Model Combination by

Decomposition and Aggregation

by
Mingyang Xu

B.E., Engineering Physics
Tsinghua University, 1997

M.E., Nuclear Science and Engineering
Tsinghua University, 2000

SUBMITTED TO THE DEPARTMENT OF NUCLEAR ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

AUGUST 2004
[September 2004]

Author..
Department of Nuclear Engineering
August 25, 2004

Certified by..
Michael W. Golay
Professor of Nuclear Engineering
Thesis Supervisor

Certified by..
George Apostolakis
Professor of Nuclear Engineering
Thesis Reader

Accepted by..
Jeffrey A. Coderre
Chair, Committee on Graduate Students

Model Combination
by Decomposition and Aggregation

by

Mingyang Xu

Submitted to the Department of Nuclear Engineering
on August 25, 2004 in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy at the Massachusetts Institute of Technology

ABSTRACT

This thesis focuses on a general problem in statistical modeling, namely model combination. It proposes a novel feature-based model combination method to improve model accuracy and reduce model uncertainty. In this method, a set of candidate models are first decomposed into a group of components or features and then components are selected and aggregated into a composite model based on data. However, in implementing this new method, some central challenges have to be addressed, which include candidate model choice, component selection, data noise modeling, model uncertainty reduction and model locality. In order to solve these problems, some new methods are put forward. In choosing candidate models, some criteria are proposed including accuracy, diversity, independence as well as completeness and then corresponding quantitative measures are designed to quantify these criteria, and finally an overall preference score is generated for each model in the pool. Principal component analysis (PCA) and independent component analysis (ICA) are applied to decompose candidate models into components and multiple linear regression is employed to aggregate components into a composite model. In order to reduce model structure uncertainty, a new concept of fuzzy variable selection is introduced to carry out component selection, which is able to combine the interpretability of classical variable selection and the stability of shrinkage estimators. In dealing with parameter estimation uncertainty, exponential power distribution is proposed to model unknown non-Gaussian noise and parametric weighted least-squares method is devise to estimate parameters in the context of non-Gaussian noise. These two methods are combined to work together to reduce model uncertainty, including both model structure uncertainty and parameter uncertainty. To handle model locality, i.e. candidate models do not work equally well over different regions, the adaptive fuzzy mixture of local ICA models is developped. Basically, it splits the entire input space into domains, build local ICA models within each sub-region and then combine them into a mixture model. Many different experiments are carried out to demonstrate the performance of this novel method. Our simulation study and comparison show that this new method meets our goals and outperforms existing methods in most situations.

Thesis Supervisor: Michael W. Golay
Title: Professor of Nuclear Engineering

Acknowledgements

Here is my opportunity to thank many people for their support and assistance during my thesis work.

First and foremost, I thank my advisor, Prof. Michael Golay, not only for giving me the opportunity to explore a wide range of very interesting topics, but also for his invaluable advice and guidance over the last four years. He has always shown me how to reach beyond what I thought was possible.

I also thank my other thesis committee members – Prof. George Apstolakis and Kent Hansen -- for their time, interest and advice.

I am deeply indebted to Prof. Richard Dudley for his invaluable instruction in probability and statistics. His classes have proved to be one of my best learning experiences at MIT. From the collaboration with him on term papers, I learned a lot about how to write a good paper.

I am grateful to all my friends, who offered so much help during these years. Especially, I would like to thank my girlfriend Ling, without whom this journey would have been very lonely.

Finally, I must thank my mother, father and sisters, who have always been there for me, for their encouragement and understanding throughout this endeavor.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem formulation

In general, models are simplified representations of physical systems, which can be in a variety of forms such as mathematical formula, computer codes, electronic systems, physical constructions, and even visual pictures or descriptions. Models are created based upon some theories and observations, thereby reflecting our knowledge about a real physical system. They can be used to test the validity of theories. Meanwhile, models can also predict behaviors of real systems.

Mathematical model is perhaps the most familiar type to most of us. Roughly speaking, it is a mathematical function mapping from some input x to output y, i.e. $f(x)$: $x \rightarrow y$. Sometimes a model itself includes some tunable parameters, for example, $f(x;\theta)$: $x \rightarrow y$ where the parameter θ is usually learned from observations. Following Bamber and van Santen [1985], a model, M, can be formalized as an ordered triple, (P, F, Q), where $P \subseteq R^k$ is the parameter space consisting of all conceivable combinations of k parameter values of the model, F is the prediction function defined on P such that $F(P) \subseteq Q$, and $Q \subseteq R^n$ is the n-dimensional response surface in its n-dimensional outcome space.

Since the structure of a model, namely, its mathematical form, is determined by the theory underlying it, it is proper to say that models are built on two bases, that is, theories and observations. As for models in a general sense, the main purpose of mathematical models is to estimate, estimate the states of a system given observations, or predict, forecast what will occur given some inputs.

From a theoretical point of view, it is advantageous to view models as sets of probabilistic, or statistical, hypotheses [Forster, 2000], that is, given inputs a statistical model produces the distribution of outputs $p_{Y|X}(y \mid x)$. In fact, more generally a model only delivers mean values $f(x)$ rather than a distribution function. For example, in physics in order to interpret models in statistical context error distributions are associated with models. In other words, in any case a deterministic equation can be regarded as mean values of the dependent response variables given a set of input variables. In fact, any measurement involves measurement errors. In this sense, a model can be reduced to a probability distribution governed by a group of parameters. In addition, the input X, which can be a vector, are often assumed to be randomly drawn from distribution $p_X(x)$. Thus, throughout this thesis, we restrict our attentions to statistical mathematical models, unless stated otherwise.

It is not surprising that for a specific real physical system, there exist many different models, which might have different parameters or even have different model structures. According to pragmatic epistemology [James, 1907], no model can ever hope to capture all relevant information, and even if such a complete model would exist, it would be too complicated to use in any practical way. Therefore we must accept the parallel existence of different models, even though they may seem contradictory. In this situation the model selection problem arises naturally, that is, which model should be chosen to use? For example, consider the following scenario: an individual is engaged in evaluating seismic risk. To this end, he needs first to predict the group motion at a specific site given an earthquake. There are actually many seismic attenuation models available, some of which are empirical models based upon historical data and others are created upon some theories in earth science like geognosy. Which model will make him more confident? The answer depends on the problems that are to be solved. The basic criterion is that the model should produce correct (or approximate) predictions (which may be tested) or problem-solutions, and be as simple as possible. Since a major purpose of models is to predict the future, it is quite natural to evaluate a model's performance using its predictive accuracy, that is, how good it can predict the upcoming events.

A model is not a complete expression of the reality, but merely reflects a perspective, an aspect of reality that may prove more or less fruitful depending on the circumstances. On the other hand, although a model does not deliver the whole truth, but it certainly conveys some information about the truth. This implies every model might be useful. Therefore, besides selecting an individual model another good solution is to combine multiple competing models.

In particular, when a new data set is available, one can also build a more accurate model using the data combination approach [Ting and Low, 1997], that is, put together the new data set with the old one and learn the model from all the available data. However, sometimes one might have no access to the old data set. Moreover, in this case we leave the model structure unchanged, and thus it does not benefit from other competing models. In contrast, for model combination approach each competing model can make contribution to the composite model. In this sense, model combination is able to combine both theories and data.

From the angle of information, one can piece together information in different competing models as well as new observations and finally come up with a better model, even without the emergence of new theories. Recent research in machine learning shows that the performance of the final model can be improved not by choosing the model structure which is expected to predict the best but by creating a model whose output is the combination of the output of models having different structures [Ting and Low, 1997]. Therefore, the remain problem is to find out an approach that makes more effecient use of information one is in possession of currently and thus improves accuracy and precision of models especially when he has multiple competing models and a new data set at hand.

This chapter is organized as follows. In section 1.2, many model selection and model combination methods will be first reviewed, and then model selection and model combination will be compared from several angles to show why model combination should be favored over model selection. In section 1.3, a new feature-based model combination method will be proposed. At last, some other important issues about this new method will be discussed in section 1.4.

1.2 Model selection and model combination

As we mentioned, in face of multiple competing models, in order to improve model accuracy one might select a single best model or combine the group of competing models. In this section, we will make a survey on methods for both model selection and model combination.

1.2.1 Model selection

The goal of model selection is to find out the best model among a set of competing models. A natural criterion for model selection is the predictive accuracy, as the major purpose of models is prediction. Before putting in practice this criterion, there are still two free factors we need make sure of, namely, error measure and generalizability estimation method. The error metrics is used to measure the discrepancy between two models especially a candidate model and the true model; generalizability estimation method is used to estimate the generalization or predictive error, in fact the expected generalization error, based upon finite samples but without knowing the true model.

1.2.1.1 Model distance measurements

In mathematics, lots of different distances arise in all sorts of contexts, and one usually requires these to be a 'metric', which is a distance function $D(\cdot, \cdot)$ that has the following desirable properties [cf. Dudley, 1989]:

1. positivity: $D(a,b) > D(a, a)=0$ when $a \neq b$
2. symmetry: $D(a,b) = D(b,a)$
3. triangle inequality: $D(a,b) \leq D(a,c)+D(c,b)$

where a and b are any two objects like vectors in a Euclidean space.

Suppose there are two distinct models $f(x)$ and $g(x)$. In the following, we will review some important distance measurements which can be employed to measure the similarity between these two functions with or without being tailored.

Minkowski-form distance

The Minkowski-form distance is defined based on L_p norm:

$$d_p(f,g) = \left(\int \left|f(x) - g(x)\right|^p dx\right)^{1/p}. \tag{1.1}$$

It has the following special case:

(1) Absolute, city block, or Manhattan distance:

$$d_1(f,g) = \int \left|f(x) - g(x)\right| dx \tag{1.2}$$

(2) Euclidean distance:

$$d_2(f,g) = \left(\int (f(x) - g(x))^2 dx\right)^{1/2} \tag{1.3}$$

(3) Maximum value distance:

$$d_\infty(f,g) = \sup_x \left|f(x) - g(x)\right| \tag{1.4}$$

The Minkowski-form distance naturally measures how a function approximates another one.

The Euclidean distance is often called mean squared error. If we assume x is uniformly distributed, e.g. $p_X(x)$=1, then

$$\begin{aligned} d_2(f,g) &= E\left[(f(x) - g(x))^2\right] = E\left[(f(x) - g(x) - E(f) + E(g) + E(f) - E(g))^2\right] \\ &= (E(f) - E(g))^2 + \mathrm{var}(f(x) - g(x)) \\ &= bias^2 + variance, \end{aligned} \tag{1.5}$$

which implies mean-squared error can be decomposed into two parts, namely bias and variance. This result will have further application in bias-variance tradeoff later on.

Another property of the Euclidean distance is that by virtue of Parseval's theorem the Euclidean distance between two functions $f(x)$ and $g(x)$, $d_2(f, g)$, is equal to the Euclidean distance between their Fourier transforms.

Non-parametric test statistics

(1) Kolmogorov-Smirnov distance

The Kolmogorov-Smirnov distance is the maximum distance between two functions over the input domain,

$$d_{KS}(f,g) = \sup_{x} |f(x) - g(x)|. \tag{1.6}$$

Note that the Kolmogorov-Smirnov distance is equivalent to the Minkowski-form distance with $p \rightarrow \infty$.

In statistics, it is often used for Kolmogorov-Smirnov tests.

(2) χ^2 (chi-square) distance

$$d_{\chi^2}(f,g) = \int \frac{(f(x) - g(x))^2}{g(x)} dx \tag{1.7}$$

Note that χ^2-distance is not symmetric, and thus not a metric.

Like Kolmogorov-Smirnov distance, it is often used for goodness-of-fit test, namely, χ^2 goodness-of-fit test.

Information-theoretic divergence

(1) Kullback-Leibler divergence [Kullback and Leibler, 1951]

In information theory, the entropy is defined as the expected value of log-likelihood, i.e. $\int \log f(x) \cdot f(x) dx$. The Kullback-Leibler divergence measures the relative entropy between two probability distributions,

$$d_{KL}(f,g) = \int \log \frac{f(x)}{g(x)} f(x) dx \tag{1.8}$$

The Kullback-Leibler distance was introduced in statistics as early as in 1951, and its use in hypothesis testing and model evaluation was propagated strongly by Kullback [1959].

It measures the degree of approximation or similarity between two probability distributions. However, in reality it is easy to extend it to general functions by normalizing them as long as $\int f(x)dx$ and $\int g(x)dx$ exist.

Nevertheless, it is noteworthy that the KL discrepancy is not a metric because it does not satisfy either symmetry or triangular inequality although a nonnegative distance.

(2) Jeffrey divergence

The Jeffrey divergence [Jeffreys, 1946] is empirically derived from the K-L divergence such that it is symmetric, stable and robust with respect to noise. It can be written as

$$d_J(f,g) = \int \left[f(x)\log\frac{2f(x)}{f(x)+g(x)} + g(x)\log\frac{2g(x)}{f(x)+g(x)} \right] dx \,. \tag{1.9}$$

It is also known as Jensen-Shannon divergence [Lin, 1991].

Empirical study shows that in spite of difference these various discrepancy measurements generally give consistent results.

1.2.1.2 Generalizabililty estimation methods

Generalizability is a mean discrepancy between the true model and the best-fitting member of the model class of interest, averaged across all possible data that could be observed under the true model. The basic tenet of model selection is that among a set of competing model classes, one should select the one that optimizes generalizability. However, generalizability is not directly observable and instead one must estimate the measure from a data sample by considering the characteristics of the model class under investigation.

Before we turn to generalizability estimation methods, let's first introduce Occam's razor and model complexity, which are crucial to most model selection criteria.

Occam's razor

Occam's razor, a principle also called principle of parsimony, can be dated back to the mediaeval philosopher William of Occam. It states that of two theories that describe the data equally well the simpler one should be preferred. To date, it has been underlying all scientific modeling and theory building. In any given model, Occam's razor helps us to "shave off" those concepts, variables or constructs that are not really needed to explain the phenomenon. By doing that, developing the model will become much easier, and there is less chance of introducing inconsistencies, ambiguities and redundancies.

For a given set of observations or data, there are always an infinite number of possible models that might explain those data with the same accuracy. Occam's razor admonishes us to select the simplest one among the set of otherwise equivalent competing models. Thus, Occam's razor is realized such that parsimony or simplicity is somehow balanced against goodness-of-fit, which refers to how well a model fits the particular data set.

Based upon this principle, many model selection approaches have been proposed and developed. All these methods, which overlap with one another, provide an implementation of Occam's razor in one way or another.

According to Occam's razor, simplicity is another desired property besides accuracy. Thus, defining and measuring model complexity has become an integral part in most model selection criteria.

Measures of Model complexity

Another basic issue is the complexity of a model. Model complexity is conceptualized as the capacity of a model to fit any conceivable data set. Alternatively, Myung and Pitt [1997] define model complexity as "the flexibility inherent in a model that enables it to fit diverse patterns of data".

In order to measure model complexity, Myung and Pitt [1997] have suggested three factors that affect a model's complexity, namely, the number of parameters, the parameter space, and the functional form of a model. First, the degree of freedom quantified by the number of unknown parameters in a model is a classical measure of model complexity. For example, in the multiple polynomial regression analysis it is obvious that the more items included, the more complex a regression model. Second, as for the parameter space, there is no doubt that the wider the space, the more data patterns a model can fit. It is not hard to verify this point using the example of polynomial regression. Finally, models with different functional forms have different ability to fit arbitrary patterns of data. For instance, generally nonlinear functions are more complex than linear functions. Again in the multiple regression example, if cosine series instead of polynomial terms are used as regressors, the regression model's capability of diverse fitting data patterns also varies.

Similarly, Brooks and Tobias [1996] defines the overall complexity of a model as a combination of three elements: size (the number of components), connectedness (which components are related), and calculational complexity (the complexity of the calculations determines the relationships). Based on their arguments, they also propose a graph theory measure of model complexity.

From the point view of coding theory, model complexity can be represented by its description length, which is formulated in Kolmogorov complexity. Kolmogorov complexity is a modern notion of randomness dealing with the quantity of information in individual objects; that is, pointwise randomness rather than average randomness as produced by a random source [Li and Vitanyi, 1997]. It was proposed by A.N. Kolmogorov in 1965 to quantify the randomness of individual objects in an objective and absolute manner. Kolmogorov complexity is also known variously as algorithmic complexity, Turing complexity and others. It is defined as the minimum number of bits into which a string can be compressed without losing information. The Kolmogorov complexity $C(s)$ of any arbitrary string $s \in \{0,1\}^n$ is defined as the length of the shortest computer program s^* that can produce this string on the Universal Turing Machine (UTM), which is not a real computer but an imaginary reference machine [Grunwald, 2000]. Generally, Kolmogorov complexity is not computable because we cannot compute the output of every program.

Since a model can be geometrically represented multidimensional response surface, another natural way to measure model complexity is to use the roughness of a fitted curve. It does not distinguish the contribution to the model complexity from different factors like degree of freedom and functional form. However, it might be able to reflect model complexity resulting from other factor than those discussed above. On the other hand, this measure has different values for models with the same model structure but various model parameters.

Bias-variance tradeoff

How model complexity affects a model's generalizablity can better understand in light of a well-known bias-variance tradeoff [see e.g. Geman *et al.*, 1992].

When a model is too complex for the amount of training data at hand, it learns parts of the noise as well as the true model structure, resulting in poor generalizability. The model ends up with being very sensitive to the training samples we use and has a lot of variance across training samples of a fixed size. This is often called overfitting.

In contrast, when our model is not complex enough, it cannot capture the structure in the training data. Therefore, no matter how much data we feed there will be always some error between the true model and our approximating model. In other words, so trained model has a lot of bias. This is usually called underfitting.

Figure 1.1 illustrates the difference between underfitted and overfitted models in regression.

Generally, a fitted model starts with underfitting and ends up with overfitting with the model complexity growing. In the example of multiple linear regression, the prediction error decreases at first and goes up at last by adding more predictors, which is shown in Figure 1.2. The balance point between underfitting and overfitting is considered optimal. To understand this, it is helpful to take a look at the bias-variance tradeoff.

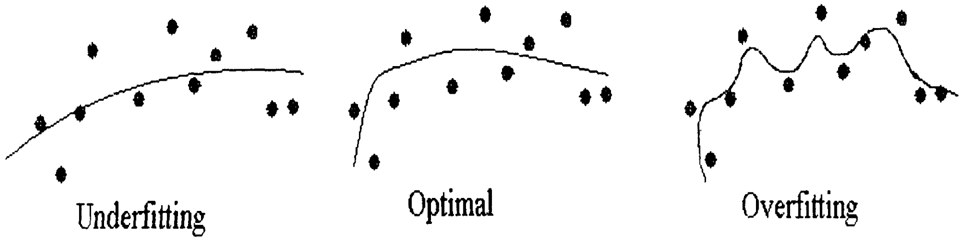


Figure 1.1 Illustration of the difference between underfitted and overfitted models

As we mentioned earlier, the predictive error can be decomposed to bias and variance. In general, using more complex models can reduce the model bias, the first term, or in other words achieve better fit, but in the meantime model variance is increased because the sample size gets smaller relative to the number of model parameters to be estimated. In the case of underfitting, the bias in parameter estimation is generally substantial while the variance is underestimated. As for overfitting, the parameter estimation is usually free of bias but have large variance. Figure 1.2 may give us an intuitive sense of the relationship between underfitting and overfitting as well as how the model complexity affects the generalization error.

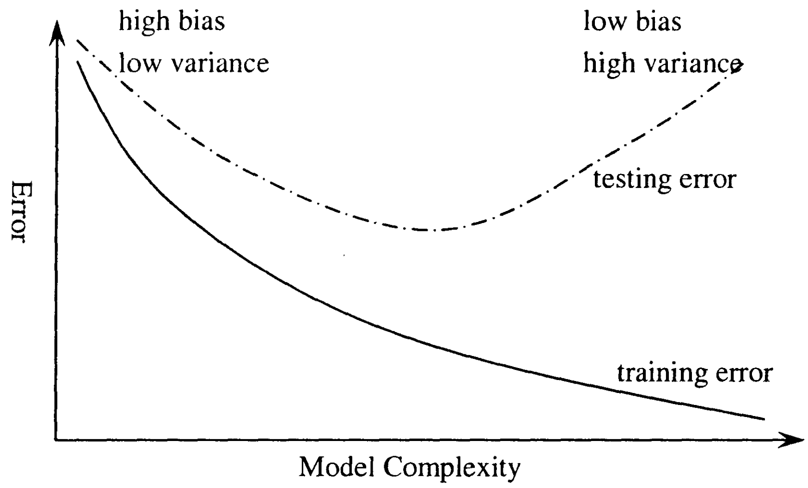


Figure 1.2 Bias-Variance tradeoff

In view of this trade-off, we need to find out a balance point in this tradeoff, which is considered optimal, thereby minimizing expected predictive squared error in the future. From another angle, this is also a tradeoff between goodness-of-fit and model complexity.

Generalizability, or predictive accuracy, refers to a model's ability to predict future, unseen yet, data samples generated from the same underlying process. Mathematically, generalizability is the mean discrepancy between a candidate model and the true model. Thus, the purpose of generalizability estimation methods is to estimate the generalization or prediction error based upon empirical training errors. Since empirical errors are based upon finite samples, thus integrals in distances formula should be replaced by sum.

To date, many different types of generalizability estimates have been proposed, and in the following we will briefly review some of them.

(1) Resampling method:

To obtain robust estimators with finite samples, statisticians start to resort to data resampling methods, such as cross-validation and bootstrap. Basically, resampling methods mimic the future data by resampling technique to estimate the generalization error.

Cross-validation

Cross-validation [see Stone, 1974] is a natural method for estimating generalization error based on resampling a limited pool of data, and has been widely used for model selection. There are lots of variants of cross-validation methods, including leave-one-out cross-validation or jack-knife, generalized cross-validation and K-fold cross-validation.

The basic idea of cross-validation is to test a trained model using samples different from those for training. In cross-validation, the original data set is split into two parts, namely, the calibration samples and the validation samples. The model of interest is fitted to the calibration samples and tested on the validation samples with the estimated parameters. The test error serves as the generalization error. Some researchers suggest using 2/3 for training and 1/3 for testing.

In order to improve the efficiency of the use of data, rather than setting aside a separate validation set, one might leave out part of the original data, train on the rest, measure errors on the part left out, and then repeat leaving out a different bunch of data. If we break our data into K equal groups, and cycle through them all, leaving one out at a time, this is known as K-fold cross-validation. Our final generalization error is equal to the average of all validation errors.

Moreover, if we leave out only one observation at a time or equivalently make K equal to the sample size, this leads to leave-one-out (LOO) cross-validation or jack-knife. The estimation of generalization error is similar to K-fold cross-validation.

In practice, cross-validation can be very time consuming. However, in some special situations there are some efficient tricks that can save one lots of work over brute-force retraining on all K possible LOO datasets. In the case of multiple linear regression, that is, $Y=X^T\beta$, there is a simple expression of LOO, that is, the PRESS (PREdiction Sum of Squares) proposed by Allen [1974], which is defined as

$$PRESS = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{-i})^2 , \tag{1.10}$$

where y_i is the i-th data point and $\hat{y}_i^{-i}$ is the prediction corresponding to y_i by the fitted model with the i-th pair (x_i, y_i) left out. This is actually the sample mean of prediction errors for 1-fold cross-validation. Furthermore, one can show that

$$y_i - \hat{y}_i^{-i} = \frac{y_i - \hat{y}_i}{1 - h_{ii}} \tag{1.11}$$

where h_{ii} denotes the ii-th element of the "hat" matrix $H=x(x^Tx)^{-1}x^T$ [Cook and Weisberg, 1982]. Then

$$PRESS = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_{ii}}\right)^2 \tag{1.12}$$

With the above formula, PRESS can be calculated without fitting the model n times, each time deleting one of the n cases. If we replace h_{ii} by the average of H's diagonal entries $\bar{h} = \frac{1}{n}\sum_{i=1}^{n} h_{ii}$, we obtain the error prediction as

$$GCV = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - \bar{h}}\right)^2 = \frac{SSE}{n(1-\bar{h})^2} = \frac{n \cdot SSE}{(n - trace(H))^2}, \tag{1.13}$$

which is termed Generalized Cross-Validation (GCV) [see Golub,Heath, and Wahba, 1979].

When we do not have enough data to make separate training and testing sets, another resampling method which can make efficient use of limited use can be applied instead.

Bootstrapping

The bootstrap method, which can be first dated back to Efron [1979], has been become a popular and practical tool of inference and gained wide application in estimating standard errors, confidence intervals, biases, and prediction errors. Roughly speaking, bootstrapping is a statistical resampling technique based on randomly sampling from the empirical distribution with replacement. Given an original set of independent and identically distributed (i.i.d.) observations x_i , i=1 ...,n , the unknown cumulative distribution function (CDF) $F_x(x)$ that generates the observed data can be first estimated by putting mass $1/n$ at each data points. Then bootstrap samples, denoted x^b, are repeatedly drawn from the original sample x according to the empirical distribution, with the number of bootstrap replications $N \geq n$.

The fundament of the bootstrap is the "plug-in" principle [see Efron and Tibshirani, 1993], which allows for the estimation of a statistics according to an empirical distribution, such as estimation of median values or confidence intervals. For the purpose of model selection, bootstrapping is used to estimate the generalization error. To this end, Efron defines the bootstrap estimator of the generalization error (or prediction error) [Efron, 1983 and Efron, 1986] as

$$\hat{e}_{gen} = e_{app} + \omega, \tag{1.14}$$

where $\hat{e}_{gen}$ denotes the bootstrap generalization error, e_{app} is the apparent error and ω, called optimism, is a correction term for the difference between the training error and the generalization error. e_{app} is the training error of the model $f_b(x_i)$ learned on the original sample x_i, y_i, i=1 ...,n , that is,

$$e_{app} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_b(x_i))^2 . \tag{1.15}$$

The optimism is intended to approximate the difference of errors obtained on the finite sample x and an unknown infinite ideal sample. It is estimated by bootstrap method. At first, one draws randomly N samples with replacement from the original dataset. These new samples form a new learning set with the same size as the original one. The original training set serves as the validation set. This procedure is called re-sampling. After training model on the bootstrap replications and testing it using the original dataset, we obtain the difference between training error and testing error as optimism, denoted as ω_k, a measure of performance degradation (for the same model) between a learning and a validation set,

$$\omega_k = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_b(x_i))^2 - \frac{1}{n}\sum_{i=1}^{n}(y_i^b - f_b(x_i^b))^2 . \tag{1.16}$$

This process is repeated K times with K as large as possible and we obtain the average optimism as ω

$$\omega = \frac{1}{K}\sum_{k=1}^{K}\omega_k \ . \tag{1.17}$$

Note that both the bootstrap learning set and the evaluation set has observations in common, and thus bootstrap method still subjects to underestimation of the error due to overfitting.

A particular case of bootstrap method is the .632+ bootstrap method [see Efron and Tibshirani, 1997], in which the generalization error is estiamted as

$$\hat{e}_{gen} = 0.368 e_{app} + 0.632\omega', \tag{1.18}$$

where the apparent error rate e_{app} remains the same as above but the optimism ω' is estimated only on the data that are not selected during the re-sampling. This is a weighted average of in-sample error and out-of-sample error.

Resampling methods do not take model complexity into account explicitly. A common drawback of resampling mehtods is their high computational load. In addition to resampling methods, there are other analytic approaches, which are computationaly more efficient.

(2) Statistical measurements

In statistics, R^2 is often used to measure the proportion of variance of a given data set explained by a model. For the purpose of model selection, R^2 is adjusted by incorporating a penalty for additional predictions, attempting to adjust R^2 for capitalization on chance in a sample data and give an estimate of R^2 in the population. In mathematics, it is written as

$$AdjR^2 = 1 - \frac{\hat{\sigma}^2}{S_y^2} = 1 - \frac{SSE/(n-D)}{S_y^2} \tag{1.19}$$

where SSE is the sum of squared error, i.e. $\sum_{i=1}^{n}(y_i - f(x_i))^2$, n is the sample size, D is the model dimension, S_y^2 is the sample variance of the response variable y, and $\hat{\sigma}^2$ is called unbiased estimation of variance

$$\hat{\sigma}^2 = \frac{SSE}{n-D}. \tag{1.20}$$

Mallows' C_p [Mallows, 1973] is concerned with total mean squared error of fitted values, which is also closely related to adjusted R^2 and Akaike's AIC. Mallows' C_p criterion is to minimize

$$C_p = \frac{SSE}{\hat{\sigma}^2_{full}} - [n - 2D] \tag{1.21}$$

where $\hat{\sigma}^2_{full}$ is estimated from the model with all the predictor variables and used to estimate the true variance. If a model is good, $C_p \approx D$, while a model with bad fit will have C_p much bigger than D. In general, C_p is a good indicator for determining when a model is underfitted.

(3) penalty-based methods

A general form for this class of methods can be expressed as

(generalizability) = (goodness-of-fit) + λ·(model complexity)

which formalizes the principle of Occam's razor. In other words, they penalized a model's empirical accuracy by its complexity. Almost all information theoretic criteria can be included in this class.

As we discussed earlier, Kullback-Leibler divergence is a natural distance to measure how a model approximate the true model. The goal of Akaike's Information Criterion (AIC) [Akaike, 1973] is to minimize the expected Kullback-Leibler distance. By penalizing empirical K-L discrepancy with model complexity, AIC provides an asymptotic estimate of the mean Kullback-Leibler divergence between a fitted model and the true model. The mathematical expression of AIC is quite simple and can be viewed as an extension of general maximum likelihood with a complexity penalty term

$$AIC = -2\log L(\hat{\theta}_k \mid x) + 2k \ , \tag{1.22}$$

where $\log L(\hat{\theta}_k \mid x)$ is the maximum log-likelihood of a model with k model parameters based on data $x = (x_1, \ldots, x_n)$, that is,

$$\log L(\hat{\theta}_k \mid x) = \sum_{i=1}^{n} \log p_{x|\theta}(x_i \mid \hat{\theta}_k) \ , \tag{1.23}$$

and $\hat{\theta}_k$ is the maximum likelihood estimate of that model. The criterion chooses the model with the smallest value of AIC.

Stone [1977] proves that AIC is asymptotically equivalent to LOO cross-validation. Furthermore, in regression variable selection, AIC is essentially equivalent to Mallow's C_p [Shibata,1981] as well as to cross-validation and generalized cross-validation asymptotically [Li, 1987].

Several other variations on the AIC exist. In AIC, it is assumed that models are faithful, i.e. the learning target can be expressed by the model [Murata et al., 1994]. Takeuchi [1976] extended AIC to be applicable to unfaithful models and proposed TIC (Takeuchi's Information Criterion) as a more accurate estimate than AIC, which is expressed as

$$TIC = -2\log L(\hat{\theta}_k \mid x) + 2trace\left(I(\theta_0)J(\theta_0)^{-1}\right) \tag{1.24}$$

where J and I are the expected values of kxk matrices based, respectively, upon first and second partial derivatives of $\log p_{X|\theta}(x \mid \theta)$ with respect to θ, i.e.,

$$\begin{aligned} J(\theta_0)_{ij} &= E_{p_X}\left[\frac{\partial \log p_{X|\theta}(x \mid \theta_0)}{\partial \theta_{0i}} \frac{\partial \log p_{X|\theta}(x \mid \theta_0)}{\partial \theta_{0j}}\right], \\ I(\theta_0)_{ij} &= E_{p_X}\left[\frac{\partial^2 \log p_{X|\theta}(x \mid \theta_0)}{\partial \theta_{0i} \partial \theta_{0j}}\right], \text{where } i,j = 1,\ldots,k \end{aligned} \tag{1.25}$$

evaluated at the true model parameter $\theta_0 = (\theta_{01}\, \theta_{02} \ldots \theta_{0r}) \in \Theta$,the parameter space, and the expectations are with respect to the true distribution $p_X(x)$.

Note that TIC does not assume the true model is in the set of candidate models and in fact when the true data generating model is a member of the model class under consideration, a candidate model tends to the true model $p_X(x) \approx p_{X|\theta}(x \mid \theta)$ and $\hat{\theta} \approx \theta_0$, and then $J(\theta_0)$=- $I(\theta_0)$, the Fisher information matrix evaluated at the maximum likelihood estimate $\hat{\theta}$,. At last, -$J(\theta_0)I(\theta_0)^{-1}$ becomes a kxk identity matrix, whose trace is exactly –k, and thus TIC simplifies to AIC.

Murata *et al.* [1994] generalized the loss function of TIC, and proposed the network information criterion (NIC). In NIC it is assumed that the quasi-optimal estimator minimizing the empirical error, say the maximum likelihood estimator when the log loss is adopted as the loss function, has been exactly obtained.

To improve the performance of AIC under the small-sample situation, Hurvich and Tsai [1989] propose a small sample version of AIC, namely AIC$_C$. AIC and AIC$_c$ differ in that the AIC$_c$ contains correction for finite sample bias, although both provide asymptotically unbiased estimates of expected KL distance. Similar to AIC, AIC$_c$ takes the form

$$AIC_c = -2\log L(\hat{\theta}_k \mid x) + 2k + \frac{2k(k+1)}{n-k-1} \tag{1.26}$$

where n is the sample size.

Note that when n is very large relative to k, AIC$_c$ reduces to AIC. When the number of free parameters is relatively large compared to sample size, Burnham and Anderson [2002] strongly recommend AIC$_c$.

Since Akaike's seminal paper, some other information criteria were proposed later on, including Bayesian Information Criterion (BIC) and Minimum Description Length (MDL). BIC has a similar form to AIC although derived from a very different prospective, a Bayesian framework. In fact, this is not surprising if we notice the close connection between information and likelihood [Kullback, 1959]. BIC was first derived by Schwarz in a Bayesian context with a uniform prior probability on each competing model and priors with everywhere positive densities on the model parameters θin each model and choosing the model dimensionality with the highest posterior probability leads to the BIC criterion of Schwarz [1978],

$$BIC = -2\log L(\hat{\theta}_k \mid x) + k\log n\ . \tag{1.27}$$

In comparison to AIC, BIC has a different model complexity-based penalty term, which depends on both model dimensionality, the number of parameters, and sample size.

Later on we will show that BIC can also be derived using Laplace's method of approximation.

AIC was proven to be inconsistent, e.g. by Shibata [1976] and Woodroofe [1982] for i.i.d data, while BIC was shown to be consistent by Woodroofe [1982]. However, inconsistency may not affect the use of AIC for the purpose of prediction, and indeed there is some evidence that in certain predictive context AIC is asymptotically optimal [e.g. see Shibata, 1981 and Geisser and Eddy, 1979].

In the above information-theoretic criteria, the number of parameters (k) and the sample size (n) are the only relevant factors of complexity. However, they neglected another important facet of model complexity, namely, the functional form of the model expression [Myung and Pitt, 1997], which refers to how model parameters are used in the model formulation. For example, two models, $y=ax+b$ and $y=ax^b$, differ in functional form and thus in model complexity although have the same number of parameters.

Jorma Rissanen specially addresses a model's simplicity develops the idea of minimizing the generalization error of a model by penalizing it with its description length, which estimates the Kolmogorov Complexity by replacing algorithmic complexity with stochastic complexity (the shortest obtainable description x by a model class M) [Rissanen, 1978]. In Minimum Description Length (MDL), both models and data are viewed as codes that can be compressed, and correspondingly the objective of model selection is to choose the model that permits the greatest compression of the data in its description.

In MDL, the model complexity is penalized not only according to the number of parameters but also both parameters and precision, and the MDL takes the familiar form of a penalized likelihood [Rissanen, 1996]

$$MDL = -\log L(\hat{\theta}_k \mid x) + (k/2)\log(n/2\pi) + \log\int\sqrt{\left|\det(I(\theta))\right|}d\theta + o(1)\,, \tag{1.28}$$

in which $|I(\theta)|$ is the determinant of the Fisher information matrix and $o(1)$ becomes negligible for n large.

Compared to AIC and BIC, the model complexity term, stochastic complexity, in MDL is

$$SC = \frac{k}{2}\ln\frac{n}{2\pi} + \ln\int\sqrt{|\det I(\theta)|} \tag{1.29}$$

which is viewed as the combination of complexity due to the number of parameters (k) and complexity due to the functional form of the model equation reflected through $I(\theta)$.

In statistics, the term

$$\tau(\theta) = \sqrt{|\det(I(\theta))|} \tag{1.30}$$

is called the Rao measure [Amari, 1985]and the Riemannian volume of the parameter manifold can be obtained by integrating the Rao measure over the parameter space [see Myung *et al.*, 1999]

$$V_M = \int \sqrt{|\det(I(\theta))|}\, d\theta\,. \tag{1.31}$$

Therefore, corresponding to the measure of model complexity, this term reflects the model complexity due to parameter space.

The second term is often difficult or impossible to compute, but a reasonable practical version views stochastic complexity as a two-stage description of the data, consisting of the encoding of a model and the encoding of the data using that models [Grunwald, 2000]. This leads to an approximation of the MDL as

$$-\log L(\hat{\theta}_k \mid x) + (k/2)\log(n) \tag{1.32}$$

which is identical to one half of the BIC.

In addition to the above model complexity-based penalties, one can also penalize the training error by roughness of a fitted curve. In fact, roughness is directly connected to model complexity, although it does not explicitly consider the number of parameters and functional form of model equations. Intuitively, the more complex is a model, the rougher it can be. This idea can be traced back to spline smoothing [e.g. see Reinsch, 1967 and Silverman, 1985].

The roughness measure can be defined based on local variation, which can be quantified by the first, second, and so forth derivative. In order to explicate the main ideas the integrated squared second derivative is most convenient, that is, the roughness penalty $\int f''(x)^2 dx$ is often used to quantify local variation. Using this measure, define the generalization error

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx \tag{1.33}$$

where λ denotes a smoothing parameter, specifying the degree of preference of smoothness.

In fact, the model complexity ascribed to the functional form is in terms of not only how parameters are combined but also how the inputs are used in a model. Since the Riemannian volume is appropriate for measuring the complexity of functional forms as shown in [Myung et al., 1999], we can use it to measure part of the model complexity in terms of inputs

$$V_I = \int \sqrt{\left|\det(d^2 f(x)/dx^2)\right|}\, d\theta \tag{1.34}$$

or

$$V_I = \int \sqrt{\left|\det((df(x)/dx)^2)\right|}\, d\theta \tag{1.35}$$

where $df(x)/dx$ and $d^2f(x)/dx^2$ denotes Jacobian vector and Hessian matrix, respectively. Note that V_I reflects roughness, similar to roughness penalty in equation (1.33).

Following this line of reasoning, a more comprehensive version measure of model complexity of model $f(x;\theta)$ can be proposed as

$$SC = \frac{k}{2}\ln\frac{n}{2\pi} + V_M + V_I\,. \tag{1.36}$$

In the above, we reviewed many different generalizability estimation methods. Most of them converge to the true model with the sample size growing, mainly due to the law of large numbers which forces the statistics of samples to converge with the statistics of the source. Furthermore, empirical study shows that in spite of difference they usually produce consistent results, especially in asymptotic cases.

Any model performance evaluation includes two elements, namely, distance function and generalizability estimation method. By varying discrepancy measurements and generalization methods, we can construct many different model selection criteria. Although they are introduced specifically for the purpose of model selection, in fact they can be employed as general model performance evaluation methods, i.e. estimate the generalization or prediction error. For example, they can be used to evaluate the performance of a new composite model.

1.2.2 Model combination

Besides model selection, another strategy commonly used to arrive at improved model performance is to combine multiple competing models. As opposed to model selection, which uses training data merely to select a single best model among a group of competing models, model combination produces a composite model based on original models. Recent research in machine learning shows that the performance of the final model can be improved not by choosing the model structure which is expected to predict the best but by creating a model whose output is the combination of the output of models having different structures.

Combining multiple candidate models can be implemented by a variety oftechniques. In the following we will briefly discuss some popular methods.

Majority voting is a weighting scheme, but unlike the weighted average the one receiving the maximum votes wins. The basic idea is to improve the probability of making correct decision by combining decisions from multiple experts. The simple majority voting counts individual votes supporting each decision, and the one receiving majority votes ends up as the final decision.

If we take into account the different competences of individual experts, this leads us to the weighted majority voting in which voting weights are decided according to one's competence. If we denote as d_{ik} the kth expert's decision to supporting the ith decision, then the total support that the ith decision receives takes the form of

$$d_i = \sum_k w_k \cdot d_{ik} \text{ , with } d_{ik}\text{=0 or 1,} \quad (1.37)$$

where w_k refers to the weights of individual experts.

The final decision is therefore the one that receives the most support

$$j = \arg\max_{i=1,\dots,n} d_i \, . \quad (1.38)$$

Majority vote is originally an effective strategy in making decision, and recently it has been introduced to pattern recognition, in particular in combining multiple classifiers or in other words classifier fusion [e.g. see Kuncheva *et al.*, 2001]. Roughly speaking, pattern recognition is to represent objects by a finite number of real-valued measurements called features, and then classify objects of interest into one of a number of categories or classes. Thus, the classification problem is to assign an input, a feature vector, to one of the given classes. The gain of accuracy by majority voting in classification can be exemplified by the following simple example which is given in [Dietterich, 2000].

If we have a dichotomic classification problem and assume n independent classifiers have the same probability p of being correct, the overall error of the resulting majority ensemble can be given by the area under the binomial distribution where more than $n/2$ classifiers are wrong:

$$P_{error} = \sum_{i=[n/2]}^{n} \binom{n}{i} p^{i}(1-p)^{n-i} . \tag{1.39}$$

Condorcet [1785] is usually credited with first recognizing this fact and the Condorcet Jury Theorem attributed to him proved that the judgment of a group is superior to those of individuals provided that the individuals have reasonable competence.

If weighted majority voting is applied, weights of individual classifiers can be determined according to their training accuracy.

The application of majority voting to the cases where there are a finite number of different possible discrete outputs is obvious. Actually, one might modify it slightly to make it suitable for continuous-valued or infinite-value cases. One possible solution is to choose the center of outputs of multiple models as the combined output, i.e. the point that has the minimum total distance to all outputs. If we apply the Euclidean distance, mathematically we have

$$y_c = \min_{y} \|y - y_i\|^2 , \tag{1.40}$$

where y_c is the combined estimate and y_i's are individual estimates.

The solution can be easily obtained as

$$y_c = \frac{1}{m} \sum_{i=1}^{m} y_i , \tag{1.41}$$

where m is the number of competing estimators.

This is exactly the simple average or unweighted average of all estimates. Averaging is a classical way to reduce variance. For example, consider two estimators of an unknown parameter θ, say θ_1 and θ_2, which are unbiased and having the same variance , i.e. E(θ_1)= E(θ_2)= θ and $var(\theta_1)=var(\theta_2)=v$.

We can build a combined estimator of θ using unweighted average as $\theta_c=(\theta_1+\theta_2)/2$, which remains unbiased and has variance

$$\mathrm{var}(\theta_c)=v/2+\mathrm{cov}(\theta_1,\theta_2)/2. \qquad (1.42)$$

It is easy to see that as long as $\mathrm{cov}(\theta_1,\theta_2) < v$ or equivalently the correlation coefficient ρ <1, the composite estimator has a reduced variance. In the case of physical models for the same system of interest, $\mathrm{var}(\theta_1)\approx\mathrm{var}(\theta_2)$ holds in general and the correlation coefficient ρ is also high.

This argument can be easily extended to the case of more than two candidate models.

Simple average method is usually applied where there is no or very few new data is available and none of the competing models dominates others. However, when we learn that the performances of models might be significantly different, we have no reason to assign uniform weights to each model indiscriminatingly; rather, we would like to assign higher weights to some models and lower to others, which leads to the weighted average.

Mathematically, weighted average is the linear combination of a number of candidate models with a normalization constraint on weights, i.e.

$$f_C(x)=\sum_{j=1}^{m} w_j f_j(x), \text{ subject to } \sum_{j=1}^{m} w_j = 1, w_j \geq 0, j = 1,\dots,m \qquad (1.43)$$

Choosing MSE as the error measurement, we obtain the empirical mean-squared error of the combined model given observations $D=\{(x_i, y_i)\}_{i=1}^{n}$ as

$$MSE=\sum_{i=1}^{n}\left(y_i-\sum_{j=1}^{m} w_j f_j(x_i)\right)^2. \qquad (1.44)$$

It becomes a constrained least-squares problem. The above MSE can be minimized under the linear constraint using the Lagrangian method

$$L=\sum_{i=1}^{n}\left(y_i-\sum_{j=1}^{m} w_j f_j(x_i)\right)^2+\lambda\left(\sum_{j=1}^{m} w_j-1\right), \qquad (1.45)$$

where λ is a Lagrangian multiplier.

Optimum is achieved by equating the gradient with respect to w_j's to 0. For convenience, let's rewrite the above equation in matrix notions

$$L = (Fw - y)^T (Fw - y) + \lambda(u^T w - 1), \tag{1.46}$$

where $w = [w_1, ..., w_m]^T, y = [y_1, ..., y_n]^T, u = [1, ..., 1]^T$ is an m-dimensional vector of ones, and $F = \begin{bmatrix} f_1(x_1) & \cdots & f_m(x_1) \\ \vdots & \ddots & \vdots \\ f_1(x_n) & \cdots & f_m(x_n) \end{bmatrix}$.

Thus, we have

$$\begin{cases} 2F^T(Fw - y) + \lambda u = 0 \\ u^T w = 1 \end{cases} \text{ or } \begin{bmatrix} F^T F & u \\ u^T & 0 \end{bmatrix} \begin{bmatrix} w \\ \lambda / 2 \end{bmatrix} = \begin{bmatrix} F^T y \\ 1 \end{bmatrix}. \tag{1.47}$$

After some matrix manipulations, we obtain

$$\begin{bmatrix} w \\ \lambda / 2 \end{bmatrix} = \begin{bmatrix} F^T F & u \\ u^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} F^T y \\ 1 \end{bmatrix}. \tag{1.48}$$

In fact, it can also be solved using an iterative procedure by taking advantage of $\lambda = 2(y^T - w^T F^T)Fw$ and $w = (2F^T F)^{-1}(2F^T y - \lambda u)$.

Note that if the individual competing models are unbiased, so is the combined model, which is the main reason for employing the constraint $\sum_{j=1}^{m} w_j = 1$. Meanwhile, this method produces interpretable composite models.

In practice, the "optimal" weights derived above are not really optimal, because weights are learned from a limited number of data points. Thus, in reality other weighting strategies based upon predictive accuracy measurements are applied instead. Consistent with one's intuition, models of higher predictive accuracy are assigned higher weights. For example, performance of each model can be evaluated using Akaike's information criterion (AIC) and assigned different weights based on their AIC value [Burnham and Anderson, 2002], for instance,

$$w_k = \frac{\exp(-\frac{1}{2}AIC_k)}{\sum_{j=1}^{m}\exp(-\frac{1}{2}AIC_j)}, \tag{1.49}$$

which is called Akaike weights.

Certainly, other predictive accuracy measurements like cross-validation, Mallow's C_p, and MDL can be employed in place of AIC to generate predictive performance-based weights. Since all weights lie in between 0 and 1, it is obvious that within any certain sub-region the composite model is at best as good as the best model within that sub-region. To overcome such weakness, a possible way is to remove the constraints on weights and make them any real value, that is,

$$f_C(x) = \sum_{j=1}^{m} w_j f_j(x), w_j \in R, j = 1,...,m\,, \tag{1.50}$$

which is called linear combination of experts or models in some literature. Although combination of models arrives at improved accuracy compared to weighted average, the coefficients w_j lose their meaning in weighted average scheme, thereby making the composite model less interpretable.

The coefficients w_j can be learned from data using various algorithms, for example, regressing a data set y on the m competing models $f_j(x)$ using ordinary least squares, in which the training set is made by $D = \{f_i, y_i\}_{i=1}^{n}$ where f_i, i=1,…,n, is an m-dimensional vector, i.e. f_i=[$f_1(x_i)$,…, $f_m(x_i)$]^T. However, this simple least-squares approach might not produce satisfactory results especially when the training sample size n is small, because it learns from a limited number of data by minimizing squared-error rather than prediction error, which makes parameters data-specific and thus suffer from high variance and instability. In order to address these problems, a variety of approaches have been brought forward up to now, for example, ridge regression [Hoerl and Kennard, 1970 and Tikhonov and Arsenin, 1977], Bayesian regression [Lindley and Smith, 1972], M-estimate [Huber, 1964], weighted least-squares regression [Carroll and Ruppert, 1988], and so on. We are not going to review these regression techniques, but instead we will go on introducing some general model combination techniques, which, however, can also be employed to create the combined model in equation (1.50).

Bagging

"Bagging" is the abbreviation of "Bootstrap Aggregating". The idea behind it is quite straightforward. It is well known that in the bias-variance tradeoff any reduction in the prediction variance is usually along with an increase in the expected bias for the future predictions. Breiman [1996b] introduced bagging to reduce the prediction variance without increasing the prediction bias. Basically, the bagging procedure is to learn multiple models from bootstrap samples of the original data set, and combine them with uniform weight. Individual models are trained on slightly different samples of the available data set, which are generated by bootstrapping The generalization performance obtained by the "average model" is usually better than the one that would result from training a single model on the full data set.

Certainly, bagging method is not restricted to regression models, but suitable to learn any parametric model like $h(x;\theta)$. Instead of making inference from a single fitted model, a set of repeated bootstrap replicates are drawn from the original data set with replacement and then a model $h(x;\theta)$ is trained based upon for each bootstrap replication with parameter θ_k , and finally the predictions are averaged over all of the fitted models to obtain the bagged prediction,

$$f_{Bag}(t) = \frac{1}{m}\sum_{j=1}^{m} f_k(x;\theta_j)\,. \tag{1.51}$$

According to Breiman [1996b], bagging works well for unstable modeling procedures with respect to the data, i.e., small changes in the data can result in significant change in model estimation, but it leads to no substantial improvements in linear regression, which is a stable procedure. Intuitively, bagging uses bootstrap replicates to mimic instability caused by data and tries to avoid it by averaging. If perturbing the learning set can cause significant changes in the predictor constructed, then bagging can improve accuracy. On the other hand it can slightly degrade the performance of stable procedures. There is a cross-over point between instability and stability at which bagging stops improving. In addition, according to his experiments, Breiman [1996b] also suggested the number of bootstrap replicates m to be about 50.

Note that bagging is intended to combine models learned from different batches of data using the same learning algorithm and thus result in models with the same model structure but different parameters. However, it can be easily generalized it for combining a given class of competing models, by making it work together a model selection procedure, which is unstable. Model selection procedure is repeated many times on bootstrap replications and the final model is obtained by averaging over all the selected models. Since in each model selection procedure the selected model might be different due to instability, the resultant model is actually a combined model in the form

$$f_c(x) = \sum_{j=1}^{m} w_j f_j(x)\,, \qquad (1.52)$$

where the weight w_j is proportional to the times that a specific model $f_j(x)$ is chosen.

Boosting

Boosting technique, attempting to boost the accuracy of a learning algorithm, was originally proposed as a multiple prediction and aggregation scheme for classification problems and it has proven to be effective for reducing bias and variance and improving misclassification rates [Bauer and Kohavi, 1999]. Recently, this technique has been extended to regression problems. For example, Freund and Schapire [1997] suggested how boosting method can be applied to regression using their algorithm AdaBoost (Adaptive Boost); Drucker [1997] applied an ad hoc modification of AdaBoost to some regression problems and obtained promising results; Breiman [1997] proposed another Arcing (stands for Adaptive Resampling and Combining) algorithm as a modification of the original AdaBoost algorithms to apply boosting in regression. Recently, adaptive boosting algorithms have been derived from the viewpoint of gradient descent methods [see Friedman, 1999 and Mason 1999].

To address the special challenges posed by regression problems, some methods are often used, for example, converting the problem into a series of binary classification problems [Freud and Schapire, 1997], scaling each learner's errors based on its maximal error [Drucker, 1997], and using a threshold to evaluate a response as correct or incorrect [Avnimelech and Intrator, 1999]. Alternatively, several recent regression boosting methods adopt a residual-fitting strategy [Karakoulas and Shawe-Taylor, 1999], in which one trains learners sequentially to produce the residual error $|y\text{-}f(x)|$, instead of target output y, and finally linear combination will approximate y.

On the whole, boosting is different from re-sampling methods in that it reweights smoothly. In each boosting iteration a regression model is constructed on different weights on the dataset.

A typical boosting procedure can be described as follows:

1) Initialize the weights on the dataset uniformly, that is, $w_i^{(1)} = \frac{1}{n}$,

2) For t in 1 to T, construct a regression model $f_t(x_i)$ using the weights,

3) Compute the regression error of $f_t(x_i)$ as $\varepsilon_k = \sum_{i=1}^{n} w_i^{(t)} \frac{(y_i - f(x_i))^2}{\max\limits_{i}(y_i - f(x_i))^2}$,

4) Let $\lambda_t = \frac{\varepsilon_t}{1-\varepsilon_t}$ which is a measure of confidence in the predictor and

$\alpha_t = \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$ and update the weights of each observation as $w_i^{(t+1)} = w_i^{(t)} \lambda_t^{1-(y_i - f_t(x_i))^2}$.

5) Normalize $w^{(t+1)}$ so that they sum to one and then repeat steps 2 through 5.

Finally, it outputs a weighted ensemble predictor

$$f(x) = \frac{\sum_{t=1}^{T} \alpha_t f_t(x)}{\sum_{t=1}^{T} \alpha_t} \qquad (1.53)$$

Note that in each iteration the weights of those observations poorly predicted by $f(x_i)$ are increased and helps speed up the learning procedure. In a typical boosting algorithm, weighting data works in conjunction with regressor combination to improve a regression model.

In the above example, we applied the quadratic loss function, and certainly other loss functions such as absolute error can also be applied and the procedure is very similar. Although in the above regression models are used to illustrate the boosting procedure, extending it to a general model procedure is trivial.

No matter how various and complex these boosting algorithms, the basic idea remains the same, that is, to establish some weight function on the observations through some procedure and combine the simple regressors into a composite one. A common problem inherent in boosting is that it seems be especially susceptible to noise, because it gives more emphasis to those "difficult" data points, which is more likely to be contaminated by noise.

As for bagging, the main effect of boosting is to reduce variance. According to Breiman's work, it seems to do better than bagging. However, the actual performance of boosting on a particular problem clearly depends on the data and the weak learner. For example, given insufficient data or overly complex weak learner boosting might fail to perform well.

Stacking

Stacking [Wolpert, 1992] is not a particular algorithm, but a generic method of combining a collection of *m* different models, that could have been obtained by training on different subsets of data or by using different techniques. The purpose of stacking is to find out a better way to combine them rather than using simple averaging as in bagging or the weighted mean in boosting. Stacked regression [Breiman, 1992] combines linearly the models as

$$f_s(x) = \sum_{j=1}^{m} w_j h_j(x), w_j \in R, j = 1,\dots,m \,, \tag{1.54}$$

where $\{h_j(x)\}$, j=1,…,m denotes m different models and w_j are their individual weights. The optimal combining weights are estimated as

$$\hat{w} = \arg\min_{w} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{m} w_j h_j^{(i)}(x_i)\right)^2 \text{, with } w_j \geq 0 \tag{1.55}$$

where $h_j^{(i)}(x_i)$ is a predicted value by the *j*th model corresponding to data y_i, for example, the *j*th model is calibrated with the *i*th observation deleted. In other words, parameters are obtained by performing a least-squares regression of the output *y* on the *m* inputs $h_j^{(i)}(x_i)$.

By using cross-validated predictions, stacked regression actually tries to minimize prediction error by combining multiple models. Thus, rather than choose a single model, stacking combines them with estimated optimal weights. As shown by Breiman [1992], the performance of the stacked regressor improves when the weights are constrained to be non-negative, but the composite model is less interpretable than the choice of only one of the *m* models. In effect, if we restrict the minimization to weight vectors *w* that have one unit weight and the rest zero, this reduces to a winner-take-all model selection method based on the leave-one-out (LOO). As compared to the linear combination of models in equation (1.50), stacking tries to minimize LOO error rather than empirical error.

As we mentioned, stacking is a general method for combining models and can be employed together with other methods to improve a learning algorithm, for example, in Wolpert and Macready [1996] stacking was combined with bagging to obtain better accuracy and Drucker [1997] empirically shows that stacking does improve both bagging and boosting in some situation.

The proof of the utility of these three procedures is that they work well in some certain circumstances in the real world. In most empirical studies, the improved performance through the above procedures has often been demonstrated to be impressive. It is worth noting that they work mainly through two mechanisms, that is, data weighting and estimator ensemble. It is seen that all the above procedures including, bagging, boosting and stacking, combine multiple regressors to overcome instability or in other words reduce variance in parameter estimation.

By the no-free-lunch theorems by Wolpert and Macready [1997], there are similar no-free-lunch results concerning quadratic error, which means any improve-a-learning –algorithm procedure, including bagging, stacking, boosting and so on, hurts as often as it helps [Wolpert and Macready, 1996]. The only possible way to improve a learning procedure is to incorporate new information.

Bayesian Model Averaging

Bayesian theory provides us a natural and easy way to integrate the information from several different sources. It allows us to combine new observations with any prior information, which can be generic information about the system of interest, previous experience or expert judgment. In model combination context, we can define model probability $Pr(M_j)$ for each candidate model and treat the mean model as the optimal combined model

$$f_{BMA}(x) = \sum_{j=1}^{m} Pr(M_j)h_j(x)\,. \tag{1.56}$$

Model probability $Pr(M_j)$ can be interpreted in a similar way to that for a random variable. In the probabilistic world, just like a random variable a true model is assumed to never appear exactly as it is. If we can define the distance of two models in the model space somehow, for example, using some kind of norm, the model probability is actually converted to the probability of random variables. As such, the prior model probability distribution expresses our prior knowledge about the true model probability distribution in the model space.

After collecting a new dataset D, the posterior model probability $Pr(M_j|D)$ can be obtained to replace the model probability in the above equation. According to the Bayesian updating formula the posterior probability $Pr(M_i|D)$ can be calculated as

$$Pr(M_j \mid D) = \frac{Pr(D \mid M_j)Pr(M_j)}{\sum_{j=1}^{m} Pr(D \mid M_j)Pr(M_j)}, \tag{1.57}$$

where $Pr(M_j)$ is the prior probability of model M_j and $Pr(D|M_j)$ is the likelihood of the data set D given model M_j. Defining Bayes factor $B_{j0} = Pr(D \mid M_j) / Pr(D \mid M_0)$ and prior odd $\alpha_j = Pr(M_j)/ Pr(M_0)$, equation (1.57) can be rewritten as

$$Pr(M_j \mid D) = \frac{\alpha_j B_{j0}}{\sum_{j=1}^{m} \alpha_j B_{j0}}\,. \tag{1.58}$$

This is exactly the basic idea of a recent model combination method, namely, Bayesian Model Averaging (BMA) [Hoeting *et al.*, 1999] or Bayes factor [Kass and Raftery, 1995] weighting.

The difficulty of implementing BMA partly consists in the computation of the integral

$$Pr(D \mid M_j) = \int Pr(D \mid \theta_j, M_j) Pr(\theta_j \mid M_j) d\theta_j, \tag{1.59}$$

where $Pr(\theta_j|M_j)$ is the prior density and θ_j is the vector of parameters of model M_j, because the probability distribution functions might assume overly complicated high- dimensional functional forms, thereby making integral analytically intractable.

Fortunately, nowadays with the dramatically growing computational capability of model computers and especially with the invention of the Markov Chain Monte Carlo (MCMC) technique [Gilks, Richardson, and Spiegelhalter, 1998] numerical solution of the integral has become computationally possible. Basically, MCMC methods are sampling methods for multivariate probability distribution function, which attempt to simulate direct draws from some complex distribution of interest. MCMC approaches are so-named because one uses the previous sample values to randomly generate the next sample value, generating a Markov chain (as the transition probabilities between sample values are only a function of the most recent sample value). One particular MCMC method, the Gibbs sampler [Geman and Geman 1984], is very widely applicable to a broad class of Bayesian problems. At the same time, Monte Carlo integration is a numerical integration method, which computes complex integrals by expressing them as expectations for some distribution and then estimate this expectation by drawing samples from that distribution, that is, $\int_a^b h(x)dx = \int_a^b f(x)p(x)dx = E_{p(x)}[f(x)] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)$ with n very large.

The integral in equation (1.59) can be computed numerically by using Monte Carlo integration working together with MCMC method.

In regular statistical models, roughly those in which the MLE is consistent and asymptotically normal, the integral in equation (1.59) can be approximated via the Laplace method [Tierney and Kadane, 1986], i.e.,

$$\int e^{f(u)}du \approx (2\pi)^{d/2}|A|^{1/2}\exp\{f(u^*)\}, \tag{1.60}$$

where $f(u)$ is a real-valued function of d-dimensional vector u, u^* is the value of u at which $f(u)$ attains its maximum, A is minus the inverse Hessian of $f(u)$ evaluated at u^*.

Applying the Laplace approximation to equation (1.59) yields

$$Pr(D \mid M_j) = (2\pi)^{d_j/2}|\Psi_j|^{1/2} Pr(D \mid \tilde{\theta}_j, M_j)Pr(\tilde{\theta}_j \mid M_j)O(n^{-1}), \tag{1.61}$$

where d_j is the dimension of θ_j, $\tilde{\theta}_j$ is the posterior mode of θ_j, and ψ_j is minus the inverse Hessian matrix of $h(\theta_j)=\log\{Pr(D|\theta_j)Pr(\theta_j|M_j)\}$ evaluated at $\theta_j=\tilde{\theta}_j$.

Meanwhile, let's define

$$R_j = -\frac{1}{n}\left(\frac{\partial^2 \log h(\theta_j)}{\partial\theta_{jr}\partial\theta_{js}}\right)_{\theta_j=\hat{\theta}_j} = -\frac{1}{n}\sum_{i=1}^{n}\frac{\partial^2 \log h(\theta_j, x_i)}{\partial\theta_{jr}\partial\theta_{js}}\Bigg|_{\theta_j=\hat{\theta}_j}, \tag{1.62}$$

where $D=\{x_i|i=1,\dots,n\}$ and $\hat{\theta}_j$ is the MLE of θ_j.

From the law of large numbers, it follows when n tends to infinity,

$$R_j = -\frac{1}{n}\sum_{i=1}^{n}\frac{\partial^2 \log h(\theta_j, x_i)}{\partial\theta_{jr}\partial\theta_{js}}\Bigg|_{\theta_j=\hat{\theta}_j} \rightarrow -E_{\theta_0}\left[\frac{\partial^2 \log h(\hat{\theta}_j)}{\partial\hat{\theta}_{jr}\partial\hat{\theta}_{js}}\right]. \tag{1.63}$$

Furthermore, if $\hat{\theta}_j$ is in a close neighborhood of θ_0, i.e. $\|\hat{\theta}_j - \theta_0\| < \varepsilon$, which is very small, we have $R_j \rightarrow -E_{\theta_0}\left[\frac{\partial^2 \log f(\hat{\theta}_j, X)}{\partial\hat{\theta}_{jr}\partial\hat{\theta}_{js}}\right] \approx -E_{\theta_0}\left[\frac{\partial^2 \log f(\theta_0, X)}{\partial\theta_{0r}\partial\theta_{0s}}\right] = I(\theta_0)$,which is the well-known Fisher information matrix, whose determinant is bounded. As such, ψ_j is asymptotically equal to n times the inverse of the observed information matrix.

Therefore, when n is large, we have $\tilde{\theta}_j \approx \hat{\theta}_j$ and

$$\log|\psi_j| = \log|nR_j| \approx \log|nI(\theta_0)| = d_j\log n + \log|I(\theta_0)| = d_j\log n + o(1). \tag{1.64}$$

Finally, we obtain

$$\log \Pr(D \mid M_j) = \log \Pr(D \mid \hat{\theta}_j, M_j) - d_j \log n + o(1), \tag{1.65}$$

which is exactly the same as the BIC formula derived by Schwarz [1978].

With this approximation of posterior likelihood, we obtain the posterior model probability as

$$Pr(M_j \mid D) = \frac{\alpha_j \exp(-BIC_j)}{\sum_{j=1}^{m} \alpha_j \exp(-BIC_j)}, \tag{1.67}$$

which is very similar to AIC-based weighting method except that it allows us to incorporate prior preference among candidate models via α_j. Kass and Raftery [1995] discussed the relative merits of AIC and BIC in this context.

Volinsky *et al.* [1997] shows that Bayesian model averaging produces better models than selecting a single model.

Bayesian information aggregation

As mentioned earlier, combining candidate models is to integrate information contained in each model. Meanwhile, Bayesian method is a good way to combine information. This idea leads to another class of model combination methods, Bayesian information-aggregation, pioneered by Morris original papers [Morris, 1974, 1977]. Suppose θ is a continuous quantity to be estimated, and we obtain a group of cstimates $x_1,\ldots,x_K$ from a class of competing models, say, $M_1,\ldots,M_K$, respectively. According to the Bayesian formula, the posterior distribution of θ is

$$\Pr(\theta \mid x_1,\ldots,x_K) = \frac{\Pr(x_1,\ldots,x_K,\theta)}{\Pr(x_1,\ldots,x_K)} = \frac{\Pr(x_1,\ldots,x_K,\theta)}{\int \Pr(x_1,\ldots,x_K,\theta)d\theta}, \tag{1.68}$$

The capability of prediction of so constructed models comes from the statistical dependence between models and the truth. The central idea of these methods lies in modeling the dependence among models through $Pr(x_1, \ldots ,x_K, \theta)$. In light of the convenience of modeling dependence through the covariance matrix, many researchers assume the normal distribution of estimates $x_1,\ldots,x_K$, for example, French [1981], Winkler [1981] and Lindley [1983]. Influence diagram becomes a useful graphical tool in modeling covariance structure [Burns and Clemen, 1993]. A typical way to evaluate the joint distribution is to assess marginal and conditional distribution and aggregate by the Markov's property

$$\Pr(x_1,...,x_K,\theta) = \Pr(x_K \mid x_{K-1},...,x_1,\theta) \cdots \Pr(x_2 \mid x_1,\theta)\Pr(x_1 \mid \theta)\Pr(\theta). \qquad (1.69)$$

Clemen and Winkler [1993] propose to model the dependence based upon the conditional mean dependence assumption (CMDA), that is,

$$E(X_i \mid X_{i-1},...,X_1,\theta) = \beta_{i,0} + \beta_{i,1}X_1 + ... + \beta_{i,i-1}X_{i-1} + \alpha_i\theta \qquad (1.70)$$

where α_i and $\beta_{i,j}$ are coefficients to be evaluated.

By the above equation (1.70), the knowledge about the information sources is incorporated in aggregation. Thus, if we know the distribution of X_i in advance, such as Normal, Student T, Logistic, Laplace, Gamma and Beta, we can obtain its conditional distribution $\Pr(X_i|X_{i-1},...,X_1,\theta)$ with the expected value determined by equation (1.70). Finally, we obtain the posterior distribution of θ.

Although the above approach permits considerable modeling flexibility by allowing arbitrary distribution, it does not facilitate the modeling of exchangeability among information sources. Therefore, later on Jouini and Clemen [1995] propose to apply the theory of copulas to model dependence among the experts' opinions. A copula is a function that connects marginals with joints cumulative distribution function (CDF), so it is the copula that models the dependence among the random variables. For details about copula, see Dall'Aglio *et al.* [1991]. According to Sklar's theorem [Sklar, 1959], for any joint distribution function $F(x_1, x_2)$ with marginal distribution functions F_{X_1} and F_{X_2}, there exists a copula C with

$$F(x_1,x_2) = C(F_{X_1}(x_1),F_{X_2}(x_2)) \qquad (1.71)$$

for every $x_1, x_2 \in R$. If F_{X_1} and F_{X_2} are continuous, then C is unique. On the other hand, if C is a copula and F_{X_1} and F_{X_2} are distribution functions, then the function F defined by equation (1.71) is a joint distribution function with marginals F_{X_1} and F_{X_2}.

Therefore, with the copula which represents dependence among expert opinions one can construct the joint probability distribution of these opinions from the univariate distributions of individual expert's assessments. There exist a number of families of copulas in the literature, but the type of stochastic dependence and the degree of dependence they are able to capture vary. For a given problem, a crucial issue is to choose a suitable family of copula to construct the joint distribution. Jouini and Clemen [1995] recommend the use of Frank's copula

$$C_{n|\alpha}(u_1,...,u_n) = \log_\alpha\left[1+\frac{(\alpha^{u_1}-1)...(\alpha^{u_2}-1)}{(\alpha-1)^{n-1}}\right], 0 < \alpha < 1 \qquad (1.72)$$

where n is the number of experts, $u_1,\ldots,u_n$ represent individual marginals, and α captures the dependence whose value can be obtained from the Table I in [Jouini and Clemen, 1995].

Mixture of experts (MoE)

In machine learning, there are also some ensemble methods, which combine multiple simpler learners to improve predictions. The Mixture of Experts (MoE) architecture proposed by Jacobs *et al.* [1991] is one of such methods, which is a modular artificial neural network where each module is called an expert and is a parametric function of the inputs. As shown in Figure 2.3, the gate is also a parametric function and typically receives the same inputs as the expert networks. The gate network chooses the weights of each expert in the output of the mixture and for each input it determines which expert to use. The Mixture of Experts (MoE) architecture illustrated in Figure 1.3 can be formulated as follows,

$$f(x)=\sum_i g_i(x,\theta_i)\cdot f_i(x,\alpha_i),\quad \sum_i g_i(x,\theta)=1 \text{ with } g_i(x,\theta)>0 \qquad (1.73)$$

where f_i are the experts and g_i are the gate functions. Gating functions generate probabilities, based on which the input space is partitioned "softly". In other words, gating networks can be thought of as classifiers. Therefore, the major difference between MoE and ensemble of learner is that it is a nonlinear mixture of learners since the weight functions or the gate functions also depend on inputs x. In this aspect, it is similar to the locally weighted least squares (WLS). Each expert network is local in the sense that they fit to the data not equally well.

If the gating networks are also generalized liner, then the normalized gating function is a "softmax" function as

$$g_i = \frac{e^{v_i x}}{\sum_{i=1}^{m} e^{v_i x}}, \tag{1.74}$$

where v_i is a weight vector. Such gating networks can be interpreted as providing a soft split of the input space.

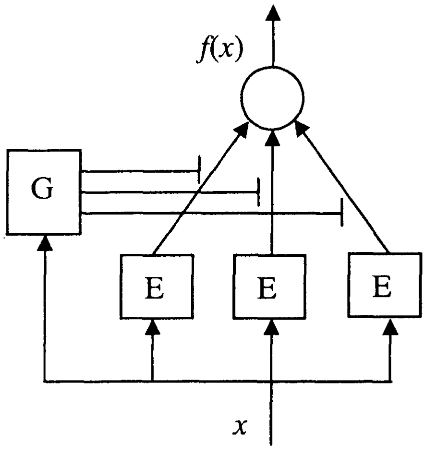


Figure 1.3 Mixture of experts architecture

In the case where each expert is a linear function and the gate chooses just one expert for a given input, the MoE constructs a piecewise linear approximation of the learned mapping.

Learning mixture of experts consists of learning the parameters of individual expert networks as well as learning the parameters of the gating network. As usual, the objective of this architecture is specified by defining an error function and then many algorithms can be applied to optimize the system. In order to encourage localization, Jacobs *et al.* [1991] a different error function which gives better performance:

$$e(x_i) = -\log \sum_j g_j(x_i, \theta_j) \exp\left(-\left\| y_i - f_j(x_i, \alpha_j) \right\|^2 / 2\sigma^2\right), \tag{1.75}$$

which is simply the negative log probability of generating the desired output vector under a mixture of Gaussians models of the probability distribution of possible output vectors given the current input, since errors for different experts are assumed to be normally distributed with the same variance and g_i can be viewed as the probability of selecting expert i for a particular case. The output vector of experts specifies the mean of a multidimensional Gaussian distribution. This objective function is certainly different from the traditional one for model combination

$$E=\sum_i\left(y_i-\sum_j g_j(x_i,\theta_j)f_j(x_i,\alpha_j)\right)^2 . \tag{1.76}$$

The error functions can be minimized by performing gradient descent [Jacob *et al.*, 1991]. At the same time, if we assume that errors for different expert networks are normally distributed with the same variance and thus the output of the whole network is a mixture of Gaussians, the learning of mixture of expert networks can be treated as a maximum likelihood problem. Since Expectation-Maximization (EM) algorithm is a general technique for maximum likelihood estimating especially suitable for mixture of Guassian problems, Jordan and Jacobs [1994] present EM algorithm for learning of the parameters of the architecture, where the hidden variables are identities of expert networks responsible for data points (x_i, y_i), i=1,.., n. In general, EM algorithm includes two steps, namely, Expectation and Maximization. It is in particular suitable for problems with "incomplete data". In the case of mixture-of-experts architecture learning, the "missing" or "hidden" variables are identities of expert networks responsible for a training case.

Empirical studies show the training of mixture of experts is significantly faster than the back-propagation networks and the EM algorithm is faster than the gradient descent learning algorithm.

The distribution over experts can be hierarchical, as in a hierarchical mixture model, giving a Hierarchical Mixture of Experts [Jordan and Jacobs, 1994] as shown in Figure 1.4. It can be interpreted as providing a nested "soft" partitioning of the input space within the partitioning providing by the higher-level gating network. The same algorithms can be applied for learning the hierarchical architecture.

The MoE method follows the divide-and-conquer principle to the problem of learning from examples. It can be considered a general method for combining local models learning from examples in small regions of the input space.

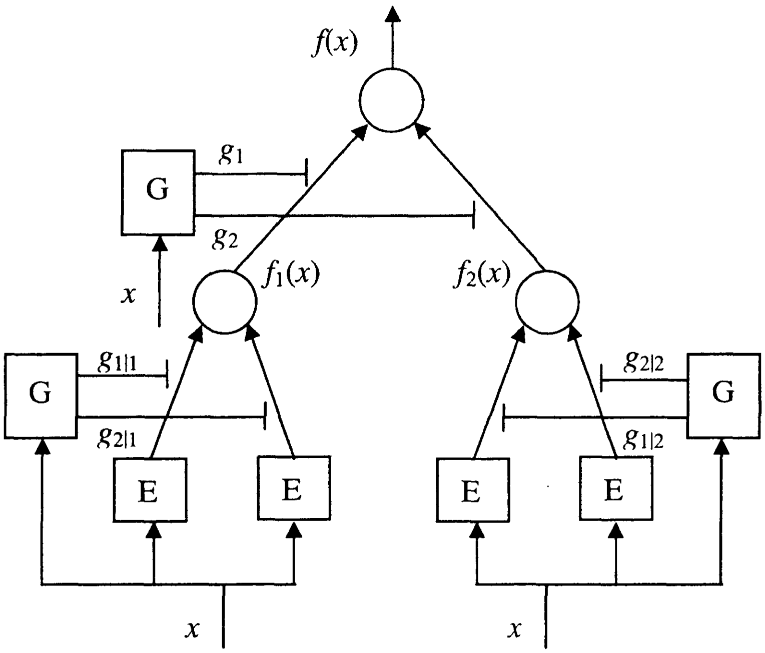


Figure 1.4 Hierarchical mixture of experts

1.2.3 Comparisons

It is central for model combination to improve performance in that it tries to exploit the information contained in all candidate models, avoiding the loss of information that might result if a single best model is chosen while the rest discarded. In the context of combining forecasts, Makridakis and Winkler [1983] conclude that combining forecasts seems to be a reasonable practical alternative when the true model of the data-generating process cannot be identified. This statement is equally applicable to combining general models. In this section, we will discuss why combining models should be preferred to selecting a single best model.

Heuristically, any model tries to explain a physical system from a certain angle. To this end, some assumptions and simplifications must be made. In addition, models may be created based upon distinct theories or calibrated using different observations. Thus, although a model can not capture all the properties of a real system, it does deliver lots of information concerning the modeled system. These distinct competing models are able to complement each other and come up with a better composite model. Just as in the Bayesian information-aggregation methods, each model depends on the true model somehow but in different ways and these dependences can be combined to tell us more information about the truth.

Moreover, the strategy of combining multiple models can be considered following the divide-and-conquer principle, that is, simple models try to embody some certain aspects of a complicated physical system and then they are combined into a more complex composite model. Alternatively, one might attempt to let a single model to incorporate all the features, but this perhaps makes the modeling process intractable because the model dimensionality of the true model is infinite, or even induces theoretical conflict inside a model. Therefore, an alternative philosophy is to divide the task of a complex system into simpler modeling processes whose results can be combined relatively easily to yield a satisfactory model. The hierarchical mixture of experts architecture [Jordan and Jacobs, 1994] is a good example, in which only the input space is partitioned though. This philosophy has been proven successful in many areas, for example, fast Fourier transform, multi-scale modeling, algorithm design and software development.

Evidence from recent research in machine learning also shows that the performance of the final model can be improved by creating a composite model by combining a group of competing models having different structures more than by choosing the model structure which is expected to predict the best [e.g. Abbott, 1999]. In the neural network community, “ensembles” of neural networks has been proven effective in improving performance, see for instance [Hansen and Salamon, 1990] and [Perrone and Cooper, 1993].

A single model might work better than all others within a certain domain, but it is hard for it to outperform others over the whole region. Besides, a composite model is usually more stable than a single model, i.e. smaller average variance over the data domain. Here is a simple example to illustrate the superiority of model combination over model selection. Suppose $f_b(x)$ is the best model among a group of competing models $f_1(x), \ldots, f_N(x)$, and we build a composite model as follows

$$f_c(\mathrm{x})=\sum w_j f_j(x) \tag{1.77}$$

where $f_j(x)$ refers to candidate models and correspondingly w_j denotes the weight of each candidate model. w_j's can be learned based upon a data set in some manner, for example, multiple regression method.

Certainly, within some region of x, for instance $x\in \Omega$, the best model $f_b(x)$ works better than the composite model $f_c(x)$ in terms of some measure of model performance, but it is not the case across the whole region, just because in some other regions other competing models might work better. On the whole, the global performance of this composite model will be better than that of the single best model by choosing appropriate weights; at worst, the composite model is at least as good as the best model by simply setting the weight of the best model 1.0 and all others 0.

To compare model selection and model combination in a formal way, we'd better first define some general modeling method evaluation criteria. In addition to predictive accuracy, generally there are other important criteria, namely, consistency, stability and globality, according to which to assess model selection methods. Actually, they are easy to be extended to model combination approaches.

(1) Predictive accuracy

The goal of both model selection and model combination is to maximize the model performance measurement. A model can be viewed as combination of two parts, namely the reasonable part and the error part. The reasonable parts in competing model overlaps and thus are highly dependent, which the error parts, due to modeler's bias or mistake, are independent. To improve a model's performance is to improve the reasonable part while suppressing the error part.

Let's first revisit the generalization error of models or model error in short. Generally, a model $f(x;\ \theta)$ consists of model structure as well as model parameters, i.e. $M=(S,\ \theta)$, where the model structure, or the functional form, is usually based on some theories and assumptions and parameters are estimated from observations. Model error can be decomposed into bias and variance, i.e.

$$\begin{aligned} MSE &= E\left[(f(x;\theta)-g(x))^2\right] = \left(E[f(x;\theta)]-g(x)\right)^2 + E\left[f(x;\theta)-E(f(x;\theta))\right]^2 \\ &= \{bias\}^2 + \{estimation\ variance\} \end{aligned} \tag{1.78}$$

Bias can be further decomposed

$$\begin{aligned} bias^2 &= [E(f(x;\theta))-g(x)]^2 = [f(x;\theta^*)-g(x)]^2 + [f(x;\theta^*)-E(f(x;\theta))]^2 \\ &= \{model\ bias\}^2 + \{estimation\ bias\}^2, \end{aligned} \tag{1.79}$$

where g(x) denotes the unknown true model and $f(x;\theta^*)$ refers to the pseudo-true model, the closest model to the true model given a model structure $f(x;\theta)$,

$$\theta^* = \inf_{\theta\in\Theta} E[(f(x;\theta)-g(x))^2]. \tag{1.80}$$

Model bias results from model structure, namely misspecification bias. In the context of model selection, it is also known as model selection bias.

Since model structure is based on some beliefs and theories, model bias is due to the modeler's lack of enough knowledge or bias in knowledge. Because data used for parameter estimation is randomly collected, this leads to the estimation variance of a model. If the collection of data is not random, there may exist some bias in data, which contributes to the estimation bias. Another source of estimation bias comes from estimation methods, which might result in biased estimators

In the example of linear regression model, the linear assumption and the choice of regressors constitutes the model structure. Regression coefficients are model parameters, estimated from data. The sources of bias and variance are clear.

In the following, we will discuss how model selection and model combination affect bias and variance, respectively.

Zucchini [2000] points out that all model selection criteria suffer from model selection bias. The more competing models, the more risky model selection. To see how combining model having different structures helps reduce model bias, we just take BMA as an example. Suppose we have a class of competing models $h_j(x)$ with different model structures, each with independent bias $b_j(x)$ with E[$b_j(x)$]=0. We construct a composite model by weighted average

$$f_c(x) = \sum_{j=1}^{m} p_j h_j(x). \tag{1.81}$$

Then, the bias of the combined model is $\sum_{j=1}^{m} p_j b_j(x)$, which tends to E[$b_i(x)$]=0 as m increases.

Another extreme is that the competing models have complete dependent bias, $b(x)$, and then the bias of the combined models is equal to $b(x)$. However, this is rare the case. A real case is most likely in between the two extremes. But whatever, combining models facilitates model bias reduction. This conclusion agrees with Wasserman [2000] that BMA as a weighted average diminishes the problem of model selection bias due to fact that it is not a risky kind of winner-takes-all procedure favored by model selection.

In real world, the model bias results from modeler's lack of knowledge or bias in beliefs. However, different models might be created based upon distinct theories or learned from different data, and thus modelers have different bias, so it is reasonable to assume that individual biases are independent.

At the same time, combining multiple models is a variance reduction technique as shown in the example of simple average method. Suppose there are m competing unbiased models, denoted as $f_j(x)$, j=1,...,m, having uncorrelated errors of the same variance, $\text{var}(f_j(x)) = v$ and $\text{cov}(f_j(x), f_k(x)) = 0$.

If we again create a weighted average model

$$f_c(x) = \sum_{j=1}^{m} p_j f_j(x). \tag{1.82}$$

Then the variance of the combined model is

$$\text{var}(f_c(x)) = \sum_{j=1}^{m} p_j^2 \cdot \text{var}(f_j(x)) = v \sum_{j=1}^{m} p_j^2 . \tag{1.83}$$

If $p_j < 1/2$, then

$$\mathrm{var}(f_c(x)) < v/2. \tag{1.84}$$

Similar assumption to that for bias can be made for model variance, which is more precisely parameter estimation variance. Since the data used for parameter estimation is randomly gleaned, models can be considered to be uncorrelated in reality as far as the random part is concerned.

However, on the other hand, model combination tends to increase variance because more parameters need to be estimated from the same number of data. Taylor and Siqueira [1996] discuss in detail about the cost of adding parameters to a model. So the advantages of model combination come at the expense of perhaps making the predictions more imprecise. This raises an interesting theoretical question about the overall effect of model combination on the predictive error. We will give a heuristic answer in the following.

To assess a model, the first thing we are concerned with is its predictive accuracy, because the goal of models is to predict the future. However, we can never know the true model or collect an infinite number of samples, and thus the predictive accuracy has to be estimated based upon a limited number of samples by some model performance evaluation method discussed earlier. Just as we pointed out earlier, any 2-tuple model assessment method, i.e. a pair of discrepancy function and generalizablity estimation method, can be used to assess models, which is also suggested by Browne [2000]. Before comparing a single best model and a combined model, we also need first to choose a model performance evaluation method.

Can model combination improve the model performance in terms of a certain model performance evaluation method? The answer is yes. Let's show this as follows:

Suppose we have a group of competing models, denoted as $f_j(x)$. We construct a composite model as

$$f_c(x) = \sum\nolimits_{j=1}^{m} w_j \cdot f_j(x), \tag{1.85}$$

where w_j are individual coefficients.

Let's designate as *MP* the model performance evaluation method we choose. Given a model $f(x)$, we can obtain its performance by $MP(f(x))$. Now, the goal of model combination is to maximize the performance of the composite model or equivalently minimize its generalization error, that is,

$$v = \arg\max_{w \in R^m} MP\left(\sum_{j=1}^{m} w_j \cdot f_j(x)\right), \text{ where } w=[w_1, \ldots, w_m], \tag{1.86}$$

by whatever optimization algorithms.

A model selection procedure is to find out the a single best mode that has the greatest model performance,

$$MP_{\max} = \max_{j=1,\ldots,m} MP(f_j(x)), \tag{1.87}$$

and the "best" model denoted as $f_b(x)$ corresponding to coefficient u=[0…1…0].

There is no doubt that

$$MP\left(\sum_{j=1}^{m} v_j \cdot f_j(x)\right) \geq MP\left(\sum_{j=1}^{m} u_j \cdot f_j(x)\right) = MP_{\max}, \tag{1.88}$$

which means the performance of the combined model is better than the single best model in terms of the chosen model assessment criterion, because otherwise v=u.

This completes the proof of our argument. Even if finally it turns out that the performance of the combined model deteriorate, the problem does not lie in model combination but stems from the inappropriate choice of generalizability estimation method.

(2) Consistency

In statistics, an estimator $\hat{\theta}$of a parameter θ is said to be consistent if $\hat{\theta} \to \theta$as the sample size $n \to \infty$. Likewise, in model selection a modeling procedure is consistent if the models it produces gets closer and closer to the true model as the sample size n tends to infinity. Another simpler and more straightforward concept is dimension consistency. If as in many cases model complexity is only affected by degree of freedom, model dimension is defined to be the number of free parameters in a model. A criterion is dimension consistent if and only if its estimate of the model dimension converges to the correct value as the number of data tends to infinity if a true model is among those being considered [see Woodroofe 1982]. For example, AIC was proven to be inconsistent, e.g. by Shibata [1976] and Woodroofe [1982] for i.i.d data, while BIC was shown to be consistent by Woodroofe [1982].

The consistency of unbiased estimators is guaranteed by the law of large numbers. Although sufficient, this is not necessary, for example, asymptotically unbiased estimators can also be consistent. However in model selection, the situation is somehow different. Denote as M the set of all multivariate models and as $f(x;\theta)$ a subset of M fully specified by parameters θ. A model $f(x;\theta)$ is correctly specified if the true model $f(x) \in f(x;\theta)$, and otherwise misspecified. A model $g(x;\theta)$ is nested under $f(x;\theta)$ if $g(x;\theta)\subset f(x;\theta)$. For a model selection procedure to be consistent, it is required that the model is correctly specified, or at least asymptotically so. In other words, model bias is required to converge to 0. Therefore, for any model selection criterion to be consistent, first it is necessary that candidate models are correctly specified. Otherwise, even with an infinite number of observations, the final model can only converge to the pseudo-true model $f(x;\theta^*)$, rather than the true model f(x).

Model bias is a common caveat for the consistency of all model selection criteria. The only solution is to reduce or eliminate model bias somehow. As we discussed earlier, model combination can potentially meet this goal. Intuitively, integrating information in all distinct candidate models help create a more complete composite model. To illustrate this point, let's take the linear regression model as an example. Suppose the true model can be expressed as

$$f(x)=\sum\nolimits_{j=1}^{3}\beta_j h_j(x)\,, \tag{1.89}$$

where $h_j(x)$'s are regressors and β_j's are corresponding coefficients.

We have three competing models $f_1(x)=\beta_{11}h_1(x)+\beta_{12}h_2(x)$, $f_2(x)=\beta_{22}h_2(x)+\beta_{23}h_3(x)$ and $f_3(x)=\beta_{31}h_1(x)+\beta_{33}h_3(x)$, and then combine these three models

$$f_c(x)=\sum\nolimits_{j=1}^{3}w_j f_j(x)\,. \tag{1.90}$$

Note that since each candidate model is biased in structure since they incorporate only part of the predictors. Whatever the model selection criterion and no matter how many observations are obtained, the final model is biased. In contrast, the combined model involves no model bias since

$$f_c(x) = \sum_{j=1}^{3} w_j f_j(x) = \sum_{j=1}^{3} w'_j h_j(x) , \tag{1.91}$$

which has the same model structure as the true model. Hence, it results in an unbiased model.

In the case where one of the competing models is correctly specified, if a consistent model assessment method is chosen, the composite model will converge to the true model definitely due to the law of large numbers. For example in the linear combination, the weight of the correctly specified candidate model converges to 1.0 and others 0, which results in the true model.

(3) Stability

A learning algorithm, including both model selection and model combination, is unstable if small changes in the training data lead to significantly different models and relatively large change in accuracy. Typically, instability in learning can be attributed to high variance. Stability can be tested by perturbation.

The desirability of stability can be seen from some theoretical results that relate the generalization error to the stability of a modeling procedure. For example, Bousquet and Elisseeff [2002] show that the generalization error bound decreases exponentially for algorithms with uniform stability, which implies for models having the same empirical error the generalization error bound will be significantly tighter for those algorithms with uniform stability. Recently, Mukherjee *et al.* [2002] even further pointed out that stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization (ERM).

Many researchers found that model selection methods were instable and the predictive error was remarkably large [e.g. see Miller, 1984 or Breiman, 1996]. In the following, let's heuristically analyze what causes instability in model selection.

Model selection is often done based on some data set. The variance of the final model is composed of model selection variance and parameter estimation variance [e.g. see Burnham and Anderson, 2002].

Model selection variance also results from the random fluctuation in small sample, just like the variance of estimators given a small sample. The winner-take-all principle favored by all model selection criteria make it risky in that a slight different in data might results in choosing a different competing model. If we incorporate this winner-take-all procedure in a weighted scheme, the weights are discrete and can only take value either 1 or 0. A natural way to smooth it is to use continuous weights instead, similar to applying ridge regression in variable selection problem [Breiman, 1996a]. This idea leads us to model combination. Thus, model combination does not suffer from model selection variance.

Parameter estimation variance is due to the variance in sampling data. For already efficient estimators, it is can not be improved without increasing the sample size, because the minimum variance of unbiased estimator is lower bounded by the reciprocal of the Fisher information, know as Cramer-Rao bound [Rao, 1945 and Cramér, 1946]. However, if the learning algorithm is unstable, combining multiple models is a variance reduction technique, for example using bagging method [Breiman, 1996b]. More generally, unstable learning algorithms can improve their accuracy by perturbing (i.e. producing multiple models by perturbing the training set or learning method) and then combining. Breiman [1996c] call such techniques P&C methods. For example, the bagging is a P&C technique, which helps improve the performance of unstable learning algorithms by averaging over multiple models learned from many bootstrap replications.

Since we have learned that the model selection procedure is unstable, we are able to design a modeling procedure using bagging, in which model selection procedure is repeated many times on bootstrap replications and the final model is obtained by averaging over all the selected models. Since in each model selection procedure the selected model might be different, the resultant model is actually a combined model in the form $f_c(x) = \sum_{j=1}^{m} w_j f_j(x)$.

(4) Globality/locality

Another important issue in assessing a model is its globality, or uniform consistency and efficiency across the input domain, because the predictive capability of a calibrated model relies on the validity of interpolation and extrapolation. The failure of a model to extrapolate or generalize beyond the range of the observed data arises not only from small sample fluctuation, but also from the failure of the sample data to properly represent the domain of prediction. This limitation is shared by all standard model selection method, because model selection is not supposed to manage this kind of error. This problem is well identified by Busemeyer and Wang [2000] in the context of time series.

Actually, this phenomenon is quite common in the physical world. Each model has its own applicable domains, outsides of which its performance might deteriorate steeply. This might be because the law governing the modeled system changes from an input domain to another, or because of the poor extrapolatability of empirical models. As we mentioned earlier, any model is built based on some theories and learned from observations. In different sub-regions the mechanisms underlying the physical system might be different and correspondingly models have varied features. A certain theory usually takes into account one kind of mechanism, which leads to the locality of the models based on it. In addition, data collected in a certain domain might not represent those in other domains. For example, the noise variance or even error distribution varies over the input space. Models learned from such data certainly suffer from localization.

In order to show why model combination might help mitigate such problem, let's look at a very simple example. Suppose we have two competing models, one of which works better within a sub-region I and the other has better performance in the other sub-region II. Obviously, using weighted average method we can create a composite model $f_c(x)=w_1f_1(x)+ w_2f_2(x)$, with $w_1+ w_2=1$.

In an extreme case, we can set w_I=1 in region I and 0 in region II, which is certainly the optimal composite model we can have. This combined model works better than either of the candidate models as far as the whole input region is concerned.

In the above extreme case, weights are not constant over the input region. A possible solution is to partition the input region into two sub-regions. However, even if we use constant weights, it can be much better than candidate models with the bottom line of as good as the better one between them.

In the above, we argued that model combination is a better choice than model selection from many different angles. Although in argument we used weighted averaging or BMA as an example to show the advantages of model combination over model selection, it is not the only approach, and usually not the best way, to implement model combination. As mentioned earlier, the goal of model combination is to build a more accurate and precise model by integrating information in multiple distinct models and data sets, and so we need to find out an effective way to realize this goal.

1.3 Feature-based model combination method

Before we propose a new model combination method, let's first discuss how a physical model is created, what might cause its failure to predict the future, and how they can be improved with regard to a specific problem. Here, when we speak of the failure of a model to predict the future, we mean the prediction error is beyond an acceptable level.

Conceptually, a model is a hypothesis about how a system works or responds to changes in its inputs. Usually, a model is expressed in a form of mathematical formulations. In practice, each model is built on some scientific or technical assumptions, thereby significantly simplified. However, with such simplifications and plausible assumptions, models are just approximations to the truth. Furthermore, these simplifications and assumptions even turn out to be inappropriate or even erroneous later on, especially for a model that is built when science is immature and data are lacking for model testing and validation.

The model bias and uncertainty may stem from both model structure and model parameters. For scenarios and phenomena of interest, alternative sets of scientific or technical assumptions may be available for developing a model, which leads to different mode structure. Model structure is embodied in the form of the equations used and in the selection of variables, which serve as model inputs. Meanwhile, it is possible to parameterize model structures into a higher order model. Both bias and uncertainty in model structure and parameter estimation may result in the failure of a model to predict the future, that is, the expected error is unacceptably great. In the following some possible causes are listed and their remedies are also discussed.

1) Error

In a modeling process, some mistake can be made occasionally or due to one's misunderstanding or biased belief concerning a real physical system. This might lead to model misspecification, big bias in model structure.

For this reason, a model must be validated before being used to make prediction. If an effect is supported by data, it is valid, but otherwise it is most likely to be spurious and should be removed.

2) Incompleteness

A model may miss some features due to the lack of enough knowledge about the truth. For example, the significance of these missing features may vary under different conditions and thus it is not present in the data available to the model creator.

With a class of diverse competing models, this weakness can be improved in some degree, because in so doing it is possible to aggregate those effects contributed by individual models, which finally results in a better approximation to the full truth.

3) Bias

Bias includes both biases in model structure and model parameters. Besides errors and the incompleteness, there are other causes that might result in model bias, for example, incorrect assumptions or overly simplification. Model parameter bias refers to bias resulting from estimating model parameters based on data. A model learned from data, which may be not collected randomly but discriminatively, will have large bias in parameter estimation. Consequently, a resultant model may include some specific features pertaining to certain domain or data pattern and therefore cannot be generalized to the whole region. In other words, spurious features tend to be included in a model.

4) Uncertainty

Uncertainty refers to both model structural uncertainty and model parameter uncertainty. Model structural uncertainty results from some random factors that affect the choice of model structure. Model parameter uncertainty is due to the random fluctuation in finite sample, which is almost unavoidable. But, as indicated earlier, combining different models helps reduce uncertainty.

5) Localization

A model, which is validated for a certain input region, may be proven to be completely inappropriate when extrapolated to other regions. For example, some assumption underlying a model can only be met in a certain range of inputs. Hence, in other regions this model may turn out to be invalid.

It is also conceivable that a real-world system may enter a different phase with inputs varying, and even follows different laws in another phase. This may lead to the variation of model structures. In such case, it is possible to find out some missing hidden variables to characterize such phase-switching phenomenon.

Another possible reason may be that effects underlying a system have different influence over the input range. To address such problem, nonlinear combination may be helpful.

We may propose a mixture of local model method, in which the whole range of inputs is divided into several partitions. Instead of using hard partition, we may apply soft partition by means of defining a weighting function like member function in fuzzy theory. A weighted loss function will also be used to calibrate regression models in each partition. The weighting function also serves as the weights in the step of mixing local models.

From the above analysis, it is possible to mitigate all weaknesses by combining multiple competing models.

At this point, it is necessary to clarify some concepts, namely "dependence" and "independence" among competing models. A model can be thought of as composed of two parts, reasonable part and error part

$$f(x) = h(x) + \varepsilon(x)\,, \tag{1.92}$$

where the reasonable part $h(x)$ is the valid part that reflects the truth in a correct way, while the error part $\varepsilon(x)$ is due to the modeler's bias and some errors. Such decomposition is similar to Mosleh and Apostolakis [1986]' additive error model of experts. When we say two competing models are highly dependent, we mean the reasonable part is highly correlated; when we say two competing models are independent, we mean the error part does not depend on each other. To better understand this, let's consider a simple example. Suppose a random variable Y_1 is governed by a normal distribution $N(ax, \sigma^2)$ and another random variable Y_2 follows a normal distribution $N(bx, \sigma^2)$, where a and b are constants, x is a variable, and σ^2 is the variance. If at a certain time we know the realization of Y_1, say y_1, we can predict Y_2 more accurately, which implies the dependence between Y_1 and Y_2. On the other hand, knowing the random part in Y_1 does not help predict the random part in Y_2, because they are independent.

The basic idea behind model combination is to aggregate all available information effectively, which, however, may contain errors or noises, and then obtain a new model as good as possible. In other words, we just integrate the valid information in the reasonable part $h(x)$ but reduce the effect of the error part $\varepsilon(x)$. A model combination system acts like function that maps a group of candidate models and a data set into a new composite model. The goal of this system is to minimize the generalization error or the expected error. In some sense, model combination is an optimization problem.

Before performing model selection we need first to make certain the model performance assessment method. As pointed out earlier, a model performance assessment method can be designed by combining its two elements, distance function and generalizability estimation method. In fact, the choice of distance function and generalizability estimation method highly depends on specific problems.

In addition, a good model combination method had better have the following desirable properties:

(i) It is able to aggregate information in all competing models, thereby improving model performance.

From the angle of information, the merit of model combination comes from its capability of integrating information. The more information incorporated, the better. Therefore, if a method can make use of information in all candidate models and data in a more efficient way, it can produce a more complete and accurate composite model.

(ii) It should be able to detect errors in competing models in some degree, thereby reducing model bias;

As mentioned in the beginning of this section, each candidate has a reasonable part and an error part. The purpose of model combination is to combine the reasonable part but get rid of the error part at the same time. Otherwise, the errors will contaminate the composite model.

(iii) It can model dependence among competing models and thus reduce information redundancy;

A model combination strategy must be able to robustly handle the inherent dependence, or correlation, among candidate models, otherwise multicollinearity will cause lots of trouble just as in linear regression.

As pointed out in [Hogarth, 1987], the poor performance of human judges relative to statistical models stems largely from an inability to recognize and process redundant information appropriately, which, in part, reflects the importance of reducing information redundancy. Furthermore, reducing information redundancy helps reduce model dimensionality, e.g. the number of factors in a factor model, and thus reduce the variance in estimating model parameters.

(iv) It is able to combine different kinds of information, including models and data;

Since information about a physical system might come in the forms of theories, data, or even expert judgments, a good model combination method should be able to incorporate all these kinds of information.

(v) It has robust performance when having different sets of data;

In the previous section, we argued that stability is an important criterion by which to assess a modeling procedure. A stable model combination method is robust to changes in training data set.

(vi) It is objective, involving no subjective judgment.

Ideally, a model selection process should be objective and therefore repeatable.

To achieve the above goals, basically improving both accuracy and precision, we will propose a new model combination method. It is worth noting that this method is mainly intendcd for the situations where there is no well-founded theory and no enough data is available, because otherwise we may be able to derive a more exact theoretical model.

1.3.1 Model structure analysis

In general, a model consists of model structure, functional form of model formulation, and model parameters. Model structure is worked out on the basis of some theories or hypothesis, while model parameters are estimated from observations. Here, let's analyze model structure from a new prospective.

In practice, object can be efficiently modeled as combination of effects or features, for instance in pattern recognition. Similarly, a model can also be views as an ensemble of features arranged in some way. To simplify the problem a lot, we assume features are linearly mixed in a model, that is,

$$f(x,\beta)=\sum_{j=1}^{m}\beta_j h_j(x)\,, \tag{1.93}$$

where $h_j(x)$ are features, which is nonlinear functions of inputs, and β_j are corresponding feature coefficients. Mathematically, features are function basis, which are not general but specific to problems. For the true model, whose dimensionality is infinite, the number of features tends to infinity. For a real model, it might only include a finite subset of features. Distinct competing models might incorporate a different subset of features and have different feature coefficients.

In fact, the linear assumption is not accidental. At first, mathematically no matter how complicated a function is, it can always be expanded in some basis functions and finally expressed as sum of some simpler nonlinear functions. In some sense, features can also be thought of as basis function, which, however, depends on the specific problem under investigation. Therefore, a complicated nonlinear model can also be expressed as an additive model of simpler nonlinear functions, namely features. If some features are entangled together so tightly, they can be combined into a single feature. For example, in energy functions mass and velocity have to be so combined as to have energy units.

Another often used technique for approximating a nonlinear system is to divide the input space into many small regions and in each small region linear approximation gives satisfactory accuracy according to the Taylor expansion.

Thus, the linear assumption is appropriate in most cases. In such a model framework, each candidate model can be written as

$$f_i(x,\beta_i) = \sum_{j=1}^{k} \beta_{ij} h_{ij}(x) + \varepsilon_i(x)\,, \tag{1.94}$$

where $h_{ij}(x)$ are features and β_{ij} are corresponding feature coefficients, $\varepsilon_i(x)$ is the additive error in the candidate model $f_i(x,\beta_i)$. Candidate models are approximations of the true model in the sense that they only incorporate a subset of features of the true model and the factor loadings are not precise. Distinct models might contain a different subset of features or have different factor loadings.

Generally, different candidate models might incorporate different subsets of features, and the composite composed of the union of the subsets of features will produce a better approximation to the true model. Therefore, the information in multiple competing models is aggregated.

In order to aggregate information in all competing models, it is important to model dependence among multiple competing models and reduce information redundancy. In our feature-based model structure, competing models depend on each other through common features as in factor analysis [Everitt, 1984]. Therefore, feature extraction enables us to model the dependence among a group of competing models.

Another advantage of feature extraction is dimensionality reduction. By summarizing a physical system in the form of features, we can effectively decrease model dimension with the minimum loss of information by discarding those trivial features.

In integrating features into a composite model, the data comes into play its role. The feature coefficients can be estimated based upon observations.

Detecting errors in candidate models and eliminating them is another important thing in model combination. In our feature-based scheme, we are able to test the validity of features using new samples, that is, feature selection. In so doing, spurious features can be removed from the feature set.

So, this feature-based scheme can meet most of our goals in model combination. However, as in pattern recognition problems, the features that are needed depend on the specific problem that one wants to solve, and designing a good set of features is more of an art than a science.

1.3.2 Identify model features

Unlike general function expansion, features are completely problem-dependent. Since each candidate model is an approximation to the true model although the feature sets are incomplete and feature coefficients are imprecise, it is possible to extract features from the group of competing models. In cases where there are no or very few existing models, we might use features extracted from models for similar systems based upon an assumption that similar phenomena have similar features.

There are many statistical methods that can be applied for feature extraction, including principal component analysis (PCA), factor analysis (FA), independent component analysis (ICA) and independent factor analysis (IFA).

In most of these methods, the generative model of data variables X_i is assumed to be linear mixture of underlying components or factors S_j

$$X_i = \sum_{j=1}^{m} w_{ij} S_j + \varepsilon_i \text{, or in matrix notation } X=W\cdot S+\varepsilon, \qquad (1.95)$$

where the mixing matrix element w_{ij} is also termed factor loading in factor analysis and ε is the noise.

The purpose of feature extraction is to identify unobserved underlying factors S given some observed data variables X.

(1) Principal component analysis (PCA)

In principle component analysis (PCA), data variables are assumed to be exact linear mixture of factors that are assumed uncorrelated, that is, ε is assumed to be zero. Principal components turn out to be linear combinations of the observed variables, having the maximum variation in the smallest number of variables. For instance, the first principal component can be obtained by maximizing the variance of a combined variable of observed variables. To ensure uniqueness, all principal components must be orthogonal.

In mathematics, the principal components can be shown to be the eigenvectors of the covariance matrix of observed variables and the eigenvalues are equal to the variance of each component. The first component is corresponding to the eigenvector associated with the largest eigenvalue.

A nonlinear version of PCA is principal curve. Principal curves are smooth curves that minimize the average squared orthogonal distance to each point in a data set. It can be done by nonlinear regression with Gaussian noise on both x and y.

Most often, a small number of principal components are enough to explain most of the variation in the original data, thereby resulting in reduced data dimensionality. Actually, this analysis is the same as the discrete Karhunen-Loeve expansion, and therefore in the context of signal processing or pattern recognition, it can be used to perform data compression, optimal pattern representation and feature extraction.

(2) Factor analysis (FA)

Factor analysis, which is similar to PCA except that it includes Gaussian noise, ε, which is allowed to have an arbitrary diagonal covariance matrix. PCA is actually is a noiseless version of FA. Thus, values of factors can not be directly computed from observed variable due to the existence of noise.

In the context of FA, the unobserved sources are called "common factors" and the noise "unique factors". Both factors and factor loadings are estimated from the data by methods like maximum likelihood. If we set unique factors equal to zero, FA reduces to PCA.

FA is intended to find out a small number of latent variables to explain the data. However, unlike principal component analysis which is intended to explain data variables, FA is used to identify underlying factors that explain the correlations among data variables. Since noise or unique factors affect the variance of observed variables and correlations among them due to common factors, in performing FA we first need to estimate corrected covariance matrix from data and then conduct principal component analysis afterwards.

To aid in interpreting the factors, FA attempts to make factor loadings either very high or very low. This can be done by rotation techniques, for example, varimax, quartimax or equimax. In reality, after so doing those unimportant factors can be possibly eliminated, which helps data reduction.

Principal component analysis is often preferred in data reduction, while factor analysis is often favored when the purpose of analysis is to detect structures.

(3) Independent component analysis (ICA)

One inadequacy of both PCA and FA stems from the assumption of Gaussian factors, since they do not require the factors to be mutually independent but merely uncorrelated. Consequently they only exploit second-order statistics of the observed data.

As in factor analysis, in ICA the data variables are assumed to be linear or non-linear mixtures of some unknown latent variables, which are assumed non-Gaussian and mutually independent as compared to uncorrelated and Gaussian components in PCA and

FA.

In ICA, components are non-Gaussian and therefore it goes beyond second-order statistics of data for fitting model. In effect, statistical independence is inherently linked to higher order statistics. For two independent random variable Y and Z, it holds that for two arbitrary functions $f(\cdot)$ and $g(\cdot)$ $\mathrm{E}[YZ]=\mathrm{E}[f(Y)]\cdot\mathrm{E}[g(Z)]$. In particular, we have $\mathrm{E}[Y^m Z^n]=\mathrm{E}[Y^m]\cdot\mathrm{E}[Z^n]$.

Since in ICA, components are assumed to be non-Gaussian and independent, there are two directions in implementing ICA, namely maximization of non-Guaussianity measured by kurtosis or negentropy and minimization of dependence measured by mutual information. In fact, these two techniques do not conflict with each other and rather Hyvärinen and Oja [2000] show that they are equivalent. Based upon these objectives, many efficient algorithms have been proposed to perform ICA.

ICA has been widely used for blind-source separation, imaging and signal processing and even forecasting.

(5) Independent factor analysis (IFA)

The above methods are closely related to each other and they can be unified in independent factor analysis (IFA) proposed by Attias [1999].

In the framework of IFA, each factor is modeled as a mixture of Gaussian, therefore it can learn arbitrary factors, both Gaussian or Non-Gaussian. This constitutes a big advantage of IFA over ICA. Maximum likelihood is employed to estimate factors as well as mixture matrix W in a probabilistic context. In particular, an efficient algorithm, namely Expectation-Maximization (EM), is applied to implement ML estimation.

In addition, it is superior to ordinary ICA in some context because it can deal with noise. When the source factors become Gaussian, IFA reduces to ordinary FA; when there is no noise present, IFA is equivalent to ICA in non-Gaussian cases and PCA in Gaussian cases.

Although all the above approaches are feature extraction and data reduction methods, intuitively independent component or independent factor analysis is a better choice for our purpose, because there is no mechanism forcing underlying factors to be Gaussian.

1.3.3 Construct a composite model based upon components

After obtain a set of problem-specific features, the next step is to aggregate them into a composite model under the guide of data.

Since features result from linear decomposition of candidate models and we assume candidate models are similar to the true model, it is reasonable to assume that the true model is also a linear combination of features, i.e.

$$f_c(x) = \sum_{j=1}^{m} \beta_j h_j(x), \tag{1.96}$$

where $h_j(x)$'s are features or factors and β_j's are factor loadings.

Factor loadings can be estimated by various methods, for example, multiple linear regression, Bayesian regression, bagging or stacking. The choice of estimation method depends on specific problems. However, one thing worth emphasizing is that the calibration must be done based on data. For example, if an effect is supported by data, it is valid, but otherwise it is most likely to be spurious and should be removed.

1.4 Discussion

In this chapter, we first reviewed many different model selection criteria and model combination method, and then show the advantages of model combination over model selection by comparing them in terms of several different standards, and at last we proposed a new feature-based model combination method, in which, rather than combining multiple competing models directly, candidate models are first mapped into a feature space, and then features are selected to construct a new composite model.

According to the efficiency in aggregating information and other standards mentioned in the section 1.2.3, heuristically we think the methods are ordered as follows in preference: a single model, select the best model, simple average, weighted average (different weighting strategies including bagging and boosting), linear combination model or regression using candidate models (including stacking), and feature-based model combination. First, we have already argued that model combination is superior to model selection, which in turn is obviously better than picking up one from a group of competing models randomly. Second, the preference of linear combination method over weighted averaging can be seen from the fact that stacking performs a bit better than both bagging and boosting [see e.g. Zenko *et al.*, 2001], mainly because it removes the constraint that coefficients sum up to 1. Finally, the superiority of feature-based model combination method mainly comes from its ability to detect errors and reduce dimensionality.

Among all these approaches, only the new feature-based model combination method can potentially meet the goals we put forward in the beginning of section 1.3. However, to implement the feature-based model combination efficiently, there are still some important issues that we have not touched on until now.

(1) Candidate model choice

By now, we just assume we have a group of competing models at hand, and we never mention where and how we get them. In fact, the choice of candidate model is also important. For example, if we pick up a terrible candidate model carelessly, it might result in misleading features or the big error will ruin the composite model. Therefore, a candidate model should be at least competitive.

For another example, in order to make the set of features more complete, the more candidate models, the better. But, does it mean we should include any competitive model regardless of other properties?

(2) Model assessment criterion

We have already mentioned that a model assessment method can be designed by combing a distance function and a generalizability estimation method. Whereas, how to choose distance function and generalizability estimation method for a certain problem is not clear yet. It also needs further investigation.

(3) Feature selection

The feature selection serves two purposes: dimensionality reduction and error elimination. By feature selection we are able to detect spurious features and get rid of them. Meanwhile, it can reduce model dimensionality with the minimum loss of valid information. We are not sure which feature selection methods are suitable to our cases.

(4) Feature integration

The composite model is a linear combination of features. Just as in weighted averaging, there might be many different methods to produce factor loadings. It is desired to be unbiased, efficient and robust.

(5) Uncertainty analysis

After one creates a new model, another important thing is to specify its precision, or commit uncertainty analysis. As we know, in model selection it consists of model uncertainty and parameter uncertainty. In model combination, how should we quantify it?

(6) Model Locality

As mentioned earlier, usually a real model may be suitable for use in a certain domain but not others due to its locality. It might be beneficial to partition the input space into some small regions and learn local models in each region individually. However, other problems arise. For example, usually divide-and-conquer technique tends to increase variance, and how should local models be combined to reduce this effect?

All the above issues will be discussed in the subsequent chapters.

Chapter 2

Candidate Model Choice in Feature-based Model Combination

2.1 Introduction

Models are approximations of a real system based upon some assumptions and simplification. They are widely used for prediction. Usually, for a certain phenomenon there are many different models available. A natural question arises from such a situation that how to maximize the model performance as allowed by information at hand. In practice, there are two different actions, namely model selection or model combination. Just as in safe-critical industry reliability can be enhanced through redundancy, model performance can also be improved by combination. The reasons and advantages of combining models have been discussed in detail in term of accuracy, stability, consistency as well as globality in the previous chapter.

Feature-based model combination is an indirect mode combination method, which is so devised as to improve the efficiency in aggregating information from candidate models and observations. In that method, candidate models are first mapped into a feature space, rather than combining candidate models directly as in weighted averaging, and then features are selected to build a new composite model. According to feature-based model combination, a model can be expressed as a linear combination of features, i.e.

$$f_c(x) = \sum_{j=1}^{m} \beta_j h_j(x), \tag{2.1}$$

where $h_j(x)$'s are features or factors and β_j's are factor loadings.

In such a scheme, each candidate model can be written as

$$f_i(x, \beta_i) = \sum_{j=1}^{k} \beta_{ij} h_{ij}(x) + \varepsilon_i(x), \tag{2.2}$$

where $h_{ij}(x)$ are features and β_{ij} are corresponding feature coefficients, $\varepsilon_i(x)$ is the additive error in the candidate model $f_i(x,\beta_i)$. Candidate models are approximations of the true model in the sense that they only incorporate a subset of features of the true model and the factor loadings are not precise. Distinct models might contain a different subset of features or have different factor loadings.

Using some feature extraction methods, we are able to extract features from a group of candidate models and then form a feature set to build a new composite model. However, before proceeding to perform model combination, the first practical problem one has to face is where and how to obtain a group of candidate models, which is common to any model combination method. Candidate models might come from a wide range of sources including scientific literature, results of manipulative experiments, personal experience, and scientific debate and so on. Some of them may be theoretical models from different schools of though, or empirical models learned based upon different data or using different algorithms. Some model combination methods like bagging and boosting treat the generation of candidate models as a part of themselves.

In view of rich model sources, it is usually not hard to obtain a number of competing models. However, another problem arises again, that is, should we include all these competing models? For example, the previous performance of a model is not satisfactory, two candidate models are very similar, and two models are created by the same modeler or learned using the same algorithm. Should we choose them without any discrimination? If not, how should we preselect candidate models? These questions are absolutely not new, but have been raised in various contexts in the literature, for example, in combining forecasts [Clemen and Winkler, 1985], classifier ensembles [Kuncheva and Whitaker, 2003] and neural network ensembles [Opitz and Shavlik, 1996].

In this chapter, we will discuss in detail how to choose candidate models for model combination and some criteria will be proposed.

This chapter is arranged as follows. In section 2.2, four criteria for choosing candidate models are presented. In section 2.3, bootstrap method is applied to estimate generalization error based on testing error, which is combined with expert judgment by Bayesian theorem. In section 2.4, both subjective judgment of model diversity and objective measure of diversity will be presented and at last combined with a utility function. In section 2.5 and 2.6, we will discuss how to test model independence based on data and how to make sure of completeness by finding the saturated number of candidate models. Finally, a stepwise forward candidate model choice procedure is put forward in section 2.7. A summary in section 2.8 concludes this chapter.

2.2 Choosing candidate models

Model combination is an effective way to improve model performance. However, how good a composite model can be depends on the choice of candidate models, because obviously how much information is contained in models and how they complement each other affect the efficiency of information extraction and information aggregation. In order to facilitate improving model performance, it is of great help to follow some criteria in choosing candidate models, which include competence, diversity, independence and completeness.

2.2.1 Competence

By competence we mean that each candidate model should be competent compared to their peers in term of model performance, rather than its absolute predictive accuracy. If a terribly inaccurate candidate model gets involved accidentally, it might make negative contribution and result in misleading features. At worst, the big error will ruin the composite model. As a simple example, let's consider the unweighted averaging

$$f_C(x) = \frac{1}{m}\sum_{j=1}^{m} f_j(x)\,. \tag{2.3}$$

Clearly, if one of the candidate models has big error, the expected error of the combined model will be much bigger than the average error of the rest candidate models.

If a candidate model has big errors, it is more likely because it contains spurious features. Although in feature-based model selection feature selection helps detect and remove errors, it might fail to do so due to small sample of data.

Therefore, if a model is rather weak compared to others, it should not be chosen as a candidate model. In order to ensure competence, one first needs to assess all potential models. This can be done based upon their historical performance, their relationship as well as testing. We will discuss model assessment in detail later on.

2.2.2 Diversity

Besides competence, diversity is another important criterion we need to take into account during preselecting candidate models. The role of model diversity in model combination is just like what diversity has in natural evolution.

If we view models as composed of two parts, reasonable part and error part

$$f_i(x) = g_i(x) + \varepsilon_i(x)\,, \tag{2.4}$$

where the reasonable part $g_i(x)$ is the valid part that reflects the truth in a correct way, while the error part $\varepsilon_i(x)$ is due to the modeler's bias and some errors. By diversity, we specifically refer to different valid model structure, $g_i(x)$. According to diversity criterion, we should choose candidate models as different as possible as long as they have good competence.

Just as in safe-critical industry where diversity plays an essential role in improving reliability, diversity in model combination is also central. As we know, the gain of accuracy in model combination comes from the fact multiple candidate models complement each other. It is clear that combining models is only useful if they disagree on some inputs. Obviously there is no more information to be gained from a million identical models than from just one of them.

Diversity can ensure that candidate models do not all fail in a certain situation. To see how this is advantageous, let's look at an example. Suppose we have two candidate models, one of which works badly within a sub-region I and the other deteriorates in the other sub-region II. Using weighted average method produces a composite model $f_c(x)=w_1f_1(x)+w_2f_2(x)$, with $w_1+w_2=1$, such that $f_c(x)$'s performance in both sub-regions will be not so bad. In an extreme case, we can set w_I=0 in region I and 1 in region II, which is certainly the optimal composite model we can have. This combined model works better than either of the candidate models as far as the whole input region is concerned. In this extreme case, weights are not constant over the input region. A possible solution is to partition the input region into two sub-regions.

In regression problems, mean squared error is generally used to measure accuracy, and variance is utilized to measure diversity. In the context of neural network, Krogh and Vedelsby [1995] show that the generalization error, e_{gen} ,can be expressed as $e_{gen} = \bar{e} - \bar{d}$, where $\bar{e}$ and $\bar{d}$ are the average squared error and diversity of the ensemble, respectively. According to this result, it is obvious that increasing the diversity while maintaining the average error of a group of candidate models leads to a decrease in the generalization error of the final composite model.

Diversity also helps reduce the problem of multicollinearity in direct model combination methods where

$$f_c(x) = \sum_{j=1}^{m} w_j \cdot f_j(x) , \tag{2.5}$$

in which w_j are individual coefficients, because $f_j(x)$'s become less linearly dependent.

From the angle of features, diverse candidate models might have different subsets of features. By combining these feature subsets, it is possible to come up with a more complete subset of features, which enables us to make a better approximation to the true model.

Thus, heterogeneity among candidate models is highly desirable. To ensure model diversity, one should try to collect candidate models from various resources, for example models built upon different theories, learned from different data or using different algorithms, or created by different modelers. The measure of diversity will be presented in detail in the coming sections.

2.2.3 Independence

By independence, we specifically mean each candidate model makes independent error. In other words, $\varepsilon_i(x)$ in equation (2.2) is independent of each other. In the literature, error independence is often included in model diversity. However, it is worth distinguishing diverse model structure and independent error, since we explicitly divide a model into two parts, namely reasonable part and error part. Throughout this chapter, model independence specifically refers to independent errors, unless stated otherwise.

As pointed out by Clemen and Winkler [1985], the reduction in variance or gain in precision by combining multiple dependent models will be significantly less than could be obtained if the models were independent. Hansen and Salamon [1990] and Krogh and Vedelsby [1995] also showed the model combination works best when the candidate models are fairly accurate but fairly independent in the errors they make. Therefore, error independence plays an important role in reducing both bias and variance. This point can be easily caught from some simple examples.

To see how combining model having different structures helps reduce model bias, we just take Bayesian Model Averaging (BMA) as an example. Suppose we have a class of competing models $f_j(x)$ with different model structures, each with independent bias $b_j(x)$ with E[$b_j(x)$]=0. We construct a composite model by weighted average

$$f_c(x) = \sum_{j=1}^{m} p_j f_j(x)\,. \tag{2.6}$$

Then, the bias of the combined model is $\sum_{j=1}^{m} p_j b_j(x)$, which tends to E[$b_i(x)$]=0 as m increases.

However, if the competing models have complete dependent bias, $b(x)$, and then the bias of the combined models is equal to $b(x)$, which does not reduce to 0 as m increases.

To illustrate the role of error independence in variance reduction, let's suppose there are m competing unbiased models, denoted as $f_j(x)$, j=1,...,m, having correlated errors of the same variance, $\mathrm{var}(f_j(x)) = v$ and $\mathrm{cov}(f_j(x), f_k(x)) = c$.

Again we create a weighted average model as in equation (2.6). Then the variance of the combined model is

$$\mathrm{var}(f_c(x)) = \sum_{j=1}^{m} p_j^2 v + \sum_{i=1}^{m} \sum_{j=1, j \neq i}^{m} p_i p_j c \,. \tag{2.7}$$

Clearly, if $p_j < 1/2$ and c=0, then

$$\mathrm{var}(f_c(x)) < v/2 \,. \tag{2.8}$$

Furthermore, if c<0, the variance of the combined model will be even smaller. On the other hand, if c>0, the variance of $f_c(x)$ can be either greater or smaller than v. Thus, it is important to ensure that candidate models make independent errors, or even negatively correlated errors.

2.2.4 Completeness

As implied by its name, completeness means we should choose as many diverse candidate models as possible. Again let's see from examples how bias reduction and variance reduction can benefit from completeness.

Suppose we have a class of competing models $f_j(x)$ with different model structures, each with independent bias $b_j(x)$ with E[$b_j(x)$]=0. We construct a composite model by unweighted average

$$f_c(x) = \frac{1}{m} \sum_{j=1}^{m} f_j(x) \,. \tag{2.9}$$

Then, the bias of the combined model is $b(x) = \frac{1}{m} \sum_{j=1}^{m} b_j(x)$, which tends to E[$b_i(x)$]=0 as m increases by the law of large number. According to the Markov's inequality, the larger the number of candidate models m, the closer $b(x)$ is to 0 in probability.

To illustrate how the number of candidate models variance reduction, let's suppose there are m competing unbiased models, denoted as $f_j(x)$, j=1,...,m, having uncorrelated errors of the same variance, $\mathrm{var}(f_j(x)) = v$ and $\mathrm{cov}(f_j(x), f_k(x)) = 0$.

Once again we create an unweighted average model as in the above equation (2.7), and then the variance of the combined model is

$$\mathrm{var}\left(f_c(x)\right) = \frac{v}{m} . \tag{2.10}$$

Clearly, the larger the number of candidate models m, the smaller the variance of the composite model.

From the angle of feature extraction, in order to make the set of features more complete, the more candidate models the better, because the union of more feature subsets can approximate the complete feature set better.

However, does it mean we should include all competitive models regardless of other properties? For example, in direct model combination example where

$$f_c(x) = \sum_{j=1}^{m} w_j \cdot f_j(x) , \tag{2.11}$$

according to the bias-variance tradeoff, although the model bias can be reduced by including more candidate models, the variance of estimated parameters w_j will rise because more parameters need to be estimated from the same data.

Therefore, given model diversity, there might be a saturated number of candidate models. In the context of aggregating multiple experts, Makridakis and Winkler [1983] and Clemen and Winkler [1985] demonstrate the diminishing marginal returns associated with large numbers of experts, which is further supported by Ferrell [1985] who suggests using three to five experts. This will be discussed in more detail.

2.3 Model assessment

Certainly, in order to keep competence of candidate models, first we should know how to assess a model, that is, to tell how good a model is. Naturally, the predictive accuracy or equivalently generalization error is a good criterion to evaluate the performance of a model. However, if we are in possession of only small sample, this can be incorrect and instable. Furthermore, unlike the usual case where the generalization error is estimated somehow based upon in-sample error or training error, here we have got to estimate it from small sample testing error, and therefore the problem is somewhat different and the usual methods can not be applied directly.

The possible solution to make model assessment correct and stable is to incorporate some prior information as well as employ some stabilization procedure. In this section, we will propose a method to combine subjective judgment with some empirical objective evaluation in model assessment. In fact, the concepts of objectivity and subjectivity have always coexisted uneasily in scientific disciplines. Although many scientists and engineers would argue that subjective judgment has no place in objective scientific endeavors, the reality is that some measure of subjective judgment is inevitable because appropriate empirical data are simply not always available to characterize everything quantitatively. For example, in model assessment objective quantification fails to incorporate a model's theoretical foundation. In the past several decades, numerous technical disciplines have recognized the role that expert judgment, in particular, plays in their fields and have engaged in formal studies of its use, for example, medical decision making, weather forecasting, climate change analysis, safety and reliability analyses for nuclear power plants, stock price forecasting, to name just a few.

2.3.1 Subjective judgment

As we argued, subjective judgment plays an important part in assessing models, because it helps incorporate some information that can not be characterized by empirical data and make model assessment robust to random small sample. Especially in some cases where it is too expensive or even impossible to collect observations, subjective judgment may be the only way to assess a model.

Usually, subjective judgments are expert opinions from domain experts. Such expert opinions can be made in terms of the following information:

(1) Theoretical foundation

Obviously, if a model is built on sound theoretical foundations, people tend to feel more confident in it.

(2) Interpretable structure

Without surprise, one would favor a model that has interpretable structure, because if one can understand it he will feel more assured. That is part of the reasons why theoretical models are often preferred over empirical models.

(3) Validation

If a model is well tested after calibration, the chance of large errors gets lower. Certainly, a validated model is more likely to have good performance.

(4) Past performance

The past performance of a model is another good information source based on which for an expert to judge a model. Record about a model's past performance can come from one's personal experience as well as public data set.

(5) The history of a model

The development history of a model can be helpful. Just for example, a newer model is likely to be better than older ones. This is because if there is no new information worth being incorporated, it is not necessary to create a new model. Furthermore, sometimes a new model is explicitly intended to replace old ones.

Subjective judgment can be made in the form of a point estimate of its generalization error. Alternatively, experts often provide these judgments in the form of probability density function of the expected error. In some other cases, expert opinions concerning the performance of a model can be summarized in a score.

2.3.2 Model error evaluation

Objectively the only way to evaluate the performance of a model is to compare it against observations, so the objective evaluation of model performance is to estimate the generalization error based on empirical validation errors. It is worth noting that in the present case the estimation process is different from the usual model selection procedure that occurs together with model training. This is because in the real world, models are already ready to use and their parameters are not tunable. It is almost always the case that the new test data set is different from the training data set, and therefore what we have is a problem of out-of-sample model testing. In the literature, sometimes this is called post-sample forecasting error or post-sample model validation [see e.g. Ashley, 1997].

Besides the above difference, there are another two challenges, namely small sample setting and noisy data or even the presence of outliers. The small sample of noisy data poses at least two problems. First, the test data set is not sampled evenly across the input space, and since each candidate model does not work uniformly well over the input space, consequently the comparative performance of models might vary with different test sets. Second, due to the random fluctuation in sampling noisy data the rank of models in terms of performance can be quite different from sample to sample, especially in small sample settings. Due to the limited amount of data, the testing error might depart from the expected error. In both cases, the new data set might not be representative sample of the cases that we want to generalize to and therefore the testing error will be biased, that is, deviate from the expected error.

A classical way to estimate generalization error is to penalize in-sample calibration errors by model complexity in terms of Occam's razor, for example Akaike's Information Criterion (AIC) [Akaike, 1973]. When estimating generalization error from testing error, likewise we can apply the principal of parsimony, but the measure of model complexity is different because the number of parameter cannot represent model complexity anymore. Rather, the functional form or functional roughness as in regularization can be utilized instead. In such a framework, the generalization error can be expressed as

$$[\textit{generalization error}] = [\textit{testing error}] + \lambda \cdot [\textit{model complexity}], \qquad (2.12)$$

where λ is a constant which specifies the weight of model complexity.

In this method, for models with the same testing error, the smoother a model, the smaller its generalization error. However, in practice it is quite hard to figure out what value λ, which indicates the importance of smoothness, should take,. Therefore, we would rather favor another simpler method.

An alternative possible way to improve small-sample accuracy and stability is to apply the method of bootstrap, which is a resampling technique specifically suitable for small sample situations. In this method, bootstrap replications are repeatedly sampled from the original data set with replacement according to its empirical distribution function. Bootstrap is an effective way to stabilize instable procedures [Breiman, 1996] by mimicking the random fluctuations in real testing data set. In addition, inference and estimates obtained using the bootstrap replications are often more accurate in small samples, in some cases dramatically so. [e.g. Freedman and Peters, 1984].

As for outliers, a useful approach to make estimate robust is to employ different loss functions just as in *M*-estimator [Huber, 1964], which reduces the influence of outliers. For example, in the presence of outliers we may prefer the absolute error to the squared error.

To apply the bootstrap method is pretty simple. Suppose the original data set is $\{(x_i, y_i)\}$, i=1,…,n. Bootstrap samples, denoted $\{(x_{bi}, y_{bi})\}$, are repeatedly drawn from the original samples by putting mass $1/n$ at each original data point, with the number of bootstrap replications equal to the original size n. Then, the testing error of the bootstrap data set can be evaluated as

$$e_j = \sum_{i=1}^{n} \left(f_j(x_{bi}) - y_{bi}\right)^2 . \tag{2.13}$$

The above procedure is repeated for 50 times and we equate the generalization error of model $f_j(x)$ to the average testing error.

2.3.3 Combining subjective judgment and objective evaluation

Bayesian updating is a natural way to aggregate information from different sources and thus improve estimate. It is often applied in combining multiple experts as well as empirical evidence, for example, see [Clemen and Lichtendahl, 2002], [Clemen and Winkler, 1999] and [Morris, 1977].

In Bayesian framework, both prior information and likelihood function are expressed in the form of probabilistic density function. According to the Bayesian updating formula, the posterior probability density function is obtained as

$$p_{X|D}(x \mid d) = \frac{p_X(x) \cdot p_{D|X}(d \mid x)}{\int p_X(x) \cdot p_{D|X}(d \mid x) dx}, \tag{2.14}$$

where $p_X(x)$ denotes the prior distribution of a random variable x, $p_{D|X}(d|x)$ refers to the likelihood of data d given x, and $p_{X|D}(x|d)$ is the posterior probability density function of x.

Therefore, the posterior distribution incorporates all the information we have, both prior information and empirical evidence, and thus Bayesian estimates can be made based on it. This method can be directly applied to our case, combining subjective expert judgment and objective evaluation. To illustrate how to use it, consider an example. Suppose a random variable x follows normal distribution $N(\mu, \sigma_x^2)$, where μ is the parameter to be estimated from samples $d=(x_1,\ldots,x_n)$. Furthermore, we know in advance μ is also normally distributed as

$$p_\mu(\mu) = \frac{1}{\sqrt{2\pi}\sigma_\mu} \exp\left(-\frac{(\mu-\eta)^2}{2\sigma_\mu^2} \right). \tag{2.15}$$

Applying the Bayesian formula yields the posterior distribution $X \mid D \sim N(\tilde{\mu}, \sigma'^2)$,

with $\tilde{\mu} = \dfrac{\dfrac{\sigma_x^2}{n}\eta + \sigma_\mu^2 \bar{x}}{\dfrac{\sigma_x^2}{n} + \sigma_\mu^2}$.

Based on the posterior distribution, the Bayesian point estimate of the location parameter μ is equal to $\tilde{\mu}$. Note that it is actually a weighted average of the prior mean η and the sample mean $\bar{x}$

$$\tilde{\mu} = w_1 \eta + w_2 \bar{x}, \tag{2.16}$$

where

$$w_1 = \frac{\operatorname{var}(\bar{x})}{\operatorname{var}(\mu) + \operatorname{var}(\bar{x})} \text{ and } w_2 = \frac{\operatorname{var}(\mu)}{\operatorname{var}(\mu) + \operatorname{var}(\bar{x})}. \tag{2.17}$$

In this example, both prior distribution and likelihood function are assumed to be Gaussian. In fact, generally it is also true that the posterior mean is a linear combination of the prior mean and maximum likelihood estimate. In this sense, Bayes estimate is a shrinkage estimator with the difference that the Bayes estimate shrinks towards the prior mean while in other shrinkage estimators like James-Stein type estimators towards zero.

A difficulty in applying Bayesian method is that we have to specify prior density and likelihood function in the form of probability density function, which is sometimes impossible without enough information. For example, it is usually hard to know the distribution of an expert judgment. In reality, experts often provide their judgments in the form of quantiles of the distribution (e.g., 5th, 50th, and 95th percentiles). In addition, we can not always know the error distribution, either. In fact, there is an easier method closely related to the Bayesian approach, which is called Bayesian Method of Moments (BMOM) [Zellner, 1994 and 1997]. It is specifically introduced to compute post-data densities for parameters when not enough information is available to formulate a likelihood function and a prior density, for which it is impossible to use Bayes' theorem. BMOM differs from traditional Bayesian analyses in that it is based on two weak assumptions about moment conditions of data without specifying the prior density and the likelihood function. In the BMOM approach, post-data moments of parameters given a data set are first calculated, and then, the maximum entropy approach is applied to choose a proper posterior density that maximizes entropy under the constraints of given moments. As derived in [Zellner, 1994], the posterior mean is equal to the sample mean under assumptions, where no prior information is incorporated.

To overcome these disadvantages, we would propose a more flexible method by extending the case with both normal prior and likelihood function to general ones. A generalized Bayesian estimate of the generalization error is constructed as a weighted average of prior mean μ_e from expert judgment and mean sampling error $\bar{e}$

$$GE = w_1 \mu_e + w_2 \bar{e} , \tag{2.18}$$

where variances are utilized to construct weights, i.e.

$$w_1 = \frac{\operatorname{var}(\bar{e})}{\operatorname{var}(\mu_e) + \operatorname{var}(\bar{e})} \text{ and } w_2 = \frac{\operatorname{var}(\mu_e)}{\operatorname{var}(\mu_e) + \operatorname{var}(\bar{e})}. \tag{2.19}$$

Heuristically, it is easy to understand the meaning of weights. As we know, the variance reflects confidence. The smaller is the variance, the smaller the expected error of an unbiased estimator. Likewise for experts, the smaller is the uncertainty in their judgments, the higher the confidence. Intuitively, an estimator with more confidence deserves higher weight. Furthermore, we can check some extreme cases. If there is no evidence but only subjective judgment, clearly the best guess of the expected model error is the expert opinion. On the other hand, if we have infinite number of samples, which is only an ideal case though, generalization error can be evaluated exactly only with data and expert opinion will be of no use.

Therefore, it does make sense to use confidence to control weights in generalized Bayesian estimate. Its biggest advantage lies in that it only requires the first and second moments without the necessity of specifying prior distribution and likelihood function, thereby making the problem much easier.

2.4 Model diversity measurement

It is essential to maintain diversity in combining multiple models. Model diversity can be analyzed both qualitatively and quantitatively depending on from which angle to assess it. Correspondingly, both subjective judgment and objective measures are necessary.

2.4.1 Subjective judgment concerning model diversity

Model diversity stems from many aspects. Some of them can not be directly characterized by real values. However, based on this information, a subjective judgment concerning model diversity can be made. In assessing model diversity, the following factors should be taken into account. Based on one's qualitative analysis, diversity scores denoted as $ds(f_i, f_j)$ can be assigned to models. Subjective diversity scores range from 0 to 1, with 0 meaning all the same.

2.4.1.1 Model source

By model source, we mean how models are created. Models built upon different theories, learned from different data or using different algorithms, or created by different modelers.

First, models having different underlying theories can be expected to fail in different situations. For example, in atomic physics atom models build on classical mechanics and quantum mechanics tend to have quite different behaviors.

Second, data and learning algorithms play an essential role in creating models, especially empirical or semi-empirical models. As we know, it is very likely that randomly collected data convey different patterns, which in turn result in varied models. For example, data sets from different small regions in the input space will lead to different local models. Similarly for learning algorithms, since they differ in their capabilities and properties of learning, they usually produce various kinds of models. For example, multiple linear regression using polynomials and nonlinear artificial neural network tend to spot different patterns in the same samples and the resultant models will have different behaviors.

Third, the importance of modelers in modeling is unassailable. Experts who are very similar (in modeling style, philosophy, access to data, etc.) tend to provide redundant information in modeling.

Therefore, model source is an important factor in assessing model diversity.

2.4.1.2 Model structure

Compared to model source, diversity in model structure seems more obvious. Model structure here refers to both functional form and parameter vector. If two models take on different function forms, for example multiplicative or exponential, clearly they will behave differently. Meanwhile, if two models have different parameter vector, they will have different model dimensionality in terms of the number of independent parameters and thus have different freedom in modeling.

Thus, in analyzing model diversity, it is central to consider model structures.

2.4.1.3 Error distribution

Model diversity can be also seen from error patterns. Usually, model error is only expressed in the form of mean squared error. However, two models with the same mean squared error can be pretty different in terms of error patterns in the form of empirical probability density function of error or exceedence probability of error.

The empirical probability density function of error is histogram of testing errors with mass $1/n$ on each sample point. One of its examples is shown in

Figure 2.1

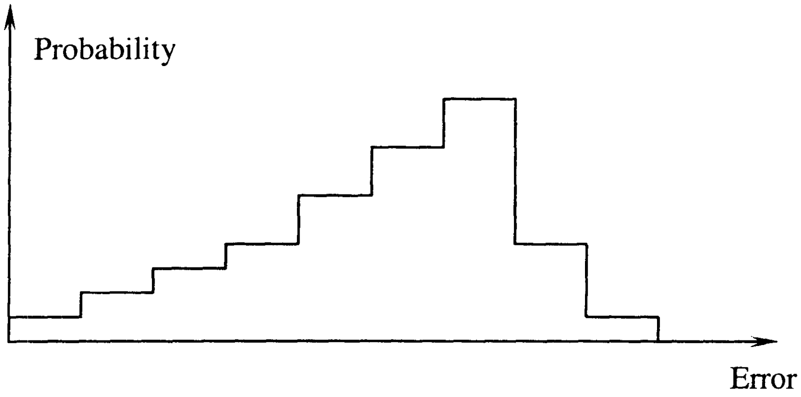


Figure 2.1 Typical empirical probability density function of error

The error distribution shows us how the errors are distributed and it helps identify the effect of outliers. Comparing the empirical probability density function of testing errors of two models, it is easy to see the difference in error patterns.

2.4.1.4 Localization

As argued in section 2.2.2, diversity in localization is also desired in model combination. Locality can be expressed in the change of the predictive error over the input space.

Usually, the generalization error is defined as

$$GE = \int E_{\varepsilon}\left(f_i(x) - f(x)\right)^2 p_X(x)dx = \int \left(g_i(x) - f(x)\right)^2 p_X(x)dx + \int E_{\varepsilon}\left(\varepsilon^2\right) p_X(x)dx \,, \qquad (2.20)$$

where $f_i(x)=g_i(x)+\varepsilon(x)$ is a candidate model and $f(x)$ is the true model. To analyze the localization, we will apply an idea similar to moving averaging of errors with regard to inputs x with the difference that all data points will be used and locally weighted average will be utilized to make the moving average smoother because only a small testing data set is available. In particular, a radial function, which is a function of the difference between data points, is used as the weighting function. At last, a localized error with regard to x is defined as

$$e(x_k) = \sum_{i=1}^{n} w_{ki}\left(f_j(x_i) - f(x_i)\right)^2 \text{, with weights } w_{ki} = \frac{|x_k - x_i|^2}{\sum_{i=1}^{n} |x_k - x_i|^2}. \qquad (2.21)$$

A typical example of such localized errors is shown in Figure 2.2.

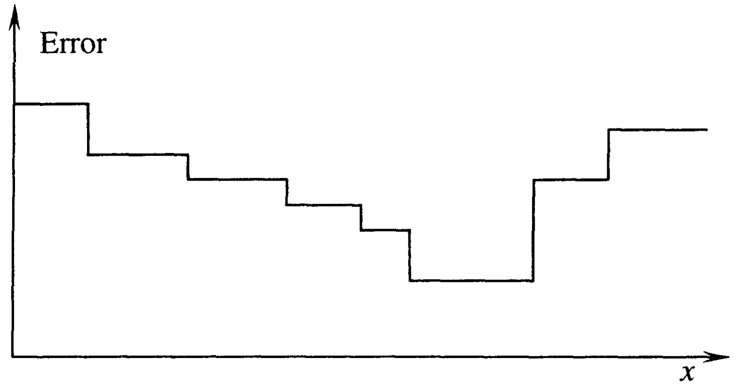


Figure 2.2 Localized average errors

Localized errors helps evaluate model performance over different regions, which in turn helps determine the property of localization of each model. With the aid of the graph of localized errors, it is easy to see how diverse models are in terms of localization.

2.4.2 Objective measure of model diversity

Besides subjective judgment regarding model diversity from aspects that can not be expressed explicitly in numbers, it can also be evaluated by some objective measures, for example discrepancy between two models. There are many different kinds of distance measures that quantify the dissimilarity between models, among which mean squared error (MSE) is the most widely used and also the simplest one. However, mean squared error only considers the mean distance and omits other information, thereby making it not sufficient for evaluating model diversity. For example, model A has the same MSE distance from two models B and C. However, suppose there is only a constant difference between model A and B while the difference between A and C is a complex function of x. Therefore, the similarity between model A and B and that between A and C is actually quite different.

The problem of MSE distance is that it does not take into account the difference in trends over x.

The dissimilarity in trends can be accounted for in at least two ways. One is the correlation between two models. Similar to covariance between two random variables, let's first define variance and covariance of functions. Suppose there are two models $f_i(x)$ and $f_j(x)$ defined on $x\in[a, b]$. We define the function variance as the sample variance with infinite number of samples

$$\begin{aligned}\mathrm{var}(f_i(x)) &= \lim_{N\to\infty}\left(\frac{1}{N}\sum_{k=1}^{N}\left(f_i(x_k)-E(f_i(x_k))\right)^2\right) = \lim_{N\to\infty}\left(\frac{1}{N}\sum_{k=1}^{N}\left(f_i(x_k)\right)^2\right)-\left(E(f_i(x))\right)^2 \\ &= \frac{\int_a^b f_i^{\,2}(x)dx}{b-a}-\left(\frac{\int_a^b f_i(x)dx}{b-a}\right)^2 = \frac{(b-a)\int_a^b f_i^{\,2}(x)dx-\left(\int_a^b f_i(x)dx\right)^2}{(b-a)^2},\end{aligned} \tag{2.22}$$

where the expectation is in turn defined as

$$E(f_i(x)) = \lim_{N\to\infty}\left(\frac{1}{N}\sum_{k=1}^{N} f_i(x_k)\right) = \lim_{N\to\infty}\left(\frac{1}{N\Delta x}\sum_{k=1}^{N} f(x_k)\Delta x\right) = \frac{\int_a^b f(x)dx}{b-a}. \tag{2.23}$$

Likewise, we can define the function covariance between two models

$$\begin{aligned}Cov(f_i(x), f_j(x)) &= \lim_{N\to\infty}\left(\frac{1}{N}\sum_{k=1}^{N} f_i(x_k)\cdot f_j(x_k)\right)-E(f_i(x))\cdot E(f_j(x)) \\ &= \frac{\int_a^b f_i(x)\cdot f_j(x)dx}{b-a}-\frac{\int_a^b f_i(x)dx\cdot\int_a^b f_j(x)dx}{(b-a)^2} = \frac{(b-a)\int_a^b f_i(x)\cdot f_j(x)dx-\int_a^b f_i(x)dx\cdot\int_a^b f_j(x)dx}{(b-a)^2}\end{aligned} \tag{2.24}$$

If $cov(f_i(x),f_j(x))=0$, $f_i(x)$ and $f_j(x)$are uncorrelated.

At last, the function correlation between two models can be defined as

$$corr(f_i(x), f_j(x)) = \frac{\mathrm{cov}\left(f_i(x), f_j(x)\right)}{\left(\mathrm{var}(f_i(x))\cdot\mathrm{var}(f_j(x))\right)^{1/2}}, \tag{2.25}$$

which takes values from -1 to 1.

It is clear that if $f_i(x) = f_j(x) + C$ with C a constant, $corr(f_i(x),f_j(x))=1$, in which $f_i(x)$ and $f_j(x)$ have exactly the same trend.

Another way to take into account the difference in trends is by the variance of error between two models, that is, $cov(f_i(x)\text{-}f_j(x))$. Obviously, if $f_i(x) = f_j(x) + C$ with C constant, $cov(f_i(x)\text{-}f_j(x))=0$.

In fact, these two methods are closely related except in different scales, because changing the geometric average to arithmetic average and division to subtraction in the definition of the correlation yields

$$\begin{aligned} corr2(f_i, f_j) &= \operatorname{cov}(f_i, f_j) - \frac{1}{2}\left(\operatorname{var}(f_i) + \operatorname{var}(f_j)\right) \\ &= \frac{1}{2}\left(\operatorname{cov}(f_i - f_j, f_i) - \operatorname{cov}(f_i - f_j, f_j)\right) = \frac{1}{2}\operatorname{var}(f_i - f_j) \end{aligned} . \tag{2.26}$$

However, since the error variance has the same scale as the mean squared error, it is easy to play with. Therefore, at last we can define the dissimilarity between $f_i(x)$ and $f_j(x)$ as

$$d(f_i, f_j) = E\left[(f_i(x) - f_j(x))^2\right] + \operatorname{var}(f_i(x) - f_j(x)), \tag{2.27}$$

which is positive and symmetric, similar to the bias-variance decomposition except that the first term is the mean squared error rather than the square of the mean error.

Earlier we defined the diversity as the difference between the reasonable part $g(x)$ of two models, rather than between outputs of two models. However, here we define diversity directly on outputs of two models. Actually it does matter in most situations, because if we suppose $f_i(x) = g_i(x) + \varepsilon_i(x)$, then

$$E(f_i(x) - f_j(x))^2 = E(g_i(x) - g_j(x))^2 + E(\varepsilon_i(x) - \varepsilon_j(x))^2 \tag{2.28}$$

where the second term is unknown but can be assumed to be constant.

In the above, we defined the dissimilarity between two models, or pair-wise diversity, which incorporates both mean distance and dispersion of distance. In practice, we care more about the diversity of a model ensemble, or group-wise diversity. Based on the pair-wise diversity, it is easy to define the group-wise diversity as the average diversity between any two models

$$D = \frac{1}{N(N-1)} \sum_i \sum_{j \neq i} d(f_i, f_j), \tag{2.29}$$

where N is the total number of models in an ensemble.

2.4.3 Combine subjective and objective diversity measures

Now we have both subjective diversity score denoted as $ds(f_i, f_j)$ and objective diversity measure denoted as $d(f_i,f_j)$ready and just like in evaluating model error we need to integrate subjective and objective information together. However, unlike in that case diversity score and objective diversity measure at the present are in different scales and Bayesian method is not suitable. This problem can be addressed by a utility function if we view it as a multi-attribute decision problem. Let's define final diversity as

$$div(f_i, f_j) = \exp(ds(f_i, f_j)) \cdot d(f_i, f_j)\,, \tag{2.30}$$

which is nonnegative and symmetric, monotonic with regard to both diversity score and diversity measure. In terms of so defined diversity, when two models are the same, $div(f_i,f_j)$ is equal to zero, and $div(f_i,f_j)$ can tend to infinity.

2.5 Test error independence

From the arguments and examples in section 2.2.3, it is seen that independence plays a similar essential role to that of diversity in model combination. Once again, here independence specifically refers to independence among total errors of models compared to the true model, that is, $e_i(x)=f_i(x)-f(x)$. Although the true model is unknown, we have data sampled from it. Therefore, the error independence can be tested by sample covariance. The smaller the covariance, the better.

Another problem is that data might be noisy, but this does not matter since we are not interested in absolute values of covariance, but rather in their difference, for example, the difference between $cov(e_i(x), e_j(x))$ and $cov(e_i(x), e_k(x))$. Suppose the data contains additive noise $\varepsilon(x)$, and we have data $y=f(x) + \varepsilon(x)$. Then, if we assume $\varepsilon(x)$ is independent of $f(x)$, we obtain

$$\mathrm{cov}\big(e_i(x), e_j(x)\big) = \mathrm{cov}\big(f_i(x) - f(x), f_j(x) - f(x)\big) + \mathrm{var}(\varepsilon(x)\,, \tag{2.31}$$

where the LHS can be estimated from samples directly and the second term on the RHS does not depend on models. Therefore,

$$\operatorname{cov}\left(f_i(x)-f(x), f_j(x)-f(x)\right)=\operatorname{cov}\left(e_i(x), e_j(x)\right)-\operatorname{var}(\varepsilon(x)) . \tag{2.32}$$

Thus, the existence of noise in data does not affect the difference between covariance.

According to results in section 2.2.2.3, the covariance is desired to be close to 0 or even better to be negative. If $\operatorname{cov}\left(e_i(x), e_j(x)\right) < \operatorname{var}(\varepsilon(x))$, then $\operatorname{cov}\left(f_i(x)-f(x), f_j(x)-f(x)\right) \leq 0$, which means uncorrelated or negatively correlated errors. In fact, like in regression model the noise level $var(\varepsilon(x))$ can be estimated in the model combination procedure.

Sometimes, we want to know the covariance between a model and a group of models. It can be defined as the average covariance between that model and every member in the group of models.

2.6 Saturated number of candidate models

In model combination, the model accuracy is improved through redundancy. The reason reduction works in improving model performance is because diversity and independence are also introduced at the same time. It is clear that without model diversity and independence redundancy alone can only worsen model performance. We suspect given the level of diversity, there exist a saturated number of candidate models, beyond which adding more candidate models does more harm than good. This suspension is supported by our empirical studies. There is also some evidence for this from the literature. For example, in the context of aggregating multiple experts, Makridakis and Winkler [1983] and Clemen and Winkler [1985] demonstrate the diminishing marginal benefits associated with large numbers of experts. Moreover, Ferrell [1985] suggests using three to five experts.

The existence of a saturated number can be explained heuristically by a tradeoff between redundancy and diversity. Generally, the more redundancy information a group candidate models contain, the worse in combining models; in the mean while the higher the level of diversity and independence, the better. Adding more candidate models tends to increase information redundancy as well as model diversity and independence. Beyond some point, the influence of redundancy increase dominates that of increase in diversity and independence, and thus adding more candidate models does more harm than good.

In fact, this redundancy-diversity tradeoff is equivalent to the well-known bias-variance tradeoff in that the more diversity, the smaller the bias of the resultant composite model because more valid features are included, while at the same time the more redundancy is introduced by adding more candidate models the higher the variance because more parameters need to be estimated.

According to both redundancy-diversity and bias-variance tradeoff, there is an optimal balance point. Currently, we are not able to provide some theoretical results concerning how to find the saturated number. However, it can be done by empirical test easily. In reality we can test if the generalization error decreases when adding one more candidate model to determine the saturation number. This test might tell us if it is worth adding one candidate model to the pool or not.

2.7 Candidate model choice procedure

In the above sections, we have already discussed some important criteria we'd better follow in choosing candidate models and how to quantify these criteria. Now we are ready to come up with a complete candidate model choice procedure. This procedure is a stepwise forward procedure beginning with an empty candidate model pool. Before describing it in detail, let's first propose an overall preference score of models incorporating model accuracy, diversity and independence. The definition of preference score is also in the form of a utility function as in multi-attribute decision making problems

$$PS(f_i(x)) = -a \cdot GE(f_i(x)) + b \cdot div(f_i(x), F) - c \cdot \mathrm{cov}(f_i(x), F), \qquad (2.33)$$

where coefficients a, b and c are all nonnegative, which specify the individual importance of accuracy, diversity and independence, $div(f_i(x),F)$ and $cov(\underline{f_i}(x),F)$ refer to average diversity and covariance, respectively, between a model $f_i(x)$ and the current candidate model pool F. This choice of additive utility function is because the measures of generalization error, diversity and independence have the same unit. It helps resolve conflicts among model accuracy, diversity and independence when choosing the next model.

Basically, the stepwise forward candidate model choice procedure follows the steps:

(1) Choose the model with the smallest generalization error $GE(f_i(x))$ and add it to the empty pool of candidate models, F.

(2) Pick the model having the largest preference score $PS(f_i(x))$ among the rest models.

(3) Combine models in the pool and evaluate the generalization error of the resulting composite model;

(4) If the generalization error gets larger than that of the previous composite model, then eliminate the newly added model and go to (5); otherwise, go to step (2);

(5) Combine models in the pool and return an optimal composite model.

It is seen from the above procedure that the testing of completeness is actually integrated with the model combination procedure.

2.8 Summary

Combining multiple models is a productive way to improve model performance. However, its efficiency highly depends on the choicc of candidate models. In this chapter, we proposed some desirable properties a group of candidate models should possess, which include accuracy, diversity, independence as well as completeness. To facilitate the choice with the use of these criteria, some quantitative measures are put forward. Meanwhile, Bayesian method and utility function are employed to aggregate information. Finally, a stepwise forward candidate model choice procedure is proposed to realize all these criteria in a procedure.

Chapter 3

Data-guided Model Combination by Decomposition and Aggregation

3.1 Introduction

A model, which is usually in a mathematical form, is a proposed explanation of a particular phenomenon. This proposed explanation is also used to predict future events. As more evidence is gathered by observing later on, that model can be validated against data. If the prediction error, i.e. difference between the prediction and the observation, is not tolerable, the model has be to calibrated or modified in model structure by incorporating new observed data. This cycle is then repeated again and again with more observations available until the model provides a satisfactory explanation of all events, which can be observed. Scientific progress requires that scientific models be updated as new observations show their deficiencies. As we see, both models and data play crucial roles in this progress.

Most often scientists bring up various models to explain a certain phenomenon from different angles, based upon varied theories, or due to the use of different sets of data. For example, in thermal hydraulics many different models were put forward to describe the behavior of two-phase flows and predict their pressure drops through a flow passage. Another example can be the probabilistic seismic hazard analysis, where many ground motion attenuation models were developed independently to predict the ground shaking given an earthquake. In such a situation, we are facing a thorny problem, namely, choosing the best model to predict the future events. In general, we may also have some, but often sparse, data at hand, and thus we can test the models against those data to select an optimal one. Several model selection methods and procedures have been developed to achieve this goal.

However, selecting a single best model is not so desirable as it does not make efficient use of those information at hand, e.g. a class of competing models and a set of new data. Therefore, alternatively model combination is proposed to further improve model performance. The benefits of model combination concerning augmenting model accuracy and reducing model uncertainty has been pointed out a lot of literatures [cf. Madign and Raftery, 1994 and Clemen, 1986]. As we know, model uncertainty, categorized into "epistemic uncertainty", stems from incomplete or imprecise knowledge and can be lowered by improvements in data measurements and model formulation. It is not surprising to see that combining different information sources including candidate models and data could result in a better model. Up to date several model combination methods have been proposed, for example equally-weighted combination and Bayesian Model Averaging (BMA) [Hoeting, 1999] or Bayes factor [Kass and Raftery, 1995] weighting method.

The basis idea behind model combination is to efficiently aggregate all available information, which, however, may contain errors or noises, and then build a new model somehow as good as possible. A good model combination method should have the following desirable properties:

(i) It is able to aggregate information in all competing models and therefore reduce model bias and uncertainty.

(ii) It should be able to detect errors in competing models in some degree, thereby reducing model bias.

(iii) It can model dependence among competing models and thus reduce information redundancy.

As pointed out by Hogarth[1987], the poor performance of human judges relative to statistical models stems largely from an inability to recognize and process redundant information appropriately. Furthermore, reducing information redundancy helps reduce model dimensionality, e.g. the number of factors in a factor model, and thus reduce model uncertainty.

(iv) It is able to combine different kinds of information, including models and data;

(v) It has robust performance when having different sets of data;

(vi) It is objective, involving no subjective judgment.

Ideally, a model selection process should be objective and therefore repeatable.

To achieve the above goals, basically improving both accuracy and precision, we will propose a new model combination method by means of decomposition and aggregation based upon data. This method is mainly suitable to the situations where there is no well-founded theory and only sparse data is available, because otherwise we may be able to derive a more exact theoretical model.

The paper is organized as follows. In section 3.2, a brief review about related work will be given. In section 3.3, dependence among candidate models will be analyzed using influence diagram and factor model will be proposed to model such dependence. In section 3.4, we will continue to talk about how to decompose a class of candidate models to factors. After that, section 3.5 will present a regression method to aggregate factors based upon data. Finally, a numerical example will be given in section 3.6 to show how this model combination method works.

3.2 Related work

Closely relevant problems to this chapter include model evaluation, model selection and model combination. Up to date lots of efforts have been devoted to these problems. In this section, we will briefly review some of them.

A model can be evaluated based upon how well the resulting prediction agrees with future observations [Dawid, 1984]. In the case where the same group of data is used for both model calibration and validation, model selection method or its variants are widely applied.

By now a variety of model selection methods have been developed, including classical hypothesis testing, penalized maximum likelihood, Bayes methods, information criteria and cross-validation. All these methods, which overlap with one another, provide an implementation of Occam's razor [Madign and Raftery, 1994] in one way or another, in which parsimony or simplicity is somehow balanced against goodness-of-fit.

Among those model selection methods, information criteria are considered to be novel and promising, and thus draw much attention. The name, i.e., information criterion, arise from its close connection to the information theory. This class of model evaluation and selection methods was pioneered by Akaike's Information Criterion (AIC) [Akaike, 1973], and afterwards many other similar information criteria were derived from different perspectives, for example, Bayesian Information Criterion (BIC) [Schwarz, 1978], Takeuchi's Information Criterion (TIC) [Takeuchi, 1976], Minimum Description Length (MDL) [Rissanen, 1978], Hannan and Quinn criterion (HQ)[Hannan and Quinn, 1979] and so on. Basically, all these criteria can be expressed as

$$IC = -2\log(Maximum\ likelihood) + penalty(k, n) \tag{3.1}$$

where maximum likelihood is the likelihood f(θ,x) evaluated at the maximum likelihood estimate $\hat{\theta}$, and penalty term is a function of the model dimension k, the number of model parameters, and the sample size n. From equation (3.1), it is easily seen that this class of methods can be viewed as modified maximum likelihood or penalized maximum likelihood methods.

All these approaches select the model that minimizes this quantity based on available data. The only difference between them lies in the second term, that is, different evaluation methods use different penalty terms as correction.

As we mentioned, in addition to model selection another class of approaches is model combination, which includes, for instance, equally weighted combinations, combinations based on information criteria evaluation, Bayesian model averaging.

Equally weighted combination is the simplest one in this class, because each model is assigned the same weight. This approach does not involve new data, and thus is usually applied in cases where there is no data available and all competing models have the same preference. When some data is gathered, this approach is ready to extended to a weighted combination. For example, each model can be evaluated using Akaike's information criterion (AIC) and assigned different weights based on their AIC value [Burnham and Anderson, 2002], for example

$$w_i = \frac{\exp(-\frac{1}{2}AIC_i)}{\sum_{j=1}^{K}\exp(-\frac{1}{2}AIC_j)}. \qquad (3.2)$$

A recently developed model combination method is Bayesian Model Averaging (BMA) [Hoeting, 1999] or Bayes factor [Robert, 1995] weighting, which became computationally possible since the invention of the Markov Chain Monte Carlo algorithm [Gilks, Richardson, and Spiegelhalter, 1998]. The basic idea of BMA is very straightforward, that is, to calibrate the probabilities of competing models using Bayesian updating method. After obtaining the posterior model probability, the composite model can be expressed as

$$f(y \mid D) = \sum_{i=1}^{K} f_i(y)\Pr(M_i \mid D), \qquad (3.3)$$

where D is the observed data, K is the number of competing models, $f_i(y)$ is the ith model, and according to the Bayesian formula the posterior probability $\Pr(M_i|D)$ can be calculated as

$$\Pr(M_i \mid D) = \frac{\Pr(D \mid M_i)\Pr(M_i)}{\sum_{i=1}^{K}\Pr(D \mid M_i)\Pr(M_i)}, \qquad (3.4)$$

where $\Pr(M_i)$ is the prior probability of model M_i. The difficulty of implementing BMA partly consists in the computation of the integral

$$\Pr(D \mid M_i) = \int \Pr(D \mid \theta_i, M_i)\Pr(\theta_i \mid M_i)d\theta_i, \qquad (3.5)$$

where $\Pr(\theta_i|M_i)$ is the prior density and θ_i is the vector of parameters of model M_i.

Another class of methods of model combination is Bayesian information- aggregation, which is also based upon the Bayesian method [Morris, 1977 and Clemen and Winkler, 1993]. Suppose θ is a continuous quantity to be estimated, and we obtain a group of estimates $x_1,\ldots,x_K$ from a class of competing models, say, $M_1,\ldots,M_K$, respectively. According to the Bayesian formula, the posterior distribution of θ is

$$\Pr(\theta \mid x_1,\ldots,x_K) = \frac{\Pr(x_1,\ldots,x_K,\theta)}{\Pr(x_1,\ldots,x_K)} = \frac{\Pr(x_1,\ldots,x_K,\theta)}{\int \Pr(x_1,\ldots,x_K,\theta)d\theta} , \tag{3.6}$$

where according to the Markov's property

$$\Pr(x_1,\ldots,x_K,\theta) = \Pr(x_K \mid x_{K-1},\ldots,x_1,\theta)\cdots\Pr(x_2 \mid x_1,\theta)\Pr(x_1 \mid \theta)\Pr(\theta) . \tag{3.7}$$

The central idea of these method lies in modeling the dependence among models, which is termed the conditional mean dependence assumption (CMDA) in [Clemen and Winkler, 1993], that is,

$$E(X_i \mid X_{i-1},\ldots,X_1,\theta) = \beta_{i,0} + \beta_{i,1}X_1 + \ldots + \beta_{i,i-1}X_{i-1} + \alpha_i\theta . \tag{3.8}$$

By the above equation (3.8), the knowledge about the information sources is incorporated in aggregration. Thus, if we know the distribution of X_i in advance, such as Normal, Student T, Logistic, Laplace, Gamma and Beta, we can obtain its conditional distribution $\Pr(X_i|X_{i-1},\ldots,X_1,\theta)$ with the expected value determined by equation (3.8). Finally, we obtain the posterior distribution of θ.

Unfortunately, none of the above methods can give us a satisfactory solution to the problem mentioned earlier. For example, the model selection methods of information criterion can only choose a single best model, the BMA method cannot model the dependence in model structure among candidate models, and Bayesian information-aggregation methods cannot incorporate information in new data. These weaknesses are part of reasons that motivated the research of this chapter.

3.3 Model structure analysis

To proceed, it is time to further clarify our problem. Suppose we have a set of competing models, denoted as $M_1,\ldots,M_K$, which can be expressed in mathematical forms as $f_1(x),\ldots,f_K(x)$, and gather a new set of data, i.e. $\{(x_i,y_i)$: $i=1,\ldots,n\}$, where x_i and y_i can be vectors of input variables and response variables, respectively. Now our question is

***Question*: Given a class of competing models and a set of sparse data, how can we construct a more accurate composite model with smaller uncertainty as well?**

Note that here we evaluate a model using both accuracy and uncertainty, which is certainly consistent with the goals mentioned in section 1.

Generally, it is convenient and beneficial to deal with modeling in a statistical way. In a statistical scheme, (x_i,y_i) can be interpreted to be generated by random variables, say, (X,Y), which can be carried out in two different angles. First, the input variable X can be viewed as a random variable with the probability distribution function (pdf), $f_X(x)$, which is the same as its sampling density. Usually, $f_X(x)$ can be assumed to be a uniform distribution, i.e. $U[a, b]$. Correspondingly, if $Y=h(X)$, the pdf of $Y, f_Y(y)$, is given by [Papoulis,1991]

$$f_Y(y) = \frac{f_X(x_1)}{|h'(x_1)|} + \cdots + \frac{f_X(x_n)}{|h'(x_n)|} + \cdots \tag{3.9}$$

where $x_1,\ldots,x_n,\ldots$ are the real roots of the equation $y=h(x)$.

Second, besides randomness in X, Y might have other sources of uncertainty, for example, Y is mapped from X by a random rather than deterministic function or Y includes random error or noise.

Therefore, a data set $\{(x_i,y_i): i=1,\ldots,n\}$ can be viewed as realization of a random process, and the same for models. Consequently, this model combination problem can be dealt with in a statistical framework.

In this section, we will first make two arguments, namely, (i) candidate models are dependent on each other and (ii) such dependence can be modeled using common factors, and then we will propose a model structure formula, base upon which an ideal model is obtained.

3.3.1 Model dependence

Intuitively, the dependence among competing models is obvious. First, each model is built on the basis of some theories and data, which may be available to all modelers. This means the competing models share the common, at least partly if not the whole, knowledge base.

Second, all candidate models have the same purpose, i.e. describe the same phenomena and predict the same future events. It is not surprising that different models produce very similar results, as every model is trying its best to approximate the same truth.

As we know, good approximating models, each representing a scientific hypothesis, in conjunction with a good set of relevant data can provide insight into the underlying truth. To explain information source and the dependence among competing models, it is useful to introduce influence diagram [Howard and Matheson 1984, Schachter 1986,1988], which offers a convenient graphical tool to model the dependence among different information sources. Figure 3.1 shows a typical example, where each circle or oval represents nodes, which can be the truth or the full reality, a theory, a set of data or a model, and each directed arc refer to conditional dependence between a chance node and a decision node, which conveys information from a node to another and implies causality. In this example, Model 1 is created based upon Theory 1 and Data 1; Model 2 is build upon Theory 1 and Data 2; and Model 3 is produced in view of Theory 2 and Data 2. The purpose of models is to approximate the truth and predict the future. In such a framework, the overlapping information source, including theory and data, serves as a vehicle for representing dependence among the models. Note that here we use theory to denote any set of statements or principles devised to explain a group of facts or phenomena, while model refers to mathematical models in particular. Furthermore, different theories are brought up to explain the same phenomena, termed as truth in Figure 3.1, and different sets of data are generated by the same true model, so in fact there also exists dependence even among different theories and different sets of data. Therefore, influence diagram gives us a clear idea where the information sources for modelers come from.

Such influence diagrams were also called "knowledge maps" by Howard [1989], when they are used to describe a modeler's knowledge about a particular system.

If we introduce the concept of "information", the purpose of a model is to express all the information we have in a more compact and understandable form. A set of data contains only a finite fixed amount of information. In addition, most often data include noise as well as information. In practice the part that cannot be explained is considered to be noise, which actually may contain useful information.

In reality, such dependence is quite common. For example, the correlation coefficients among economic forecasters are usually around 0.9 [Clemen and Winkler 1986, Figlewski and Urich 1983].

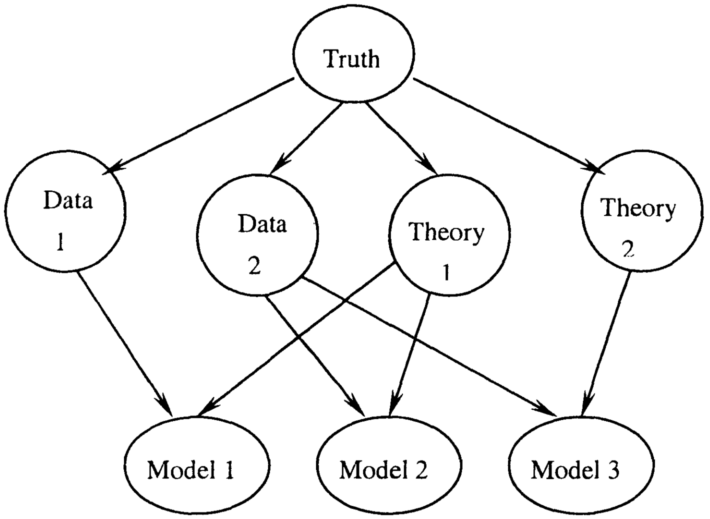


Figure 3.1 Influence diagram

3.3.2 Factor model

In the previous subsection, we have analyzed the dependence among candidate models, and now we will apply latent factors model to model such dependence. Our key argument is that the propagation of information beginning at the truth and ending up with models is just through latent factors, or components, and the dependence among candidate models is due to their sharing of common factors. This factor model is absolutely not a new idea, which has been applied in many areas. For example, factor analysis is widely used in such areas as psychology, chemistry and economics.

It is easy to slightly modify the above influence diagram into a factor diagram as in Figure 3.2. Note that in this factor diagram, information is characterized by factors and correspondingly each directed arc is associated with a set of pairs of factor and weight, i.e. $Q_{ij}=\{(f_{ij}, w_{ij})\}$. In such a scheme, information is propagated from the node of truth to the nodes of model in the form of factors, but it is obvious that the sets of factor received by candidate models are not necessary to be the same. The loss of information and misspecification can also be described in terms of factor.

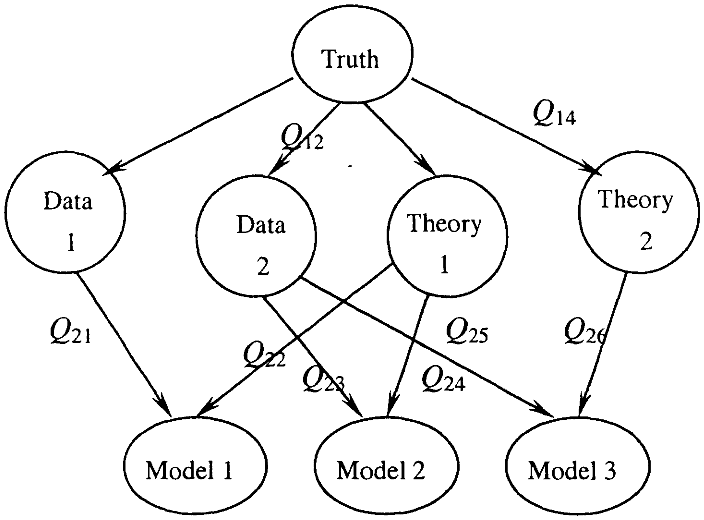


Figure 3.2 Factor diagram

3.3.3 Model structure

Now that we analyzed how the truth is reflected in data and theory and how the data and theory are incorporated into models by means of influence diagram and factor diagram, we are ready to analyze model structures of candidate models.

First, let's define the full truth or the true model. The observed data arise from the full truth, or in other words, generated by the "true model". In terms of factors, the true model can be expressed as

$$M_T(x) = \sum_{i=1}^{N} w_i(x) f_i(x), \tag{3.10}$$

where $f_i(x) \in F$, the factor set, and $w_i(x)$ is its corresponding weight, intensity, or factor loading as in factor analysis [See Bartholomew,1999], N is the number of total factors, x is an input variable. In a linear case, $w_i(x)$ is constant, independent of x. Actually, in a nonlinear case we can divide the range of input variables and approximate each subrange with a linear model. Therefore, in this chapter we will assume the true model is linear with respect to factors.

We believe that usually "truth" or full reality has essentially infinite dimension, i.e. N tends to infinity, and therefore it cannot be revealed with only finite samples of data and a limited set of candidate models. At best, we can only building a model providing a good approximation to the data available. Thus, a candidate model, an approximate representation of the system, can be expressed in a similar way, for example the kth candidate model

$$M_k(x) = \sum_{i=1}^{N_k} w_{ki}(x) f_{ki}(x) \quad , \tag{3.11}$$

with $N_k \leq N$, where $f_i(x) \in F_k$.

As we pointed out, a model is only a simplification or approximation of the reality and hence certainly will reflect the full truth in some degree. Whether a model is good or not depends on the quality of the data and the theoretical foundation that went into the modeling. In the factor model framework, the disagreement between a candidate model and the true model can be caused by the following:

(i) The set of factors contained in a certain candidate model is incomplete;

(ii) The factor loadings are imprecise, i.e. $w_{ki} \neq w_i$ for the same factor;

(iii) A candidate model incorporates an erroneous or spurious factor, i.e. $f_{ki} \notin F$.

3.3.4 Construct an optimal composite model

With the above model structure in mind, our question now is how we can construct an optimal composite model to overcome or mitigate the problems in candidate models, thereby meeting the goals to the extent permitted by the amount of information available.

Before we begin to construct such a composite model, let's have another look at the set of factors captured by a certain candidate model. As we mentioned earlier, on one hand candidate models depend on each other through factors and on the other hand the sets of factors may be different, which can be easily seen with the aid of the following Venn diagram as in Figure 3.3.

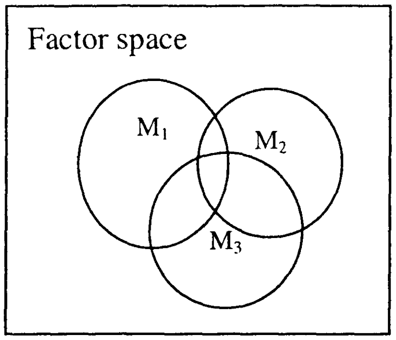


Figure 3.3 Factor space diagram

In light of this, it is convenient to divide a set of factors into two parts, i.e. common factors, which are shared by all the candidate models, and unique factors. Actually, unique factors are so called only in the sense that they are not shared by all the candidate models. This differentiation can be easily understood intuitively because candidate models are created based on common knowledge and individual intelligence. Therefore, every competing model can potentially contribute to the composite model. It is obvious that the union of $F_1,\ldots,F_K$ give us a better approximation to F than any single subset.

Meanwhile, as we pointed out a while ago, each candidate model may contain erroneous or spurious factors. For example, the unique factors may be attributed to personal bias and incorrect. Such bias or error is also what we must try to rule out in combination. At this point, data come to play their crucial role just as in a general modeling process data are used to calibrate and validate a model. The detection of erroneous factors is together with composite model construction.

Once a set of factors is ready, we can construct an optimal composite model by aggregating factors, i.e.

$$M(x) = \alpha + \sum_{i=1}^{N_c} w_i f_i(x)\,. \qquad (3.12)$$

In the above equation (3.12), the factor weights and constant αcan be determined based upon data. In the course of aggregating factors, whether a factor is valid or not is determined by its agreement with the data.

With these manipulations, the incompleteness, imprecision and error mentioned in the previous subsection can be reduced, and at last we can obtain an optimal composite model, which is closer to the true model than any other candidate model individually. But, we still have not solved two difficult key issues, namely,

(1) How can we extract factors from a class of candidate models? and

(2) How should we integrate factors and detect erroneous ones?

In the coming sections, we will propose some methods to attack these two obstacles.

3.4 Model Decomposition

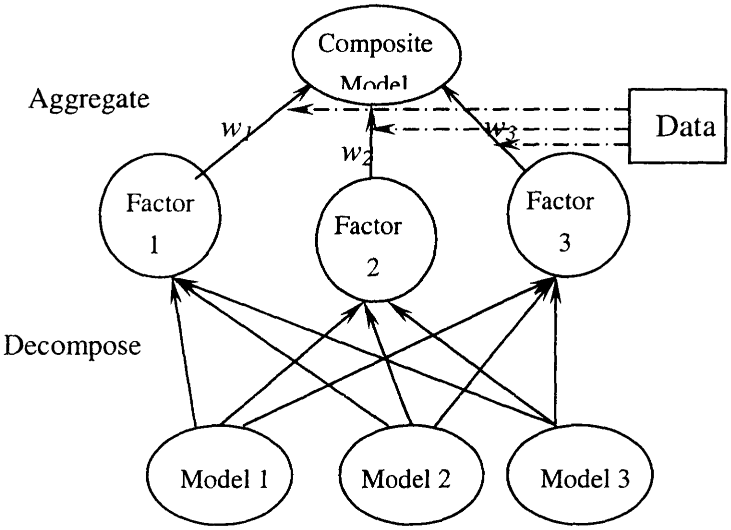


Figure 3.4 Model decomposition and aggregation

The process of model decomposition and aggregation can be described reversing the factor diagram in Figure 3.2. In Figure 3.2, the dashed lines means the pointed arc is under the guide of data.

In the above figure, it is clear that this method consists of two stages. In this section, we will propose different approaches to commit model decomposition and factor aggregation.

3.4.1 Model factor extraction

In section 3.3.3, we model the dependence among candidate models by means of common factors, and thus factors can be extracted from candidate models by taking advantage of this relationship. Before introducing factor extraction method, let's further clarify model structure and also assume some simplification to make it mathematically tractable.

If we fuse all unique factors of the kth candidate model into a single one, say, $f_{ku}(x)$, we obtain a simpler model structure as

$$M_k(x) = \sum_{i=1}^{Nc} w_{ki}(x) f_{ci}(x) + f_{ku}(x), \tag{3.13}$$

where $f_{ci}(x)$'s are common factors and $f_{ku}(x)$ is the single unique factor of kth candidate model, and N_c is the number of common factors extracted, and k=1,...,K. If we rewrite the above equation (3.13) in a matrix notation, we have

$$M = Wf_c + f_u, \tag{3.14}$$

where candidate model vector $M=[M_1,\ldots,M_K]^T$, common factor vector $f_c=[f_{c1},\ldots,f_{cN}]^T$, the unique factor $f_u=[f_{u1},\ldots,f_{uK}]^T$, and the factor loading matrix $W=[w_1,\ldots,w_K]^T$ with $w_k=[w_{k1},\ldots,w_{kN}]^T$. In particular, the equation (3.14) reduces to a linear transformation from common factors to candidate models when assuming no unique factors.

To further simplify, we might assume the unique factors follow the same probability distribution and independent just as in factor analysis, and then treat the average of the unique factors as another common factor and rewriting the equation (3.14) we obtain

$$M = Wf_c, \tag{3.15}$$

which is much simpler linear transformation. If W is invertible, we obtain

$$f_c = W^{-1}M. \tag{3.16}$$

By doing this, we significantly simplify the problem of factor extraction, although compromising some generality.

Actually, up to now we have not precisely defined dependence yet. In the following, we will define dependence in two different cases and propose a second-moment as well as a higher-order statistical method to perform factor extraction.

3.4.2 Principal Component Analysis (PCA)

As we know, if a variable X has a normal or Gaussian distribution, its distribution is completely determined by its mean value and variance. Furthermore, if X is vector of

Gaussian random variables, their joint distribution is also completely determined by the covariance matrix. Under this classical assumption of Gaussianity, being uncorrelated is equivalent to being independent statistically. Thus, dependence can be modeled only through pairwise correlation.

Principal Component Analysis (PCA) [Jolliffe, 1986 and Christensen, 2001] is the most commonly used subspace-related techniques for dimensionality reduction, filtering, data modeling. The basic idea of PCA is to find the components that can explain the maximum amount of variance of original variables, e.g. $M_1,\ldots,M_K$ in the current case. PCA can be defined in a recursive way as described below. The direction of the first principal component (PC) is so defined that the variance of the projection on that direction is maximized, i.e.

$$w_1 = \arg\max_{\|w\|=1} Var(w^T M) = \arg\max_{\|w\|=1} \left\{ E\left[(w^T M)^2 \right] - E\left[w^T M \right]^2 \right\} , \qquad (3.17)$$

where w is a vector of same dimension as M, and it is normalized. In this sense, PCA method is also termed Varimax rotation method. Then, the first principal component is given by $f_1 = w_1^T M$. After this, it finds the orthogonal direction with the second largest variation, or equivalently the principal component of the residual:

$$w_2 = \arg\max_{\|w\|=1} Var\left(w^T (M - w_1 w_1^T M) \right) . \qquad (3.18)$$

Or in general, after determining the first k-1 principal components, the kth component can be determined similarly as:

$$w_k = \arg\max_{\|w\|=1} Var\left(w^T (M - \sum_{i=1}^{k-1} w_i w_i^T M) \right). \qquad (3.19)$$

This continues until the dimension in the space is used up. The PCA is thus a rotation to new coordinates and a sorting with respect to the variance.

In practice, the computation of the w_i can be simply accomplished by the singular value decomposition of the (sample) covariance matrix of M, i.e. $\Sigma_M = E[(M-E(M))(M-E(M))^T]$. Σ_M can be decomposed as :

$$\Sigma_M = W^T \Lambda W \qquad (3.20)$$

where Λ is a diagonal matrix made of eigenvalues and W is made of eigenvectors, which correspond to $w_1,\ldots,w_k$ found by equation (3.17-19). Finally, principal components, or factors, are obtained as $f=W^TM$. It is easy to check that the covariance matrix of f is diagonal, which means factors are uncorrelated.

PCA can at least serve two purposes in our case. First, it helps reduce model dimension and reduce information redundancy, since the first several components, having the largest variance, contain most information. Second, noise or error may be reduced by removing the principal components ranking in the tail, which are more likely due to error or bias.

3.4.3 Independent Component Analysis (ICA)

For non-Gaussian variables, independence is not the same as uncorrelatedness; rather, uncorrelated variables are only partially independent. In probability or statistics, two random variables are considered to be independent if and only if their joint distribution is equal to the product of their marginal distributions. For example, $E(XY)=E(X)E(Y)$ if X and Y are uncorrelated; while $E(g_1(X)g_2(Y))= E(g_1(X))E(g_2(Y))$ holds for any arbitrary functions $g_1(\cdot)$ and $g_2(\cdot)$ if X and Y are statistically independent. Therefore, in the case of non-Gaussian random variables, much more sophisticated techniques have to be devised, which can incorporate the information of higher-order moments. Independent Component Analysis (ICA) is developed to meet this purpose.

Nowadays, lots of different ICA methods have been developed. In spite of such diversity, the basic idea of ICA remains the same, that is, components or factors $f_c = W^{-1}M$ are so determined that f_{ci} is independent of f_{cj} for $i\neq j$. The diversity of ICA methods is due to different independence measures and various optimization algorithms used to maximize these independence measures. Two widely applied independence measures are nongaussianity and mutual information [Hyvärinen, 1999]. Mutual information is a natural measure of the dependence between random variance, which is defined as

$$I(Y,X)=H(Y)-H(Y|X)=H(X)+H(Y)-H(XY), \qquad (3.21)$$

where X and Y are two random variables, and the entropy $H(\cdot)$ is defined as

$$\begin{aligned} H(X) &= \int \log f_X(x) \cdot f_X(x)dx \\ H(XY) &= \int\int \log f_{XY}(x,y) \cdot f_X(x,y)dxdy \ . \\ H(Y \mid X) &= \int \log f_{Y|X}(y \mid x) \cdot f_{Y|X}(y \mid x)dy \end{aligned} \tag{3.22}$$

Obviously, $I(Y,X)=0$ only when X and Y are independent.

The independence measure of nongaussianity is based upon the argument that nongaussian is independent. This is based upon the fact that a component will be uninteresting if it is random and a Gaussian random is the most random according to one standard measure of randomness, entropy [see Diaconis and Friedman, 1984]. The nongaussianity can be measured by differential entropy or negentropy, which is defined as

$$J(Y) = H(Y_G) - H(Y), \tag{3.23}$$

where Y is a non-Gaussian random variable and Y_G is a Gaussian random variable of the same variance as Y. A fundamental result of information theory is that a Gaussian variable has the largest entropy among all random variables of the same variance. Thus, the negentropy $J(Y)$ in equation (3.23) is always positive and is zero only when Y is Gaussian.

In fact, [Hyvärinen and Oja, 2000] shows that the above two measures are equivalent.

In practice, because of the difficulty of calculating it directly some approximation methods are applied to estimate the negentropy. A classical method approximating negentropy is to use higher-order moments [Jones and Sibson, 1987], for example,

$$J(Y) \approx \frac{1}{12}E\left[Y^3\right]^2 + \frac{1}{48}Kurt(Y)^2, \tag{3.24}$$

where the kurtosis of Y is defined by

$$Kurt(Y) = E\left[y^4\right] - 3\left(E\left[y^2\right]\right)^2 \tag{3.25}$$

Later on, [Hyvärinen, 1998] proposed a new approximation of negentropy as

$$J(Y) \approx \sum_{i=1}^{p} k_i \left(E[G_i(Y)] - E[G_i(Y_G)]\right)^2, \tag{3.26}$$

where k_i are some positive constants, both Y and Y_G are of zero mean and unit variance, and the functions $G_i(\cdot)$ are some nonquadratic functions. If we use only one nonquadratic function G, the above approximation becomes

$$J(Y) \propto \left(E[G(Y)] - E[G(Y_G)]\right)^2 . \tag{3.27}$$

By choosing G wisely, equation (3.27) can give us better approximations than equation (3.24). [Hyvärinen and Oja, 2000] suggested two nonquadratic functions:

$$G_1(u) = \frac{1}{a_1} \log\cosh a_1 u, \quad G_2(u) = -\exp(-\frac{u^2}{2}), \tag{3.28}$$

where $1 \leq a_1 \leq 2$ is some suitable constant.

Before applying any ICA algorithm, it is useful to do some preprocessing, which mainly includes centering and whitening. This preprocessing is actually very simple. Centering M, i.e. subtracting its mean vector $E[M]$, is to make M a zero-mean variable. Whitening M is to transform M to a new vector M' such that its components are uncorrelated and their variance equal unity. This can be accomplished by means of eigenvalue decomposition (EVD) of the covariance matrix $\Sigma_M = W^T \Lambda W$, just as in PCA.

Based upon the above discussions, maximizing the nongaussianity of $w^T M$ gives us one of the independent components. To accomplish this optimization, a FastICA algorithm is developed in [Hyvärinen and Oja, 2000]. The FastICA tries to find a direction, i.e. a unit vector w, such that the projection $w^T M$ maximizes nongaussianity measured by equation (3.27), in a fixed-point iteration scheme [Hyvärinen and Oja, 1997]. The basic steps of the FastICA algorithm is as follows:

1. Randomly choose an initial weight vector w ;
2. Let $w^+ = E\left[Mg(w^T M)\right] - E\left[g'(w^T M)\right]w$;
3. Normalize w^+ as $w = w^+ / \left\| w^+ \right\|$;
4. If not converged, go back to step 2.

Note that g is the derivatives of the nonquadratic functions G, defined as in equation (3.28),

$$\begin{aligned} g_1(u) &= \tanh(a_1 u) \\ g_2(u) &= u \exp(-u^2/2) \end{aligned}, \tag{3.29}$$

where $1 \le a_1 \le 2$ is some constant, often taken as $a_1=1$.

In the above convergence means that the old and new valued of w point in the same direction, i.e. their dot-product get close enough to 1.

The above algorithm is only for one unit, which can estimate one of the independent components. The algorithm for several units is very similar but needs some extra steps. To prevent different vectors from converging to the same maxima we decorrelation the outputs $w_1{}^T M, \ldots, w_n{}^T M$ after every iteration in a deflation scheme based on a Gram-Schmidt-like decorrelation, i.e., when we run the one-unit FastICA algorithm for w_{p+1} after estimating p independent components, or p directions $w_1, \ldots, w_p$, we subtract from w_{p+1} the "projections" $w_{p+1}^T w_j w_j$, $j=1,\ldots,p$ and then renormalize w_{p+1}:

1. $w_{p+1} = w_{p+1} - \sum_{j=1}^{p} w_{p+1}^T w_j w_j$,

2. $w_{p+1} = w_{p+1} / \sqrt{w_{p+1}^T w_{p+1}}$.

The above two steps are accomplished after every iteration step in one-unit FastICA algorithm.

In so doing, we can estimate all the independent components one by one. However, sometimes it may be desired to use a symmetric decorrelation [Karhunen, 1997]. This can be done by the classical method using matrix square root,

$$W = (WW^T)^{-1/2} W \tag{3.30}$$

where W is the matrix $(w_1, \ldots, w_K)^T$ and the inverse square root $(WW^T)^{-1/2}$ is obtained from the eigenvalue decomposition of $WW^T = U \Lambda U^T$ as $(WW^T)^{-1/2} = U \Lambda^{-1/2} U^T$.

With this decorrleation, the several-unit FastICA algorithm, fixed-point for equations, is obtained by replacing vector w in one-unit FastICA algorithm with a matrix W, thereby giving us all the dependent components in one time. In this chapter, we use $g_2(u)$ and a deflation method for decorrelation.

ICA is often considered a tool for explanatory data analysis. This is not surprising because cause can be defined in terms of conditional probability or dependence in some circumstance [see e.g. Ellery,1991 and Forster, 1984]. ICA is also efficient for redundancy reduction as each components (features) are independent from each other, i.e. they provide no information to predict one variable using another one [Deco and Obradovic, 1995]. Another wide application of ICA is in noise reduction. Such denoising capability of ICA was particularly noted in blind source separation [Jutten and Herault, 1991]. With these desirable properties, ICA can serve as a good tool for us to extract factors from candidate models.

3.5 Factor selection and aggregation

Applying the methods proposed earlier, we can extract factors from a class of candidate models. With factors ready, the next step is to select a subset of factors and integrate them based upon available data.

The factors will be aggregated in a linear form as shown in equation (3.3), in the same manner the candidate models are decomposed. The popular regression method will be used to estimate factor loadings. The basic idea of factor selection is to check if a factor is supported by the empirical data or not. In other words, if inclusion of a factor makes the resultant composite model worse, it is likely to be an erroneous one and should be ruled out. Some criterion will be introduced to accomplish factor selection.

As mentioned earlier, factor selection is not a separate activity that precedes the model calibration; rather it is a critical and integral part of model building. In the context of multiple regression analysis, it is specially known as variable selection.

3.5.1 Sorting factors

In factor selection and assembly, the importance rank of factors becomes an important issue, especially when the pool of factors is quite big. For example, suppose we have N factors, then the total number of subset of factors is equal to 2^N, which means we will have to compare 2^N possible composite models to choose the optimal one. However, if factors are ranked, a stepwise factor selection can be applied, which makes the procedure of factor selection and aggregation substantially easier and accordingly save computing costs dramatically, i.e. we, for example, have only N possible composite models. Besides, in so doing we are able to construct a sequence of subsets of factors, which are nested, and therefore some statistical model selection method can be applied.

The principal components are naturally sorted by their capability of explaining variance of original variables, or equivalently the variance of components. Typically, the variance of components, or the eigenvalues of the covariance matrix in equation (3.20), drops very fast. This implies a principal component contains more information about a system and therefore more important than those ranked behind it.

As pointed out by some authors [cf. Hyvärinen, 1999 and Cheung and Xu, 2001], one of the drawbacks of ICA is that components resulting from ICA are not sorted, all of which have the mean value zero and unit variance. Here we will propose two simple methods to order independent components (ICs) based on their contribution to reconstruction of original data, which is similar to the ordering of principal components.

The first method is based on the mixing matrix W as in equation (3.15). An assumption behind this method is that the candidate models are close to the true model and thus factors make similar contributions to both the true model and the candidate models. Meanwhile, we can expect that the larger the absolute value of an entry W_{ji} in the mixing matrix W, the greater the contribution the ith factor makes to the jth candidate model, because all the ICs have the mean value zero and unit variance. Therefore, we define as a component importance measure (*CIM*) the average coefficients of an IC in reconstructing the candidate models,

$$CIM_i = \frac{1}{K}\sum_{j=1}^{K}\left|W_{ji}\right|, \tag{3.31}$$

where $|\cdot|$ stands for the absolute value.

The second method is based on the sample correlation between a factor and the observed data. As we will see later on, the contribution of an IC to the composite model is determined by how it is supported by the data. Thus, it is reasonable to rank ICs based on their agreement with data. In this method, the CIM is defined as

$$CIM_i = \frac{1}{n}\sum_{j=1}^{n}\left(f_i(x_j) - \bar{f}_i\right)\cdot\left(y_j - \bar{y}\right), \tag{3.32}$$

where $\bar{f}_i = \frac{1}{n}\sum_{j=1}^{n} f_i(x_j)$ and $\bar{y} = \frac{1}{n}\sum_{j=1}^{n} y_j$, and $(x_j, y_j), j=1,\ldots,n$ are observed data.

The ordering of the importance of independent components can be verified using a little more complicated method. The rank checking can be accomplished both forwards and backwards. The forward method starts with an empty queue and ranks components based on their squared error reduction worth (ERW) ΔL_{2-}, that is, how much squared error defined in equation (3.37) is reduced by adding a certain component to a set of factors. Obviously, the larger the error reduction worth is, the more important a factor. In contrast, the backward method begins with a full queue and ranks factors based on their squared error achievement worth (EAW) ΔL_{2+}, i.e., how much squared error is increased by deleting a certain factor from a set of factors. Once again, the larger the error achievement worth is, the more important a factor. In our empirical study, it shows that the above methods give out consistent results.

However, we have to point out that the ranking of ICs is far from so simple. As we will note afterwards in our numerical study, the ordering of the non-dominant ICs according to the methods introduced above is quite subtle and even changes with regard to data, although the ordering of the dominant ICs is in good agreement with that resulting from the above component importance measures. Here, we define dominant ICs as those whose *ICMs* are significantly larger than that of others. Therefore, ICs can only be partially ranked in advance.

Another important issue is that unlike PCs ICs cannot be determined uniquely, as demonstrated in FastICA algorithm [Hyvärinen and Oja, 2000]. Applying different nonlinearity function *g* leads to a different group of ICs, and furthermore even with the same function *g*, different initial guess of *w* in iteration will also lead to varied optimum although those dominant ICs will remain very similar. Thus, it is beneficial to repeat ICA many times so as to choose a group of ICs which include as many dominant ICs as possible.

3.5.2 Model calibration

As in equation (3.3), a composite model of factors can be expressed as

$$M_c(x) = \alpha + \sum_{i=1}^{N} w_i f_i(x) = w^T f(x), \tag{3.33}$$

where $w=[w_1,\ldots,w_N]^T$ and $f(x)=[f_1(x),\ldots,f_N(x)]^T$.

In the above equation, $f_i(x)$ is a function of input variable x, and so is the composite model. Furthermore, $f_i(x)$, $i=1,..,K$, forms a set of orthogonal base functions. The factor loadings w_i are assumed to be constant over the range of input variable x. Now the task of model calibration is to estimate factor weights, i.e. w_i, given a data set $\{(x_i,y_i)\text{: } i=1,\ldots,n\}$. In the face of such a problem, a general solution is first to define a loss function and then design an algorithm to search for parameters such that minimize the loss function. The most widely used loss function is the mean squared error loss function, i.e.

$$L_2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - M_c(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - w^T f(x_i)\right)^2. \tag{3.34}$$

Thus, the estimated factor weights are

$$\hat{w} = \arg\min_{w} L_2 = \arg\min_{w} \frac{1}{n}\sum_{i=1}^{n}\left(y_i - w^T f(x_i)\right)^2. \tag{3.35}$$

The above equation can be solved analytically, and we obtain

$$\hat{w} = (F^T F)^{-1} F^T y, \tag{3.36}$$

where $F=[f_1,\ldots,f_N]$ with $f_i=[f_i(x_1),\ldots, f_i(x_n)]^{\mathrm{T}}$ and $y=[y_1,\ldots,y_n]^{\mathrm{T}}$.This is exactly the well-known Ordinary Least Squares (OLS) method.

With factor weights estimated, we are able to calculate the estimated mean squared loss as

$$\hat{L}_2 == \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{w}^T f(x_i)\right)^2. \tag{3.37}$$

If we denote a subset of factors as Γ, then we designate $\hat{L}_2(\Gamma)$ as the estimated squared loss of the composite model using all factors in Γ, whose factor weights are estimated by equation (5.6). It is easy to see that

$$\hat{L}_2(\Gamma) \geq \hat{L}_2(\Omega)\text{, for } \forall \Omega \supset \Gamma. \tag{3.38}$$

This means adding more factors will definitely reduce the estimated loss using the same data set as for calibration, but the capability of predicting the future data is not necessarily improved, or likely deteriorated. This phenomenon is called overfitting in statistical literature [Burnham and Anderson, 2002]. Literally, a selected model is said to be overfitted if it involves more factors than the true model or spurious factors. The danger of overfitting is that it tends to identify spurious features unique to a single data set and so calibrated model cannot be generalized. In contrast to overfitting, an underfitted model fails to identify effects or factor that are actually supported by the data set. Generally, a fitted model starts with underfitting and end up with overfitting with the number of variables increasing. Quantitatively, the prediction error decreases at first and goes up at last by adding more predictors. The balance point between underfitting and overfitting is considered optimal. To understand this, it is helpful to take a look at the bias-variance tradeoff.

Let's first define the expected prediction error at x as $E[(y\text{-}M_c(x))^2]$, which can be decomposed as follows:

$$\begin{aligned}E\left[(y(x)-M_c(x))^2\right] &= E\left[(y(x)-E(M_c(x))+E(M_c(x))-M_c(x))^2\right]\\ &= (y(x)-E(M_c(x)))^2+E\left[(E(M_c(x))-M_c(x))^2\right]\\ &= \{Bias(M_c(x))\}^2+Variance(M_c(x))\end{aligned}\quad, \tag{3.39}$$

where $E[M_c(x)]$ is the expected composite model given a certain subset of factors and given the sample size. The expectation and variance of $M_c(x)$ is with respect to observed data, because the composite changes from data set to data set.

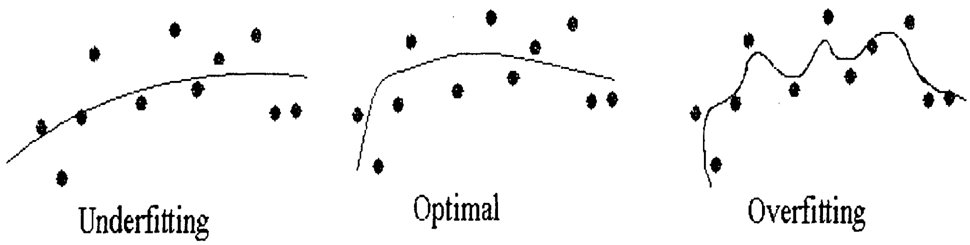


Figure 3.5 Models fitted to a set of data

In general, adding more factors can reduce the model bias, the first term, or in other words achieve better fit, but in the meantime model variance is increased because the sample size gets smaller relative to the number of model parameters to be estimated. In the case of underfitting, the bias in parameter estimation is generally substantial while the variance is underestimated. As for overfitting, the parameter estimation is usually free of bias but have large variance. In view of this trade-off, we need to find out a balance point in this tradeoff, which is considered optimal, thereby minimizing expected predictive squared error in the future. Figure 3.5 may give us an intuitive sense of the relationship between underfitting and overfitting.

3.5.3 Factor selection

Here factor selection is actually the same as variable selection in regression. However, since the factors have been already ranked in terms of their importance, the factor selection process gets much simpler. We will design a stepwise factor selection procedure to complete this.

Before designing factor selection procedure, let's first work out how to evaluate a composite model. To this end, we would apply some statistical model selection method or criterion. The first criterion we would use is Schwarz's Bayesian Information Criterion (BIC) [Schwarz, 1978], which is simple in computation and was proven to be consistent [Woodroofe, 1982]. Similar to a general information criterion as in equation (3.1), BIC is expressed as

$$BIC = -2\log L(\hat{\theta} \mid x) + k \cdot \log(n)\,, \tag{3.40}$$

where $\log L(\hat{\theta} \mid x)$ is the maximum log-likelihood of a model with k model parameters based on data $x = (x_1, \ldots, x_n)$, that is,

$$\log L(\hat{\theta} \mid x) = \sum_{i=1}^{n} \log f(x_i \mid \hat{\theta}), \tag{3.41}$$

and where $f(\cdot \mid \hat{\theta})$ is a conditional pdf and $\hat{\theta}$ is the maximum likelihood estimate of that model. In the current case, $\theta = (w_1, \ldots, w_k)$. In the case of linear regression model, under the assumption of Gaussian error the BIC can be derived as follows:

The likelihood

$$L(X, Y, \theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y_i - \hat{y})^2}{2\hat{\sigma}^2}\right] = \left(\frac{1}{\sqrt{2\pi\hat{\sigma}^2}}\right)^n \exp\left[-\frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{2\hat{\sigma}^2}\right]$$

$$\log(L(X, Y, \theta)) = -\frac{n}{2}\log(2\pi\hat{\sigma}^2) - \frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{2\hat{\sigma}^2} = -\frac{n}{2}\log(2\pi\hat{\sigma}^2) - \frac{RSS}{2\hat{\sigma}^2} \tag{3.42}$$

$$= -\frac{n}{2}\left[\log\left(\frac{2\pi}{n}\right) + \log(RSS) + 1\right]$$

where residual sum of squared error is defined as $RSS = \sum_{j=1}^{n}(y_j - \hat{y}_j)^2$, and $\hat{\sigma}^2 = \frac{RSS}{n}$.

Thus, we obtain the BIC as

$$BIC = -2\log(L(X, Y, \theta)) + k\log n = -n[\log(n/2\pi) - \log RSS - 1] + k\log n \tag{3.43}$$

In our case, the model dimension is equal to the number of factors plus 2 (one constant α and σ^2 are also estimated). According to this criterion, a model having a smaller BIC value is thought of as better than others with larger BIC values.

Another by far the most natural model evaluation method is Cross-Validation (CV). Cross-Validation can be used to estimate the generalization error or expected prediction error. In a simple version of cross-validation, the data set is divided into two parts: one part for model calibration, training set, and another part for model validation, test set. That is,

$$\hat{L}_2 = \frac{1}{L}\sum_{i=1}^{L}\left(y_l - \hat{w}^T f(x_l)\right)^2 , \tag{3.44}$$

where y_l are data points in the test set and L is the size of test set, and $\hat{w}$ is estimated using training data set.

As rule of a thumb, one-third of the data should be used for the purpose of validation. Although simple, this version requires that the data set be large. In a more complicated L-fold cross-validation scheme, the data set is randomly broken into L partitions, and then train on all the points not in the l-th partition with the l-th partition serving as test set, and at last find the test-set sum of errors. In L-fold cross-validation, the procedure of model calibration and cross-validation test should be repeated L times. In this version, the estimated generalization error is

$$\hat{L}_2 == \frac{1}{L}\sum_{l=1}^{L}\frac{1}{J}\sum_{j=1}^{J}\left(y_{lj} - \hat{w}_l^T f(x_{lj})\right)^2 , \tag{3.45}$$

where L is the total number of data partitions and J is the partition size, (x_{lj}, y_{lj}) are data points in the l-th partition, the model parameters $\hat{w}_l$ are estimated using the L-1 partitions of data excluding the l-th partition.

In the current situation of sparse data, L-fold cross-validation is more data-efficient, and thus a better choice. In this chapter, we set L as 5.

With model evaluation approach ready, now let's start to design factor selection procedure with supposing the factor $f_1,\ldots,f_K$ have already been ordered. The stepwise factor selection procedure starts with an empty subset of factors and let $k=0$, and then goes through the following steps:

(1) Add factor f_{k+1} to the subset and estimate a composite model M_{ck+1} by OLS;

(2) Evaluate the newly created model M_{ck+1} by BIC or cross-validation;

In the case of ICs, each unused IC can be a candidate for f_{k+1}, and therefore we have to try several different M_{ck+1} correspondingly and then choose the best one among them.

(3) If according to the above assessment, M_{ck+1} is worse than M_{ck}, then stop; otherwise go to step (1).

(4) $f_1,\ldots, f_k$ are selected as good factors and correspondingly M_{ck} is considered to be the optimal composite model.

If we apply cross-validation, the above step (4) is slightly different. That is, after an optimal subset of factors is determined, we will use the whole data set, instead of L-1 partitions, to fit a composite model, M_{ck}.

By such a procedure, we can avoid exhaustive combination of factors and thus save computation cost. In such a stepwise factor selection, those factors ranked in the tail have much slighter chance to be included in the optimal subset of factors, which seems reasonable. Because those factors assigned smaller importance have smaller contribution to explain the observed data and are more likely to be corrupted by noise or error.

3.6 Numerical results

By now we have already developed the data-guided model combination method by decomposition and aggregation, and in this section we will demonstrate its performance with both artificial and real examples.

3.6.1 An artificial example

In the following we will first use an artificial example to illustrate the model combination method and also show how it works. Based on Monte Carlo simulation results, some general conclusions will be drawn. For the purpose of demonstration, we would like to use an artificial example, where the true model is supposed to be known.

Suppose for some certain system the true model is already known and a set of observations are also obtained somehow. Let's assume the true model can be expressed in mathematics as

$$\begin{gathered} y(x) = 150\text{-}150\exp(\text{-}2x) + x^2\text{-}0.1x^3 + 4x + \\ 30\exp(\text{-}x/3)\cdot\sin(x) + 15\sin(1.5x)\text{-}20\ln(x+1) \end{gathered}, \tag{3.46}$$

where the real number $x \in [0,10]$.

Correspondingly, its realistic data generative model can be written as

$$y = y(x) + \varepsilon , \tag{3.47}$$

where ε is supposed to assume a normal distribution, i.e. $N(0,\sigma^2)$, where σ^2 is set as 64 in the current example. From this generative model, we gathered a set of data with the sample size n=20, i.e. (x_i, y_i), where x_i is evenly distributed within [0,10]

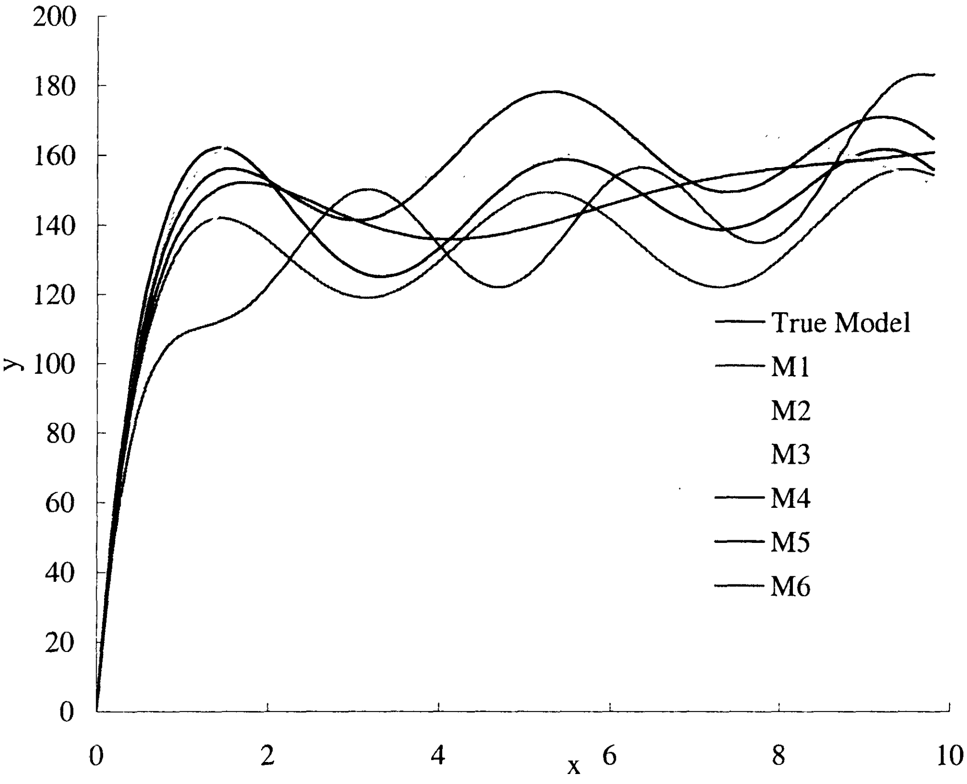


Figure 3.6 Artificial candidate models

Meanwhile, suppose we also collected a class of competing models, all of which can predict the system behavior to a similar degree.

$$\begin{aligned}
&y_1(x) = 150\text{-}150\exp(\text{-}2x) + 4x + 15\sin(1.5x)\text{-}20\log(x+1);\\
&y_2(x) = 150\text{-}150\exp(\text{-}2x) + x^2\text{-}0.1x^3;\\
&y_3(x) = 150\text{-}150\exp(\text{-}2x) + x^2\text{-}0.1x^3 + 30\exp(\text{-}x/3)\cdot\sin(x);\\
&y_4(x) = 150\text{-}150\exp(\text{-}2x) + 6x + 30\exp(\text{-}x/3)\cdot\sin(x)\text{-}20\cdot\log(x+1);\\
&y_5(x) = 150\text{-}150\exp(\text{-}2x) + x^2\text{-}0.1x^3 + 15\sin(1.5x);\\
&y_6(x) = 150\text{-}150\exp(\text{-}2x) + 15\cos(2x)\text{-}15 + 0.004x^2;
\end{aligned} \tag{3.48}$$

Note that each candidate model is either incomplete or erroneous, or both. Now we will apply our model combination method to derive a composite model, which will give us better prediction of future data.

If we plot both the truth and the candidate models in a single figure as in Figure 6.1, it is seen that each candidate model does approximate the true model in some degree.

Next we will apply both PCA and ICA to extract orthogonal factors from the class of candidate models. The mixing matrices obtained in PCA and ICA are as follows:

$$W_P = \begin{bmatrix} 0.3572 & -0.2513 & 0.6192 & -0.2198 & -0.1678 & -0.5911 \\ 0.4335 & -0.0932 & -0.42 & 0.5593 & 0.3426 & -0.4436 \\ 0.4119 & -0.2443 & -0.4771 & -0.1913 & -0.701 & 0.123 \\ 0.3888 & -0.0061 & -0.1883 & -0.6863 & 0.5695 & 0.1339 \\ 0.4326 & -0.3134 & 0.3968 & 0.3563 & 0.1096 & 0.6468 \\ 0.4202 & 0.8776 & 0.1403 & 0.0656 & -0.1636 & 0.0497 \end{bmatrix}$$

and

$$W_I = \begin{bmatrix} -0.1349 & -0.2476 & 0.1194 & -0.0169 & 0.2302 & 0.0390 \\ -0.0977 & -0.0689 & -0.0261 & 0.0319 & 0.1082 & 0.0056 \\ -0.1817 & -0.1311 & 0.0182 & 0.1508 & 0.1524 & -0.0279 \\ -0.0720 & 0.1390 & -0.3287 & 0.2578 & 0.0731 & -0.0681 \\ -0.2512 & -0.1761 & 0.0450 & 0.0580 & 0.2294 & 0.0673 \\ -0.3456 & -0.1843 & 0.0539 & 0.0397 & 0.3815 & 0.0039 \end{bmatrix},$$

respectively, and corresponding separating matrices are the inverse of mixing matrices and thus factors $f=W^{1} M$.

The resultant principal components and independent components are shown in Figure 3.7and Figure 3.8, respectively.

As for independent components, before proceeding we'd better rank them first. Based upon the mixing matrix W_I, the ICs can be ordered as IC2, IC1, IC4, IC3, IC5 and IC6 according to the labels in Figure 3.8. This is consistent with the result from the backward approach shown in Table 3.1. According to this ordering, IC6 is the least important one, but as we mentioned earlier, the ordering of those non-dominant ICs, such as IC3, IC5, and IC6, is in fact rather subtle.

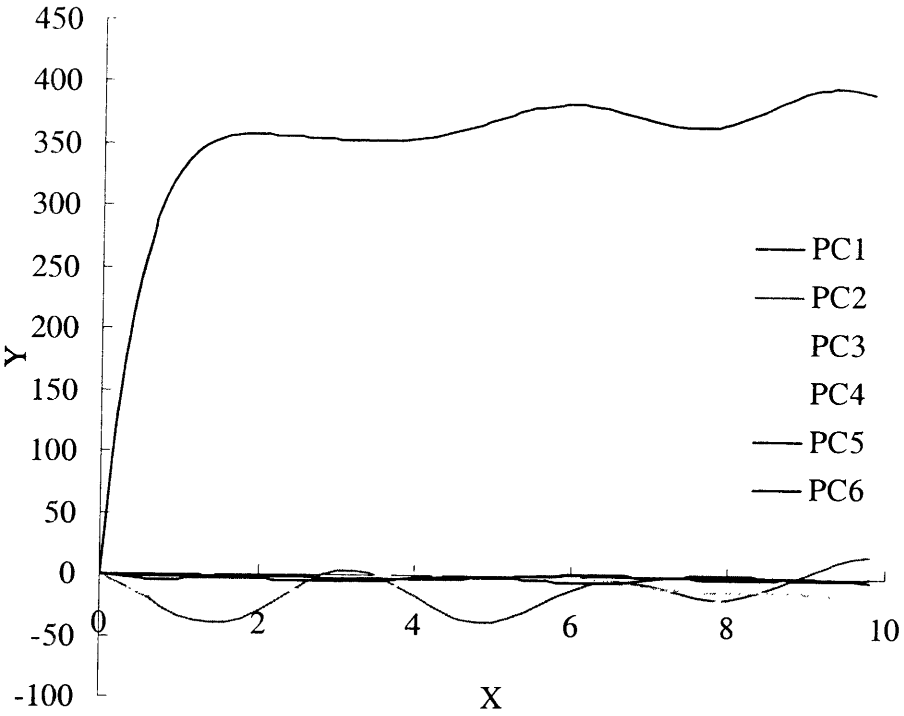


Figure 3.7 Principal components

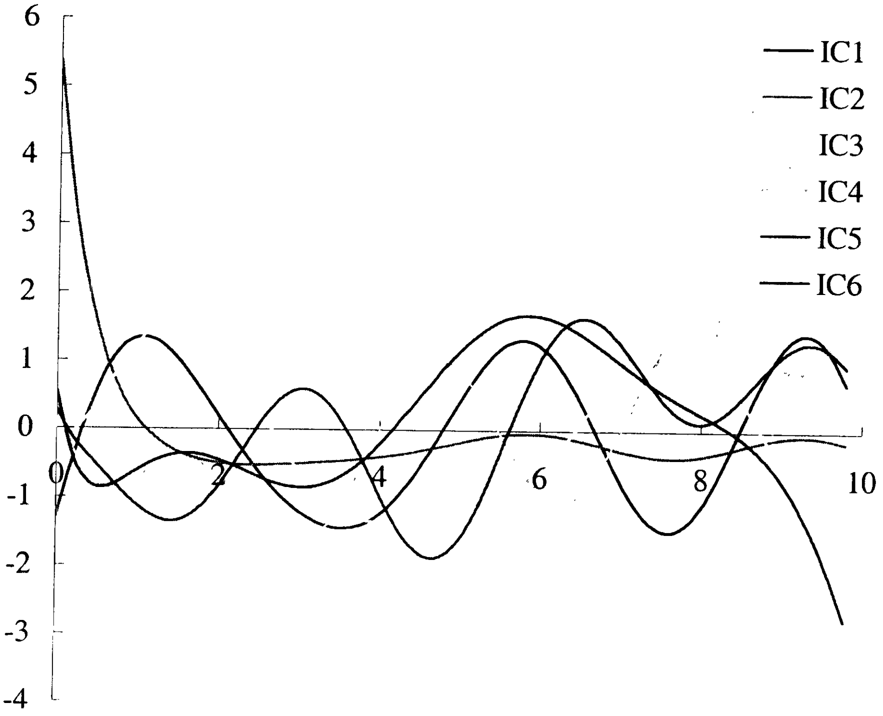


Figure 3.8 Independent components

Table 3.1 Residual sum of squared errors of composite models

Models	-IC2*	- IC1	- IC4	- IC3	- IC5	- IC6
RSS	19838	3665	1278.7	717.2	679	671.3

* Sign minus means that IC is excluded.

Alternatively, if we regress on data using all the six factors, we obtain:

$$y = 143.97 + 12.0926\cdot IC1 - 24.4966\cdot IC2 - 0.6979\cdot IC3 + 5.6838\cdot IC4 - 1.5551\cdot IC5 + 0.3299\cdot IC6$$

Therefore, the ordering of the estimated factor loading is the same as the result implied by Table 3.1, but this provides us a simpler way.

After factor ordering, we are ready to construct a composite model by aggregating factors. The result is shown in Table 3.2

Table 3.2 Evaluation of composite models

PC models	1 PC	2 PCs	3 PCs	4 PCs	5 PCs	6 PCs
RSS	2418.8	1585.4	955.2	671	670.2	669.5
k	3	4	5	6	7	8
BIC	161.7	156	149	145	148	151
CV	108.8	58.9	49.9	23.8	22.4	22.6
GMSE	107.8	67.9	40.9	17.7	17.1	18.3
IC models	1 IC	2 ICs	3 ICs	4 ICs	5 ICs	6 ICs
RSS	4289.2	1354.8	728.9	681.4	671.3	669.5
k	3	4	5	6	7	8
BIC	173.1	153	143.6	145.3	148	151
CV	179.4	21.48	12.2	16.2	18.8	22.5
GMSE	204.5	35	16.1	16.6	19	18.3

Note that the IC models listed in the above table are the best ones among those having the same number of factors.

Table 3.2 also shows the evaluation of composite models using both BIC and Cross-Validation together with global mean squared error (GMSE), which is computed against the true model as

$$GMSE = \frac{1}{m}\sum_{i=1}^{m}\left(y(x_i) - M_c(x_i)\right)^2, \tag{3.49}$$

where m is so large as to give us a satisfactory approximation.

Based upon this information, an optimal composite model can be determined for both cases. As for the PC models, 4 PCs should be selected in terms of BIC and 5 PCs should be chosen in terms of CV to be optimal; while for the IC models, both BIC and CV suggest that 3 dominant ICs form the optimal subset. Since the true model is supposed to be known, and thus we can compare our composite models directly with the truth, which tells us that the optimal numbers of factors are 5 and 3 for PC model and IC model, respectively. Therefore, both model evaluation methods, say BIC and CV, are acceptable, except that BIC chooses the second best option in the case of PC models.

In addition, the optimal IC model has better performance, or smaller GMSE, than the optimal PC model, this may tell us that ICA is potentially more efficient in factor extraction. What is more, the optimal number of ICs is 2 less than the optimal number of PCs, which means ICA is more efficient in information redundancy reduction.

Finally, let's compare the performance of the two composite models with any single model. Before comparison, let's suppose a candidate model can also be calibrated based upon data in the following way:

$$M_i'(x) = a_i + b_i M_i(x). \tag{3.50}$$

After such calibration, we found that the best calibrated model had GMSE 40.4, which is much greater than the two composite models. This means combination does improve the model performance.

In the above, we demonstrated the performance of this new method by an example, but usually in statistics a single specific case might not be so meaningful. Thus, in order to get a general result the above procedure is repeated a great number of times with different training data sets by the means of Monte Carlo simulation. The average errors of resultant composite models are listed in Table 3.3, for two different factor extraction methods and with different sample sizes.

Table 3.3 Monte Carlo simulation results of average errors

Sample Size	All Models	PCA	ICA
20	24.95	19.82	13.18
50	9.78	9.62	7.30

From the above table, we can draw some conclusions. First, the new model combination method outperforms the simple linear combination of all models. Second, ICA leads to better composite models than PCA. Third, the smaller the sample size, the more effective the new method is and also the more advantageous ICA is relative to PCA.

3.6.2 Real example

In the above subsection, we demonstrated our method using some toy data. Now let's apply to a real example and see if it works.

The real example we use here is the attenuation models in seismology. In this example, the purpose is to build a more accurate composite model which is applicable to south California in the United States. A sample data set of size 102 is obtained from the literature [Steidl and Lee, 2000]. Correspondingly, the candidate attenuation models include the attenuation relations by Boore et al. [1997], Sadigh et al. [1997], Abrahamson and Silva [1997], Campbell and Bozorgnia [1997], Spudich et al. [1997] and Idriss [1995]. All of these attenuation relations may be found in Seismological Research Letters, Volume 68, Number 1, January/February, 1997. All these attenuation relationships were developed for shallow crustal earthquakes in active tectonic regions, and thus they should be applicable to southern California.

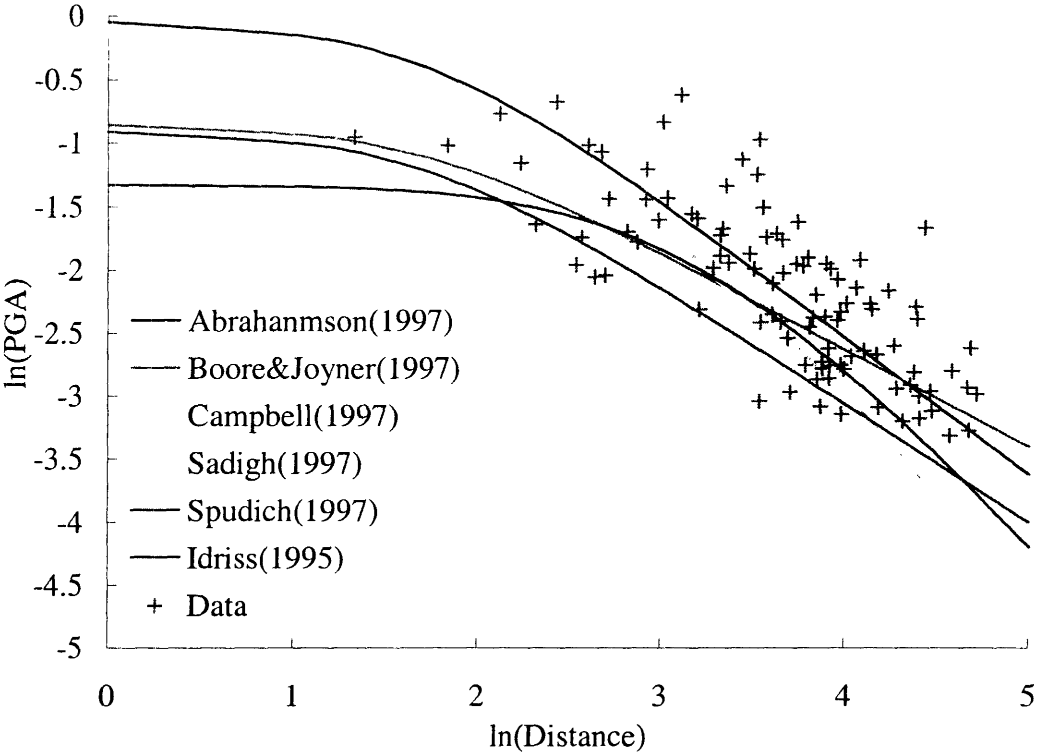


Figure 3.9 Candidate attenuation models and data

Both the candidate models and the sample data are plot in the same Figure 3.9. From Figure 3.9, it is easy to note that all the models are close to be a straight line, which means unlike the artificial example the dependence among candidate models are mostly linear.

Table 3.4 Comparison of models

	A single best model	Linear combination of all models	New method with PCA	New method with ICA
Test error	0.1935	1.63	0.146	0.140

Once candidate models and sample data are ready, we apply the same procedure as in the artificial example to combine candidate models under the guide of sample data, namely decomposing candidate models, selecting factors and aggregate factors into a composite model by multiple linear regression method. In order to evaluate the resultant composite model, the cross-validation is applied, in which two-thirds of the data set is used for model calibration and the rest one third is used to test the model. The results are shown in Table 3.4, where the test error is the mean squared error.

In this example, the same conclusion can be drawn that this new method outperforms both selecting a single best model and simple linear combination of all models. Meanwhile, ICA seems work better than PCA again. However, it is noteworthy that since the non-uniqueness of FastICA, ICA is committed several times and the best one is chosen. Compared to the artificial example, the advantage of ICA over PCA is not so significant in the current case. In fact, this observation is in agreement with our expectation. In general, the advantage of ICA compared to PCA is to incorporate higher order nonlinear dependence, but in the current case the models look like straight lines and there is only very little if not without at all nonlinear dependence. As a result, ICA simply reduces to PCA. Therefore, in cases where more nonlinear dependence is involved, the strength of ICA will be more significant.

Meanwhile, in the case of ICA, three independent factors are chosen while with PCA four uncorrelated factors are used. This once again verifies our expectation that ICA is more efficient in information compression, which leads to less valid factors.

3.7 Conclusions and discussion

In this chapter we developed a model combination method by taking advantage of the factor extraction, denoising and information redundancy reduction capability of both PCA and ICA. By some numerical results, we also show that this method works. But, some problems still remain unsolved, which include

(1) Nonlinear factor loadings, which depend on input variables x. For example, factors play different roles over the range of input variables. In our current method, we only suppose the linear assumption is valid based upon our assumption that the candidate models are similar

to the true model in some degree. To taking into account this nonlinear possibility, a design of mixture of composite local model will be helpful [Jacobs, 1991 and Jordan, 1994].

(2) Unique factors issue. In our current method, we reduce the general model structure to a linear transformation by treating equally weighted unique factors as another common factor, but obviously in so doing we may loss some useful information. To address this problem, a hierarchical model structure may help, which is similar to hierarchical factor analysis [Schmid and Leiman, 1957 and Ghahramani and Hinton, 1997].

(3) Explanation of factors. In our above discussion, although we extracted some factors from a class of candidate models, we have no idea what these factors are physically, or what effects they proxy for. Although explanation of factor entails knowledge about a specific system, it will help us interpret factors and further refine a composite model, for instance factor selection.

(4) Factor selection. Factor selection is always a tough job just as in general model selection. Although our numerical study both BIC and CV seem satisfactory, we need a more robust factor selection method, thereby helping reduce model uncertainty.

(5) Composite model uncertainty. Model uncertainty is key to the performance of a model. How to reduce such model uncertainty is a current active research problem.

Solving these problems can further refine this method or extend it to more general cases. These remaining issues serve as our works in the following chapters

Chapter 4

Shrinkage-based Fuzzy Regression Variable Selection Strategy

4.1 Introduction

In science, it is a central task to develop models to explain the observed data and predict future events. Usually, a model, which includes some free parameters, is first proposed to be true and then is fit to a set of data. When there exist multiple competing models, the problem of model selection arises naturally, that is, which model is the best for the future use. In the case of multiple linear regression models, the purpose of model selection is specially to select an optimal subset of predictor variables or regressors such that construct a model with the smallest expected prediction error.

Up to now, a variety of model selection criteria or strategies have been developed, which include classical hypothesis testing, penalized maximum likelihood, Bayes methods, information criteria and cross-validation. All these methods, which overlap with one another, somehow provide an implementation of Occam's razor [Madign and Raftery, 1994], in which parsimony or simplicity is somehow balanced against goodness-of-fit.

Basically, the goal of model selection is to minimize the expected predictive error. To this end, an alternative way is to apply shrinkage estimator, which is to balance the well-known tradeoff between Bias and Variance, thereby minimizing the overall prediction error. Such shrinkage estimators include ridge regression [Hoerl and Kennard, 1970 and Tikhonov and Arsenin, 1977], LASSO [Tibshirani, 1996] and negative garotte [Breiman, 1995], all of which add some kind of constraints on regression coefficients to the Ordinary Least Square (OLS) method.

As pointed out by many authors, for example, see [Miller, 1984] and [Breiman, 1996a], in linear regression the predictive error for subset selection is considerably larger than that for ridge regression, or put in other words, subset selection is unstable, while such shrinkage estimator as ridge regression is stable. Here, a procedure is considered unstable if a small change in the data can cause large changes in a calibrated regression model. It is also well known that regression models resulting from subset selection procedure suffer from selection bias [Miller, 1984].

On the other hand, model selection delivers a more interpretable framework than shrinkage estimators, especially when the predictor variables have certain corresponding physical meanings.

To combine the individual advantages of the above two classes of methods in a linear regression context, we will propose a shrinkage-based fuzzy model selection strategy, which is intended to reduce predictive error and provide an interpretable final model as well.

This chapter is organized as follows. In section 4.2, a fuzzy variable selection scheme will be proposed by generalizing the classical model selection method. Section 4.3 will define an effective model dimensionality for the fuzzy case. In section 4.4, a method of generalized shrinkage will be given to estimate coefficients in a fuzzy regression model. Right after that, a numerical optimization algorithm will be introduced to estimate individual shrinkage parameters numerically in section 4.5. Section 4.6 will present some numerical simulation study results. Finally, a brief summary will be given in section 4.7.

4.2 Fuzzy variable selection

In a typical regression situation, we have a set of predictor variables $X_1,\ldots,X_p$ and a response variable Y, and also gather a group of data, say (x_i,y_i), where $x_i=(x_{i1},\ldots, x_{ip})$. A multiple linear regression model is usually in the form

$$Y = \alpha + \sum_{i=1}^{K} \beta_i X_i \,. \tag{4.1}$$

If we further assume that X_i's are orthonormal, that is, $E(X_i)$=0 and the $cov(X_i,X_j)$=0 for $i \neq j$ and 1 for $i=j$, the above equation (4.1) reduces to

$$Y = \sum_{i=1}^{k} \beta_i X_i = X^T \beta \,, \tag{4.2}$$

where $\beta = (\beta_1, \ldots, \beta_k)^T$ and $X=(X_1, \ldots, X_k)^T$.

In fact, assuming the orthonormality does not lead to the loss of generality, because we can always make them orthonormal by applying Principal Component Analysis (PCA) or Singular Value Decomposition (SVD) and scaling if for any $i \neq j$ $cov(X_i, X_j) \neq 1$ holds, that is, $X=U \cdot Z$, where $Z=(Z_1, \ldots, Z_k)^T$ is not orthonormal and U is a rotation matrix. Therefore, in this chapter we will focus on orthonormal regressors.

If we apply the quadratic loss function, correspondingly we can define the empirical Sum of Squared Error (*SSE*) as

$$SSE = \sum_{i=1}^{n} \left| y_i - x_i^T \beta \right|^2 , \tag{4.3}$$

where $x_i=(x_{i1}, x_{i2}, \ldots, x_{ik})^T$.

Likewise, we can define the expected squared Prediction Error (*PE*) as

$$PE = E\left[(Y - X^T \beta)^2\right]. \tag{4.4}$$

The regression coefficients, i.e. β, in equation (4.2) is usually estimated by the Ordinary Least Square (OLS) method, which minimize *SSE* as in equation (4.3). According to OLS, it is easy to obtain that

$$\hat{\beta}_{OLS} = \left(x^T x\right)^{-1} x^T y , \tag{4.5}$$

where x is a $n \times k$ matrix, i.e. $x=(x_1, \ldots, x_n)^T$, and y is a column vector $y=(y_1, \ldots, y_n)^T$.

It is easy seen that the *SSE* is monotonically decreasing with adding more regressors, but our final purpose is to minimize the expected Prediction Error, which can be decomposed as

$$PE = E\left[(Y - X^T \hat{\beta})^2\right] = E\left[(Y - X^T E(\hat{\beta}))^2\right] + Var(X^T \hat{\beta}) , \tag{4.6}$$

where the first term is due to bias and the second is due to variance of estimated regression coefficients. In general, adding more regressors can reduce the model bias, the first term, or in other words achieve better fit, but in the meantime model variance is increased because the sample size gets smaller relative to the number of model parameters to be estimated. This is what is termed bias-variance tradeoff in the literature.

In view of this tradeoff, the task of model selection is to find out an optimal subset of predictor variables to reach a balance point. Because of this, we treat model selection and variable selection as interchangeable in the case of linear regression. In classical model selection, or variable selection in a linear regression case, there are total 2^p possible subset choices, which can be represented by a vector $\gamma = (\gamma_1,...,\gamma_p)$, where γ_i=1 means the *i*-th regressor is chosen while γ_i=0 means it is not.

With this notation, a regression model can be generally expressed as

$$Y = \sum_{i=1}^{k} \gamma_i \hat{\beta}_i X_i, \quad \text{with } \gamma_i\text{=1 or 0,} \tag{4.7}$$

where $\hat{\beta}_i$'s are specially estimated by OLS given γ.

In any classical model selection strategy, a γ is selected based upon some criterion such that the expected Prediction Error is expected to be minimized. If we denote the full set of predictor variables as U and likewise the selected subset of predictor variable as U_{opt}, then each regressor X_i, i=1,...,p, in U is either included in or excluded by U_{opt}. That is, U_{opt} is a crisp set. In this sense, we refer all the classical model selection methods as crisp model selection.

The γ can be also interpreted as follows: if γ_i=1, the *i*-th regressor, i.e. X_i, makes its full contribution to the explanation of the response variable Y; in contrast, if γ_i=0, X_i makes no contribution to explaining the response variable Y. In other words, γ controls the contribution of all predictor variables. Therefore, γ can be termed contribution factor.

As mentioned earlier, many researchers found that such crisp model selection methods were instable and the predictive error was remarkably large [see Miller, 1984 or Breiman, 1996a]. Such instability can be easily observed in Monte Carlo simulation, where different data sets result in different optimal subsets of predictor variables. To overcome the instability of model selection and improve its accuracy, Breinman [1996b] proposed to stabilize model selection using Bagging [Breiman, 1996a], which is an example of general P&C technique [Breiman, 1996c]. In this chapter, we explain the instability from another angle and propose a new variable selection method.

Heuristically, it is quite understandable that the instability of classical crisp model selection method is partly due to drastic change of γ_i between 0 and 1. Moreover, we may doubt that the contribution factor of a certain regressor changes from 1 to 0 just because a small change in the observed data. Even intuitively, that a regression has either full or no contribution should not always be the case and it is pretty reasonable that some regressor may have partial explanational capability. Thus, in practice we want a continuous factor to control contribution of regressors.

Following this line of thought, we propose a fuzzy variable selection strategy, which is quite different from those crisp variable selection methods in that it chooses a fuzzy subset of predictor variables instead of a crisp one. The elements of this fuzzy subset are the same as those of the full set of predictor variables, i.e. *U*, but with an associated membership function $\mu_U : X \rightarrow [0,1]$, which is in the current case discrete and maps each regressor to a membership grade, or degree of belonging. Let's denote a membership grade as m_i, which is continuous in [0,1]. In practice, for convenience we remove a certain regressor from the fuzzy set if its membership grade is small enough.

Like the γin a crisp case, we use the membership grade m_i to control the contributions of regressors as in

$$Y = \sum_{i=1}^{p} m_i \hat{\beta}_{OLSi} X_i \text{, with } m_i \in \ [0,1], \tag{4.8}$$

where each m_i functions as a continuous shrinkage factor since it is less than 1. It is clear that the crisp model selection is actually a special case of this general fuzzy framework with m_i either 0 or 1. We term this general model as a fuzzy model.

As we mentioned earlier, the ultimate performance measure of a model is its expected Prediction Error, and thus we need to find

$$\hat{m} = \arg\ \min_{m} E\left[(Y - \sum_{i=1}^{p} m_i \hat{\beta}_i X_i)^2\right] = \arg\ \min_{\beta} E\left[(Y - \sum_{i=1}^{p} \beta_i X_i)^2\right] \text{ ,}$$

$$\text{subject to } \sum_{i=1}^{p} m_i \leq \lambda \text{ ,where } \ 0 \leq m_i \leq 1 \ . \tag{4.9}$$

where the constraint 0≤ m_i≤1 simply follows the definition of m_i.

In reality, the expectation *E*[·] is unknown, therefore we have to estimate those parameters *m* in other practical ways, such as model evaluation methods that will be discussed later on.

In above discussion, we learned that the expectation *E*[·] can be minimized at some balance point in the Bias-Variance tradeoff. Therefore, the problem is actually how to find out such a balance point. Generally, the bias can be controlled by shrinking the size of the regression model space somehow. Corresponding to the two key elements determining the regression model space, namely predictor variables and regression coefficients, two different existing techniques can meet this goal. One is the crisp variable selection that has been already mentioned. Another popular technique is shrinkage estimate or regularization, which imposes constraints on the regression coefficients in some form. This technique works very well in practice. To illustrate this, it is well-known that shrunken regression estimator can have smaller future prediction error [Copas, 1983]. In fact, in the case of crisp model selection it is obvious that $\sum_{i=1}^{p} m_i \leq p$, which can also be interpreted as some kind of shrinkage. Therefore, in order to parameters *m*, it is plausible to add another constraint to least square method, that is, to minimize

$$SSE = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 \tag{4.10}$$

under some additional constraints. For example, the crisp model selection can be viewed as adding a certain constraint, i.e. γ_i=0 or 1 for i=1,...,p, under which the equation (4.10) is minimized. Some other different kinds of constraints can be found in the literature, some of which is imposed directly on *m*, for example, $\sum_{i=1}^{p} m_i \leq t$ where $m_i \geq 0$ as in Breiman's non-negative garotte [Breiman, 1995], and others are imposed directly on β, for instance, $\sum_{i=1}^{p} \beta_i^2 \leq t$ as in standard ridge regression [Copas, 1983] and $\sum_{i=1}^{p} |\beta_i| \leq t$ as in Tibshirani's LASSO [Tibshirani, 1995]. Seemingly, our fuzzy model is closest to Breiman's non-negative garotte with an additional constraints $1 \geq m_i \geq 0$ among all existing methods. In following, let's see how our fuzzy variable selection is related to these existing methods.

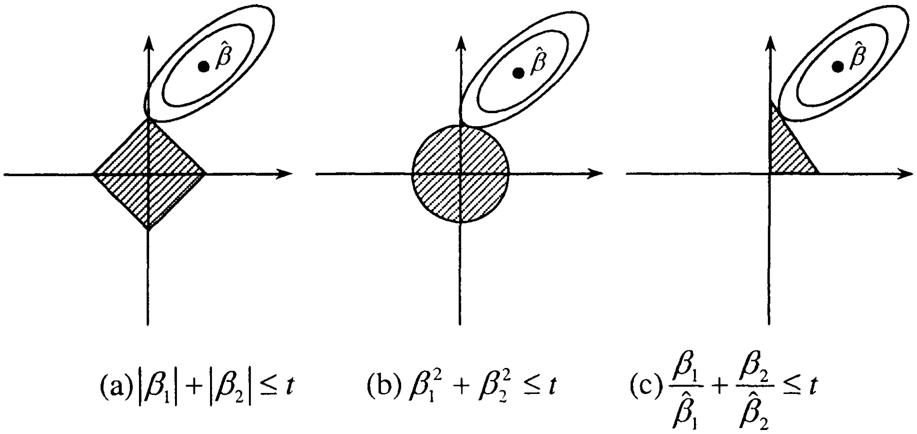

$\bullet\hat{\beta}$
$\bullet\hat{\beta}$
$\bullet\hat{\beta}$
(a) $|\beta_1| + |\beta_2| \leq t$
(b) $\beta_1^2 + \beta_2^2 \leq t$
(c) $\frac{\beta_1}{\hat{\beta}_1} + \frac{\beta_2}{\hat{\beta}_2} \leq t$

Figure 4.1 Graphical representations of three different shrinkage methods

In fact, the model $\sum_{i=1}^{p} m_i \hat{\beta}_i X_i$ with constraint $\sum_{i=1}^{p} m_i \leq t$ is equivalent to the model $\sum_{i=1}^{p} \beta_i X_i$ with constraint $\sum_{i=1}^{p} \frac{\beta_i}{\hat{\beta}_{OLSi}} \leq t$. To illustrate graphically the relationship as well as difference between these three different constraints, let's consider a very case with p=2. The Figure 4.1 shows the parameter optimization pictures of these three methods, where the hatched regions stand for constraint region.

In the case of non-negative garotte, the sign of each $\hat{\beta}_i$ is retained and therefore β is restricted in the quadrant where $\hat{\beta}$ is. Furthermore, each β_i is scaled by a factor of the reciprocal of the corresponding $\hat{\beta}_i$.

Subject to these different constraints, minimizing equation (4.10) results in different regression coefficients, such as

$$\beta_i^{LASSO} = sign(\hat{\beta}_i)(\left|\hat{\beta}_i\right| - \lambda)^+, \tag{4.11}$$

$$\beta_i^{RR} = \frac{1}{1+\lambda}\hat{\beta}_i \tag{4.12}$$

and

$$\beta_i^{NNG} = \left(1 - \frac{\lambda^2}{\hat{\beta}_i^2}\right)^+ \hat{\beta}_i \tag{4.13}$$

for LASSO, standard Ridge Regression and non-negative garotte, respectively [cf. Tibshirani, 1996]. In the above equations (4.11-4.13), λ is determined by t and the superscript + indicates the positive part of the expression, i.e. $f^+ := max(f, 0)$. Note that the above results are derived only for our orthonormal design case.

Consequently, we obtain corresponding membership grades as

$$m_i^{LASSO} = \left(1 - \frac{\lambda}{\left|\hat{\beta}_i\right|}\right)^+, \tag{4.14}$$

$$m_i^{RR} = \frac{1}{1+\lambda} \tag{4.15}$$

and

$$m_i^{NNG} = \left(1 - \frac{\lambda^2}{\hat{\beta}_i^2}\right)^+ \tag{4.16}$$

by LASSO, ridge regression and non-negative garotte, respectively. For each case, we have $\lambda > 0$ and therefore it is easy to check that all of these solutions satisfy $0 \le m_i \le 1$ and $\sum_{i=1}^{p} m_i \le t < p$. However, in above solutions all m_i's are controlled by a single parameter, namely λ, which does not meet our requirement each regressor can determine its own membership grade independently. Therefore, we would like to extend the above solutions to generalized and flexible ones by replacing λ with λ_i, i=1,...,p that is,

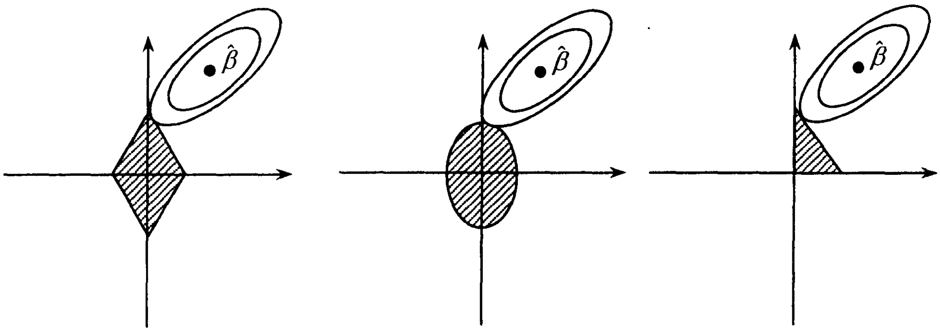
$$m_i^{LASSO} = \left(1 - \frac{\lambda_i}{\left|\hat{\beta}_i\right|}\right)^+ , \tag{4.17}$$

$$m_i^{RR} = \frac{1}{1+\lambda_i} \tag{4.18}$$

and

$$m_i^{NNG} = \left(1 - \frac{\lambda_i^{\,2}}{\hat{\beta}_i^2}\right)^+ , \tag{4.19}$$

where λ_i can be tuned independently.



(a) $\lambda_1|\beta_1| + \lambda_2|\beta_2| \le t$ (b) $\lambda_1\beta_1^2 + \lambda_2\beta_2^2 \le t$ (c) $\frac{\lambda_1\beta_1}{\hat{\beta}_1} + \frac{\lambda_2\beta_2}{\hat{\beta}_2} \le t$

Figure 4.2 Graphical representations of different generalized shrinkage methods

In order to gain some insight into this modification, let's take p=2 as an example to compare a generalized case to its corresponding standard case. The graphical representations of these generalizations are shown in Figure 4.2. Note that in Figure 4.2, square and circle are stretched to be diamond and ellipse, respectively.

Meanwhile, if we further restrict that λ_i be less than $|\hat{\beta}_i|$, which is true most time according to our simulation and other authors [Copas, 1983], we can rewrite the above solutions as

$$m_i^{LASSO} = \left(1 - \frac{\lambda_i}{|\hat{\beta}_i|}\right)^+ = 1 - \rho_i\,, \tag{4.20}$$

$$m_i^{RR} = \frac{1}{1+\lambda_i} = 1 - \frac{\lambda_i}{1+\lambda_i} = 1 - \rho_i \tag{4.21}$$

and

$$m_i^{G} = \left(1 - \frac{\lambda_i^2}{\hat{\beta}_i^2}\right)^+ = 1 - \rho_i \ . \tag{4.22}$$

With this transformation, now it is clear that in a generalized case all of the above methods will converge to the same solution, that is, the optimization procedure will reach the same value of ρ_i. It is easy to see that in all cases $0 \le \rho_i \le 1$ and thus $0 \le m_i \le 1$, therefore all these methods will lead to the same result as our fuzzy model.

Furthermore, note that membership grades obtained from LASSO and non-negative garotte depend on $\hat{\beta}_i$, which in turn depends on the given data set, while the membership grades estimated by generalized ridge regression does not. In addition, although we assume that X_i's are orthonormal, in practice the sample variances are usually not exactly zero, in which case the calculation of LASSO and non-negative garotte turns out to be more complicated, while generalized ridge regression is still able to deal with it easily. In view of these facts, in this study we would apply the generalized ridge regression [Tikhonov, 1963], or local ridge regression as another name [Orr, 1995].

By now, we have discussed how to estimate both membership grade and coefficients of regressors. The practical calculation of solutions is actually a global parameter optimization problem with linear inequality constraints, and we will further discuss it later on.

In our fuzzy variable selection scheme, a slight variation in data set may lead to the change of membership grade of some regressors, but not dramatically from 1 to 0 or vice versa as in crisp model selection methods. Thus, it can be expected such fuzzy variable selection method will give us much more stable results.

Except for what we mentioned about the relationship between crisp variable subset selection methods and fuzzy variable selection scheme, there is actually a closer, and sometimes more useful, connection. This connection can be realized by taking advantage of the concept of α-cut. As we know, give any number $\alpha \in [0,1]$ the α-cut of a fuzzy set U is a crisp set, which is defined as

$$^{\alpha}U = \{X \mid \mu_U(X) \geq \alpha\}. \tag{4.23}$$

Suppose we have already estimated the membership grades $m=(m_1,\ldots,m_p)$, and if we set an appropriate small cut level α such as 0.1 , we obtain a crisp set of predictor variable, which is consistent with a result produced directly by a crisp model selection method. However, the great advantage of this indirect method lies in that it eliminates the need of comparing all possible linear regression models, and therefore significantly save computation cost especially when p is very large.

Furthermore, if the true model can be expressed by some γ, the optimized membership function m will converge to γ.

4.3 Membership grade and effective model dimensionality

In the case of classical model selection, the model dimensionality is defined as the number of free parameters, that is,

$$D = \sum_{i=1}^{p} \gamma_i \,. \tag{4.24}$$

Likewise, we can define the effective model dimensionality in our fuzzy case as

$$D_{eff} = \sum_{i=1}^{p} m_i \,. \tag{4.25}$$

As discussed in section 4.2, these membership grades can be estimated by adding some form of constraint. The local ridge regression [Tikhonov, 1963], which is different from the standard ridge regression in that it applies different individual shrinkage parameter to each regressor, minimizes

$$\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 \text{, subject to } \sum_{j=1}^{p} \omega_j \beta_j^2 \le t, \tag{4.26}$$

which is equivalent to add a Lagrangian penalty term to the sum of squared error as $\lambda \sum_{j=1}^{p} \omega_j \beta_j^2$ with λ depends on t [see Gill, 1986].

Now to solve penalized least squares problem, that is, to estimate β_j is to minimize penalized sum of squared error

$$\begin{aligned} PSSE &= \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} \omega_j \beta_j^2 \\ &= \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \sum_{j=1}^{p} \lambda_j \beta_j^2 \end{aligned} \quad . \tag{4.27}$$

Like OLS, by setting the first derivative of PSSE with respect to each β_j, finally we can obtain

$$\beta = (x^T x + \Lambda)^{-1} x^T y = (x^T x + \Lambda)^{-1} x^T x \hat{\beta}_{oLS} = M\hat{\beta}_{oLS}, \tag{4.28}$$

where Λ is a diagonal matrix, i.e. Λ=diag(λ_1,..., λ_p) and $M = (x^T x + \Lambda)^{-1} x^T x$ Thus, in an orthonormal design case we have

$$M = diag(m_1, m_2, ..., m_p), \text{where } m_i = \frac{1}{1+\lambda_i} \ . \tag{4.29}$$

Then, according to our definition of the effective model dimensionality, we obtain

$$D_{eff} = \sum_{i=1}^{p} m_i = \sum_{i=1}^{p} \frac{1}{1+\lambda_i}, \tag{4.30}$$

which is equal to $trace((x^T x + \Lambda)^{-1} x^T x)$. Therefore, in a general case, without the orthonormal assumption, the effective model dimensionality can be defined as

$$D_{eff} = trace((x^T x + \Lambda)^{-1} x^T x), \tag{4.31}$$

which is in agreement with definitions of the effective dimensions of linear smoothers in [Hastie and Tiibshirani, 1990], the effective number of parameters in nonlinear learning systems in [Moody, 1992], and the number of good parameter measurements in Bayesian context as in [MacKay, 1992]. In the case of OLS, Λ=0 and therefore

$$D_{eff} = trace((x^T x)^{-1} x^T x) = p = D \qquad (4.32)$$

As we know, in some sense the task of model selection is to estimate the true dimensionality based upon data [Schwarz, 1978], and the most classical model selection methods heavily depend on the definition of model dimensionality. Since we argued that the classical model selection is just a special case of the fuzzy model selection scheme, it is quite important to define appropriate model dimensions to keep the fuzzy model selection strategy consistent with classical ones. In terms of our definition in equation (4.30) or (4.31), in any crisp model selection scheme, where m_i is either 0 or 1, or equivalently λ_i is either very large or 0, the effective model dimensionality D_{eff} amounts to the number of free parameters, i.e. the model dimension in a classical variable selection method. Therefore, our definition of the effective model dimensionality is consistent.

4.4 Model evaluation

Any model selection method is based upon its own model performance evaluation criterion, and so is our fuzzy model selection strategy. Only if we can evaluate a fuzzy model, we will be able to estimate an optimal membership function such that maximizes a model's performance.

As we mentioned in section 4.2, a model should be evaluated in terms of its future performance, instead of goodness of fitting. However, except for a limited number of observations we have no idea what the future data would be, and thus a model's future performance certainly cannot be known exactly. In practice, a model's performance can be estimated based on available information by applying some principles, for example the principle of parsimony or Occam's razaor [Madign and Raftery, 1994] and preference of smoothness [O'Sullivan, 1986 and Eilers, 1991], which turns out to be effective.

Under the guide of these principles, a variety of classical model selection methods have been developed. Besides the quadratic loss function as in section 4.2, many other metrics were also applied to measure a model's performance, which include absolute prediction error, expected likelihood, information deviance, for instance, Kullback-Leibler distance[Kullback, 1959].

In statistics, R^2 is often used to measure the proportion of variance of a given data set explained by a set of regressors. For the purpose of model selection, R^2 is adjusted by incorporating a penalty for additional predictions, attempting to adjust R^2 for capitalization on chance in a sample data and give an estimate of R^2 in the population. In mathematics, it is written as

$$AdjR^2 = 1 - \frac{\hat{\sigma}^2}{S_y^2} = 1 - \frac{SSE/(n-D)}{S_y^2} \tag{4.33}$$

where SSE is the sum of squared error as in equation (4.3), n is the sample size, D is the model dimension, S_y^2 is the sample variance of the response variable, and $\hat{\sigma}^2$ is called unbiased estimation of variance

$$\hat{\sigma}^2 = \frac{SSE}{n-D}. \tag{4.34}$$

Mallows' C_p is concerned with total mean squared error of fitted values, which is also closely related to adjusted R^2 and Akaike's AIC [Mallows, 1973]. Mallows' C_p criterion is to minimize

$$C_p = \frac{SSE}{\hat{\sigma}_{full}^2} - [n - 2D] \tag{4.35}$$

where $\hat{\sigma}_{full}^2$ is estimated from the model with all the predictor variables and used to estimate the true variance. If a model is good, $C_p \approx D$, while a model with bad fit will have C_p much bigger than D. In general, C_p is a good indicator for determining when a model is underfitted

Another class of model selection criteria is information criteria, which are so-called due to their close connection to information theories. Most of them can be expressed in a uniform form as

$$IC = -2\log(Maximum\ likelihood) + penalty(D,n)\ , \tag{4.36}$$

where the first term is from a Maximum Likelihood Estimate (MLE) and the second term is a function of model dimension and sample size. In some sense, it can be considered to be an extension of the general maximum likelihood principle. Among them, Akaike's AIC and Schwarz's BIC are two widely applied criteria, for which the penalty term is equal to $2 \cdot D$ and $D \cdot \log(n)$, respectively.

Cross-Validation is a natural way to estimate a model's future performance. Its basic idea is quite straightforward, that is, to evaluate a model using a test data set other than the training set used in model calibration based upon the assumption that the test data can respect future data. A widely used version is K-fold cross-validation, in which the entire data set is broken into K partitions. In each time, one partition is left out as a test set and others are used to calibrate a model, and then the resultant model is tested against the test set. This procedure is repeated K times for each partition and finally the cross-validation generalization error is obtained as the average test error. Leave-One-Out (LOO) cross-validation is a special case of general K-fold cross-validation when K is equal to the sample size.

In the case of multiple linear regression, there is a simple expression of LOO, that is, the PRESS (prediction sum of squares) proposed by Allen [1974], which is defined as

$$PRESS = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{-i})^2 , \tag{4.37}$$

where y_i is the i-th data point and $\hat{y}_i^{-i}$ is the prediction corresponding to y_i by the fitted model with the i-th pair (x_i, y_i) left out. This is actually the sample mean prediction error for 1-fold cross-validation. Furthermore, one can show that

$$y_i - \hat{y}_i^{-i} = \frac{y_i - \hat{y}_i}{1 - h_{ii}} \tag{4.38}$$

where h_{ii} denotes the ii-th element of the "hat" matrix $H=x(x^Tx)^{-1}x^T$ [Cook and Weisberg, 1982]. Then

$$PRESS = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_{ii}}\right)^2 \tag{4.39}$$

With the above formula, PRESS can be calculated without fitting the model n times, each time deleting one of the n cases. If we replace h_{ii} by the average of H's diagonal entries $\bar{h} = \frac{1}{n}\sum_{i=1}^{n} h_{ii}$, we obtain the error prediction as

$$GCV = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1-\bar{h}}\right)^2 = \frac{SSE}{n(1-\bar{h})^2} = \frac{n \cdot SSE}{(n - trace(H))^2}, \tag{4.40}$$

which is termed Generalized Cross-Validation (GCV) [see Golub,Heath, and Wahba, 1979]. Since the trace of the product of a sequence of matrixes is unaffected by their orders, for example, $trace(A_{n\times m} \cdot B_{m\times n}) = trace(B_{m\times n} \cdot A_{n\times m})$, therefore

$$trace(H) = trace((x^T x)^{-1} x^T x) = D. \tag{4.41}$$

As we notice, most of model selection criteria are functions of the model dimension. With the aid of the definition of the effective model dimensionality given in the previous section, it is easy to extend all these model evaluation criteria to the fuzzy model selection, which is realized by simply replacing the model dimension by the effective model dimension, D_{eff}. For example, GCV becomes

$$GCV_{Fuzzy} = \frac{n \cdot SSE}{(n - D_{eff})^2} \tag{4.42}$$

and by some manipulation we obtain an equivalent BIC in multiple linear regression case as

$$BIC_{Fuzzy} = -n \log SSE + D_{eff} \log n. \tag{4.43}$$

We can see that maximize the adj-R^2 in (4.33) is equivalent to minimizing $\frac{SSE}{n - D_{eff}}$, because the sample variance of the response variable $S_y{}^2$ is fixed given a data set. Thus, in a fuzzy case, GCV is equivalent to adj-R^2 method.

Meanwhile, if we assume the noise variance $\sigma^2$is already know, not necessary to be estimated by $\hat{\sigma}^2_{full}$ as in (4.35), then it is easy to derive that AIC can be expressed as

$$AIC_{Fuzzy} = \frac{SSE}{\sigma^2} + 2D_{eff}, \tag{4.44}$$

which is actually, as we pointed out a little earlier, the same as Mallow's C_p criterion. Since the use of an estimate of σ^2 in place of its true value will not lead to significant difference, we will consider these two criteria to be the same in the current study.

In view of the above equivalences in our linear case with Gaussian noises, in this chapter we will exclude adj-R^2 and Mallow's C_p in the empirical study.

Finally, each model evaluation method can be expressed in a unified form

$$MEV = f(SSE, D_{eff}, n), \tag{4.45}$$

where MEV stands for model evaluation value, SSE is the sum of squared error, D_{eff} is the effective model dimension, n is the sample size and $f(\cdot)$ is a function. The MEV, which is depending on the choice of regularization parameters through SSE and D_{eff}, serves as our objective function, i.e. the smaller it is, the better a fuzzy model.

It is worth mentioning that like any crisp model selection method, its performance significantly depends on the choice of model evaluation method.

Although different objective functions in the current case produce similar results, they do have varied properties in terms of rate of convergence of optimization algorithms, stability and probability of outperforming corresponding classical methods, as we will see later on. It is possible to employ hybrid objective functions to combine advantages of individual objective functions, for example, employ two objective functions in a certain order or alternatively during GA simulation or utilize one to search and another one to pick the best, which is somewhat like the cross-validation method in data manipulation. In our simulations, for the purpose of illustration BIC is applied to global search and 10-CV is subsequently utilized for local searching in view of their different individual advantages.

4.5 Individual shrinkage parameter estimation

By now, we have already developed a fuzzy regression model as

$$Y = \sum_{i=1}^{p} m_i \hat{\beta}_{OLSi} X_i =, \sum_{i=1}^{p} \beta_i X_i \text{ with } m_i \in [0,1], \tag{4.46}$$

where coefficients β_{OLSi}'s can be estimated given membership grades m_i's or equivalently individual shrinkage parameters λ_i's by local ridge regression method, i.e.

$$\hat{\beta} = \arg\min_{\beta}\left\{\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \sum_{j=1}^{p}\lambda_j\beta_j^2\right\} = (x^T x + \Lambda)xy\,, \tag{4.47}$$

where Λ=diag(λ_1,...,λ_p) and λ_i's are tunable. To build an optimal fuzzy regression model, its regularization parameters, namely m_i's or equivalently λ_i's, should be optimized so as to maximize its future model performance measured by model evaluation methods discussed in the last section. That is,

$$\hat{\lambda} = \arg\min_{\lambda} MEV(\lambda)\text{, with } \lambda_i \geq 0,\ i=1,\ldots p, \tag{4.48}$$

where the model evaluation method can be any one described earlier.

Therefore, our variable selection problem now turns to be a multidimensional global optimization problem. Although it is also possible to optimize m_i directly, using λ_i imparts some advantages. First, it has fewer constraints and thus simplifies the optimization problem. Another advantage of using λ_i is that it can help handle situations where predictor variables are not orthonormal.

For this nonlinear optimization, the classical gradient-based methods fail to apply, as it is hard to solve the nonlinear multivariate functions of λ_i's, especially when K-fold cross-validation is utilized. To attack this difficulty, Orr [1995] proposed to optimize each parameter by itself one by one with others fixed and repeat doing this until they converge. Nevertheless, as pointed out by the author [Orr, 1996], that algorithm tends to get stuck in local minima, which depend on the initial guess. To see this characteristic of multiple local minima, let's take the BIC as the model evaluation method and analyze the objective function. In this case, the optimization problem becomes

$$\begin{aligned}\hat{\lambda} &= \arg\min_{\lambda} MEV(\lambda) = \arg\min_{\lambda} -n\log SSE + D_{eff}\log n \\ &= \arg\min_{\lambda} -n\log\sum_{i=1}^{n}(y_i - (x^T x + \Lambda)x^T y x_i)^2 + trace((x^T x + \Lambda)^{-1}x^T x)\log n\end{aligned}\ . \tag{4.49}$$

To make explicit the dependence of the objective function on $\{\lambda_j\}_{j=1}^{p}$, let

$$\Lambda = \Lambda^{-j} + \Lambda^{j}, \tag{4.50}$$

where $\Lambda^{-j} = diag(\lambda_1,\ldots,\lambda_{j-1},0,\lambda_{j+1},\ldots,\lambda_p)$ and $\Lambda^{j} = diag(0,\ldots 0,\lambda_j,0,\ldots,0)$. Therefore, it gives

$$\hat{\beta} = x^T x x^T y + \Lambda^{-j} x^T y + \Lambda^{j} x^T y \qquad (4.51)$$

Substituting equation (4.51) into (4.49) and taking into account the fact that $x^T x = I$, we obtain the objective function

$$-n \log \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \hat{\beta}_j x_{ij})^2 + \log n \sum_{i=1}^{p} \frac{1}{1+\lambda_i}, \qquad (4.52)$$

where among all elements in $\hat{\beta}$ only $\hat{\beta}_k = \left[x^T x x^T y + \Lambda^{-k} x^T y\right]_k + \lambda_k \left[x^T y\right]_k$ depends on λ_k.

Note that $[a]_k$ refers to the element with index k in the vector a.Equating as zero the first derivative of equation (4.52) with regard to λ_k gives

$$\frac{\left[x^T y\right]_k \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \hat{\beta}_j x_{ij}) \cdot x_{ik}}{\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \hat{\beta}_j x_{ij})^2} - \frac{\log n}{2n} \frac{1}{(1+\lambda_k)^2} = 0 \qquad (4.53)$$

With fixed λ_i, $i \neq k$, equation (4.53) turns out to be a cubic equation of λ_k, which in general has three roots. However, according the relationships between roots and coefficients, equation (4.53) can have at most two positive solutions. If we substitute m_i into above equation, this becomes clearer. But at the same time, we have to be careful that the boundaries 0 or $+\infty$ can be other local minima if the derivatives at them are positive or negative, respectively. Because one can derive an equation for k from 1 through p, then number of local minima will almost surely be greater than 1.

In light of the existence of multiple local minima, in this chapter we will propose a hybrid optimization algorithm, which combines a global as well as a local searching algorithm. The basic idea is that the outputs of the global optimization algorithm sever as the inputs or initial settings of the local optimization method. In so doing, we will gain both global and local optimization capability and save computation cost at the same time.

It is well known that the Genetic Algorithm (GA) has great global searching capability, and it does not require the objective function to be differentiable [Goldberg, 1989]. However, its local optimization is relatively poor and time-consuming. To overcome its drawbacks, we will design a simple adaptive derivative-free local searching algorithm to combine with GA.

4.5.1 Global optimization

Basically, GA is a numerical version of natural selection, which is believed to be the ultimate optimizer based upon the idea of Darwin's revolutionary writing *Origin of Species.* According to Darwin, the power of evolution lies in the continuing struggle for survival and some "variation", such as mutation of genes, of an organism increases its chances for survival. Therefore, key to such evolution is the concept of larger numbers, i.e. large population and many generations, and randomness, such as the probabilistic selection, mixing, and mutation. Likewise, GA first generates a large set of random parameters as a generation, and then randomly select parameters in terms of their fitness to reproduce a new generation of parameters by means of random crossover and mutation. In GA, the fitness-based selection will ensure the consistent direction of evolution, or guarantee the increase of the average fitness of a generation. Meanwhile, crossover and especially mutation enables the GA to avoid being stuck at a local minimum and search for the global optimum.

Usually, a chromosome in GA is represented by a binary string consisting of 0s and 1s, but in the current case each parameter to be optimized is a positive floating point number, and therefore floating-point coding or double-precision representation will be used rather than binary coding. This is because as pointed out by some research, binary coding is less suited for numerical optimization problems [Garcia, 1999], although a floating-point number can also be expressed in a binary form somehow. Therefore, each chromosome is a vector of floating-point parameters.

The first crucial issue of GA is the fitness function, which tells how good or bad a candidate solution is. It is this fitness function that determines the goal of optimization. As we discussed, the goal of model selection is to minimize the objective function, i.e. prediction error, but GA works by maximizing the fitness. Such conflict can be resolved by a simple transformation

$$fitness_i = Max(MEV) - MEV_i \ , \tag{4.54}$$

where *Max(MEV)* stands for the maximum *MEV* in a population and MEV_i refers to the MEV of the *i*-th individual.

The definition of fitness significantly influences the behavior of convergence. For example, in the early stage few "super individuals" tend to dominate the selection process leading to premature, whereas later when the population is less diverse, the simulation tends to lose focus [Goldberg, 1989]. Therefore, in practice we would like to apply a more general and flexible fitness function by scaling and shifting, i.e.

$$F_i = b + a \cdot (MEV_{\max} - MEV_i), \tag{4.55}$$

where the scaling factors *a* and shifting factor *b* are so adjusted adaptively during simulation as to avoid premature convergence early on and encourage convergence in later stages.

As for selection, we utilize the fitness-weighted roulette wheel method, which is conceptually equivalent to giving each individual a slice of a roulette wheel equal in area to the individual's fitness. The wheel is spun and the ball comes to rest on the wedge shaped slice, and the corresponding individual is selected. Therefore, the probability for a chromosome to be chosen is proportional to its fitness. A pair of "parents" is selected by spinning the wheel two times to reproduce a pair of "children" by recombination and mutation.

As we know, the GA success is also sensitive to the two operators, namely, recombination operator and mutation operator. For example, it is found that the general, fixed, problem-independent recombination operators often break partial solutions and slow down convergence. Thus, in view of the fact that each parameter falls in the interval $[0, \infty)$, we will design a recombination and mutation operator suited to our case.

The recombination strategy we applied is the one-point arithmetic crossover. Let the parents be P_1=[P_{11},...,P_{1L}] and P_2=[P_{21},...,P_{2L}], respectively. Then, the two children are

$$C_{1i} = \begin{cases} P_{1i} & i \le t \\ a \cdot P_{1i} + (1-a) \cdot P_{2i} & i > t \end{cases}, \tag{4.56}$$

and

$$C_{2i} = \begin{cases} P_{2i} & i \le t \\ b \cdot P_{1i} + (1-b) \cdot P_{2i} & i > t \end{cases}, \tag{4.57}$$

where t is a random integer number among $1,2,\ldots,L$, and a and b are two random floating-point numbers between [0,1].

The crossover rate, i.e. the probability that crossover happens, is generally around 0.5, and in this chapter we set it as 0.6.

Mutation operator is defined as multiplication of a log-normal distributed factor with median value 1, i.e. $D_i' = D_i \cdot \exp(\varepsilon)$, where D_i is an original parameter and D_i' is the mutated one, ε is a Gaussian random number, i.e. $N(0,\sigma^2)$, where σ^2 is tunable.

The mutation operator is so defined that all the resultant parameter stays in the interval $[0,\infty)$, and can be larger or smaller than the original one. For practical purpose, we set the range of parameters as $[l,u]$, where l is very small, say 10^{-6}, while u is very large such as 10^7. In fact, searching for parameters out of that range is meaningless and impractical. In practice, if some λ is greater than u, the corresponding predictor variable will be removed from the regression model.

Hessner and Manner [1991] suggested that the optimal mutation rate, i.e. the probability that mutation occurs for a single gene in a chromosome, is approximately $(M \cdot L^{1/2})^{-1}$, where M is the population size and L is the length of the chromosome. In this chapter, we will follow this "rule of thumb".

If there is no crossover and mutation, a chromosome is simply copied to the next generation. For the above operators, we see that regularization parameters evolve almost independently except for selection operator.

The stopping rule for the current case is relatively simple, as our purpose is to search for promising initial inputs for a local optimization algorithm. Thus, when we observe that the convergence of GA becomes very slow, it will be the time to stop it. However, it is should be met that the *MEV* of the best solution be at least lower than that of the corresponding crisp one.

Finally, the main steps of GA are follows:

(1) Build an initial population of *M* chromosomes randomly;

(2) Calculate the fitness of each chromosome;

(3) Select chromosomes from the parent generation to reproduce a child generation:

(i) Select two parent chromosomes,

(ii) Generate a random number between [0,1]. If it is smaller than the crossover rate, recombine them by one-point arithmetic crossover; otherwise, enter the next step;

(iii) Generate a random number between [0,1]. If it is smaller than the mutation rate, perform mutation on a gene in a chromosome. Repeat this for each gene in both chromosomes.

(iv) Add the two resulting chromosome to the next generation.

Repeat the above (*i*) through (*iv*) steps until *M* new chromosomes are reproduced.

(4) If the stopping criterion is met, then exit; otherwise, return to step (2).

4.5.2 Local optimum search

By GA optimization, we obtain a set of global good initial guesses of the best vector of parameters, namely the last generation out of GA. In practice, it is also useful to keep tract of the "best" chromosome throughout the whole GA simulation history. The next task is to search for the optima around these good initial guesses.

In this stage, for some model evaluation methods classical gradient-based methods can be applied directly. For example, the equation of local minima has been derived for BIC in equation (4.43). Likewise, we can derive it for AIC and GCV, respectively, as follows:

$$\frac{[x^T y]_k \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \hat{\beta}_j x_{ij}) \cdot x_{ik}}{\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \hat{\beta}_j x_{ij})^2} = \frac{1}{n} \frac{1}{(1+\lambda_k)^2} \qquad (4.58)$$

and

$$\left(n-\sum_{i=1}^{p}\frac{1}{1+\lambda_i}\right)\frac{\left[x^T y\right]_k \sum_{i=1}^{n}(y_i-\sum_{j=1}^{p}\hat{\beta}_j x_{ij})\cdot x_{ik}}{\sum_{i=1}^{n}(y_i-\sum_{j=1}^{p}\hat{\beta}_j x_{ij})^2}=\frac{2}{(1+\lambda_k)^2} \tag{4.59}$$

However, it is still not easy to analytically solve the above equations for λ_k with others fixed by expressing the coefficients explicitly, but this difficulty can be overcome by an ordinary iterative method for equation like $x=f(x)$. To this end, the above equations should be rewritten as

$$\lambda_k=\sqrt{\frac{\log n\sum_{i=1}^{n}(y_i-\sum_{j=1}^{p}\hat{\beta}_j x_{ij})^2}{2n\left[x^T y\right]_k \sum_{i=1}^{n}(y_i-\sum_{j=1}^{p}\hat{\beta}_j x_{ij})\cdot x_{ik}}}-1, \tag{4.60}$$

$$\lambda_k=\sqrt{\frac{\sum_{i=1}^{n}(y_i-\sum_{j=1}^{p}\hat{\beta}_j x_{ij})^2}{n\left[x^T y\right]_k \sum_{i=1}^{n}(y_i-\sum_{j=1}^{p}\hat{\beta}_j x_{ij})\cdot x_{ik}}}-1 \tag{4.61}$$

and

$$\lambda_k=\sqrt{\left(n-\sum_{i=1}^{p}\frac{1}{1+\lambda_i}\right)^{-1}\frac{2\sum_{i=1}^{n}(y_i-\sum_{j-1}^{p}\hat{\beta}_j x_{ij})^2}{\left[x^T y\right]_k \sum_{i=1}^{n}(y_i-\sum_{j=1}^{p}\hat{\beta}_j x_{ij})\cdot x_{ik}}}-1. \tag{4.62}$$

In order to optimize the λ_i's jointly, a similar iterative scheme can be applied for solving equations of multivariate, that is, letting $\lambda^{(k+1)}=g(\lambda^{(k)})$ where λ is a vector though.

Nonetheless, this approach cannot always work. For example, if k-fold cross-validation is used to assess a model, the objective function will be quite complicated and thus it is hard to apply classical gradient-based methods. However, it can be solved numerically by some derivative-free approach, for example, hill climbing. What we will propose here is similar to that in [Orr, 1995] in that both of us try to optimize parameters individually one by one with others fixed. However, since each parameter is not independent of each other, the overall optimization has to be done in an iterative way. Furthermore, the step size is also tuned adaptively. Our algorithm described below is simple, self-adaptive and fast.

The whole process consists of multiple cycles, in each of which the individual parameters are optimized one at a time. Suppose we are optimizing λ_i and let its current value as $\lambda_i^{(0)}$ and the current model evaluation value as $MEV^{(0)}$. Let k=1 and $\lambda_i^{(k)} = \lambda_i^{(k-1)} \cdot q^k$, where q is slightly bigger than 1, and keep the other p-1 shrinkage parameters unchanged, and then recalculate the model evaluation value as $MEV^{(k)}$. If $MEV^{(1)} > MEV^{(0)}$, that is, the fuzzy model gets worse, then return to $\lambda_i^{(0)}$ and k=1and replace q by q^{-1}; otherwise, continue to search in the same direction within the interval $[l,u]$ until $MEV^{(k+1)} > MEV^{(k)}$. The final $\lambda_i^{(k)}$ is taken as the optimal value in the current loop. After λ_i is optimized, we turn to the next parameter λ_{i+1}. Each loop starts with λ_0 and ends up with λ_p. Once a loop is done, another one will be started depending on the stopping criterion.

At the beginning of each loop, we calculate the resultant model's MEV, and the same for the end of each loop. If the difference between these two values is small enough, for example,

$$\frac{\left|MEV^{(j+1)} - MEV^{(j)}\right|}{MEV^{(j+1)}} < \delta, \tag{4.63}$$

where δ is very small, say 10^{-5}, we would say that the minimum has been reached and therefore stop the local searching process.

In view of the facts that (i) There exists a lower bound for $MEV^{(k)}$, although unknown, and (ii) the sequence of $MEV^{(k)}$ is non-increasing, the convergence is guaranteed according to the Cauchy convergence criterion.

After accomplishing both global and local optimization procedure, a group of good candidate solutions are obtained, from which the solution with the smallest MEV can be easily chosen as the optimal one. Finally, an optimal fuzzy regression model can be created.

4.6 Numerical simulation study

In the above sections, we have already completely developed a fuzzy model selection method, and now we will assess its performance by means of numerical simulation. For the purpose of demonstration, we apply cosine series to fit some data set generated by an even function plus Gaussian noise, and we also test the fuzzy model selection performance by varying noise variances as well as sample size.

Fourier series, made up of both sine and cosine components, is widely applied to approximate periodic functions. Suppose $f(t)$ is a periodic function of t with period τ, and then we have

$$f(t)=\sum_{i=0}^{\infty}a_i\cos(2\pi\cdot it/\tau)+\sum_{i=1}^{\infty}b_i\sin(2\pi\cdot it/\tau). \qquad (4.64)$$

Furthermore, if $f(t)$ is an even function in $[-L, L]$, then the sine terms will vanish and we obtain

$$f(t)=\sum_{i=0}^{\infty}a_i\sqrt{2}\cos(\pi\cdot it/L). \qquad (4.65)$$

In the above equation (4.65), cosine terms can be viewed as predictor variables. It is easy to check that $\sqrt{2}\cos(\pi\cdot it/L)$ are orthonormal.

After gathering a set of data generated by $f(t)+\varepsilon$, where $\varepsilon\sim N(0,\sigma^2)$, the coefficients a_i's in equation (4.64) can be estimated using some approach, for example, OLS, ridge regression, and the fuzzy model selection.

Because of the symmetry of cosine functions, in sampling data, special attention should be made to ensure that the rank of the design matrix x is not smaller than the number of predictor variables p, and otherwise $x'x$ will be singular. Therefore, in this study we simply restrict the total number of regressors p to be not greater than $n/2$-1, where n is the number of data points, that is,

$$f(t)=\sum_{i=0}^{p}a_i\sqrt{2}\cos(\pi\cdot it/L),\text{ with p}\leq n/2\text{-1}. \qquad (4.66)$$

Since in the current simulation study we assume the true model is already known, therefore we can define a global mean prediction error to measure the performance of a model selection method. Conceptually, the global mean prediction error is defined as

$$GMPE=\frac{1}{2L}\int_{-L}^{L}(f(t)-\hat{y}(t))^2dt, \qquad (4.67)$$

where $\hat{y}(t)$ is the estimated model, but in practice, it can be approximated by

$$GMPE \approx \frac{1}{m}\sum_{i=1}^{m}(f(t_i) - \hat{y}(t_i))^2, \tag{4.68}$$

where t_i is typically evenly distributed in [-L, L] and m is large enough.

Besides evaluating the performance of the fuzzy model selection method using GMPE, we will also compare it to that of a corresponding crisp model selection using the same model evaluation method. To this end, we define a *C/F* ratio of *GMPE*s, i.e.

$$R_{C/F} = \frac{GMPE_{Crisp}}{GMPE_{Fuzzy}}. \tag{4.69}$$

The average *C/F* ratio is defined as its geometric mean

$$\overline{R}_{C/F} = \exp\left(\frac{1}{N}\sum_{i=1}^{N}\log(R_{C/F})_i\right), \tag{4.70}$$

where N is total number of simulations. The average *C/F* ratio is so defined that it can reduce the influence of some unusual cases.

Table 4.1 Comparison results for a smooth function with sample size 20

Model Evaluation Method	σ=0.2 and n=20			σ=0.05 and n=20		
	$AGMPE_C$[(1)]	$AGMPE_F$[(2)]	$R_{C/F}$[(3)]	$AGMPE_C$	$AGMPE_F$	$R_{C/F}$
AIC	0.018	0.014	1.34	1.18E-3	8.86E-4	1.39
BIC	0.015	0.011	1.47	1.14E-3	8.51E-4	1.39
10-CV	0.018	0.014	1.31	1.28E-3	9.91E-4	1.28
GCV	0.018	0.015	1.22	1.25E-3	9.88E-4	1.28
Hybrid	0.016	0.011	1.52	1.08E-3	7.38E-4	1.5

(1)*Average GMPE of Crisp Model Selection Methods*

(2)*Average GMPE of Fuzzy Model Selection Methods*

(3) *Average C/F Ratio of Prediction Errors*

The first function we used is $f_1(x)$=sinc(3x)=sin(3x)/(3x), $x \in$ [-1,1]. In our simulation, two different noise variances, i.e. σ=0.2 and 0.05, and two different sample sizes, i.e. n=20 and 50 , are used. For each case, the simulation was repeated for 500 times to obtain an average value. The results are shown in Table 4.1 and Table 4.2.

Also note that in our simulation when a shrinkage parameter was over 10^7, the corresponding regressor was removed from the fuzzy set for the practical purpose.

From the simulation results, it is seen that on average the fuzzy model selection scheme outperforms crisp model selection methods by a factor of around 1.4.

Table 4.2 Comparison results for a smooth function with sample size 50

Model Evaluation Method	σ=0.2 and n=50			σ=0.05 and n=50		
	$AGMPE_C$	$AGMPE_F$	$R_{C/F}$	$AGMPE_C$	$AGMPE_F$	$R_{C/F}$
AIC	1.40E-2	8.56E-3	1.65	1.04E-3	6.77E-4	1.6
BIC	1.52E-2	7.86E-3	1.94	7.41E-4	5.70E-4	1.29
10-CV	1.25E-2	8.83E-3	1.37	8.96E-4	6.17E-4	1.51
GCV	1.54E-2	9.46E-3	1.65	1.13E-3	7.2E-4	1.64
Hybrid	1.03E-2	6.99E-3	1.43	8.03E-4	5.44E-4	1.48

If we define the signal-noise ratio as the ratio of response variable variance to the noise variance, we can see the influence of the signal-noise ratio on the increased performance of the fuzzy model selection compared to the corresponding crisp model selection in some situations.

The second function we used is

$$f_2(x) = \begin{cases} 1, \mathit{if}\ x \in [-0.5, 0.5] \\ -1, \ \mathit{otherwise} \end{cases}, \tag{4.71}$$

which is called potential well function.

Once again, two different noise variances and two different sample sizes are used. After repeating the simulation 500 times for each case, the results were obtained as shown in Table 4.3 and Table 4.4.

Table 4.3 Comparison results for a non-smooth function with sample size 20

Model Evaluation Method	σ=0.2 and n=20			σ=0.05 and n=20		
	$AGMPE_C$	$AGMPE_F$	$R_{C/F}$	$AGMPE_C$	$AGMPE_F$	$R_{C/F}$
AIC	0.116	0.09	1.3	0.104	0.066	1.61
BIC	0.115	0.092	1.23	0.104	0.063	1.66
10-CV	0.114	0.085	1.34	0.105	0.058	1.8
GCV	0.118	0.086	1.39	0.104	0.06	1.74
Hybrid	0.11	0.082	1.32	0.104	0.06	1.73

Table 4.4 Comparison results for a non-smooth function with sample size 50

Model Evaluation Method	σ=0.2 and n=50			σ=0.05 and n=50		
	$AGMPE_C$	$AGMPE_F$	$R_{C/F}$	$AGMPE_C$	$AGMPE_F$	$R_{C/F}$
AIC	3.75E-2	3.35E-2	1.12	1.66E-2	1.64E-2	1.01
BIC	4.26E-2	3.37E-2	1.26	1.74E-2	1.71E-2	1.014
10-CV	3.94E-2	3.71E-2	1.07	1.72E-2	1.66E-2	1.035
GCV	3.61E-2	3.16E-2	1.146	1.65E-2	1.61E-2	1.03
Hybrid	4.12E-2	3.38E-2	1.22	1.72E-2	1.69E-2	1.013

From the above two tables, we can see that for non-smooth functions, the fuzzy variable selection method still significantly outperform corresponding crisp variable selection, especially when the sample size is relatively small.

Furthermore, the performance of the fuzzy method also depends on what kind of model evaluation method it applies. In our first example, BIC works the best among all model evaluation methods, while in the second example cross-validation seems perform a little better than others. Meanwhile, most often the hybrid objective function outperforms that of

BIC method alone.

To further test its performance, we also compared fuzzy variable selection with bagged variable selection [Breiman, 1996b]. In the context of regression models, there are two different ways to generate bootstrap samples. One is to directly bootstrap the data pairs (x_i, y_i) and another one is to bootstrap the residual errors ε_i and finally produce (x_i, $x_i \hat{\beta} + \varepsilon_{bi}$). In light of the nonrobustness of the second bootstrap method, Efron and Gong [1983] seem to prefer the first one. In this study, we will apply the former method. The Monte Carlo simulations are only done for the case with the sample size of 50 and noise level of 0.2, and for both the smooth and non-smooth function. The results are shown in Table 4.5.

Table 4.5 Comparing fuzzy variable selection and bagged variable selection

Model Evaluation Method	Smooth function			Non-smooth function		
	$AGMPE_C$	$AGMPE_B$	$AGMPE_F$	$AGMPE_C$	$AGMPE_B$	$AGMPE_F$
AIC	3.75E-2	4.19E-2	3.35E-2	1.66E-2	1.60E-2	1.64E-2
BIC	4.26E-2	4.73E-2	3.37E-2	1.74E-2	1.77E-2	1.71E-2
GCV	3.61E-2	3.82E-2	3.16E-2	1.65E-2	1.66E-2	1.61E-2

Table 4.5 shows that in our simulations fuzzy variable selection significantly outperforms classical variable selection, while bagged variable selection deteriorates the performance a little.

Comparing the two functions we used in our simulation, it was also found that for the first function, which is smooth, the optimal solution in the last generation always outruns the optimal one in the whole history of optimization, while for the second function, for which usually more predictor variables are selected in the crisp model selection procedures, the optimal solution over whole history of optimization always outruns the optimal one in the last generation. This gives us some insight concerning how to select the best one among those candidate vectors of shrinkage parameters.

In our simulation, we also found a close relation between fuzzy and crisp model selection methods. If we set some α, say α=0.1, then the α-cut of the fuzzy variable set is the same as the variable subset selected by the corresponding crisp model selection method. Particularly in cases of overfitting, the best subset, which can be found by the true model, is nested in the optimal subset, which is determined by a crisp variable selection method, and both can be regarded as α-cut of the fuzzy variable set with some α. This property enables us perform crisp model selection by taking advantage of the fuzzy method, while avoiding combinatorial explosion especially when the full predictor variable set is large.

4.7 Conclusion

In this chapter, we developed a fuzzy variable selection scheme for multiple linear regression models by generalizing classical crisp model selection methods. Besides producing interpretable models, this method also incorporates the favorable stability of shrinkage estimators. With the definition of the effective model dimension, each classical model evaluation criterion can be easily extended to the fuzzy case. Based upon these model evaluation methods, the coefficients in a fuzzy model can be optimized by a hybrid optimization algorithm having both global and local searching capability, which is realized by combining together a global and a local optimization algorithm. The numerical study further shows that this fuzzy model selection strategy significantly improves the accuracy as well as precision of a predictive model.

Chapter 5

Dealing with Non-normality, Outliers and Heteroscedasticity in Partly Linear Regression

5.1 Introduction

Multiple linear regression is widely used for modeling and prediction. Usually, a linear regression model is expressed in the following form

$$y = x\beta + \varepsilon, \tag{5.1}$$

where y is $n\times1$ vector of observations on a dependent response variable, x is a $n\times p$ matrix of observed regressors, β is the $p\times1$ vector of regression parameters to be estimated , and ε is the $n\times1$ noise vector, which is usually assumed to be $\varepsilon \sim N(0,\sigma^2\Sigma)$. Furthermore, without loss of generality it is convenient for us to assume $X=[X_1,\ldots,X_p]$ comprises an orthogonal set of regressors, which is easy to be done by some transformation.

Ordinary Least Squares (OLS) is certainly the most widely applied method for estimating β in multiple linear regression. The regression coefficients are estimated so as to minimize the sum of squared errors (SSE), i.e.

$$\hat{\beta}_{OLS} = \arg\min_{\beta} SSE = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i\beta)^2 . \tag{5.2}$$

It is easy to derive that

$$\hat{\beta}_{OLS} = (x^T x)^{-1} x^T y . \tag{5.3}$$

An estimator $\hat{\beta}$ of β is considered to be the Best Linear Unbiased Estimator (BLUE), if

i) It is a linear function of the random variables, x and y;

ii) $E(\hat{\beta}) = \beta$, that is, it is unbiased; and

iii) $Var(\hat{\beta}) < Var(\hat{\beta}^*)$ for any other $\hat{\beta}^*$ that is linear and unbiased, i.e. $\hat{\beta}$ is the most efficient.

According to the Gauss-Markov theorem, the OLS estimators of β are BLUE if the following Gauss-Markov (GM) assumptions are satisfied:

A1) ε_i is Gaussian with $E(\varepsilon_i \mid x_i) = 0$,

A2) ε_i is independent, i.e. $Cov(\varepsilon_i, \varepsilon_j) = 0 \quad \forall\ i \neq j$,

A3) ε_i is homoscedastic, i.e. $Var(\varepsilon_i \mid x_i) = \sigma^2\ \forall\ i$,

A4) $Cov(\varepsilon_i, x_i) = 0$, and

A5) The regression model is properly specified.

With all the above GM assumptions holding, OLS is regarded as BLUE and naturally any other method cannot outperform OLS in terms of generalization error. However, as matter of fact this cannot always be the case. For example, in the presence of heteroscedasticity, the GM assumption A3 is violated, and therefore the OLS estimators are not BLUE anymore, although they remain unbiased.

In reality both model and data problems may lead to the failure of the Gauss-Markov assumptions. In this chapter we will mainly focus our attention on some of them, including model misspecification, non-normal errors, outliers and heteroscedasticity. To describe these problems, let's first introduce a more general generating model

$$Y(t) = X\beta + g(t) + u \tag{5.4}$$

where Y is a response variable depending on a variate t, X is a row vector of predictor variables, β is the vector of parameters of the regression model, u is an error term, and $g(t)$ denotes the overall effect of predictor variables excluded from X and therefore serves as the bias term. Furthermore, $g(t)$ might be non-linear, and thus such a model is partially linear.

The ideal conditions about both model specification and error distribution may not be fulfilled, and here we will consider these potential problems and their corresponding effects on the modeling one by one.

(1)Non-normal error distributions

Usually, the error u in the generating model in equation (5.4) is assumed to be normally distributed, but in fact there is no reason for us to stick to such an assumption. Therefore, we will extend the normality assumption to any arbitrary symmetric error distribution, for example, some heavy-tailed distribution like the Cauchy distribution.

Without the assumption of normal error holding, the OLS estimate is not efficient any more, but maximum likelihood estimates can be applied instead just as for Generalized Linear Models (GLM) [see Nelder and Wedderburn, 1972]. However, another problem is that the type of error distribution may be unknown.

(2) Outliers

To explicitly distinguish outliers from observations resulting from heavy-tailed error distribution, outliers here specially refer to bad data points due to some unusual error, or something like surprises of stock return in a financial market. In this case, the normal error can be considered to be contaminated by some unknown distribution as follows

$$e = \varepsilon + s , \tag{5.5}$$

where ε denotes the normal error and s refers to the contaminating error generated by some unspecified distribution.

Since outliers have undeserved heavy influence on the estimation of regression coefficients when using ordinary least squares, it is important to detect outliers and assign them appropriately smaller importance, and even rule them out. In reducing or eliminating the influence of outliers, some robust regression methods like M-estimators [Huber, 1964] turn out to be effective. Although we model outliers explicitly differently from heavy-tailed error distributions, in practice heavy-tailed error distributions can also be used to accommodate outliers.

(3) Heteroscedasticity

Heteroscedasticity is another central issue in regression analysis, which has captured lots of attention among statisticians. The presence of inconstant variance of the assumed normal error distribution violates the GM assumption A3.

A usual way to deal with heteroscedasticity is to estimate the covariance matrix of errors as in generalized least squares (GLS) [see Carroll, Wu and Ruppert, 1988], or alternatively model the variance deterministically if one is in possession of some knowledge about the variance or randomly when there is less information. For example, assuming that the σ_i are random, Hooper [1993] showed that empirical Bayes estimators improves the Fuller and Rao weighted least squares estimator (WLSE) [Fuller and Rao, 1978] provided that we can correctly specify the distribution type of the random σ_i. Verbyla [1993] modeled the variance component ε as log-linearly dependent on explanatory variables. In both cases, the overall error distribution is non-normal, although still symmetric.

In reality, we can also find some examples. A Gaussian variable whose variance fluctuates over time in general generates a super-Gaussian distribution and real signals such as oscillations have sub-Gaussian distributions. [see e.g. Parra et al., 2001]. In addition, Beale and Mallows [1959] show that mixture of symmetric distributions, or equivalently varying the variance of a symmetric distribution, always leads to increasing kurtosis, i.e. super-Gaussian.

In the cases where the variance is modeled as a discrete random variable, the overall error can follow an arbitrary distribution according to the theory of mixture of Gaussian [see McLachlan and Peel, 2000].

(4) Incomplete set of predictor variables

Compared to the model in equation (5.4), the model in equation (5.1) is partly linear in that $g(t)$ is actually omitted, thereby representing part of model misspecification. If we assume X is orthogonal to $g(t)$, X can be considered to be uncorrelated with g(t) statistically. Accordingly, the error term in the model in equation (5.1) is in fact equal to

$$\varepsilon = g(t) + u \,. \tag{5.6}$$

If we assume t is uniformly sampled in a range, some functions $g(t)$ may generate a distribution close to normality, and consequently so does ε. However, in most cases the distribution of $g(t)$ will deviate from normality. Thus, in the presence of incomplete predictor variables even if u is normally distributed, ε is not statistically normal any more.

Besides the methods mentioned earlier, there is another possible way to solve the problem of non-normality, heteroscedasticity, outlier and removable non-additivity, which is pioneered by Box and Cox [1964]. They proposed a parametric family of transformations of variables, which were intended to provide homogeneity of variance, additive model and normality of the errors. Such transformation methodology has been proven quite successful in some contexts, but it has some limitations, for example, the physical meaning of predictor variables may get lost and variables have to be positive.

In summary, all the problems listed above can result in breakdown of some of GM assumptions, and some existing techniques might help handle some situations, but none of them are able to deal with all these problems. Furthermore, in practice it is generally hard to tell what causes the violation of GM assumptions and apply corresponding effective technique. Hence, it is of practical interest to come up a uniform framework to deal with all these problems simultaneously, which is possible if we note that all of them lead to non-normal overall error.

With the GM assumptions violated, the usual OLS estimator of β is in general biased and inconsistent. However, Schick [1996] shows that in the partly linear regression model with heteroscedastic errors an appropriately constructed weighted least squares (WLS) estimator can be consistent and more efficient than OLS estimators. Following a similar idea, we will generalize the usual weighted least squares, where weights are given, to parameterized weighted least squares (PWLS) and then estimate an optimal weighting function before performing WLS analysis, which will produce a better regression model than OLS.

To present and demonstrate the PWLS method, this chapter will be organized as follows. In section 5.2, the method of PWLS will be proposed as a uniform framework to handle the data and model problems mentioned above. Two methods, that is, maximum likelihood estimate and residual maximum likelihood estimation, will be applied to estimation the hyperparameter in the proposed family of error distributions in section 5.3. In section 5.4, a likelihood ratio test is applied to perform a significance test. Section 5.5 reports Monte

Carlo experiments results. The last section concludes this chapter.

5.2 Parameterized weighted Least-Squares

To define parameterized weighted least squares (PWLS), it is helpful to begin with usual weighted least squares.

In OLS, all the observations of data are treated equally without discrimination, or in other words, a flat weight function is assigned over the data. Assigning different weights to different data points gives rise to the weighted least squares (WLS), which is to minimize the weighted squared error

$$RSS_w = \sum_{i=1}^{n} w_i e_i^2 = \sum_{i=1}^{n} w_i (y_i - x_i \beta)^2, \tag{5.7}$$

where the weights associated with each data point are incorporated into the fitting criterion. The size of the weight indicates the precision of the information contained in the associated observation.

By a transformation on the variables, i.e. $y_i' = \sqrt{w_i}\, y_i$ and $x_i' = \sqrt{w_i}\, x_i$, with the aid of the solution of OLS it is easy to derive that

$$\hat{\beta}_{WLS} = (x^T W x)^{-1} x^T W y, \tag{5.8}$$

where W is a diagonal weight matrix that takes the weights as its diagonal entries.

WLS can be applied to diminish the effects of outliers and therefore gives a robust regression model.

In general, the weights are empirically chosen. Usually, empirical weights are a function of the residual error, which is in turn a function of the estimated coefficients that depends on the weights. In such a circumstance, an iterative method known as iteratively reweighted least squares (IRLS) can be utilized to estimate regression coefficients [for example, see Green, 1984 or Chen and Shao, 1993]. Basically, it works as follows:

1. Select initial weights $w^{(0)}$, such as uniform weights, i.e. w_i=1.

2. At each iteration t, solve for the WLS estimates of β as

   $$\hat{\beta}_{WLS}^{(t)} = (x^T W^{(t)} x)^{-1} x^T W^{(t)} y$$

3. Calculate the residuals $e_i^{(t)}$ and associated weights $w_i^{(t+1)} = w[e_i^{(t)}]$ for the next iteration.
4. Step 2 and 3 are repeated until the estimated coefficients converge.

In the above, the iteration starts with an OLS estimate as in Chen and Shao [1993], and as many authors noted it can result in different accuracy and efficiency when starting with different initial estimates [for example, Carroll, Wu and Ruppert, 1988], for instance, a feasible GLS estimate (FGLS) suggested by Inoue [1999] or more generally a WLS estimate proposed by Inoue [2003].

In the above procedure, w[·] refers to a weight function. For example, $w(e)=1/|e|$ is often used to deal with heteroscedasticity [Fuller and Rao,1978] so that the weights are inversely proportional to the magnitude of the residuals. In this case, after convergence the weighted residual sum of squares is

$$RSS_w = \sum_{i=1}^n w_i e_i^2 = \sum_{i=1}^n |y_i - x_i \beta|, \tag{5.9}$$

which is exactly the sum of absolute deviations. That is, if $w(e)=|e|^{-1}$, the WLS reduces to Least Absolute Error method.

Note that in the above W or Σ is assumed to be a diagonal matrix with all the off-diagonal elements zero, which implies errors are independent across observations. Often, however, this may be an unreasonable assumption (e.g. in time series, or in clustered data). We can relax this assumption and thus extend our results to a non-diagonal weight matrix, which corresponds to the case where the covariance matrix of noise Σ is not diagonal. we will obtain a similar generalized least squares (GLS) estimator

$$\hat{\beta}_{GLS} = (x^T W x)^{-1} x^T W y. \tag{5.10}$$

Aitken [1934] has shown that in general the generalized least squares estimate is the BLUE of β when $W = \Sigma^{-1}$, i.e. its variance achieves the lower bound $(\sum_{i=1}^n \sigma_i^{-2} x_i^T x_i)^{-1}$.

However, in practice we rarely possess exact knowledge of Σ, and therefore GLS is not operable, but an estimated or feasible generalized least squares (FGLS) should be employed instead, which is suggested by Inoue [1999]. FGLS estimators are often implemented in multiple steps: (1) an OLS analysis to yield estimated $\hat{e}_i$; (2) estimate Σ from the analysis of the $\hat{e}_i$ as $\hat{\Sigma}$, for example, regressing $\hat{e}_i^2$ on x_i or squares of x_i, which is based on the assumption that $\hat{e}_i^2$ depends on x_i; and finally, (3) computing the FGLS estimator with the estimated $\hat{\Sigma}$, i.e. $\hat{\beta}_{FGLS} = (x^T \hat{\Sigma}^{-1} x)^{-1} x^T \hat{\Sigma}^{-1} y$. It can be shown that if the heteroscedastic regression is correctly specified, FGLS is asymptotically equivalent to GLS, i.e. $\sqrt{n}(\hat{\beta}_{GLS} - \hat{\beta}_{FGLS}) \to 0$ in probability [e.g. see Davidson and MacKinnon, 1993].

To estimate the covariance matrix, it is beneficial to model it as a tunable function of the residual errors. That is, we can extend the usual weights in IRLS, i.e. $w(e)=1/|e|$, to a parametric family of weights

$$w_\alpha(e) = |e|^\alpha . \tag{5.11}$$

Such a weighting function finally makes the weighted sum of squared residual error in IRLS procedure converge to

$$RSS_w = \sum_{i=1}^n w_{\alpha i} e_i^2 = \sum_{i=1}^n |y_i - x_i\beta|^{2+\alpha} = \sum_{i=1}^n |y_i - x_i\beta|^\gamma , \tag{5.12}$$

which is often called L^γ fitting.

Although there exist various weighting strategies, all of them generally fall into two categories, no matter how complicated

(i) decrease the weights of those data points having large residual error, e.g. perhaps outliers, as in some robust regression methods like M-estimation [Huber, 1964]

(ii) increase the weights of those observations hard to learn as in boosting [see Drucker, 1997 and Avnimelech and Intrator, 1999]

If we set the range of α in (5.12) as from -1 to 1, we can achieve both classes in a unified weighting function, that is,

(i) if $\alpha<0$, the larger the residual error, the smaller the weight, while

(ii) if $\alpha>0$, the larger the residual error, the greater the weight.

Therefore, the advantage of such a weighting function as in (5.12) is that it enables us to realize both classes of weighting strategies through a single tunable hyperparameter.

Such a parametric family of weight functions can be also justified from another angle. It is well known that MLE is efficient if an efficient estimator exist, and further it is at least asymptotically efficient. In addition, as shown by Bradley [1973], in the presence of a noise with any distribution in an exponential family the MLE of β_i's can be estimated by weighted least squares. In order to estimate the optimal weights, we have to know the error distribution in advance. However, usually we have no precise information about either the error distribution or the variance component, and therefore what we can do is to estimate them from observations.

Without knowledge about the form of the error distribution, a possible way to estimate the error is to utilize a parametric family of error distributions, whose parameters can be estimated based upon data. A good candidate for such parametric family is the exponential power distribution (EPD) [Albers, 1998], because of its simplicity, close connection to WLS and other desirable properties.

EPD has probability density function

$$f(e\,|\,\gamma,\mu,\sigma)=\frac{\gamma}{2\sigma\cdot\Gamma[1/\gamma]}\exp\left(-\left|\frac{e-\mu}{\sigma}\right|^{\gamma}\right), \qquad (5.13)$$

where γ is the shape parameter, μ is the location parameter and σ is the scale parameter.

This family of distributions has the following statistical properties:

(1) It includes Gaussian (γ=2), Laplace (γ=1), and uniform distributions ($\gamma\to\infty$) as special cases;

(2) It is symmetric about μ but the scale parameter σ is not equal to the standard deviation;

(3) In view of the fact that $\Gamma(t)=\int_0^\infty e^{-u}u^{t-1}du$, it is easy to obtain

$$E(|e-\mu|^k \mid \sigma,\gamma)=\sigma^k \frac{\Gamma[(k+1)/\gamma]}{\Gamma[1/\gamma]}, \text{ where } k>1 \tag{5.14}$$

If we define kurtosis as

$$Kurtosis=\frac{E\left((e-\mu)^4\right)}{\left[E\left((e-\mu)^2\right)\right]^2}, \tag{5.15}$$

the kurtosis for EPD as a function of γ is computed as

$$Kurtosis_\gamma=\frac{\Gamma(5/\gamma)\Gamma(1/\gamma)}{\Gamma(3/\gamma)^2}. \tag{5.16}$$

From the above equation (5.16), it is easy to see that when γ is greater than 2, EPD is more peaked than the normal distribution, while when γ is smaller than 2, EPD is flatter than the normal distribution. Thus, comparing to the usual normal family and double exponential family, apart from location and scale parameters EPD has an extra kurtosis parameter, which can controls the distribution's deviation from normality and affects the tail of the distribution. By varying γ, it is possible to describe Gaussian, platykurtic and leptokurtic distributions (kurtosis larger than 3). The smaller is γ, the heavier the tails.

EPD provides us a wide class of statistical distribution to model both sub-Gaussian and super-Gaussian densities. It also enables us to approximate non-Gaussian error distribution by choosing appropriate values of γ.

Figure 5.1 shows a class of EPD density functions with μ zero, σ unit and γ ranging from 1 to positive infinity.

Now let's suppose the error ε in the equation (5.1) follows an EPD distribution with a given γ. Then the log-likelihood of an observation (x_i, y_i) is

$$LL(y_i,x_i \mid \beta,\gamma,\sigma^2)=\log\left(\frac{\gamma}{2\sigma\cdot\Gamma[1/\gamma]}\right)-\left|\frac{y_i-x_i\beta}{\sigma}\right|^\gamma. \tag{5.17}$$

By maximizing the log-likelihood of the observations

$$LL(y,x \mid \beta,\gamma,\sigma^2) = \sum_{i=1}^{n} LL(y_i,x_i \mid \beta,\gamma,\sigma^2)$$
$$= n\log\left(\frac{\gamma}{2\sigma \cdot \Gamma[1/\gamma]}\right) - \sum_{i=1}^{n}\left|\frac{y_i - x_i\beta}{\sigma}\right|^{\gamma} \quad , \qquad (5.18)$$

the maximum likelihood estimate of regression coefficients can be obtained as

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} \left|y_i - x_i\beta\right|^{\gamma}, \qquad (5.19)$$

which is exactly equivalent to L^{γ}fitting in the equation (5.12).

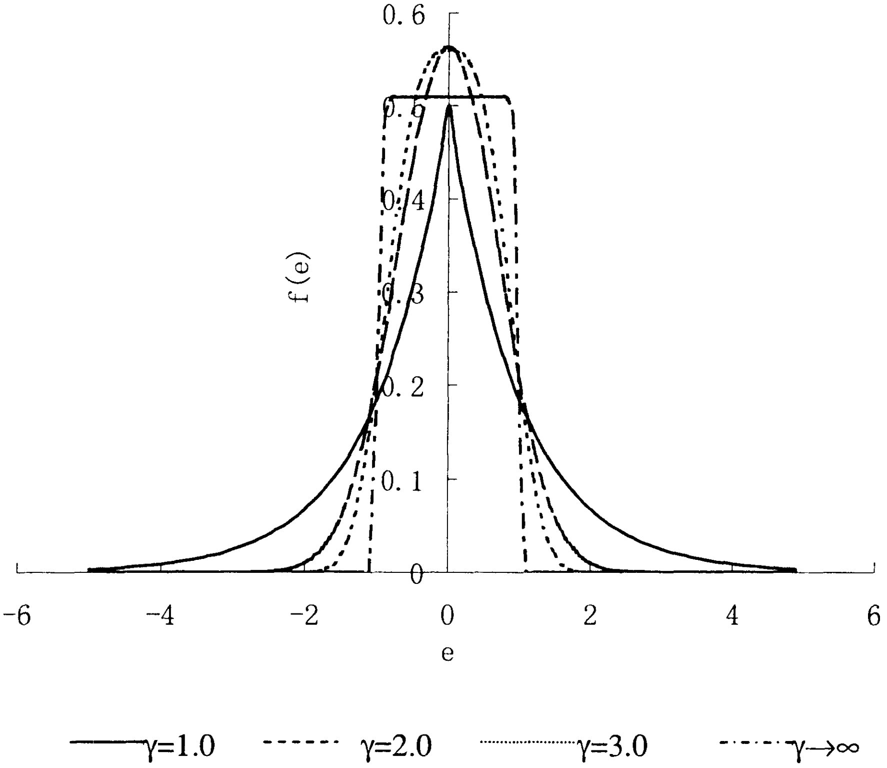


Figure 5.1 A class of EPD density functions

Therefore, with the shape parameter γalready known, it is easy to construct a regression model by WLS. However, this is rarely the case as we have very little information regarding the error distribution except for observations. Thus, we have to estimate the shape parameter γ somehow before performing WLS analysis.

5.3 Estimating optimal weight functions

As we mentioned earlier, we need to first estimate the error distribution and then estimate regression coefficients. To estimate the error distribution, we propose to apply the exponential power distribution with tunable parameters to approximate the error distribution. A classical way to do so is to follow the maximum entropy principle, which states that when an inference is made on the basis of incomplete information, it should be drawn from the probability distribution that maximizes the entropy, subject to constraints on the distributionTo maximize entropy is equivalent to minimizing the Kullback-Leibler (KL) divergence [Kullback, 1959] between the true distribution and the approximating distribution, which is defined as

$$I(g,f)=E_Y\left\{\log\frac{g(Y)}{f(Y)}\right\}=\int_{-\infty}^{+\infty}\log\frac{g(y)}{f(y)}g(y)dy, \qquad (5.20)$$

where g refers to the true distribution and f denotes the statistical model.

The KL divergence has the following properties:

(i) $I(g,f)\geq 0$,

(ii) $I(g,f)=0 \Leftrightarrow g(y)=f(y)$.

Note that

$$I(g,f)=E_Y\{\log g(Y)-\log f(Y)\}=\int_{-\infty}^{+\infty}\log g(y)\cdot g(y)dy-\int_{-\infty}^{+\infty}\log f(y)\cdot g(y)dy$$

$$=Const.-\int_{-\infty}^{+\infty}\log f(y)\cdot g(y)dy \qquad (5.21)$$

where $\int_{-\infty}^{+\infty}\log g(y)\cdot g(y)dy =: H(Y)$, which is constant although unknown.

Hence, in order to minimize the KL divergence, we need to maximize the empirical likelihood of the observations, which is $\frac{1}{n}\sum_{i=1}^{n}\log f(y_i)$.

In the current case, if we suppose the true error distribution is $g(Y|X,\beta)$, then we obtain

$$I(g(Y\mid X,\beta),f(Y\mid\beta,X,\gamma,\sigma))=E_Y\{g(Y\mid X,\beta)\}-\int_{-\infty}^{+\infty}\log f(y\mid\beta,x,\gamma,\sigma)g(y\mid x,\beta)dy$$

$$\approx Const. - \frac{1}{n}\sum_{i=1}^{n} \log f(y_i \mid \beta, x_i, \gamma, \sigma). \tag{5.22}$$

Finally, we obtain the estimates as

$$\begin{aligned}(\hat{\gamma}, \hat{\beta}) &= \arg\max_{\gamma,\beta} \sum_{i=1}^{n} \log f(y_i \mid \beta, x_i, \gamma, \sigma) \\ &= \arg\max_{\gamma,\beta} \left\{ n \log\left(\frac{\gamma}{2\sigma \cdot \Gamma[1/\gamma]} \right) - \sum_{i=1}^{n} \left| \frac{y_i - x_i\beta}{\sigma} \right|^{\gamma} \right\},\end{aligned} \tag{5.23}$$

which is exactly the maximum likelihood estimate of parameters. So even if the true error distribution is not in the parametric family, asymptotically the one closest to the truth with the family will be found out.

5.3.1 Maximum likelihood estimation

The MLEs in equation (5.23) can be analytically obtained by setting each first partial derivative to zero, that is

$$\frac{\partial l(y)}{\partial \beta_j} = \gamma \sum_{i=1}^{n} sign(y_i - \sum_{j=1}^{p} x_{ij}\beta_j) \cdot \left| y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right|^{\gamma-1} \cdot x_{ij} = 0, \text{ for } j=1,\dots,p \tag{5.24}$$

$$\frac{\partial l(y)}{\partial \sigma} = -\frac{n}{\sigma} + \frac{\gamma}{\sigma^{\gamma+1}} \sum_{i=1}^{n} \left| y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right|^{\gamma} = 0, \tag{5.25}$$

and

$$\frac{\partial l(y)}{\partial \gamma} = \frac{n}{\gamma} - n(\log \Gamma(1/\gamma))' - \sum_{i=1}^{n} \left| \frac{y_i - x_i\beta}{\sigma} \right|^{\gamma} \log\left| \frac{y_i - x_i\beta}{\sigma} \right| = 0. \tag{5.26}$$

These equations are hard to solve simultaneously for γ, σ, and β_j, $j=1,\dots,p$, due to their complex forms. However, the difficulty can be circumvented by some kind of iterative procedures, in which we

(1) first set some initial value for γ,

(2) solve the equation (5.24) for β_j, $j=1,\dots,p$, given γ,

(3) solve the equation (5.25) for σ, given γ and β_j, and finally

(4) solve the equation (5.26) for γ given β_j and σ.

(5) repeat the above steps (2)-(4) until convergence.

Even with such an iterative procedure, the equations (5.24) and (5.25) are still not easy to solve analytically, and thus some numerical methods have to be applied.

Note that equation (5.24) is actually equivalent to equation (5.19), an L_p norm optimization problem, and therefore it can be solved by iterative re-weighted least squares (IRLS), with the weights set as $w_\gamma(e) = |e|^{\gamma-2}$ in each iteration.

Meanwhile, since equation (5.26) involves the power function and the derivative of the Gamma function, another numerical iterative method has to be utilized to solve it for γ. To do so, we can rewrite it as

$$\frac{1}{\gamma^{(k+1)}} = \left(\log\Gamma(1/\gamma^{(k)})\right)' + \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - x_i\beta}{\sigma}\right|^{\gamma^{(i)}} \log\left|\frac{y_i - x_i\beta}{\sigma}\right|, \tag{5.27}$$

where k is the number of iteration. Furthermore, the derivative of the gamma function can be computed numerically, i.e.,

$$\Gamma'(z) = -\Gamma(z)\left[\frac{1}{z} + \lambda + \sum_{n=1}^{\infty}\left(\frac{1}{n+z} - \frac{1}{n}\right)\right], \tag{5.28}$$

where λ is the Euler-Mascheroni constant, approximated by 0.5772.

In practice, the exact value of γ may be not so important as long as we can know whether γ is greater or smaller than 2, for some reasons that will be discussed in more detail later on in section 5.3.3. Therefore, simple numerical grid search procedure can be applied in place of the one above. The procedure can be described as follows.

(1) Generate a grid of values of γ evenly distributed between [1,3].

For each γ solve the equation (5.24) for β_j, $j=1,\ldots,p$, using IRLS, that is

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} |y_i - x_i\beta|^{\gamma}. \tag{5.29}$$

(2) Estimate σ as

$$\hat{\sigma} = \left(\frac{\gamma}{n}\sum_{i=1}^{n} | y_i - x_i\beta |^{\gamma}\right)^{1/\gamma}. \tag{5.30}$$

Compute the maximum likelihood for each γ as

$$l(y) = n\log\left(\frac{\gamma}{2\hat{\sigma}\cdot\Gamma[1/\gamma]}\right) - \sum_{i=1}^{n}\left|\frac{y_i - x_i\hat{\beta}}{\hat{\sigma}}\right|^{\gamma}. \tag{5.31}$$

The γ corresponding to maximum $l(y)$ will be taken as the MLE of γ.

5.3.2 Residual maximum likelihood estimation (REML)

In the above section, we described an MLE procedure to estimate both the error distribution and the regression model. However, it is well known that usually MLE does not produce unbiased estimators because it does not take into account the loss of degrees of freedom that results from estimating other parameters [Harville, 1977]. For example, the ML estimate of the variance for a normal distribution is $S_{MLE}^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$ whereas the unbiased estimate of variance is $S_{unb}^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$. The estimator in this example is biased because it ignores the loss of one degree of freedom due to the estimation of the sample mean. This disadvantage of the MLE can be overcome by the Restricted Maximum Likelihood (REML) technique, which was proposed by Patterson and Thompson [1971] to estimate the variance components in a linear mixed-effects model. Since then, REML has become the method of choice for estimation of the variance components in a linear mixed model. For derivation of REML, refer to Patterson and Thompson [1971] and Harville [1974]. In this chapter, we will extend the REML method in the current case to estimate γ.

To review a little about REML, let's first present an explicit definition of linear mixed-effects models, or simpler linear mixed model. A general form is

$$y = x\beta + zb + \varepsilon, \tag{5.32}$$

where y is the n-dimensional response vector, z is an $n \times p$ model matrix for the fixed-effect and z is an $n \times q$ model matrix for the random-effect, β is a vector of fixed effect coefficients , b and ε are vectors of random variables with mean 0, assumed Gaussian usually and uncorrelated with each other.

Let's suppose $b \sim N(0, \sigma^2\Gamma)$ and $\varepsilon \sim N(0,\sigma^2 I)$, and $cov(b, \varepsilon)=0$. If we let $v=zb+\varepsilon$, then $v \sim N(0,\sigma^2(z\Gamma z^T+I))$ and $y \sim N(x\beta,\sigma^2(z\Gamma z^T+I))$. Knowing this, it is easy to find out the

MLE of β by generalized least squares.

For example, we further let $\Gamma=\lambda I$ as in Patterson and Thompson [1971]. Hartley and Rao [1967] estimate β, λ and σ^2 by maximum likelihood, while Patterson and Thompson [1971] propose to use REML, which separates the estimation of β from that of λ and σ^2.

REML differs from ML in that the likelihood of the data is maximized only for the random effects, thus REML is a restricted solution. Rather than using y directly, REML is based on linear combinations of y, say, Sy, where S is an $n\times n$ matrix. S is chosen in such a way that Sy does not depend on the mean value of y, in other words, independent of β no matter what their value is, i.e. $Sy=S(x\beta+v)=Sv$. If we suppose the log-likelihood of a data vector y is $LL(y)$, the log-likelihood of Sy is

$$LL(Sy) = LL(y) + \log\det(S)\,, \tag{5.33}$$

where S is independent of β and thus independent of the data y. Therefore, maximizing $LL(y)$ with regard to λ amounts to maximizing $LL(Sy)$ with regard to λ. In so doing, the estimation of other hyperparameters like λ in the current case can be separated from the estimation of β.

S can be chosen by finding an S such that

(i) $var(Sy)$ is positive definite.

(ii) $E(Sy)=0$, i.e. $Sx=0$.

(iii) $Rank(S)=n\text{-}p$, that is, S has $n\text{-}p$ linearly independent rows.

A suitable matrix S was suggested by Patterson and Thompson [1971] as

$$S = I - x(x^T x)^{-1} x^T\,. \tag{5.34}$$

It is easy to verify that $Sx=0$ holds. From $Sx=0$, we know that S is orthogonal to the column space of x, whose rank is equal to p, therefore S lies in the $(n\text{-}p)$-dimensional orthogonal complement of the column space of x, i.e. $rank(S)=n\text{-}p$.

With such an S, Sy turns out to be equivalent to the residuals obtained by fitting the data using OLS. Therefore, the basic idea of REML is to perform maximum likelihood on the predicted residual, which is calculated by OLS method. This is why REML is also called residual maximum likelihood. A good thing is that our estimator is invariant to the choice of Sy except for a constant, provided that we satisfy the three conditions above.

Now the REML equations can be directly derived from the ML equations in (5.17) by making suitable replacements, i.e. y by Sy and x by Sx. Following a similar idea, we can separate the estimation of the error distribution from the estimation of regression coefficients, because

$$Sy = S(x\beta + \varepsilon) = S\varepsilon, \tag{5.35}$$

where ε follows an exponential power distribution and thus it is easy to obtain the likelihood function for Sy.

Finally, we obtain

$$LL_{REML}(y_i, x_i \mid \beta, \gamma, \sigma^2) = \log\left(\frac{\gamma}{2\sigma \cdot \Gamma[1/\gamma]}\right) - \left|\frac{e_i}{\sigma}\right|^{\gamma}, \tag{5.36}$$

where e_i is calculated after fitting using OLS, i.e. $e = \lfloor I - x(x^T x)^{-1} x^T \rfloor y$.

Similar to MLE, maximizing the above LL_{REML} with regard to both $\gamma$and σproduces the REML estimator of γ. Since the REML estimator of γ is obtained by maximizing the marginal log-likelihood, β is not involved in it. After estimating γ, the next step is to estimate the linear regression model given the error distribution, which can be done by an IRLS procedure.

Once again, maximizing LL_{REML} involves differentiating the gamma function as well as calculating the root of a complex function, and it is helpful to apply a simple numerical line search method instead, which can be listed as follows:

(1) Generate a set of values of γ evenly distributed between [1,3].

(2) Estimate σ as

$$\hat{\sigma} = \left(\frac{\gamma}{n}\sum_{i=1}^{n} | e_i |^{\gamma}\right)^{1/\gamma}. \tag{5.37}$$

(3) Compute maximum likelihood for each γ as

$$l_{REML}(y) = n\log\left(\frac{\gamma}{2\hat{\sigma}\cdot\Gamma[1/\gamma]}\right) - \sum_{i=1}^{n}\left|\frac{e_i}{\hat{\sigma}}\right|^{\gamma}. \qquad (5.38)$$

The γcorresponding to maximum $l_{REML}(y)$ will be taken as the REML estimate of γ.

Under mild conditions, the REML and maximum likelihood estimators are asymptotically equivalent [see Cressi and Lahiri, 1993, Richardson and Welsh, 1994]. However, it is easy to note that this method based on the residuals is simpler than the ML method as it performs IRLS only once. In addition, it separates the estimation of error distribution, i.e. γ, and regression coefficient, namely β, thereby simplifying the problem a lot.

Our simulation study shows that when the sample size is relatively small, ML method is a little more stable, while when the sample size is large, REML performs better.

5.3.3 Shrinking weight functions

A weakness of WLS observed by many researchers, for example, Hooper [1993], is that sometimes the error of some data points tend to be underestimated and therefore are assigned overly great weights. For example, if the estimated error of an observation happens to be near 0, its corresponding weight will be markedly high. This is part of the reason why the WLS estimator may not be better than OLS estimator when the group size is no greater than 2 as indicated by Chen and Shao [1993]. To address this problem, Hooper [1993] proposed to apply empirical Bayesian estimator and based upon his study he recommended shrinking large weights towards 0 by, for example, adding a positive constant to each residual error estimate. If we extend this idea, we can construct a shrunk weight function as

$$w_{\alpha}(e) = |\lambda + e|^{\alpha} \qquad (5.39)$$

where λ is an appropriately chosen positive constant, for example, the estimated scale parameter.

If we plug the above weights into the equation (5.12), we obtain

$$RSS_{w} = \sum_{i=1}^{n} w_{\alpha i} e_i^2 = \sum_{i=1}^{n} \frac{|y_i - x_i\beta|^{2+\alpha}}{|\lambda / | y_i - x_i\beta | + 1|^{-\alpha}} = \sum_{i=1}^{n} \rho(e_i, \lambda) |y_i - x_i\beta|^{\gamma}, \tag{5.40}$$

where when α<0, $\rho(e_i,\lambda)$<1 and increases monotonically with e_i increasing, and on the other hand when α>0, $\rho(e_i,\lambda)$>1 and decreases with e_i. The above equation (5.40) can be viewed as weighted maximum likelihood estimate. Therefore, the final effect is equivalent to shrink γ towards 2, i.e. the normal distribution. As a matter of fact, if we apply Bayesian approaches and assume a probability density function for λ with mean value 2, we will end up with a similar result, i.e. shrinking γ towards prior mean 2.

In fact, in our simulation study the hyperparameter γ tends to be overestimated especially with maximum likelihood estimate (MLE). This may be due to the inefficiency of MLE under some situations. By shrinking γ towards 2 under mild deviation of the error distribution from normality, better performance of PWLS can be achieved.

In fact, the shrinkage of the hyperparameter γ depends on the log-likelihood difference that will be introduced in the next section. If difference is significant and we would say that one hypothesis is dominating the other, and in such a case the shrinkage is not necessary.

5.4 Significance test

In the last section, we applied Maximum Likelihood method to estimate the approximate error distribution or equivalently the hyperparameter γ in parametric WLS. However, this class of ML methods can not guarantee us that so estimated PWLS regression model will certainly outperform OLS estimate at least for two reasons:

(1) The GM assumptions do hold and the deviation from normality is just attributed to statistical fluctuation especially resulting from small sample size;

(2) Adding more parameters to the model increases the variance of estimation of parameters, which might finally lead to larger overall mean squared error.

Taylor and Siqueira [1996] discusses in detail about the cost of adding parameters to a model, which is also applicable to our case.

According to statistics text, we may define this kind of error as Type I error or false alarm. One way to avoid such error and increase our confidence is to perform significance test. A test of choice is the likelihood ratio test as it is a powerful, very general likelihood-based method of testing model assumptions. In the current case, we will test the null hypothesis that the error distribution is best represented by a normal distribution against a composite alternative hypothesis that the error distribution is not normal. Equivalently, the hypothesis of interest is $H_0 : \gamma = 2$ and $H_1 : \gamma \neq 2$.

In Likelihood Ratio Tests (LRT), an important assumption can be restated as a reduction or restriction on the number of parameters used to formulate the likelihood function of the data. In all these cases, there is a simple and very useful way to test whether the assumption is consistent with the data. If we define L_1 be the maximum value of the likelihood of the data without the additional assumption and also let L_0 be the maximum value of the likelihood when the parameters are restricted (and reduced in number) based on the assumption H_0. Assume k parameters were lost by adding restrictions (i.e., L_0 has k less parameters than L_1). According to the Wilks theory [Wilks, 1963], 2 times the log maximum likelihood ratio approximately follows a Chi-square distribution with k degrees of freedom as n tends to infinity, i.e.

$$LR = -2\log\lambda = 2(\log L_1 - \log L_0) \xrightarrow{D} \chi^2(k), \tag{5.41}$$

where $\lambda = \frac{L_0}{L_1}$, which is always between 0 and 1, and $\xrightarrow{D}$ means converges in distribution. The above approximation in (5.41) is usually good, even for small sample sizes.

The null hypothesis, H_0, will be rejected if LR is larger than a Chi-Square percentile with k degrees of freedom, where the percentile corresponds to the confidence level chosen by the analyst such as 95%.

In the present case, for ML method we define

$$\begin{aligned}\log L_1 &= LL(y,x \mid \hat{\beta}_{PWLS}, \hat{\gamma}, \hat{\sigma}^2) = \sum_{i=1}^{n} LL(y_i, x_i \mid \hat{\beta}_{PWLS}, \hat{\gamma}, \hat{\sigma}^2) \\ &= n \log\left(\frac{\hat{\gamma}}{2\hat{\sigma} \cdot \Gamma[1/\hat{\gamma}]} \right) - \sum_{i=1}^{n} \left| \frac{y_i - x_i \hat{\beta}_{PWLS}}{\hat{\sigma}} \right|^{\hat{\gamma}}\end{aligned} \tag{5.42}$$

and

$$\begin{aligned}\log L_0 &= LL(y,x \mid \hat{\beta}_{OLS}, \hat{\sigma}^2) = \sum_{i=1}^{n} LL(y_i, x_i \mid \hat{\beta}_{OLS}, \hat{\sigma}^2) \\ &= \frac{n}{2} \log\left(\frac{1}{2\pi\hat{\sigma}^2} \right) - \sum_{i=1}^{n} \frac{(y_i - x_i \hat{\beta}_{OLS})^2}{2\hat{\sigma}^2},\end{aligned} \tag{5.43}$$

where $\hat{\beta}_{\bullet}, \hat{\gamma}, \hat{\sigma}$ denote maximum likelihood estimates under respective hypotheses.

Since the lost degrees of freedom k by setting γ as 2 is just 1, we have approximately

$$LR = 2 \log \frac{LL(y,x \mid \hat{\beta}_{PWLS}, \hat{\gamma}, \hat{\sigma}^2)}{LL(y,x \mid \hat{\beta}_{OLS}, \hat{\sigma}^2)} \sim \chi^2(1). \tag{5.44}$$

If we choose 95% as the confidence level, then we obtain the decision rule, i.e. if

$$LR > \chi^2_{0.95}(1) = 3.841, \tag{5.45}$$

the assumption that the error distribution is normal will be rejected with a confidence of 95% and the parametric WLS method rather than OLS should be applied instead. If we choose 90% as confidence level, the criterion becomes LR>2.706.

In fact, if we reject the normal assumption if $LR > 2$, this decision rule is equivalent to Akaike's Information Criterion (AIC) [Akaike, 1973].

As for RMEL, we developed corresponding residual maximum likelihood ratio test, in which L_0 and L_1 are defined as follows

$$\log L_1 = LL(e \mid \hat{\gamma}, \hat{\sigma}) = n \log\left(\frac{\hat{\gamma}}{2\hat{\sigma} \cdot \Gamma[1/\hat{\gamma}]} \right) - \sum_{i=1}^{n} \left| \frac{e_i}{\hat{\sigma}} \right|^{\hat{\gamma}}, \tag{5.46}$$

and

$$\log L_0 = LL(y,x \mid \hat{\beta}_{OLS}, \hat{\sigma}^2) == \frac{n}{2} \log\left(\frac{1}{2\pi\hat{\sigma}^2} \right) - \sum_{i=1}^{n} \frac{e_i^2}{2\hat{\sigma}^2}, \tag{5.47}$$

where e_i's are equivalent to residuals resulting from OLS.

Similar to the significance test for MLE, if

$$LR_{REML} = 2(\log L_1 - \log L_0) > \chi^2_{0.95}(1) = 3.841,$$

the assumption that the error distribution is normal will be rejected at a confidence level of 95%.

With this significance test, our procedure will adaptively pick PWLS if the error distribution severely deviates from normality, or select OLS if the departure is just mild.

5.5 Monte Carlo simulation study

In the above sections, we developed a uniform framework, i.e. parametric weighted least-squares (PWLS), to handle the departure of error distribution from normality caused by non-Gaussian error, outliers, heteroscedasticity or incomplete predictor variables. In this section, we will show how this method works in practice by means of Monte Carlo simulations.

To provide a general example, we will use cosine series as predictor variables to fit a data set generated by a smooth function plus some disturbance, i.e.

$$y = f(x) + \varepsilon, \tag{5.48}$$

where ε can be any arbitrary but symmetric distribution.

The set of predictor variables we utilized in our simulations is $\{\cos(\pi x), \cos(2\pi x), \cos(3\pi x), \cos(4\pi x), \cos(5\pi x)\}$, and the smooth function is

$$f(x)=\mathrm{sinc}(3x)=\sin(3x)/(3x),\ x\in[-1,1] \tag{5.49}$$

which can be well approximated by the chosen cosine series and therefore the set of predictor variables can be deemed as complete.

Since in the current simulation study we assume the true model is already known, therefore we can define a global mean prediction error to measure how good an estimated regression model is. Conceptually, the global mean prediction error is defined as

$$GMPE = \frac{1}{2L}\int_{-L}^{L}\left(f(t) - \hat{y}(t)\right)^2 dt, \tag{5.50}$$

where $\hat{y}(t)$ is the estimated model, but in practice, it can be approximated by

$$GMPE \approx \frac{1}{m}\sum_{i=1}^{m}\left(f(t_i)-\hat{y}(t_i)\right)^2, \tag{5.51}$$

where t_i is typically evenly distributed in [$-L$, L] and m is large enough.

To evaluate the performance of PWLS method, we used OLS as a benchmark, and compared their performance by defining a ratio as

$$R_{OLS/PWLS} = \frac{GMPE_{OLS}}{GMPE_{PWLS}}. \tag{5.52}$$

The average ratio is defined as its geometric mean

$$\overline{R}_{OLS/PWLS} = \exp\left(\frac{1}{N}\sum_{i=1}^{N}\log(R_{OLS/PWLS})_i\right), \tag{5.53}$$

where N is total number of simulations. The average ratio is so defined that it helps reduce the influence of some unusual cases.

To see how this uniform method can handle all these problems, we designed four different experiments.

The first one is intended to test its performance in face of non-Gaussian error distribution. To this end, we generated errors of Laplace or double exponential distribution.

Secondly, to test its ability to detect outliers and reduce their influence, we let the contaminant distribution as

$$f_s(s) = (1-2p)\delta(s) + p\delta(s-c) + p\delta(s+c), \tag{5.54}$$

where p=0.1 and c=0.3.

The third one is to show how this method handles heterogeneity of variance. To do this, the variance component is modeled as log-normal and mixture of Gaussian (MoG) with median value σ^2, that is, $e \sim N(0,\sigma_e^2)$, where $\ln(\sigma_e) \sim N(0,1)$

and

$$f_e(e) = p \cdot N(0,\sigma_{e1}^2) + (1-p)\cdot N(0,\sigma_{e2}^2). \tag{5.55}$$

The last experiment is to show the influence of omission of predictor variables and how PWLS is also able to deal with it. In this experiment, to simulate the effect of omitted predictor variables another function is chosen as $f(x)$ in (5.48),

$$f(x)=\begin{cases}1, if\ x\in[-0.5,0.5] \\ -1, \quad otherwise\end{cases}, \tag{5.56}$$

which can not be approximated by the included predictor variables so well. f(x) is so chosen that the distribution of g(t) in the equation (5.6) departs from normality.

The results for this experiment show that PWLS can improve OLS very little if those omitted predictor variables are not so significant. In this experiment, γ is shrunken to 0.5 or –0.5.

The shrinkage of the hyperparameter γ is implemented by shrinking γ by 0.2 towards 0 in all these experiments.

The simulation results of all these experiments are shown in Table 5.1 and Table 5.2 for both ML and REML, respectively. From these results, ML and REML are easy to be compared as well.

Table 5.1 Monte Carlo simulation results for ML

Noises		ML		
		$AGMPE_{OLS}$(1)	$AGMPE_{PWLS}$(2)	$R_{OLS/PWLS}$(3)
Non-normal		1.64E-03	1.41E-03	1.18
Outliers		4.58E-03	4.23E-03	1.12
Hetero-skedasticity	Log-normal	1.19E-02	2.36E-03	3.61
	MOG	7.25E-03	6.24E-03	1.2
Partly linear		7.79E-02	7.49E-02	1.04
Ideal		1.60E-03	1.64E-03	0.98

(1)Average GMPE of OLS Method

(2)Average GMPE of PWLS Method

(3) Average C/F Ratio of Prediction Errors

The last interesting question is that what the PWLS would do if the errors do follow the normal distribution. This is to test the probability of making wrong decision on the error distribution. For this purpose, simulations with ideal normal errors were also performed, whose results are shown in Table 5.1 and Table 5.2. The results show that when it is applied to the normal regression, it amounts to the OLS.

Table 5.2 Monte Carlo simulation results for REML

Noises		REML		
		$AGMPE_{OLS}$	$AGMPE_{PWLS}$	$R_{OLS/PWLS}$
Non-normal		1.64E-03	1.52E-03	1.1
Outliers		4.58E-03	4.46E-03	1.044
Hetero-skedasticity	Log-normal	1.18E-02	2.86E-03	3.21
	MOG	7.25E-03	6.90E-03	1.07
Partly linear		7.79E-02	7.48E-02	1.04
Ideal		1.60E-03	1.61E-03	0.999

In all the above experiments, we first collect a group of sample data points from the generating model and fit a multiple regression model to the data set using different methods. The sample size is set to be 40 and each experiment is repeated many times to obtain an average performance evaluation.

From these Monte Carlo simulations, we can see some facts.

First of all, the PWLS estimate using both ML and REML performs better than OLS estimate in all the four kinds of contexts where the overall distribution deviates from normality. In the presence of heteroskedasticity, the improvement of PWLS over OLS is the most remarkable.

Secondly, under the assumption of ideal normal errors the PWLS method is almost equivalent to OLS. In other words, if we define the error that PWLS decides that errors are not normal while in fact they are as type I or false alarm error, the probability of such error is very low, which means PWLS succeeds in avoiding the downside.

Finally, in most situations ML is on average a little bit better than REML, but REML has a slightly lower probability of type I error.

As expected based upon our derivation, it was also observed in our numerical experiment that the improvement of PWLS over OLS depends on how severely the error distribution departs from normality, that is, the more severe the deviation, the greater the improvement. In the case of mild deviation, the PWLS may not help or even get worse because the more parameters to estimate, the higher the variance of the estimate.

5.6 Conclusion

To remedy some data problems or predictor variable problems, such as non-normal error, outliers, heteroskedasticity as well as incomplete predictor variables, a parametric weighted least squared method is proposed, which is built on weighted least squares with estimated optimal weights or equivalently approximates the error distribution with exponential power distribution. Two methods, namely ML and REML, are also suggested to estimate the hyper-parameter γin the family of exponential power distribution or αin the weight function. Finally, Monte Carlo simulations study is conducted to test the performance of the PWLS method. Based upon the comparison of PWLS and OLS, we can conclude that in those contexts described in this chapter PWLS does outperform OLS. From the simulation results, we also obtain some insight about where PWLS can be of great help.

Chapter 6

Regression Model Uncertainty Reduction by Constrained Parametric Weighted Least Squares

6.1 Introduction

A model typically in a mathematical form is a simplified representation of a system of interest intended to help understand and predict the behavior of the system. Generally, a model consists of two parts, i.e. model specification or model structure, which reflects important structural assumptions about the system under consideration, and model parameters, which are usually estimated based upon some data.

Therefore, intuitively a model formulates how a response variable is related to inputs, but it cannot predict the response variable precisely given inputs x. This is how model uncertainty arises. Corresponding to the constituents of a model, model uncertainty comes from two sources, i.e. model structural uncertainty and model parameter uncertainty. The uncertainty in the choice of link function and error distribution in a generalized linear model [Nelder and Wedderburn, 1972] serves as an example of model structural uncertainty and the uncertainty in regression coefficients exemplifies model parameter uncertainty.

As many authors pointed out, for example, Draper [1995] and Burnham and Anderson [2002], it is common in statistics to acknowledge model parameter given a specific assumed model structure, while it is less common to admit model structure uncertainty. Consequently, the model structure uncertainty fails to be incorporated into model uncertainty analysis, thereby leading to the underestimate of predictive uncertainty about the response variable and therefore overconfident decisions. Draper [1995] illustrates the consequences of unacknowledged structural uncertainty with some examples.

Model uncertainty is one kind of epistemic uncertainty, which is attributed to lack of knowledge. Since the cause of model uncertainty is lack of knowledge, increasing the knowledge base might reduce such uncertainty. Again take the generalized linear model as an example. If we understand the system much better and thus we can specify the link function and error distribution more appropriately, the model structural uncertainty can be reduced significantly; likewise, if we can collect more data or extract more useful information out of data, we can estimate the regression parameter more precisely as well.

It is worth noting that besides model uncertainty, another important component of prediction error of a model is model bias. However, in most cases bias is dominated by variance, although generally both model bias and uncertainty contribute to the predictive error of a model. As such, model bias is not of our concern in this chapter and we will primarily focus our attention on model uncertainty. Therefore, throughout this chapter, we will treat the prediction error and model uncertainty equivalently.

Following this line of thought, the prediction error of a multiple regression model has two parts, namely model structure and model parameter. Besides the assumption of linearity, model structure in a multiple regression model also includes the choice of a set of predictor variables and the specification of the error distribution. The model parameter refers obviously to regression coefficients, which are usually estimated from data.

There are some existing techniques that help reduce the model structural uncertainty, such as model selection and shrinkage estimator. For this reason, model structural uncertainty is sometimes called model selection uncertainty, for example see Burnham and Anderson [2002]. Since variable subset selection methods are instable and the predictive error is remarkably large [see Miller, 1984 or Breiman, 1996], although they can reduce model structural uncertainty, shrinkage methods, for example, ridge regression [Hoerl and Kennard, 1970 and Tikhonov and Arsenin, 1977], LASSO [Tibshirani, 1996] and negative garotte [Breiman, 1995], are usually preferred, because shrinkage is a generalized smoothing method and helps reduce the instability of regression coefficient estimation. In order to combine the advantages of these two classes of methods, chapter 4 puts forward a new shrinkage-based fuzzy variable selection strategy, which is proven to outperform classical model selection methods by numerical simulations. Such a framework incorporates both model selection and constrained regression method and model structure is simply represented by a vector of membership grade.

The other source of model uncertainty, namely parameter uncertainty, results from parameter estimation, which can be expressed as a function of the variance of estimated regression coefficients. Under Guassian-Markov (GM) assumptions, the ordinary least-suqares (OLS) estimator has been proven the best linear unbiased estimator (BLUE) and statistically we can do nothing more to improve its efficiency. However, in reality GM assumptions cannot always be met, and thus the OLS estimator is not BLUE any more, for example, non-normal error, outliers, inconstant variance as well as incomplete set of predictor variables. There exist some techniques, such as generalized linear models [see Nelder and Wedderburn, 1972], robust regression like *M*-estimator [Huber, 1964], transformation of response variables [Box and Cox, 1964] and generalized least-squares, that may help handle some of the above problems, but cannot handle them all at a time. For this reason, chapter 5 proposed a new method, called parametric weighted least-squares, as a uniform framework to deal with all those problems, thereby reducing the parametric uncertainty.

In order to reduce the overall prediction error, we need to reduce both its two components, that is, model selection uncertainty and parameter estimation uncertainty. To this end, combining a stable model selection method with a more efficient regression method might be a feasible way, for example, combining such constrained least squares as Ridge Regression, LASSO or Non-negative garotee with weighted least squares. Unfortunately, up to now this so called constrained weighted least squares has been rarely utilized directly in multiple linear regression analysis, although we do witness some application in maximum likelihood estimation with constraints. For example, Tibshirani [1997] suggested using lasso for variable selection in the Cox model, which is finally realized by a constrained weighted least squares procedure.

In this chapter, we will make the fuzzy variable selection method and parametric weighted least-squares work together to reduce both components of model uncertainty.

This chapter is organized as follows. In section 6.2, the newly developed fuzzy variable selection and parametric weighted least-squares (PWLS) will be briefly reviewed. Section 6.3 will generalize the fuzzy variable selection method to the case of PWLS and present a two-stage optimization algorithm to implement the new method in practice. In section 6.4, an approach is proposed to evaluate the model uncertainty, which takes into account both model structure uncertainty and parameter uncertainty. In section 6.5, the results of a numerical study will be presented. Finally, section 6.6 concludes this chapter.

6.2 Constrained parametric weighted least squares (CPWLS)

In section, we will first review something about fuzzy variable selection and parametric weighted least squares and then generalize them, and finally come up with a two-stage iterative procedure, which incorporates both methods.

6.2.1 Fuzzy variable selection

A multiple linear regression model is usually written as

$$Y = \sum_{i=1}^{p} \beta_i X_i + \varepsilon , \tag{6.1}$$

where Y refers to the response variable, X_i's are predictor variable, β_i's are regression coefficients to be estimated, and ε denotes the error term.

Variable selection is an essential part of regression analysis. The purpose of variable selection is to choose an optimal subset of predictor variables such that the prediction error of the resultant regression model is minimized. To achieve goal, a variety of statistical model selection methods have been developed, for example, classical hypothesis testing, penalized maximum likelihood, Bayes methods, information criteria like Akaike's Information Criterion (AIC) [Akaike, 1973] and Schwarz's Bayesian Information Criterion

(BIC) [Schwarz, 1978], and cross-validation. Most of these methods follow the principle of parsimony, in which model simplicity is somehow balanced against goodness-of-fit.

If we use $C:=\{c_1, c_2, \ldots, c_p\}$ to denote the full set of potential predictor variables in the case of linear regression, there are total 2^p possible subset choices. Simply, we may characterize classical variable selection with a model selection vector $m = (m_1,\ldots,m_p)$, where m_i is either 1 or 0 for i=1,…,p. if m_i=1, it means that the i-th regressor is chosen; while if m_i=0, it is not.

In both statistical theory and practice, it is shown that an appropriate variable selection method can improve the efficiency of a model and reduce its prediction error significantly. Further more, classical model selection produces an interpretable framework, that is, each predictor is either chosen or not, especially when the predictor variables have certain corresponding physical meanings. However, as we mentioned earlier, subset selection is instable mainly because it only select a single best model, and may have considerably large prediction error compared to some other shrinkage estimator such as ridge regression and other constrained least squares. In order to combine the advantages of classical model selection and shrinkage estimator methods, chapter 4 generalized the conventional model selection by introducing the concept of fuzzy model selection. Again, we can characterize classical variable selection with a vector of membership grade, i.e. $m = (m_1,\ldots,m_p)$, where m_i can be any value between 0 and 1 rather than restricted to 0 or 1 as in classical variable selection. In the fuzzy model selection scheme, m_i specifies in which degree the corresponding predictor variable belongs to the best subset, and thus it can be used to control a regressor's contribution to the model.

Similar to the crisp model selection, we can construct a selected regression model as

$$Y = \sum_{i=1}^{p} m_i \hat{\beta}_i X_i \text{, with } m_i \in [0,1]. \qquad (6.2)$$

where each m_i functions as a continuous shrinkage factor since it is less than 1. So, it is easy to note that $\sum_{i=1}^{p} m_i \leq p$, and therefore this model is a shrunken model and shares some desirable properties with shrinkage estimators.

The model selection factor and regression coefficients can be estimated based upon data as follows

$$(\tilde{m},\tilde{\beta}) = \arg\min_{m,\hat{\beta}} E\left[(Y - \sum_{i=1}^{p} m_i\hat{\beta}_i X_i)^2\right] = \arg\min_{\beta} E\left[(Y - \sum_{i=1}^{p} \beta_i X_i)^2\right] \ ,$$

$$\text{subject to } \sum_{i=1}^{p} m_i \le t \text{ ,where } 0 \le m_i \le 1 \tag{6.3}$$

Meanwhile, the generalized ridge regression can be expressed as

$$\hat{\beta}^{RR} = \arg\min_{\beta} E\left[(Y - \sum_{i=1}^{p} \beta_i X_i)^2\right], \text{ subject to } \sum_{i=1}^{p} \lambda_i \beta_i^2 \le t \,. \tag{6.4}$$

The solution of the generalized ridge regression given regularization factors is simply

$$\hat{\beta}^{RR} = (x^T x + \Lambda)^{-1} x^T y \,, \tag{6.5}$$

where x is a $n\times k$ matrix, i.e. $x=(x_1,\ldots,x_n)^T$ where each x_i represents n realization of X_i, i.e. $x_i=(x_{i1},x_{i2},\ldots,x_{ik})^T$, y is a column vector $y=(y_1,\ldots,y_n)^T$, and $\Lambda=\mathrm{diag}(\lambda_1,\ldots,\lambda_p)$.

In view of the simplicity of the generalized ridge regression, we can convert a fuzzy model selection problem into a problem of generalized ridge regression

$$(\tilde{\lambda},\tilde{\beta}) = \arg\min_{\lambda,\beta} \sum_{j=1}^{n} (y_j - \sum_{i=1}^{p} \beta_i x_{ji})^2 \text{ , subject to } \sum_{i=1}^{p} \lambda_i \beta_i^2 \le t \tag{6.6}$$

by the following transformation

$$m_i = \frac{1}{1+\lambda_i} \,, \tag{6.7}$$

where satisfies $0 \le m_i \le 1$ since $\lambda_i \ge 0$.

Minimizing the empirical prediction error over both shrinkage factors and regression coefficients under the quadratic constrains is actually a multi-dimensional global optimization problem. Thus, the Genetic Algorithm with floating-point coding can be applied to search for the optimal solution. To construct the fitness for a candidate solution, chapter 4 defines an effective model dimensionality for shrunken regression models, which then allows extending classical model selection methods like cross-validation and information criterions to evaluate a shrunken regression model.

6.2.2 Parametric weighted least squares

The classical way to estimate regression coefficients in a multiple linear regression model is the ordinary least squares (OLS), and in fact it is also the most widely used method. Under the Gauss-Markov (GM) assumptions, namely,

(i) ε is Gaussian independent of X and with $E(\varepsilon|X)=0$.

(ii) The covariance of ε is constant.

(iii) The regression model is properly specified.

OLS estimator of regression coefficients has been proven to be the best linear unbiased estimator (BLUE). In other words, with these assumptions holding, we can do nothing to improve its efficiency.

However, in reality GM assumptions cannot always be met. For example, non-Gaussian error, outliers, heteroscedasticiy and incomplete set of predictor variable might result in the break-down of these ideal assumptions. Although some existing techniques mentioned earlier might be useful in face of some of the above problems, but none of them can handle all these problems. Meanwhile, since these problems have the same consequence, that is, non-Gaussian effective error, it is usual difficult to distinguish them from each other only based upon observations and apply appropriate technique. Therefore, chapter 5 proposes a new method, namely, parametric weighted least-squares (PWLS), as a uniform framework to remedy these problems and to obtain a more efficient regression model.

A model method for estimating regression coefficients is weighted least squares, which minimizes the weighted sum of squared error

$$\hat{\beta}_{WLS} = \arg\min_{\beta} WSSE = \arg\min_{\beta} \sum_{i=1}^{n} w_i (y_i - x_i \beta)^2 , \tag{6.8}$$

where usually $w_i = |e_i|^{-1} = |y_i - x_i\hat{\beta}|^{-1}$. In PWLS, the weight function is extended to be

$$w_i = |e_i|^{\alpha} = |y_i - x_i \hat{\beta}|^{\alpha}, \tag{6.9}$$

where α is between [-1, 1]. Although such a weigh function is simple, involving only a single hyperparameter, but it can realize two classes of weighting strategies by varying α, that is

(i)Decrease the weights of data points having large residual error, or (ii)Increase the weights of those observations hard to learn.

Noting that asymptotically $WSSE = \sum_{i=1}^{n} w_{\alpha i} e_i^2$ converges to $\sum_{i=1}^{n} |e_i|^{2+\alpha} = \sum_{i=1}^{n} |e_i|^{\gamma}$, it is learned that the corresponding estimation of β is equivalent to the maximum likelihood estimation (MLE) if we assume the error distribution as

$$e \sim f(e \mid \gamma, \mu, \sigma) = \frac{\gamma}{2\sigma \cdot \Gamma[1/\gamma]} \exp\left(-\left| \frac{e-\mu}{\sigma} \right|^{\gamma} \right) , \tag{6.10}$$

which is termed exponential power distribution (EPD). This parametric exponential power distribution can model a wide class of distributions including uniform, Gaussian, Laplace and other sub- and super-Gaussian densities.

Now it is clear that estimating the optimal weight function within this parametric family is equivalent to approximating the error distribution using the parametric EPD family. To this end, in chapter 5 two methods are suggested to accomplish this, that is, maximum likelihood estimator (MLE) and alternatively residual maximum likelihood estimator (REMLE). The idea of ML method is somewhat straightforward, but because the score functions corresponding to each parameter cannot be solved simultaneously, an iterative procedure should be applied to estimate (γ, σ, β) as follows

(1) first set some initial value for γ,

(2) solve for β_j, j=1,...,p, given γ,

(3) solve for σ, given γ and β_j, and finally

(4) solve for γ given β_j and σ.

(5) repeat the above (2)-(4) steps until the solution converges.

Because the ML method is a little complex and ML estimator is usually biased, REML is proposed as an alternative. Residual Maximum Likelihood (REML) is very similar to ML except that it applies ML method after transforming the original data by some properly chosen matrix S, i.e. Sy. The likelihood function for the transformed data is obtained simply by replace x and y by Sx and Sy in the likelihood function for the original data,

$$f(Sy \mid \gamma, \sigma, \beta) = \prod_{i} \frac{\gamma}{2\sigma \cdot \Gamma[1/\gamma]} \exp\left(- \left| \frac{(Sy)_i - (Sx)_i \beta}{\sigma} \right|^{\gamma} \right). \tag{6.11}$$

where S can be so chosen that satisfies the following conditions

(i) $Var(Sy)$ is positive definite.

(ii) $E(Sy)=0$,i.e. $Sx=0$.

(iii) $Rank(S)=n-p$, that is, S has $n-p$ linearly independent rows.

A suitable matrix S was suggested by Patterson and Thompson [1971] as

$$S = I - x(x^T x)^{-1} x^T . \tag{6.12}$$

The rest procedure is similar to that of ML method. With REML, the estimation of β is separated from that of γ, therefore simplifying the problem a lot.

In order to avoid the downside of the performance of PWLS and reduce the probability of error when GM assumptions hold, a likelihood ratio test is also designed to conduct significance test. With this test, PWLS will be applied in the situations where the overall error distribution significantly departs from normality; otherwise, we should stick to the normal assumption and employ OLS method.

6.2.3 Generalized fuzzy model selection scheme

As we note, PWLS differs from OLS in that PWLS extends the error distribution from Gaussian to a parametric family of exponential power distributions and the hyperparameter has to be estimated from observations. Meanwhile, fuzzy variable selection is basically based upon constrained least squares. Therefore, in order to combine the procedures of fuzzy model selection and parametric weighted least squares, we actually need to extend fuzzy model selection to regression models where the error might not be normal and is not known with certainty.

Before we proceed, let's first consider a simpler case, where the error is not Guassian but its type has been already known. In such a case, the Maximum Likelihood Estimator is not equal to OLS estimator and accordingly the fuzzy variable selection method in chapter

4 can be applied directly, but it is not hard to generalize the fuzzy model selection scheme to situations with non-normal error if we notice the following facts.

First, as we've already pointed out, fuzzy model selection actually solves a problem of constrained least-squares with constrains on parameters as in equation (6.3). It is natural to extend this idea to maximum likelihood estimation, that is,

$$\hat{\beta} = \arg\max_{\beta} l(\beta) \ s.t. \ c(\beta) \geq 0, \qquad (6.13)$$

where $l(\beta)$ is the likelihood or log-likelihood function given a data set

$$l(\beta) = \sum_{i=1}^{n} \log f(y_i, x_i, \theta(\beta)). \qquad (6.14)$$

In the literature, this estimator is sometimes called constrained maximum likelihood.

However, usually the likelihood function does not have a simple form like least squares, and thus we still have difficulty solving this constrained optimization problem and also defining effective model dimensionality as in fuzzy model selection method. Fortunately, this maximum likelihood estimation can be converted into an iteratively reweighted least squares (IRLS) procedure. For example, in case of the linear regression with independent observations

$$\begin{aligned}\hat{\beta} &= \arg\max_{\beta} l(\beta) = \arg\max_{\beta} \sum_{i=1}^{n} \log f(y_i, x_i, \theta(\beta)) \\ &= \arg\min_{\beta} \sum_{i=1}^{n} \left(c - \log f(y_i, x_i, \theta(\beta))\right) \\ &= \arg\min_{\beta} \sum_{i=1}^{n} \left(\frac{\left(c - \log f(y_i, x_i, \theta(\beta))\right)}{(y_i - x_i^T \beta)^2} (y_i - x_i^T \beta)^2 \right) \qquad (6.15) \\ &= \arg\min_{\beta} \sum_{i=1}^{n} w_i (y_i - x_i^T \beta)^2,\end{aligned}$$

where $w_i = \dfrac{\left(c - \log f(y_i, x_i, \theta(\beta))\right)}{(y_i - x_i^T \beta)^2}$ and c is so chosen as to make sure that w_i is positive.

Therefore, if we let

$$w_i^{(t+1)} = \frac{\left(c - \log f(y_i, x_i, \theta(\beta^{(t)}))\right)}{(y_i - x_i^T \beta^{(t)})^2}, \qquad (6.16)$$

ML estimation can be implemented via an IRLS procedure, but the rate of convergence might not be satisfactory.

For general maximum likelihood estimation, the likelihood function can be maximized by zeroing its first derivatives with regard to β, i.e.,

$$\frac{\partial l(\beta)}{\partial \beta} = 0, \tag{6.17}$$

which can be solved for β by applying the iterative Newton-Raphson algorithm as follows

$$\beta_{t+1} = \beta_t - \left[\nabla^2 l(\beta_t)\right]^{-1} \nabla l(\beta_t), \tag{6.18}$$

where $\nabla^2 l(\beta_t)$ is the Hessian matrix and $\nabla l(\beta_t)$ is Jacobian vector evaluated at the previous value of β.

In practice, a popular approximation of the Newton-Raphson algorithm is the Fisher's scoring algorithm, which may be easier to apply in some cases. With Fisher's scoring technique and some other approximations, Green [1984] shows how an iterative reweighted least squares algorithm can be employed to implement the Newton-Raphson method for an iterative solution to the likelihood equations (6.17). Its applications can be found in generalized linear models [McCullagh and Nelder, 1983] and generalized additive models [Hastie and Tibshirani, 1990].

With this conversion, the constrained maximum likelihood in the equation (6.17) can be easily solved by a constrained iteratively reweighted least squares procedure. In each step, given weights and shrinkage factors a constrained weighted least squares problem needs to be solved, i.e.

$$\hat{\beta}_{CWLS} = \arg\min_{\beta} \sum_{j=1}^{n} w_j (y_j - \sum_{i=1}^{p} \beta_i x_{ji})^2 \text{, subject to } \sum_{i=1}^{p} \lambda_i \beta_i^2 \leq t, \tag{6.19}$$

which looks very similar to the constrained least squares in the equation (6.4) except for weights w_j. Following straightforward Lagrange multiplier approach, a constrained weighted least squares problem can be easily converted to a penalized weighted least squares problem. By some transformation on the variables, i.e. $y_i' = \sqrt{w_i}\, y_i$ and

$x_i' = \sqrt{w_i} x_i$, with the aid of the solution of the generalized ridge regression it is easy to derive that

$$\hat{\beta}_{CWLS} = (x^T Wx + \Lambda)^{-1} x^T Wy, \tag{6.20}$$

which incorporates both model selection and data influence controlling information.

Now, generalized fuzzy model selection scheme is actually represented by a constrained iteratively reweighted least squares procedure. If we note that

$$\hat{\beta}_{CWLS} = (x^T Wx + \Lambda)^{-1} x^T Wy = (x^T Wx + \Lambda)^{-1} (x^T Wx)(x^T Wx)^{-1} x^T Wy = M\hat{\beta}_{WLS}, \tag{6.21}$$

where $M = (x^T Wx + \Lambda)^{-1} x^T Wx$ and $\hat{\beta}_{WLS} = (x^T Wx)^{-1} x^T Wy$. Just as chapter 4, let's define the generalized effective model dimensionality as

$$D_{eff} = Trace(M) = Trace[(x^T Wx + \Lambda)^{-1} x^T Wx], \tag{6.22}$$

which is similar to what was suggested by Tibshirani [1997].

It is easy to see that in constrained least squares $W=I$ and thus

$$D_{eff} = Trace[(x^T x + \Lambda)^{-1} x^T x], \tag{6.23}$$

which is the same as in chapter 4.

Another modification we need to make to generalize fuzzy variable selection method is concerning the empirical likelihood, since the error distribution is not assumed to be normal. However, it should be easy to replace normal distribution with other distributions and obtain corresponding maximum likelihood. With model likelihood and effective model dimensionality ready, a model can be assessed using most classical model selection methods.

With all the above modifications, now we are able to extend fuzzy variable selection to any regression model with non-Gaussian error. Once the generalized fuzzy variable selection is ready, it seems trivial to apply it to PWLS.

6.2.4 PWLS with a fuzzy set of predictor variables

PWLS in chapter 5 is developed assuming that the predictor variables are totally valid, or equivalently supposing m=[1,...,1], although perhaps incomplete, and therefore each predictor variable makes its full contribution. In this section, we will generalize PWLS to the cases where the set of predictor variables is a fuzzy set. For this purpose, several modifications have to be made to the PWLS.

Just as in chapter 5, here we also assume the model structure has already been known, although in fact not, or in other words a fixed m vector is already given. In order to take advantage of iterative reweighted generalized ridge regression, it is convenient to convert membership grade matrix M=diag(m_1,...,m_p) in the equation (6.4) to individual regularization parameters matrix $\Lambda = (M^{-1} - I)x^T Wx$, or even express the model structure in terms of Λ directly.

If we assume the error follows an exponential power distribution in the equation (6.10), without constraints the ML estimation is easy to be obtained by a procedure of iterative reweighted least squares with setting $w_i = |e_i|^{\gamma-2}$. When the model structure is specified in the form of a membership grade vector m or equivalently an individual regularization parameter vector λ, we are facing a problem of constrained maximum likelihood estimation, which, as we discussed earlier, can be also solved by a constrained iterative reweighted least squares procedure with weights $w_i = |e_i|^{\gamma-2}$, i.e.

$$\hat{\beta}_{CWLS} = (x^T Wx + \Lambda)^{-1} x^T Wy . \tag{6.24}$$

It is straightforward that in order to generalize PWLS using ML to the cases with a fuzzy set of predictor variables the only modification is to replace $\hat{\beta}_{WLS} = (x^T Wx)^{-1} x^T Wy$ in the step (2) in the iterative (γ, σ, β)-estimating procedure by $\hat{\beta}_{CWLS}$ in the above equation.

When we want to apply REML in estimating γ in the generalized PWLS, some problems arise. Although $S = I - x(x^T x)^{-1} x^T$ still meets the requirements for S in the section 6.2.2, it does not reflect the change of model structure, which seems inappropriate. As we know, the major purpose of REML in PWLS is to decouple the estimation of γ and β by choosing such an appropriate matrix that S does not depend on data y and $Sx=0$. Unfortunately, for the current case, i.e. constrained weighted least squares, it is not easy to find such S. Therefore, only maximum likelihood method will be used for the current case.

6.2.5 Two-stage optimization algorithm

By now, we have already generalized fuzzy variable selection and parametric weighted least squares and at this point we are ready to combine these two methods.

Before we proceed, let's define our problem explicitly. Suppose we have a multiple regression model

$$Y = \sum_{i=1}^{p} \beta_i X_i + \varepsilon , \tag{6.25}$$

where Y is the response variable, X_i's are predictor variables, and β_i's are regression coefficients to be estimated, and ε is the error term, whose distribution is symmetric but unknown.

Furthermore, a group of data, (x_i, y_i) for i=1,…,n, is collected, based upon which we want to estimate a regression model with as smaller mean prediction error as possible. For this purpose, we need to reduce both model structural uncertainty by model selection and model parameter uncertainty by iteratively weighted least squares, as we discussed earlier. Combining fuzzy model selection and parametric weighted least squares might serve this purpose. The fuzzy model selection procedure will deliver us the model structure in the form of a membership grade vector m or equivalently a vector of regularization parameters λ, while the parametric weighted least squares procedure produces the estimation of error in term of γ as well as regression coefficients β. Thus, the overall problem can be regarded as an optimization problem with regard to m, γ and β, i.e.

$$(\tilde{m},,\tilde{\gamma},\tilde{\beta}) = \arg\min_{m,\beta,\gamma} E\left[(Y - \sum_{i=1}^{p} m_i \beta_i X_i)^2\right], \tag{6.26}$$

where the optimizations of m, γ and β are coupled together. In reality, with finite samples we have no idea of the expected prediction error given a regression model and thus some empirical model evaluation methods, for example those in chapter 4, should be utilized instead.

In fact, in order to combine fuzzy model selection and parametric weighted least squares, we may have two different ways to do so.

One is to plug the PWLS procedure into the fuzzy variable selection procedure, that is, each time when evaluating fitness for a candidate solution in the Genetic Algorithm, first perform PWLS given a vector of regularization parameters as in the generalized PWLS to estimate $\gamma$and then calculate maximum likelihood or cross-validation mean squared errors. Thus, on the whole we need to go through only one optimization procedure and we may call this method one-stage optimization procedure.

An alternative way is to optimize model selection and error distribution estimation separately and alternately. The overall iterative procedure actually includes two stages, fuzzy variable selection given error structure and parametric weighted least squares given model structure. The two steps alternate until reaching convergence. We might call this method two-stage optimization procedure.

Comparatively, the one-stage method is more computationally expensive, because in searching for an optimal model structure using Genetic Algorithm, the one-stage method needs to conduct PWLS for each candidate solution in every generation, while the two-stage method needs only to do weighted least squares. In addition, we suspect that the two-stage optimization is more robust and converges faster. This is because for the one-stage method the estimated error distributions, or γ, for each candidate solution may be quite different, which probably slows down the convergence. It seems good to fix the error structure during variable selection.

Accordingly, in this chapter we will favor the two-stage optimization procedure, and let's formulate it clearly in the following.

(1) Assume the error is Gaussian, i.e. $\gamma=2$.

(2) Given the error distribution in terms of γ, conduct the generalized fuzzy variable selection, which delivers a vector of regularization parameters, λ.

(3) Given the model structure in terms of λ, perform the generalized parametric weighted least squares, which produces a new estimation of the error distribution.

(4) Repeat the above steps (2)-(3) until convergence.

6.3 Uncertainty evaluation

In the above sections, we were concentrating on working out some method that helps reducing both model structural uncertainty and parameter uncertainty. For this purpose, fuzzy variable selection is combined together with parametric weighted least squares. From now on, we will turn to the quantification of model uncertainty and extend point estimate to interval estimate.

The expected generalization error or prediction error

$$\begin{aligned} E_D[(y-\hat{y})^2] &= E_D[(y-E_D(\hat{y})+E_D(\hat{y})-\hat{y})^2] \\ &= (y-E_D(\hat{y}))^2+E[(\hat{y}-E_D(\hat{y}))^2] \end{aligned} \tag{6.27}$$

where $E_D(\cdot)$ is evaluated with regard to observations. The bias term $(y-E_D(\hat{y}))^2$ is usually dominated by variance $E_D[(\hat{y}-E_D(\hat{y}))^2]$, and thus, in this chapter we will consider $\hat{y}$ to be unbiased, that is,

$$E_D[(y-\hat{y})^2] \approx E_D[(\hat{y}-E_D(\hat{y}))^2]. \tag{6.28}$$

As we mentioned earlier, in effect a model is made up of two parts, i.e. model structure and model parameter. Following the notations in Draper [1995], we formulate a model as $M=(S, \theta)$. In this chapter, both model structure S and model parameter θ are estimated based upon the same data set D. Thus, different data set D results in different model estimation $M=(S, \theta)$. In other words, random fluctuation in data leads to uncertainty in both model structure selection and parameter estimation. $E_D(\cdot)$ is in fact equivalent to $E_{S\theta}(\cdot)$, which is computed with regard to all possible model structures S and model parameters θ, i.e. $E_D(\cdot) \equiv E_{S\theta}(\cdot)$.

By iterated expectation, we further obtain

$$\begin{aligned} E_D[(\hat{y}-E_D(\hat{y}))^2] &= E_S\{E[(\hat{y}-E_{\theta|S}(\hat{y}\,|\,S)+E_{\theta|S}(\hat{y}\,|\,S)-E_{S\theta}(\hat{y}))^2\,|\,S]\} \\ &= E_S\{E_{\theta|S}[(\hat{y}-E_{\theta|S}(\hat{y}\,|\,S))^2\,|\,S]+(E_{\theta|S}(\hat{y}\,|\,S)-E_{S\theta}(\hat{y}))^2\} \\ &= E_S\{E_{\theta|S}[(\hat{y}-E_{\theta|S}(\hat{y}\,|\,S))^2\,|\,S]\}+E_S\{(E_{\theta|S}(\hat{y}\,|\,S)-E_{S\theta}(\hat{y}))^2\} \\ &= E_S[Var(\hat{y}\,|\,S)]+Var_S[E_{\theta|S}(\hat{y}\,|\,S)] \end{aligned} \qquad (6.29)$$

where the first term represents the mean variance given model structure and the second terms refers to the variance between model structures. By now, it is clear that the overall uncertainty consists of two components, model structural uncertainty and model parameter uncertainty, corresponding to the two terms in the RHS of the above equation. It is noteworthy that the uncertainty obtained in equation (6.29) is the uncertainty of a model construction procedure or a learning algorithm, rather than that of a specific model.

Usually, we just take into account $Var(\hat{y}\,|\,S)$ and on the other side ignore the model structural uncertainty, thereby overstating the precision of an estimated model. Draper [1995] tries to attack this problem in a Bayesian framework, where the predictive distribution is formed by using as weights the posterior model probability *p(M|D)*, that is,

$$\begin{aligned} p(y\,|\,D) &= \int p(y\,|\,D,M)p(M\,|\,D)dM = \iint p(y\,|\,D,S,\theta)p(S,\theta\,|\,D)dSd\theta \\ &= \iint p(y\,|\,D,S,\theta)p(\theta\,|\,D,S)p(S\,|\,D)dSd\theta, \end{aligned} \qquad (6.30)$$

which is the same as Bayesian Model Averaging (BMA) [Hoeting, 1999].

Here, model probability *p(M)* can be interpreted in a similar way to that for a random variable. In the probabilistic world, just like a random variable a true model is assumed to never appear exactly as it is. If we can define the distance of two models in the model space somehow, for example, using some kind of norm, the model probability is actually converted to the probability of random variables. As such, the prior model probability distribution expresses our prior knowledge about the true model probability distribution in the model space.

However, usually the space of all models is "too big", and at the same time a single structural choice S^* may be too small to be well calibrated, and thus Draper [1995] proposed an intermediate position based on model expansion [e.g. see Box, 1980], i.e. beginning with a single structural choice S^* and then expanding it in some directions suggested by the data analytic search that resulted in S^*.

Then, Draper [1995] further utilized discrete model expansion to approximate a continuous expansion. If we suppose the set of alternative structures as $\{S_1, \ldots, S_m\}$, then equation (6.30) can be rewritten as

$$p(y \mid D) = \sum_{i=1}^{m} \int p(y \mid D, S, \theta) p(\theta \mid D, S_i) p(S_i \mid D) d\theta = \sum_{i=1}^{m} p(y \mid D, S_i) p(S_i \mid D) \qquad (6.31)$$

Regarding the choice of alternative structures $\{S_1, \ldots, S_m\}$, Draper [1995] also makes some general comment on it, such as

(i) the set of alternative structures should not be small,

(ii) S_i should have high posterior probability $p(S_i|D)$, and

(ii) the predictive consequence $p(y|D,S_i)$had better to be substantially different from that of the single structural choice S^*.

Applying the Laplace approximation [e.g. see Tierney and Kadane, 1986], Draper [1995] obtained

$$\begin{aligned} \ln p(D \mid S) &= \ln \int p(D \mid S, \theta) p(\theta \mid S) d\theta \\ &\approx \frac{1}{2} k \ln(2\pi) - \frac{1}{2} k \ln(n) + \ln p(D \mid S, \hat{\theta}) + O(1), \end{aligned} \qquad (6.32)$$

where $k=dim(\theta)$, n is the sample size, and $\hat{\theta}$ is the mode of the posterior probability $p(\theta|D,S)$. The above approximation actually forms the basis of the Bayesian Information Criterion (BIC) for model selection [see Schwarz, 1978 and Chow, 1981].

Supposing $p(y|D,S_i)$ has mean μ_i and variance σ_i^2 and also the posterior probability $p(S_i|D)=\pi_i$, Draper [1995] derived

$$\begin{aligned} Var(y \mid D) &= E_S[Var(y \mid D, S)] + Var_S[E(y \mid D, S)] \\ &= \sum_{i=1}^{m} \pi_i \sigma_i^2 + \sum_{i=1}^{m} \pi_i (\mu_i - \mu)^2. \end{aligned} \qquad (6.33)$$

In fact, noting that in the derivation of Draper [1995], if we assume $n>>2\pi$or k is equal for all S, then we obtain

$$\ln p(S \mid D) \approx \ln p(D \mid S, \hat{\theta}) - \frac{1}{2} k \ln(n) + C = -\frac{1}{2} BIC + C\,, \tag{6.34}$$

where C is a constant and BIC is exactly the same as defined in Schwarz [1978], i.e.

$$BIC = -2 \ln p(D \mid S, \hat{\theta}) + k \ln(n)\,. \tag{6.35}$$

In view of the constraint that $\sum_{i=1}^{m} \pi_i = 1$, we have

$$\pi_i = \frac{\exp(-BIC_i / 2)}{\sum_{j=1}^{m} \exp(-BIC_j / 2)}\,, \tag{6.36}$$

which is very simple and directly related to what we have done in fuzzy model selection.

As we noticed, Draper [1995] incorporates model structural uncertainty straightforwardly within Bayesian Model Averaging (BMA). Buckland, Burnham and Augustin [1997] think BMA suffers from the sensitivity to the choice of priors and two many possible models, and they seek to average models using some weights

$$\hat{y} = \sum_{i=1}^{m} w_i \hat{y}_i\,, \tag{6.37}$$

where $\hat{y}_i$ is the estimate of y under model M_i=(S_i, θ_i) and w_i are respective weights.

Thus,

$$Var(\hat{y}) = \sum_i w_i^2 Var(\hat{y}_i) + \sum_i \sum_{j \neq i} w_i w_j Cov(\hat{y}_i, \hat{y}_i) \leq \left(\sum_{i=1}^{m} w_i \sqrt{Var(\hat{y}_i)} \right)^2, \tag{6.38}$$

where the equality holds if we conservatively assume that alternative models are perfectly correlated. Actually, it should be true that the covariance among alternative models is high since each model is fitted to the same data set.

If the independence of estimators is realized somehow, for example, by Bootstrap or Cross-Validation, the variance of the weighted model is

$$Var(\hat{y}) = \sum_i w_i^2 Var(\hat{y}_i)\,. \tag{6.39}$$

In the above derivations, the weights w_i are assumed to be known, whereas in practice they have to be estimated as well. Buckland, Burnham and Augustin [1997] prefer to use information criteria of the form,

$$I = -2 \ln(L) + q\,, \tag{6.40}$$

where L is the likelihood function, evaluated by plugging in maximum likelihood estimate of parameters, and q is a penalty term which is usually a function of the number of parameters. Such information criteria include Akaike's Information Criterion (AIC) [Akaike, 1973] and Schwarz's BIC.

Finally, Buckland, Burnham and Augustin [1997] proposes a plausible choice for weight w_i as

$$w_i = \frac{\exp(-I_i / 2)}{\sum_{j=1}^{m} \exp(-I_j / 2)}. \tag{6.41}$$

If we use BIC in place of I in the above equation, what we obtain is exactly the same as equation (6.36).

Both above methods have their own strengths and weaknesses. For example, the good thing about the second method is to apply weights, which renders the problem simple, but the assumptions of both perfect correlation and independence are not satisfactory. In this chapter, we will propose a new method intended to combine the favorable advantages of these two approaches. This new method is supposed to have the following merits

(i) Using Weighted Model Averaging rather than Bayesian Model Averaging, but the weights are not restricted to constructing using information criteria but a wider class

(ii) Interpreting normalized weights as probability and thus the formula (6.29) can be applied to evaluate model uncertainty.

(iii) Furthermore, the variance conditional on model structure in equation (6.29) will be replaced by expected prediction error, which may includes bias and in practice estimated via Cross-Validation.

In the following, we will describe the method fitted into our situation in detail.

As we know, evaluation of model selection uncertainty must be based upon multiple alternative models. So, the first thing we need to do is to choose an appropriate set of alternative structures $\{S_1, \ldots, S_m\}$, neither too big nor too small. In the current case, model structure is represented by a membership grade vector m or a shrinkage parameter vector λ, and thus we can express the model space in R^P, which is a continuous p-dimensional Euclidean vector space. This model space seems too big to be operable, and following Draper [1995] we might apply discrete approximation. Actually, we have a good candidate at hand, if we note that in the optimization procedure of Genetic Algorithm a group of candidate model structures are generated. This group of candidate solutions generally has high posterior probability since they fit to the data not badly and remain diverse at the same time, thereby meeting the comment by Draper [1995] on the choice of alternative structures. Another advantage of choosing this group of alternative structures is that uncertainty evaluation adds very little extra work, which will become clear later on.

In chapter 4, four methods are used to conduct model evaluation, namely, AIC, BIC, 10-fold Cross-Validation as well as Generalized Cross-Validation (GCV). For AIC and BIC, formula (6.41) can be directly applied to construct weights. For Cross-Validation method, we want to find a function, which map Cross-Validation errors (CVE) to weights, such that the weights are nonnegative and the larger the Cross-Validation error the smaller the weight. Meanwhile, to avoid dominance of some model, which happens to have very small *CVE*, we apply the softmax weights based on generalization errors

$$w_i = \exp(-CVE). \qquad (6.42)$$

As we already mentioned, the $Var(\hat{y} \mid S)$ in equation (6.29) is replaced by expected prediction error, which is estimated by Cross-Validation in practice, and thus

$$Var(\hat{y} \mid S) = CVE\,. \qquad (6.43)$$

Finally, we define posterior model probability as

$$\pi_i = p(S_i \mid D) = \frac{w_i}{\sum_{i=1}^{m} w_i}\,. \qquad (6.44)$$

Consequently, equation (6.29) becomes

$$\begin{aligned} E_D[(\hat{y} - E_D(\hat{y}))^2] &= E_S[Var(\hat{y} \mid S)] + Var_S[E_{\theta|S}(\hat{y} \mid S)] \\ &\approx \sum_{i=1}^{m} \pi_i CVE_i + \sum_{i=1}^{m} \pi_i \left(E_{\theta|S}(\hat{y} \mid S_i) - \sum_{j=1}^{m} \pi_j E_{\theta|S}(\hat{y} \mid S_j) \right)^2 \end{aligned} \qquad (6.45)$$

If the sample size is equal to *n*, an average value over samples should be used to estimate the model uncertainty. Again it is worth pointing out that the variance in equation (6.45) refers to the uncertainty our constrained parametric weighted least square method rather than that of an individual model. However, it does represent the uncertainty of the model produced by this model construction procedure. If we want to estimate the uncertainty of a specific model, we can simply replace the first term in equation (6.45) with individual predictive error like CVE_i.

Alternatively, we might want to evaluate the variance between alternative model structures, i.e. the second term in the RHS of equation (6.45) using some other resampling methods by the mechanism of perturbation, such as Cross-Validation and Bootstrap. For example, Burnham and Anderson [2002] explored Bootstrap to evaluate model selection probability. However, in the current case, since the original data set is not too big resampling with replacement is not effective in general.

6.4 Numerical simulation study

In the above, we have developed a new variable selection method for PWLS as well as a model uncertainty quantification method. In this section, we will demonstrate how they work in the context of PWLS by numerical study.

In order to show the performance of the new fuzzy variable selection approach, we will first compare it with classical or crisp variable selection methods in PWLS by simulation.

In this numerical study, the standard Fourier series fitting is used, where cosine series are applied as predictor variables to fit a data set generated by a smooth function plus some disturbance, i.e.

$$y = f(x) + \varepsilon, \tag{6.46}$$

where ε can be any arbitrary but symmetric distribution.

The set of predictor variables we utilized in our simulations is $\{\cos(k\pi x)\text{: } k < n\}$, and the smooth function is

$$f(x)=sinc(3x)=sin(3x)/(3x),\ x\in[-1,1]. \tag{6.47}$$

Because in PWLS all the problems lead to the same consequences, that is, non-normal overall error and PWLS has shown effective under all situations, in the current study we only need to test its performance in face of non-Gaussian errors. To this end, we generated errors of Laplace or double exponential distribution.

Since in the current simulation study we assume the true model is already known, therefore we can define a global mean prediction error to measure how good an estimated regression model is. Conceptually, the global mean prediction error is defined as

$$GMPE = \frac{1}{2L}\int_{-L}^{L}(f(t) - \hat{y}(t))^2 dt, \tag{6.48}$$

where $\hat{y}(t)$ is the estimated model, but in practice, it can be approximated by

$$GMPE \approx \frac{1}{m}\sum_{i=1}^{m}(f(t_i) - \hat{y}(t_i))^2, \tag{6.49}$$

where t_i is typically evenly distributed in [$-L$, L] and m is large enough.

To evaluate the performance of new variable selection method, we used crisp model selection as a benchmark, and compared their performance by defining a ratio as

$$R_{C/F} = \frac{GMPE_C}{GMPE_F}. \tag{6.50}$$

The average ratio is defined as its geometric mean

$$\overline{R}_{C/F} = \exp\left(\frac{1}{N}\sum_{i=1}^{N}\log(R_{C/F})_i\right), \tag{6.51}$$

where N is total number of simulations. The average ratio is so defined that it helps reduce the influence of some unusual cases.

In order to compare the performance of the new method with that of classical crisp variable selection under the various situations with non-Gaussian noise, three classes of experiments were carried out, including heteroskedastic Gaussian noise, double exponential noise, and Gaussian noises with outliers. For the heteroskedastic Gaussian noises, the variance is modeled in two different ways, namely log-normal distribution and binomial distribution. In the case of outliers, two extremely large errors are added to the normal noises with probability 0.1, respectively.

In all the experiments, a sinc function is used as the target function and the sample size is assumed to be 20. The simulations are repeated for 100 times and the average results are shown in the following Table 6.1.

Table 6.1 Simulation results for model selection under unknown error distribution

Non-Gaussian noise		GMPE of classical variable selection	GMPE of fuzzy variable selection	Average $R_{C/F}$
Hetero-skedasticity	Log-Normal	0.16	0.122	1.35
	MoG	0.054	0.047	1.19
Double Exponential		0.103	0.072	1.50
Outliers		0.065	0.028	2.41

As expected, from the simulation results we note that in all these contents the mean generalization error of fuzzy variable selection is significantly smaller than that of classical variable selection, especially with the presence of outliers. This means compared to the classical approaches our new method is more robust under various situations.

6.5 Conclusion

In this chapter, we intend to solve two problems, namely variable selection in the context of parametric weighted least-squares and model uncertainty evaluation. A new method called constrained parametric weighted least-squares is developed by generalizing the newly developed fuzzy variable selection method. Another new approach is also proposed to evaluated model uncertainty, which is able to consider both model structure and parameter uncertainty, thereby avoiding underestimating the model uncertainty. Our analysis and simulation results show that the new method is superior to the classical crisp variable selection methods in the presence of non-normal errors and outliers.

Chapter 7

Adaptive Fuzzy Mixture of Local ICA Models

7.1 Introduction

Models are very useful in explaining systems and predicting their future behaviors. Usually, for a certain system there are many different competing models. In such a situation, a question arises naturally, that is, how to improve model performance with all the available information. One strategy is to select a single best model among the group of competing models and an alternative way is to combine multiple competing models. In both theoretical and empirical researches, it is shown that model combination can lead to better models than model selection. However, how to aggregate information contained in candidate models and new observations in an efficient way is still an open research problem. To this end, we proposed a feature-based model combination method, which first extracts statistically independent features from a group of candidate models by Principal Component Analysis (PCA) or Independent Component Analysis (ICA) and then aggregate features into a new composite model based on data through regression. Our simulation studies show that this method outperforms model selection and other existing model combination methods.

However, there are some weaknesses inherent in this approach that limit its application. Two major weaknesses among them are the employment of a single global linear model and the application of global linear PCA or ICA.

In the above feature-based model combination method, a global linear model is used to describe the system. However, in reality for a complicated system its true model will be nonlinear and highly complex. In addition, in real physical systems phase transition is a ubiquitous phenomenon. This is because the mechanisms underlying a system vary from a small region to another resulting in different local features and as a consequence a single global model is far from enough to capture them. Therefore, it is far from justified to use a single global linear model to describe a complex system over the entire domain. In many cases, it is more appropriate to use local models based upon operating regimes.

Furthermore, the global linear model directly results from the application of the global linear ICA for feature extraction. Although the global linear ICA has been applied to many areas successfully, certainly its efficiency is restricted by its inherent weaknesses. The limitations of standard ICA have been investigated by Karunen and Malaroiu [1999] and other authors, and at least two of them are relevant to the present case.

First, the standard PCA and ICA assume that the data x are linear superposition of independent components s, i.e.

$$x(t)=Ws(t) \tag{7.1}$$

where $x(t)=[\, x_1(t),\ldots, x_n(t)]^T$, $s(t)=[s_1(t),\ldots,s_m(t)]^T$, and W is the mixing matrix.

However, it is not natural to assume this linearity, even though in many cases linear ICA delivers meaningful results. For general nonlinear data structures, it can provide only a crude approximation and cannot describe nonlinear characteristics adequately. In view of this drawback, during the recent years researchers attempt to generalize linear ICA to nonlinear ICA,

$$x(t)=f(s(t)), \tag{7.2}$$

where $f(\cdot)$: $R^m \rightarrow R^n$ can be an arbitrary nonlinear mixing function.

Unfortunately, nonlinear ICA is much more difficult than linear ICA because the solution of nonlinear ICA problem is usually highly non-unique [see e.g. Hyvärinen and Pajunen, 1999] fundamentally because any functions of independent components remain independent and solving a nonlinear ICA problem is computationally rather demanding [Karunen and Malaroiu, 1999b]. Thus, by now only limited success of nonlinear ICA has been seen.

Second, the standard ICA tries to describe all the data using a single group of independent component called global features. This means the mixing matrix W in equation (7.1) and the corresponding separating matrix are assumed to be the same over the entire region. However, usually real systems have varying characteristics and thus varying mixing matrices in qualitatively different domains of the entire region, which calls for using different local features in each domain for efficient representation.

In this chapter, we will propose mixture of local PCA or ICA models, which helps to overcome the above drawbacks. Local models are built to approximate the complex system within operating regimes and then are combined by smooth interpolation into a complete global model. The split of the input space enable us to characterize the nonlinearity of the system although in a somewhat coarse manner and local component analysis produces different sets of local features, which lead to different local models.

The outline of this chapter is as follows: in section 7.2, at first the idea of mixture of local models is proposed and justified from different angles; an adaptive fuzzy parametric clustering algorithm is presented to split the entire input space into sub-domains; and then local ICA is put forward to extract local features, which constitute local models; finally, a method will be proposed to piece local models together into a mixture model. In section 7.3, a three-stage optimization algorithm will be applied to implement the procedure. Both an artificial example and a real case study will be presented in section 7.4 to demonstrate the performance of this new approach. The last section will summarize this chapter.

7.2 Combining local models

7.2.1 Mixture of local models

In this section, we will introduce a mixture of local models as the generative model, the underlying process that generates the observations. Local models only valid in different operating regimes are built to approximate the complex system locally and then are combined by smooth interpolation into a complete global model. In the following, we will argue why local models can help overcome the problems the previous method fails to solve, namely nonlinearity and phase transition.

7.2.1.1 Local models

In general, it is usually preferred to build a single global model to describe a system over its entire input space. However, in reality a model might not be able to cover the full range of the input space with limited complexity because of the need to describe the interactions between a large number of phenomena that appear globally, and rather a well-defined model is only appropriate over a certain prescribed subspace, namely its operating regime. For example, a model, which is fitted quite well to the data in a region, may end up with poor performance when extrapolated to other regions, which is perhaps, for instance, because some assumption underlying a model can only be met in a certain range of inputs. It is a usual case that for a complicated system the true model is nonlinear and highly complex and a global linear model is far from adequate.

It is also conceivable that with parameters varying a real-world system may undergo different phases, which are governed by different underlying laws. This phenomenon is regarded as phase transition. Meanwhile, from the angle of features, the system is governed by different sets of features over different domains or the same set of features but with non-constant influences. Both cases result in varied local patterns. Correspondingly, it leads to the variation of model structures, that is, different models over different regions. Thus, in the presence of phase transition, it is difficult to incorporate the properties of distinct phases in a single global model.

Therefore, a single global linear model cannot describe a nonlinear system adequately or capture all the local features. To deal with this nonlinearity and locality, it might be possible to find out some missing hidden variables to characterize nonlinearity or phase transition phenomenon and come up with a comprehensive complex global model. Such variables are called Arrhenius-type terms by Johansen and Foss [1997]. However, it is usually hard to find such extra hidden variables, not only because it requires increase knowledge concerning the system under investigation, but also because even if we do, we may end up with an overly complicated global or even intractable model. Following the philosophy of divide-and-conquer, an alternative simpler way to capture the locality is to divide the whole input space into several small regions and perform local analysis in each local region, for example, classification and regression tree (CART) and hierarchical mixture of experts (HME). Local analysis leads to simple local models, which try to characterize a complicated physical system over a certain regime called operating regime, and then local models are combined into a global composite model. In contrast to a global model that is valid in the full range of the input space, a local model is valid only in a predefined operating region smaller than the input space. Then, local models, which, Compared to global modeling, this local modeling can be considerably simpler because there may be a smaller number of phenomena are relevant and their interactions are simpler [Johansen and Foss, 1997]. Therefore, this divide-and-conquer principle simplifies the modeling problem by transforming the task of modeling a complex system into simpler modeling processes whose results can be finally combined relatively easily to yield a satisfactory model.

As usual, in local learning local models are constructed as linear functions of local features. Once local models are ready, they are pieced together somehow to form a global linear mixture model [McLachlan and Basford, 1988]. Although simpler, this mixture model improves the model accuracy because it reduces the model bias by specifying model more properly. By examining bias/variance tradeoff for local and global learning, Murray-Smith and Johansen [1995] show that local learning can be viewed as a simple form of regularization and produce models with higher accuracy and greater robustness than global learning methods.

However, as noted by Jordan and Jacobs [1994], divide-and-conquer tends to increae the variance with hard partition, and a remedy to it is to employ soft split of the input space. A simple version of soft partition is to overlap the operating regimes of local models somehow, which helps smoothen the switching of local models and thereby reduce variance.

It is noteworthy that local models here are different from those in local modeling [Fan, 1995] where a parametric function is fitted to the neighborhood around a query point x, locally weighted regression [Cleveland et al., 1988] where the weights in weighted least squared (WLS) depend on the distance from a data point to the query point x, and mixtures of local experts [Jacobs et al, 1991] where local experts are fitted to all data but not equally well in some local regions. A common drawback of these local learning methods is the complete lack of interpretability of the resultant models.

7.2.1.2 Phase transition

Usually, local models are combined into a global model by smooth superposition. The main motivation for this is that the system usually has some smoothness properties, i.e. with the operating point changing the phenomena or behavior change smoothly. However, one may occasionally come across processes that are non-smooth, in the sense that they exhibit abrupt changes in dynamics, for example phase transition or flow pattern changes [Söderman et al., 1993]. Below a mixturc of phases model will be introduce to characterize phase transition, which lead to mixture of local models with overlapping.

In general, phase transition means a system undergoes discontinuous changes in its behaviors as a result of continuously changing parameters, transforming from one phase to another. For example, a liquid flow changes from layered flow to turbulent flow with the Ronald number increasing. In the current case, by phase transition we mean a system undergoes a change in its underlying local features, thereby leading to different local models.

According to the modern classification scheme, phase transitions fall into two broad categories, namely the first-order phase transitions and the second-order phase transitions. Under this scheme, phase transitions were labeled by the lowest derivatives of the free energy that is discontinuous at the transition. First-order transitions exhibit a discontinuity in the first derivative of the free energy, or the value of the response variable, with a thermodynamic variable. In contrast, second-order transitions are continuous in the value of the response variable but might be not in the second derivative of the free energy.

In this scheme, first-order transitions are associated with "mixed-phase regimes", in which some parts of the system have completed the transition and others have not. A typical example of this class of transitions is water boiling, in which with temperature increasing the water does not instantly turn into gas but forms a turbulent mixture of water and water vapor. Mixed-phase systems are instable and difficult to study, because their dynamics are violent and hard to control. However, many important phase transitions fall in this category, including the solid/liquid/gas transitions.

Similar to soft partition, in the present case we would adopt soft phase boundaries, where there coexist multiple phases. This is consistent with overlapping operating regimes mentioned earlier. During the coexistence region, the system can be considered to be a random mixture of multiple phases governed by different local models. This stochastic mixture model of phase transitions helps explain instability during the transitional regime, because inside this coexistence region the system behaves like one of the distinct phases with certain probabilities and distinct phases are quite different in behaviors. Therefore, the expected behavior of the system is simply a weigthed average of those of distinct phases. At the sametime, outside the coexitence regime the system is dominated by a single phase and thus a local model can be applied deterministically. This statistical mixture model of phase transition is not short of evidence from the real world. For example, Harrington et al. [1997] reported that in liquid-liquid phase transition they found the coexistence of two different phases inside an unstable region.

In addition, generally we have no idea in advance where and how wide the transitonal regions are and thus they have to be estiamted based on data. Conveniently, each transitional region can be represented by two parameters, the outset and the end-point of a phase transition. In fact, this parametric model is able to characterize both categories of phase transitions, namely first-order or second-order transition. If the width of the coexistence regime turns out to be 0, it is a first-order phase transition; otherwise, it is a second-order phase transition.

Based on the above argument, the framework of mixture of local models is able to model the phase transition phenomenon with the addition of dynamic identification of operating regimes.

7.2.2 Adaptive fuzzy parametric clustering

In our local model scheme, the input space is split into partitions, which overlap somehow because of coexistence region. Each partition corresponds to a distinct operating regime, in which the system can be described by a local model. In implementing this scheme, the first and a key problem is how to split the input space, which is actually a problem of clustering.

Clustering can be considered the most important unsupervised learning problem, which is intended to organized objects into groups whose members are similar in some way. Clustering involves the task of dividing data points into homogeneous classes or clusters so that items in the same class are as similar as possible and items in different classes are as dissimilar as possible. Thus, the goal of clustering is to find common patterns or similarity.

The measure of similarity plays an essential role in clustering algorithms. Usually, distance is employed as the similarity criterion. However, it seems inappropriate for the current situation. In our scheme, if two points in the input space belong to the same phase, they are thought of behaving similarly and hence as in the same cluster. From the perspective of local features, points in the same cluster have the same underlying local features. Furthermore, since currently local features are extracted from candidate models through ICA, similarity in local features is equivalent to similarity in the separating matrix W. Therefore, it is more reasonable to cluster data based on the similarity of the mixing matrix W.

To meet different needs, many clustering algorithms have been proposed, which can be roughly grouped into two classes, namely hard clustering and soft clustering.

Hard clustering assumes exclusive assignment of each datum to clusters, which means that a certain datum belonging to a definite cluster could not be included in another cluster simultaneously. So hard clustering results in crisp clusters, where each data point belongs to exactly one cluster. An example of this class is the *K*-means clustering algorithm. Its application is mainly in pure local piecewise models such as CART [Breiman et al., 1984].

A special case of hard clustering algorithms is hierarchical clustering [Johnson, 1967]. In hierarchical clustering the data are not partitioned into a particular cluster in a single step. Instead, a series of partitions takes place, which may run from a single cluster containing all objects to *n* clusters each containing a single object. The clusters in each step can be organized in a hierarchical tree.

On the contrary the soft clustering, also called overlapping clustering, allows each point to belong to two or more clusters simultaneously. Corresponding to the two existing uncertain reasoning techniques, fuzzy set theory and probability theory, this class of clustering algorithms can be further divided into fuzzy clustering and probabilistic clustering. In fuzzy clustering, fuzzy clusters are identified and the data points can belong to more than one cluster associated with membership grades, which indicate the degree to which the data points belong to the different clusters. The fuzzy c-means algorithm is one of the most widely used fuzzy clustering algorithms, which is developed by Dunn [1973] and improved by Bezdek [1981].

Similar to the fuzzy clustering, in probabilistic clustering each data point has certain probability of belonging to a particular cluster. This probabilistic reasoning is implied by restricted amount of evidence. A most used probabilistic clustering algorithm is the mixture of Gaussians, where the well-known Expectation-Maximization algorithm is applied to estimate parameters.

As pointed out in [Jordan and Jocob, 1994], divide-and-conquer technique tends to increase the variance and a simple remedy to this problem is to apply soft partition. This is also applicable to the current case and thus we favor soft clustering. Another fact making soft clustering appealing is that many systems change behaviors smoothly as a function of inputs and soft transition between regimes introduced by the fuzzy set representation characterize this feature in an elegant fashion. However, both fuzzy clustering and probabilistic clustering cannot be applied directly, because fundamentally they measure the similarity based on distance and are separated from the modeling process, which is inappropriate in our case. Meanwhile, in comparison to probabilistic algorithms, fuzzy clustering is more natural and flexible in the current case. Therefore, we will propose a new adaptive fuzzy clustering algorithm below to identify different phases over the entire input space. The characteristics of the new fuzzy clustering algorithm are described in detail in the following.

(1) Fuzzy clustering

A natural way to interpret overlapping operating regimes is to apply fuzzy set, because an operating point falls in two or more operating regime simultaneously. According to our scheme, in the overlapping regions multiple local models might be relevant while outside the coexistence regions only one local model, which is called dominant local model, is valid. The simple trapezoid fuzzy membership function is specifically suitable to characterize such operating regimes. Furthermore, the choice of trapezoid shape membership function produces more interpretable local models than other functions like Gaussians [see e.g. Honda et al., 2000]. It is also worth noting that other kinds of fuzzy membership functions can also be employed so as to make the global model in possession of some meaningful characteristics. This will be discussed later on.

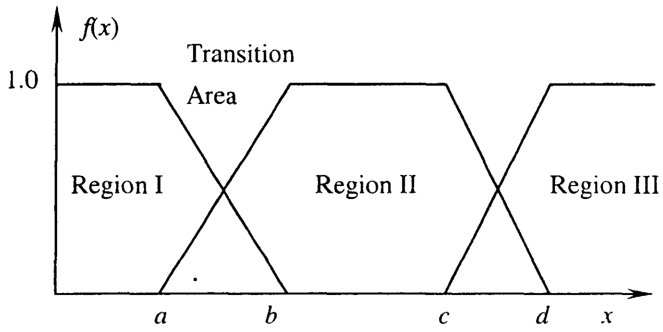


Figure 7.1 Fuzzy clustering

In order to be consistent with the concept of mixture model and superposition, a constraint on the fuzzy membership functions is imposed, which requires that at any point x,

$\sum_{m=1}^{M} \mu_m(x) = 1$. This results in smooth transition between operating regimes.

Clusters or operating regimes are represented by fuzzy sets. A typical fuzzy clustering with three overlapping operating regimes is depicted in Figure 7.1. Any point in the input space might belong to multiple clusters with memberships simultaneously. From another practical angle, the membership can be interpreted as how possible an observation was possibly generated by a certain local model.

(2) Parametric

In our fuzzy clustering scheme, the fuzzy membership functions are parameterized by the number of clusters and the locations of splitting points.

As usual, the main task of fuzzy clustering is to identify fuzzy sets characterized by parametric fuzzy membership functions. For each cluster, the parameters include the location of boundaries and their widths. The importance of the locations of boundaries is obvious. They can be chosen such that the similarity within a cluster is maximized while the patterns of different clusters should be as dissimilar as possible. Only so, in each cluster the system can be better represented by a local model and the bias will be decreased.

The overlap or the width of coexistence region plays a major role in smoothening the transition between local models. Murray-Smith and Johansen [1995] further argue that overlap has a regularizing effect in the ill conditioning in a learning problem and the level of overlap determines the amount of regularization. High level of overlap leads to high level of correlation between neighboring local models and decreased transparency of the local models, i.e. compatibility with the understanding of a system [Johansen and Foss, 1997], but on the other hand low level of overlap results in non-smooth transition between models. Hence, the optimal degree of overlap and softness depends on the modeling problem through the objective function.

This algorithm is called adaptive in the sense that in addition to the separators the number of regions is determined based on data. If the number of clusters is not large enough, the nonlinearity of the system cannot be caught adequately. On the other hand, an increasing number of operating regimes increase the model complexity. The overall effect of an increasing number of local models depends on where the decrease in bias is more significant than the increase in variance.

Thus, the number, location and overlap of the operating regimes should be so tuned dynamically as to reach optimal values, which is determined by objective functions. In so doing, it can be ensured that there are adequate amount of data within each operating regime to get a good local model.

(3) Objective function

The goal of modeling processes is to minimize the predictive error. Likewise, local modeling also aims to minimize the generalization error across the entire input space. Both the splitting of the input space and the building of local models should be determined by this overall goal. Nevertheless, in most previous work such as local PCA [Kambhatla and Leen, 1997] or local ICA [Karhunen and Malaroiu , 1999], local models [McLachlan and Basford, 1988], clustering is treated as a separate optimization problem from local learning and global mixture, which thus causes sub-optimality. In this chapter, we will optimize both problems jointly by incorporating them in a single objective function, which reflects the overall goal of minimizing the global generalization error. This goal can be realized by two steps, namely estimation of clustering parameters given the number of clusters and estimation of the number of clusters. For the first step, it can be done by minimizing the empirical error.

Until now, we have not discussed how to create local models and the global mixture model. For the time being, let's suppose the global mixture model as $f_g(x, \alpha, \beta)$, where α denotes the clustering parameters and β refers to other local model parameters. Therefore, the partial objective can be expressed as

$$\arg \min_{\alpha} \min_{\beta} \sum_{i=1}^{n} \left(y_i - f_g(x_i, \alpha, \beta)\right)^2 , \tag{7.3}$$

where the squared loss is applied.

If the normal distribution of the data is assumed, minimizing the sum of squared error is equivalent to maximizing the likelihood. So, equivalently all parameters can be estimated by MLE. This will turn out to be very useful later on.

From the partial objective function in equation (7.3), it is seen that the input space is so split as to minimize the empirical error. In this sense, this clustering algorithm is data-driven.

Therefore, through a same objective function fuzzy clustering is closely bound up with modeling process. This is exactly in agreement with Johansen and Foss [1997] that the creation of local models should not be separated from the choice of operating regimes. In this aspect, it is similar to MoE, where probabilistic clustering is mixed with learning.

Nevertheless, the partial objective function in equation (7.3) does not involve the number of clusters, which is in fact another important part of our fuzzy clustering. This is because a different number of operating regimes lead to varied model structures, which cannot be reflected in the empirical error. Following the principle of parsimony in model selection, we will bring forward a complete objective function, which incorporates the effect of the number of clusters.

The divide-and-conquer principle reduces the model bias by specifying the model structure more properly, but the variance will be increased at the same time, because with an increasing number of local models (increasing model structure) more parameters need to be estimated, which leads to larger variance on the parameter estimate. This phenomenon is well known in the literature of statistics as bias/variance tradeoff [see e.g. Geman *et al.*, 1992]. On one hand, if the input space is split into too many regions, overfitting will occur; on the other hand, too few regions might not capture the structure in enough detail and thus leads to underfitting. The task here is to find out the optimal balance point within the bias/variance tradeoff given a finite number of samples. To this end, generally two different strategies can be employed, namely model selection or regularization. In the present case, our purpose is to choose the optimal number of operating regimes, and thus model selection seems more appropriate.

As mentioned earlier, with an increasing number of operating regimes local models can fit the data much better, but the model complexity is also increased, which usually leads to deteriorating generalization. Generally, the higher the model complexity is, the smaller the bias but the larger the variance. Most model selection criteria realizes Occam's razor by penalizing the goodness-of-fit with model complexity, thereby minimizing the generalization error. Among all model selection methods, the information theoretical criteria like AIC [Akaike, 1973] or BIC [Schwarz, 1978] have a close connection to the maximum likelihood method, which seems to many statisticians an advantage. As a result, these information criteria can be easily applied in many circumstances without any additional computation. In addition, some of these information criteria including AIC and Bayesian Information criterion (BIC) can be justified under a Bayesian framework, which is also viewed by many statisticians as another big advantage [see Akaike, 1978 and Schwarz, 1978]. Therefore, in this chapter, we will only utilize the BIC for the purpose of demonstration.

BIC was first derived by Schwarz in a Bayesian context with a uniform prior probability on each competing model and priors with everywhere positive densities on the model parameters θin each model. Choosing the model dimensionality with the highest posterior probability leads to the BIC criterion of Schwarz [1978],

$$BIC = -2\log L(\hat{\theta} \mid x) + k \log n \,, \tag{7.4}$$

where $L(\hat{\theta} \mid x)$ is likelihood function of data x and maximum likelihood estimate $\hat{\theta}$, k is the number of parameters or model dimension, and n is the sample size. Note that the first term on the RHS comes directly from the general maximum likelihood and the second term is a complexity penalty term.

Assuming that the errors in data are Gaussian, we can obtain the BIC formula after some mathematical manipulations

$$BIC = -n[\log(n/2\pi) - \log RSS - 1] + k \log n \,, \tag{7.5}$$

where RSS is sum of empirical squared error, i.e. $RSS = \sum_{i=1}^{n} \left(y_i - f_g(x_i, \alpha, \beta)\right)^2$.

In the current situation where the Gaussian errors are not homogeneous, the likelihood function can be written as

$$L(\hat{\theta} \mid y, x) = \prod_{i=1}^{M} \prod_{j=1}^{n} L_i(\hat{\theta} \mid y_j, x_j)^{\mu_{ji}}, \tag{7.6}$$

and thus the log likelihood becomes

$$l(x, y) = \log L(\hat{\theta} \mid y, x) = \sum_{i=1}^{M} \sum_{j=1}^{n} \mu_{ji} \log L_i(\hat{\theta} \mid y_j, x_j), \tag{7.7}$$

where parameters θ, including β and σ, are estimated by MLE.

Because the log likelihood for a local model can be expressed as

$$\begin{aligned} l_i(x, y) &= \sum_{j=1}^{n} \mu_{ji} \log L_i(\hat{\theta} \mid y_j, x_j) \\ &= \sum_{j=1}^{n} \left(-\mu_{ji} \log \sqrt{2\pi} - \mu_{ji} \log \sigma_i - \frac{\mu_{ji}(y_j - \beta_i^T x_j)^2}{2\sigma_i^2} \right), \end{aligned} \tag{7.8}$$

and therefore

$$\hat{\beta}_i = \arg\min_{\beta_i} \sum_{j=1}^{n} \mu_{ji} (y_j - \beta_i^T x_j)^2 \tag{7.9}$$

and the variance for the ith local model

$$\sigma_i^2 = \frac{\sum_{j=1}^{n} \mu_{ji} (y_j - \beta_i^T x_j)^2}{\sum_{j=1}^{n} \mu_{ji}} = \frac{WRSS_i}{\sum_{j=1}^{n} \mu_{ji}}, \tag{7.10}$$

where $WRSS_i = \sum_{j=1}^{n} \mu_{ji} (y_j - \beta_i^T x_j)^2$.

At last, we will have a slightly different formula

$$BIC = \sum_{i=1}^{M} \left\lfloor \sum_{j=1}^{n} \mu_{ji} \cdot \log(2\pi) + \sum_{j=1}^{n} \mu_{ji} \cdot (\log \sigma_i + 1) \right\rfloor + k(M) \log n, \tag{7.11}$$

Finally, we obtain the complete objective function as

$$\arg \min_{M} \min_{\alpha} \min_{\beta} \sum_{i=1}^{M} \left\lfloor (\log(2\pi) + \log \sigma_i + 1) \cdot \sum_{j=1}^{n} \mu_{ji} \right\rfloor + k(M) \log n, \tag{7.12}$$

where M refers to the number of operating regimes and the model dimensionality $k(M)$ is a function of it.

Usually, the penalty term is equal to the number of free parameters that need to be estimated based on data. In the current case model parameters include clustering parameters and local model parameters. Therefore, the model complexity can be evaluated by

$$k(M) = 2(M-2) + \sum_{m=1}^{M} p_m \,, \tag{7.13}$$

where the first term refers to the number of clustering parameters specifying the boundary positions and the p_i denotes the model dimensionality of each local model. However, in fact the parameters in local models are not free, because once the operating regimes and regression methods are specified, the local model parameters are already determined, which means the local models cannot be tuned independent of clustering parameters, but fully determined by operating regimes separation and regression methods. Therefore, in choosing the number of clusters the appropriate model complexity can be expressed as

$$k(M) = 2(M-2) \tag{7.14}$$

In a special case where the width of the coexistence region is zero, we just count the number of parameters as 1.5 rather than 2 as implied by the above equation (7.14).

Note from the above that model selection is only for choosing the optimal number of operating regimes, because the penalty term only depends on the number of free parameters rather than their values.

By the complete objective function, not only will this algorithm determine the regime location, size and overlap, but it will also determine the number of regimes. The number of clusters depends on the sample size. Its upper bound should be such that in each cluster the number of data points having non-zero membership should be greater than the number of features. Note that such an objective function favors parsimonious models rather than under or over-parameterized model structures by optimizing the number of local models.

7.2.3 Local analysis

7.2.3.1 Local component analysis

ICA is a successful technique in reducing statistical dependence, and hence redundancy, between the candidate models. Dimension reduction is also achieved by eliminating a subset of independent components without significant loss of information.

PCA [see Hotelling, 1933 and Jolliffe, 1986] is another popular dimension reduction technique, which only relies on second order statistics and helps remove linear dependency. As a result, the principal components, although uncorrelated, can be highly statistically dependent. In contrast, ICA takes into account higher order statistics and is able to eliminate non-linear dependency. Therefore, ICA can produce a more compact representation of the data and outperforms PCA in statistical redundancy as well as dimension reduction. However, PCA is much easier than ICA and in some cases where no significant nonlinear dependence is involved, PCA can produce satisfactory results. Thus, although in the following we will mainly focus on local ICA, it is also directly applicable to local PCA.

However, ICA still has some limitations as we pointed out earlier, namely its linearity and globality. To overcome the limitation of global linearity, recently researchers propose nonlinear ICA as in equation (7.2). However, the difficulty of nonlinear ICA consists in the fact that the solution of nonlinear ICA problem is usually highly non-unique [see e.g. Hyvärinen and Pajunen, 1999] and is computationally rather demanding [Karunen and Malaroiu, 1999b]. In order to develop non-linear extensions of ICA, we propose to use a local linear ICA, in which the data space is first partitioned into disjoint regions somehow and then ICA is performed within each cluster.

Local linear ICA can provide an approximation of nonlinear ICA because based on Taylor expansion the nonlinear mixing function f in equation (7.2) can be approximated locally at any point by linear functions. By choosing the number of regions adaptively, the nonlinear characteristic can be represented adequately in the sense of accuracy given limited observations. At the same time, linear ICA is utilized to extract local features within each more homogeneous domain. Thus, multiple sets of local features rather than global features are produced.

Therefore, local ICA can overcome some weaknesses of linear ICA while avoiding the problems associated with general nonlinear ICA. Local ICA usually works in conjunction with a suitable clustering algorithm, which is responsible for partitioning the data space into clusters. For example, Karhunen and Malaroiu [1999] proposed to use k-means clustering algorithm, and Honda et al. [2000] suggested using fuzzy c-varieties clustering [see Bezdek, 1981], which partition the data space based on the similarity of the mixing matrix.

In this chapter, based on our overall model a different adaptive fuzzy clustering is proposed, which is described in a previous section. After clustering, Fast ICA algorithm [Hyvärinen, 1999] is applied in each cluster to extract local independent components. It seems more appropriate to employ weighted ICA based on fuzzy memberships, because even intuitively the points having smaller fuzzy memberships, in coexistence region for example, should have smaller influence on feature extraction. However, since the coexistence regions are not so wide, for simplicity we apply the standard ICA in each cluster.

7.2.3.2 Local models

Once local features are extracted, we can proceed to construct local models, each pertaining to a different, somehow overlapping though, operating regimes of the input space. Since each local model is only relevant to one cluster, it is reasonable to train local models using data points belonging to that cluster. Thus, before building local models, all observations need to be first assigned to the fuzzy clusters, which constitutes a fuzzy multi-category classification problem. Because from the step clustering we know the membership functions $\mu_j(x)$ of all fuzzy clusters, the classification task is simply to evaluate the membership of each data point in all clusters, that is,

$$\mu_{ij} = \mu_j(x_i), \tag{7.15}$$

which specifies the influence of each data point in building local models pertaining to all operating regimes.

Just as in the global version of model combination method by decomposition and aggregation (MCDA) in chapter 3, local models are created by multiple linear regression models,

$$f_m(x) = \sum_{j=1}^{p} \beta_{mj} h_{mj}(x), \tag{7.16}$$

where $h_{mj}(x)$'s refer to local independent features in the m-th fuzzy operating regime.

Taking into account varied influence of data points, the parameters in local models can be estimated by weighted least squares

$$\beta_m = \arg\min_{\beta_m} \sum_{i=1}^{n} \mu_{ij}\left(y_i - f_m(x)\right)^2, \tag{7.17}$$

which further encourages the locality of local models.

Local feature selection is an integral part of building local models. The purpose of feature selection is to eliminate non-informative features and noise and remove redundant information, thereby reducing model dimensionality. Since multiple linear regression method is used in constructing local linear models, feature selection is actually a variable selection problem. Feature selection results in parsimonious models, which are known to yield improved generalization.

From the above, it is easy to see that the building of local models are separated from each other expect for somewhat overlapping. Thus, local feature selections can be also performed separately in each operating regime. As a result, the local feature selection only depends on empirical errors of each observations falling in a certain cluster.

Up to date, there are a variety of variable selection methods. In order to keep the interpretability of variable selection while improve its stability, the newly developed fuzzy variable selection in Chapter 4 is applied.

7.2.4 Combine local ICA models

Once local models are built for all clusters, they can be combined into a global mixture model, or so-called operating based model according to Murray-Smith and Johansen [1997], based on our phase transition model.

Based on the fuzzy membership functions, the final global mixture model can easily be formulated as a fuzzy weighted average model

$$f_g(x) = \sum_{m=1}^{M} \mu_m(x) f_m(x), \tag{7.18}$$

where fuzzy membership function $\mu_m(x)$ characterizes the operating regime of the m-th local model $f_m(x)$. Each local ICA model can be expressed as

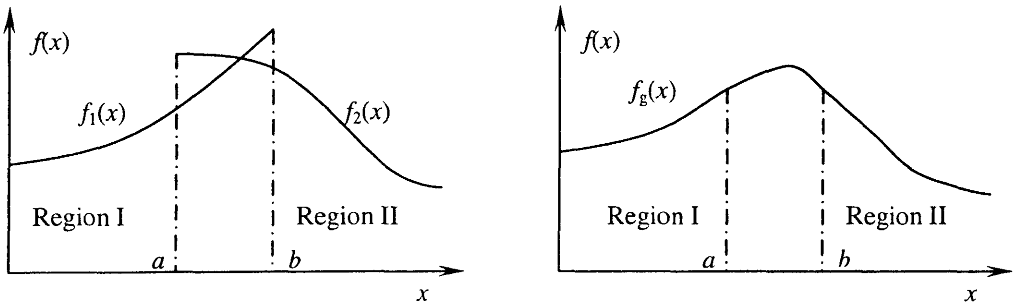
$$f_m(x) = \sum_{j=1}^{p} \beta_{mj} h_{mj}(x), \tag{7.19}$$

where $h_{mj}(x)$ denotes j-th local independent component for m-th cluster and β_{mj} is its corresponding regression coefficient.

In the present case simple trapezoid membership function is applied to represent fuzzy operating regimes and for any point x a constraint is imposed such that $\sum_{m=1}^{M} \mu_m(x) = 1$, so the mixture of local models turns out to be simple. If for given point x, which is outside unstable phase transition regions, some $\mu_m(x)=1$, the m-th local model $f_m(x)$ is dominant and thus $f_g(x) = f_m(x)$; otherwise, if x is inside a transition region, two different phases characterized by two different local models coexist. Based on our stochastic modeling of phase transition, the mixture model $f_g(x)$ can be constructed by a weighted linear superposition of local models with nonzero membership by their degrees of membership,

$$f_g(x) = \mu_i(x) f_i(x) + \mu_j(x) f_j(x). \tag{7.20}$$

An example with two operating regimes is shown in Figure 7.2.



(a) Local models

(b) Global mixture model

Figure 7.2 An example with two operating regimes

From the above example, it can be seen that the global mixture model is continuous. In fact, it is true as long as the width of the coexistence region is not zero, because, for example,

$$f_g(x_a) = \mu_1(x_a)f_1(x_a) + \mu_2(x_a)f_2(x_a) = f_1(x_a), \qquad (7.21)$$

where $\mu_1(x_a)$=1 and $\mu_2(x_a)$=0.

Nevertheless, in general the first derivates are not continuous. This is because with trapezoid membership functions we have

$$\begin{aligned} f_g'(x_a) &= \mu_1'(x_a)f_1(x_a) + \mu_1(x_a)f_1'(x_a) + \mu_2'(x_a)f_2(x_a) + \mu_2(x_a)f_2'(x_a) \\ &= c(f_1(x_a) - f_2(x_a)) + f_1'(x_a) \end{aligned} \qquad (7.22)$$

where $\mu_1'(x_a)$=-c and $\mu_2'(x_a)$=c.

Certainly, if we want the global mixture model is smooth in the sense of first derivatives, we should choose other membership functions than linear ones in the coexistence region such that $\mu_1'(x_a)$=$\mu_2'(x_a)$=0. A candidate of such membership functions can be

$$\mu_1(x) = \frac{1}{2}\left[1 + \cos\left(\frac{\pi(x - x_a)}{x_b - x_a}\right)\right], \mu_1(x) = \frac{1}{2}\left[1 - \cos\left(\frac{\pi(x - x_a)}{x_b - x_a}\right)\right], \text{for } x \in [x_a, x_b]. \qquad (7.23)$$

Following this strategy, global model with even higher orders of smoothness can be produced.

7.3 Three-stage optimization algorithm

In the problem under investigation, we need to jointly optimize the number of operating regimes, the locations of boundaries as well as parameters in local models. Seemingly, it is somehow analogous to adaptive regression splines with free knots [cf. Jupp, 1978 and Friedman, 1991]. Splines are piecewise polynomial functions that are constrained to join smoothly at points called knots. In particular, a free-knot spline is a spline where the knot locations are considered parameters to be estimated from the data. Freeing the knots greatly improves the spline's approximating power [Burchard, 1974]. However, it poses a very difficult problem, that is, estimating the optimal number of knots and their locations, which is similar to the problem we are facing.

Nonetheless, their differences are also obvious. First, our model is more flexible without smoothness constraints, which, on the other hand, leads to more free parameters. Second, in our model the regressors nonlinearly depend on the boundary locations while for splines regressors are fixed as polynomials. Therefore, it is expected that the current optimization problem is even more difficult than that of free-knot splines.

The difficulty lies in some undesirable characteristics of the complete objective function in equation (7.12). To help analyze its properties, let's substitute $f_g(x, \alpha, \beta)$ in equation (7.5) and rewrite it as

$$\arg \min_M \min_\alpha \min_\beta n \log \sum_{i=1}^{n} \left(y_i - \sum_{m=1}^{M} \mu_m(x_i, \alpha) \sum_{j=1}^{p_m} \beta_{mj} h_{mj}(x_i, \alpha) \right)^2 + k(M) \log n . \quad (7.24)$$

First, it is a nonlinear function of boundary locations, because the membership functions and the local independent components nonlinearly depend on α and they appear inside the square. This makes it a complex system.

Second, it is non-differentiable partly because of the non-differentiable membership function. Furthermore, the explicit dependence of the local independent components, or equivalently the separating matrices, on the fuzzy clustering parameters α can be never known. Because of this non-differentiability, all gradient-based optimization algorithms such as steepest descent, Newton-Raphson method and conjugate gradients will certainly fail. This excludes the application of most gradient-based method, but some numerical searching algorithms are still possible.

Finally, the objective function is not strictly convex or concave but has many local optima. This can be seen by applying the "lethargy" theorem introduced by Jupp [1978]. Similar to free-knot splines [Jupp, 1978], the existence of multiple optima is the objective surface is related to the symmetry introduced by the exchangeability of the boundary parameters. For example, in a simple case with two clusters the objective surface is symmetric along any normal to the line defined by two equal parameters. Consequently, the derivative along the normal at the intersection to the equal-parameter line is zero. This property, called "lethargy" by Jupp [1978], results in many stationary points and ridges along lines or planes in the parameter space where two or more parameters coincide.

This property will result in the failure of all local optimization algorithms like gradient-based methods, line search and hill climbing. Local optimization algorithms easily get stuck at local optima and with different initialization they will converge to varied local optima.

In order to overcome this problem, global optimization algorithms like simulated annealing and genetic algorithms should be applied instead.

The above are some interrelated reasons that make it so difficult to find out the global optimum. Since there are many local optima in the objective surface, good starting parameter values are essential for finding the global optimum. Unfortunately, it is usually difficult to construct good starting values that will converge to the global optimum. One possible way is to construct starting values based on data. First, we sort the inputs x and split the input range to segments s_1 through s_{n-1}. Clearly, every parameter α_i can be within each one of the segments. Then, the entire parameter space of the vector α will be divided into $(n-1)^{2M-2}$ pieces of subspaces. If we pick an initial value for α within each subspace, we will end up with a local optimum within that subspace. Comparing all these local optima will give us an approximate global optimum. However, because we cannot make sure there is only one local optimum within each subspace, the global optimum is not guaranteed. Furthermore, it is computationally very expensive because there are $O(n^{n/k})$ possible initial values in total.

Originally developed by Holland [1975], genetic algorithm (GA) is a global stochastic search algorithm, which is less susceptible to getting 'stuck' at local optima than gradient search methods. But on the other hand they tend to be computationally expensive. In practice, genetic algorithms work very well on mixed (continuous and discrete), combinatorial problems. For example, Pittman [1999] suggests using GAs to optimize the knot locations in adaptive splines.

Here we will propose a similar hybrid genetic algorithm, which combines global optimization together with local search. It is different from that in [Pittman, 1999] in that a distinct genetic chromosome representation and correspondingly different genetic operators are defined. Furthermore, in this scheme GAs are only used to find out good starting values for the local search rather than the global optimum, which significantly cut down the computational time because of the slow convergence of GAs.

Following a strategy of problem splitting, the optimization problem can be solved through three stages.

First, let's optimize local model parameters given fuzzy clusters. In order to encourage competition and locality of local models, we might approximate the original objective function with a slightly different one

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} \left(y_i - f_g(x_i,\alpha,\beta)\right)^2 \approx \arg\min_{\beta} \sum_{m=1}^{M} \sum_{i=1}^{n} \mu_{im}\left(y_i - f_m(x,\beta,\alpha)\right)^2 . \quad (7.25)$$

The only difference between the two objective functions is in the coexistence regions. For convenience, let's denote $E(\theta_i) = \sum_{m=1}^{m} \mu_{im}\theta_i$, and then we have

$$\begin{aligned}
&\sum_{m=1}^{M} \mu_{im}\left(y_i - f_m(x,\beta,\alpha)\right)^2 = E\left(y_i - f_m(x_i,\beta,\alpha)\right)^2 \\
&= \left(y_i - E\left(f_m(x_i,\beta,\alpha)\right)\right)^2 + E\left(f_m(x_i,\beta,\alpha) - E\left(f_m(x_i,\beta,\alpha)\right)\right)^2 \\
&= \left(y_i - f_g(x_i,\beta,\alpha)\right)^2 + \sum_{m=1}^{M} \mu_{im}\left(f_m(x_i,\beta,\alpha) - f_g(x_i,\beta,\alpha)\right)^2
\end{aligned} \quad (7.26)$$

where the second term is usually small within the coexistence regions.

In fact, another important thing is that this change makes the optimization problem much easier, because the membership function is moved out of the square. Moreover, another good thing about it is that local models can be built independently from each other, which is consistent with section 7.2.3.2.

In this stage, local model parameters β can be optimized as functions of α, which can be easily done by WLS as described in section 7.2.3.

Second, we need to find out the best fuzzy clustering given the number of clusters, which is actually the toughest stage. Here, we can rewrite the sub-optimization problem as

$$\hat{\alpha} = \arg\min_{\alpha} \sum_{i=1}^{n} \left(y_i - \sum_{m=1}^{M} \mu_m(x_i,\alpha) f_m(x_i,\hat{\beta}(\alpha),\alpha)\right)^2 = \arg\min_{\alpha} F(\alpha) , \quad (7.27)$$

where $\alpha=(\alpha_1,\ldots, \alpha_{2M-2})$ with $x_{\min}<\alpha_1\leq \alpha_2<\ldots< \alpha_{2M-3}\leq\alpha_{2M-4}\leq x_{\max}$.

Note that under the assumption of Gaussian errors the fuzzy clustering parameters are actually estimated by Maximum Likelihood. Nevertheless, a well-known problem with

MLE is the danger of overfitting. For a simple example, suppose we have two operating regimes and the number of data points falling in the first regime is equal to the number of candidate models and all others fall in the other regime. In this case, the data can be fitted perfectly in the first regime and a little better in the other regime than in the single regime case. Therefore, as a result the maximum likelihood is increase dramatically, but the resultant model most likely becomes worse.

In order to overcome this pitfall, we apply the cross-validation approach, that is, using testing likelihood in place of maximum likelihood. Correspondingly, the sub-optimization problem can be expressed as

$$\hat{\alpha} = \arg\min_{\alpha} \sum_{i=1}^{n_t} \left(y_{ti} - \sum_{m=1}^{M} \mu_m(x_{ti}, \alpha) f_m(x_{ti}, \hat{\beta}(\alpha), \alpha) \right)^2 = \arg\min_{\alpha} F_t(\alpha) \qquad (7.28)$$

where the subscribe t denote the out-of-sample test.

The sub-objective function $F_t(\alpha)$ in equation (7.28) is in possession of all the unpleasing properties mentioned earlier. To address this challenge, we will propose a hybrid optimization algorithm combining genetic algorithms with multi-dimensional hill climbing, which will be describe in detail a while later.

After accomplishing both global and local optimization procedure, a group of good candidate solutions are obtained, from which the solution with the smallest $F_t(\alpha)$ can be easily chosen as the optimal one.

The last stage is to choose the optimal number of local models, which is treated as a model selection problem. Since we restrict that in each operating regime the number of data points must be greater than the number of local independent components, therefore there exists an upper bound M_m much smaller than the sample size n. Thus, this optimization problem can be expressed as

$$\hat{M} = \arg \min_{1 \le M \le M_m} n \log \sum_{i=1}^{n_t} \left(y_t - f_g(x_t, \hat{\alpha}, \hat{\beta}) \right)^2 + k(M) \log n_t = \arg \min_{1 \le M \le M_m} G(M) \qquad (7.29)$$

To determine the optimal number of clusters, we simply repeat the second stage for M ranging from 1 to M_m and finally choose as the optimal the one corresponding to the minimal model selection criterion value in equation (7.29). In practice, a forward stepwise process can be utilized, which increases M from 1 and stops until the model selection criterion value increases.

7.3.1 Real-coded genetic algorithm (GA)

Basically, GAs are stochastic global search and optimization methods that mimic the metaphor of natural biological evolution, which are believed to be the ultimate optimizers based upon the idea of Darwin's revolutionary writing *Origin of Species*. According to Darwin, the power of evolution lies in the continuing struggle for survival and some "variation", such as mutation of genes, of an organism increases its chances for survival. Therefore, key to such evolution is the concept of larger numbers, i.e. large population and many generations, and randomness, such as the probabilistic selection, mixing, and mutation. Likewise, GA first generates a large set of random parameters as a generation, and then randomly select parameters in terms of their fitness to reproduce a new generation of parameters by means of random crossover and mutation. In GA, the fitness-based selection will ensure the consistent direction of evolution, or guarantee the increase of the average fitness of a generation. Meanwhile, crossover and especially mutation enables the GA to avoid being stuck in a local minimum and search for the global optimum.

Usually, a chromosome in GA is represented by a binary string consisting of 0s and 1s, but in the current case each parameter to be optimized is a floating point number, and therefore floating-point coding or double-precision representation seems more appropriate than binary coding. As pointed out by some research, binary coding is less suited for numerical optimization problems [Garcia, 1999], although a floating-point number can also be expressed in a binary form somehow, for example in Pittman [1999]. Therefore, each chromosome directly represents a vector of floating point parameters.

The first crucial issue of GA is to define a proper fitness function, which tells how good or bad a candidate solution is. It is this fitness function that determines the goal of optimization. Usually, GA works by maximize the fitness, but we intend to minimize the objective function or equivalently maximize the likelihood. Therefore, the fitness function can be constructed from the objective function as

$$fitness(\alpha^{(k)}) = Max(F_t(\alpha^{(j)})) - F_t(\alpha^{(k)}), \tag{7.30}$$

where $F_t(\alpha)$ is the objective function in equation (7.30), $Max(F_t(\alpha^{(j)}))$ stands for the maximum $F_t(\alpha)$ in a population and $fitness(\alpha^{(k)})$ refers to the fitness of the k-th individual chromosome.

The definition of fitness significantly influences the behavior of convergence. For example, in the early stage few "super individuals" tend to dominate the selection process leading to premature, whereas later when the population is less diverse, the simulation tends to lose focus [Goldberg, 1989]. Therefore, in practice we would like to apply a more general and flexible fitness function by scaling and shifting, i.e.

$$fitness(\alpha^{(k)}) = b + a\left(Max(F_t(\alpha^{(j)})) - F_t(\alpha^{(k)})\right) \tag{7.31}$$

where the scaling factors a and shifting factor b are so adjusted adaptively during simulation as to avoid premature convergence early on and encourage convergence in later stages.

As for selection, we utilize the fitness-weighted roulette wheel method, which is conceptually equivalent to giving each individual a slice of a roulette wheel equal in area to the individual's fitness. The wheel is spun and the ball comes to rest on the wedge shaped slice, and the corresponding individual is selected. Therefore, the probability for a chromosome to be chosen is proportional to its fitness. A pair of "parents" is selected by spinning the wheel two times to reproduce a pair of "children" by recombination and mutation.

As we know, the GA success is also sensitive to the two operators, namely, recombination operator and mutation operator. For example, it is found that the general, fixed, problem-independent recombination operators often break partial solutions and slow down convergence. In order to avoid such a pitfall, we design problem-specific crossover and mutation operators.

The recombination strategy we applied is the one-point arithmetic crossover. Let the parents be $P_1=[P_{11},\ldots,P_{1L}]$ and $P_2=[P_{21},\ldots,P_{2L}]$, respectively. Then, the two offspring are

$$C_{1i} = \begin{cases} P_{1i} & ,i \le t \\ P_{1t} + \dfrac{x_{\max} - P_{1t}}{x_{\max} - P_{2t}}(P_{2i} - P_{2t}), & i > t \end{cases}, \tag{7.32}$$

and

$$C_{2i} = \begin{cases} P_{2i} & i \le t \\ P_{2t} + \dfrac{x_{\max} - P_{2t}}{x_{\max} - P_{1t}}(P_{1i} - P_{1t}) & i > t \end{cases}, \tag{7.33}$$

where t is a random integer number among $1,2,\ldots,L$.

This crossover operator is so designed that it guarantee the resulting children are still ordered sequenced of real numbers within a valid range, the number of clusters is maintained and good patterns in parents can be kept as well. In fact, it is especially suitable for chromosome representations of ordered sequences of real numbers.

The crossover rate, i.e. the probability that crossover happens, is generally around 0.5, and in this chapter we set it as 0.6.

Mutation operator is defined as addition of a normally distributed factor with mean value 0, i.e. $D_i' = D_i + \varepsilon$, where D_i is an original parameter and D_i' is the mutated one, ε is a Gaussian random number, i.e. $N(0,\sigma^2)$, where σ^2 is tunable. ε plays a similar role of the step size in line search. In this study, we choose

$$\sigma = \frac{x_{\max} - x_{\min}}{3n}, \tag{7.34}$$

where n denotes the sample size.

Hessner and Manner [1991] suggested that the optimal mutation rate, i.e. the probability that mutation occurs for a single gene in a chromosome, is approximately $(S \cdot L^{1/2})^{-1}$, where S is the population size and L is the length of the chromosome. In this chapter, we will follow this "rule of thumb".

Since the current optimization problem is constrained by $x_{\min}<\alpha_1\le \alpha_2<\ldots< \alpha_{2M-3}\le \alpha_{2M-4}\le x_{\max}$, besides selection, crossover and mutation operator, we need another check operator before a new valid child chromosome is really created.

A valid chromosome has to meet some constraint. First, a chromosome must be an order sequence of real numbers. Second, each element must lie in between $x_{\min}$ and $x_{\max}$. Finally, another constraint is that the number of data points falling into each cluster must be greater than the number of local independent components.

If there is no crossover and mutation, a chromosome is simply copied to the next generation.

The stopping rule for the current case is relatively simple, as our purpose is to search for promising initial inputs for a local optimization algorithm. Thus, when we observe that the convergence of GA becomes very slow, it will be the time to stop it.

Finally, the main steps of GA are follows:

(1) Build an initial population of S chromosomes randomly between $x_{\min}$ and $x_{\max}$;

(2) Calculate the fitness of each chromosome;

(3) Select chromosomes from the parent generation to reproduce a child generation:

(i) Select two parent chromosomes,

(ii) Generate a random number between [0,1]. If it is smaller than the crossover rate, recombine them by one-point arithmetic crossover; otherwise, enter the next step;

(iii) Generate a random number between [0,1]. If it is smaller than the mutation rate, perform mutation on a gene in a chromosome. Repeat this for each gene in both chromosomes.

(iv) Add the two resulting chromosome to the next generation.

Repeat the above (i) through (iv) steps until S new chromosomes are reproduced.

(4) If the stopping criterion is met, then exit; otherwise, return to step (2).

7.3.2 Adaptive multi-dimensional hill climbing

By GA optimization, we obtain a set of global good initial guesses of the best vector of fuzzy clustering parameters, namely the last generation out of GA. In practice, it is also useful to keep tract of the "best" chromosome throughout the whole GA simulation history.The next task is to search for the optima around these good initial guesses.

As we noticed, our objective functions are quite complicated and thus it is hard to apply classical gradient-based methods. However, it can be solved numerically by some derivative-free approach, for instance, hill climbing. We will propose a derivative-free method to optimize the parameters one by one while keeping other fixed. Furthermore, the step size is adaptively tuned. However, since each parameter is not independent of each other, the overall optimization has to be done in an iterative way. Our algorithm described below is actually a multi-dimensional version of adaptive hill climbing, which is simple, self-adaptive and fast.

The whole process consists of multiple loops, in each of which the individual parameters are optimized one at a time. Suppose we are optimizing α_i and let its current value as $\alpha_i^{(0)}$ and the current model evaluation value as $F_t(\alpha)^{(0)}$. Let $k=1$ and $\alpha_i^{(k)}= \alpha_i^{(k-1)}+kd$, where d is small positive number, and keep the other p-1 parameters unchanged, and then recalculate the model evaluation value as $F_t(\alpha)^{(k)}$. If $F_t(\alpha)^{(1)} > F_t(\alpha)^{(0)}$, that is, the fuzzy model gets worse, then return to $\alpha_i^{(0)}$ and let k=1and replace d by -d; otherwise, continue to search in the same direction within the interval $[x_{\min}, x_{\max}]$ until $F_t(\alpha)^{(k+1)} > F_t(\alpha)^{(k)}$. The final $\alpha_i^{(k)}$ is taken as the optimal value in the current loop. After α_i is optimized, we turn to the next parameter α_{i+1}. Each loop starts with α_0 and ends up with α_{2M-2}. Once a loop is done, another one will be started depending on the stopping criterion.

At the beginning of each loop, we calculate the resultant model's $F_t(\alpha)$, and the same for the end of each loop. If the difference between these two values is small enough, for example,

$$\frac{\left|F_t(\alpha)^{(j+1)} - F_t(\alpha)^{(j)}\right|}{F_t(\alpha)^{(j+1)}} < \delta , \tag{7.35}$$

where $\delta$is very small, say 10^{-5}, we would say that the minimum has been reached and therefore stop the local searching process.

In view of the facts that (i) There exists a lower bound for $F_t(\alpha)^{(k)}$, although unknown, and (ii) the sequence of $F_t(\alpha)^{(k)}$ is non-increasing, the convergence is guaranteed according to the Cauchy convergence criterion.

7.3.3 Mixture of local ICA models procedure

Up to now, we almost complete the development of a new method of mixture of local ICA models, from model structure to parameter estimation. Basically, the input space of the system is first decomposed into fuzzy subspaces by adaptive fuzzy clustering algorithm and then in each subspace the system is approximated by a local linear ICA model. This is somewhat analogous to what is called Takagi-Sugeno fuzzy model [Takagi and Sugeno, 1985] in the context of predictive control.

The entire modeling procedure can be summarized as follows.

(1) Set M=1

(2) Optimize the fuzzy clustering with hybrid GA given the number of fuzzy clusters

(a) Build an initial population of S chromosomes randomly;

(b) Calculate the fitness of each chromosome;

(i)Classify data points into each fuzzy cluster

(ii)Perform local ICA within each fuzzy cluster

(iii)Create local ICA models based on data by multiple regression method with fuzzy variable selection

(iv) Mixing local ICA models

(v) Calculate the weighted residual sum of squared error

(c) Generate the next generation by selection, crossover and mutation

(d) If stop criterion is met, then go to (e); otherwise go to (b)

(e) Treat the last generation of GA as starting parameter values and find out local minima around them

(3) Choose the best local optimum as the global optimum and then assess the resulting optimal model by $G(M)$ in equation (7.29) . If $G(M) < G(M\text{-}1)$ for $M\geq2$, go to step (4); otherwise, go to step (5)

(4) Set M=M+1 and go to (2)

(5) Return the final optimal mixture model including the optimal fuzzy clustering

7.4 Numerical simulation study

In this section, we will present some results in our numerical simulation studies. This method will be first applied to an artificial example, where the true model is supposed to be known, to demonstrate how it works and its advantage over global models. And then, it is will be used in a real case. From our numerical simulation, we also intent to justify our argument that mixture of local models is suitable to situations where severe nonlinearity is involved, because the linear local models are supposed to catch its nonlinear characteristics.

7.4.1 Artificial example

In this example, artificial models and data will be used to demonstrate the effectiveness of the new method. Let's assume the true model is expressed in mathematics as

$$\begin{aligned} y(x) &= 150\text{-}150\exp(\text{-}2x) + x^2\text{-}0.1x^3 + 4x + \\ &30\exp(\text{-}x/3)\cdot\sin(x) + 15\sin(1.5x)\text{-}20\ln(x+1) \end{aligned}, \tag{7.36}$$

where real number $x \in$ [0,10]. From the expression, it is easy to note that it involves complex nonlinearity.

Correspondingly, its realistic data generative model can be written as

$$y = y(x) + \varepsilon, \tag{7.37}$$

where ε is supposed to assume a normal distribution, i.e. $N(0,\sigma^2)$ where σ^2 is set as 64 in the current example. From this generative model, we gathered a set of data with n=50, i.e. (x_i, y_i), where x_i is evenly distributed between [0,10].

Meanwhile, suppose we also collected a class of candidate models as follows:

$$f_1(x) = 150\text{-}150\exp(\text{-}2x) + 4x + 15\sin(1.5x)\text{-}20\log(x+1);$$

$$f_2(x) = 150\text{-}150\exp(\text{-}2x) + x^2\text{-}0.1x^3;$$

$$f_3(x) = 150\text{-}150\exp(\text{-}2x) + x^2\text{-}0.1x^3 + 30\exp(\text{-}x/3)\cdot\sin(x);$$

$$\begin{aligned} f_4(x) &= 150\text{-}150\exp(\text{-}2x)+6x+30\exp(\text{-}x/3)\cdot\sin(x)\text{-}20\cdot\log(x+1); \\ f_5(x) &= 150\text{-}150\exp(\text{-}2x)+x^2\text{-}0.1x^3+15\sin(1.5x); \\ f_6(x) &= 150\text{-}150\exp(\text{-}2x)+15\cos(2x)\text{-}15+0.004x^2; \end{aligned} \tag{7.38}$$

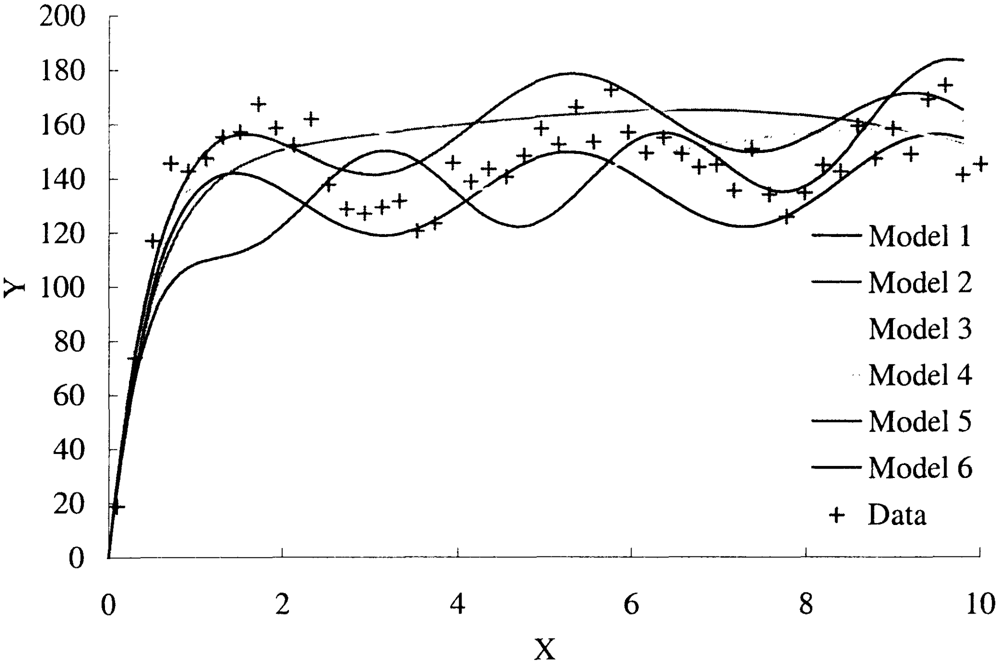


Figure 7.3 Artificial candidate models and data

Note that each candidate model is either incomplete or erroneous, or both. All these models are shown in Figure 7.3 together with a set of sample data.

Once the candidate models are formulized and data are collected, we are ready to employ our new approach to find out local domains by fuzzy clustering, build local models within each fuzzy cluster and finally mix local models into a global model. The results are shown in Table 7.1.

Table 7.1 Simulation results of the artificial example

Number of domains	Domains	BIC	Test error
A single best model	[0, 10]	N/A	104.67
1	[0, 10]	57.83	18.02
2	[0, 0.57, 4.865, 10]	55.15	11.68

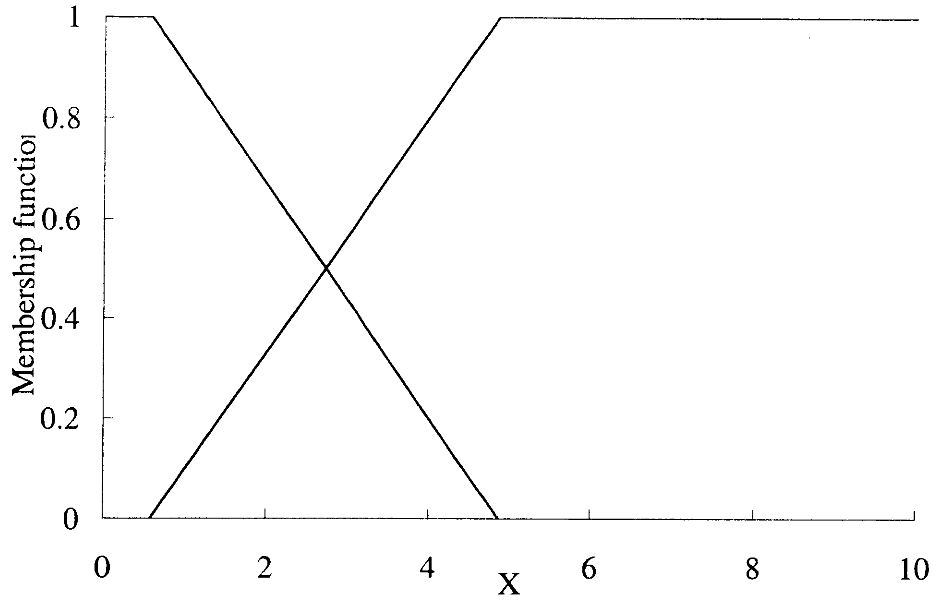


Figure 7.4 Membership functions

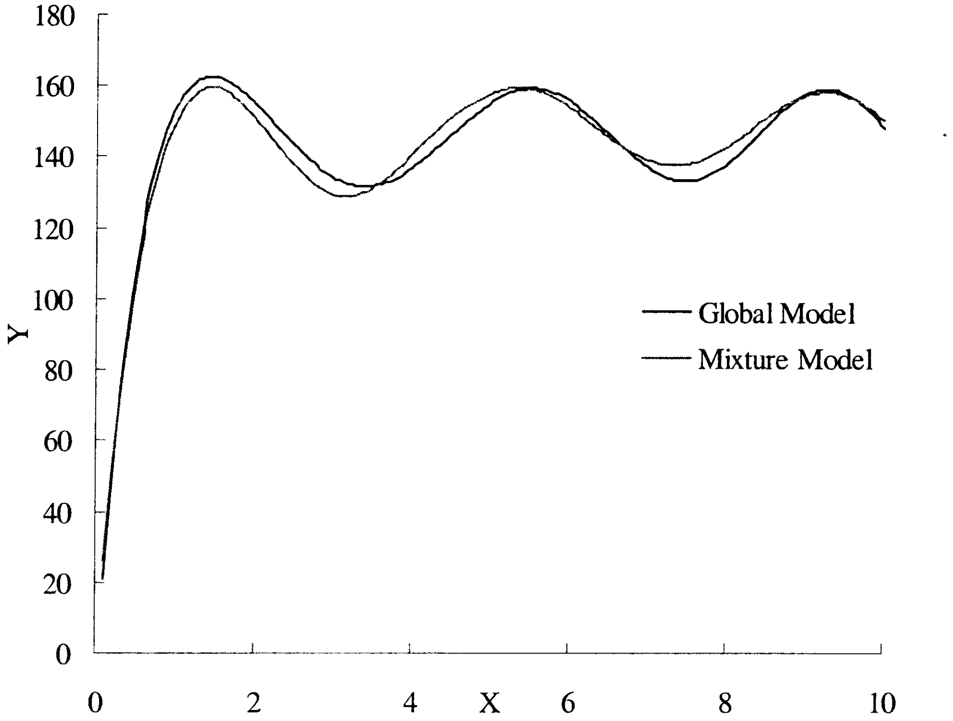


Figure 7.5 Local models and the mixture

The fuzzy membership functions and resultant mixture models are plot in Figure 7.4 and Figure 7.5, respectively. From the results in Table 7.1, we see that the new method does work pretty well as expected. This is quite understandable, because in this example highly non-linear dependence among candidate models and severe nonlinearity are involved, which are exactly the two primary problems that our new approach is supposed to deal with.

7.4.2 Real case study

In the above subsection, we demonstrated the effectiveness of our method using some toy data. Certainly, that is not enough without showing real applications. Now let's turn o a real example and see if it works.

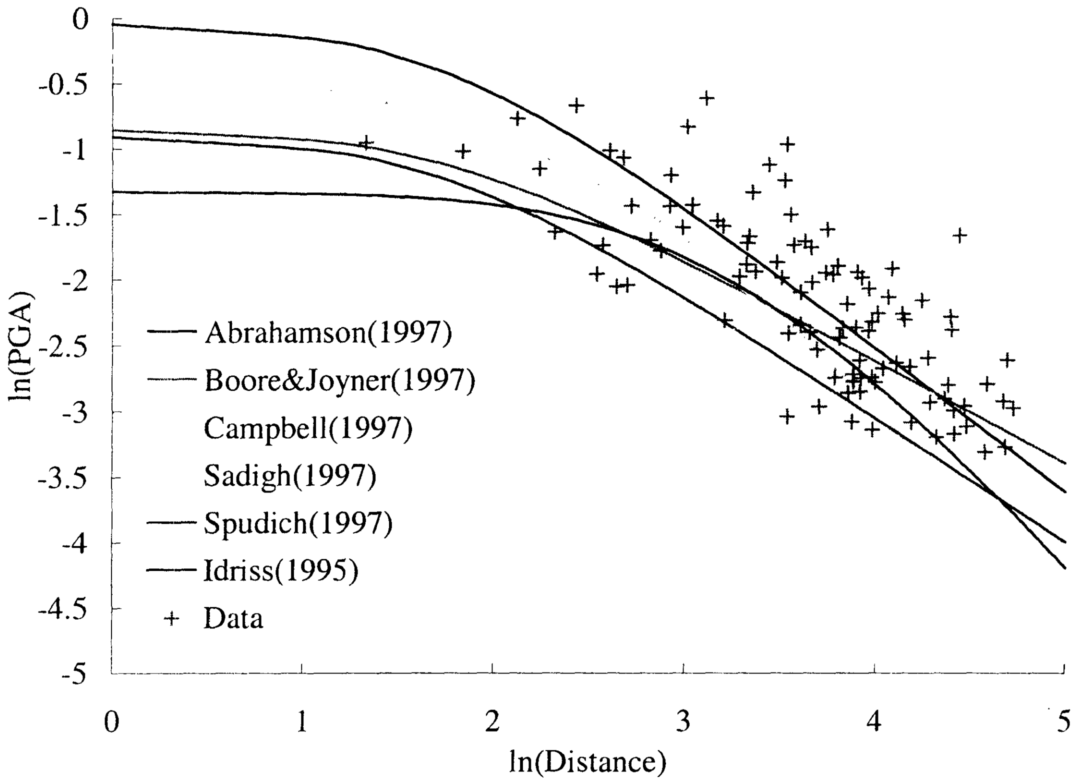


Figure 7.6 Candidate attenuation models and data

The real example we use here is the attenuation models in seismology. In this example, the purpose is to build a more accurate composite model, which is applicable to south California in the United States. A sample data set of size 102 is obtained from the literature [Steidl and Lee, 2000], whose logarithms are assumed to include Gaussian noise. Correspondingly, the candidate attenuation models include the attenuation relations by

Boore et al. [1997], Sadigh et al. [1997], Abrahamson and Silva [1997], Campbell and Bozorgnia [1997], Spudich et al. [1997] and Idriss [1995]. All of these attenuation relations may be found in Seismological Research Letters, Volume 68, Number 1, January/February, 1997. All these attenuation relationships were developed for shallow crustal earthquakes in active tectonic regions, and thus they should be applicable to southern California.

The candidate models are plot together with the sample data in Figure 7.6. From Figure 7.6, it is easy to note that all the models are close to be a straight line, which means unlike the artificial example the dependence among candidate models are mostly linear.

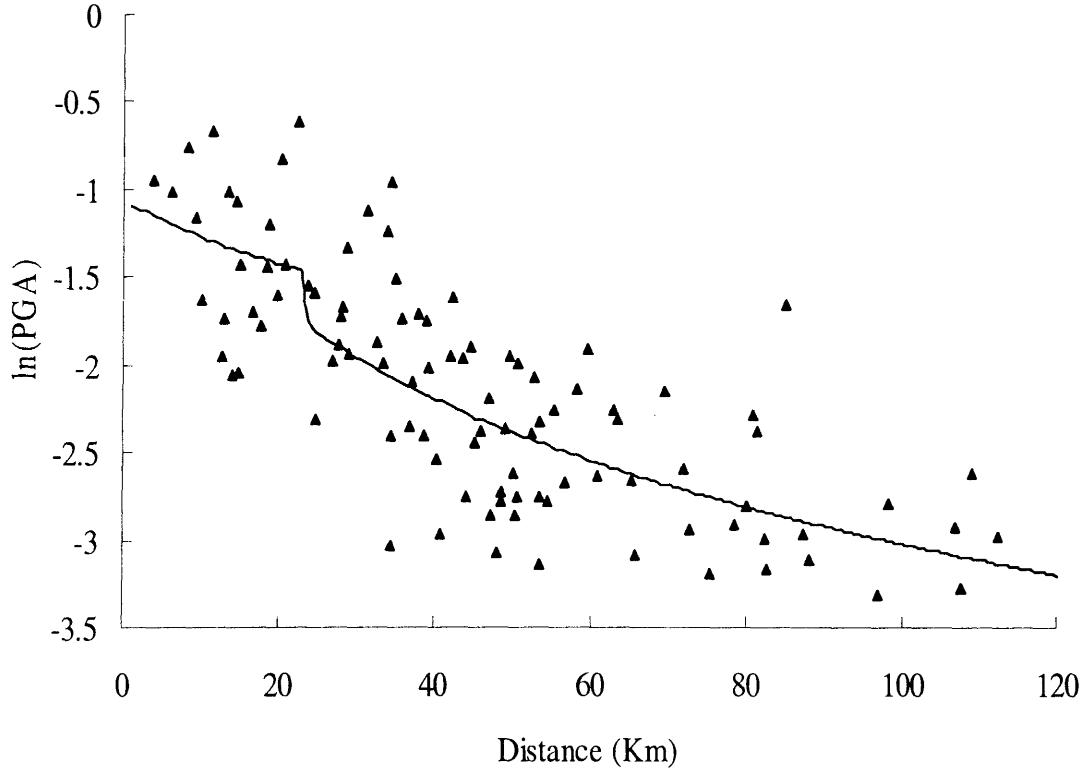


Figure 7.7 Local models and mixture model

As in the artificial example, we apply the new method to optimize fuzzy domains, create local models and finally come up with a mixture model. The simulation results are shown in Table 7.2 and the mixture model is plotted in Figure 7.7 together with the data.

Table 7.2 Simulation results of real case study

Number of domains	Domains	BIC	Test error
Best single model	[0,120]	N/A	0.1935
1	[0, 120]	-66.72	0.1567
2	[0, 23, 23.9, 120]	-69.37	0.1303

From the above results, we see that the test error of the mixture model with two operating regimes is smaller than that of the global model by about 12%. Besides, the models plotted in Figure 7.7 do make sense. First of all, the peak ground acceleration (PGA) goes down with the distance from the epicenter increasing. Meanwhile, it also shows that in two regions, namely near the epicenter and far from the epicenter, the attenuation model with regard to the distance are somewhat different. One of the possible reasons is the effect of the depth of the seismic source. Likewise, a possible explanation of the turning point around 23km is that for shallow crustal earthquakes the average depth of ruptures is about 25km [Campbell, 1997].

In this case study, once again we observed the significant influence that the choice of candidate models might have on the performance of the resultant mixture model. The choice of candidate models has been already fully discussed in chapter 2. Following the procedure proposed in chapter 2, we are able to come up with an optimal choice.

7.5 Summary

In this chapter, we propose a mixture model of local ICA models to overcome the weakness of global models in dealing with nonlinearity and locality. Basically, this method consists of three components: clustering, local ICA and model combination. Adaptive fuzzy parametric clustering algorithm is put forward to divide the whole input space into operating regimes, local ICA analysis are carried out and the feature-based model combination method is applied to create local ICA models in each individual region, and finally the local ICA models are combined into a mixture model. Correspondingly, a three-stage optimization procedure is designed to optimize the complete objective function, which is actually a hybrid GA algorithm.

In the end, to demonstrate its effectiveness this new method was applied to both an artificial example and a real case in seismic study. Our simulation results show that the adaptive fuzzy mixture of local ICA models turns out to be superior to global models.

Chapter 8

Summary and Contributions

This thesis mainly focuses on statistical modeling problems. The basic problem it is intended to solve can be expressed as follows:

Question: Given a class of competing models and without enough data, how can we construct a more accurate and precise composite model?

By competing, it is meant that none of the models is significantly superior to others. In fact, this is a very general problem in both science and engineering. This thesis is motivated by a project in seismic risk analysis, in which multiple different attenuation functions are available and people are not sure which one should be used or how to build a better one in terms of accuracy and uncertainty.

In general, there are two different ways to attack this problem, namely model selection and model combination. Clearly, model combination should be a better choice as it can incorporate more information. However, in order to combine multiple models, there are some challenges one has to face, which include

(1) Candidate model choice

There might be many different models for a certain system or phenomenon. They can be built at different times, by different people, in different disciplines, or on the basis of different theories and observations. This diversity can be greatly useful, but if we include every model without discrimination most likely we cannot succeed. Therefore, the first challenge is how to choose a group of candidate models to combine into a composite model.

(2) Model combination method

In order to create a better model, more information or more efficient use of existing information is necessary. The second challenge is how to aggregate information contained in candidate models to reduce model bias and uncertainty. Moreover, different kinds of information are involved, namely candidate models and data, which poses another difficulty.

(3) Model performance evaluation

Model performance evaluation sits in the core of model selection. It is also a central issue in model combination. Without model performance evaluation criteria, one cannot tell if a composite model has better future performance than those candidate models and which composite model is better. There are some existing methods, which, however, do not work equally well in different situations. So, it is important to apply an appropriate model assessment method to compare model performances.

(4) Model redundancy and uncertainty reduction

It can be good to take advantage of model diversity to improve model performance. However, at the same time information redundancy is also introduced because of dependence among candidate models, which might lead to high model uncertainty and overconfidence. In integrating information, how to detect errors in candidate models constitutes another major concern.

(5) Data uncertainty

Usually, data are contaminated by noises, with unknown probability distributions. If they are not treated properly, the results can be misleading and lead to poor generalized model performance. Therefore, another problem is how to model the noises in data appropriately and thus reduce model bias and uncertainty.

(6) Model locality

Generally, each candidate model has its own operating domain, within which it can work very well but outside of which it may work poorly. For example, some attenuation function in seismology can only be used within some region in terms of the distance from the epicenter. It might be helpful to combine candidate models having different favorite regimes. But, the challenge is how to identify their operating regimes and how to combine them in a consistent way.

(7) Objectivity

Usually, in model combination subjective judgments like expert opinions are involved, which makes the procedure of modeling subject to personal bias and unrepeatable. As pointed out by many authors, experts tend to be overconfident. To overcome this weakness, some objective methods should be preferred. But, the question is how to do that.

By now, lots of work has been done in model selection and model combination. As mentioned earlier, model selection tends to select a single best candidate model, without incorporating the contributions from other peer models. In contrast, model combination is able to somehow integrate information in all candidate models as well as new observations. Therefore, the model combination is preferred. There are many model combination methods, but none of them can address all of the challenges listed above satisfactorily and have their own limitations in terms of accuracy, stability and locality. For example,

(1) Equally weighted average

In this method, all the candidate models make equal contribution to the composite model. Some big errors in candidate models will ruin the composite model. Meanwhile, it can not take advantage of newly collected data.

(2) Weighted average

Because of the dependence among candidate models, a direct weighted average of candidate models based on data suffers from information redundancy, high model dimensionality and large model uncertainty.

(3) Bagging (Bootstrap aggregation), Boosting and Stacking

Bagging fits the same parametric model to bootstrap replicates, thereby stabilizing the model procedure and reducing model uncertainty. It is unable to combine different calibrated models. Similarly, boosting is another machine learning algorithm, which combines poor learners into a better one.

Stacking can be viewed as a modified version of weighted average, whose weights are based on leave-one-out cross-validation. Therefore, it shares the weaknesses with other method.

(4) Bayesian Model Averaging

BMA is a Bayesian method applied in the model space. The first problem in BMA is to assign prior probabilities to candidate models, which is usually done arbitrarily. Another problem of BMA is that it does not take into the dependence among candidate models. A practical difficulty of BMA is that it is hard to implement and computationally expensive.

(5) Bayesian information aggregation

This class of methods also applies Bayesian approach. They try to model the dependence based upon the conditional mean dependence assumption (CMDA), which is not always the case in reality. Besides, they do not incorporate new observations either.

(6) Mixture of Experts (MoE)

MoE follows the philosophy of divide-and-conquer. It trains local models for individual sub-regions and finally mix them. It cannot be directly applied to combine multiple candidate models.

In order to address the challenges mentioned earlier and overcome those limitations of existing approaches, a new feature-based model combination method is proposed. This new approach has (1) higher predictive accuracy, (2) stability, (3) consistency and (4) ability to deal with locality. Basically, it has the following advantages over the others

(i) It is able to aggregate information in all competing models, thereby improving model performance.

(ii) It is able to detect errors in competing models to a degree, thereby reducing model bias;

(iii) It can model dependence among competing models and thus reduce information redundancy;

(iv) It is able to combine different kinds of information, including models and data;

(v) It has robust performance when having different sets of data;

(vi) It is objective, involving no subjective judgment.

In order to overcome those challenges, in this thesis the following new methods are put forward

(1) A candidate model assess procedure is proposed to choose an appropriate group of candidate model for combination;

(2) Principal component analysis and independent component analysis are used to extract features from candidate models and then features are aggregated into a composite model using linear regression;

(3) Fuzzy variable selection method is put forward to perform feature selection and thus reduce information redundancy and help get rid of errors as well;

(4) Exponential power distribution is applied to model noises contained in data as well as deal with non-normality, outliers and heteroscedasticity in partly linear regression;

(5) A new method called Constrained Parametric Weighted Least Squares is proposed to reduce both model structural uncertainty and parameter uncertainty;

(6) An adaptive fuzzy mixture of local models is proposed to combine models in the presence of model locality.

Besides, three kinds of examples are used in this thesis for numerical simulations.

(i)To demonstrate the effectiveness of this method, an artificial example is used to illustrate the model combination procedure.

(ii)Fourier series are used for standard testing of performance of the new method;

(iii)A real case study in seismic risk analysis is carried out to show its application in reality.

Certainly, this approach can have applications in wider areas, for example, in finance.

Let's summarize what have been done in the proceeding chapter and also the contributions we made in each of them.

In chapter 1, at the beginning the thesis problem is formulated and two different kinds of solutions, namely model selection and model combination, are discussed. After reviewing many model performance evaluation methods, either statistical or information-theoretic, most of which follow the principle of parsimony, a generalized flexible framework for designing model performance evaluation approach is brought forward. Then, many existing model combination methods are also briefly reviewed and their individual strengths and weaknesses are pointed out. By comparing model selection and model combination from several perspectives, a conclusion is drawn that model combination is a more effective way to improve model performance than simple model selection. Some objectives of model combination are put forward and a novel feature-based model combination method is proposed to meet these goals. However, to implement this new approach, several important issues have to be addressed. And each issue will constitute the topic of the following chapters.

In chapter 2, the first central problem, that is, the candidate model choice, is solved. Through our analysis, we first conclude that the efficiency of model combination highly depends on the choice of candidate models. Some desirable properties are then proposed to assess a group of candidate models, which include accuracy, diversity, independence as well as completeness. To facilitate the choice with the use of these criteria, some quantitative measures are put forward. Meanwhile, Bayesian method and utility function are employed to aggregate information to obtain an overall evaluation of models. Finally, a stepwise forward candidate model choice procedure is proposed to realize all these criteria in a procedure, which chooses a group of candidate models out of a model pool.

In chapter 3, a new data-guided model combination method by decomposition and aggregation is described in detail. With the aid of influence diagram, we will analyze the dependence among candidate models and apply latent factors to characterize such dependence. After analyzing model structures in this framework, we derive an optimal composite model. Two widely used data analysis tools, namely, Principal Component Analysis (PCA) and Independent Component Analysis (ICA) are applied to commit factor extraction from the class of candidate models. Once factors are ready, they are sorted and aggregated to produce composite models. During the course of factor aggregation, another important issue, namely factor selection, is also touched on. Finally, a numerical study is given to show how this method works.

Chapter 4 is dedicated to solving the problem of component selection in this new method. The classical model selection such as variable selection in multiple linear regression delivers interpretable models, but its instability is also well known. Meanwhile, some shrinkage estimators enjoy the property of stability. To combine these two strengths, we generalize the conventional variable selection by introducing the concept of fuzzy variable selection. After constructing a fuzzy model, we will show that the optimal membership function can be estimated by taking advantage of the generalized ridge regression. By defining the effective model dimensionality, almost all the classical model selection method can be easily extended to our fuzzy scheme. To make the multiple dimensional optimization problem tractable, we also propose a hybrid optimization process, which occupies both the global and local searching capability. Finally, our numerical study shows its advantages over classical method selection methods.

In chapter 5, we deal with another issue of data uncertainty, especially those data contaminated by noises with unknown distributions. It is well known that under Gauss-Markov (GM) assumptions, Ordinary Least Squares (OLS) OLS estimators of regression coefficients are BLUE. But those assumptions cannot always be met in reality. For example, non-Gaussian error, outliers, heteroscedasticiy and incomplete predictor variables can result in the failure of these ideal assumptions. Since these problems lead to the same consequences, that is, non-normal overall error, it is usually difficult to distinguish them from each other only based upon the data. Therefore, a parametric weighted least squares (PWLS) method is proposed as a uniform framework in this chapter to remedy these problems and to obtain a better estimated regression model. Basically, we model the underlying error distribution that departs from normality with a continuously defined parametric exponential power distribution, which includes uniform, Gaussian, Laplace and other sub- and super-Gaussian densities. It is actually unusual to know the error distribution and therefore we have to estimate it from the data, i.e. mainly estimate the shape parameter in the current case. To this end, maximum likelihood estimator (MLE) and alternatively the residual maximum likelihood estimator (REMLE) are put forward. After estimating the error distribution, the maximum likelihood is applied to estimate regression coefficients by a weighted least squares procedure. Furthermore, a significance test based upon the likelihood ratio test is designed to avoid the downside of the performance of this method compared to OLS. Finally, Monte Carlo simulation results show that PWLS outperforms OLS in some cases of interest.

In the proceeding two chapters, we deal with model structure uncertainty resulting from variable selection and parameter and data uncertainty separately. In chapter 6, these two methods are put together to reduce overall model uncertainty, or in other words, apply fuzzy variable selection to the situation where the noise distribution is unknown. A model generally consists of two parts, i.e. model structure and model parameters. Correspondingly, model uncertainty includes both modcl structural uncertainty and model parametric uncertainty. In order to reduce the overall prediction error, we need to reduce both components of model uncertainty. Therefore, in multiple linear regression models, combining a stable model selection method with a more efficient regression method might be a feasible way to improve precision. In this chapter the fuzzy model selection method and the parametric weighted least squares are both generalized to work together to deal with both model structure and data problems simultaneously. A two-stage optimization procedure is designed to ease the numerical realization of the combination. In addition, given an estimate from a model it is usually important to know its associated uncertainty. As pointed out by some authors, model structural uncertainties are sometimes ignored in model uncertainty evaluation, thereby resulting in overstating the precision of a model. In view of Pros and Cons of some existing methods such as Draper [1995] and Buckland [1997], we come up with a new model uncertainty evaluation method, which fits into our situation.

In chapter 7, another important issue in model combination, that is, model locality, is addressed. Since usually a global linear model is unable to catch nonlinearity and characterize local features especially in a complex system, we propose a mixture model of local ICA models to overcome these weaknesses. The basic idea is to split the entire input space into operating domains and the model combination method developed in chapter 3 is applied to build local models for each region. To realize this idea, three steps are required, which include clustering, local modeling and model combination. Some new approaches are developed or existing methods are applied to carry out these steps. Adaptive fuzzy parametric clustering algorithm is put forward to divide the whole input space into operating regimes, local ICA analysis are carried out and the feature-based model combination method is applied to create local ICA models in each individual region, and finally the local ICA models are combined into a mixture model. Correspondingly, a three-stage optimization procedure is designed to optimize the complete objective function, which is actually a hybrid GA algorithm. Our simulation results show that the adaptive fuzzy mixture of local ICA models turns out to be superior to global models.

Bibliography

Abbott, Dean W. (1999). Combining models to improve classifier accuracy and robustness. In *Proceedings of Second International Conference on Information Fusion, Fusion '99*, Volume I, Sunnyvale, CA, 289-295.

Abrahamson, N.A., and Silva, W.J. (1997). Empirical response spectral attenuation relations for shallow crustal earthquakes, *Seism. Res. Letters*, 68(1), 94-127.

Akaike, H. (1973). Information Theory and an Extension of the Maximum Likelihood Principle. 2nd *International Symposium on Information Theory* (B. N. Petrov and F. Czáki, eds), pp. 267-281. Budapest: Akademiai Kiadó.

Akaike, H. (1978). A Bayesian analysis of the minimum AIC procedure. *Annals of the Institute of Statistical Mathematics* 30, 9-14.

Allen, D.M. (1974) The relationship between variable selection and data augmentation and a method of prediction. *Technometrics* 16(1), 125-127.

Amari, S. I. (1985). *Differential Geometrical Methods in Statistics.* Springer-Verlga.

Ashley, R. (1997). *A New Technique for Postsample Model Selection and Validation.* V.P.I. & S.U. Economics Department Working Paper #E97-01.

Attias, H. (1999). Independent factor analysis. Neural Computation, 11(4),803-851.

Avnimelech, R. and Intrator, N. (1999). Boosting Regression Estimators. *Neural Computation*, 11, 499-520.

Bamber, D. and van Santen, J.P.H. (1985). How many parameters can a model have and still be testable? *Journal of Mathematical Psychology,* 29, 443-473.

Bartholomew, David J. (1999). *Latent variable models and factor analysis.* London: Arnold; New York: Oxford University Press.

Bauer, E. and R. Kohavi (1999). An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, 36, 105-142.

Beale, E. M. L. and C. L. Mallows (1959). Scale Mixing of Symmetric Distributions with Zero Means. *The Annals of Mathematical Statistics*, 30 (4), 1145-1151.

Bezdek, J. C. (1981). Pattern Recognition with Fuzzy Objective Function Algoritms, Plenum Press, New York.

Boore, D.M., Joyner, W.B., and Fumal, T.E. (1997). Equations for estimating horizontal response spectra and peak acceleration from western North American earthquakes: A summary of recent work, *Seism. Res. Letters*, 68(1), 128-153.

Bousquet, O. and A. Elisseeff (2002). Stability and Generalization. *Journal of Machine Learning Research* 2, 499-526.

Box, G. E. P. and D. R. Cox (1964). An Analysis of Transformations. *J. R. Statist. Soc.* B, 26, 211-243.

Breiman, L. (1992). Stacked Regression. University of California, Berkeley, Dept. of Statistics, Technical Report 367.

Breiman, L. (1995). Better Subset Regression Using the Nonnegative Garrote. *Technometrics*, 37(4), 373-384.

Breiman, L. (1996). *Arcing Classifiers* (Technical, Report). University of California, Department of Statistics.

Breiman, L. (1996a). Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, 24, 2350 - 2383.

Breiman, L. (1996b). Bagging predictors. Machine Learning, 24(2), 123-140.

Breiman, L. (1996c). *Arcing Classifiers* (Technical, Report). University of California, Department of Statistics.

Breiman, L. (1997). *Prediction Games and Arcing Algorithms.* University of California, Berkeley, Dept. of Statistics, Technical Report 504.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and Regression Trees*. Belmont CA: Wadswörth.

Brook, R. J. and A. M. Tobias (1996). Choosing the Best Model: Level of Detail, Complexity, and Model Performance. *Mathematical and Computer Modeling*, 24(4), 1-14.

Browne, M. (2000). Cross-validation methods. *Journal of Mathematical Psychology*, 44, 108-132.

Buckland, S. T., K. P. Burnham and N. H. Augustin (1997). Model Selection: An Integral Part of Inference. *Biometrics*, 53, 603-618.

Burchard, H. G. (1974). Splines (With Optimal Knots) Are Better. *Applicable Analysis*, 3, 309-319.

Burnham, K. P., and D. R. Anderson (2002). *Model selection and multimodel inference: a practical information-theoretic approach.* Springer-Verlag, New York.

Burns, W. J. and R. Clemen (1993). Covariance Structure Models and Influence Diagrams . *Management Science*, 39, 816-834.

Busemeyer, J. R. and Yi-Min Wang (2000). Model comparisons and model selections based on generalization test methodology, *Journal of Mathematical Psychology*, 44, 171-189.

Campbell, K.W. (1997). Empirical near-source attenuation relations for horizontal and vertical components of peak ground acceleration, peak ground velocity, and pseudo-absolute acceleration response spectra. *Seism. Res. Letters*, 68(1), 154-179.

Carroll, R.J. and Ruppert D. (1988) *Transformation and Weighting in Regression*, Chapman and Hall, New York.

Carroll, R.J., C.F.J. Wu and D. Ruppert (1988). The effect of estimating weights in generalized least squares. *J. of the Amer. Statist. Assoc.*, 83, 1045-1054.

Chan, ai-Wan and Siu-Ming Cha (2001). Selection of Independent Factor Model in Finance. In proceedings of *3rd International Conference on Independent Component Analysis and blind Signal Separation,* December 9-12, 2001 -- San Diego, California, USA.

Chen, Jiahua and Jun Shao (1993). Iterative Weighted Least Squares Estimators. *The Annals of Statistics*, 21(2), 1071-1092.

Cheung, Yiu-ming and Lei Xu (2001). Independent component ordering in ICA time series analysis. *Neurocomputing*, 41, 145-152.

Chow,Gregory C. (1981). A comparison of the information and posterior probability criteria for model selection. *Journal of Econometrics*, 16, 21-33.

Christensen, R.(2001). *Linear models for multivariate, time series, and spatial data.* New York : Springer.

Clemen, R. T. (1986). Combing Overlapped Information. *Management Science*, 33(3), 373-379.

Clemen, R. T. and K. C. Lichtendahl (2002). Debiasing Expert Overconfidence: A Bayesian Calibration Model. *PSAM6.*

Clemen, R. T. and R. L. Winkler (1985). Limits for the Precision and Value of information from Dependent Sources. *Operations Research*, 33, 427-442.

Clemen, R. T. and Winkler, R. L. (1999). Combining Probability Distributions From Experts in Risk Analysis." *Risk Analysis*, 19, 2, 187–203.

Clemen, R.T. and R.L. Winkler(1986).Combining Economic Forecasts.*J. Business and Economic Statistics*, 4, 39-46.

Clemen, R.T. and R.L. Winkler(1993).Aggregating Point Estimates: A Flexible Modeling Approach. *Management Science*, 39:4,501-515.

Cleveland, W. S. and S. J. Devlin (1988). Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting, *Journal of the American Statistical Association*, 83, 596-610.

Condorcet, N. C. de (1785). *Essai sur l' application de l' analyse à la probabilité des decisions rendues àprobabilité des voix.* Imprimerie Royale, Paris.

Cook, R. Dennis, Sandford Weisberg (1982). *Residual and Influence in Regression.* Chapman & Hall, New York.

Copas, J. B. (1983). Regression, Prediction, and Shrinkage (with discussion). *J. R. Statist. Soc. B*, 45(3), 311-354.

Cramér, H. (1946). Mathematical Methods of Statistics. Princeton University Press, Princeton, NJ.

Cressie, N. and S.N.Lahiri (1993). The asymptotic distribution of REML estimators. Journal of Multivariate Analysis 45,217 –233.

Dall'Aglio, G., S. Kotz and G. Salinetti (1991). *Advances in Probability Distribution with Given Marginals: Beyond the Copulas*. Kluwer, Dordrecht, Netherlands.

Davidson, Russell and James G. MacKinnon (1993). *Estimation and Inference in Econometrics*. New York: Oxford University Press.

Dawid, A. P. (1984). Present position and potential developments: some personal views. Statistical theory. The prequential approach (with discussion). *J. R. Statistical Soc.* A, 147, 178-292.

Deco G., and D. Obradovic (1995). Linear Redundancy Reduction Learning. *Neural Networks*, 8(5): 751-755.

Diaconis, P. and D. Freedman (1984). Asymptotics of Graphical Projection Pursuit. *Ann. Statist.* , 12, 793-815.

Diday, Edwin (1974). Recent Progress in Distance and Similarity Measures in Pattern Recognition. *Second International Joint Conference on Pattern Recognition*, 534-539.

Diertterich, T. G. (2000). Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, 1-15. Springer-Verlag.

Draper, D. (1995). Assessment and Propagation of Model Uncertainty. *J. R. Statist. Soc.* B, 57(1): 45-97.

Drucker, H. (1997). Improving Regressors Using Boosting Techniques. *Proceedings of the Fourteenth International Conference on Machine Learning*, 107-115.

Dudley, R.M.(1989). *Real Analysis and Probability*, Wadsworth & Brooks/Cole, Pacific Grove, California.

Dunn, J. C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in detecting Compact Well-Separated Clusters, *Journal of Cybernetics*, 3, 32-57.

Efron, B. (1983). Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. *Journal of the American Statistical Association*, 78(382): 316-331.

Efron, B. (1986). How Biased is the Apparent Error Rate of a Prediction Rule? *J. of the American Statistical Association*, 81, 461-470.

Efron, B. and G. Gong (1983). A Leisurely Look at the Bootstrap, the Jackknife and Cross-validation. *Amer. Statist.*, 37, 36-48.

Efron, B. and R. J.Tibshirani (1993). *An Introduction to the Bootstrap*. Monographs on Statistics and Applied Probability, 57. Chapman & Hall, New York.

Efron, B. and R. Tibshirani (1997). Improvements on Cross-Validation: The .632+ Bootstrap Method. *Journal of the American Statistical Association*, 92(438):548-560.

Eilers, P.H.C. (1991). Penalized Regression in Action: Estimating Pollution Roses from Daily Averages. *Environmetrics*, 2, 25-48.

Ellery, Eells (1991). *Probabilistic Causality*. Cambridge: Cambridge University Press.

Everitt, B. S. (1984). *An Introduction to Latent Variable Models*. Chapman and Hall, London.

Fan, J. (1995). Local Modelling. In *Encyclopidea of Statistical Science*.

Ferrell, W. R. (1985) Combining individual judgments. In G. Wright (Ed.), *Behavioral Decision Making* (pp. 111–145). New York: Plenum.

Figlewski, S. and T. Urich (1983). Optimal Aggregation of Money Supply Forecasts: Accuracy, Profitability and Market Efficiency. *J. Finance*, 28, 695-710.

Forster, Malcolm R. (2000). Key Concepts in Model Selection: Performance and Generalizability. *Journal of mathematical psychology*, 44,205-231.

Forster, Malcolm R.(1984). *Probabilistic Causality and the Foundation of Modern Science*. Ph.D. Thesis, University of Western Ontario.

Freedman, D. A. and Peters, S. C. (1984). Bootstrapping a Regression Equation: Some Empirical Results. Journal of the American Statistical Association, 79, 97-106.

French, S. (1981). Consensus of Opinion. *European J. Opnl. Res.* 7,332-340.

Freund, Y and R. Schapire (1997). A decision-theoretic Generalization of Online Learning and an Application to boosting. *Journal of Computer and System Sciences*, 55(1): 119-139.

Friedman, J. (1991). Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, 19, 1-141.

Friedman, J. H. (1999). *Greedy Function Approximation: a Gradient Boosting Machine.* Technical report, Dept. of Statistics, Stanford University.

Fuller, W. A. and Rao, J. N. K. (1978). Estimation for A Linear Regression Model with Unknown Diagonal Covariance Matrix. *Ann. Statist.* 6, 1149-1158.

Garcia, S. (1999). *Experimental Design Optimization and Thermophysical Parameter Estimation of Composite Materials Using Genetic Algorithms.* Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Geisser, S. and W. F. Eddy (1979). A Predictive Approach to Model Selection. *Journal of the American Statistical Association*, 74(365), 153-160.

Geman, S. and D. Geman (1984). Stochastic relaxation, Gibbs distribution and Bayesian restoration of images. *IEE Transactions on Pattern Analysis and Machine Intelligence.* 6, 721–741.

Geman, S., E. Bienenstock and R. Doursat (1992). Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4, 1-58.

Ghahramani,Z. and G. E. Hinton (1997). Hierarchical non-linear factor analysis and topographic maps. *Advances in Neural Information Processing Systems 10, NIPS*97*, 486-492.

Gilks, W.R. S. Richardson, and D.J. Spiegelhalter (1998). *Markov chain Monte Carlo in practice.* Boca Raton, Fla.: Chapman & Hall.

Gill, P. E., W. Murray and M. H. Wright (1986). *Practical Optimization.* Academic Press, 1986.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, New York.

Golub, G.H., M. Heath, and G.Wahba (1979). Generalised cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215-223.

Green, P. J.(1984). Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and Some Robust and Resistant Alternatives. *J. Royal Stat. Soc. Ser.* B, 46(2):149-192.

Grunwald, P. (2000). Model selection based on minimum description length. *Journal of Mathematical Psychology, 44*, 133-170.

Hannan, E. J. and Quinn, B. (1979). The determination of the order of an autoregression, *Journal of the Royal Statistical Society, Series B* 41, 190–191.

Hansen, L. and P. Salamon (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, 993-1001.

Harrington, S., R. Zhang, P. H. Poole, F. Sciortino, and H. E. Stanley (1997). Liquid-Liquid Phase Transition: Evidence from Simulations. *Physical Review Letters*, 78(12), 2409-2412.

Hartley, H. O. and J. N. K. Rao (1967). Maximum Likelihood Estimation for the Mixed Analysis of Variance Model. Biometrika, 54, 93-108.

Harville, D. A. (1974). Bayesian Inference for Variance Components Using Only Error Constrasts. *Biometrika*, 61, 383-385.

Harville, D. A. (1977). Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems. *Journal of the American Statistical Association*, 72, 320-338.

Hastie, T. J. and R. J. Tibshirani (1990). *Generalized Additive Models.* Chapman and Hall, New York.

Hessner, J., and R. Manner (1991). Choosing optimal mutation rate. In *Proceedings of the first workshop on parallel problem solving from nature*, edited by P. Schwefel and R. Manner, 23-31. Berlin: Springer-Verlag.

Hjorth, J.S.U. (1994), Computer Intensive Statistical Methods: Validation, Model Selection, and Bootstrap, London: Chapman & Hall.

Hoerl, A.E. and R.W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(3): 55-67.

Hoeting, J., D. Madigan, A. Raftery and C.T. Volinsky(1999). Bayesian model averaging: A tutorial (with discussion), *Statistical Science* 14(4) 382-417.

Hoeting, Jennifer, David Madigan, Adrian Raftery and Chris Volinsky (1999). Bayesian Model Averaging. *Statistical Science* 14, 382-401.

Hogarth, R. M. (1986). Generalization in decision research: The role of formal models. *IEEE Transactions on Systems, Man, and Cybernetics*, 16, 439–449.

Holland, J. M. (1975). *Adaptation in Nature and Artificial Systems.* Ann Arbor, MI: The University of Michigan Press.

Honda, K., H. Ichihashi, M. Ohue and K. Kitaguchi (2000). Extraction of Local Independent Components Using Fuzzy Clustering. Proc. of 6th International Conference on Soft Computing (IIZUKA2000).

Hooper, P. M. (1993). Iterative weighted least squares estimation in heteroscedastic linear models. *Journal of the American Statistical Association*, 88, 179-184.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24, 417-441.

Howard, R. and J. Matheson (1984). Influence Diagrams. In R. Howard and J. Matheson (Eds.), *The Principles and Applications of Decision Analysis, SDG Systems*, Menlo Park, CA, 719-762.

Howard, R.(1989). Knowledge Maps. *Management Science*,35, 903-922.

Huber, P. J. (1964). Robust Estimation of a Location Parameter. *Annals of Mathematical Statistics*, 35, 73-101.

Hurvich, C. M., & Tsai, C-L. (1989). Regression and time series model selection in small samples. *Biometrika, 76*, 297-307.

Hyvärinen, A. and Erkki Oja (1997). A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7): 1483-1492.

Hyvärinen, A. and Erkki Oja (2000). Independent Component Analysis: Algorithms and Applications. *Neural Networks*, 13(4-5): 411-430.

Hyvärinen, A. and P. Pajunen (1999). Nonlinear Independent Component Analysis: Existence and Uniqueness Results. *Neural Network,* 12 (2), 209-219.

Hyvärinen, A.(1998). New approximation of differential entropy for independent component analysis and project pursuit. In *Advance in Neural Information Processing Systems*, 19, 273-279.

Hyvärinen, A.(1999). Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks* 10(3), 626-634.

Hyvärinen, A.(1999). Survey on independent component analysis. *Neural Computing Surveys*, 2:94-128.

Idriss, I. M. (1995). An overview of earthquake ground motion pertinent to seismic zonation, Presented at 5th International Conference on Seismic Zonation, 17-19, October, Nice, France.

Inoue, K. (1999). Asymptotic Improvement Of the Graybill-Deal Estimator. Comm. Statist. Theory Methods, 28(2), 388-407.

Inoue, K. (2003). Iterative weighted least-squares estimates in a heteroscedastic linear regression model. *Journal of Statistical Planning and Inference* 110, 133-146.

Jacobs, R. A. and M. I. Jordan, S. J. Nowlan, and G. E. Hinton (1991). Adaptive mixtures of local experts. *Neural Comput.*, 3(1) ,79–87.

James, William (1907). *Pragmatism: A New Name for Some Old Ways of Thinking.* New York: Longmans, Green.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proc. Roy. Soc.* A, 186, 453–461.

Johansen, T. A. and B. A. Foss (1997). Operating regime based process modeling and identification. Computers and Chemical Engineering, 21, 159-176.

Johnson, S. C. (1967). Hierarchical Clustering Schemes. *Psychometrika*, 2, 241-254.

Jolliffe, I. T.(1986). *Principal Component Analysis.* New York: Springer-Verlag.

Jones, M.C. and R. Sibson (1986). What is projection pursuit? *J. of the Royal Statistical Society, ser. A*, 150: 1-36.

Jordan, M. I. and R. A. Jacobs (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Comput.*, 6(2), 181–214.

Joyner, W.B. and D.M. Boore (1993). Method for regression analysis of strong motion data, *Bull. Seism. Soc. Am.*, 83, 469-487.

Jupp, D. L. B.(1978). Approximation to data by splines with free knots. *SIAM Journal on Numerical Analysis*, 15(2), 328-343.

Jutten, C. and J. Herault (1991). Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24: 1-10.

Kailath, T.(1967). The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communications Technology*, 15(1),52-60.

Kambhatla, N. and T. Leen (1997). Dimension Reduction by Local Principal Component Analysis, Neural Computation, 9, 1493-1516.

Karakoulas, G. and J. Shawe-Taylor (1999). Towards a Strategy for Boosting Regressors. In *advances in Large Margin Classifiers*, Smola, Bartlett, Schölkopf and Schuurmans (Eds.).

Karhunen, J. and E. Oja, L. Wang, R. Vigário, and J. Joutsensalo (1997). A class of neural networks for independent component analysis. *IEEE Trans. On Neural Networks*, 8(3): 486-504.

Karhunen, J. and S. Malaroiu (1999a). Locally Linear Independent Component Analysis. In *Proc. of the Int. Joint Conf. On Neural Networks (IJCNN'99).*

Karhunen, J. and S. Malaroiu (1999b). Local Independent Component Analysis Using Clustering. In *Proc. First Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, 43-48.

Kass, Robert E. and Adrian E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association*, 90 (430): 773-795.

Krogh, A. and J. Vedelsby (1995). Neural network ensembles, cross-validation, and active learning. In Tesauro, G., Touretzky, D. and Leen, T. editors, *Advances in Neural Information Processing Systems (Volume 7)*, Cambridge, MA, MIT Press.

Kulkarni, Lugosi and Venkatesh (1998). Learning Pattern Classification - A Survey. *IEEE Tran. IT*, 44, 2178-2206.

Kullback, S. (1959).*Information Theory and Statistics.* Wiley, New York.

Kullback, S. and R. Leibler (1951). On information and sufficiency. *Ann. Math. Statist.*, 22,79-86.

Kuncheva, L., Bezdek, J.C., Duin, R.P.W. (2001). Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34, 299-314.

Kuncheva, L.I. and C.J. Whitaker (2003). Measures of diversity in classifier ensembles. *Machine Learning*, 51, 181-207.

Lam, L. and C.Y. Suen (1997). Application of Majority Voting to Pattern Recognition : an Analysis of Its Behavior and Performance. *IEEE Trans.Systems Man Cybern.*Part A, 27 (5), 553-568.

Lee, V.W. and M.D. Trifunac (1995). Frequency Dependent Attenuation Function and Fourier Amplitude Spectra of Strong Earthquake Ground Motion in California, Dept. of Civil Engineering *Report No CE 95-03*, University of Southern California, Los Angeles.

Li, Ker-Chau (1987). Asymptotic Optimality for C_p, C_L, Cross-Validation and Generalized Cross-Validation: Discrete Index Set. *The Annals of Statistics*, 15(3), 958-975.

Li, M. and P.M.B. Vitanyi (1997). *An Introduction to Kolmogorov Complexity and its Applications*, Springer-Verlag, New York.

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, **37**,145 -151.

Lindley, D. V. (1983). Reconciliation Probability Distributions. *Opns. Res.*, 31,866-880.

Lindley, D. V. and A. F. M. Smith (1972). Bayes Estimates for the Linear Model. *Journal of Royal Statistical Society*, 67, 1-19.

Luo, Z. and Wahba, G. (1997). Hybrid Adaptive Splines. *J. Amer. Statist. Assoc.*, 92(437), 107-115.

MacKay, D.J.C.(1992). Bayesian interpolation. *Neural Computation*, 4(3):415-447.

MacQueen, J. B. (1967). Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1, 281-297.

Madigan, D. and Raftery, A. E. (1994). Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam's Window. *J. Amer. Statist. Assoc.*, 89, 1535-1546.

Makridakis, S. and R. L. Winkler (1983). Averages of forecasts: Some empirical results. *Management Science*, 29(9), 987-996.

Mallows, C.(1973). Some comments on C_p. *Technometrics*, 15:661-675.

Mason, L., J. Baxter, P. Bartlett and M. Frean (1999). Boosting Algorithms as Gradient Descent in Function Space. *NIPS* 11.

McCullagh, P. and J. A. Nelder (1983). *Generalized Linear Models.* Chapman and Hall, New York.

McLachlan, G. J. and K. E. Basford (1988). *Mixture Models: Inference and Application to Clustering.* New York: Marcel Dekker.

McLachlan, G. J. and Peel, D. (2000). *Finite Mixture Models.* Wiley, New York.

Miller, A.J. (1984). Selection of subsets of regression variables (with Discussion). *Journal of the Royal Statistical Society A*, 147, 389-425.

Moody, J.E.(1992). The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In J.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems* 4, pages 847-854. Morgan Kaufmann, San Mateo, CA.

Morris, P. (1997). Combining Expert Judgments: A Bayesian Approach. *Management Science*, 23, 679-693.

Morris, P. A. (1974). Decision Analysis Expert Use. *Management Science,*20, 1233-1241.

Morris, P. A. (1977). Combining Expert Judgments: A Bayesian Approach. *Management Science*, 23,679-693.

Mosleh, A. and G. Apostolakis (1986). The Assessment of Probability Distribution from Expert Opinions with an Application to Seismic Fragility Curves. *Risk Analysis*, 6(4), 447-461.

Mukherjee, S., P. Niyogi, T. Poggio and R. Rifkin (2002). Statistical Learning: Stability is Sufficient for Generalization and Necessary and Sufficient for Consistency of Empirical Risk Minimization. *AI/CBCL Memo.*

Murata, N., Yoshizawa, S., and Amari, S. (1994). Network information criterion Determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks*, 5(6), 865-872.

Murray-Smith, R. and T. A. Johansen (Editors)(1997). *Multiple Model Approaches to Nonlinear Modeling and Control*, Taylor and Francis, London, UK.

Murray-Smith, R. and T. A. Johansen(1995). Local Learning in Local Model Networks, *Proc. IEE Int. Conf. Artificial Neural Networks*, Cambridge, UK, 40-46.

Myung, I. J. and M. A. Pitt (1997). Applying occam's razor in modeling cognition: A Bayesian approach. *Psychonomic Review and Bulletin*, 4, 79-95.

Myung, I. J., Balasubramanian, V., & Pitt, M. A. (1999) Counting probability distributions: Differential geometry and model selection. Proceedings of *the National Academy of Sciences* USA, 97, 11170-11175.

Nelder, J.A. and R.W.M. Wedderburn (1972). Generalized Linear Models. *Journal of the Royal Statistical Society, Series A* 135, 370-384.

O'Sullivan, F. (1986). A Statistical perspective on ill-posed inverse problems (with discussion). Statistical Science. 1, 505-527.

Opitz, D. and J. Shavlik (1996). Generating accurate and diverse members of a neural network ensemble, in D. Touretzky, M. Mozer and M. Hasselmo, eds., Advances in Neural Information Processing Systems 8, 535-541, MIT Press.

Orr, M. J. L.(1996). *Introduction to radial basis function networks.* Technical report, Center for Cognitive Science, University of Edinburgh.

Orr, M.J.L.(1995). Local Smoothing of Radial Basis Function Networks (long version).In *International Symposium on Artificial Neural Networks*, Hsinchu, Taiwan.

Papoulis, A. (1991). *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, New York.

Parra, L., C. Spence and P. Sajda (2001). High-order Statistical Properties Arising from the Non-stationarity of Natural Signals. In T. Leen, T. Dietterich, and V. Tresp ed. *Advances in Neural Information Processing Systems 13,* MIT Press, Cambridge, MA, 786-792.

Patterson, H. D. and Robin Thompson (1971). Recovery of Inter-Block Information when Block Sizes are Unequal. *Biometrika*, 58, 545-554.

Perrone, M. P. and Leon N Cooper (1993). When networks disagree: Ensemble method for neural networks. In R. J. Mammone, editor, *Neural Networks for Speech and Image processing.* 126-142, Chapman-Hall.

Pittman, J. (2002). Adaptive Splines and Genetic Algorithms. *Journal of Computational and Graphical Statistics*, 11, **3**, 1-24.

Rao, C. R. (1945). Information and accuracy attainable in the estimation of statistical parameters, *Bull. Calcutta Math.* Soc. 37, 81-91.

Reinsch, H. (1967). Smoothing by spline functions, *Numerische Mathematik* 10: 177-83.

Richardson, A.M. and A. H. Welsh (1994). Asymptotic properties of restricted maximum likelihood (REML) estimates for hierarchical mixed linear models. *The Australian Journal of Statistics* 36,31 –43.A378.

Rissanen, J. (1978) Modeling by the shortest data description. *Automatica*, 14, 465-471.

Rissanen, J. (1996). Fisher Information and Stochastic Complexity. *IEEE transactions on information theory*, 42, 40-47.

Sadigh, K., Chang, C.-Y., Egan, J.A., Makdisi, F., and Youngs, R.R. (1997). Attenuation relations for shallow crustal earthquakes based on California strong motion data, *Seism. Res. Letters*, 68(1), 180-189.

Schick, A. (1996). Weighted Least Squares Estimates in Partly Linear Regression Models. *Statistics & Probability Letters*, 27, 281-287..

Schmid, J. and Leiman, J.M. (1957). The development of hierarchical factor solutions. Psychometrika, 22, 53-61.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics* 6, 461-464.

Shachter, R. (1986). Evaluating Influence Diagrams. *Oper. Res.*, 34, 871-882.

Shachter, R. (1988). Probabilistic Inference and Influence Diagrams. *Oper. Res.*,36, 589-604.

Shibata, R.(1981). An optimal selection of regression variables. *Biometrika*, 68(1), 45-54.

Shibata, Ritei (1976). Selection of the order of an autoregressive model by Akaike's information criterion. *Biometrika* 63 117-126.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to nonparametric regression curve fitting (with discussion), *Journal of the Royal Statistical Society*, Series B 47, 1-52.

Sklar, A. (1959). Fonctions de rèpartition à n dimensions et leurs marges, *Publ. Inst. Statist. Univ. Paris* 8, 229-231.

Söderman, U., Top, J. & Strömberg, J.-E. (1993), The conceptual side of mode switching. In *Proc. IEEE Conf. Systems, Man, and Cybernetics, Le Touquet, France*, 245-250.

Steidl, J. H. and Y. Lee (2000). The SCEC Phase III Strong-Motion DataBase. Bulletin of the Seismological Society of America, 90(6B), S113-S135.

Stone, M. (1977). An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1), 44-47.

Stone,M. (1974). Cross-validation choice and assessment of statistical procedures, *Journal Royal of Statistical Society*, 36,111-147

Takagi, T. and M. Sugeno (1985). Fuzzy Identification of Systems and Its Application to Modeling and Control. *IEEE Trans. on Systems, Man and Cybernetics*, 15, 116-132.
Takeuchi, K. (1976). Distribution of informational statistics and a criterion of model fitting. *Suri-Kagaku (Mathematical Sciences)* 153,12-18. (in Japanese)

Taylor, J. M. G. and Arminda L. Siqueira (1996). The Cost of Adding Parameters to a Model. *J. R. Statist. Soc.* B, 58(3), 593-607.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the LASSO. *J. R. Statist. Soc. B*, 58(1), 267-288.

Tibshirani, R. (1997), A Proposal for Variable Selection in the Cox Model. *Statistics in Medicine*, 16, 385-395.

Tierney, L., and Kadane, J.B. (1986) Accurate approximations for posterior moments and marginal densities. *J. Amer. Statist. Ass.*, 81, 82-86.

Tikhonov, A.N. (1963). Solution of incorrectly formulated problems and regularization method. *Soviet math. Dokl.* 4, 1036-1038.

Tikhonov, A.N. and V.Y. Arsenin (1977). *Solutions of Ill-Posed Problems.* Winston, Washington.

Ting, K. M. and B. T. Low (1997). Model Combination in the Multiple-Data-Batches Scenario. In Proceedings of the Ninth European Conference on Machine Learning (ECML-97). Springer-Verlag, 250-265, Prague, Czech Republic.

Verbyla, A. P. (1993). Modelling Variance Heterogeneity: Residual Maximum Likelihood and Diagnostics. *Journal of the Royal Statistical Society*, Series B 55, 493–508.

Volinsky, C. T., D. Madigan, A. E. Raftery, and R. A. Kronmal (1997). Bayesian Model Averaging in proportional hazard models: Assessing the risk of a stroke. *Applied Statistics*, 46(3), 443-448.

Wasserman, L. (2000). Bayesian model selection and model averaging. *Journal of Mathematical Psychol-ogy*, 44, 92-107.

Weiss, S.M. & Kulikowski, C.A. (1991). *Computer Systems That Learn*, Morgan Kaufmann.

White, H. (2000). A Reality Check for Data Snooping, *Econometrica*, 68, 1097-1126.

Wilks, S. S. (1963). *Mathematical Statistics.* New York : Wiley.

Winkler, R. L. (1981). Combining Probability Distributions from Dependent Information Sources. *Mgmt. Sci.* 27, 479-488.

Wolpert, D. (1992). Stacked Generalization. *Neural Networks*, 5, 241-259.

Wolpert, D. and W. Macready (1996). *Combining stacking with bagging to improve a learning algorithm*, Technical Report, Santa Fe Institute, Santa Fe.

Wolpert, D. H. and W. G. Macready (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1): 67-82.

Woodroofe, M. (1982) On model selection and the arcsine laws. *Ann. Statist.* 10, 1182-1194.

Zellner, A. (1994). Bayesian Method of Moments (BMOM) Analysis of Mean and Regression Models. in Lee, J., Johnson, W. and Zellner, A. (eds.) Prediction and Modeling Honoring Seymour Geisser, New York: Springer Verlag, 61-74.

Zellner, A. (1997). The Bayesian Method of Moments (BMOM): Theory and Applications. Fomby, T. and Hill, R., eds., *Advances in Econometrics*, 12, Greenwich, CT: Jai Press, 85-105.

Zenko, B., L. Todorovski, and S. Dzeroski (2001). A comparison of stacking with meta decision trees to bagging, boosting, and stacking with other methods. In *Proc. IEEE International Conference on Data Mining*, pages 669-670. IEEE Computer Society, Los Alamitos.

Zucchini, W. (2000). An introduction to model selection. *Journal of Mathematical Psychology*, 44, 41-61.