Prediction Interval Modeling Using Gaussian Process Quantile Regression

By

Joan Aguilar Fargas

Engineering in Electronics
Universitat Ramon Llull (2008)
Technical Engineering in Electronic Systems
Universitat Ramon Llull (2006)

SUBMITTED TO THE SYSTEM DESIGN AND MANAGEMENT PROGRAM
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN ENGINEERING AND MANAGEMENT
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2014

Signature of Author: Signature redacted
Joan Aguilar Fargas
System Design and Management Program
June 2014

Certified by: Signature redacted
Moshe E. Ben-Akiva
Thesis Supervisor
Edmund K. Turner Professor of Civil and Environmental Engineering

Certified by: Signature redacted
Francisco C. Pereira
Thesis Supervisor
Research Affiliate, Department of Civil and Environmental Engineering

Accepted by: Signature redacted
Patrick Hale
Director
System Design and Management Program

This page has been intentionally left blank

Prediction Interval Modeling Using Gaussian Process Quantile Regression

By

Joan Aguilar Fargas

Submitted to the System Design and Management Program
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering and Management

Abstract

In this thesis a methodology to construct prediction intervals for a generic *black-box* point forecast model is presented. The prediction intervals are learned from the forecasts of the black-box model and the actual realizations of the forecasted variable by using quantile regression on the observed prediction error distribution, the distribution of which is not assumed.

An independent *meta-model* that runs in parallel to the original point forecast model is responsible for learning and generating the prediction intervals, thus requiring no modification to the original setup. This meta-model uses both the inputs and output of the black-box model and calculates a lower and an upper bound for each of its forecasts with the goal that a predefined percentage of future realizations are included in the interval formed by both bounds.

Metrics for the performance of the meta-model are established, paying special attention to the conditional interval coverage with respect to both time and the inputs.

A series of cases studies are performed to determine the capabilities of this approach and to compare it to standard practices.

Thesis supervisor: Moshe E. Ben-Akiva

Title: Edmund K. Turner Professor of Civil and Environmental Engineering

Thesis supervisor: Francisco C. Pereira

Title: Research Affiliate, Department of Civil and Environmental Engineering

Acknowledgements

I would like to thank my thesis advisors Professor Moshe Ben-Akiva and Francisco Pereira for their guidance, support and help during my stay at the Intelligent Transportation Systems Lab and especially the months after.

I would also like to thank Katie Rosa, Eunice Kim and Jorge Santos without whom the MIT ITS Lab wouldn't be the same.

For his help with some of the case studies presented in this thesis, I would like to thank Filipe Rodrigues from the Universidade de Coimbra.

For their support and for being always available I am very thankful to the MIT's System Design and Management Program staff, especially Pat Hale who is always looking after all the SDM community.

Last, but not least, my most sincere appreciation for all the friends, classmates and colleagues that I have met and had the opportunity to spend time with during my stay at MIT.

Table of contents

List of figures

List of tables

1 Introduction

1.1 Problem statement

Forecasts are present in our everyday lives. From the moment we wake up and check the predicted maximum temperature for the day, the probability of rain or the time it will take to get to the office we are using forecasts. In economics, currency exchange rates or stock price future values are constantly evaluated. In marketing and supply chain management, product sales forecasts are vital and the energy companies base their infrastructure expansion on electricity consumption forecasts, just to name a few.

Of all the different forms of predictions, point forecasts are perhaps the most commonly used. They are single numeric values that represent the expected future value of a given numeric variable, and they are very useful because they are intelligible and easy to communicate. Because they are so natural, many forecasting techniques have been developed to provide point forecasts. Also, they are easy to treat mathematically.

One of the problems of point forecasts, however, is that they provide a limited amount of information. Unless the forecaster knows what the future will be (which would mean it is not a forecasting after all) the probability of an exact point prediction becoming a reality is not one. In fact, for continuous variables this probability is zero. This is because a forecast is a best-effort attempt to predict the future, but that future is nevertheless unknown until it realizes.

The use of alternative forecast methods that can provide more information about the *uncertainty* associated to a prediction is becoming more popular, mainly thanks to advances in computation and the needs of the financial industry. The most popular approaches are the use of prediction intervals, or interval forecasts, and density forecasts. Both are an attempt to provide more information about the uncertainty associated to the future value of the variable that is forecasted.

Nevertheless, point forecast models are used everywhere and most of the time generated by models with a solid theoretical and practical background. Replacing them is most of the times not feasible or not straightforward, and perhaps not even desirable.

The focus of this thesis is precisely the generation of prediction intervals to be used as a complement to a point forecast model in a way that this does not require any modification. Furthermore, the goal is that the generated prediction intervals should be accurate not only on average but also under any circumstances.

1.2 Contributions

In this thesis a methodology to construct prediction intervals for a generic black-box point forecast model is presented. The prediction intervals are learned by an independent meta-model from the forecasts of the black-box model and the actual realizations of the forecasted variable by using quantile regression on the observed prediction error distribution, the distribution of which is not assumed.

This methodology differs from most of the existing approaches, which focus on specific models, time-series or error distributions with well-known properties. In comparison to similar approaches that use quantile regression, this methodology is more general in both its applications and its use of quantile regression.

Last, an argument in favor of prediction interval evaluation metrics that are not widespread is made, and other evaluation methods are proposed.

1.3 Thesis outline

The rest of this thesis is organized as follows. Chapter 2 gives a review on the literature and provides a background on the concepts that we'll be used in later chapters. Chapter 3 introduces a methodology for modeling prediction intervals using Gaussian process quantile regression. Chapter 4 applies this methodology to some practical cases and analyses the results. Finally, chapter 5 contains the conclusions and possible future research.

2 Background and literature review

This chapter presents a review of past research on prediction intervals, forecast evaluation methods and quantile regression with the objective to provide the necessary background for the rest of chapters in this thesis.

2.1 Terminology

There does not seem to be a complete agreement on what the difference between forecasting and predicting is. Even though "forecast" tends to be used more frequently in certain fields like meteorology and finance, both terms are used indistinctly throughout the literature. A review of different English dictionaries does not provide a definitive answer either. Therefore, given a lack of consensus and in order to improve the readability of the text both terms will be used interchangeably in this thesis with the exemption of "point forecast" and "forecast uncertainty", for which the alternative is rarely found in the literature.

According to the Cambridge dictionary of English, a prediction is "a statement about what you think will happen in the future" whereas a forecast is "a statement of what is judged likely to happen in the future, especially in connection with a particular situation, or the expected weather conditions". In the context of this thesis, more specifically, a prediction or forecast is an estimation of the future value of a target variable generated by a model based on some past information, where this target variable is expressed either numerically or categorically.

The "realization" or "observation" of the target variable for a specific time in the future is the actual value of the variable when that time arrives.

The "innovation" or "observed prediction error" is the difference between the realization of the target variable and a prediction made a priori.

2.2 Forecast uncertainty

Because a forecast is an attempt to estimate what the value of a variable will be in the future and that future is not known a priori, any forecast comes inevitably associated with some degree of uncertainty. In other words, if we were totally certain that a forecast would become true there wouldn't be any need for a forecast in the first place.

The importance of quantifying forecast uncertainty has been long acknowledged (see [1] just as an example), but the literature has historically avoided this topic (see [2] for a review). According to [2] there seem to be several reasons for that: point forecasts are easy to understand, exact analytical expressions that describe uncertainty cannot be obtained for the vast majority of prediction models, and empirically-based uncertainty calculations have been historically too computationally expensive, amongst others.

2.2.1 Prediction intervals

Nevertheless, research on forecast uncertainty has intensified during the last two decades, especially on interval forecasts, which are by far the most common way of representation (see [3] for reviews). An interval forecast, or prediction interval (abbreviated *PI*), is a range of possible outcomes that includes the realization of the predicted variable with a certain probability. Assuming we know the conditional probability distribution of the predicted value, a prediction interval is characterized by two of its (generally extreme) quantiles. The wider a prediction interval is, therefore, the higher the forecast uncertainty, and vice versa.

For a given distribution with finite variance, it can be demonstrated (Wilks, [4]) that an interval can be found such that the amount of probability within the limits is bounded based on Chebyshev's inequality. Given a random variable Y with mean μ and variance σ^2:

$$P[|(Y-\mu)/\sigma| \geq \epsilon] \leq 1/\epsilon^2$$

This result implies that one can in general find an appropriate prediction interval provided some information about the conditional target variable probability distribution is known. For practical reasons, though, a normal distribution is usually assumed (see [3], for example), which makes generating more accurate (narrower) prediction intervals compared to Wilks' straightforward:

$$PI = \mu \pm z\sigma$$

where z is called the standard score and can be easily found by using mathematical software packages or alternatively looking up pre-calculated tables in statistics books (for example [5] page 472). For the 95% prediction interval, a commonly used PI, the standard score is 1.96.

It is important to distinguish the concept of prediction intervals from that of confidence intervals. This is a topic that has been thoroughly discussed in the literature (e.g.[6], [7]) and that still nowadays is sometimes misunderstood. As an example, a 90% confidence interval is expected to contain the population mean in at least 90% of repeated sample experiments while a 90% prediction interval should contain the realization of the predicted value at least 90% of the times.

2.2.2 Heteroscedasticity

A random variable Y is said to be heteroscedastic if its conditional variance given a vector of random variables X is not constant for all possible values of X. [8] provides a good overview of the topic. In many practical situations the errors of a predictive model are heteroscedastic. There are several reasons for that, including: missing explanatory variables, non-stationary processes or inability to model high-frequency dynamics ([8], [9]). This fact has been acknowledged for a long time in the literature, and most prediction interval research focuses on the conditional variance. Examples of this are [10], [11], [12], [13], [14], [15].

[14] introduced a very simple method to obtain PIs for a time-series by calculating separately the variances of the different step-ahead errors assuming normality and then using a standard score to generate different step-ahead intervals. Baillie and Bollerslev ([1]) developed analytical asymptotic prediction intervals for the ARMA-GARCH class of models, whereas other procedures have been developed for other classes of ARCH models. See for example [16], [17], [18], [19]. In [20] the bootstrap method is used to find the prediction intervals for any ARCH model. [11] introduces a "nonlinear conditional heteroscedastic

forecasting" model (NCHF) that is estimated using support vector machines and forecasts intervals.

A considerable amount of research has also focused on generating prediction intervals specifically for neural networks. [21] provides a methodology for constructing prediction intervals for neural networks and quantifying the extent to which each source of uncertainty contributes to total prediction uncertainty. Khosravi et al. ([22]) present two techniques, (i) delta, based on the interpretation of neural networks as nonlinear regressors, and (ii) Bayesian, for the construction of prediction intervals to account for uncertainties in transportation travel time prediction. A review of neural network–based prediction interval methods can be found in [15].

2.2.3 Beyond symmetric prediction intervals

Interval predictions are a simple way to characterize the uncertainty around a forecast, but not the only one. A more generic approach is to consider the full probability distribution, namely all possible quantiles. This type of forecasts, called density forecasts, has also gained increasing attention in the last years, especially in finance (see [23] for an example of the need for density forecasts). However, in part because evaluating the performance of density forecasts is not straightforward, their use has been limited. See [24] for a review of density forecasts and their evaluation.

At this point, it is important to note that heteroscedasticity does not imply a symmetric error distribution. Therefore, even though there is a considerable amount of literature dedicated to both heteroscedastic prediction errors and prediction intervals, it is still difficult to find prediction interval modeling approaches that do not assume symmetrically distributed (most of the times Gaussian) errors conditional on the inputs and past information. Under the assumption that the prediction errors are distributed symmetrically conditional on the inputs and past information and that the point forecast is an unbiased estimator of the true regression mean, the prediction intervals are calculated as the point forecast plus/minus the standard deviation of the error multiplied by some constant. Whereas these assumptions are widespread and are acceptable in some cases, they are not in many others. See [3], [25], [26] and [27] for additional background on this topic.

Research on prediction intervals for asymmetrically distributed prediction errors has focused mainly on time-series models that forecast conditional quantiles directly, rather than the mean. Examples of research on asymmetrical prediction errors: [28], [29], [30], [31].

Taylor ([28]) proposes a quantile regression model for the prediction error distribution where the regressor is the lead-time or a basic transformation of it. Engle ([30]) introduces a conditional quantile autoregressive model (CAViaR) to estimate the value at risk (VaR) of a portfolio in financial applications, although it is worth noting that Gorr and Hsu ([32]) had previously introduced a very similar adaptive filter approach. All these approaches are only applicable to time-series and most of the times to specific models.

2.3 Forecast evaluation

Forecast evaluation, validation or verification is a subject that had been widely neglected in the literature until recently ([2] provides the most comprehensive study on this subject). The need for a mechanism or mechanisms to evaluate both single forecasts and to compare different forecast models is very present, especially in fields like economics and atmospheric science. In this section the few most commonly used forecast evaluation metrics are presented.

The Mean Absolute Percentage Error (MAPE) is frequently used to evaluate the performance of point forecast models, usually time-series, and is defined as

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

where y_i are the realizations of the forecasted variable and $\hat{y}_i$ are the corresponding point forecasts. A low MAPE is therefore desirable for a point forecast model.

The Prediction Interval Coverage Probability (PICP), is used to evaluate interval forecasts and is defined as

$$PICP = \frac{1}{n}\sum_{i=1}^{n} I_i$$

where I_i is an indicator variable that can be either 0 or 1, depending on whether the realization of the target variable, y_i is included in the prediction interval PI_i:

$$I_i = \begin{cases} 0, & \text{if } I_i \notin PI_i \\ 1, & \text{if } I_i \in PI_i \end{cases}$$

If a prediction interval is designed to cover X*100% of the future values of the target variable, an optimal interval forecast model would have a PICP value of X.

The Mean Prediction Interval Length (MPIL) is a measure of the narrowness of a set of prediction intervals. It is defined as

$$MPIL = \frac{1}{n}\sum_{i=1}^{n}(PI_i^U - PI_i^L)$$

where PI_i^U and PI_i^L are the upper and lower bounds of the prediction interval PI_i, and it is frequently used together with the PICP. For the same y_i and PICP, it is generally accepted that the prediction intervals that minimize the MPIL are better because they don't overestimate the prediction intervals. However, as it is shown later in this thesis, this is not always a valid assumption.

In most work on prediction interval evaluation both the PICP and MPIL, or slight variations of them, are used in conjunction to evaluate the performance of interval forecast models. This poses a problem when trying to compare two different models, since more than two numbers need to be compared. Questions such as which of the two measures should have more weight when making a decision and how to compare all the values are a concern. There are several different measures that combine the PICP and MPIL throughout the literature, but they are not detailed here (see [22]).

All the aforementioned measures, with the exception of MPIL to a certain extent, provide an evaluation of the unconditional forecast performance. That is, the performance is assumed to be independent of other variables such as time, the actual predicted value or any other variables. This is not usually a realistic assumption and measures that take into account this dependence are available already, although their use is not widespread. Section *3* Methodology explores this topic more in detail.

2.4 Gaussian processes

A Gaussian process (GP) is a stochastic process, i.e. a distribution over functions, such that any finite subset of the domain range follows a multivariate Gaussian distribution. A Gaussian multivariate distribution deals with vectors, and a Gaussian process with function. [33] provides a deep study of Gaussian processes and their practical applications for machine learning, but in a nutshell a Gaussian process can be used to describe any given function. Given a mean and covariance function (analogous to the mean vector and covariance matrix used in Gaussian distributions), a Gaussian process can model in a nonlinear way the relationship between a variable and any number of independent covariates.

A Gaussian process y is typically defined as $y \sim \mathcal{GP}(0, k(x, x'))$, where $k(x, x')$ is the covariance function. The mean function is set to zero in practical applications to allow for an easier treatment of the GP (and easier regression).

Even though a Gaussian process is a non-parametric model and can model any function in theory, a covariance function needs to be specified which limits the possible outputs of the GP. The selection of covariance function or functions when performing GP regression is one of the main areas of research in the field, and [33] provides a very detailed introduction on the topic. Two of the most widely used covariance functions are the squared exponential and the rational quadratic. However, there are many others, like the periodic exponential kernel, Matérn, etc.

Also known as the Radial Basis kernel, the Squared Exponential kernel is the most commonly used covariance function for Gaussian processes, as it provides good results in a wide variety of cases. It is defined as:

$$k_{SE}(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$

where the hyper-parameters ℓ and σ^2 are the *lengthscale* and the output variance, respectively. The length-scale is a hyper-parameter that appears in many kernels and determines how smooth the sample paths generated by the Gaussian process are.

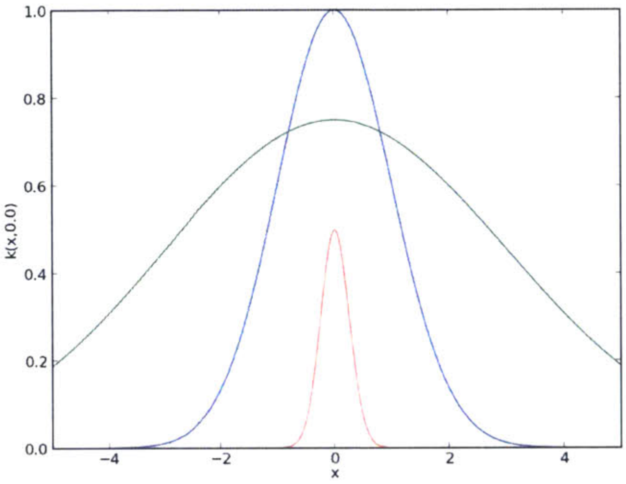

Figure 2-1. Sample squared exponential kernels. Source: (http://gpy.readthedocs.org/en/latest/tuto_kernel_overview.html)

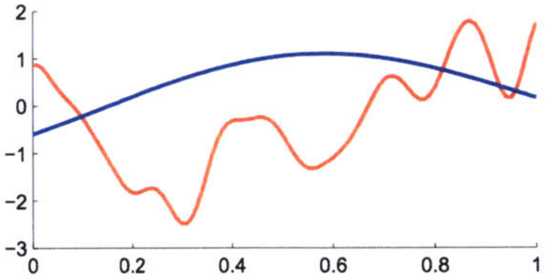

Figure 2-2. Sample paths generated from a squared exponential kernel with lengthscale 0.05 and 0.5 (red and blue, respectively). Source: http://www.cs.toronto.edu/~hinton/csc2515/notes/gp_slides_fall08.pdf

Very popular as well, the Rational Quadratic kernel works with functions that are smooth across a range of lengthscales. When $\alpha \to \infty$ this kernel is equivalent to a squared exponential.

$$k_{RQ}(x, x') = \sigma^2 \left(1 + \frac{(x - x')^1}{2\alpha\ell^2}\right)^{-\alpha}$$

In some cases it is useful to work with more complex covariance functions that can model specific features of the data better. To achieve this, one can add and/or multiply as many kernels as she wishes. The end result is a more complex model with more hyperparameters to learn, but in terms of treatment there is no difference whatsoever.

2.5 Quantile regression

In contrast to least-squares regression methods, which fit a specific function to the conditional mean, quantile regression fits it to a given quantile ([34]). By using a different objective function that takes into account the absolute error as opposed to the squared error, one can come up with a very intuitive, albeit not easy to solve, way to perform quantile regression. If the quantile of interest is the mean, then the objective function is simply the absolute value of the error. However, a function can be fitted to any quantile by using the *tilted loss function*:

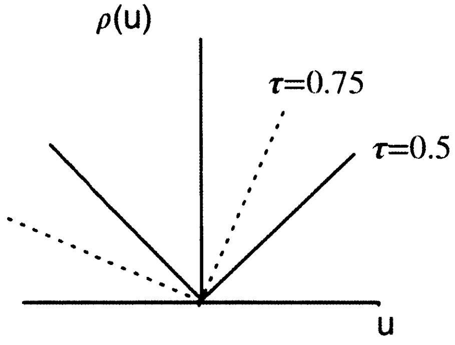

Figure 2-3. Tilted loss function; setting τ to the quantile of interest yields a loss function that can be used for quantile regression.

Koenker ([34]) introduced a mechanism to perform the optimization for linear regression using the above loss function by using linear programming, and since then the use of quantile regression has spread considerably. Nonlinear regression is also possible, as described in the following subsections and more in detail in [35].

2.5.1 Linear quantile regression

In linear quantile regression the conditional quantile is modeled as a linear function

$$Q_y(\tau|x) = x^T\beta(\tau)$$

Where $\hat{\beta}(\tau)$ is obtained by minimizing the tilted loss function ρ_τ:

$$\underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} \rho_\tau(y_i - x_i^T \beta)$$

2.5.2 Locally weighted linear quantile regression

Even though a linear model of the quantiles is an improvement over the unconditional exact quantiles, it is still far from realistic in many cases. In general, the relationships between the input variables and the probability distribution of the prediction error may not have a nature and for that reason more complex functions are needed to accurately model the quantiles of these distributions.

One simple way to create a nonlinear model is to perform a locally weighted liner regression ([36]). For each input value x, the corresponding modeled output is the value at x of a linear quantile that is obtained by assigning different weights to the sample data points based on their distance to x. A distance function, or kernel, can be used to determine the weights. One of the most popular kernels is the *squared exponential*, that is defined as

$$s(x_i) = e^{-\frac{\|x_i - x\|}{2h^2}}$$

Where h is known as the bandwidth and x_i is any sample data point.

While this method is more flexible than the basic linear regression and can produce better results in terms of unconditional coverage of the prediction interval, it can also result in overfitting (and therefore poorer performance) if the bandwidth is not properly chosen. For this reason, it is important to perform cross-validation as part of the modeling process. Cross-validation is a widely used technique for assessing the generalization performance of a model that consists basically in diving the sample data into multiple random subsets and using these subsets for training or validation. For more details, see [37].

2.5.3 Splines quantile regression

A natural step forward from linear regression is polynomial regression. Polynomials provide more flexibility and can fit more complex patterns. Nevertheless, composition of polynomials of low degree known as *splines* can be used instead to obtain better generalization properties.

A splines function is decomposed into a sequence of piece-wise polynomial functions that are linked together at *M knot* points so that it is differentiable everywhere, including the knot points. By choosing carefully the knot points, splines allow for capturing local effects while still being smooth functions and using relatively low-degree polynomials, resulting in better generalization properties.

Any spline function can be decomposed as a linear combination of basis splines, or B-Splines, of the same degree and smoothness. Using B-Splines, a quantile can be expressed as

$$Q_y(\tau|x) = \sum_{i=1}^{M+2} a_i(\tau)B_i(\tau, \mathrm{x})$$

Where i includes all $M + 2$ B-Spline functions and $a_i(\tau)$ is a coefficient for each of the B-Spline functions, $B_i(\tau)$. Note that $a_i(\tau)$ and $B_i(\tau)$ are different for each quantile τ.

B-spline functions are the subject of extensive literature. For further details, the reader is referred to [38].

To make the model more flexible, a combination of linear and B-Splines models can be used for different input variables. In this case, the quantile can be expressed as

$$Q_y(\tau|x) = x_{lin}^T\beta(\tau) + \sum_i a_i(\tau)B_i(\tau, x_{spl})$$

Where x_{lin}^T is the subset of features used in the linear part of the model and x_{spl} the one used in the B-Splines part.

2.5.4 Gaussian process quantile regression

Gaussian process quantile regression ([39]) minimizes the expected tilted loss using an Asymmetric Laplace Distribution likelihood and a GP prior. Although Boukouvalas ([39]) uses Expectation Propagation (see [40]) to approximate the posterior using an exponential-family distribution, Markov Chain Monte Carlo (MCMC) can be used as well whenever the EP approach is unstable (see [39] for more details).

Given a Gaussian process prior in the following form,

$$Q_y(\tau|x) \sim \mathcal{GP}(0, k(x, x'))$$

where k is the covariance function. The zero mean assumption is used throughout the Gaussian process literature and makes it much easier to perform regression.

The GP quantile regression process finds the posterior distribution of the quantile regression function q

$$p(q|\mathcal{D}, \theta) = Z^{-1} p(\mathcal{D}|q, \theta) p(q|\theta)$$

Where for simplicity q is the quantile $Q_y(\tau|x)$, $\mathcal{D}$ is the training dataset, θ are the hyper-parameters of both the covariance function *and* the likelihood function (see below) and Z is the normalization factor, defined as

$$Z = \int p(\mathcal{D}|q, \theta) p(q|\mathcal{D}, \theta) dq$$

The hyper-parameters are obtained by MAP (Maximum A Posteriori) estimation:

$$\arg\max_{\theta} p(q|\mathcal{D}, \theta)$$

As mentioned before, the likelihood used for quantile regression is the multivariate Asymmetric Laplace Distribution (ALD) because, as Yu and Moyeed ([41]) show, maximizing the ALD is equivalent to minimizing the tilted loss function. For the entire training dataset, the likelihood then becomes

$$p(\mathcal{D}|q, \theta) = \left(\frac{\tau(1-\tau)}{\sigma}\right)^n e^{\left[-\sum_i \frac{u_i}{\sigma}(\tau - I(u_i < 0))\right]}$$

Where τ is the quantile of interest, σ is the standard deviation of the ALD (and part of the hyper-parameter vector θ), n is the size of the training dataset $\mathcal{D}$, u_i is the difference between each of the training data points' observed value y_i and the corresponding estimate by q ($u_i = q_i - y_i$), and I is zero when the condition is false and one otherwise.

For prediction purposes the predictive mean and variance of the Gaussian process at a new point x_* are calculated. See [39] for details on the process.

3 Methodology

In this chapter, a methodology for estimating the prediction interval of a generic real-valued point forecast model (*black-box* model) is presented. This methodology consists of a *meta-model* that runs in parallel to the original black-box model and is able to generate prediction intervals for each of the original point forecasts based on the observed prediction error of the black-box model.

3.1 Overview

In its basic configuration, the inputs to the prediction interval meta-model are:

- The inputs used by the original black-box model
- The output of the black-box model, namely a point forecast
- The desired prediction interval, which is a configuration parameter of the meta-model

Given only these inputs, the meta-model generates a prediction interval that complements the original one step ahead point forecast. Figure 3-1 illustrates the high-level methodology.

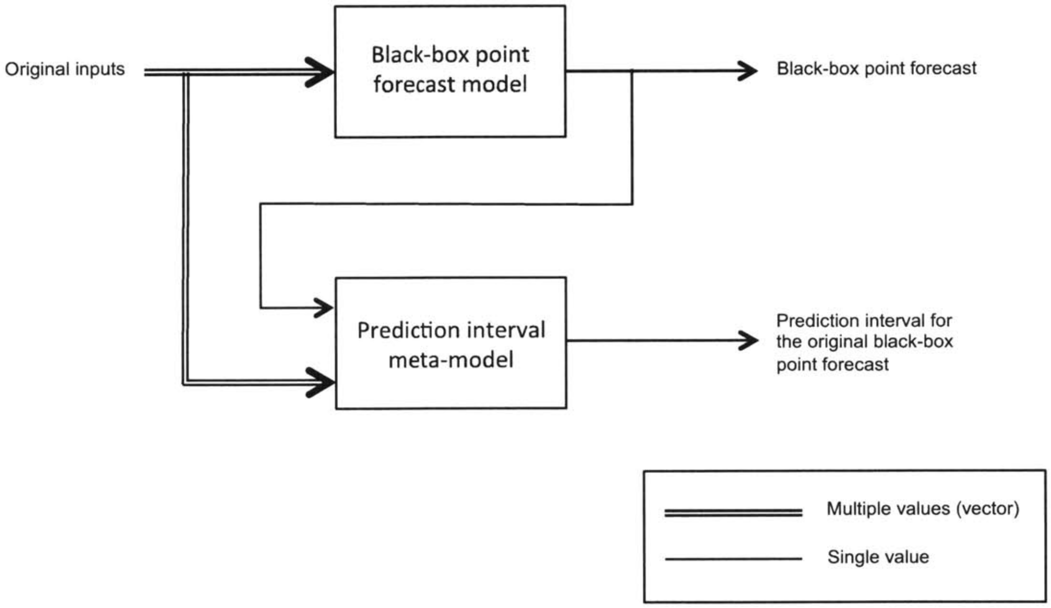
Original inputs
Black-box point forecast model
Black-box point forecast
Prediction interval meta-model
Prediction interval for the original black-box point forecast
Multiple values (vector)
Single value

Figure 3-1. Block diagram of the meta-model approach.

Note that unless stated otherwise, the term "input" will be used from now on to refer to the all the inputs of the meta-model, and not those of the original black-box model.

The meta-model is trained in a supervised way and the target variable is the *observed prediction error distribution* of the black-box model. More specifically, the goal of the training process is to learn two specific quantiles of the prediction error distribution, which will then be used to generate the prediction intervals, using quantile regression.

For a $100\times(1-\alpha)\%$ desired prediction interval, the $\alpha/2$ and $(1-\alpha/2)$ quantiles are modeled independently. For example, for a 95% prediction interval the quantiles of interest are 0.025 and 0.975. In most practical applications α will be relatively small since the prediction intervals are expected to cover most of the future observations.

Once trained, the meta-model generates the prediction interval by estimating the upper and lower quantiles given the inputs and adding them to the black-box model point forecast.

Throughout the rest of this section the following notation will be used:

- y is the target variable of the black-box model
- $\hat{y}$ is the black-box model point forecast
- ε is the black-box model prediction error defined as $\hat{y}-y$
- E is the prediction error ε cumulative probability distribution
- $\mathbf{x}_{bb}$ is a the input vector of the black-box model
- $\mathbf{x}$ is the input vector of the meta-model and it is composed of $\mathbf{x}_{bb}$ and y
- $\tau\in[0,1]$ is a quantile
- $Q_\varepsilon(\tau)=E^{-1}(\tau)$ is the τ-th quantile of the error probability distribution E
- $Q_\varepsilon^U=Q_\varepsilon(1-\alpha/2)$ is the quantile used to generate the upper bound of the prediction interval
- $Q_\varepsilon^L=Q_\varepsilon(\alpha/2)$ is the quantile used to generate the lower bound of the prediction interval

3.2 Motivation

The motivations for modeling the observed prediction error distribution quantiles are:

a) In many fields, point forecast models that have been improved over the years are being used already with good results. By working on the observed prediction error of these models we avoid creating a completely alternative model of the quantiles and take advantage of the mean predictions.

b) In many situations, even if desired, these new models of the quantiles might be very hard to obtain. In traffic prediction, for example, complex statistical traffic models can't easily be adapted to produce quantiles.

c) A generic meta-model approach can provide good results for a wide variety of problems and black-box models and still be simple to implement.

3.3 Quantile modeling

The overall meta-model methodology introduced in this chapter relies on the accurate estimation of the prediction error distribution quantiles, regardless of the actual modeling approach used to obtain these estimations. This section describes some of these approaches, all based on quantile regression, and justifies the use of Gaussian process quantile regression in favor of the other methods.

It is important to note that specific models may give better results depending on the situation, and therefore there is no one-size-fits-all approach when it comes to quantile modeling, just as there is no single modeling approach in any branch of statistics and machine learning. However, a flexible model is preferred over more restricting ones if the goal is a generic methodology that will work for a broader range of problems, and that is precisely the intent of the methodology presented in this thesis.

As opposed to probability distribution modeling, where the entire distribution is estimated, the methodology described in these pages focuses on the independent estimation of two quantiles of a distribution. For this reason, one can apply different models to each of the quantiles if that proves beneficial, even if sticking to Gaussian processes only, as there are plenty of covariance functions that can be chosen.

3.3.1 Homoscedastic approach

The simplest way to model a quantile of the prediction error distribution is to assume that this is constant (i.e. homoscedastic). Under this assumption, calculating any exact quantile of the sample is straightforward. Given the (right-continuous) distribution function of the prediction error $\mathrm{E} = P(y - \hat{y})$

$$F(\varepsilon) = P(\mathrm{E} \leq \varepsilon)$$

And the τ-th quantile defined as

$$F^{-1}(\tau) = \inf\{\varepsilon : F(\varepsilon) \geq \tau\}$$

The prediction interval for a given point forecast $\hat{y}_i$ is given by:

$$PI = \left(\hat{y}_i + \mathrm{E}^{-1}(\alpha/2), \hat{y}_i + \mathrm{E}^{-1}(1 - \alpha/2)\right) = (\hat{y}_i + Q_\varepsilon^L, \hat{y}_i + Q_\varepsilon^U)$$

Note the positive sign of both sums, given that we are using directly the quantiles of the prediction error distribution.

This approach is straightforward and fast to compute, and in some cases the calculation of the quantiles can be simplified even further by assuming an error distribution with well-known properties. For instance, if a normal distribution is assumed, the quantiles can be obtained as the mean plus/minus the standard deviation times a constant (standard score) that depends on the quantiles of interest. In fact, this normality assumption is widely used both in the prediction interval literature and in practice.

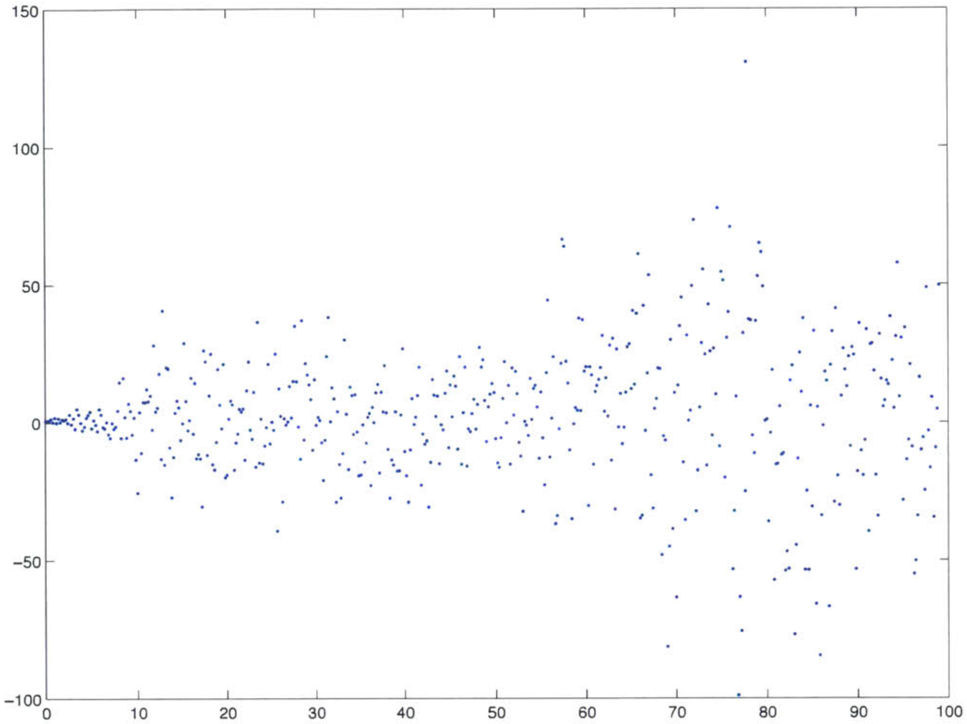

Figure 3-2. 0.025 and 0.975 constant quantiles of toy univariate prediction error.

Note that the prediction intervals derived using this method will contain overall the desired percentage of predictions with a high accuracy. This is what is known as unconditional interval coverage. However, this is not necessarily true if the coverage is tested under certain specific conditions (for some input values or certain periods of time) and the prediction error is heteroscedastic. The coverage under certain conditions is known as the conditional coverage. Figure 3-2 is an example of good unconditional coverage but bad conditional coverage, as the constant quantiles are not a good approximation to the real quantiles of the error distribution for some input values.

3.3.2 Linear quantile regression

As described in *2.5.1* Linear quantile regression, using linear quantile regression the conditional quantile is modeled as

$$Q_{\varepsilon}(\tau|\mathbf{x}) = \mathbf{x}^T\beta(\tau)$$

Figure 3-3 shows the conditional linear quantiles of the same toy example. Using a linear approximation of the quantiles, the unconditional coverage of the prediction interval defined by the two quantiles over the training set (the one depicted in the figure below) is 93%, which is worse than using constant quantiles but still reasonably good. However, one can sense that the conditional coverage has improved and that makes the approach better overall. In *3.4 Model performance* different methods to precisely quantify the conditional coverage are described, but for the purposes of comparing different quantile models for this toy example a simple visual inspection will be enough.

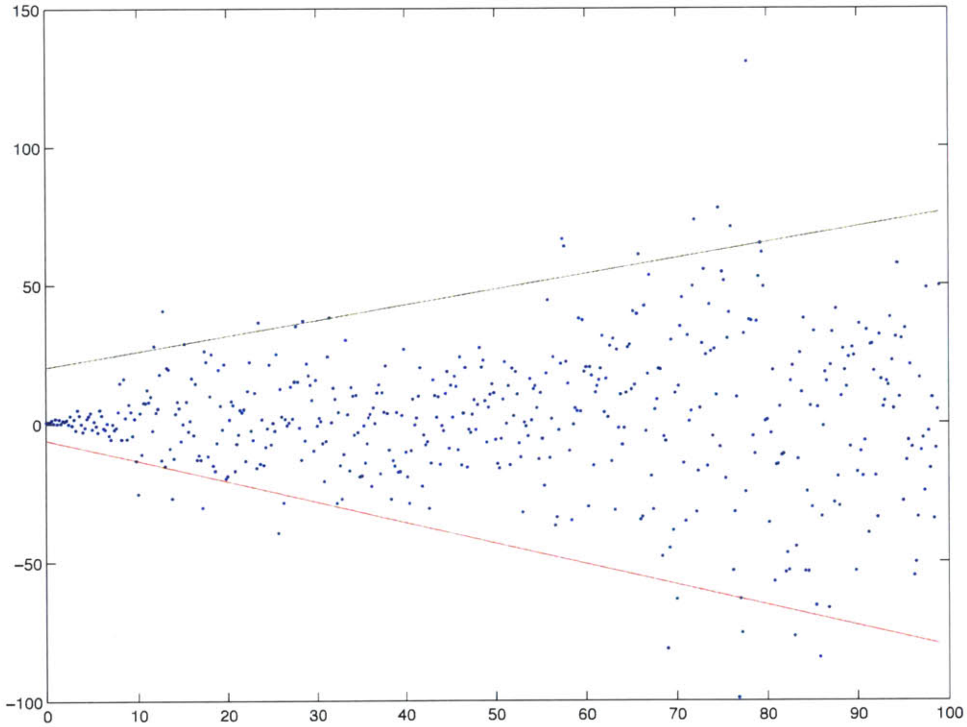

Figure 3-3. Linear quantile regression for the 0.025 and 0.975 quantiles of toy univariate prediction error.

3.3.3 Splines quantile regression

Figure 3-4 shows an example of prediction interval modeling using splines quantile regression on the same toy example used so far:

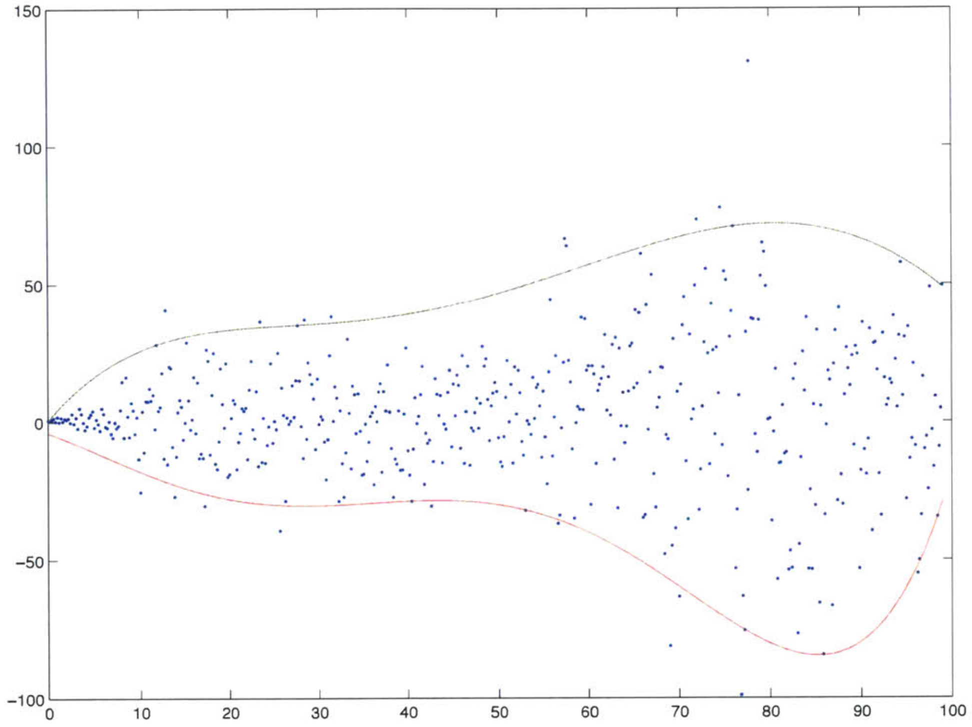

Figure 3-4. B-Splines quantile regression for the 0.025 and 0.975 quantiles of sample univariate prediction error.

From a practical standpoint, one of the challenges associated to the use of splines is the selection of the number and placement of the knots. As a rule of thumb, it is better to place more knots in regions where the quantile is supposed to vary more rapidly and, again, it is recommendable to perform cross-validation to make sure the model generalizes well.

3.3.4 Gaussian process quantile regression

Compared to the methods seen so far, GPs are an appealing choice for prediction interval modeling because they can model a wide variety of complex functions and behave well in high-dimensional input spaces, while still being relatively easy to use. Chapter 4 compares in detail GP quantile regression for prediction interval modeling for real-world point forecast models to the rest of quantile modeling approaches seen so far, but the following toy example shows a good prediction interval performance using GPs.

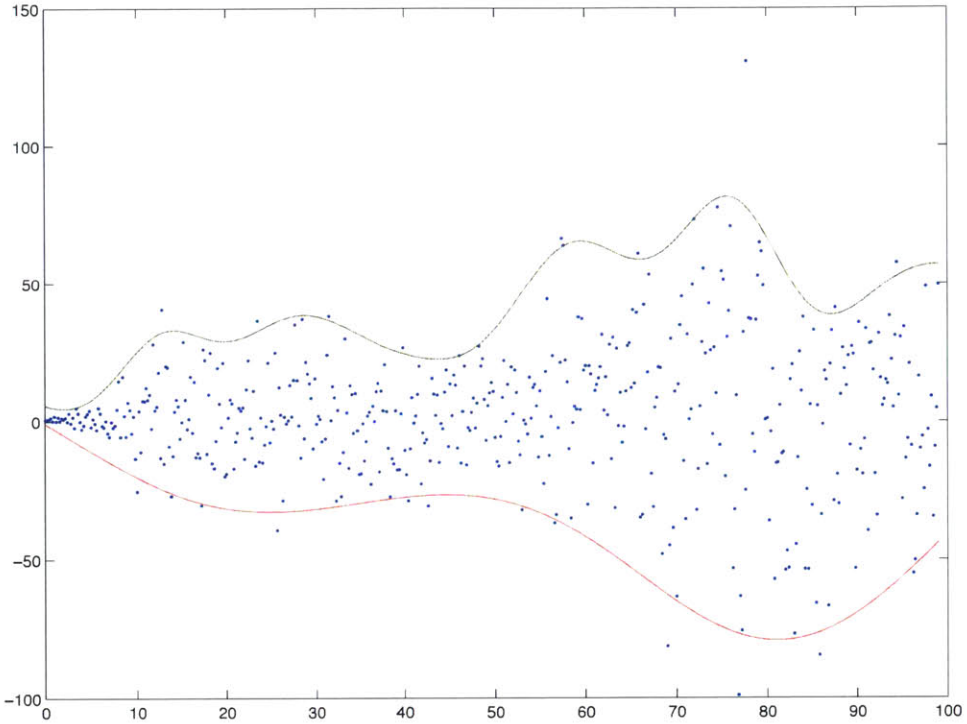

Figure 3-5. Gaussian process quantile regression for the 0.025 and 0.975 quantiles of toy univariate prediction error using a squared exponential covariance function.

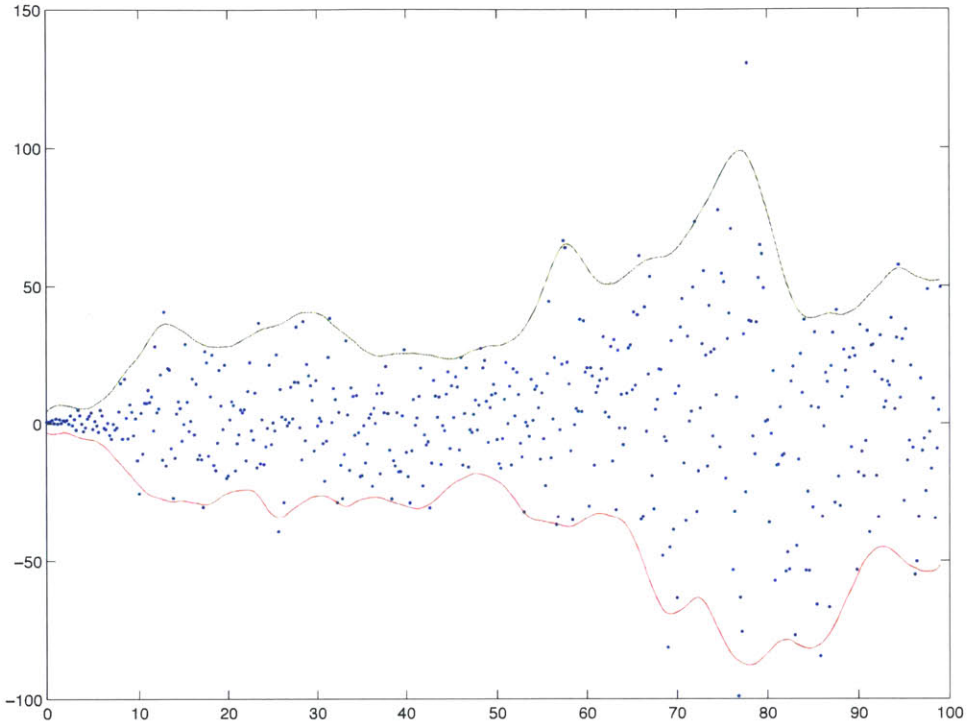

Figure 3-6. Gaussian process quantile regression for the 0.025 and 0.975 quantiles of toy univariate prediction error using a Matérn 5/2 covariance function.

For an overview of Gaussian processes and GP quantile regression see *2.4 Gaussian processes* and *2.5.4 Gaussian process quantile regression*, respectively.

3.3.5 Practical considerations and limitations when using quantile regression

3.3.5.1 Extreme quantiles

For the purpose of generating prediction intervals the quantiles of interest are generally extreme, either close to zero or one. As it has been acknowledged in the literature, however, quantile regression for extreme quantiles in the tail of the distribution yields suboptimal results (see [35] page 146 and [42]). Therefore, the use of quantiles that are as central as possible is recommended, regardless of the quantile regression approach.

3.3.5.2 Quantile Crossing

By definition, any cumulative probability distribution is a monotonically increasing function, and even in scenarios where there is heteroscedasticity the probability distribution is assumed to be constant at a given point. Therefore, for any fixed point the quantiles also increase monotonically, which implies that two real quantile functions never cross.

However, quantile regression is concerned with a function that fits as efficiently as possible one single quantile and not the whole probability distribution. In the methodology detailed in this chapter two quantiles are modeled independently and, given that the result of the regression does not necessarily match the real quantiles, it should not be surprising to find that they cross at certain points. This can easily be seen when using linear regression: as long as the two quantiles have different slopes they will inevitably cross at some point.

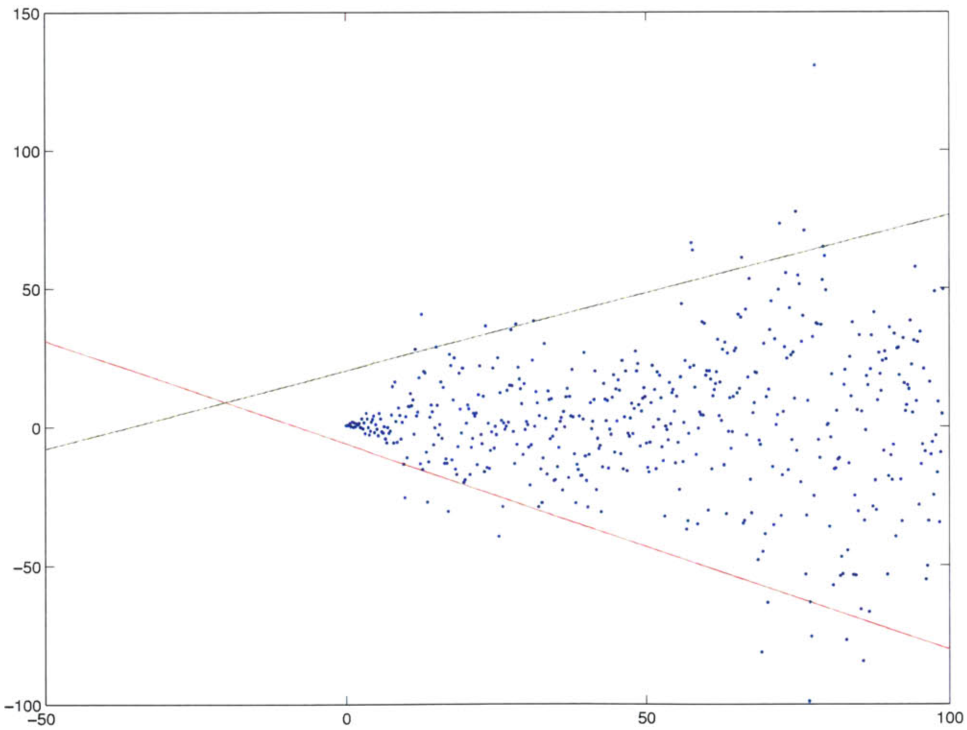

Figure 3-7. Quantile crossing example.

Quantile crossing has been the focus of some research and, according to Koenker ([35]), it is most of the times the result of model misspecification. Intuitively, it is easy to see that the more similar the modeled quantiles are to the real ones the less probability of crossings.

It can be shown that for at least linear regression the conditional quantile function is monotone in τ at the centroid of the design $\bar{x} = N^{-1} \sum \mathbf{x}_i$. Also, for nearest neighbor approaches, like locally weighted regression or GPs, the strong local influence of the data makes it more difficult for the quantiles to cross. However, it is in general not possible to mathematically ensure that two quantiles will never cross when they are modeled using quantile regression (i.e. independently).

The practical implications of quantile crossing for prediction interval generation are:

- The quantile models should be selected so that there is no quantile crossing for the input values of interest (i.e. the input region for which training data is available).
- The meta-model should be checked for quantile crossing.
- Even when all precautions are taken, crossing may occur. When this happens, the meta-model needs to avoid generating prediction intervals that make no sense (lower bound higher than the upper bound) by using some predefined prediction interval. Many solutions are possible, but a good, intuitive one is the use of the static quantiles of the homoscedastic approach.

3.3.5.3 GP regression using MCMC vs. EP

The integrals described *2.5.4 Gaussian process quantile regression* are analytically intractable and for this reason numerical methods like Markov Chain Monte Carlo ("MCMC", [43], [44]) or approximation methods need to be used. Taking into account that the Asymmetric Laplace Distribution likelihood used for quantile regression is not differentiable at all points, Laplace's method or variational Bayes cannot be used. Instead, Expectation Propagation ("EP", [40]) needs to be used.

Although Boukouvalas ([39]) uses EP for GP quantile regression, MCMC is recommended for prediction interval generation purposes since EP proves unstable in practice when modeling extreme quantiles with available software packages as it's shown in chapter 4.

MCMC is computationally more expensive, but provides results good results for moderately extreme quantiles.

3.3.5.4 Fast approximations for GP regression

The main practical limitation of GP regression is its basic complexity ($O(N^3)$, where N is the size of the training dataset). For this reason, for large datasets ($N > 10000$) GPs are not viable.

In order to overcome this problem several approximations have been proposed (see [33] for a full review), including:

- Reduced-rank matrix approximations
- Subset of data points
- Approximating the marginal likelihood (using EP in the case of quantile regression)

3.3.5.5 GP kernel

A critical decision when performing GP quantile regression, and GP regression in general, is the selection of the covariance function or kernel. While the training process will learn the hyper-parameters, the covariance function itself remains the same from the beginning. Therefore, it is important to select a kernel, or combination of kernels, that represent as accurately as possible the relationships between the different variables.

Since this is not a simple task in some cases, especially with high-dimensional data where the relationships between the different variables can't be easily visualized, an automated search for the most efficient kernel combination has been implemented as part of the meta-modeling approach and it is described in *3.5 Automatic feature and GP kernel* selection.

3.3.5.6 Data normalization

Gaussian process regression requires the dataset to be normalized to achieve good results. In this case normalization means zero mean and unit standard deviation for all dimensions.

3.3.5.7 Overfitting

In any machine learning problem overfitting is always a risk. The literature on this topic is extensive and it is out of the scope, other than suggesting the use of cross-validation ([37])

and trying different initial values for the hyper-parameters of the Gaussian process covariance function to minimize the chances of finding local maxima.

3.4 Model performance

So far the quantile modeling methodology has been discussed, but nothing has been said about how to measure the performance of the prediction intervals generated by the meta-model. In order to make a decision on whether the meta-model is accurate enough and to select the best model amongst several different ones, a performance metric needs to be defined.

Several measures of the performance of prediction intervals can be found in the literature, although there does not seem to be any consensus around which one to use. For a review please see *2 Background and literature review.*

In this work, two different methods that can be found in the literature but are rarely used are recommended and used:

- Christoffersen's likelihood ratio framework, used to quantitatively evaluate the overall performance of the generated prediction intervals over time
- A proper interval score, used to evaluate the performance of the generated prediction intervals with respect to the meta-model input variables

Also, two additional methods to measure the performance of the model with respect to the inputs are proposed: subspace division unconditional coverage and projection onto one dimension.

The rest of this section is dedicated to explain why introduce this methods and the motivations for using them in favor of more popular measurements, and why it is important to measure the conditional interval coverage.

3.4.1 Problem definition

Given a sequence of prediction intervals and corresponding sequence of future realizations, how do we evaluate the performance of these predictions? Moreover, how do we evaluate the performance of the predictions under specific conditions?

As it has been seen already, the fact that the prediction intervals generated by the meta-model cover, on average, the right proportion of future realizations does not imply that in some specific scenarios the prediction intervals over or underestimate the actual prediction error.

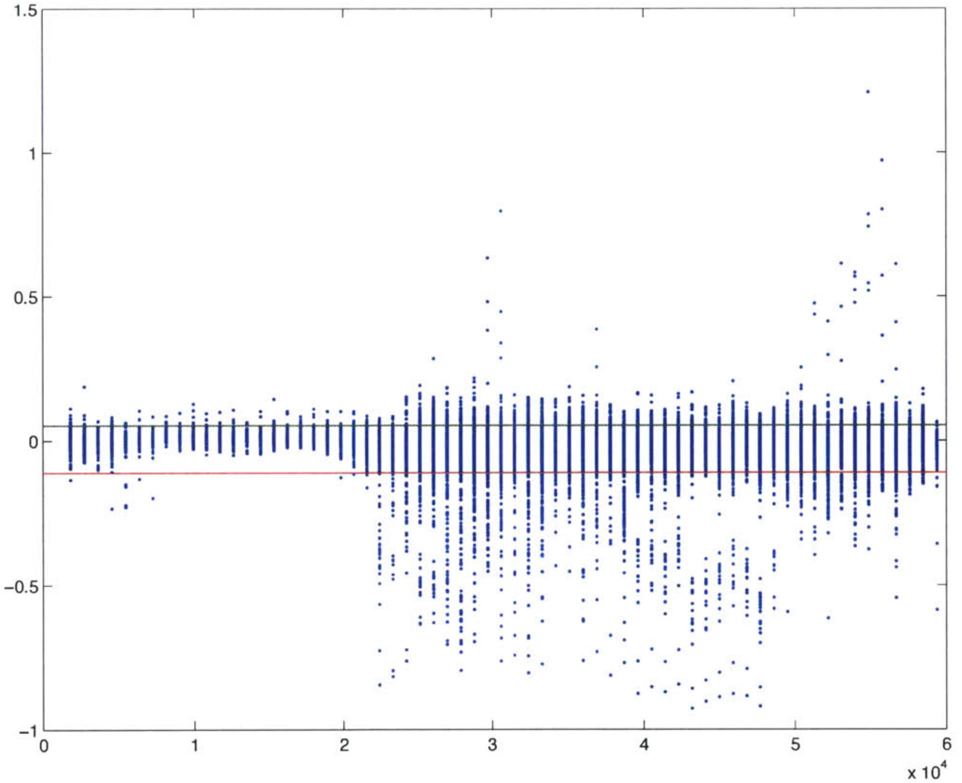

Figure 3-8. Example prediction relative error of a traffic model during the span of a day. The horizontal axis represents the time of the day in seconds since midnight.

The example in Figure 3-8 shows the one-step-ahead prediction error of a traffic forecast model during the span of a day. The target variable of the traffic model is the travel time through a segment of the road network. The two horizontal lines are the 0.05 and 0.95 quantiles of the entire sample. In this example, the unconditional coverage of the two quantiles combined together (prediction interval) is exactly 90%, which is perfect because we are testing against the training data but nevertheless will still be very accurate for another day of data with similar characteristics. However, it is easy to see that the prediction intervals are not accurate during rush hour (i.e. the conditional coverage is not good).

Even though the conditional coverage of prediction intervals is always analyzed as a function of time in the literature (at least in part because time-series are almost always the subject of study) this is not necessarily the only dimension that should be taken into account. Back to our traffic-modeling example, one may find that the prediction error is higher when it rains, or when special events like concerts are going on, or even when the forecasted value itself is higher than normal. If these or other variables are used as inputs to the prediction interval meta-model, it is necessary to evaluate the interval coverage as a function of these variables because the effects may be unnoticeable if analyzed over time only. Moreover, for some models time may not be a suitable variable at all.

3.4.2 Performance for time-series

Whenever the conditional coverage of the prediction intervals is only needed as a function of time, Christoffersen's framework ([9]) is recommended to determine the performance of the meta-model.

Given the interval coverage indicator $I_t = \begin{cases} 1, & \text{if } (y_t - \hat{y}_t) \in \left[Q_y^L(x_t), Q_y^U(x_t)\right] \\ 0, & \text{if } (y_t - \hat{y}_t) \notin \left[Q_y^L(x_t), Q_y^U(x_t)\right] \end{cases}$

where Q_y^L and Q_y^U are the lower and upper quantiles of the meta-model, respectively, and x_t are the meta-model inputs at time t Christoffersen establishes that a sequence of prediction intervals is efficient if

$$E[I_t | I_{t-1}, I_{t-2}, \dots] = p$$

where $p = (1-\theta)$, the targeted prediction interval coverage.

He proves that testing for the above condition is equivalent to testing

$$\{I_t, I_{t-1}, I_{t-2}, \dots\} \sim Bern(p)$$

and proposes a likelihood ratio framework to perform this test:

$$LR_{CC} = -2log\left(\frac{(1-p)^{n_0}\, p^{n_1}}{(1-\hat{\pi}_{01})^{n_{00}}\, \hat{\pi}_{01}^{n_{01}}\, (1-\hat{\pi}_{11})^{n_{10}}\, \hat{\pi}_{11}^{n_{11}}}\right) \sim \chi^2(2)$$

Where:

n_i = number of observations with value i in $\{I_t, I_{t-1}, I_{t-2}, \dots\}$

n_{ij} = number of observations with value i followed by j in $\{I_t, I_{t-1}, I_{t-2}, \dots\}$

$\hat{\pi}_{01} = \frac{n_{01}}{n_{00}+n_{01}}$

$\hat{\pi}_{11} = \frac{n_{11}}{n_{10}+n_{11}}$

Therefore, LR_{CC} gives us a way to determine if a meta-model is efficient and also it allows us to compare the performance of two or more different models. Note that this statistic measures both the efficiency of the conditional and unconditional coverage simultaneously. For a detailed proof see [9].

3.4.3 Performance with respect to the inputs

In multidimensional scenarios the measure of the conditional coverage efficiency using Christoffersen's framework is not straightforward because there is not a sense of a sequence of observations in multiple dimensions. In this case it is not only necessary to test

$$E[I_t | I_{t-1}, I_{t-2}, \dots] = p$$

as in the time-series case if a time sequence of observations can be built, but it is also necessary to test

$$E[I_n | x_n] = p$$

where I_n is any given observation and x_n is the corresponding input vector of the meta-model, including any lagged inputs (see chapter 4).

The two equations above test the *dynamic* and *static* performance of the meta-model, respectively, and it is worth noting that the fact that one of them is good does not imply that the other is. For instance, if the static performance is bad only under specific input values and those values are rarely seen consecutively the dynamic performance might still be good. Conversely, if some input values for which an otherwise very good static performance is not so good are often seen consecutively, the dynamic performance would be bad.

Testing for $E[I_n|x_n] = p$ is straightforward if x is unidimensional. In this case, Christoffersen's framework can be applied to the sequence of observations ordered according to this dimension, and not time.

For higher-dimensional problems, Christoffersen's framework can't be applied because there is no such thing as a unique unidimensional sequence of observations, and even if a sequence is built from the sample, which is the one that should be chosen out of all the possible combinations, if any?

Alternatively, autocorrelation measures have been used before to test the independence of the sequence of indicators in time-series ([16]), and some multidimensional autocorrelation measures and statistics have been used in fields like geography and ecology. However, most of the used methods assume a 2 or 3-dimensional space and data points located at the vertices of a lattice structure (see [45], [46]), which are not applicable to the study of prediction intervals in the multidimensional input meta-model scenario.

Therefore, given the lack of a robust statistical test for $E[I_n|x_n] = p$, different methods are hereby suggested.

3.4.3.1 Subspace division

Given the input space X, multiple divisions X_i are made and the unconditional coverage is tested against the null hypothesis

$$\frac{1}{N} \sum_{n \mid x_n \in X_i} I_n = p$$

where I_n are the observations whose input vector x_n is included in subspace X_i and N is the total number of observations in subspace X_i.

The following statistic can be used to determine if the null hypothesis is rejected for any of the subspaces ([9]):

$$-2log\left(\frac{(1-p)^{n_0}\,p^{n_1}}{\left(\frac{n_0}{n_0+n_1}\right)^{n_0}\,\left(\frac{n_1}{n_0+n_1}\right)^{n_0}}\right)\sim\chi^2(1)$$

Where:

n_i = number of observations I_n with value i

The advantage of using this method is that it can locate those subspaces for which the meta-model does not perform. On the other hand, the selection of the subspaces poses a difficulty. As a general rule of thumb, any subspace should contain enough observations so that the test can be performed with some level of confidence.

3.4.3.2 Projection onto one dimension

If the entire sample is projected onto one dimension, Christoffersen's framework can be applied to detect dependencies between observations.

Ordering the observations according to the position they occupy results in a sequence that can be used to test their independence, as long as this dimension is continuous. For discrete dimensions more than one observation may occupy the same position, and other information would be required to determine the order of the sequence, which cannot be random because we would remove any trace of autocorrelation. In these cases, the order can be based on the value of any of the other dimensions.

The advantage of this method is that is very easy to apply. The disadvantage is that one cannot expect to find all spatial dependencies since many can be masked out during the projection.

3.4.3.3 Proper interval score

In order to provide a single quantitative measure of the performance of the meta-model, a different approach can be taken. Intuitively, it is easy to see that in a heteroscedastic scenario like the one assumed throughout this text the conditional prediction interval coverage is somehow tied to the width of the prediction intervals: as long as the total (unconditional) coverage meets the expectation, narrower prediction intervals generally leads to better conditional coverage. Figure 3-9 illustrates this intuition:

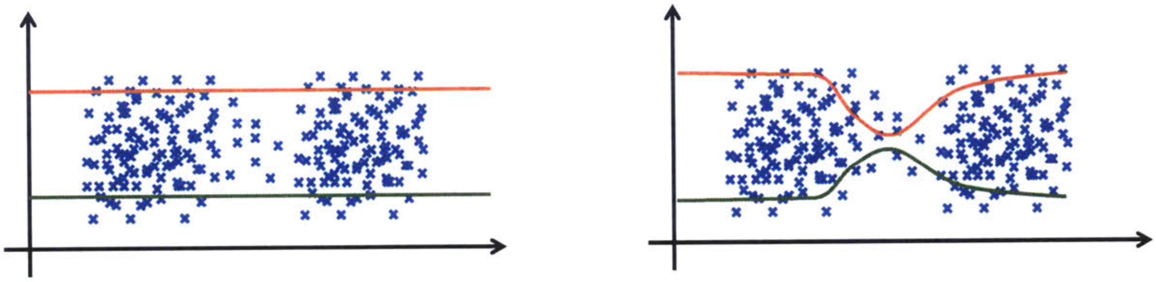

Figure 3-9. Intuitive graphical explanation supporting narrower prediction intervals.

In both examples above the unconditional coverage (or PICP) is the same. On average, the prediction intervals on the right are narrower or, equivalently, the area between the upper and lower bounds is smaller. Since the conditional coverage is clearly better on the right-hand example, one may arrive to the conclusion that narrower prediction intervals are always better, given the same unconditional coverage. However, this view of the prediction interval width, even though relatively widely assumed in the literature, is misleading, as the following simple example illustrates:

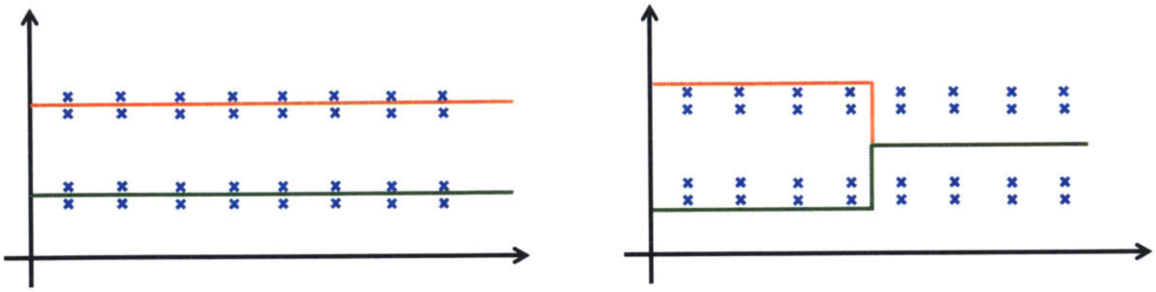

Figure 3-10. Example showing why a narrower average interval width is not necessarily good.

In Figure 3-10 two prediction intervals for a very simple dataset are shown. While both of them cover the same amount of observations, one is on average narrower. The narrower one, however, turns out to be clearly worse in terms of conditional coverage.

A way to correct this is to penalize according to the distance between the observation and the closest interval bound for those observations that fall outside the interval. The proper interval score (*2.3 Forecast evaluation*, [47]) does precisely that. Put simply, it is a scoring measure that combines the unconditional and conditional coverage and that rewards narrow prediction intervals and at the same time penalizes observations that miss the interval with a value that is proportional to the distance between the observation and the closest interval bound. In our setting:

$$S_\tau^{int}(l,u;y-\hat{y}) = (u-l) + \frac{1}{\tau}\big(l-(y-\hat{y})\big)I(y-\hat{y}<l) + \frac{1}{\tau}\big((y-\hat{y})-u\big)I(y-\hat{y}>u)$$

Where $l = Q_y^L(x)$, $u = Q_y^U(x)$ and I is zero when the argument is false and one otherwise.

The score for the entire dataset is, therefore

$$S_\tau^{int}(\mathcal{D}) = \frac{1}{N}\sum_i S_\tau^{int}\big(Q_y^L(x_i), Q_y^U(x_i); y_i - \hat{y}_i\big)$$

The proper interval score has three main advantages over the more popular metrics PICP and MPIL (*2.3* Forecast evaluation):

- It combines in a single measure the unconditional coverage of the model and the width of the intervals.
- By penalizing differently based on the distance to the interval, narrower prediction intervals are not favored when they should not.
- It is a proper score (see [47])

Therefore, the proper interval score is preferred over these metrics, or exotic combinations of both, for optimization purposes or to compare two different models.

However, in many cases it is still useful to know what the PICP and MPIL for a given model and dataset are since they can be easily understood.

3.5 Automatic feature and GP kernel selection

3.5.1 Feature selection

In many practical situations a dataset contains multiple features whose relevance is not fully understood. It is often the case that some of these features do not provide any additional information about the behavior of the target variable, but if included in the model they make it more complex and sometimes less efficient. For this reason, it is always a good idea to remove those features that do not carry any additional information, especially in high-dimensional problems.

The literature on this subject is extensive, and many methods have been introduced over the years (see, for example, [48] for a review), and there is no restriction on the feature selection method that can be used, but for illustration purposes the Best First algorithm is chosen and used in chapter 4.

Starting from zero selected features, this algorithm keeps adding the feature that increases the performance of the model the most at each step, out of all the remaining features. If no feature can be found that improves the performance or if the increase in performance is negligible, the algorithm stops.

This greedy algorithm requires training the model at each step $n - k$ times, namely as many as features left. Therefore, it can be computationally very expensive and only practical for small datasets.

3.5.2 GP kernel search

In order to select the best covariance functions for GP regression a visual analysis of the data is recommended. In cases when this is not possible due to the high-dimensionality of the problem or any other reason, an automated kernel selection mechanism can prove useful.

The selection algorithm can be as simple as a brute force test of all combinations of a certain list of predefined kernels, which is the method that is used in chapter 4. For example, three covariance functions can be combined in 13 different ways: A, B, C, A+B, A+C, B+C, AB, AC… However, it is not necessary to encode all combinations and a list of "potentially good" kernel combinations can be predefined and automatically tested.

As a rule of thumb, the different kernels should be chosen to have quite different characteristics so that one can explain features of the data that the rest cannot, such as a squared exponential and a linear kernel.

In some cases using different kernels for all or some of the dimensions can achieve better results as well. Moreover, different combinations of kernels can be used for each of the two quantiles of the meta-model. All this results in more computational resources and depending on the application the extent of the search may need to be limited.

3.6 Extensions to the basic model

3.6.1 Lagged inputs

Using the same framework seen so far, one can add memory to the meta-model by using lags of any of the original meta-model inputs, which include the inputs and output of the black-box model. This is useful whenever the black-box prediction error exhibits autocorrelation.

Therefore, it is not only useful to use the lags of the mentioned inputs but more importantly the past observed prediction errors can be used as well. This can be in practical scenarios useful to account for unexpected events. For instance, in a traffic prediction model like DynaMIT ([49]), the sudden increase of the observed prediction error for the last few iterations may indicate that an incident occurred in the network. If there is no explicit information about incidents available, the meta-model can use this change in the observed prediction error to adjust the prediction intervals temporarily.

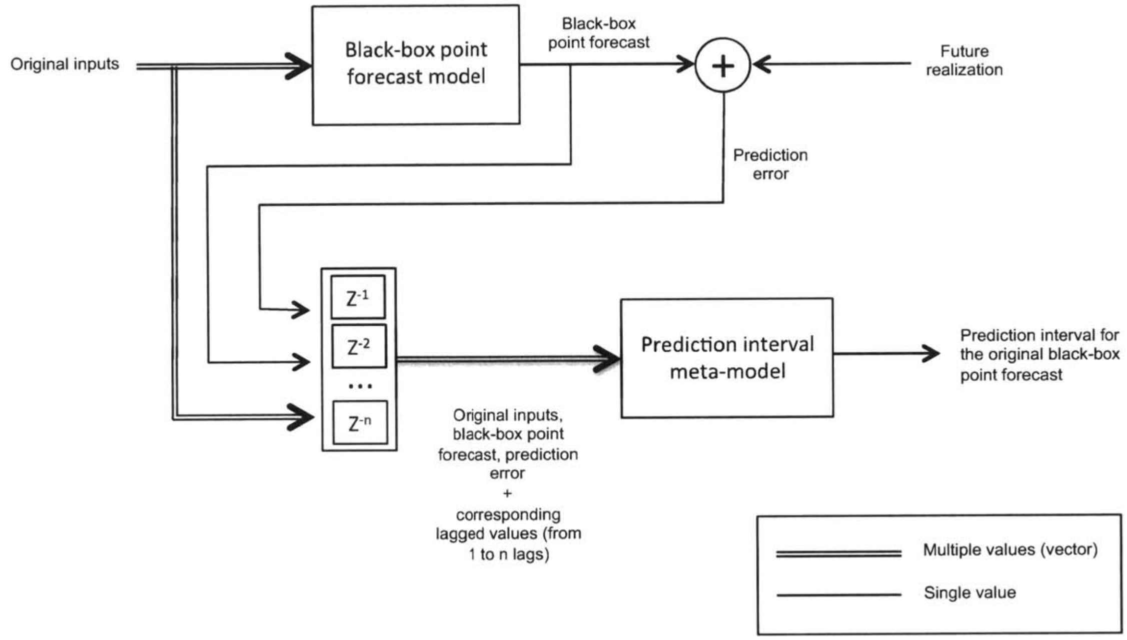
Original inputs
Black-box point forecast model
Black-box point forecast
+
Future realization
Prediction error
Z-1
Z-2
...
Z-n
Prediction interval meta-model
Prediction interval for the original black-box point forecast
Original inputs, black-box point forecast, prediction error + corresponding lagged values (from 1 to n lags)
Multiple values (vector)
Single value

Figure 3-11. Meta-model framework using past information.

It is worth noting that adding past information into the model does not affect the conditional coverage criteria established earlier in section 3.4, given that the lagged inputs are treated just like any other input by the meta-model.

3.6.2 Multiple predictions

Some black-box models may generate predictions for more than one variable simultaneously. For example, DynaMIT produces multiple predictions for each of the different network locations at each step. In these cases, it may be a good idea to use all the predictions (and prediction errors) or a subset of them at once as inputs to the meta-model in order to take advantage of any correlations that may exist between them.

In DynaMIT's road network example it is easy to imagine that the past observed prediction errors in one part of the network might be correlated with the future prediction errors at adjacent road segments, and how one could take advantage of this correlation to generate better prediction intervals.

Of course, if the black-box model outputs more than one point forecast the meta-model needs to generate the same number of prediction intervals. Instead of having a single meta-model, though, multiple ones are used. Therefore, the extended meta-model framework looks like the following diagram.

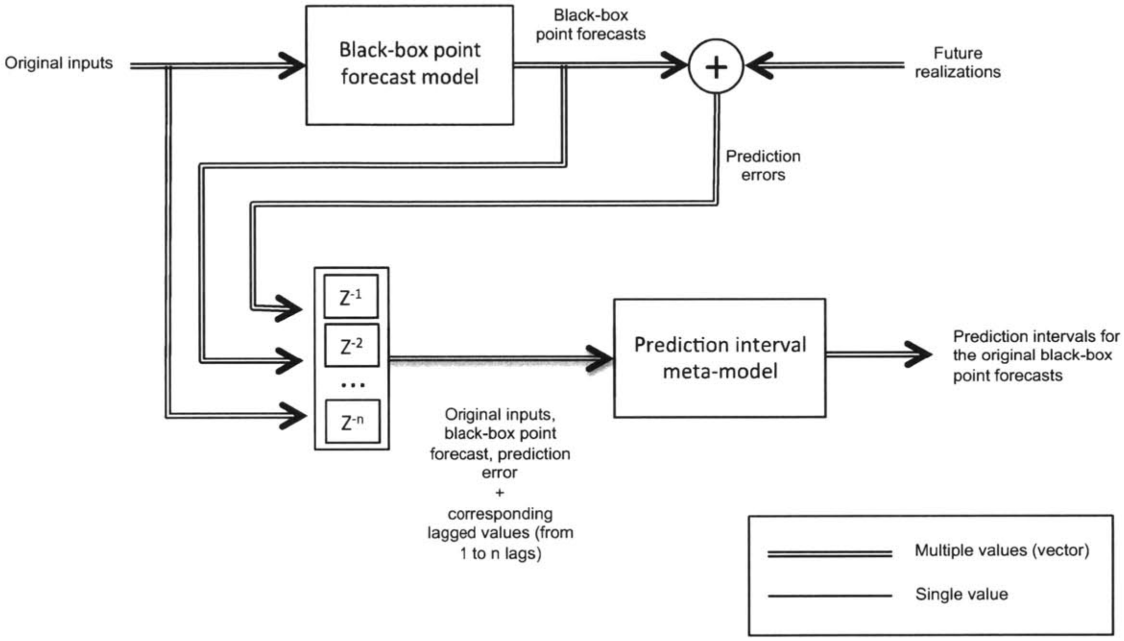
Original inputs
Black-box point forecast model
Black-box point forecasts
+
Future realizations
Prediction errors
Z^{-1}
Z^{-2}
...
Z^{-n}
Prediction interval meta-model
Prediction intervals for the original black-box point forecasts
Original inputs, black-box point forecast, prediction error + corresponding lagged values (from 1 to n lags)
Multiple values (vector)
Single value

Figure 3-12. Meta-model framework for multiple predictions and using past information.

When deciding what subset of point forecasts to use for each of the different meta-models it is important to take into consideration whether new useful information is provided and how much more complex will the model be. Although automatic feature selection methods can do the job, it is recommendable to perform previous analysis to make an educated decision, especially when the number of point forecasts is high.

If multiple step-ahead prediction intervals are needed, they are treated the same way as just explained. For each step-ahead prediction, a different meta-model is used, each potentially using all the step-ahead point forecasts as inputs.

3.6.3 Categorical variables

Very often one or more of the black-box model (and meta-model) input variables are categorical. In these cases, it is a standard practice to convert these features into vectors of binary variables. For example, if one of the inputs can have any of three possible values {A, B, C}, three new features would be used as inputs to the meta-model. Whenever the original input is A the three new features would be (1,0,0). For B, (0,1,0) would be used, and so on.

4 Case studies

In this chapter the prediction interval meta-modeling methodology described in last chapter is applied to different point forecast models and the results are analyzed. In each case, the results obtained with different quantile regression models are compared.

4.1 Freeway speed-density

Two freeway datasets from Irvine, CA, and Tel Aviv in Israel have been used for this research. In both cases, weekday data were used. The Irvine data set includes five days of sensor data from freeway I-405. Data from 10am to 12midnight have been used, since this period includes the (pm) peak flow for this direction. Speed, occupancy and flow data over 2–minute intervals were available for calibration and validation. The occupancy data has been converted to density using a relationship from May ([50], eq. 7.2 in p. 193).

The second dataset was collected in Highway 20 (Ayalon Highway), a major intra–city freeway running through the center of Tel Aviv in Israel. Speed, occupancy and flow data were available and were aggregated over 5-minute intervals. Occupancy data has been converted to density using the same relationship as above.

For each of the datasets, three different speed-density frameworks were used to determine the speed based on the density:

1. Typical speed-density relationship ([51]): $u = u_f \left[1 - \left(\frac{\max\,(0, k - k_{min})}{k_{jam}}\right)^{\beta}\right]^{\alpha}$, where u denotes the space mean speed, u_f the free flow speed, k the density, k_{min} the minimum density, k_{jam} the jam density, and β and α are model parameters.
2. Speed prediction framework presented in Antoniou et al. [52].
3. Simplified framework presented in Antoniou et al. [52].

These three approaches will be our black box speed prediction models. Henceforth, for simplicity of reference, we will call the first approach as *spddsty*; the second one will be *LOESS* (locally weighted scatterplot smoothing) and the last one will be *NNet.*

Then, the speed prediction error is calculated as the difference between the estimated and the actual measured speed. The following figures show the prediction errors for Ayalon.

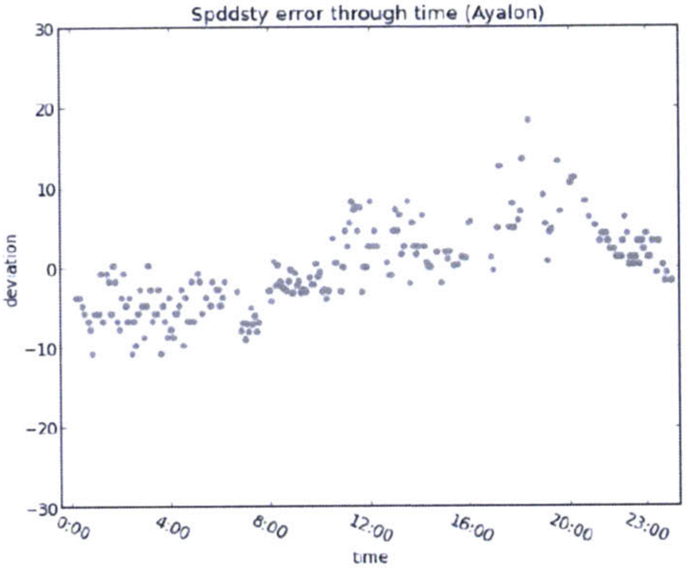

Figure 4-1. Spddst error through time (Ayalon)

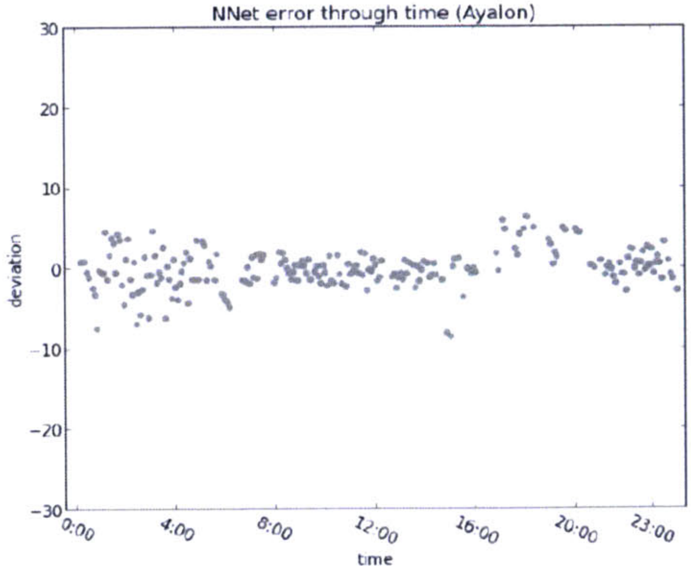

Figure 4-2. NNet error through time (Ayalon)

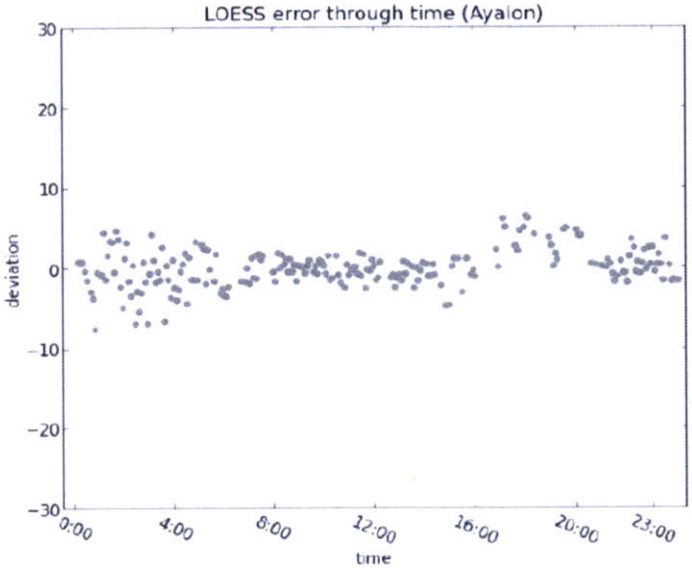

Figure 4-3. LOESS error through time (Ayalon)

In this setting, the inputs to the prediction interval meta-model are:

- Predicted speed (output of the speed-density model)
- Prediction error
- Lags of the prediction error (3 lags)

The configured prediction interval coverage was 90%.

The meta-model was implemented in Matlab and a package called *GPStuff*, which provides Gaussian process regression, including quantile regression, as well as numerous covariance functions. Linear quantile regression was handled by a different publicly available package called *quantreg*.

The entire Ayalon dataset consists of 144 training points and 71 test points, whereas the Irvine dataset consists of 270 training points and 133 test points. These numbers vary slightly depending on the speed-density model being used. The data points were randomly split.

Several different quantile models were tested for each of the datasets using the PICP and MPIL. The following tables detail the results.

4.1.1 Ayalon-spddsty

Quantile model	PICP	MPIL
Homoscedastic constant quantile	0.89	17.50
Linear	0.74	8.05
Splines	0.93	12.42
GP – Squared Exponential	0.816901	9.301496
GP – Matérn 3/2	0.577465	7.06533
GP – Matérn 5/2	0.788732	9.387622
GP – Neural network	0.71831	8.485413
GP – Exponential * Matérn 5/2	0.140845	2.740635
GP – Squared Exponential * Matérn 3/2	0.112676	3.091796
GP – Squared Exponential * Matérn 5/2	0.098592	3.417025
GP – Squared Exponential * Neural network	0.15493	3.048801
GP – Squared Exponential * Linear	0.549296	7.088714
GP – Exponential + Matérn 5/2	0.169014	3.279763
GP – Squared Exponential + Matérn 3/2	0.183099	3.178908
GP – Squared Exponential + Matérn 5/2	0.211268	4.302065
GP – Squared Exponential + Neural network	0.704225	8.246474
GP – Squared Exponential + Linear	0.704225	7.923554

Table 4-1. Results for Ayalon-spddsty

4.1.2 Ayalon-loess

Quantile model	PICP	MPIL
Homoscedastic constant quantile	0.85	8.83
Linear	0.59	4.52
Splines	0.69	7.42
GP – Squared Exponential	0.619718	3.946204
GP – Matérn 3/2	0.605634	5.17169
GP – Matérn 5/2	0.605634	3.817965
GP – Neural network	0.619718	4.078775
GP – Exponential * Matérn 5/2	0.239437	1.525469
GP – Squared Exponential * Matérn 3/2	0.338028	1.794901
GP – Squared Exponential * Matérn 5/2	0.309859	1.987211
GP – Squared Exponential * Neural network	0.521127	2.901966
GP – Squared Exponential * Linear	0.28169	2.143182
GP – Exponential + Matérn 5/2	0.408451	2.201015
GP – Squared Exponential + Matérn 3/2	0.450704	2.546883
GP – Squared Exponential + Matérn 5/2	0.591549	4.213425
GP – Squared Exponential + Neural network	0.647887	4.903392
GP – Squared Exponential + Linear	0.591549	4.171585

Table 4-2. Results for Ayalon-loess

4.1.3 Ayalon-nnet

Quantile model	PICP	MPIL
Homoscedastic constant quantile	0.86	9.53
Linear	0.66	4.44
Splines	0.69	6.46
GP – Squared Exponential	0.647887	5.646912
GP – Matérn 3/2	0.732394	6.02003
GP – Matérn 5/2	0.619718	4.173414
GP – Neural network	0.647887	4.283165
GP – Exponential * Matérn 5/2	0.352113	1.463155
GP – Squared Exponential * Matérn 3/2	0.450704	2.197863
GP – Squared Exponential * Matérn 5/2	0.408451	2.085518
GP – Squared Exponential * Neural network	0.535211	3.183974
GP – Squared Exponential * Linear	0.619718	4.896985
GP – Exponential + Matérn 5/2	0.422535	1.877202
GP – Squared Exponential + Matérn 3/2	0.56338	3.574454
GP – Squared Exponential + Matérn 5/2	0.535211	3.409899
GP – Squared Exponential + Neural network	0.690141	4.924131
GP – Squared Exponential + Linear	0.676056	4.614422

Table 4-3. Results for Ayalon-nnet

4.1.4 Irvine-spddsty

Quantile model	PICP	MPIL
Homoscedastic constant quantile	0.92	20.88
Linear	0.74	15.04
Splines	0.95	21.59
GP – Squared Exponential	0.62406	15.79928
GP – Matérn 3/2	0.917293	22.448811
GP – Matérn 5/2	0.586466	12.63717
GP – Neural network	0.902256	21.304594
GP – Exponential * Matérn 5/2	0.37594	7.590461
GP – Squared Exponential * Matérn 3/2	0.390977	7.064848
GP – Squared Exponential * Matérn 5/2	0.451128	8.177132
GP – Squared Exponential * Neural network	0.43609	8.051042
GP – Squared Exponential * Linear	0.571429	13.366558
GP – Exponential + Matérn 5/2	0.37594	7.940091
GP – Squared Exponential + Matérn 3/2	0.488722	9.626574
GP – Squared Exponential + Matérn 5/2	0.62406	12.420764
GP – Squared Exponential + Neural network	0.676692	15.254791
GP – Squared Exponential + Linear	0.684211	15.234422

Table 4-4. Results for Irvine-spddsty

4.1.5 Irvine-loess

Quantile model	PICP	MPIL
Homoscedastic constant quantile	0.88	9.69
Linear	0.82	8.79
Splines	0.81	8.67
GP – Squared Exponential	0.844262	8.318334
GP – Matérn 3/2	0.836066	8.271761
GP – Matérn 5/2	0.836066	8.477813
GP – Neural network	0.811475	7.909281
GP – Exponential * Matérn 5/2	0.401639	3.321814
GP – Squared Exponential * Matérn 3/2	0.483607	3.613976
GP – Squared Exponential * Matérn 5/2	0.442623	3.74413
GP – Squared Exponential * Neural network	0.459016	4.049936
GP – Squared Exponential * Linear	0.811475	8.7081
GP – Exponential + Matérn 5/2	0.377049	2.792096
GP – Squared Exponential + Matérn 3/2	0.57377	4.159665
GP – Squared Exponential + Matérn 5/2	0.557377	4.386076
GP – Squared Exponential + Neural network	0.827869	8.33217
GP – Squared Exponential + Linear	0.811475	8.415215

Table 4-5. Results for Irvine-loess

4.1.6 Irvine-nnet

Quantile model	PICP	MPIL
Homoscedastic constant quantile	0.88	9.56
Linear	0.87	9.50
Splines	0.88	10.05
GP – Squared Exponential	0.852459	8.722825
GP – Matérn 3/2	0.844262	8.733171
GP – Matérn 5/2	0.852459	8.735568
GP – Neural network	0.811475	8.157696
GP – Exponential * Matérn 5/2	0.467213	3.162302
GP – Squared Exponential * Matérn 3/2	0.459016	3.477804
GP – Squared Exponential * Matérn 5/2	0.459016	3.614316
GP – Squared Exponential * Neural network	0.508197	4.54075
GP – Squared Exponential * Linear	0.688525	7.682049
GP – Exponential + Matérn 5/2	0.491803	3.845364
GP – Squared Exponential + Matérn 3/2	0.663934	5.01279
GP – Squared Exponential + Matérn 5/2	0.647541	4.627044
GP – Squared Exponential + Neural network	0.819672	8.076727
GP – Squared Exponential + Linear	0.827869	8.715702

Table 4-6. Results for Irvine-nnet

4.1.7 Analysis

Although the results for the PICP and MPIL cannot be compared directly amongst the different models, it is quite visible that both splines and Gaussian processes using squared exponential or Matérn covariance functions are the best performing quantile models. From a Gaussian process point of view, the choice of the optimal covariance function or combination of functions depends very much on the dataset, and an exhaustive search is required to determine the best ones.

4.2 Incident duration in Singapore's expressways

Pereira and Ben-Akiva [53] developed a model to predict the duration of traffic incidents in Singapore based on the text description updated in near real-time about the incident. This description of the incident is created by the traffic management center in Singapore based on the inputs from the police, local traffic agents, drivers, etc. Topic modeling, a text analysis technique, is used to extract information from the descriptions. From the original text, the model is able to create scores for 25 predefined topics, which are then used together with other variables to predict the duration of the incident. Given the nature of traffic incidents new pieces of information arrive at different times and are appended to the original text description and the prediction is made again, taking into account the elapsed time since the incident started.

The dataset used to evaluate the prediction interval meta-model consists of 1913 training data points and 3471 test data points. Each data point has 39 dimensions, which include:

- The scores for 25 topics
- Elapsed time since the incident was first reported
- Time of the day (morning, noon, afternoon, evening)
- Day of the week
- Road id
- Number of blocked lanes
- Length of the queue
- Capacity reduction

- Blocked shoulder (yes/no)
- Predicted duration
- A few other descriptive variables

The following figure shows the prediction error as a function of the elapsed time.

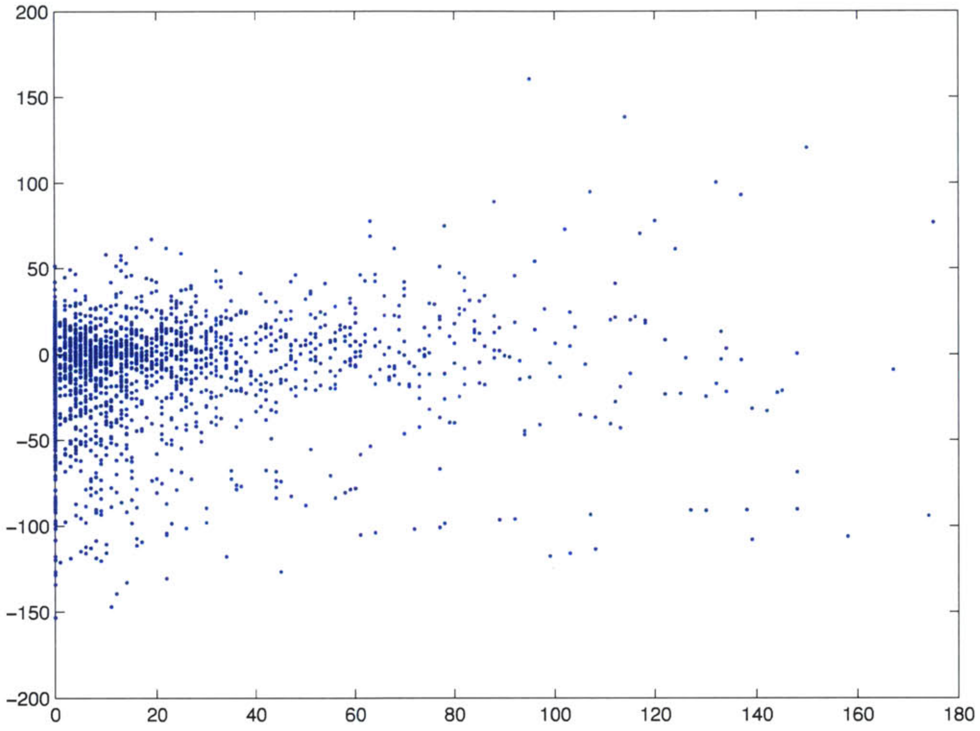

Figure 4-4. Incident duration prediction error as a function of the elapsed time

In this case, only Gaussian process regression was tested, and the obtained results were:

Quantile model	PICP	MPIL
Homoscedastic constant quantile	0.9254	107.5659
GP – Squared Exponential	0.8986	90.8857
GP – Matérn 3/2	0.751368	69.456007
GP – Matérn 5/2	0.845866	76.67643
GP – Neural network	0.761452	57.398058
GP – Exponential * Matérn 5/2	0.346586	17.179068
GP – Squared Exponential * Matérn 3/2	0.566984	34.976058
GP – Squared Exponential * Matérn 5/2	0.661769	56.962812

GP – Squared Exponential * Neural network	0.607606	44.353902
GP – Squared Exponential * Linear	0.704408	68.152062
GP – Exponential + Matérn 5/2	0.845866	76.67643
GP – Squared Exponential + Matérn 3/2	0.761452	57.398058
GP – Squared Exponential + Matérn 5/2	0.346586	17.179068
GP – Squared Exponential + Neural network	0.566984	34.976058
GP – Squared Exponential + Linear	0.661769	56.962812

Table 4-7. Results for the incident duration dataset

In this case, Gaussian processes using a squared exponential covariance function performed better than any other model.

5 Conclusions

A methodology to construct prediction intervals for a generic black-box point forecast model has been presented, where the prediction intervals are learned by an independent meta-model from the forecasts of the black-box model and the actual realizations of the forecasted variable by using quantile regression on the observed prediction error distribution, the distribution of which is not assumed.

The obtained results show that this meta-model approach is valid and its accuracy greatly depends on the underlying quantile regression model, which needs to be carefully chosen based on the data. A greedy algorithm to select the features that are most informative as well as the covariance function of the Gaussian process quantile regression is presented that can help automate this task. Given that a GP can model linear functions as well as splines and even constants, they can provide in most cases the best results.

Both Expectation Propagation (EP) and MCMC have been used to perform Gaussian process quantile regression, and the best results were obtained by using MCMC due to the problems experiences with EP in some cases (low number of data points and extreme quantiles).

The main limitation of Gaussian process quantile regression is its computational requirements, which make the use of GPs prohibitive for large datasets. In those cases, other models need to be used.

A key part of the framework presented in this thesis is the evaluation of the meta-model. Whereas simple unconditional interval coverage metrics are easy to compute and understand, conditional coverage measurements, like Christoffersen's framework and proper interval scores. Two additional methodologies, subspace division and projection onto one dimension, have been presented in this thesis.

5.1 Further research

5.1.1 Further testing

Even though the presented results are promising, further testing of the meta-model framework is required:

- on time-series models for which the theoretical prediction intervals can be derived analytically, allowing a direct comparison of the prediction intervals generated by the meta-model
- on multivariate forecast models with multiple outputs, like traffic prediction models like DynaMIT
- on models that use lagged data
- conditional coverage testing

5.1.2 Conditional coverage evaluation

The evaluation of the conditional coverage of prediction intervals for multivariate settings is a topic that requires much more research and study. Some simple ideas have been introduced in this thesis, but more advanced solutions would have a potential broad use, especially in fields like finance and atmospheric science.

5.1.3 Gaussian processes

Perhaps the most immediate improvement to the Gaussian process quantile regression framework used in this thesis is the use of fast approximations. Research on how these methodologies affect the quality of the prediction intervals would be needed.

With regards to the selection of the covariance function, there is an increasing interest in finding automated procedures to select those covariance functions that can best describe the data and there's room for a lot of research in this field.

6 References

[1] R. T. Baillie and T. Bollerslev, "Prediction in dynamic models with time-dependent conditional variances," *Journal of Econometrics*, vol. 52, no. 1–2, pp. 91–113, Apr. 1992.

[2] I. T. Jolliffe and D. B. Stephenson, *A Practitioner ' s Guide in Atmospheric Science.* John Wiley & Sons, 2003.

[3] C. Chatfield, "Prediction Intervals." 1998.

[4] S. S. Wilks, *Mathematical Statistics*. New York: Wiley, 1962.

[5] B. Kirkwood and J. Sterne, *Essential Medical Statistics*, Second. New York: Wiley, 2003.

[6] T. Heskes, "Practical confidence and prediction intervals," *Advances in Neural Information Processing Systems*, vol. 9, pp. 176–182, 1997.

[7] D. R. Cox, "Prediction intervals and empirical Bayes confidence intervals," *Perspectives in Probability and Statistics*, pp. 45–55, 1975.

[8] D. N. Gujarati, *Basic Econometrics*. McGraw-Hill, 2004.

[9] P. F. Christoffersen, "Evaluating Interval Forecasts," *International Economic Review*, vol. 39, no. 4, pp. 841–862, 1998.

[10] G. Papadopoulos, P. J. Edwards, and a F. Murray, "Confidence estimation methods for neural networks: a practical comparison.," *IEEE transactions on neural networks*, vol. 12, no. 6, pp. 1278–87, Jan. 2001.

[11] J. H. Zhao, Z. Y. Dong, Z. Xu, and K. P. Wong, "A Statistical Approach for Interval Forecasting of the Electricity Price," *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 267–276, 2008.

[12] A. A. Ding and J. T. G. Hwang, "Prediction Intervals for Artificial Neural Networks," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 748–757, 1997.

[13] J. R. Donaldson and R. B. Schnabel, "Computational Experience With Confidence Regions and Confidence for Nonlinear Intervals Least Squares," *Technometrics*, vol. 29, no. 1, pp. 67–82, 1987.

[14] E. S. Gardner, "A Simple Method of Computing Prediction Intervals for Time Series Forecasts," *Management Science*, vol. 34, no. 4, pp. 541–546, Apr. 1988.

[15] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances.," *IEEE transactions on neural networks*, vol. 22, no. 9, pp. 1341–56, Sep. 2011.

[16] C. W. J. Granger, H. White, and M. Kamstra, "Interval forecasting, an analysis based upon ARCH-quantile estimators," *Journal of Econometrics*, vol. 40, pp. 87–96, 1989.

[17] J. Geweke, "Bayesian Inference in Econometric Models Using Monte Carlo Integration," *Econometrica*, vol. 57, no. 6, pp. 1317–1339, 1989.

[18] F. X. Diebold, "Conditional Heteroskedasticity in Economic Time Series," in *Empirical Modeling of Exchange Rate Dynamics*, S. B. Heidelberg, Ed. 1988, pp. 4–32.

[19] D. F. Kraft and R. F. Engle, "Autoregressive Conditional Heteroskedasticity in Multiple Time Series." 1983.

[20] J. J. Reeves, "Bootstrap prediction intervals for ARCH models," *International Journal of Forecasting*, vol. 21, no. 2, pp. 237–248, 2005.

[21] E. Mazloumi, G. Rose, G. Currie, and S. Moridpour, "Prediction intervals to account for uncertainties in neural network predictions: Methodology and application in bus travel time prediction," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 3, pp. 534–542, 2011.

[22] A. Khosravi, E. Mazloumi, S. Nahavandi, D. Creighton, and J. Van Lint, "Prediction intervals to account for uncertainties in travel time prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 537–547, 2011.

[23] B. Hansen, "Autoregressive conditional density estimation," *International Economic Review*, vol. 35, pp. 705–730, 1994.

[24] A. S. Tay and K. F. Wallis, "Density forecasting: a survey," *Journal of Forecasting*, vol. 19, no. 4, pp. 235–254, Jul. 2000.

[25] W. H. Williams and M. L. Goodman, "A simple method for the construction of empirical confidence limits for economic forecasts," *Journal of the American Statistical Association*, vol. 66, pp. 752–754, 1971.

[26] S. Makridakis, M. Hibon, E. Lusk, and M. Belhadjali, "Confidence intervals: An empirical investigation of the series in the M-competition," *International Journal of Forecasting*, vol. 3, pp. 489–508, 1987.

[27] J. S. Armstrong and F. Collopy, "Prediction intervals for extrapolation of annual economic data: Evidence on asymmetry corrections." 1997.

[28] J. W. Taylor and D. W. Bunn, "A Quantile Regression Approach to Generating Prediction Intervals," *Management Science*, vol. 45, no. 2, pp. 225–237, 1999.

[29] J. W. Taylor, "Forecasting daily supermarket sales using exponentially weighted quantile regression," *European Journal of Operational Research*, vol. 178, no. 1, pp. 154–167, Apr. 2007.

[30] R. F. Engle and S. Manganelli, "CAViaR: Conditional Autoregressive Value at Risk by Regression Quantiles." 1999.

[31] Z. Xiao and R. Koenker, "Conditional quantile estimation for garch models." pp. 1–37, 2009.

[32] W. L. Gorr and C. Hsu, "An Adaptive Filtering Procedure for Estimating Regression Quantiles," *Management Science*, vol. 31, no. 8, pp. 1019–1029, 1985.

[33] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, vol. 14, no. 2. MIT Press, 2006.

[34] R. Koenker and G. B. Jr., "Regression Quantiles," *Econometrica: journal of the Econometric Society*, pp. 33–50, 1978.

[35] R. Koenker, *Quantile Regression*. Cambridge University Press, 2005.

[36] W. S. Cleveland, S. J. Devlin, S. Cleveland, and S. J. Devlin, "Locally Weighted Regression : An Approach to Regression Analysis by Local Fifing," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988.

[37] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[38] K. Yu and M. C. Jones, "Local linear quantile regression," *Journal of the American Statistical Association*, vol. 93, no. 441, pp. 228–237, 1998.

[39] A. Boukouvalas, D. Cornford, and R. Barillec, "Direct Gaussian Process Quantile Regression using Expectation Propagation," *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[40] T. P. Minka, "A family of algorithms for approximate Bayesian inference," Massachusetts Institute of Technology, 2001.

[41] K. Yu and R. A. Moyeed, "Bayesian quantile regression," *Statistics & Probability Letters*, vol. 54, pp. 437–447, 2001.

[42] V. Chernozhukov, "Extremal quantile regression," *The Annals of Statistics*, vol. 33, no. 2, pp. 806–839, Apr. 2005.

[43] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[44] A. E. Gelfand and A. F. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 398–409, 1990.

[45] P. A. Moran, "Notes on continuous stochastic phenomena," *Biometrika*, vol. 37, no. 1–2, pp. 17–23, 1950.

[46] D. a. Griffith, "Spatial Autocorrelation." 2005.

[47] T. Gneiting and A. E. Raftery, "Strictly Proper Scoring Rules, Prediction, and Estimation," *Journal of the American Statistical Association*, vol. 102, no. 477, pp. 359–378, Mar. 2007.

[48] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[49] M. Ben-Akiva, Bierlaire, H. M., Koutsopoulos, and R. Mishalani, "DynaMIT: a simulation-based system for traffic prediction," *DACCORS Short Term Forecasting Workshop, The Netherlands*, 1998.

[50] A. May, *Traffic Flow Fundamentals*. New Jersey: Prentice Hall, 1990.

[51] M. Ben-Akiva, H. N. Koutsopoulos, C. Antoniou, and R. Balakrishna, "Traffic Simulation with DynaMIT," in *Fundamentals of Traffic Simulation*, J. Barceló, Ed. New York: Springer, 2010, pp. 363–398.

[52] C. Antoniou, H. N. Koutsopoulos, and G. Yannis, "Dynamic data-driven local traffic state estimation and prediction," *Transportation Research Part C: Emerging Technologies*, vol. 34, pp. 89–107.

[53] F. C. Pereira, F. Rodrigues, and M. Ben-Akiva, "Text analysis in incident duration prediction," *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 177–192, Dec. 2013.