Machine Learning for Applications in Chemical and Biological Engineering

by

Kristen Ann Severson

B.S. Chemical Engineering with an additional major in French and Francophone Studies, Carnegie Mellon University (2011)

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Chemical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Signature redacted

Author ....
Department of Chemical Engineering
April 12, 2018

Signature redacted

Certified by........................
Richard D. Braatz
Edwin R. Gilliland Professor
Thesis Supervisor

Signature redacted

Accepted by ....
Patrick S. Doyle
Chairman, Department Committee on Graduate Theses

Machine Learning for Applications in Chemical and Biological Engineering

by

Kristen Ann Severson

Submitted to the Department of Chemical Engineering
on April 12, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Chemical Engineering

Abstract

Chemical and biological systems are increasingly implemented with advanced sensor systems that collect large amounts of data. For example, a single microarray can measure thousands of genes and a typical offshore oil platform generates 1 to 2 TB of data per day. New algorithms are needed to efficiently and effectively use these datasets to increase predictive capability and improve system understanding. In this thesis, algorithmic advances to bridge the gap between data and system insights are addressed in a series of case studies.

In the first case study, the problem of predicting critical quality attributes for a monoclonal antibody using data from the manufacturing process is addressed. In this setting, the main challenge is that there is only a limited dataset available for modeling. To tackle this issue, Monte Carlo sampling was used in conjunction with an elastic net approach to subset selection.

The second case study is also within the biological domain but considers a discrete outcome. The proposed algorithm addresses two common issues when building classification models for biological studies: learning a sparse model, where only a subset of a large number of possible predictors is used, and training in the presence of missing data. The resulting algorithm leverages expectation-maximization to tackle both issues simultaneously.

In the third case study, the goal was to identify anomalous operating periods using production data from an oil and gas well without access to historical examples of such periods. The proposed approach recasts the problem as a semi-supervised problem and leverages approaches from the positive and unlabeled literature.

The final case study considers the task of prediction lithium-ion battery cycle life. Cycle life is defined as the number of charge and discharge cycles the battery undergoes before 80% capacity fade. Several, difficult to identify factors can contribute to capacity fade. Even in batteries with the same chemistry, operated using the same conditions, there is considerable cycle life variability. Therefore, the challenge was to build a model to capture individual capacity trajectories.

Each case study is benchmarked using state-of-the-art approaches. In all settings, the value of data-driven methods is demonstrated.

Thesis Supervisor: Richard D. Braatz
Title: Edwin R. Gilliland Professor

Acknowledgments

I am extremely grateful to have had the opportunity to pursue my doctoral work at MIT. I have benefited from the community and opportunities here from coursework to casual research conversations. MIT is a unique place and I consider myself lucky to have been able to be a part of it.

Many people have contributed to my experience during my PhD. I would like to first thank my advisor, Richard D. Braatz. Richard created a balance of direction and freedom in pursuing my research. He has also provided candid career advice which has been invaluable in shaping my career plans. I would also like to thank my thesis committee members George Stephanopoulos and Paul I. Barton for valuable conversations and contributions to my work.

I have been fortunate to collaborate with many people over the course of my thesis work. At MIT, I would like to thank Jeremy G. VanAntwerp, Mark C. Molaro, Brinda Monian, and Wit Chaiwatanodom. Outside of MIT, I would like to thank Venkatesh Natarajan (Biogen), Richard S. Bailey (BP), Peter Attia (Stanford), Norman Jin (Stanford), Will Chueh (Stanford), and Stephen Harris (Lawrence Berkeley National Laboratory). Each of these collaborators has provided valuable insight and helped improve my work.

I have enjoyed my time in the Braatz lab and am grateful to all of my labmates for making my day-to-day work enjoyable. In particular I would like to thank Amos Lu, who has been my officemate for all five years. It was not easy to transition back to the academic world and I am grateful to my first-year classmates who provided support on problem sets and exam prep. I have served on the Graduate Student Advisory Board since my second semester and have really enjoyed contributing to the department. I have also benefited from MIT Group Exercise classes and Student Art Association to give myself a break from research.

I have been fortunate to receive support from several mentors throughout my career who supported my decision to pursue my PhD. In particular, I would like to thank Ignacio Grossmann and Mariano Martín Martín whom I worked with on my senior research at Carnegie Mellon. I would also like to thank X.B. Cox at ExxonMobil for encouraging me to pursue graduate school.

Finally, I would like to thank my friends and family for their support throughout. It hasn't always been to clear to them what I am working on, but they have provided their support regardless. I am especially grateful to my husband, Max Jordan, who has been my greatest supporter from my application to my defense.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Data science for engineering applications

The combination of improved computational power, decreased data storage costs, advances in algorithms, and new sensor technologies has led to a resurgence of interest in data-driven methods. Complex systems, which are characterized by a lack of a full physics-based description and/or high levels of uncertainty, are well-suited for data-driven approaches because other methodologies aren't readily available. This thesis focuses on the application of these methods to chemical and biological engineering problems.

The application of data-driven methods, also referred to as machine learning, to chemical and biological engineering problems has unique challenges. First, the datasets are often small because of the cost associated with generating them. Second, the cost of making an error is often high, either in financial or safety terms. Finally, there is typically chemistry/physics/physiology governing the underlying system. These factors imply that an ideal modeling approach should be able to work in settings with small amounts of data, capture the underlying science and result in an interpretable model. These goals are explored in this thesis.

The next section introduces some of the core concepts of machine learning, which will be repeated throughout the thesis. In it, statistical learning theory, probabilistic graphical models, and the bias-variance trade-off are discussed. The final section of this chapter provides a summary of the rest of the thesis chapters.

1.2 Overview of the principles of machine learning

Fundamentally, machine learning is a field dedicated to using data to make predictions and/or improve system understanding. There are many ways to sub-divide the field to relate various problems. One such sub-division is into *supervised*, *unsupervised*, and *semi-supervised* problems. In this framework, problems are organized based on whether or not the prediction task outcomes have been recorded. For instance, the task of correlating measured genes to an observed phenotype is a supervised problem, however the task of finding unexpected patterns in oil well monitoring data is an unsupervised problem. When the dataset is a combination of known and unknown outcomes, the problem is called semi-supervised.

Within the set of supervised problems, there is another common sub-division into regression and classification problems. These two classes are differentiated based on the type of outcome: a continuous outcome gives rise to a regression problem and a discrete problem gives rise to a classification problem. Often these problems can be interchanged however one approach is typically better suited to the domain application than the other.

Another possible division of problems is into *parametric* and *non-parametric* models. Parametric models have a fixed number of parameters and typically make stricter distributional assumptions. Non-parametric models do not fix the parameters but instead grow with the amount of training data. Therefore, non-parametric models are more flexible but can be computationally intractable. This poor performance in a high dimensional setting is often referred to as the curse of dimensionality.

The field of machine learning is still growing. Machine learning draws on ideas from statistics, probability theory, information theory, and decision theory. Two common frameworks for analyzing machine learning problems are presented here: statistical learning theory and probabilistic graphical models.

1.2.1 Statistical learning theory

Statistical learning theory is one framework for analyzing machine learning problems with a focus on supervised machine learning problems. A machine learning problem in the statistical learning theory framework typically has three main components: a probability space $\mathcal{X} \times \mathcal{Y}$ with probability measure ρ, a loss function $L : \mathcal{Y} \times \mathcal{Y} \rightarrow [0, \infty)$ that is a measure of success, and dataset $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where the data are assumed to be independent and identically distributed with respect to ρ. The problem is then formulated as an optimization to minimize the expected risk

$$\min_{f:X \rightarrow \mathcal{Y}} \mathcal{E} = \min_{f:X \rightarrow \mathcal{Y}} \int_{\mathcal{X} \times \mathcal{Y}} L(y, f(x)) d\rho(x, y) \tag{1.1}$$

while searching over all possible functions. This problem is intractable and therefore simplifications are made based on the specific problem of interest. Typically the functional space is restricted, often to reproducing kernel Hilbert spaces (RKHS), and the optimization uses the empirical risk as opoosed to the expected risk which is defined

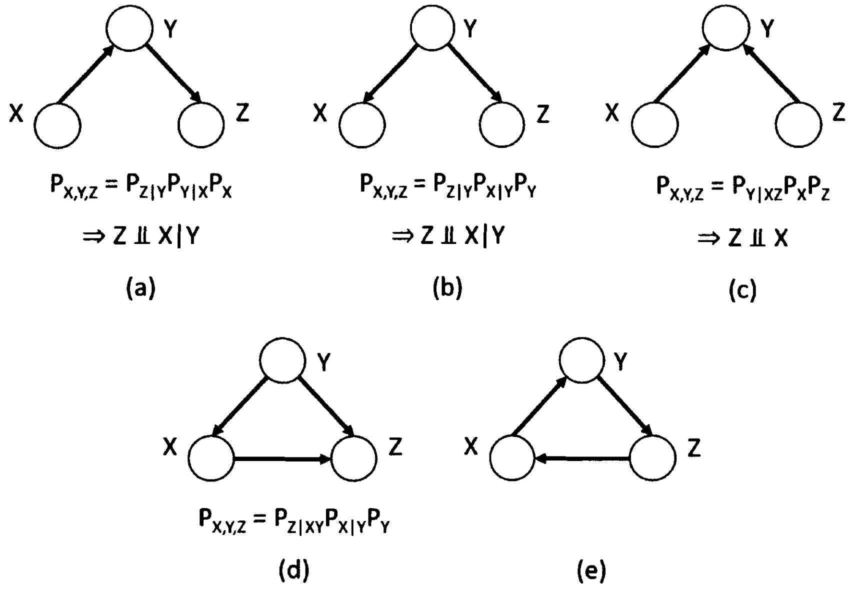
$$\hat{\mathcal{E}} = \frac{1}{n} \sum_{i=1}^{n} L(\hat{f}(x_i), y_i) \tag{1.2}$$

where $\hat{f}$ is a function of the dataset used in training. The complexity of the functions in the RKHS can be controlled using regularization. To perform regularization, penalty terms are added to the optimization objective.

$$\min_{f \in \mathcal{H}} \hat{\mathcal{E}} + \lambda R(f) \tag{1.3}$$

The popular techniques of ridge (Tikohonov) regression, lasso [246], and elastic net [302] can all be derived from this framework. Each uses a linear regression estimator, an ℓ_2 loss function and regularization penalties on the coefficient vector. In the case of ridge regression

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \frac{1}{n} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 \tag{1.4}$$



Y
X
Z
$P_{X,Y,Z} = P_{Z|Y}P_{Y|X}P_X$
$\Rightarrow Z \perp\!\!\!\perp X|Y$
(a)
Y
X
Z
$P_{X,Y,Z} = P_{Z|Y}P_{X|Y}P_Y$
$\Rightarrow Z \perp\!\!\!\perp X|Y$
(b)
Y
X
Z
$P_{X,Y,Z} = P_{Y|XZ}P_XP_Z$
$\Rightarrow Z \perp\!\!\!\perp X$
(c)
Y
X
Z
$P_{X,Y,Z} = P_{Z|XY}P_{X|Y}P_Y$
(d)
Y
X
Z
(e)

Figure 1-1: Examples of the relationships between three random variables that can be captured by directed acyclic graphs. Examples a-c show possible constraints. Example d is a fully connected graph and therefore does not have any associated independence statements. Example e is not acyclic and is therefore not a DAG.

where the prediction function is $\hat{y}_i = \mathbf{w}^T\mathbf{x}_i$, $\mathbf{y}$ is an n-dimensional vector of observed outcomes, $\mathbf{X}$ is an $n \times p$ matrix of input training data, $\mathbf{w}$ is a p-dimensional vector of coefficient weights, and λ is a non-negative scalar. Lasso uses the penalty term $\lambda||\mathbf{w}||_1$ and elastic net uses a linear combination of both forms of penalty. All regularization techniques are used to prevent over-fitting (discussed in Section 1.2.3). In the case of ridge regression, the model coefficients are biased towards zero. In lasso and elastic net, because of the functional form of the penalty, the resulting coefficient vector is more likely to be sparse, i.e. some of the values of $\mathbf{w}$ are exactly zero. This implies that the lasso and elastic net techniques perform model fitting and model selection simultaneously. These tools are used throughout the thesis.

1.2.2 Probabilistic graphical models

Probabilistic graphical models are a set of tools for machine learning problems. While not completely distinct from statistical learning theory, given the topics covered in this thesis, probabilistic graphical models merit their own introduction. Probabilistic graphical models exploit the structure of a problem to perform inference and learning tasks. In this thesis, directed acyclic graphs (DAGs) are used. DAGs are a collection of nodes, used to represent random variables, and edges, used to represent the relationships between the random variables (see Fig. 1-1). The DAG imposes constraints on the family of distributions that can be used to describe the random variables.

DAGs are a class of parametric models. Although there are methods for learning the structure of a DAG, often the structure is assumed based on the specific application. Then, given a dataset, maximum likelihood estimates of the parameters of the distributions can be made.

$$\theta^* = \arg\max_\theta \sum_{i=1}^{n} \ln p(\mathbf{x}_i|\theta) \tag{1.5}$$

where θ^* are the optimal parameters of the distribution, $\mathbf{x}_i, i = 1, \dots, n$ are the data, and $\sum_{i=1}^{n} \ln p(\mathbf{x}_i|\theta)$ is the log-likelihood function. In this thesis, DAGs are applied without full observations, i.e. there is data missing from the training dataset. To find the maximum likelihood estimate, the likelihood function should be marginalized over the missing data, $\mathbf{z}_i$. The log-likelihood is then

$$\ell(\theta) = \sum_{i=1}^{n} \ln \left[\sum_{\mathbf{z}_i} p(\mathbf{x}_i, \mathbf{z}_i|\theta) \right] \tag{1.6}$$

The marginalization is typically intractable and maximum likelihood estimates cannot be made. Expectation-maximization (EM) is one approach for estimating parameter values when some data is missing. The complete data log-likelihood is

$$\ell_c = \sum_{i=1}^{n} \ln p(\mathbf{x}_i, \mathbf{z}_i|\theta) \tag{1.7}$$

but this quantity cannot be computed because $\mathbf{z}_i$ is not observed. Instead, the expected complete data log-likelihood is considered

$$Q(\theta, \theta^{t-1}) = \mathbb{E}[\ell_c(\theta)|\mathcal{D}, \theta^{t-1}] \tag{1.8}$$

The choice of expected complete data log-likelihood is motivated by an analysis of the log-likelihood function in the presence of missing data (Eqn. 1.6). The likelihood function can be bounded by applying Jensen's inequality

$$\ell(\theta) \geq \sum_{i=1}^{n} \sum_{\mathbf{z}_i} q(\mathbf{z}_i) \ln \frac{p(\mathbf{x}_i, \mathbf{z}_i | \theta)}{q(\mathbf{z}_i)} \tag{1.9}$$

where $q(\mathbf{z}_i)$ is a distribution over the missing data. This lower bound can be rewritten

$$Q(\theta, q) = \sum_i \mathbb{E}_{q_i}[\ln p(\mathbf{x}_i, \mathbf{z}_i|\theta)] + \mathbb{H}(q_i) \tag{1.10}$$

where $\mathbb{H}(q_i)$ is the entropy of q_i. It can be shown that the best choice of $q(\mathbf{z}_i)$, in terms of finding the tightest lower bound, is $p(\mathbf{z}_i|\mathbf{x}_i, \theta)$. Because θ is unknown, it is replaced with the current parameter estimate θ^t, where t is the iteration step. Plugging this back into the lower bound gives the expected complete data log-likelihood plus the entropy term which is not a function of θ.

$$Q(\theta, \theta^{t-1}) = \mathbb{E}[\ell_c(\theta)|\mathcal{D}, \theta^{t-1}] + \mathbb{H}(q_i) \tag{1.11}$$

Because the expected complete data log-likelihood provides a lower bound, the second step of the EM algorithm is to then update the parameter values by maximizing the value of this bound

$$\theta^{t+1} = \arg\max_{\theta} Q(\theta, \theta^t) = \arg\max_{\theta} \sum_i \mathbb{E}_{q_i}[\ln p(\mathbf{x}_i, \mathbf{z}_i|\theta)] \tag{1.12}$$

This optimization is guaranteed to change the parameters, θ, such that the likelihood of the observed data increases (or the function is already at a fixed point). The EM procedure is only guaranteed to find a local maximum or saddle point (however it has been observed that many saddle points are unstable). The expectation-maximization algorithm was introduced in its general form by [56], however applications of the algorithm can be found in many publications, e.g. [13, 158, 146]. The description of the EM algorithm provided here is based on Chapter 11 of Machine Learning: A

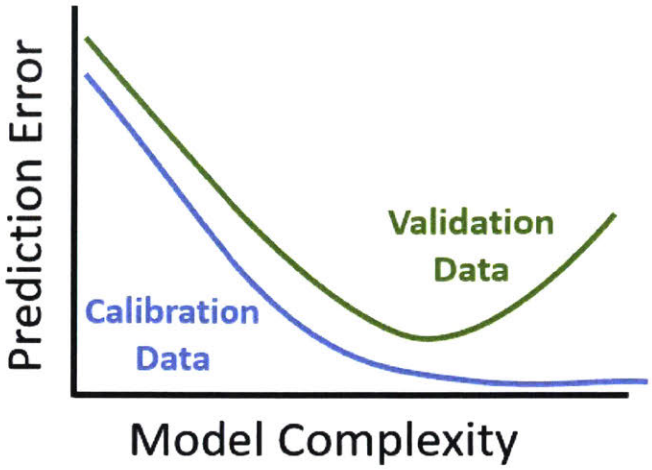


Figure 1-2: Depiction of the expected behavior for the calibration, data used to learn a model, and validation, data used to test a model, datasets. Initially the error decreases as the model captures the underlying system. The validation data prediction error increases when the model becomes overfit to the calibration data. Depiction is based on [93].

Probabilistic Perspective [169].

1.2.3 Over-fitting and the bias-variance trade-off

A common concern when fitting a data-driven model is *overfitting*. Overfitting refers to scenarios where the model has low error for the training data, i.e. the data used to learn the model, but does not *generalize* well because the model form is too complex.

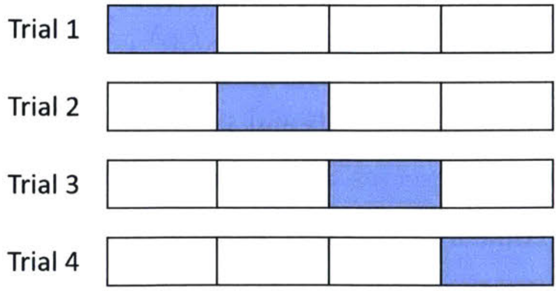


Figure 1-3: A schematic of 4-fold cross-validation. In each trial, a fourth of the data, indicated in blue, is used for validation. The model complexity is varied during each trial. The four trials are then averaged and the final model complexity is set based on the results. Depiction is based on [16].

In other words, when the model is applied to new data, or testing data, the errors are much higher than expected based on the training phase. Alternatively, models can also be *underfit*, where the model does not have enough complexity to capture the underlying phenomenon. In this case, errors in both the training data and testing data are higher than would be achieved with an appropriately selected model.

Cross-validation is one method to select the model complexity. In cross-validation, the training data is subdivided into calibration and validation sets. The calibration data is used to set the model parameters and the validation data is used to evaluate the errors. The calibration data is re-used with varying model complexity and each time the validation error is recorded. The result typically resembles Fig. 1-2, where a minimum validation data error is achieved for some complexity value. Often *k-fold cross validation* is used, where $1/k$th of the data is used for validation and the remaining data is used for calibration. k-Fold cross validation iterates through each of the sets and averages the validation errors. An example using 4-fold cross-validation is shown in Fig. 1-3.

An alternative to cross-validation is Bayesian inference. In Bayesian inference, prior distributions of the parameters are selected and the final parameter setting is based on the posterior. The name of this approach is in reference to Bayes' rule (also called Bayes' Theorem)

$$P(X = x|Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)} = \frac{P(X = x)P(Y = y|X = x)}{\sum_{x'} P(X = x')P(Y = y|X = x')} \tag{1.13}$$

which combines the definition of conditional probability with the product and sum rules. In this case, the prior serves as a guard against overfitting. Selecting an appropriate prior can be difficult and will depend on the amount of system knowledge. If a probabilistic model is chosen and a suitable prior is available, there may not be an analytical form of the posterior and sampling methods are typically employed. Cross-validation is used as the approach to select model complexity throughout this thesis.

1.3 Thesis objectives

This thesis explores ways to address some of the particular issues that arise when applying machine learning approaches to chemical and biological engineering problems. Each project addresses one or more of the three highlighted challenges: small datasets, interpretability, and an effort to capture the physical system knowledge. This is done through a series of applications. The thesis also includes reviews and tutorials to complement the analysis.

1. Chapter 2 introduces a systematic approach to data pre-processing in the context of understanding which approaches are fit for the modeling purpose. It focuses on the pharmaceutical industry.

2. Chapter 3 draws on the ideas presented in Chapter 2 and focuses on biopharmaceutical manufacturing settings where the amount of data available for modeling is often limited. The proposed approach uses sparse regression models and Monte Carlo sampling. The resulting models are shown to be simpler and more accurate than principal component regression and partial least squares models, techniques which are commonly employed in industry.

3. Chapter 4 surveys applications of principal component analysis in the presence of missing data and demonstrates the advantages and shortcomings of each on process data from the Tennessee Eastman Simulation.

4. Chapter 5 presents a new technique for binary classification with simultaneous feature selection. The method is designed such that it can easily be extended to scenarios where there are missing data by using expectation-maximization. The approach is demonstrated on three high-dimensional biological datasets.

5. Chapter 6 surveys the processing monitoring literature.

6. Chapter 7 explores an application of semi-supervised learning for production oil and gas well anomaly detection. The main problem challenge is that there are no historical examples of anomalies. However, it is shown that by labeling

a small number of nominal examples, a dramatic improvement in detection is achieved. The approach is demonstrated on data from the field.

7. Chapter 8 contains models for the accurate prediction of lithium-ion battery cycle life using data from a high-throughput cycling platform. Cycle life prediction is challenging because a number of capacity fade mechanisms may contribute to decaying performance. Furthermore, even for batteries with the same chemistry, operated in the same conditions, there is considerable variability. The proposed method achieves accurate prediction by considering the trajectory of the discharge curve.

Each case study compares the results of the proposed approach with state-of-the-art methods in the field. These case studies demonstrate the value of data-driven modeling for engineering applications. Conclusions are presented in Chapter 9.

Chapter 2

A systematic approach to data analysis in biomanufacturing

This work originally appeared as: Kristen A. Severson, Jeremy G. VanAntwerp, Venkatesh Natarajan, Chris Antoniou, Jörg Thömmes, and Richard D. Braatz. "A systematic approach to process data analytics in pharmaceutical manufacturing: The data analytics triangle and its application to the manufacturing of a monoclonal antibody." Multivariate Analysis in the Pharmaceutical Industry. Eds. A.P. Ferreira, J.C. Menezes, and M. Tobyn. In press. It has been edited to refer directly to Chapter 3 of this thesis.

2.1 Background

Data analytics refers to a set of techniques for transforming and modeling data with the objectives of discovering useful relationships and supporting decision making. Although data analytics is sometimes broadly used to include models informed by conservation equations, constitutive rate expressions, reaction networks, or other first-principles or mechanistic understanding in addition to experimental data, typically the term data analytics is used to include only models constructed completely from experimental data, which is the usage taken here. Big data analytics is a term that is widely in the data analytics field and is relevant to manufacturing processes [205, 229], but has unique challenges that are not discussed here.

Process data analytics (aka *process analytics*) are those techniques found to be useful for the analysis of data from manufacturing processes, regardless of whether the underlying phenomena are primarily biological or chemical. Process data analytics have become widely applied in the pharmaceutical industry for both chemically and biologically derived drug products. While models for chemically derived drug products, commonly referred to as small-molecule drugs, can be constructed using first-principles understanding (e.g., [129, 148]), too many molecular species are typically present in most processes in biologic drug manufacturing - especially in bioreactors and the multiple chromatography columns - to enable a first-principles model of all of the molecular species. Although significant efforts have been directed towards increasing the use of, and improving the prediction capability of, first-principles modeling in biologic drug manufacturing (e.g., [97, 149] and citations therein), process models constructed purely from experimental data are expected to be most widely used in biologic drug manufacturing for the near future.

Biologic drug products include monoclonal antibodies (mAbs), hormones, growth factors, fusion proteins, cytokines, therapeutic enzymes, blood factors, recombinant vaccines, and anticoagulants. Biologic drug products have had double-digit growth rates for many years, with mAbs being the largest category, constituting approximately 39% of biologic drug sales [1]. The pharmaceutical industry is expected to continue to shift towards increased production of biologic drugs, and mAbs in particular, for the foreseeable future. As the number of products grows, there is interest by both manufacturers and regulatory bodies to increase the use of models as a way to increase understanding and to more quickly bring new products to patients [104, 256].

As mentioned, the biopharmaceutical industry has some unique challenges for achieving process understanding: some of the processes are complex, often datasets are small and heterogeneous, within a dataset, measurements are usually collected at different sampling rates [39]. A framework is presented for analyzing experimental laboratory- and production-scale pharmaceutical manufacturing data using process data analytic techniques, which is applicable to either small-molecule or biologic drug products. Given the increased challenge associated with biopharmaceuticals, the key steps and points are illustrated for the manufacturing of a monoclonal antibody. Additionally, common pitfalls and mistakes are identified.

2.2 The data analytics triangle

Mathematical models for pharmaceutical processes can serve a variety of purposes. When the objective is to make predictions, such as the early identification of bad batches, or to modify recipes for downstream processes to improve product quality, the desired models relate *critical process parameters* (CPPs) to *critical quality attributes* (CQAs). For such purposes, the models can be dense, in which each CQA is a function of all of the CPPs, or *sparse*, in which each CQA is only a function of a small number of CPPs needed to predict the CQAs. Another purpose of such models for a process is for use in control, that is, to compute adjustments to the CPPs to move the CQAs towards desirable values. For such purposes, it can be advantageous, from a regulatory point of view, to minimize the number of process changes, in which case sparse models are preferable. A third purpose of mathematical models is to improve process understanding, which can lead to improved troubleshooting capabilities during manufacturing and improved control of the CQAs through long-lasting changes in process operating protocols. Considering this last purpose, model interpretability is given priority along with model accuracy, and sparse models are preferable.

Many specific data analytics techniques can be used to construct models to serve these purposes, and it is common practice for the data analyst to try either a single or small number of favorite techniques or a try a variety of techniques until a technique seems to give good results. A more systematic approach is to interrogate the data to discover which technique or class of techniques to apply. The decision about which technique to use should be based on the characteristics of the data. Interrogating the data to determine their characteristics is an efficient approach to direct the data analyst quickly to the technique(s) most suited to the particular dataset under consideration.

When building models from data, it is useful to categorize datasets in terms of three characteristics: correlation, nonlinearity, and dynamics [229]. After measuring the extent of each of these characteristics in the dataset, the classes of techniques most suitable to the dataset are indicated on the data analytics triangle shown in Fig. 2-1. The next section describes how to measure the extent of correlation and nonlinearity within a dataset, with examples taken from process development data collected at the laboratory scale for the manufacturing of a mAb. The extent of dynamics can be quantified between scalar variables and within single variables using serial cross-correlation and autocorrelation, respectively, which are available in MATLAB or any time-series analysis or signal-processing software package. Canonical variate analysis is a method for characterizing the extent of dynamics that takes all variables into account (e.g., see [47] and citations therein), and can be used to characterize both the extent of dynamics and the extent of correlation within a single tool.

The techniques in the data analytics triangle are selected based on the extent of the three characteristics, with the triangle labeled with representative examples of the techniques that are most appropriate. For example, consider the vertices of the triangle. If a dataset contains significant nonlinearity but minimal correlations and dynamics, then surface response methodology is one of the best techniques for building the model. If a dataset contains significant correlation but negligible nonlinearity and dynamics, then techniques such as partial least squares and principal component regression are some of the best techniques provided that a dense model is desired, whereas lasso and elastic net are some of the best techniques for the construction of sparse models from correlated data. If the primary characteristic of a dataset is dynamic linear relationships between scalar variables, then autoregressive moving average models are most appropriate.

If a dataset contains two of the three characteristics, then the techniques listed on the edge connecting the points for the two characteristics are most appropriate. For example, canonical variate analysis is one of the best techniques for data that have correlations and dynamics. If a dataset contains significant correlation and nonlinearity but minimal dynamic character, then nonlinear partial least squares and

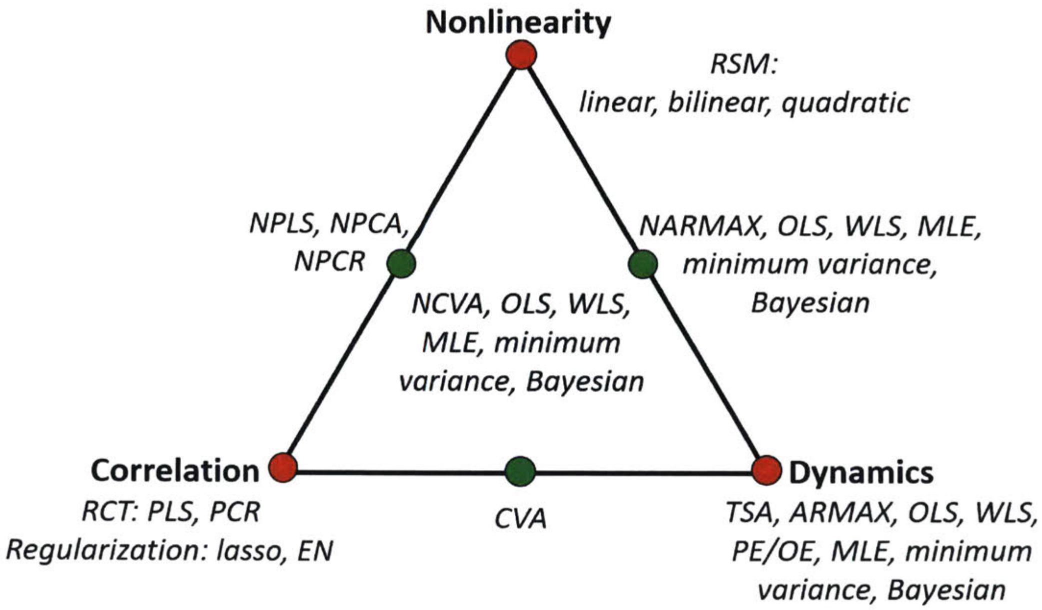


Figure 2-1: The data analysis triangle, which maps modeling techniques to data characteristics. ARMAX = autoregressive moving average model, CVA = canonical variate analysis, MLE = maximum likelihood estimation, OLS = ordinary least squares, NARMAX = nonlinear autoregressive moving average model, NCVA = nonlinear canonical variate analysis, NPLS = nonlinear partial least squares, NPCA = nonlinear principal component analysis, NPCR = nonlinear principal component regression, PLS = partial least squares (aka projection to latent structures), PCR = principal component regression, WLS = weighted least squares.

nonlinear principal component analysis are among the best data analytics techniques for constructing models from the dataset. Techniques most appropriate for datasets that contain significant nonlinearity, dynamics, and correlations are shown in the middle of the data analytics triangle. While such techniques are very powerful, a much higher level of data analytics expertise and larger quantity of data are required to be able to apply such techniques reliably, and the tendency of overfitting data is higher when the complexity of the model is higher. More generally, the simplest model able to describe the characteristics of the dataset should be used.

All models illustrated here are variations on a regression model that finds a vector of weights, $\mathbf{w} \in \mathbb{R}^p$, which can be used to predict the scalar CQA, y, using the vector of CPPs, $\mathbf{x} \in \mathbb{R}^p$. The approach for finding $\mathbf{w}$ is called ordinary least squares (OLS), which minimizes the square error in the prediction. For a static linear model, the vector w is the solution of an optimization with a quadratic objective function,

$$\hat{\mathbf{w}} = \arg\min_{w} \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i)^2 \tag{2.1}$$

A *static* model implies that the model does not contain any dynamics, that is, the output is an algebraic function of the states without any derivatives or integrals over time. For data with low correlation and number of well-designed experiments larger than the number of model parameters (that is, the number of elements of $\mathbf{w}$), Eqn. 2.1 has the unique analytical solution

$$\hat{\mathbf{w}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \tag{2.2}$$

where $\mathbf{X}$ is the $n \times p$ matrix of inputs and $\mathbf{y}$ is the n-dimensional vector of outputs. The precise mathematical definition of "well-designed experiments" within this context is that the matrix inverse in Eqn. 2.2 exists, which is equivalent to the determinant of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ being nonzero.

The case study applies a class of methods known as regularization techniques. Regularization techniques are motivated by the fact that the OLS problem can lead to solutions that are over-fit to the dataset, particularly as the number of parameters p becomes large compared to the number of experiments. To prevent overfitting, regularization adds penalties to the optimization problem of minimizing the squared error. Depending of the precise formulation of the penalties, the resulting models will have different properties. The most desirable penalty would be a so-called ℓ_0 norm, which is not actually a norm but has some of its properties, and is defined by

$$||\mathbf{w}||_0 = \text{the number of non-zero elements in the vector } \mathbf{w} \tag{2.3}$$

The desirable characteristic of this penalty is that is penalizes the complexity of the model without affecting the values of the optimal weight vector $\mathbf{w}$. The ℓ_0 norm is not differentiable, and optimizations that incorporate an ℓ_0-penalty term are computationally expensive. To produce optimizations that are much easier to solve numerically, two commonly used alternative penalties are the ℓ_1 and ℓ_2 norms, which is the sum of the absolute value of the elements of the vector w and the sum of squared values of the elements of $\mathbf{w}$, respectively.

The ℓ_2 case is referred to as ridge regression, also known as Tikhonov regularization [96] and is formulated as

$$\hat{\mathbf{w}}_{ridge} = \arg\min_{w} \sum_{i=1}^{n} (y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i)^2 + \lambda \sum_{j=1}^{p} w_j^2 \tag{2.4}$$

where λ is a a nonnegative regularization parameter and all other variables are defined as above. This problem is strictly convex and the closed-form solution can be written as

$$\hat{\mathbf{w}}_{ridge} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}_c \tag{2.5}$$

where $\mathbf{X}$ and $\mathbf{y}_c$ are the input variable matrix and output variable matrix, respectively, each mean-centered. By adding the ℓ_2 penalty term, the variance of the result is decreased, which leads to a result that is more stable. To choose the value of λ, often cross-validation strategies are used as described above based on minimizing the prediction error, over a grid of λ values. Another possible approach is to use a prior distribution on the value of λ, however this is done less frequently in practice.

Optimization using an ℓ_1 norm as the penalty is referred to as lasso [246], which is formulated as

$$\hat{\mathbf{w}}_{lasso} = \arg\min_{w} \sum_{i=1}^{n} (y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i)^2 + \lambda \sum_{j=1}^{p} |w_j| \tag{2.6}$$

where λ is a nonnegative regularization parameter and all other variables are defined as above. This penalty is similar to ridge regression, but the objective function is not strictly convex and the optimal solution typically has some coefficients in the vector w exactly zero. This approach to generating a sparse model not only prevents over-fitting but also simultaneously performs model selection, although the resulting coefficient vector w is biased due to the penalty. Often in practical applications, lasso is used only to find the model complexity and then OLS is used to find the exact values of the nonzero coefficients [67]. The value of the regularization parameter λ is obtained as in ridge regression, using cross-validation.

Many variations of regularization techniques exist that use different p-norms and combinations of penalties. The elastic net [302] has been found to be effective in biopharmaceutical applications [228] and is formulated

$$\hat{\mathbf{w}}_{EN} = \arg\min_{w} \sum_{i=1}^{n} (y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i)^2 + \lambda P_\alpha(\mathbf{w}) \tag{2.7}$$

where

$$P_\alpha(\mathbf{w}) = \sum_{j=1}^{p} (1-\alpha)\mathbf{w}_j^2 + \alpha|w_j| \tag{2.8}$$

where λ is a nonnegative regularization parameter, α is on the interval (0,1], and all other variables are defined as above. The elastic net combines the ridge and lasso penalties. This is desirable for pharmaceutical applications because there are often more measurements than observations ($p > n$), and in this case the solution to the lasso problem is not unique. It is still desirable to do model selection; by combining both terms the problem is convex and also has solutions that are sparse. Both λ

and α must be chosen to apply the elastic net and this is again done using cross-validation but using a 2-D grid over λ and α values. To better fit the application of small n additional considerations for over-fitting were added [228]. These additional considerations also leveraged Monte Carlo trials. The elastic net is applied, with a fixed α value, many times and the dimensions of the models corresponding to the lowest validation error are recorded. Then the number of dimensions is decreased by considering a threshold on the frequency with which a dimension is selected. This method helps to decrease the dependence of the model on the specific dataset that was used for training. Using this small dataset, exhaustive best subset selection can be performed to choose the final model (see Chapter 3 for full details). One of the most popular methods for quantifying potential over-fitting is to employ Monte Carlo sampling, which is applied here.

The next section describes the systematic application of data analytics to bench-scale data for the manufacturing of the same monoclonal antibody and the next chapter has a similar focus, but uses manufacturing-scale data.

2.3 Application of data analytics to laboratory-scale experiments

This section describes the application of data analytics to laboratory-scale data generated by statistical design of experiments (DOE). Data from seven of the mAb processes were available, as shown in Fig. 2-2. As is common practice, experiments on the processes were carried out in isolation, without the ability to connect the experimental data generated among different processes. None of the data involved time series, so only static models can be constructed. All data were preprocessed by z-scoring, which standardize the measurement data as

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{\sigma_j} \tag{2.9}$$

where x_{ij} is the ith observation, $\bar{x}_j$ is the mean, and σ_j is the standard deviation of

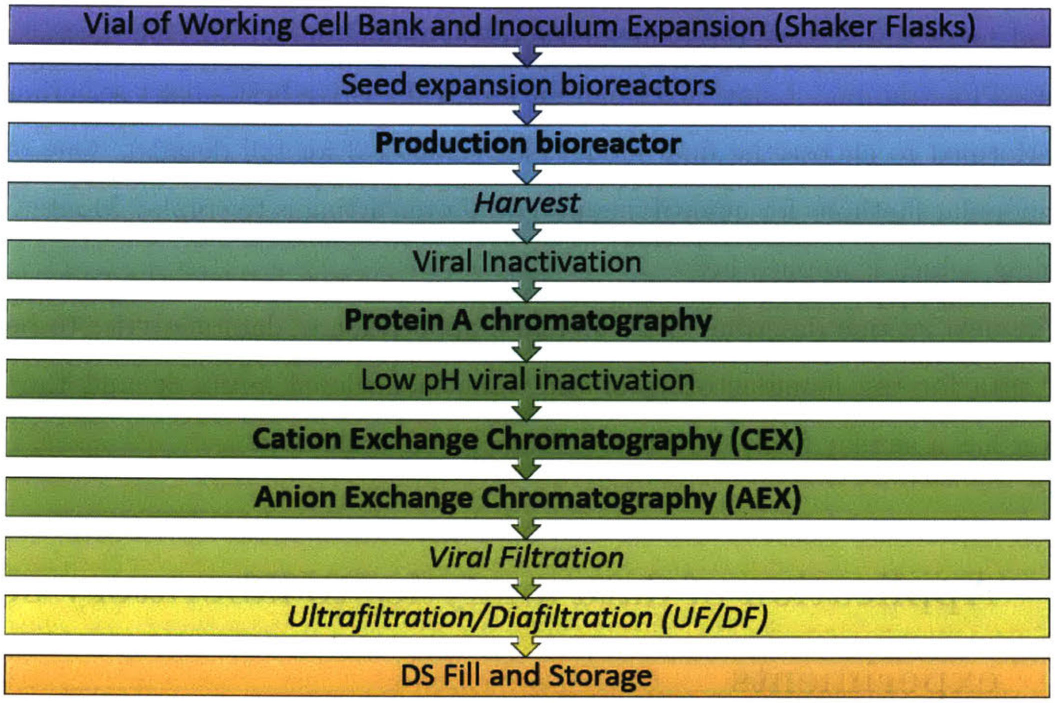

Vial of Working Cell Bank and Inoculum Expansion (Shaker Flasks)
Seed expansion bioreactors
Production bioreactor
Harvest
Viral Inactivation
Protein A chromatography
Low pH viral inactivation
Cation Exchange Chromatography (CEX)
Anion Exchange Chromatography (AEX)
Viral Filtration
Ultrafiltration/Diafiltration (UF/DF)
DS Fill and Storage

Figure 2-2: Block diagram showing the steps in the biomanufacturing process [231]. Processes in boldface are included in both the DOE and process datasets. Italicized processes are included only in the DOE dataset. Normal typeface processes are not included in either dataset.

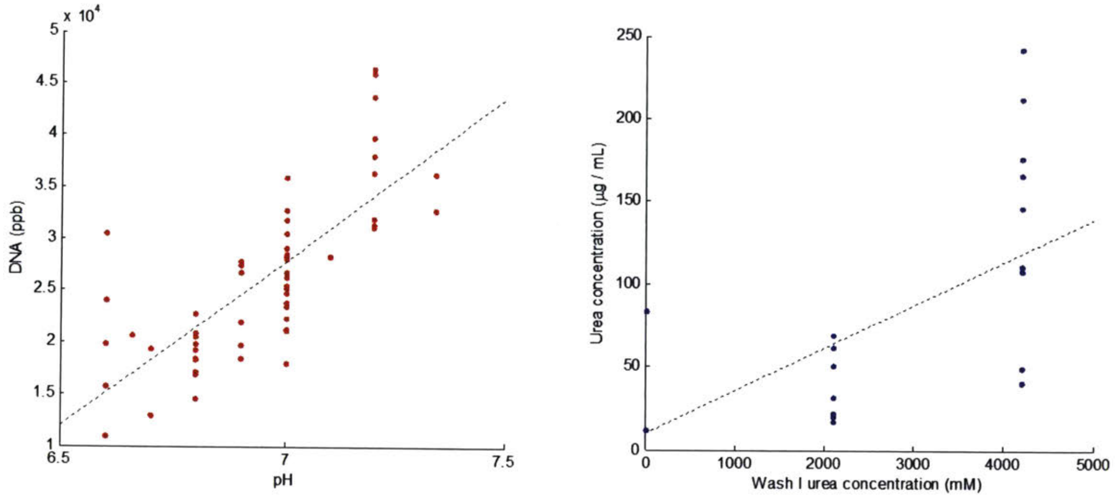


Figure 2-3: Two variables that are highly correlated. The left plot shows the correlation between DNA and pH in the bioreactor ($\rho = 0.76$). The right plot shows the correlation between urea entering and exiting the protein column ($\rho = 0.73$).

measurement j, respectively. Z-scoring is useful for measurements of different types, such as temperature and host cell protein concentration, and prevents numerical artifacts that can arise when using data of very different scales. Z-scoring is typically not useful when all of the input variables to a model are of the same type, such as absorbances at different wavenumbers in an infrared spectra.

When first presented with a dataset, it is desirable to plot the data and perform a simple correlation analysis. The correlation coefficient is

$$p(X, Y) = \frac{\operatorname{cov}(X, Y)}{\sqrt{\operatorname{var}(X)\operatorname{var}(Y)}} \tag{2.10}$$

where x and y are any two scalar variables in the dataset. The correlation coefficient ranges from $[-1, 1]$ and is a measure of linear correlation only. The correlation coefficient is trivial to compute and is an important first consideration when analyzing data, because it provides a starting point for analysis and can serve as a check for surprising behavior. Plotting the data will also serve to highlight surprising behavior as well as possible data errors.

Fig. 2-3 shows two sets of variables that have significant linear correlations, with correlation coefficients larger than 0.7. Fig. 2-3a suggests that the DNA in the bioreactor at the end of a batch run is reduced by more than a factor of 3 by operating the bioreactor at a pH of 6.6 instead of about 7.3. Fig. 2-3b shows that the urea concentration exiting the protein A column varies by about a factor of five for fixed values of the urea entering the protein A column; not surprisingly, the exiting urea concentration tends to be lower for lower values of the entering urea concentration.

Nonlinear correlations between variables in the data are also of interest. Data can be plotted for different nonlinear transformations of variables, whose forms can be suggested from plots of the original data as in Figure 3 or of z-scored data. Then correlation coefficients can be computed for the nonlinear transformed data. Here an alternative approach is presented that is fast to apply when the datasets have many variables. The approach considers bilinear and/or quadratic forms as candidate relationships and is often referred to as response surface methodology (RSM) in the literature. Quadratic functions are the second-order Taylor series approximation for any smooth nonlinearity, and provide a reasonable starting point for nonlinear analysis. To complete this analysis, test statistics can be calculated to answer the question: for each input-output pair x_1 and y_1, is $y_1 = w_2x_1^2 + w_1x_1 + w_0$ a better fit for the data than $y_1 = w_1x_1 + w_0$? The test statistic is defined

$$T_0 = \frac{w_i - w_{i0}}{\sqrt{\sigma^2 C_{ii}}} \tag{2.11}$$

where w_i is the coefficient that is being tested, w_{i0} is the coefficient value under the null hypothesis, σ^2 is the variance, which is estimated using the mean-squared error, and C_{ii} is the ith diagonal element of the covariance of the input data matrix. Consider the case where the null hypothesis is that the coefficient is zero, $H_0 : w_{i0} = 0$. If $|t_0| > t_{\alpha/2,n-p}$, the null hypothesis is rejected, indicating that there is sufficient evidence in the data to indicate that the coefficient w_i is nonzero. The value of $t_{\alpha/2,n-p}$ depends on the desired confidence level [165]. For the model $y_1 = w_2x_1^2 + w_1x_1 + w_0$, the null hypothesis is that the model is linear and w_2 is the coefficient being tested, and $|t_0| > t_{\alpha/2,n-p}$ would indicate sufficient confidence in the use of the quadratic

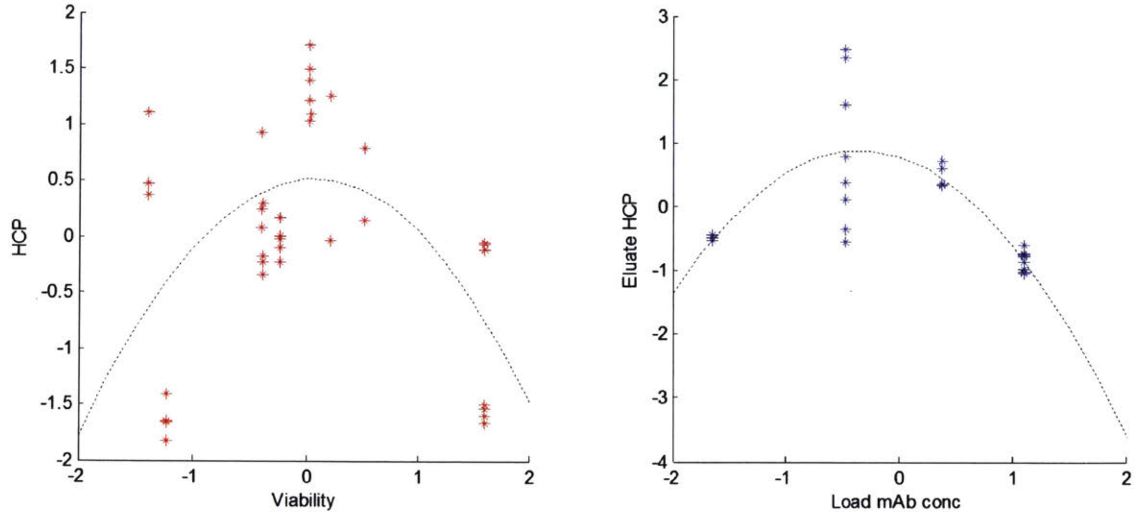


Figure 2-4: Two pairs of variables that have a statistically significant quadratic coefficient at the 95% confidence level. The quadratic regression model is shown as a dotted line.

model. Fig. 2-4 shows two sets of variables for which the hypothesis testing indicates that a quadratic model is justified.

Often there are many coefficients tested in an RSM analysis; therefore, the analyst must be cognizant of that fact that some relationships will be statistically significant by random chance, which is called a false positive. The expected value of the number of false positives can be found by multiplying the number of models that are tested by 1–confidence level, but does not enable the identification of which specific relationships are false positives. However, an estimate of the number of false positive relationships can inform modeling decisions in later analysis. A matrix of input-output pairs can be used to organize the RSM analysis. Each column of the matrix corresponds to an input set by the experimentalist and each row corresponds to a measured output. Each entry in this matrix is either the t_0 value or the α-level at which the higher order term would not be statistically significant, which is a choice made by the analyst. This matrix enables a quick scan to identify variables that are likely to be related by static nonlinear relationships, such as for the two pairs of variables related in Fig. 2-4.

Testing the statistical significance of sets of variables is also sometimes of interest.

In this case, a partial F-test is performed using the statistic

$$F_0 = \frac{SS_R(\mathbf{w}_1|\mathbf{x}_2)/r}{MS_E} \tag{2.12}$$

where

$$SS_R(\mathbf{x}_1|\mathbf{w}_2) = \mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} - \mathbf{w}_2^{\mathrm{T}}\mathbf{X}_2^{\mathrm{T}}\mathbf{y} \tag{2.13}$$

$$MS_E = \frac{\mathbf{y}^{\mathrm{T}}\mathbf{y} - \mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y}}{n-p} \tag{2.14}$$

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} \tag{2.15}$$

and the dimension of $\mathbf{w}_1$ is r and the dimension of w is p. Here the null hypothesis is that $\mathbf{w}_1 = 0$. If $F_0 > F_{\alpha,r,n-p}$, then there is sufficient confidence that at least one of the variables in $\mathbf{w}_1$ is nonzero [165].

Adding additional terms, such as quadratic terms, to a model is a specific example of the larger problem of feature selection. Feature selection refers to the problem of determining the best representation of the data for modeling purposes. Common features that are considered in biopharmaceutical modeling include bilinear and quadratic terms and nonlinear transformations such as the logarithm. Feature selection can sometimes be motivated by plotting variables against each other, as discussed above, or from an understanding of the underlying phenomena or prior knowledge. The feature selection problem is more complicated in scenarios where the quantity of data is limited. At a minimum, to solve the regression, the complete feature matrix must be full rank. As a rule of thumb, the number of experiments should be at least 1.5 times the number of coefficients in the model to be fit. Having more data than the minimum allows the model to be cross-validated as a check for over-fitting, a scenario in which the predictions of the model are much less accurate when applied to new datasets than for fit of the model outputs to the data used to fit the model. In other words, over-fitting has occurred when the model error is low for the dataset used to build the model but is high for new data points. An over-fit model has not captured the underlying phenomena well but instead is a good representation of a specific dataset used to fit the model. Testing for over-fitting is typically carried out by cross-validation, which is the evaluation of the model's performance on data that were not used for model building. Robust cross-validation involves numerous iterations in which each iteration consists of three steps: calibration, validation, and testing. Dividing the data into three parts, first models of increasing complexity (feature sets) are fit. Then each of the models is tested on the validation data, and used to compute a cross-validation error. Typically, the cross-validation error will first reduce as the model complexity increases and more accurately describes the underlying phenomena until then increase as the model begins to fit noise and biases in the specific training dataset. The optimal model complexity is that which minimizes the cross-validation error. Once this model form has been selected, the final model calibration is carried out using both the fitting and validation datasets. The testing dataset is used to characterize the error of the model, by using data that has been unseen by the model. The manner in which the dataset is divided into these three parts will depend on the quantity of data that is available. Ideally, the data are divided into three equal parts. When insufficient data are available for this division to be feasible, often the testing set is excluded and additional measures to prevent over-fitting are incorporated into the model calibration step.

The above ideas were applied to the individual processes with feature sets that included linear, bilinear, and quadratic terms. Models were built using response surface methodology (RSM) as well as elastic net with Monte Carlo sampling (ENwMC) [228]. For this case study, RSM was applied as is common in practice, to construct a sparse model without using Monte Carlo sampling.

For illustrative purposes, consider the application of the data analytics techniques from the cation exchange column, which included six independent inputs, to model the host cell protein (HCP) concentration exiting the column. The objective is to construct a quadratic function between the six inputs and one output that has good generalizability, that is, produces accurate predictions when applied to data not used to fit the model. The total number of coefficients in such a quadratic model is 28, which is larger than the number of experiments, which was 24. As such, it is impossible to directly apply ordinary least-squares to determine the values of the coefficients, because there is an infinite number of choices of the 28 coefficients that is able to exactly fit the data. This observation motivates the construction of sparse models, in which most of the coefficients in the quadratic model are set to zero.

For such datasets, special care must be taken to avoid overfitting the models. Random sampling is an effective approach for the evaluation of such models. To test the generalizability of each model, 12 experiments are randomly chosen to calibrate a model of the resulting structure. Using the remaining 12 experiments, the prediction error is calculated. These two steps were repeated 1000 times to produce statistically stable results and the test error was averaged to determine the mean squared error for the testing data, which is the metric for comparing the relative value of different models. This procedure approximates the full distribution of the testing error by subsampling the available data. The preferred method would be for the dataset to be large enough that subsampling would not be needed, but the dataset was not large enough so the subsampling procedure is a reasonable strategy.

The RSM model had nearly a factor of two lower prediction errors when applied to the data used to fit the models (see Table 2.1), whereas the ENwMC model gave nearly a factor of two more accurate predictions when applied to conditions not used to fit the model. The ENwMC model also had lower complexity, which is a common occurrence when robust cross-validation procedures are applied. The differences between the MSEs of calibration and testing demonstrate the importance of testing for over-fitting to ensure model generalizability.

In the next chapter, data analytics are applied to manufacturing-scale experiments.

2.4 Conclusions

This chapter illustrates some key points to consider when applying process data analytics. First, data analytics techniques should be selected based on the specific

Table 2.1: Models for host cell protein (HCP) exiting the cation exchange column using two different techniques. Both the average error of calibration and testing are reported.

Modeling technique	Model result	MSE of calibration	MSE of testing
RSM	HCP = 175 - 1.83(Load HCP) - 16.5(Elution pH) - 0.22(Elution NaCl) + 0.31(Load HCP)(Elution pH) + 2.4×10^{-3}(Load HCP)(Elution NaCl)	1.4×10^3	9.2×10^3
ENwMC	HCP = -247 - 1.25(Load HCP) + 0.27(Load HCP) (Elution pH) + 0.39(Elution pH)(Elution NaCl)	2.3×10^3	5.5×10^3

scenario and data availability. The important data characteristics to consider are correlation, nonlinearity, and dynamics. Once these characteristics have been identified using the discussed methods, the data analytics triangle can be used a guide for making this selection. As part of identifying characteristics, plotting the data is recommended to check for expected behaviors and possible data errors.

Chapter 3

Sparse modeling for biopharmaceutical manufacturing

This work originally appeared as: Kristen Severson, Jeremy G. VanAntwerp, Venkatesh Natarajan, Chris Antoniou, Jörg Thömmes, and Richard D. Braatz. Elastic net with Monte Carlo sampling for data-based modeling in biopharmaceutical manufacturing facilities. Computers & Chemical Engineering, 80:30-36, 2015.

3.1 Introduction

The U.S. biotechnology sector has had double-digit growth rates in recent years [106]. In 2012, sales of biologics were approximately $63.6 billion, with monoclonal antibodies (mAbs) representing the largest fraction of this market with approximately 39% of sales [1]. Modeling of the manufacturing process is one possible way to both support the growing biologics market as well as decrease costs via improved control and understanding of process operations. Modeling can play an important role in understanding, controlling, and optimizing the process steps used in these processes [254]. The U.S. Food and Drug Administration and International Conference on Harmonization recommend modeling in the development of biologics to estimate variability, provide process understanding, and establish a control strategy [177, 104].

Process modeling techniques can be grouped into two broad categories: first-principles and data-based. This article focuses on data-based modeling, which is more often applied in (bio)pharmaceutical manufacturing facilities. Data-based models have been applied to cell culture characterization [161, 123, 202], quality control [207, 42], process monitoring [202, 203, 204, 24], and downstream operations [202]. A drawback of current data-based methods applied in the biopharmaceutical industry is that the models that are produced are not easily interpretable because they rely on subspaces that do not have direct physical meaning.

With this motivation, a successful biopharmaceutical model would achieve three goals: (1) model accuracy, (2) model simplicity, and (3) model interpretability. These aims have the caveat of using only a small amount of heterogeneous data, as data for biopharmaceutical manufacturing are typically both heterogeneous and relatively limited compared to most mature industries such as in chemicals, refining, petrochemicals, and pulp and paper.

One way to achieve these goals is through the identification of the input variables in the process that exhibit the largest effects on the output variables. It is common in the biopharmaceutical industry for a dataset to have more measurements, p, than observations, N. Most measurements are only taken once in a single batch moving through the production process and few replicates are performed due to time and cost constraints. The construction of predictive models from such data sets can be made even more challenging because the collected data are typically highly correlated between batches, that is, the data sets are highly ill-conditioned. Regularization methods have been identified as possible approaches for such problems because of their ability to simultaneously handle input selection and model estimation [181]. This article first provides some background on regularization methods, specifically the lasso and elastic net. Modifications are then introduced to better handle small heterogeneous datasets. Finally, the methodology is evaluated for a manufacturing-scale process in the biopharmaceutical industry and the results are compared to other data-based modeling techniques used in the industry.

3.2 Background on regularization

The simplest form of regression finds a vector of weights, $\beta \in \mathbb{R}^p$, that can be used to predict the scalar output y using the vector in inputs, $\mathbf{x} \in \mathbb{R}^p$. The basic approach to finding β is called ordinary least squares (OLS). The OLS problem is formulated to minimize the error:

$$\text{Err}(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{x}_i^{\text{T}}\beta)^2 \tag{3.1}$$

which has the solution:

$$\hat{\beta}_{OLS} = (\mathbf{X}^{\text{T}}\mathbf{X})^{-1}\mathbf{X}^{\text{T}}\mathbf{y} \tag{3.2}$$

where $\mathbf{X} \in \mathbb{R}^{N\times p}$ and $\mathbf{X}^{\text{T}}\mathbf{X}$ is invertible. Applying this method can lead to over-fitting of the model, especially as the number of input variables (p) grows large. Regularization techniques are one method to prevent over-fitting.

Lasso [246], also known as ℓ_1 regularization, is an optimization formulation for parameter estimation that solves

$$\hat{\beta}_{lasso} = \underset{\beta,\beta_0}{\arg\min}\frac{1}{2N}\sum_{i=1}^{N}(y_i - \beta_0 - \mathbf{x}_i^{\text{T}}\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{3.3}$$

where N is the number of experiments, y_i is the ith scalar response, $\mathbf{x}_i \in \mathbb{R}^p$ is the data vector at observation i, λ is a nonnegative regularization parameter, β_0 is a scalar parameter, and $\beta \in \mathbb{R}^p$ is a vector of model parameters. By adding the penalty term to the objective, the size of the coefficient vector is effectively constrained, which helps to prevent wild fluctuations of the coefficient vector that can be due to fitting noise in the data. The penalty is equivalent to the ℓ_1-norm on the coefficient vector, hence the name.

The lasso technique is useful to choose the subset of predictors ($\mathbf{x}_i$) that exhibit the strongest effect on y because solutions to the lasso are sparse vectors, that is, the models only include a subset of the possible inputs ("dense" refers to models that include all possible inputs; these terms do not refer to the quality of data within

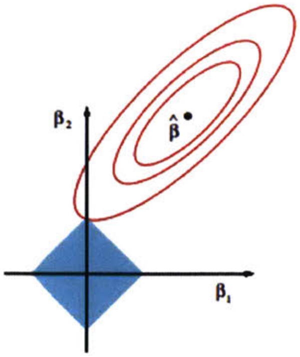


Figure 3-1: Representation of parameter selection using the lasso considering the constrained optimization problem. The blue regions represent the constraints imposed by the penalty terms and the red ellipses are the contours of the least squares error function. The solution often lies on a vertex of the constraint, causing some parameters to be exactly zero [93].

each type of measurement). Because of the ℓ_1 constraint, solutions to the lasso can be thought of as lying on a vertex point of the feasible region, leading to certain coefficients to be exactly zero [93, 201] (see Figure 3-1 for a simple representation).

The elastic net (EN) [302] is an optimization formulation for parameter estimation that is formulated as:

$$\hat{\beta}_{EN} = \arg\min_{\beta,\beta_0} \frac{1}{2N}\sum_{i=1}^{N}(y_i - \beta_0 - \mathbf{x}_i^{\mathrm{T}}\beta)^2 + \lambda P_\alpha(\beta) \tag{3.4}$$

where

$$P_\alpha(\beta) = \sum_{j=1}^{p} \frac{1-\alpha}{2}\beta_j^2 + \alpha|\beta_j| \tag{3.5}$$

N is the number of experiments, y_i is the ith scalar response,$\mathbf{x}_i \in \mathbb{R}^p$ is the data vector at observation i, λ is a nonnegative regularization parameter, β_0 is a scalar parameter, $\beta \in \mathbb{R}^p$ is a vector of model parameters, and α is on the interval $(0, 1]$. Although the elastic net is very similar to the lasso, there are some key differences. The EN is particularly useful when the number of predictors (p) is greater than the number of observations (N). If the lasso is applied to a data set where $p > N$, the solution is not unique. By adding a second term, the problem becomes convex even

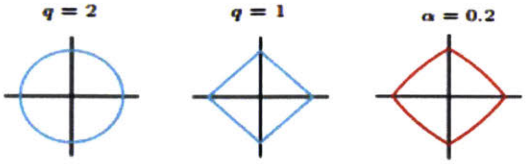


Figure 3-2: Penalty constraint contours for constant values of $\sum_j |\beta_j|^q$ (left two plots, ridge regression and lasso penalties) and $\sum_j(\alpha\beta_j^2 + (1-\alpha)|\beta_j|)$ (right plot, elastic net penalty) [93].

when $p > N$ [302].

EN is also better at handling data where the inputs are correlated. Lasso is only able to select up to N predictors and will not reveal grouping relationships. Instead, the lasso will choose one of the correlated variables, and in highly correlated cases, can switch between variables in the set. However, the EN formulation uses a strictly convex penalty function and will guarantee that equal weighting is given to inputs that are identical [302]. Figure 3-2 compares the penalty constraint contours for ridge regression which uses a quadratic penalty, lasso, and EN. EN can produce models that are sparse and can handle correlated data. These points are illustrated here using a simple four-dimensional case study where x_1 and x_2 are specified then x_3 and x_4 are calculated to equal x_1 and x_2, respectively, with a small amount of random noise. The parameter traces in Figure 3-3 show how EN groups the variables where the lasso uses one variable from each grouped set. The group is very robust to the selection of values of the penalty term λ. By including the grouped variables in the elastic net, the noise in the grouped measurements can be averaged in the calculation of the predictions produced by the model. Lasso selects a sparser model for a given value of λ, but at the cost of not being able to average noise from what are essentially duplicate measurements of the same variable.

The next section details the data-based modeling algorithms employed in this article, which are largely based on the elastic net but also draw on other algorithms.

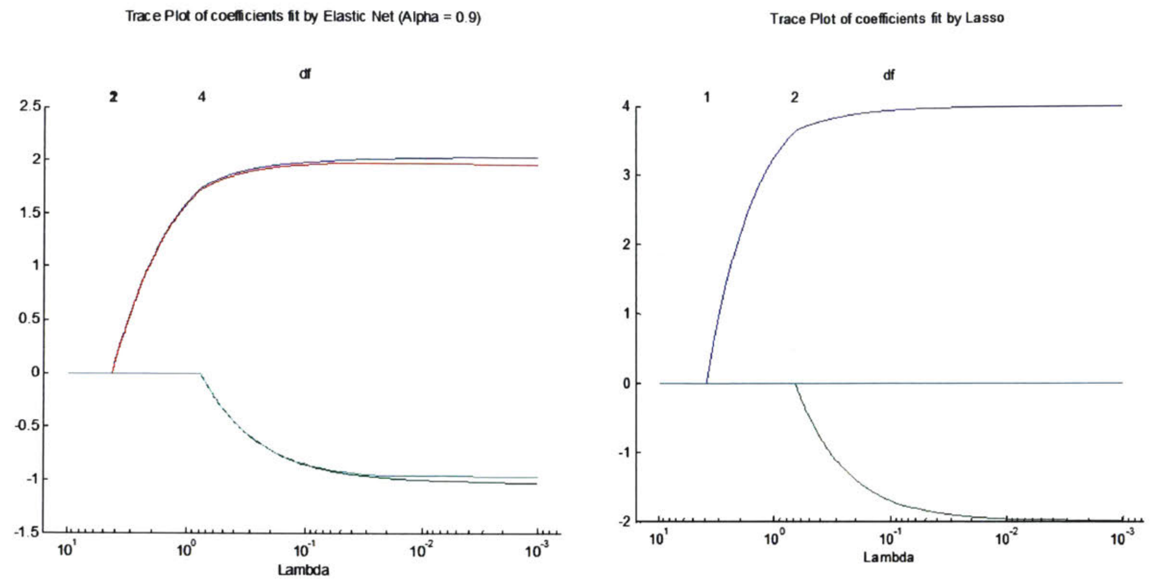


Figure 3-3: Elastic net (left) compared to lasso (right) for highly correlated input data. The left axis shows the value of the model parameter (β), the upper x-axis shows the degrees of freedom (df) in the model, and the lower x-axis is the value of the penalty term λ.

3.3 Elastic net with Monte Carlo sampling (ENwMC)

To effectively use the regularization technique on a small dataset, a multistep procedure is proposed. This procedure draws on ideas from previously proposed lasso and elastic net algorithms, notably, bolasso [10], two-stage estimation [67], and relaxed lasso [160], but differs in its exact implementation. The scalar model form is

$$y(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\beta \tag{3.6}$$

where $\mathbf{x} \in \mathbb{R}^p$, $\beta \in \mathbb{R}^p$, and the value of p is not known *a priori*. For small data sets typically common in the biopharmaceutical industry, there is often insufficient physical or data-based evidence to support a more complicated model. If there is sufficient evidence to support a more complicated nonlinear model, then nonlinear algebraic transformations can be applied to the values of x or y to generate nonlinear models using the same algorithms described in this article.

The first step in ENwMC is an application of the elastic net (4), using leave-one-out cross validation to choose the value of Îś. In leave-one-out cross validation, all but one of the experimental observations are used to fit the model, then the remaining experiment is used to calculate the error (1). This step is repeated for each possible set and then averaged. The procedure is performed for many possible combinations of the regularization parameters α and λ, where $\alpha \in (0, 1]$ and λ captures the convex behavior of the error. Because α is the weighting between the ℓ_2- and ℓ_1-norm penalties and the goal is a sparse model, a value of α close to 1 is preferable. Therefore α is chosen based on a tradeoff between model dimensionality and prediction error. In some cases, this choice is trivial, as a higher value leads to a more accurate model.

Once the value of α is fixed, a test for over-fitting is performed using k-fold cross validation. Using Monte Carlo samples [163], the data are portioned into a validation set containing $(1/k)$ proportion of the data and a calibration set containing the rest. The elastic net, with a fixed α, is then performed and the input variables corresponding to the minimum error are recorded. This step is repeated many times to converge to the distribution of models over the possible calibration and validation sets. The frequency with which each variable is selected is then calculated. In further analysis, only the variables that were selected above a threshold frequency are considered.

The subset of selected variables is considered for inclusion in a model using best subset selection. The error of all possible ordinary least squares models of size $m \in 1, 2, ..., p$ where p is now the dimensions that were chosen based on the threshold, is calculated. A model from this set is then selected based the tradeoff between increasing dimensionality and decreasing error. Increasing the number of dimensions included in the model will decrease the prediction error but trivial gains in prediction for additional dimensions likely represent over-fitting, which should be avoided. This tradeoff is easily visualized by plotting the prediction error against the model dimensions to create a Pareto curve (see Figure 3-4). Plots of this type will often exhibit an "elbow." The elbow corresponds to the model dimensionality that optimally compromises between model size and prediction error. The result of this step is the final model.

When dealing with limited data, it is important to generate model statistics that accurately quantify the prediction errors and ensure that the models generated by

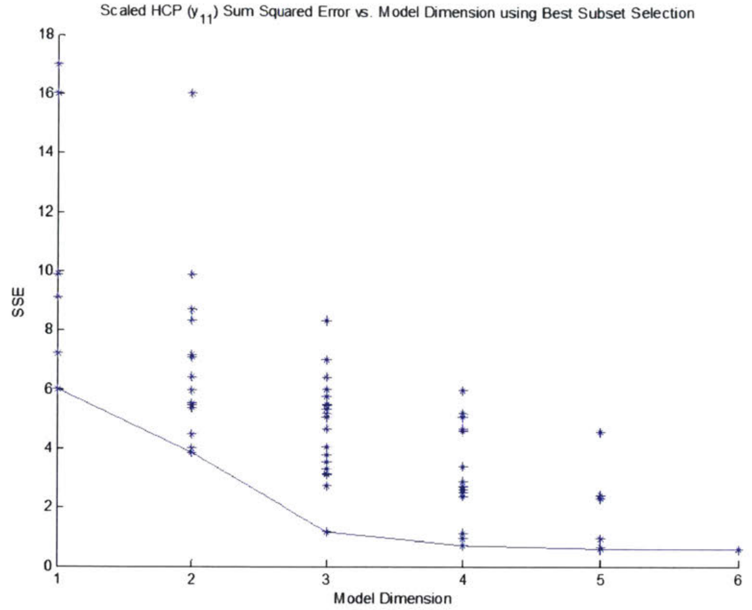


Figure 3-4: Example of a Pareto curve illustrating the tradeoff between model sparsity (horizontal axis) and model accuracy (vertical axis) for host cell protein (HCP) exiting the anion exchange column. In this instance, three input variables are used in the final model.

different data-based methods are compared on a sound statistical basis. Especially relevant are the prediction interval of the model and the covariance of the coefficients. Because of the sensitivity of the model coefficients to the data, Monte Carlo sampling is used, in a similar manner as before. The variance of the prediction is calculated by

$$\hat{\sigma^2} = \sqrt{\frac{n_c}{N}\frac{1}{N-n_\beta}\frac{\sum_{j=1}^{n_{MC}}\sum_{i=1}^{N_j}(y_i-\tilde{y}_i)^2}{n_{MC}}} \tag{3.7}$$

where n_c corresponds to the number of observations used in the calibration set. The pre-factor accounts for the fact that only a fraction of the data are used to calibrate the models. The 95% confidence interval of each prediction is equal to twice the square root of the variance of the prediction.

The covariance of the model coefficients is calculated by

$$\text{cov}(\beta) = \left(\frac{1}{N-n_\beta}\sum_{i=1}^{N}(y_i-\tilde{y}_i)^2\right)(\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1} \tag{3.8}$$

where

$$F_i = \left.\frac{\partial y_i}{\partial \beta}\right|_{\beta *} \tag{3.9}$$

N is the number of experiments, n_β is the number of coefficients in the model, F_i is the ith row of $\mathbf{F}$ and $\mathbf{F}$ is the matrix of data used to build the model.

Within this algorithm, there are many fitting parameters that will vary based on the specific application. Specifically these parameters include the number of Monte Carlo trials, the fraction of the data used for validation in the second step, and the threshold frequency. The number of Monte Carlo trials should be large enough to capture the distribution but not so large that computational resources are wasted. In the case study below, 1000 Monte Carlo samples were used. The value of k in k-fold cross validation will depend on the number of available experiments. A value of $k = 2$ provides a strong degree of validation but cannot be applied to very small data sets. In the case study below, $k = 3$ was used. The number of models that are considered in the best subset selection step scales as $2m$, and therefore should be chosen such that this step is computationally reasonable. In the study below, the threshold was 50%, i.e., any input variable that had a non-zero coefficient in 50% of the Monte Carlo samples was included in the best subset selection.

3.4 Case study

The methodology is evaluated using a dataset from Biogen Idec that involves 18 production batches, each containing 40 measurements, of an antibody manufacturing process. Of the 40 measurements, 14 were outputs and models were constructed for each output. The measurements spanned four process steps, shown in Figure 3-5. All variables were z-scored, that is, mean-centered around zero and scaled by their standard deviation. All measured variables are numerical so the data are homogeneous in type but are heterogeneous in time scale. A detailed discussion of different types of data heterogeneity that arise in bioprocesses is available [39].

For all of the variables downstream of the bioreactor, either "modular" models can be constructed, which include only the variables exiting the previous unit operation as

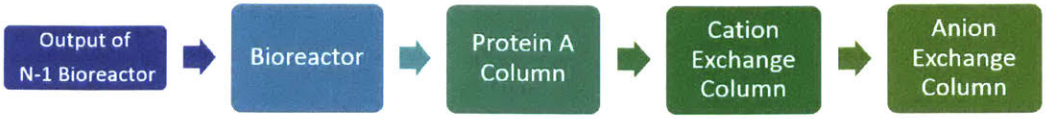


Figure 3-5: Simplified flowsheet of the antibody production process. The bioreactor size was 2000L and was operated in the fed-batch mode. The column loadings were typical of an antibody purification process [231].

Table 3.1: Summary of the ENwMC modeling results for all of the output variables in the process. All statistics are reported in scaled units [190]. HCP is host cell protein and HMW is high molecular weight impurities.

Unit Operation	Output Dimension	Num. of Input Vars Used in Final Model	Num. of Possible Input Vars	SSE	95% Pred. Interval
Bioreactor	G0 Product Quality	3	20	3.41	0.591
	Final Titer	3	20	5.40	0.836
	DNA	4	20	5.20	0.944
	Host Cell Protein	6	20	1.67	0.775
Protein A Column	DNA	4	26	2.71	0.616
	HCP	4	26	1.92	0.565
	Total Impurity	5	26	2.40	0.810
	HMW	4	26	1.11	0.424
Cation Exchange Column	HCP	4	32	1.96	0.576
	Total Impurity	2	32	7.18	0.951
	HMW	3	32	0.32	0.201
Anion Exchange Column	HCP	3	37	1.20	0.452
	Total Impurity	4	37	2.48	0.679
	HMW	2	37	0.23	0.166

well as the inputs to that unit operation, or "full process" models can be constructed, which include all of the upstream variables. In all cases, the full process models were of higher predictive accuracy than the modular process models, so only the full process model results are reported here. Table 3.1 reports the 14 outputs, the error and prediction interval for each of the 14 models. For some representative models, Figures 3-6-3-7 provide plots of measurements, predictions, prediction intervals, and residual plots.

The ENwMC modeling technique consistently chose a small number of predictors (third column in Table 3.1), which meets the goals of model simplicity and in-

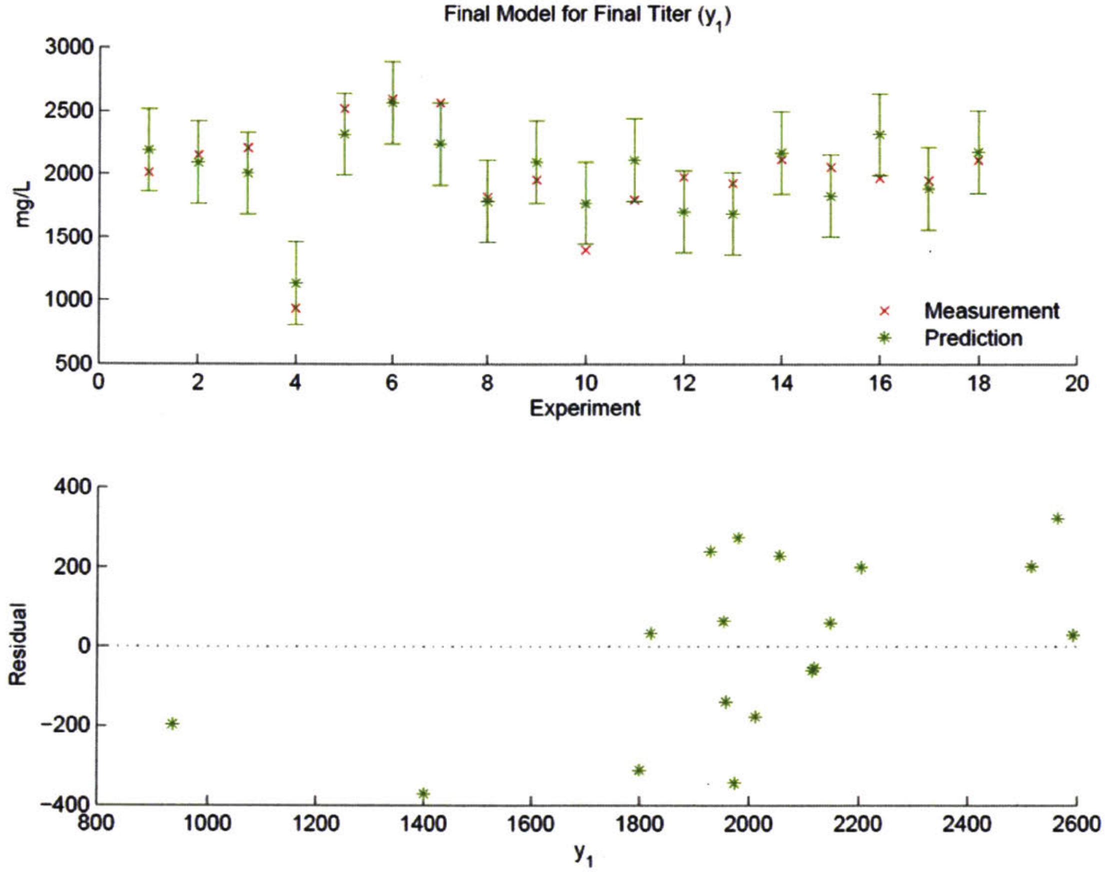


Figure 3-6: Final model, with prediction intervals (95% confidence level) and residual plot, for final titer, exiting the bioreactor.

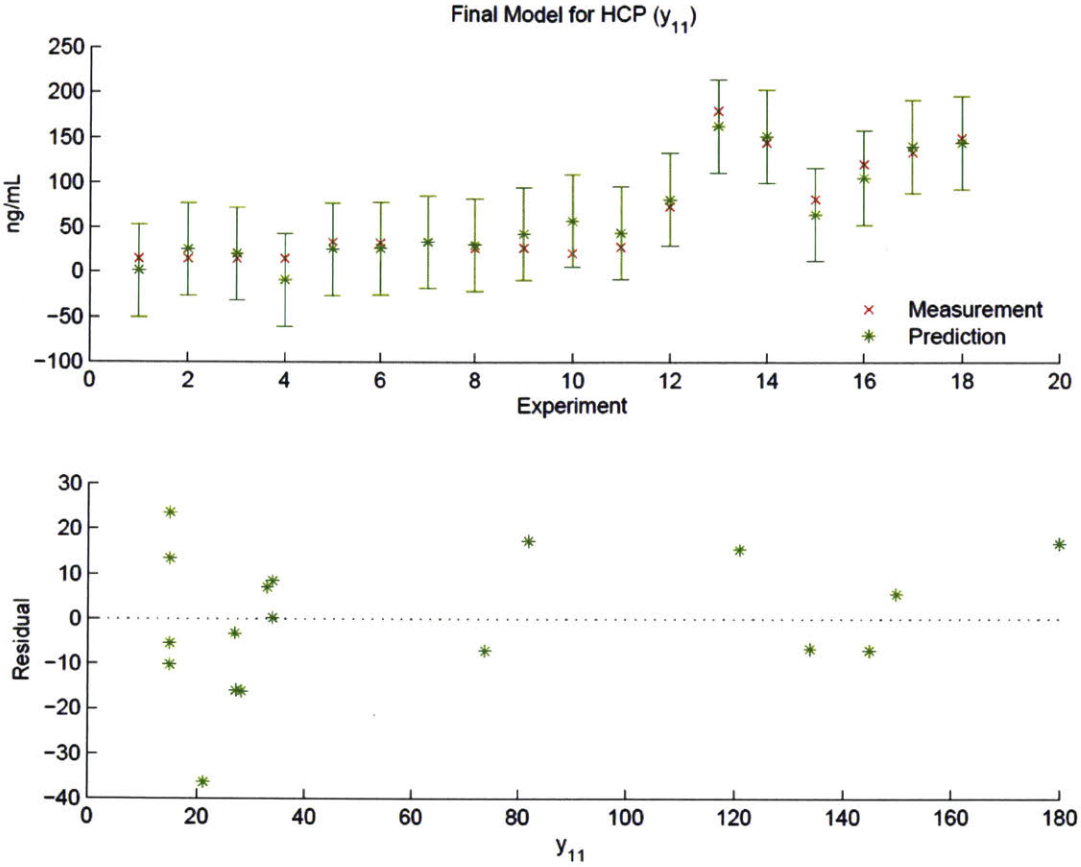


Figure 3-7: Final model, with prediction intervals (95% confidence level) and residual plot, for HCP, exiting the anion exchange column.

terpretability. Analysis of the scaled model coefficients showed clear relationships between the set of input variables and the corresponding output. The accuracy of the model was compared to two chemometrics techniques widely used in the (bio)pharmaceutical industry: principal component regression (PCR) and partial least squares (PLS, also known as projection to latent structures). Unlike regularization techniques, PCR and PLS reduce the dimensionality of the regression problem by truncating variance within the data and then performing regression. These chemometrics techniques lead to dense, rather than sparse, models and are described in [250, 29] . The number of principal components and latent variables for each model were selected using leave-one-out cross validation.

Table 3.2 shows the results of the comparison. In a majority of the cases, the proposed methodology outperforms the chemometrics techniques. The variable with the largest difference between the three data-based modeling techniques is the HMW exiting the cation exchange column, where the ENwMC model has nearly a factor of three lower percent error. In terms of variance of the prediction, the elastic net with Monte Carlo sampling outperformed PCR and PLS for all but one output variable, in some cases by more than a factor of two. For example, the ENwMC model has about a factor of six lower prediction variance than the two chemometrics methods for the HMW exiting the cation and anion exchange columns. For the one variable in which PCR and PLS gave lower predictive variation than ENwMC, which was DNA exiting the bioreactor, the differences between all three methods was very small (0.201 to 0.223). Further, the ENwMC model produces sparse models with few predictors, which is much more useful when trying to make the fewer number of changes in operations to control a process variable than the chemometrics models, which use all of the measurements as predictors.

Although the ENwMC has more hyper-parameters than PLS or PCR, experience so far has indicated that the algorithm can be largely automated. If simple heuristics are used to choose the number of trials, the CV partitioning, and the threshold, the user needs only to choose α and the final model dimensionality. If desired, α can be chosen a priori, which would leave only the model dimensionality to be chosen, which is the same decision that would need to be made for a PCR or PLS model. The computational cost is larger to implement ENwMC, but the cost is not a practical consideration when using modern personal computers for the small datasets encountered in many industrial systems such as this biomanufacturing example.

3.5 Conclusions

The elastic net with Monte Carlo sampling algorithm combines the benefits of the elastic net algorithm for simultaneous model selection and parameter estimation with the power of Monte Carlo sampling to counteract likely over-fitting of data to create an accurate, interpretable, and simple process model. This data-based modeling algorithm has potential for biopharmaceutical applications or any dataset that is small and heterogeneous and for which first-principles models are unavailable. ENwMC is demonstrated to produce more accurate predictions than chemometrics methods for a data set collected from a manufacturing-scale biopharmaceutical facility, while identifying a small number of process variables that can be used in closed-loop control. Although the models here are data-based and statistical in nature, they can still provide insight into the process and are particularly useful when only limited amounts of data are available.

Table 3.2: Comparisons of percent error and scaled variance for PCR, PLS, and ENwMC modeling techniques. The bold number marks the model with the best performance for each variable.

Unit Operation	Output Dimension	Percent error using...			Variance of the prediction using...		
		PCR	PLS	ENwMC	PCR	PLS	ENwMC
Bioreactor	G0 Product Quality	1.7% (4)	1.8% (1)	**1.5% (3)**	0.146 (4)	0.148 (1)	**0.087 (3)**
	Final Titer	9.5% (4)	**6.7% (2)**	10% (3)	0.281 (4)	0.287 (2)	**0.178 (3)**
	DNA	**60% (4)**	62% (1)	62% (4)	0.209 (4)	**0.201 (1)**	0.223 (4)
	Host Cell Protein	12% (6)	10% (2)	**9.0% (6)**	0.258 (6)	0.210 (2)	**0.150 (6)**
Protein A Column	DNA	137% (4)	**73% (1)**	102% (4)	0.151 (4)	0.143 (1)	**0.095 (4)**
	HCP	19% (6)	12% (3)	**12% (4)**	0.268 (6)	0.202 (3)	**0.080 (4)**
	Total Impurity	21% (4)	18% (1)	**11% (5)**	0.286 (4)	0.256 (1)	**0.164 (5)**
	HMW	103% (6)	105% (1)	**60% (4)**	0.117 (6)	0.092 (1)	**0.045 (4)**
Cation Exchange Column	HCP	**16% (9)**	26% (2)	20% (4)	0.226 (9)	0.132 (2)	**0.083 (4)**
	Total Impurity	25% (5)	23% (2)	**21% (2)**	0.323 (5)	0.348 (2)	**0.226 (2)**
	HMW	54% (3)	51% (1)	**18% (3)**	0.058 (3)	0.063 (1)	**0.010 (3)**
Anion Exchange Column	HCP	43% (7)	43% (2)	**42% (3)**	0.189 (7)	0.140 (2)	**0.048 (3)**
	Total Impurity	21% (4)	**11% (3)**	13% (4)	0.228 (4)	0.227 (3)	**0.115 (4)**
	HMW	30% (9)	20% (4)	**18% (2)**	0.067 (9)	0.050 (4)	**0.007 (2)**

Chapter 4

Probabilistic principal component analysis and missing data

This work originally appeared as: Kristen A. Severson, Mark C. Molaro and Richard D. Braatz. Methods for applying principal component analysis to process datasets with missing values. Special Issue on Process Data Analytics, Processes, 5:38, 2017.

4.1 Introduction

Principal component analysis (PCA) is a widely used tool in industry for process monitoring. PCA and its variants have been proposed for process control [153], identification of faulty sensors [64], data preprocessing [144], data visualization [123], model building [292], and fault detection and identification [128] in continuous as well as batch processing [173, 174]. PCA has been applied in a variety of industries including chemicals, polymers, semiconductors, and pharmaceuticals. Classic PCA methods require complete observations; however, often online process measurements or laboratory data have missing observations. Causes of missing data in this context include sensor failure, changes in sensor instrumentation over time, different sampling rates, merging of data from different systems, and samples that are flagged as poor quality and subsequently dropped from storage [107]. The nonlinear iterative partial least squares (NIPALS) algorithm was an early approach for handling missing process data when applying PCA [51, 285]. The problem started to gain more attention in the late 1990s [171, 88] and, because of the ubiquity of missing data, many PCA algorithms that can handle missing data have been proposed since. This article reviews these approaches and provides guidance to practitioners on which methods to apply.

A framework for analysis in the presence of missing data has been available since the mid 1970s [209], which introduces categories of missingness and explains when missingness can be ignored. Three categorizations of missingness are (1) missing completely at random (MCAR), (2) missing at random (MAR), and (3) not missing at random (NMAR) [141]. These categories can be described using the missing-data indicator matrix, $\mathbf{M}$, which is of the same size as the data matrix $\mathbf{X}$ where $M_{ij} = 1$ if X_{ij} is missing and 0 otherwise. The MCAR assumption applies when the independence statement

$$f(\mathbf{M}|\mathbf{X}, \phi) = f(\mathbf{M}|\phi), \quad \forall\, \mathbf{X},\ \phi, \tag{4.1}$$

is true, where f is a probability density, variables to the right of $|$ indicate the conditioning set, and ϕ are unknown parameters. MCAR implies that the missingness is not a function of the data, regardless of whether the data points are observed or missing. The MAR assumption applies when the independence statement

$$f(\mathbf{M}|\mathbf{X}, \phi) = f(\mathbf{M}|\mathbf{X}_{obs}, \phi), \quad \forall\, \mathbf{X}_{mis},\ \phi, \tag{4.2}$$

is true. MAR implies that the missingness depends on the observed data. NMAR is assumed when neither of these criteria apply [141].

Recently, access to large amounts of process data have been enabled by improved sensor technology, the Industrial Internet of Things, and decreased data storage costs. Due to an increasing number and diversity of measurements [194], data with missing elements will become increasingly common. When working with a dataset, the first step is to identify which data are missing and why. If the missingness mechanism is MCAR or MAR, a model for the missingness mechanism is not needed and is referred to as *ignorable* when performing inference. To perform inference, the quantity of interest is the *likelihood*, which is the probability of the observed data, given the distributional parameters. If the MAR assumption holds, the likelihood is proportional to the probability of the observed data given the true parameters and therefore it is not necessary to model the missingness [141]. However, when data are NMAR and the missingness mechanism is not taken into account, algorithms can lead to systemic bias and poor prediction [141]. Conclusive tests for determining the appropriate missingness categorization do not exist, and so the categorization is selected based on process understanding. The conclusions of missingness categorization depend on the specific scenario, but some typical examples for the process industry are presented here to provide guidance to practitioners. MCAR is applicable to data that are missing due to random sensor failure or mishandling of the data. MAR applies to scenarios where data are acquired sequentially, for example, a quality test that is only performed based on the results of previous testing. NMAR applies to measurements that are not recorded due to censoring, where the value is outside of limits of detection [107].

4.2 Methods

4.2.1 Introduction to PCA

Principal component analysis is a technique for dimensionality reduction. Pearson [188] and Hotelling [99] are typically attributed with the first descriptions of the technique [118]. Hotelling described PCA as the set of linear projections that maximizes the variance in a lower dimensional space. For a data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$ where d is the number of measurements and n is the number of samples, the linear projection described by Hotelling can be found via the singular value decomposition (SVD),

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^{\top}, \tag{4.3}$$

where $\mathbf{U} \in \mathbb{R}^{d \times d}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{d \times n}$ is a pseudo-diagonal matrix. The linear projection matrix $\mathbf{P} \in \mathbb{R}^{d \times a}$, also called the matrix of loading vectors, is defined by the columns of $\mathbf{U}$ that correspond to the largest a singular values. The principal components, also called the scores, are defined as

$$\mathbf{T} = \mathbf{P}^{\mathsf{T}}\mathbf{X} \tag{4.4}$$

or as the first a rows of $\Sigma\mathbf{V}^{\mathsf{T}}$. Equivalently, $\mathbf{P}$ can be found by solving the eigenvalue decomposition of the sample covariance matrix,

$$\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^{\mathsf{T}} = \mathbf{U}\Lambda\mathbf{U}^{\mathsf{T}}, \tag{4.5}$$

where the diagonal matrix $\Lambda = \Sigma^{\mathsf{T}}\Sigma$, with $\mathbf{P}$ defined as the columns of $\mathbf{U}$ that correspond to the largest a eigenvalues.

Pearson [188] described PCA as the optimal rank a approximation of a data matrix $\mathbf{X}$ for $a < d$ using the least-squares criterion. Here, the observed data are modeled as

$$\hat{\mathbf{x}}_i = \mathbf{P}\mathbf{t}_i + \mu \tag{4.6}$$

where $\hat{\mathbf{x}}_i$ is the reconstruction of a column of the previously defined data matrix $\mathbf{X}$, $\mathbf{P}$ is again an orthogonal matrix, $\mathbf{t}_i$ is the score and is equivalent to a column of the previously defined matrix $\mathbf{T}$, and μ is the mean of the observed data such that the reconstruction error

$$C = \sum_{i=1}^{n} ||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2 \tag{4.7}$$

is minimized.

PCA can also be described as the maximum likelihood solution of a probabilistic latent variable model [248, 208]. This formulation is referred to as *PPCA*. PPCA assumes the data are modeled by a generative latent variable model,

$$\mathbf{x}_i = \mathbf{P}\mathbf{t}_i + \mu + \epsilon_i, \tag{4.8}$$

where the variables are defined as above and ϵ_i is the error. The distributional assumptions are

$$\mathbf{t}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_a) \tag{4.9}$$

$$\epsilon_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d) \tag{4.10}$$

$$\mathbf{x}_i | \mathbf{t}_i \sim \mathcal{N}(\mathbf{P}\mathbf{t}_i + \mu, \sigma^2 \mathbf{I}_d) \tag{4.11}$$

$$\mathbf{x}_i \sim \mathcal{N}(\mu, \mathbf{P}\mathbf{P}^\top + \sigma^2 \mathbf{I}_d) \tag{4.12}$$

where $\mathbf{I}_k$ is the $k \times k$ identity matrix, $\mathcal{N}(\mu, \Sigma)$ indicates a normal distribution with mean μ and covariance Σ, and all other terms are defined as above. Tipping and Bishop [248] and Roweis [208] independently proposed finding the maximum likelihood estimates of the distributional parameters via expectation maximization (EM). EM is a general framework for learning parameters with incomplete data which iteratively updates the expected complete data log-likelihood and the maximum likelihood estimates of the parameters [56]. In PPCA, the data are incomplete because the principal components, $\mathbf{t}_i$, are not observed. Typically, $\mathbf{t}_i$ are referred to as latent variables, as opposed to missing data, because they cannot be observed. Generally, EM is only guaranteed to converge to a local maximum, but Tipping and Bishop [248] showed that EM converges to a global maximum for PPCA. To apply EM to PPCA, first the observed data are mean-centered using the sample mean. Then the algorithm alternates between calculating the conditional expectations of the latent variables,

$$\langle \mathbf{t}_i \rangle = \mathbf{W}^{-1}\mathbf{P}^\top(\mathbf{x}_i - \mu), \tag{4.13}$$

$$\langle \mathbf{t}_i \mathbf{t}_i^\top \rangle = \sigma^2 \mathbf{W}^{-1} + \langle \mathbf{t}_i \rangle \langle \mathbf{t}_i \rangle^\top, \tag{4.14}$$

where $\mathbf{W} = \mathbf{P}^\top\mathbf{P} + \sigma^2 \mathbf{I}_a$, and updating the parameters

$$\mathbf{P} = \left(\sum_{i=1}^{n} (\mathbf{x}_i - \mu) \langle \mathbf{t}_i \rangle^\top \right) \left(\sum_{i=1}^{n} \langle \mathbf{t}_i \mathbf{t}_i^\top \rangle \right)^{-1} \tag{4.15}$$

$$\sigma^2 = \frac{1}{nd} \sum_{i=1}^{n} \Big(\|\mathbf{x}_i - \mu\|^2 - 2\langle \mathbf{t}_i \rangle^\top \mathbf{P}^\top (\mathbf{x}_i - \mu) + \mathrm{tr}(\langle \mathbf{t}_i \mathbf{t}_i^\top \rangle \mathbf{P}^\top \mathbf{P}) \Big) \tag{4.16}$$

Before application of the PCA algorithm, each measurement (i.e., row when $\mathbf{X} \in \mathbb{R}^{d \times n}$) in the data matrix is typically mean centered around zero and rescaled to have standard deviation equal to one. For all PCA implementations, it is necessary to choose the latent dimension a, and several approaches exist. Scree plots [35] visualize the singular values in decreasing order and look for an "elbow" or "gap" and truncate at that point. The *percent variance explained* approach considers the variance, defined as the square of the corresponding singular value, of each loading vector and truncates at a specified threshold, often 90% or 95%. Cross-validation strategies choose a such that the reconstruction error of a held-out set is minimized. In the PPCA framework, the negative log-likelihood of a validation set can also be used. Parallel analysis [98] compares the scree plot of the data matrix to that of a random matrix of the same size and thresholds at the crossing point. Donoho and Gavish [57] propose an optimal threshold based on the asymptotic mean-squared error.

4.2.2 PCA methods for missing data

To apply an algorithm to a dataset with missing data, the simplest approaches are *complete case analysis*, in which only samples that have all of the measurements are used in analysis, and *mean imputation*, in which missing elements are replaced with the sample mean. These techniques can lead to large amounts of data loss or bias and are undesirable. Because complete case analysis and mean imputation first address missing data and then proceed with modeling, these techniques are referred to as *two-step* procedures. More advanced two-step procedures exist, such as *multiple imputation* [217], as well as two-step procedures that are designed for certain types of missingness, such as *lifting* [134] which is applied to multi-rate missingness. Here, the focus is on methods that integrate missing data handling and model building for PCA. All of the PCA methods in the previous section assume that the data matrix is complete, however in practice, the data matrix may not be complete and several approaches have been proposed for finding the principal components in the presence of missing data.

Grung and Manne [88] proposed an alternating least-squares type of approach.

Their algorithm is initialized by computing the singular value decomposition where missing values have been filled in using the sample mean. The algorithm then alternates between minimizing

$$C = \sum_{ij}(1 - M_{ij})\big(X_{ij} - \sum_k t_{ik}p_{jk}\big)^2 \tag{4.17}$$

with either fixed scores $\mathbf{T}$, or fixed loadings $\mathbf{P}$ where $M_{ij} = 1$ if X_{ij} is missing and zero otherwise. The first set of update equations are

$$\mathbf{t}_i^\top = \mathbf{x}_i^\top \mathbf{A}_i(\mathbf{A}_i^\top \mathbf{A}_i)^{-1} \tag{4.18}$$

where $\mathbf{t}_i$ is the ith column of $\mathbf{T}$, $\mathbf{x}_i$ is the ith column of $\mathbf{X}$, and $\mathbf{A}_i$ is a $d \times a$ matrix with elements $A_{jk} = (1 - M_{ij})p_{jk}$. The second set of update equations is

$$\mathbf{p}_j^\top = (\mathbf{B}_j^\top \mathbf{B}_j)^{-1}\mathbf{B}_j^\top \mathbf{x}_j^\top \tag{4.19}$$

where $\mathbf{p}_j$ is the jth row of $\mathbf{P}$, $\mathbf{x}_j$ is the jth row of $\mathbf{X}$, and $\mathbf{B}_j$ is a $n \times a$ matrix with elements $B_{ik} = t_{ik}(1 - M_{ij})$. To address the estimation of μ, Grung and Manne [88] suggest augmenting the model with an additional loading vector with a corresponding principal component equal to all ones. This approach leverages the reconstruction error derivation of the PCA problem and uses the change in the reconstruction error as the convergence criteria.

Another approach is to start from the SVD derivation of PCA. The origin of this method is unclear, with Troyanskaya et al. [253] and Walczak and Massart [270] both studying alternating algorithms utilizing the SVD. The algorithm is initialized as before, using mean imputation. The singular value decomposition is then performed and the data matrix is reconstructed. The missing elements are replaced using the reconstructed elements and the algorithm continues until convergence. Convergence is again based on the reconstruction error of the observed data. This approach is referred to as *SVDImpute* here.

Imtiaz and Shah [107] alter SVDImpute to account for measurement error by combining the ideas of SVD-based imputation with bootstrap re-sampling, which is referred to as *PCA-data augmentation* (PCADA). In this approach, when replacing the missing elements with the reconstructions, the estimates are augmented with residuals from the observed data. The residuals are defined as

$$R_{ij} = X_{ij}^{obs} - \hat{X}_{ij}^{obs} \tag{4.20}$$

and the missing data estimates are

$$\tilde{X}_{ij}^{mis} = \hat{X}_{ij}^{mis} + R_{kj} \tag{4.21}$$

where k is a random integer between 1 and n. The reconstruction estimates using $\tilde{X}_{ij}^{mis}$ are then used in the next iteration. To calculate the SVD, K bootstrap datasets are created by randomly drawing samples from the reconstructed data. The loading matrix is then calculated from

$$\tilde{P} = \frac{1}{K}\sum_{k=1}^{K} P_k \tag{4.22}$$

with $\tilde{P}$ then used in the reconstruction step. Convergence is based on the reconstruction error of the observed data, which is not guaranteed to decrease at each iteration due to the stochastic nature of the algorithm.

Another approach to performing PCA in the presence of missing data utilizes the PPCA formulation. The EM framework is amenable to problems with missing data and the framework as applied to PPCA can be extended to account for missing observations [249]. In the E-step, the expectation of the complete-data log-likelihood is taken with respect to the conditional distribution of the unobserved variables given the observed variables. Two approaches to this expectation calculation have been proposed in the literature. Ilin and Raiko [105] propose using an element-wise version of PPCA and taking the expectation using $\mathbf{T}$ as the unknown variables, i.e. missing data, and $\mathbf{P}$, μ, and σ^2 as the parameters. The resulting update equations are

$$\langle \mathbf{t}_i \rangle = \mathbf{W}_i^{-1} \sum_{j \in \mathbf{o}_i} \mathbf{p}_j (x_{ij} - \mu_j), \tag{4.23}$$

$$\langle \mathbf{t}_i \mathbf{t}_i^\top \rangle = \sigma^2 \mathbf{W}_i^{-1} + \langle \mathbf{t}_i \rangle \langle \mathbf{t}_i \rangle^\top \tag{4.24}$$

where $\mathbf{W}_i = \sum_{j \in \mathbf{o}_i} \mathbf{p}_j \mathbf{p}_j^\top + \sigma^2 \mathbf{I}_a$,

$$\mu_j = \frac{1}{\#(\mathbf{o}_j)} \sum_{i \in \mathbf{o}_j} \big(x_{ij} - \mathbf{p}_j^\top \langle \mathbf{t}_i \rangle \big), \tag{4.25}$$

$$\mathbf{p}_j = \big(\sum_{i \in \mathbf{o}_j} \langle \mathbf{t}_i \mathbf{t}_i^\top T \rangle \big)^{-1} \sum_{i \in M_j} \big(\langle \mathbf{t}_i \rangle (x_{ij} - \mu_j) \big), \tag{4.26}$$

$$\sigma^2 = \frac{1}{\#(\mathbf{O})} \sum_{ij \in \mathbf{O}} \big((x_{ij} - \mathbf{p}_j^\top \langle \mathbf{t}_i \rangle - \mu_i)^2 + \mathbf{p}_j^\top \sigma^2 \mathbf{W}_i^{-1} \mathbf{p}_j \big), \tag{4.27}$$

$\mathbf{O} = 1 - \mathbf{M}$ is the observed data indicator matrix, and $\#(\cdot)$ represents the number of observed elements in the set. Alternatively, the unknown variables can be taken to be $\mathbf{T}$ and the missing elements of the data matrix $\mathbf{X}$ [156, 293]. The resulting update equations are

$$\langle \mathbf{t}_i \rangle = \mathbf{W}_i^{-1} \sum_{j \in \mathbf{o}_i} \mathbf{p}_j (x_{ij} - \mu_j) \tag{4.28}$$

$$\langle x_{ij} \rangle = \begin{cases} \mathbf{p}_j \langle \mathbf{t}_i \rangle + \mu_j & \text{if } M_{ij} = 1 \\ x_{ij} & \text{if } M_{ij} = 0 \end{cases} \tag{4.29}$$

$$\langle \mathbf{t}_i \mathbf{t}_i^\top \rangle = \sigma^2 \mathbf{W}_i^{-1} + \langle \mathbf{t}_i \rangle \langle \mathbf{t}_i \rangle^\top \tag{4.30}$$

$$\langle \mathbf{x}_i \mathbf{x}_i^\top \rangle_{jk} = \begin{cases} \sigma^2 (\mathbf{p}_j \mathbf{W}_i^{-1} \mathbf{p}_k^\top) + \langle x_{ij} \rangle \langle x_{ik} \rangle & \text{if } M_{ij} = M_{ik} = 1, \ \ \forall j \neq k \\ \sigma^2 (1 + \mathbf{p}_j \mathbf{W}_i^{-1} \mathbf{p}_k^\top) + \langle x_{ij} \rangle \langle x_{ik} \rangle & \text{if } M_{ij} = M_{ik} = 1, \ \ \forall j = k \\ \langle x_{ij} \rangle x_{ij} & \text{if } M_{ij} = 1, \ M_{ik} = 0 \\ x_{ij} \langle x_{ik} \rangle & \text{if } M_{ij} = 0, \ M_{ik} = 1 \\ x_{ij} x_{ik} & \text{if } M_{ij} = M_{ik} = 0 \end{cases} \tag{4.31}$$

$$\langle \mathbf{x}_i \mathbf{t}_i^\top \rangle = \begin{cases} \sigma^2 \mathbf{p}_j \mathbf{W}_i^{-1} + \langle \mathbf{x}_i \rangle \langle \mathbf{t}_i \rangle^\top & \text{if } M_{ij} = 1 \\ \mathbf{x}_i \langle \mathbf{t}_i \rangle^\top & \text{if } M_{ij} = 0 \end{cases} \tag{4.32}$$

where $\mathbf{W}_i = \sum_{j \in \mathbf{o}_i} \mathbf{p}_j \mathbf{p}_j^\top + \sigma^2 \mathbf{I}_a$ and

$$\mu = \frac{1}{n} \sum_{i=1}^{n} \langle \mathbf{x}_i \rangle - \mathbf{P} \langle \mathbf{t}_i \rangle \tag{4.33}$$

$$\mathbf{P} = \left(\sum_{i=1}^{n} (\langle \mathbf{x}_i \mathbf{t}_i \rangle^\top - \mu \langle \mathbf{t}_i \rangle^\top) \right) \left(\sum_{i=1}^{n} \langle \mathbf{t}_i \mathbf{t}_i^\top \rangle \right)^{-1} \tag{4.34}$$

$$\sigma^2 = \frac{1}{nd} \sum_{i=1}^{n} \operatorname{tr}\Big(\langle \mathbf{x}_i \mathbf{x}_i^\top \rangle - 2 \langle \mathbf{x}_i \mathbf{t}_i^\top \rangle \mathbf{P}^\top - 2\mu \langle \mathbf{x}_i \rangle^\top + 2\mu \langle \mathbf{t}_i \rangle^\top \mathbf{P}^\top + \mathbf{P} \langle \mathbf{t}_i \mathbf{t}_i^\top \rangle \mathbf{P}^\top + \mu \mu^\top \Big). \tag{4.35}$$

Performing PPCA using this conditioning set is referred to here as *PPCA-M.*

Bayesian PCA (BPCA) is a variation on the PPCA approach [15]. A limitation of PPCA is that the method can be prone to overfitting [105], which BPCA attempts to prevent by using a prior distribution on the parameters. Conjugate priors are used for

μ and σ^2 and a hierarchical prior is used for $\mathbf{P}$. When the PPCA problem is modified in this way, the E-step no longer has a closed form and variational approaches are preferred [170]. Oba et al. [176] extended the BPCA method to cases with missing data.

The last approaches for PCA in the presence of missing data presented here are from the matrix completion literature. In matrix completion, sometimes also referred to as robust PCA, elements of a matrix are corrupted and the goal is to recover a low rank reconstruction. If the corrupted elements are treated as missing, this is exactly the same problem as has been discussed, however the problem is often framed directly as the optimization

$$\underset{\mathbf{A}}{\text{minimize}}\ \|\mathbf{A}\|_*, \qquad \text{subject to } A_{ij} = X_{ij}, \quad (i,j) \in \mathbf{O}, \tag{4.36}$$

where $\|\cdot\|_*$ denotes the nuclear norm of a matrix, which is the sum of the singular values of the matrix, X_{ij} are the observed elements in the data matrix, and $\mathbf{O}$ is the set of observed indices. An approach for solving this problem is singular value thresholding (SVT) [33], which solves

$$\underset{\mathbf{A}}{\text{minimize}}\|\mathbf{A}\|_*, \qquad \text{subject to } \mathcal{P}_{\mathbf{O}}(\mathbf{A}) = \mathcal{P}_{\mathbf{O}}(\mathbf{X}), \tag{4.37}$$

where $\mathcal{P}_{\mathbf{O}}$ is the orthogonal projector onto the span of matrices vanishing outside of $\mathbf{O}$. Cai et al. [33] propose an alternating algorithm that approximately solves (4.37) which results in a matrix that is sparse and low rank. A second approach for the matrix completion problem is the inexact augmented Lagrange multiplier method (ALM) [140], which solves

$$\underset{\mathbf{A}}{\text{minimize}}\|\mathbf{A}\|_*, \qquad \text{subject to } \mathbf{A} + \mathbf{E} = \mathbf{X}, \ \ \mathcal{P}_{\mathbf{O}}(\mathbf{E}) = 0, \tag{4.38}$$

where $\mathcal{P}_{\mathbf{O}}$ is a linear operator that also is zero outside of $\mathbf{O}$. ALM was proposed to solve the more general problem of a corrupted matrix without knowledge of which entries are corrupted but can also be applied in this setting.

4.3 Case study

The performance of the different techniques are compared in several case studies. Two types of simulations are considered: one based on distributional assumptions and one based on a chemical process simulation.

4.3.1 Simulations of Gaussian data

The design of the distributional-assumption simulations is based on the study by Ilin and Raiko [105] and uses data from multivariate Gaussian distributions. The distributional assumptions follow the development of the PPCA model. While data that exactly follow the model are idealized, the assumptions approximately hold for data that have been pre-processed using standard methods. That is, data that have been pre-processed by sub-sampling and z-scoring approximately have independent and identically distributed multivariate Gaussian (symmetric) distributions. This type of pre-processing can introduce error in the presence of missing data, particularly if missingness is due to censoring. Therefore, this analysis lays a foundation of the best-case results.

The loading matrix $\mathbf{P}$ is modeled using a random orthogonal matrix of size $d \times a$ where $a = 4$ and the columns of $\mathbf{P}$ rescaled by $1, \ldots, a$. μ is modeled using a standard normal distribution. Two scenarios are considered. In the first, $n \gg d$. Specifically, the dataset is $n = 1000$ samples from a 10-dimensional Gaussian distribution described by $\mathcal{N}(\mu, \mathbf{P}\mathbf{P}^\top + \sigma^2\mathbf{I}_d)$ where $\sigma^2 = 0.25$. In the second scenario, the opposite case is considered, $d > n$, and $n = 100$ samples from a 200-dimensional Gaussian distribution described by $\mathcal{N}(\mu, \mathbf{P}\mathbf{P}^\top + \sigma^2\mathbf{I}_d)$ where $\sigma^2 = 0.25$. For each of the scenarios, 20 simulations are used, each with four types of missingness, described below.

Ten PCA approaches were tested: mean imputation (MI), alternating least squares (ALS) as implemented by MATLAB's pca command, alternating least squares (Alternating) as implemented by Ilin and Raiko [105], SVDImpute as implemented by Ilin and Raiko [105], PCADA as implemented by the authors, PPCA as implemented by MATLAB's ppca command, PPCA-M as implemented by the authors, BPCA as implemented by Oba et al. [176], SVT as implemented by Cai et al. [33], and ALM as implemented by Lin et al. [140]. All approaches were implemented in MATLAB, used a convergence tolerance of 10^{-6}, and were limited to 1000 iterations. Alternating, SVDImpute, PCADA, BPCA, SVT, and ALM use relative change in the reconstruction error as the convergence criteria. ALS uses relative change in the reconstruction error as well as the relative change in the parameters are the convergence criteria. PPCA and PPCA-M use the relative changes in the negative log-likelihood and parameters as the convergence criteria.

To evaluate performance, two metrics were used: the root mean square error (RMSE), and the subspace angle between the true and recovered principal component loadings. The RMSE is defined

$$\text{RMSE} = \sqrt{\frac{1}{nd}\sum_{i=1}^{n}\sum_{j=1}^{d}(x_{ij} - \hat{x}_{ij})^2} \tag{4.39}$$

and is reported for only the missing data. The full definition of the subspace angle is provided in the Appendix 4.5. A subspace angle of 0 implies that the subspaces are dependent, which is the desired result here. The maximum value of the subspace angle is $\frac{\pi}{2}$. In all analysis, the subspace angle is calculated using the MATLAB function `subspace`.

4.3.2 Tennessee Eastman problem

The Tennessee Eastman problem (TEP) is a benchmark dataset that models an industrial chemical process [58]. The benchmark contains datasets both under normal operation as well as during several process faults. The process consists of five major units: reactor, condenser, compressor, separator, and stripper. There are 8 components, 41 measured variables, and 11 manipulated variables. Several control structures have been proposed for plant-wide control of the TEP. The datasets can be found online [210] and utilize "control structure 2" as described by Lyman and

Geogakis [151]. Unlike the Gaussian data simulations, the latent dimension a is unknown. To determine a, parallel analysis was used. Three missingness mechanisms were considered, as described below, and 20 simulations were used for each. The same 10 approaches for PCA as described above were implemented with a small change to the mean imputation approach. Because the data are collected in time, the last measurement before and the first measurement after the missing data point are averaged and used to fill-in. The learned model is then used in two tasks: reconstruction of a test dataset and fault detection. For the fault detection problem, the Q statistic, defined as

$$Q = \mathbf{r}^\top \mathbf{r}, \quad \mathbf{r} = (\mathbf{I}_d - \mathbf{P}\mathbf{P}^\top)\mathbf{x}_i, \tag{4.40}$$

was used. The Q statistic, also known as the squared prediction error, has been well studied in the area of fault detection [110, 285, 127, 211]. To determine the detection threshold, the tenth largest value of Q on the nominal test set was used [211].

To evaluate the performance, three metrics were used: the RMSE on a held-out test of nominal data, the detection time, and whether or not a false detection occurred. Two faults are chosen for analysis: Fault 1, which is a step change in A/C feed ratio in stream 4, and Fault 13, which is a slow drift of the reaction kinetics. In both cases, the testing dataset is used and the faults are introduced at $t = 160$. The mean detection time is defined as the average detection time for all models in which the detection time is greater than 160 and the number of false detections is defined as the number of models where there is a detection before 160. For a given model, either a detection time or a false detection time is recorded.

4.3.3 Addition of missing data

Four types of missingness were considered: random, sensor drop-out, multi-rate, and censoring. The types of missingness were chosen based on the authors' experience with realizations of missing data in process datasets. Random, sensor drop-out, and multi-rate missingness are all MCAR but have different patterns: random exhibits no pattern, sensor drop-out is correlated in time, and multi-rate has a known frequency

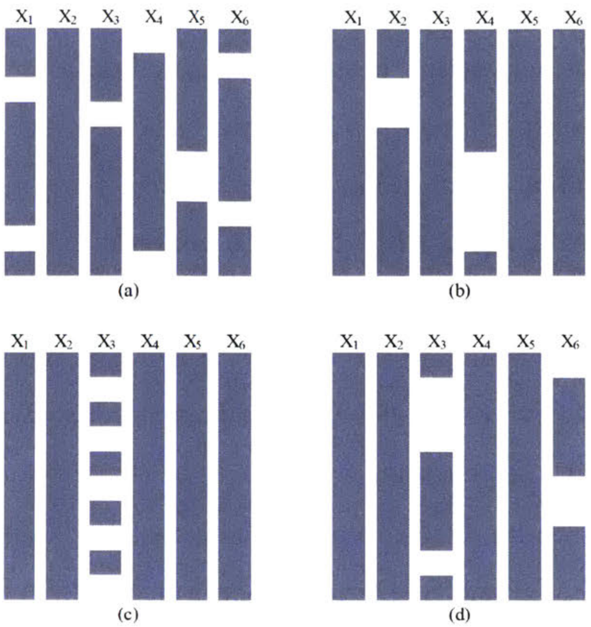

X_1 X_2 X_3 X_4 X_5 X_6
(a)
X_1 X_2 X_3 X_4 X_5 X_6
(b)
X_1 X_2 X_3 X_4 X_5 X_6
(c)
X_1 X_2 X_3 X_4 X_5 X_6
(d)

Figure 4-1: Possible realizations of the investigated missingness mechanisms: (**a**) shows random missingness; (**b**) shows sensor failure which results in missingness that is correlated in time; (**c**) shows multi-rate data, and (**d**) shows censored data.

of missingness in time. Censor missingness is NMAR. Examples of the patterns are shown in Figure 4-1. In all cases, a full dataset is generated or obtained and measurements are removed to represent the missing data mechanism. For instance, in the censoring case, a random set of variables is selected to be censored from above or below. The censoring level for each variable is then iteratively updated until the desired level of missingness is achieved. The location of the code used to introduce missing can be found in the Supplementary Materials. Missing data are introduced at levels of 1, 5, 10, and 15% for the Gaussian datasets. The multi-rate pattern is not considered for the 1% missingness level for the Gaussian datasets. The TEP is naturally a multi-rate missing data problem at a level of 21% [211]. TEP is individually combined with random, sensor drop-out, and censored missingness to total 25%.

4.3.4 Results

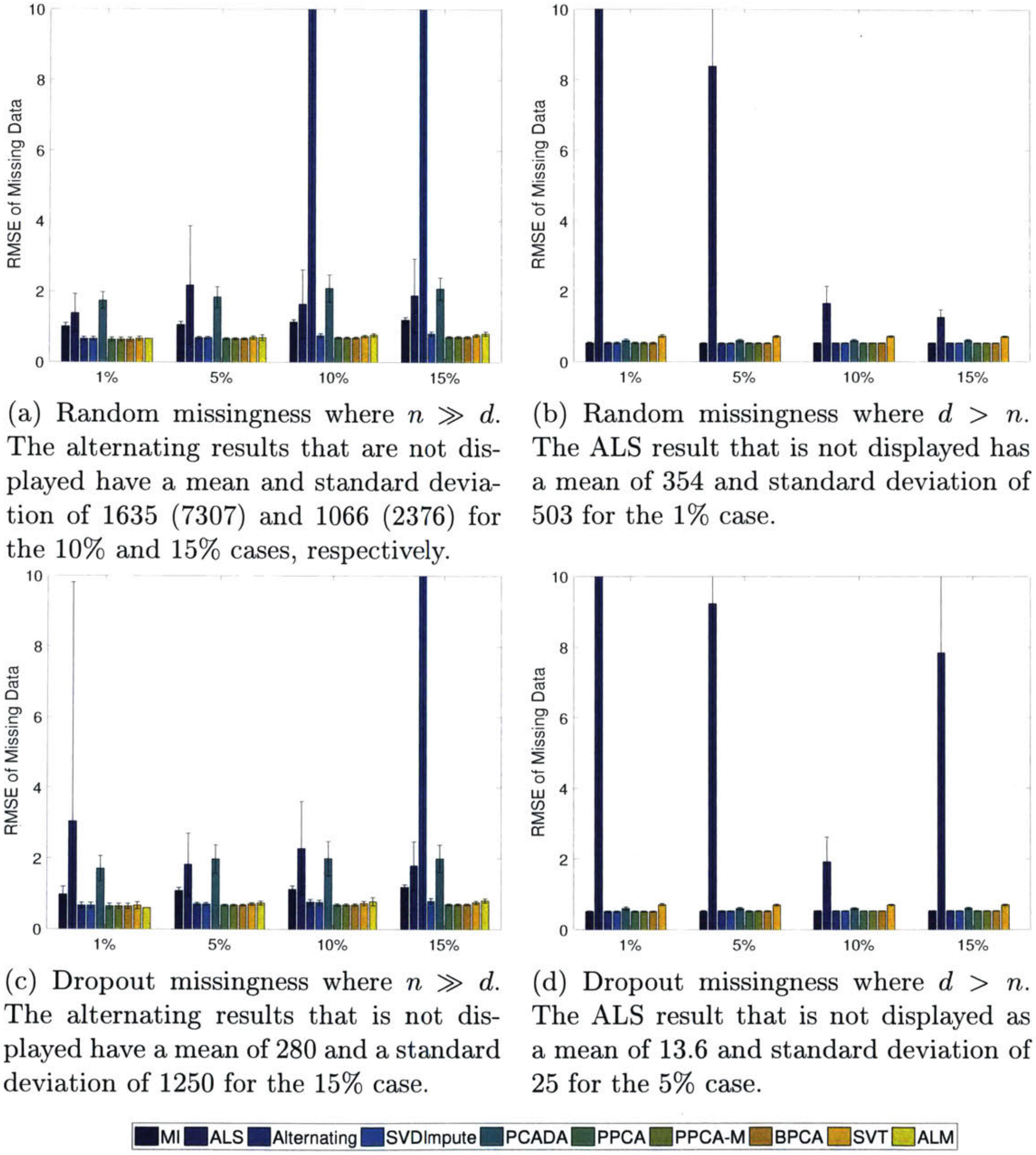
The results of the Gaussian simulations are shown in Figures 4-2–4-4. SVDImpute and the probabilistic methods (PPCA, PPCA-M, and BPCA) performed the best overall.

As the missingness level increased, the probabilistic models performed slightly better, except for SVDImpute performing better for censored data at low levels of missingness. PCADA never outperformed SVDImpute. ALS and the alternating methods both suffered from finding local optima and performed very poorly, as evidenced by the large standard deviations. ALM failed to converge in many cases, and sometimes in all cases, as in the $d > n$ scenarios. The SVT approach fell in the middle while never outperforming the best approaches. For $d > n$, most approaches did only slightly better than mean imputation whereas significant improvements were observed for $n \gg d$, especially in the censoring case.

For the TEP, the results of the reconstruction task are shown in Figure 4-5. For all missingness types, ALS and SVDImpute performed well. ALM failed to converge and Alternating and BPCA had poor results. PPCA, PPCA-M, and SVT performed moderately well, but were more affected by censoring than ALS and SVDImpute. The minimum, average, and maximum number of PCs used in the models, as determined by parallel analysis can be found in Table 4.1. The number of PCs chosen by SVDImpute, PPCA, and BPCA were very consistent whereas Alternating and PCADA had widely varying number of PCs. Across all methods, the amount of variability in the number of PCs is larger in the censoring case. The results of the fault detection task are in Tables 4.2 and 4.3. For Fault 1, ALS and SVD had the best performance overall, with low detection times and few false detections. MI performed well in terms of detection time but had many false detections. PCADA and BPCA performed the worst overall. For Fault 13, SVT performed the best in the random and drop-out cases, whereas SVDImpute performed the best for the censoring case. PCADA and BPCA again performed the worst overall. ALM was excluded from analysis as no model was learned during the training phase.

4.4 Discussion

Overall, the best technique to apply PCA in the presence of missing data can depend on the scenario. Several criteria should be considered when choosing an approach,



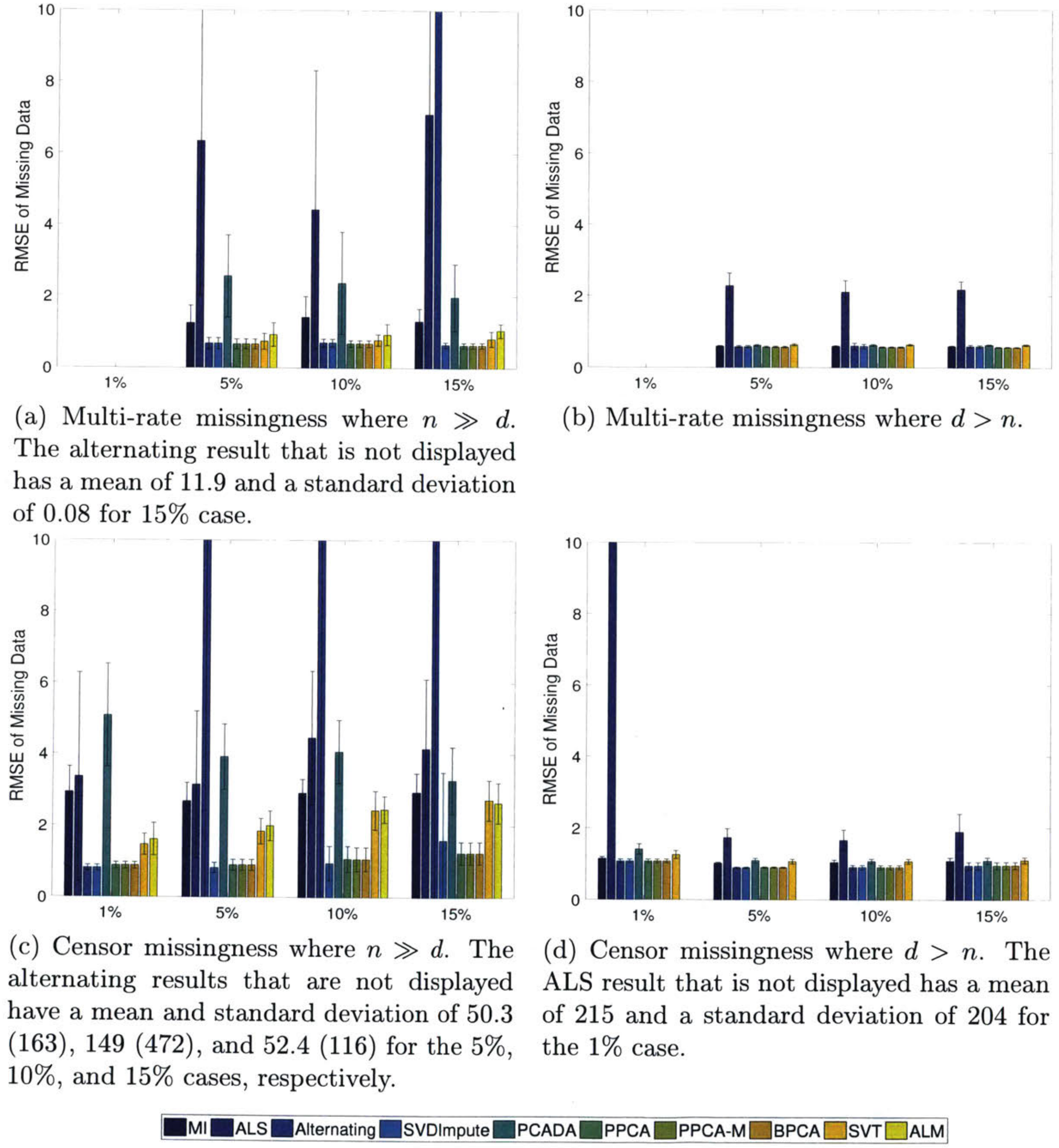
(a) Random missingness where $n \gg d$. The alternating results that are not displayed have a mean and standard deviation of 1635 (7307) and 1066 (2376) for the 10% and 15% cases, respectively.

(b) Random missingness where $d > n$. The ALS result that is not displayed has a mean of 354 and standard deviation of 503 for the 1% case.

(c) Dropout missingness where $n \gg d$. The alternating results that is not displayed have a mean of 280 and a standard deviation of 1250 for the 15% case.

(d) Dropout missingness where $d > n$. The ALS result that is not displayed as a mean of 13.6 and standard deviation of 25 for the 5% case.

Figure 4-2: Average RMSE of the missing data with standard deviation for the Gaussian cases. In the $d > n$ case, ALM never converged to a solution.



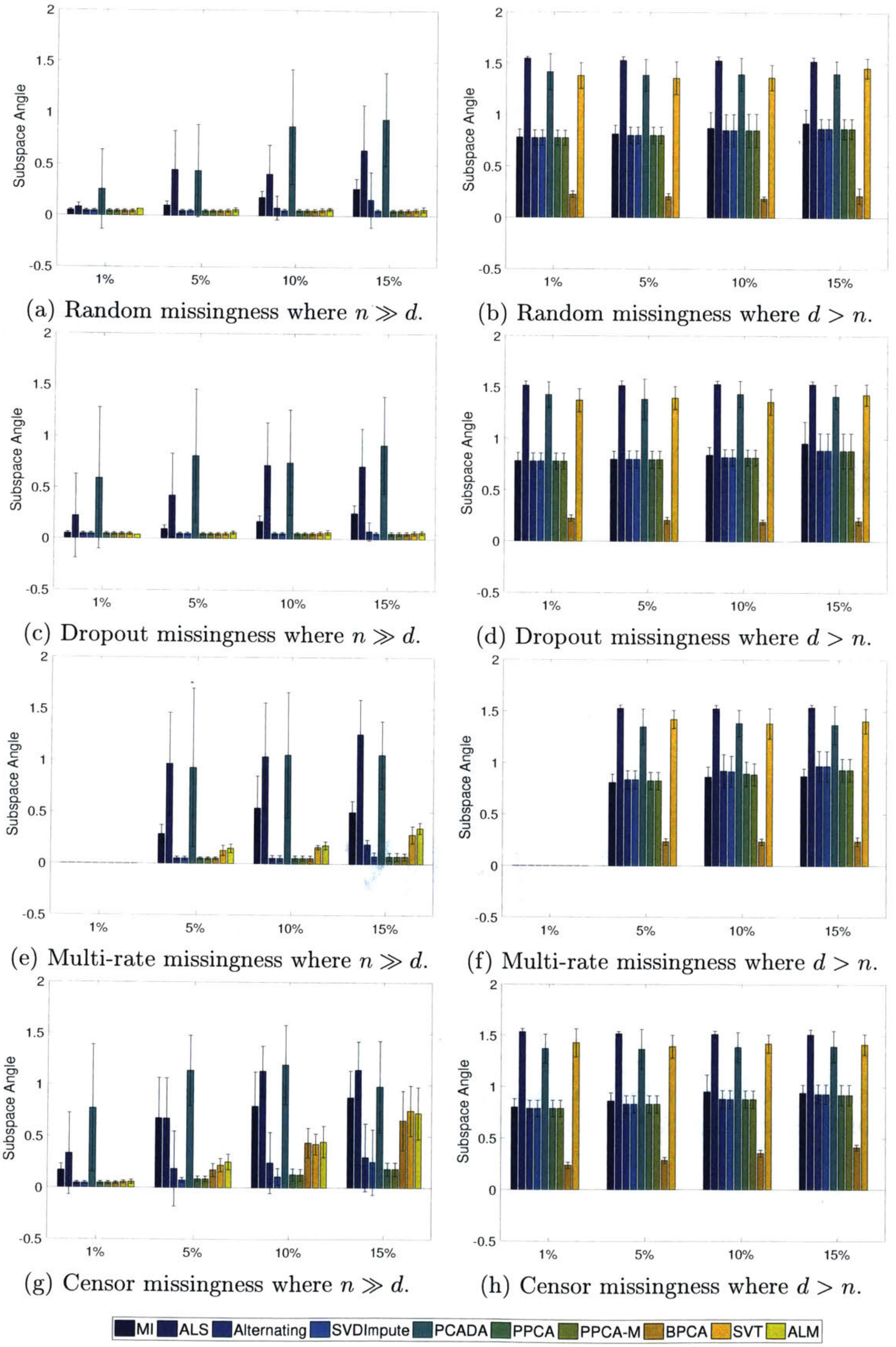
(a) Multi-rate missingness where $n \gg d$. The alternating result that is not displayed has a mean of 11.9 and a standard deviation of 0.08 for 15% case.

(b) Multi-rate missingness where $d > n$.

(c) Censor missingness where $n \gg d$. The alternating results that are not displayed have a mean and standard deviation of 50.3 (163), 149 (472), and 52.4 (116) for the 5%, 10%, and 15% cases, respectively.

(d) Censor missingness where $d > n$. The ALS result that is not displayed has a mean of 215 and a standard deviation of 204 for the 1% case.

Figure 4-3: Average RMSE of the missing data with standard deviation for the Gaussian cases. In the $d > n$ case, ALM never converged to a solution.



(a) Random missingness where $n \gg d$.

(b) Random missingness where $d > n$.

(c) Dropout missingness where $n \gg d$.

(d) Dropout missingness where $d > n$.

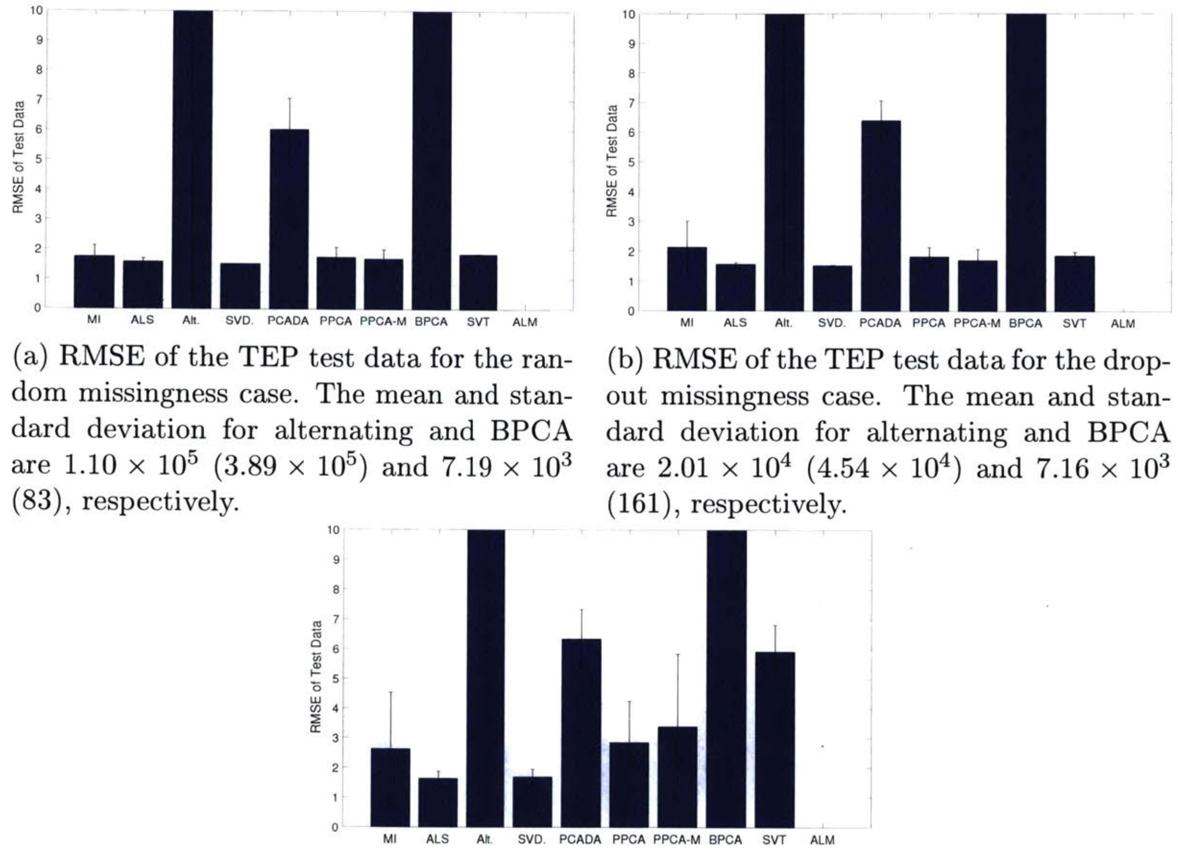
(e) Multi-rate missingness where $n \gg d$.

(f) Multi-rate missingness where $d > n$.

(g) Censor missingness where $n \gg d$.

(h) Censor missingness where $d > n$.

Figure 4-4: Average subspace angle of learned vs. true subspace with standard deviation for the Gaussian cases.



(a) RMSE of the TEP test data for the random missingness case. The mean and standard deviation for alternating and BPCA are 1.10×10^5 (3.89×10^5) and 7.19×10^3 (83), respectively.

(b) RMSE of the TEP test data for the dropout missingness case. The mean and standard deviation for alternating and BPCA are 2.01×10^4 (4.54×10^4) and 7.16×10^3 (161), respectively.

(c) RMSE of the TEP test data for the censor missingness case. The mean and standard deviation for alternating and BPCA are 3.98×10^6 (1.13×10^7) and 7.2×10^3 (685), respectively.

Figure 4-5: Average RMSE and standard deviation of the fully observed TEP test set. In all cases ALM failed to converge.

Table 4.1: The minimum, average, and maximum number of PCs chosen using parallel analysis for each method over 20 realizations of the missing data. Each missingness type is combined with the naturally arising multi-rate missingness to total 25% missing data. ALM never converged and therefore no results are reported.

	MI	**ALS**	**Alt.**	**SVD.**	**PCADA**	**PPCA**	**PPCA-M**	**BPCA**	**SVT**	**ALM**
Random										
Min	2	3	1	3	1	3	4	3	4	–
Avg	2.95	3.2	4.15	3	2.55	3	4.3	3	4.95	–
Max	3	4	7	3	4	3	5	3	5	–
Drop										
Min	1	3	1	3	1	3	3	3	4	–
Avg	3.15	3.3	4.15	3	2.65	3	4.05	3	4.9	–
Max	4	4	6	3	5	3	5	3	5	–
Censoring										
Min	1	3	1	2	1	2	1	2	1	–
Avg	3	3.5	3.65	2.9	2.6	2.85	3.3	2.9	1.65	–
Max	4	5	7	3	7	3	5	3	4	–

Table 4.2: The mean detection times for each of the methods and missingness types. Cases are marked by "–" where every trial resulted in a false detection (e.g., a detection prior to $t = 160$).

	MI	**ALS**	**Alt.**	**SVD.**	**PCADA**	**PPCA**	**PPCA-M**	**BPCA**	**SVT**
Fault 1									
Random	163.1	163	163	163	–	163.8	163.1	–	171.0
Drop	163	163	–	163	–	163.7	163.4	–	170.5
Censor	163.1	163.2	163	163.5	–	163.2	163.4	–	–
Fault 13									
Random	182	181.8	210	182	–	180.3	183.2	–	174
Drop	182	181.4	–	181.3	–	182.3	179.3	–	174.5
Censor	180.3	181.9	411	184.9	–	185	189.7	–	–

Table 4.3: The number of false detections for each of the methods and missingness types.

	MI	ALS	Alt.	SVD.	PCADA	PPCA	PPCA-M	BPCA	SVT
Fault 1									
Random	0	0	19	0	20	2	0	20	0
Drop	9	0	20	0	20	1	1	20	1
Censor	5	3	19	3	20	6	9	20	20
Fault 13									
Random	7	3	19	1	20	4	4	20	0
Drop	11	4	20	5	20	5	4	20	0
Censor	12	9	19	8	20	19	17	20	20

such as the amount of missing data, the missingness mechanism, and the available computational resources. The computational complexity per iteration for each of the algorithms can be found in Table 4.4, which should only be used as a guideline since the exact implementation will affect computational cost. For instance, SVT [33] and ALM [140] recommend using the Lanczos algorithm to compute the singular values. The Lanczos algorithm is iterative and has reported speed-up of 10 × vs. traditional calculation of the full SVD. The Lanczos algorithm returns the singular values that are larger than a certain threshold, which works well in the SVT and ALM frameworks. On the other hand, Lin et al. [140] report that the full SVD computation is faster for scenarios where greater than $0.2d$ of singular values are required. While experience indicates that a is significantly lower than d in applications, if no bound on a is known *a priori*, then the full SVD is typically calculated during procedures to select a, which impacts the computational cost. The probabilistic frameworks have the convenient relation that

$$\sigma^2_{ML} = \frac{1}{d-a} \sum_{j=a+1}^{d} \lambda_j \tag{4.41}$$

which can be used to estimate the percent variance without calculating the full SVD. Another benefit of the probabilistic frameworks is that they are generative and therefore provide parameters for estimation. For all analysis, the test data have been treated as fully observed, which may not be true in practice as new data may be subject to the same type of missingness as the data used in model building. If the data are subject to NMAR missingness, these parameters may not be useful. Note also that the probabilistic approaches can have slow convergence.

The difference in the results of the two ALS approaches also highlights the importance of the exact implementation. Both methods are using the same underlying algorithm but differ in the implementation of the update steps and convergence criteria. Empirically, this results in the Alternating algorithm finding local optima more often as the amount of missing data increases for the $n \gg d$ case and the ALS algorithm finding local optima more often for $d > n$.

It may be surprising that the robust PCA methods (SVT and ALM) did not

Table 4.4: The computational costs of each of the methods where d is the number of measurements, n is the number of samples, a is the latent dimension, and k is the number of bootstrap samples.

ALS / Alternating / PPCA / BPCA	**SVDImpute / SVT / ALM**
$O(a^2dn + a^3n + a^3d)$	$O(\min(nd^2, n^2d))$
PCADA	**PPCA-M**
$O(\min(knd^2, kn^2d))$	$O(na^3 + nda^2)$

perform better, but it is important to recognize that these methods were developed for cases with very low rank solutions, a large number of missing values, and random missingness. These assumptions are well suited to some applications such as computer vision and imaging but do not necessarily fit the assumptions of missing data in process datasets. A benefit of SVT and ALM is that they can be applied to problems where the location of the corrupt (missing) data is unknown. In the event that additional information is known about the measurement error, methods such as maximum likelihood PCA (MLPCA) [277, 5] or heteroscedastic latent variable model (HLV) [206] can be applied to leverage that information. MLPCA is suited to scenarios where the error covariance matrix is known and the errors are correlated or uncorrelated. HLV is suited to scenarios the measurement error is evolving in time. Both algorithms can be applied to scenarios with missing data.

Without additional problem information, we recommend SVDImpute for performing PCA in the presence of missing data for industrial datasets. SVDImpute can be viewed as an implementation of EM [105]. In this view, the missing observations are treated as the unknown variables and $\mathbf{P}$, μ, σ^2, and $\mathbf{T}$ are the model parameters. The corresponding cost function, for only terms involving the parameters, is

$$C = -\frac{dn}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{ij\in\mathbf{O}}(x_{ij} - \hat{x}_{ij})^2 - \frac{1}{2\sigma^2}\sum_{ij\in\mathbf{M}}\left((\bar{x}_{ij} - \hat{x}_{ij})^2 + \sigma^2\right) \qquad (4.42)$$

where $\bar{x}_{ij}$ are the imputed values from the SVD. This cost function forces the imputed terms to be near the observed terms which helps to prevent overfitting [105]. A drawback of SVDImpute is that there are many possible reconstructions that will achieve the same result for the observed data, and different results for the missing data, which implies a dependence on the initial guess [105].

In the event that the testing data will also have missing elements, PPCA or PPCA-M is recommended. PPCA-M performs slightly better in the TEP but has higher storage costs during model training. Both result in generative parameters that can be used during the testing phase.

In summary, for missing data problems, the most important step is to determine why some data are missing. If censoring is occurring and not accounted for, the results will be biased. Approaches that incorporate understanding about the underlying mechanisms are likely to perform the best. Expectation maximization frameworks are an important tool in missing data problems and can be applied generally if distributional assumptions are made.

4.5 Definition of the subspace angle

To compute the subspace angle between matrices $\mathbf{A} \in \mathbb{R}^{n\times m}$ and $\mathbf{B} \in \mathbb{R}^{n\times p}$, where $\text{rank}(\mathbf{A}) \geq \text{rank}(\mathbf{B})$, compute the orthonormal basis of each matrix using the singular value decomposition. Then compute the projection

$$\mathbf{P} = \mathbf{B} - \mathbf{A}(\mathbf{A}^{\top}\mathbf{B}). \tag{4.43}$$

The subspace angle, θ, is defined by

$$\sin\theta = \min(1, ||\mathbf{P}||) \tag{4.44}$$

where $||\cdot||$ is the 2-norm. See [17] and [276] for additional information on subspace angles.

Chapter 5

Learning sparse classification models in the presence of missing data

This work originally appeared as: Kristen A. Severson, Brinda Monian, J. Christopher Love and Richard D. Braatz. A method for learning a sparse classifier in the presence of missing data for high-dimensional biological datasets. Bioinformatics, 33:2897-2905, 2017. It has edited to include the supplemental information in the main text.

5.1 Introduction

The recent "-omics" revolution in the biomedical sciences, fueled by the decreasing cost of high-throughput technologies and an increased desire for large numbers of measurements for valuable clinical samples, has led to the prevalence of wide datasets – that is, datasets with many more measurements per sample than samples. These datasets can be generated by technologies such as microarrays and RNA-Seq, ChIP-Seq, and proteomic and metabolomic techniques (e.g., mass spectrometry, multiplexed molecular assays). Such methods are gaining widespread popularity due to their potential to unearth new molecular targets for diagnosis and treatment, and due to the possibility of discovering combinations of molecular features that contribute to a disease state.

However, having many more measurements than samples leads to ill-conditioned datasets and can introduce statistical inference challenges. Two common problems arise when attempting to build models from wide datasets: the dataset is not full rank, which limits the applicable numerical approaches, and a high-dimensional model may be difficult to interpret. Both of these issues have led to interest in learning sparse models, where the number of predictors in the final model is a subset of the training dataset.

Because of the prevalence of this problem, there are many techniques for learning a sparse model. In this work, we focus on classification models, which are of particular interest in the biomedical field due to the goal of stratifying classes of patients (e.g., healthy vs. not healthy) or treatment conditions (e.g., treated vs. untreated). One way to learn a sparse classification model is via the nearest shrunken centroids (NSC) approach [247]. This method finds a subset of predictors by penalizing, or "shrinking", the class centroids. This technique was shown by [273] to be equivalent to applying an ℓ_1 penalty to the class means. This approach is easy to implement and has a nice visual explanation. One limitation is that the method is required to assume a diagonal structure for the covariance matrix to avoid ill-conditioning. Another approach is sparse discriminant analysis (SDA) [52]. This technique simultaneously performs model fitting and feature selection, finding the k discriminant vectors β_k by solving

$$\begin{aligned} \underset{\beta_k,\theta_k}{\text{minimize}} ||\mathbf{Y}\theta_k - \mathbf{X}\beta_k||^2 + \gamma\beta_k\Omega\beta_k + \lambda||\beta_k||_1 \\ \text{subject to } \frac{1}{n}\theta_k^\top \mathbf{Y}^\top \mathbf{Y}\theta_k = 1 \\ \theta_k^\top \mathbf{Y}^\top \mathbf{Y}\theta_l = 0, \forall l < k \end{aligned}$$

where $\mathbf{Y}$ is an $n \times K$ matrix of indicator variables of the class, $\mathbf{X}$ is an $n \times p$ data matrix, Ω is a positive-definite matrix, and λ and γ are nonnegative tuning parameters. This minimization is then solved iteratively.

A limitation of both of these approaches is their ability to handle missing data. Missing data are common in biological and social data. For example, technical issues may invalidate some results of an assay, a person may drop out of a longitudinal study, a hospital may only run some diagnostic tests given the time and availability of medical equipment, or respondents may skip certain questions in a social survey [78]. In the UC Irvine Machine Learning Repository, over 20% of the datasets have missing values. Simple techniques to handle missing data involve complete case analysis, where samples with missing data are ignored, or mean imputation, where the missing data are filled in using the observed data mean. These techniques waste data and/or introduce bias.

To address these limitations, the literature contains a significant amount of work on data imputation, particularly for microarray datasets. [253] did one of the first studies and found that k-nearest neighbors (KNN) significantly improved on complete case analysis and mean imputation. More complex techniques have been presented by [176, 23, 179, 121, 226, 274, 122]. [27] surveyed these results to help practitioners decide which methods to use. The work presented here is fundamentally different than any of these techniques because it performs missing data imputation and model building simultaneously. This simultaneous approach allows for consistent assumptions in the imputation and model-building phases, and decreases the number of algorithm decisions the analyst must make. The work of Blanchet and Vignes [19] also considers simultaneous model building and handling of missing data, but does not support a sparse model, which is a key feature of the proposed methodology. To tackle the two issues simultaneously, an expectation-maximization procedure is proposed.

The expectation-maximization (EM) framework is a way to handle instances of missing data by iteratively updating the expected complete data log-likelihood and the maximum likelihood estimate of the model parameters [56]. Although EM is a local optimization technique, the likelihood can only improve at each step and the method has been applied to many problems. The challenge of using EM is to choose an appropriate model for the data. In this work, we build on probabilistic principal component analysis, a technique that uses EM to find the principal subspace, by adding sparsity-inducing priors. This method allows the learning of a subset of predictors, even in the presence of missing data. The resulting classifier is a linear discriminant analysis model.

The proposed expectation-maximization sparse discriminant analysis (EM-SDA) algorithm addresses the intersection of these ideas to be able to tackle high-dimensional datasets that may have missing elements. The proposal is foremost meant to be able to handle expected characteristics of biological datasets, red which include correlation amongst the measurements (protein, genes, etc.) and missing elements. The model assumes a symmetric distribution, which can typically be approximated via an appropriate scaling. Scaling becomes more difficult in the presence of missing data so this model is most well-suited to high-throughput assays where many measurements are performed using the same instrument and therefore the data have similar scaling. Critical care or clinical trial datasets may be more challenging to work with because of the variety of scales, however, if past information and/or intuition of scaling is available, this method would also be appropriate. As is often important in biological settings, the resulting predictions are probabilities, which are useful when more than a yes or no answer is preferred. Because the model is generative, it is also able to make predictions on new samples that also have missing elements by performing imputation.

Section 2 introduces the proposed methodology, including procedures for cases with and without missing data. Section 3 presents simulation and case studies using synthetic and real datasets. Section 4 contains discussion and conclusions.

5.2 Approach

5.2.1 Background

Principal component analysis (PCA) is a widely used technique for dimensionality reduction. PCA constructs a linear projection that maximizes the variance in a lower dimensional space [99] and is also the optimal rank a approximation of a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ for $a < p$ based on the least-squares criterion [188]. An alternative view of

PCA as a generative latent variable model [248, 208] is

$$\mathbf{x}_i = \mathbf{W}\mathbf{t}_i + \mu + \epsilon_i \tag{5.1}$$

where $\mathbf{x}_i$ are the p-dimensional observations, $\mathbf{t}_i$ is the a-dimensional latent variables, $\mathbf{W} \in \mathbb{R}^{pxa}$ are the factor loadings, μ is a constant whose maximum likelihood estimator is the mean of the data, and ϵ_i is the error. The corresponding distributional assumptions are

$$\begin{aligned}
\mathbf{t}_i &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_a) \\
\epsilon_i &\sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_p) \\
\mathbf{x}_i | \mathbf{t}_i &\sim \mathcal{N}(\mathbf{W}\mathbf{t}_i + \mu, \sigma^2 \mathbf{I}_p) \\
\mathbf{x}_i &\sim \mathcal{N}(\mu, \mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I}_p)
\end{aligned}$$

where $\mathbf{I}_k$ is the $k \times k$ identity matrix. The model parameters $\theta = [\mathbf{W}, \mu, \sigma^2]$ are found using an expectation-maximization approach [56], which is computationally more expensive than solving the PCA problem directly using the singular value decomposition but has the benefit of being able to handle missing data [105, 156, 293]. Generally, EM is only guaranteed to converge to a local maximum of the likelihood [56]; however, [248] show EM must converge to a global maximum for the PPCA problem.

5.2.2 Motivation

Let $\mathbf{x}_i \in \mathbb{R}^p$ be a vector of measurements for observations $i = 1, \ldots, n$. Let $y_i \in \{0, 1\}$ be the class label of sample i, which is observed. The classification problem is to perform supervised training to learn a model to predict the class of a new sample. To solve this problem, a linear discriminant analysis (LDA) model is used. Because the number of samples, n, may be less than the dimension of the sample, p, the model is required to be sparse. Often when LDA is applied to datasets of this type, one of two simplifying assumptions is made: the covariance matrix has a diagonal structure, as in NSC, or a regularization penalty of the form $\lambda\mathbf{I}_p$ is added, as in SDA. The use of EM allows for a structured covariance approximation [156]. Under the generative latent variable model described by eqn. 5.1, the marginal covariance is $\mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_p$. Specifying this covariance requires estimating $pa + 1 - a(a-1)/2$ parameters: pa parameters for $\mathbf{W}$ where $a \ll p$ and 1 parameter for σ^2. The $a(a-1)/2$ term is because $\mathbf{W}$ is scaled to have orthogonal columns, each with unit length, which restricts the degrees of freedom [16]. An estimation of the covariance matrix in the full data space requires the estimation of $p(p+1)/2$ parameters, therefore using the latent variable model greatly decreases the number of parameters that need to be estimated. A relaxation of the diagonal matrix constraint is desirable because the data are often known to be correlated but with too few measurements to reliably estimate the full covariance. An example is gene microarray data in which genes that participate in a pathway are expected to be correlated [283].

The LDA model makes distributional assumptions about the data, specifically

$$\begin{aligned} Y &\sim \text{Binomial}(\pi) \\ X|Y = c &\sim \mathcal{N}(\mu^c, \Sigma), \end{aligned}$$

which is specified fully by the prior probability π, class means μ^c, and the shared covariance Σ. Here, the uninformative prior of $\pi = 0.5$ is used but the model could be extended to incorporate prior class information. The method described here learns the class means and covariance to build the classifier. The dataset is modeled in a latent space using PPCA [248, 208] and a sparsity-induced prior is used for the means [72, 182].

5.2.3 Problem formulation

The data are assumed to be modeled as

$$\mathbf{x}_i^c = \mathbf{W}\mathbf{t}_i + \mu^c + \epsilon_i \tag{5.2}$$

where $\mathbf{x}_i^c, \mu^c, \epsilon \in \mathbb{R}^p$, $\mathbf{t}_i \in \mathbb{R}^a$, $\mathbf{W} \in \mathbb{R}^{p \times a}$, i represents the experiment index, and c represents the class of the observation. This PPCA formulation is typical with the small change that $\mu^c = \bar{\mu} + \Delta^c$ where $\bar{\mu}$ is the total mean and Δ^c is the class-specific deviation. Therefore the distributions of $\mathbf{x}$ are

$$\mathbf{x}_i | \mathbf{t}_i, y_i = c \sim \mathcal{N}(\mathbf{W}\mathbf{t}_i + \bar{\mu} + \Delta^c, \sigma^2 \mathbf{I}_p)$$
$$\mathbf{x}_i | y_i = c \sim \mathcal{N}(\bar{\mu} + \Delta^c, \mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I}_p)$$

The class superscripts are dropped for convenience but the analysis assumes that all observations have class-specific means and shared covariance. A prior is set for Δ as

$$\Delta | \mathbf{T} \sim \mathcal{N}(\mathbf{0}, \mathbf{T})$$
$$\tau_j \sim \text{Gamma}\left(1, \frac{\gamma^2}{2}\right)$$

where $\mathbf{T} = \text{diag}(\tau_j)$, which is chosen because [72, 169]

$$\begin{aligned} p(\Delta_j | \gamma) &= \int_0^\infty \mathcal{N}(\Delta_j; 0, \tau_j) \text{Ga}\left(\tau_j; 1, \frac{\gamma^2}{2}\right) d\tau_j \\ &= \frac{\gamma}{2} \exp\big(- \gamma |\Delta_j| \big) = \text{Laplace}\left(\Delta_j; 0, \frac{1}{\gamma}\right) \end{aligned} \tag{5.3}$$

and the Laplace distribution is known to lead to sparse solutions [169].

5.2.4 Expectation maximization

In EM, the algorithm alternates between calculating the expected complete-data log-likelihood and the maximum likelihood estimate of the parameters. For this problem, the parameters are $\theta = [\mathbf{W}, \mu, \Delta, \sigma^2]$ and the missing data are $[\mathbf{t}_i, \tau]$ (in Section 5.2.6, this set is augmented to include missing observations from the dataset). The observed data are $\mathbf{x}_i$ and the hyperparameter for the prior on τ, γ.

For the case where there is no missing data, the complete data log-likelihood is

$$\begin{aligned}\ell(\mathbf{W}, \mu, \Delta, \sigma^2 | \mathbf{x}_i, \mathbf{t}_i, \tau, \gamma, y_i) = \ln p(\mathbf{t}_i) + \ln p(\mathbf{x}_i | \mathbf{t}_i, \mathbf{W}, \mu, \Delta, \sigma^2) \\ + \ln p(\Delta | \mathbf{T}) + \sum_{j=1}^{p} \ln p(\tau_j | \gamma)\end{aligned} \tag{5.4}$$

The E-step requires the conditional distribution of the unobserved variables given the observed variables and the current values of the parameters. Given the conditioning set, the distribution of τ is independent of the distribution of $\mathbf{t}_i$:

$$q_i(\mathbf{t}_i, \tau) = p(\mathbf{t}_i | \mathbf{x}_i, \mathbf{W}, \mu, \Delta, \sigma^2) \prod_{j=1}^{p} p(\tau_j | \gamma, \Delta) \tag{5.5}$$

where τ_j^2 only appears in the complete data log-likelihood as its inverse in terms that involve the parameters. Therefore, we are only concerned with $\langle 1/\tau_j^2 \rangle$, conditioned on the current values of Δ_j and γ. [182] derive the expression

$$p(1/\tau_j | \Delta, \gamma) = \mathrm{IG}\left(\sqrt{\frac{\gamma^2}{\Delta_j^2}}, \gamma^2 \right) \tag{5.6}$$

where IG is the inverse gamma distribution.

The expected complete data log-likelihood function is formed using the posterior distribution $q_i(\mathbf{t}_i, \tau)$ and the complete log-likelihood function

$$\begin{aligned}E\big[\ell(\mathbf{W}, \mu, \Delta, \sigma^2 | \mathbf{x}_i, \mathbf{t}_i, \tau, \gamma, y_i)\big] = \sum_{i=1}^{n} \iiint q_i(\mathbf{t}_i, \tau) \\ \Bigg(\ln\ p(\mathbf{t}_i) + \ln\ p(\mathbf{x}_i | \mathbf{t}_i, \mathbf{W}, \mu, \Delta, \sigma^2) + \ln\ p(\Delta | \mathbf{T}) \\ + \sum_{j=1}^{p} \ln\ p(\tau_j | \gamma) \Bigg) d\mathbf{t}_i d\mathbf{x}_i^m d\tau\end{aligned} \tag{5.7}$$

Using the factorization of q_i, the definitions of the distributions, and the dropping of terms that do not depend on θ, this equation can be rewritten as

$$\begin{aligned} E\big[\ell(\mathbf{W},\mu,\Delta,\sigma^2|\mathbf{x}_i,\mathbf{t}_i,\tau)\big] \propto \sum_{i=1}^{n} -\frac{1}{2}\ln(|\sigma^2\mathbf{I}_p|) - \frac{1}{2}E_{q_i(\tau)}[\Delta^\top\mathbf{T}^{-1}\Delta] \\ - \frac{1}{2\sigma^2}E_{q_i(\mathbf{t}_i)}[(\mathbf{x}_i-\mu-\Delta-\mathbf{W}\mathbf{t}_i)^\top(\mathbf{x}_i-\mu-\Delta-\mathbf{W}\mathbf{t}_i)] \end{aligned} \tag{5.8}$$

To implement the result, the E-step requires the calculation of the expectations:

$$\langle\mathbf{t}_i\rangle = (\sigma^2\mathbf{I}_a + \mathbf{W}^\top\mathbf{W})^{-1}\mathbf{W}^\top(\mathbf{x}_i-\mu-\Delta) \tag{5.9a}$$

$$\langle\mathbf{t}_i\mathbf{t}_i^\top\rangle = \sigma^2(\sigma^2\mathbf{I}_a + \mathbf{W}^\top\mathbf{W})^{-1} + \langle\mathbf{t}_i\rangle\langle\mathbf{t}_i\rangle^\top \tag{5.9b}$$

$$\left\langle\frac{1}{\tau_j}\right\rangle = \frac{\gamma}{|\Delta_j|} \tag{5.9c}$$

And the M-step requires the update equations:

$$\mu^{\text{new}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i - \Delta - \mathbf{W}\langle\mathbf{t}_i\rangle \tag{5.10a}$$

$$\Delta^{\text{new}} = \mathbf{T}(\sigma^2\mathbf{I}_p + \mathbf{T})^{-1}\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i - \mu - \mathbf{W}\langle\mathbf{t}_i\rangle \tag{5.10b}$$

$$\mathbf{W}^{\text{new}} = \left[\sum_{i=1}^{n}(\mathbf{x}_i-\mu+\Delta)\langle\mathbf{t}_i\rangle^\top\right]\left[\sum_{i=1}^{n}\langle\mathbf{t}_i\mathbf{t}_i^\top\rangle\right]^{-1} \tag{5.10c}$$

$$\begin{aligned} \sigma^{2\ \text{new}} = \frac{1}{dn}\sum_{i=1}^{n}\text{trace}\Big[\mathbf{x}_i\mathbf{x}_i^\top - 2(\mu+\Delta)\mathbf{x}_i^\top + \mathbf{W}\langle\mathbf{t}_i\mathbf{t}_i^\top\rangle\mathbf{W}^\top \\ + 2(\mu+\Delta-\mathbf{x}_i)\langle\mathbf{t}_i\rangle^\top\mathbf{W}^\top + (\mu+\Delta)(\mu+\Delta)^\top\Big] \end{aligned} \tag{5.10d}$$

where

$$\mathbf{T} = \text{diag}(|\Delta_j|/\gamma). \tag{5.11}$$

It should be noted that the natural update equation for Δ^{new} is

$$\Delta^{\text{new}} = (\mathbf{I}_p + \sigma^2\mathbf{T}^{-1})^{-1}\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i - \mu - \mathbf{W}\langle\mathbf{t}_i\rangle \tag{5.12}$$

However, when implementing the update step, $\mathbf{T}^{-1} = \text{diag}(\langle 1/\tau_j \rangle) = \text{diag}(\gamma/|\Delta_j|)$ would have a numerical issue since many elements of Δ are expected to go to zero. To avoid this numerical issue [72, 169], the alternative update equation

$$\Delta^{\text{new}} = \mathbf{T}(\sigma^2 \mathbf{I}_p + \mathbf{T})^{-1} \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i - \mu - \mathbf{W}\langle \mathbf{t}_i \rangle \tag{5.13}$$

is implemented as described above [225].

The algorithm alternates between the E- and M-steps until a convergence criterion is satisfied based on the change in the negative log-likelihood (NLL) of the observed data. The change in NLL is the typical convergence criterion, but is rather expensive to calculate, which may motivate another criterion such as the change in the parameters or a fixed number of steps. Additionally, because the NLL decreases at each step, the NNL could be calculated intermittently to reduce computational cost without risk of moving away from the optimum. Once the algorithm converges, the learned parameters are used to train the classifier.

Cross-validation should be used to select the value of the latent dimension, a, and the parameter governing sparsity, γ. To test for convergence of the algorithm, the observed data negative log-likelihood (NLL) should be monitored. The observed data NLL is

$$\begin{aligned} \ell = \sum_{i=1}^{n} \Bigg[&\frac{|o|}{2} \ln 2\pi + \frac{1}{2} \ln \left| \mathbf{W}^o \mathbf{W}^{o\top} + \sigma^2 \mathbf{I}_{|o|} \right| \\ &+ \frac{1}{2} (\mathbf{x}_i^o - \bar{\mu}^o - \Delta^o)^\top (\mathbf{W}^o \mathbf{W}^{o\top} + \sigma^2 \mathbf{I}_{|o|})^{-1} (\mathbf{x}_i^o - \mu^o - \Delta^o) \Bigg] \\ &+ \frac{p}{2} \ln 2\pi - p \ln \frac{\gamma^2}{2} + \frac{1}{2} \sum_{k \in \mathcal{K}} \sum_{j=1}^{p} \left[\ln \tau_j^k + \frac{\Delta_j^2}{\tau_j} + \gamma^2 \tau_j^k \right] \end{aligned} \tag{5.14}$$

As the algorithm proceeds, many of the elements of $\mathbf{T} = \text{diag}(\tau_j)$ will go to zero, which represents a change in the degrees of freedom that needs to be reflected in the observed data NLL. The value of p should correspond to the length of the non-zero elements along the diagonal of $\mathbf{T}$. Additionally, only the corresponding values of Δ should be used.

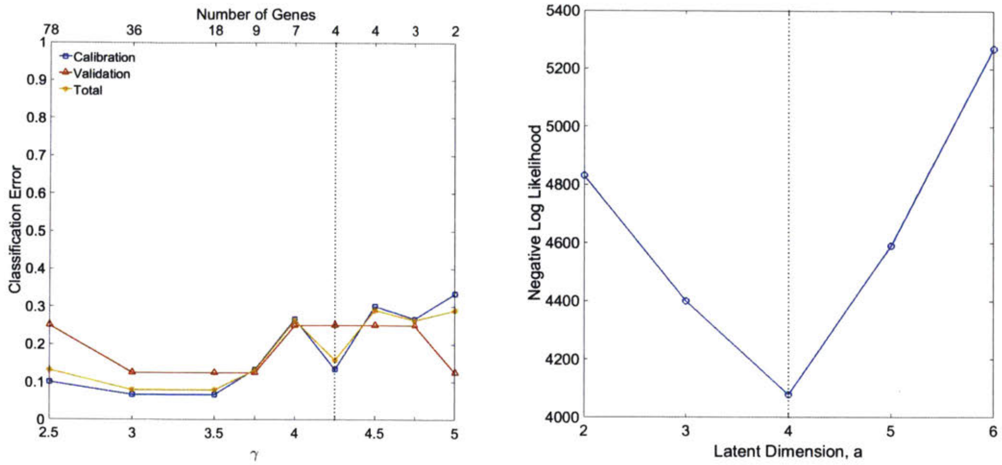


Figure 5-1: The cross-validation plots for the penalty parameter γ and the latent dimension a. The vertical dotted line indicates the selected value for the final model calibration.

To choose the values of the latent dimension a and the sparsity tuning parameter γ, a cross-validation strategy is recommended. The training dataset is partition into two parts: a $1/k$ proportion of the dataset for validation and the remaining data for training. For the presented work, k was selected as 5. The performance on the held-out validation set is used to select the values. Fig. 5-1 shows an example as applied to the Golub *et al.* dataset.

5.2.5 Classification model

LDA models are specified by two parameters, $\mathbf{w} \in \mathbb{R}^k$ and the scalar b, where k is the dimension of the vector of means whose class-specific deviations are non-zero,

$$\mathbf{w} = \hat{\Sigma}^{-1}(\mu_1 - \mu_2), \tag{5.15}$$

$\hat{\Sigma}$ is the marginal covariance of the discriminating variables which can be read from the full covariance matrix, and

$$b = -\frac{1}{2}\mu_1^\top \hat{\Sigma}^{-1} \mu_1 + -\frac{1}{2}\mu_2^\top \hat{\Sigma}^{-1} \mu_2 \tag{5.16}$$

Predictions are then made from

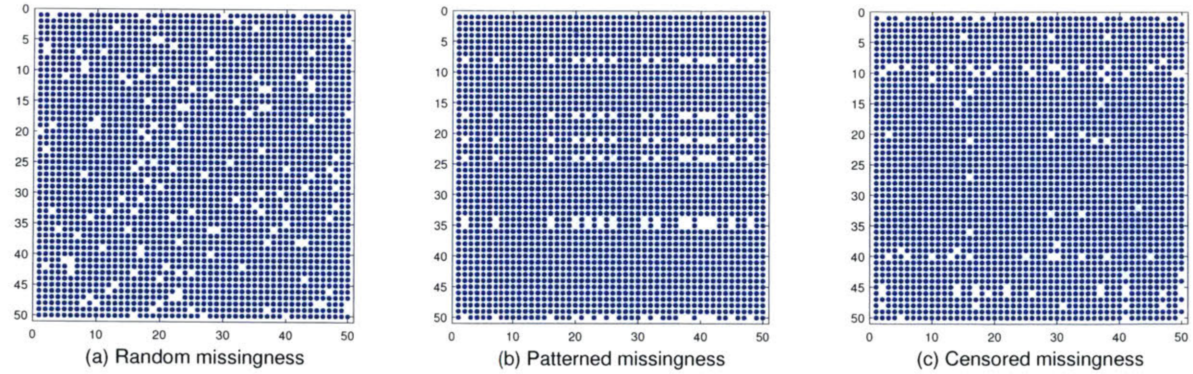
$$\hat{y} = \mathbf{w}^\top \mathbf{x}_i + b. \tag{5.17}$$

A value of $\hat{y}$ greater than 0 indicates class 1, otherwise class 2 is indicated. Its value can be converted back into a probability measure

$$P(y_i = 1 | \mathbf{X} = \mathbf{x}_i) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x}_i - b)} = \frac{1}{1 + \exp(-\hat{y})} \tag{5.18}$$

To decrease the bias of the estimator, the EM procedure can be used for model selection, and the final model is trained without a penalty term. Whether or not this step is possible depends on data availability.

5.2.6 Extension to missing data



(a) Random missingness (b) Patterned missingness (c) Censored missingness

Figure 5-2: Examples of the various types of missingness patterns considered from one of the simulation datasets with 5% missing data.

Missing data are typically described by three categories: missing completely at random (MCAR), missing at random (MAR), and not missing at random (NMAR) [209]. Each of these categories has a precise definition, however robust tests do not exist to determine which mechanism is applicable to a particular scenario and instead auxiliary problem information is used to inform which model applies. Our analysis focuses on the types of missingness that we have observed in practice for high-throughput biological assays.

First, randomly missing measurements throughout the dataset, perhaps due to inappropriate sample handling or image corruption, are considered. No pattern to the missingness is assumed for this case (Fig. 5-2a). Second, missing measurements that are subject to a pattern are considered. Patterned missingness may represent local scratches or, as is sometimes the case in clinical settings, that some patients provided smaller samples, i.e. less volume of blood, and a rank-ordered list of assays are performed until there is no sample remaining (Fig. 5-2b). Finally censoring is considered, where values that meet a certain threshold are missing. An example could be species concentrations that are below a limit of detection (Fig. 5-2c). If censoring is a known issue, other imputation techniques, such as those that generate low or high values based on prior information, may be more appropriate as the validity of the inference is no longer guaranteed [141]. However, the example remains relevant as the analyst may not realize that censoring is occurring.

To account for the introduction of missing data, let $\mathbf{x}_i \in \mathbb{R}^p$ be a vector of measurements for observations $i = 1, \ldots, n$ which may have elements that are missing. Any observation $\mathbf{x}_i$ can be permuted such that $\mathbf{x}_i = [\mathbf{x}_i^o;\ \mathbf{x}_i^m]$. The superscript notation denotes the elements of the ith observation which are missing (m) and observed (o). These elements are a function of the observation, i.e., $m = m(i)$, however this explicit dependence is dropped for simplicity.

The joint distribution is augmented to include these missing elements. To describe the joint distribution of $\mathbf{t}_i$ and $\mathbf{x}_i^m$, a new variable

$$\mathbf{z}_i = \begin{bmatrix} \mathbf{t}_i \\ \mathbf{x}_i^m \end{bmatrix}, \tag{5.19}$$

is defined where $p(\mathbf{z}_i|\mathbf{x}_i^o, \mathbf{W}, \sigma^2, \mu, y_i)$ is a Gaussian distribution described by the information form of the multivariate Gaussian distribution,

$$\Lambda_z = \begin{bmatrix} \mathbf{I}_a + \frac{1}{\sigma^2}\mathbf{W}^\top\mathbf{W} & -\frac{1}{\sigma^2}\mathbf{W}^{m\top} \\ -\frac{1}{\sigma^2}\mathbf{W}^m & \frac{1}{\sigma^2}\mathbf{I}_m \end{bmatrix} \tag{5.20}$$

$$\eta_z = \begin{bmatrix} \frac{1}{\sigma^2}\mathbf{W}^{o\top}(\mathbf{x}_i^o - \mu^o - \Delta^o) - \frac{1}{\sigma^2}\mathbf{W}^{m\top}(\mu^m + \Delta^m) \\ \frac{1}{\sigma^2}(\mu^m + \Delta^m) \end{bmatrix} \tag{5.21}$$

Using these factors, the mean and covariance of the posterior distribution can be defined by

$$\Sigma_z = \Lambda_z^{-1} = \begin{bmatrix} \sigma^2(\sigma^2\mathbf{I}_a + \mathbf{W}^{o\top}\mathbf{W}^o)^{-1} & \sigma^2(\sigma^2\mathbf{I}_a + \mathbf{W}^{o\top}\mathbf{W}^o)^{-1}\mathbf{W}^{m\top} \\ \sigma^2\mathbf{W}^m(\sigma^2\mathbf{I}_a + \mathbf{W}^{o\top}\mathbf{W}^o)^{-1} & \sigma^2(\mathbf{I}_m + \mathbf{W}^m(\sigma^2\mathbf{I}_a + \mathbf{W}^{o\top}\mathbf{W}^o)^{-1}\mathbf{W}^{m\top}) \end{bmatrix} \tag{5.22}$$

$$\mu_z = \Sigma_z\eta_z = \begin{bmatrix} (\sigma^2\mathbf{I}_a + \mathbf{W}^{o\top}\mathbf{W}^o)^{-1}\mathbf{W}^{o\top}(\mathbf{x}_i^o - \mu^o) \\ \mathbf{W}^m\langle\mathbf{t}_i\rangle + \mu^m \end{bmatrix} \tag{5.23}$$

The complete data log-likelihood does not change in this scenario but the E- and M-steps change because of the new distribution with which the expectation is taken with respect to:

$$\langle\mathbf{t}_i\rangle = (\sigma^2\mathbf{I}_a + \mathbf{W}^{o\top}\mathbf{W}^o)^{-1}\mathbf{W}^{o\top}(\mathbf{x}_i^o - \mu^o - \Delta^o) \tag{5.24a}$$

$$\langle\mathbf{t}_i\mathbf{t}_i^\top\rangle = \sigma^2(\sigma^2\mathbf{I}_a + \mathbf{W}^{o\top}\mathbf{W}^o)^{-1} + \langle\mathbf{t}_i\rangle\langle\mathbf{t}_i\rangle^\top \tag{5.24b}$$

$$\left\langle\frac{1}{\tau_j}\right\rangle = \frac{\gamma}{|\Delta_j|} \tag{5.24c}$$

$$\langle\mathbf{x}_i^m\rangle = \mathbf{W}^m\langle\mathbf{t}_i\rangle + \mu^m + \Delta^m \tag{5.24d}$$

$$\begin{aligned}\langle\mathbf{x}_i^m\mathbf{x}_i^{m\top}\rangle &= \sigma^2(\mathbf{I}_m + \mathbf{W}^m(\sigma^2\mathbf{I}_a + \mathbf{W}^{o\top}\mathbf{W}^o)^{-1}\mathbf{W}^{m\top}) \\ &\quad + \langle\mathbf{x}_i^m\rangle\langle\mathbf{x}_i^m\rangle^\top\end{aligned} \tag{5.24e}$$

$$\langle\mathbf{x}_i^m\mathbf{t}_i^\top\rangle = \sigma^2\mathbf{W}^m(\sigma^2\mathbf{I}_a + \mathbf{W}^{o\top}\mathbf{W}^o)^{-1} + \langle\mathbf{x}_i^m\rangle\langle\mathbf{t}_i\rangle^\top \tag{5.24f}$$

In the M-step, the parameters are updated by

$$\mu^{\text{new}} = \frac{1}{n}\sum_{i=1}^{n}\langle\mathbf{x}_i\rangle - \mathbf{W}\langle\mathbf{t}_i\rangle - \Delta \tag{5.25a}$$

$$\Delta^{\text{new}} = \mathbf{T}(\sigma^2\mathbf{I}_p + \mathbf{T})^{-1}\frac{1}{n}\sum_{i=1}^{n}\langle\mathbf{x}_i\rangle - \mathbf{W}\langle\mathbf{t}_i\rangle - \mu \tag{5.25b}$$

$$\mathbf{W}^{\text{new}} = \left[\sum_{i=1}^{n}\langle\mathbf{x}_i\mathbf{t}_i^\top\rangle - (\mu+\Delta)\langle\mathbf{t}_i\rangle^\top\right]\left[\sum_{i=1}^{n}\langle\mathbf{t}_i\mathbf{t}_i^\top\rangle\right]^{-1} \tag{5.25c}$$

$$\begin{aligned}\sigma^{2\text{ new}} = \frac{1}{dn}\sum_{i=1}^{n}\text{trace}\Big[&\langle\mathbf{x}_i\mathbf{x}_i^\top\rangle - 2\langle\mathbf{x}_i\mathbf{t}_i^\top\rangle\mathbf{W}^\top - 2(\mu+\Delta)\langle\mathbf{x}_i\rangle^\top\\ &+ 2(\mu+\Delta)\langle\mathbf{t}_i\rangle^\top\mathbf{W}^\top + \mathbf{W}\langle\mathbf{t}_i\mathbf{t}_i^\top\rangle\mathbf{W}^\top + (\mu+\Delta)(\mu+\Delta)^\top\Big]\end{aligned} \tag{5.25d}$$

Note that $\langle\mathbf{x}_i\rangle$ is a concatenation of the expectations for the missing elements and the observed values. Building the final model follows the same approach as in the full data case. Re-estimation of the parameters may or may not be reasonable in this case, depending on how much data are missing. In the event of missing data in the test case, the generative model can be used to impute the relevant elements.

5.3 Case study

5.3.1 Simulation

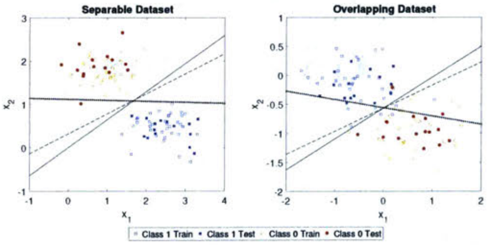


Figure 5-3: The results of the two-dimensional simulation. In both cases, the correct two discriminating variables are discovered by EM-SDA.

EM-SDA is first tested by application to synthetic data. In all cases, the dataset has 100 'experiments' and 2000 'measurements' where half of the experiments are assigned to class 1 and the other half are assigned to class 0. The data are modeled using class-specific means for the discriminating variables and zero means for the remaining data. The data are modeled using a shared covariance with a latent dimension of 5. The error and factor loadings, as described in eqn. 1, are scaled to control class overlap. To determine the covariance matrix, first a matrix of size $p \times a$ is generated where each element is uniformly distributed between $[-1, 1]$. The QR decomposition is applied to orthogonalize the result which is then rescaled. The full covariance matrix is calculated by $\Sigma = \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_p$. The rescaling and value of σ^2 are selected to meet the overlap criteria. This step specifies the class-specific means and covariance. The observed data are then simulated using these parameter is a multivariate normal distribution, conditioned on the class.

In the first application, the true model has a two-dimensional (2D) decision boundary, so that it can be visualized easily. No missing data is added and two cases are considered: separable and overlapping classes. The results of learning the decision boundary using EM-SDA and 5-fold cross validation are shown in Fig. 5-3. In both cases, EM-SDA correctly identifies the two discriminating variables from the 2000 measurements. For the separable case, NSC is unable to differentiate between a 1D and 2D model, whereas EM-SDA always correctly chooses the 2D model. SDA is less successful in finding the true discriminating variables. In the separable case, the model has 3 variables, 1 true and 2 spurious, and in the overlap case, the model has 9 variables, 2 true and 7 spurious. EM-SDA is better able to handle these cases where the measurements are correlated.

The second part of the simulation study focuses on the missingness mechanism (random, patterned, and censored) and level (i.e., percent of data missing). Five datasets were generated and 20 discriminating variables were randomly chosen. Overlap was specified such that the true LDA model would achieve at least 95% accuracy but never 100%, i.e. the data are not separable. Missingness is introduced using random, patterned, and censored assumptions into each of the cases at 5% and 15%. Cases without missing data are also tested. To train the model, 70 of the experiments are used with a 5-fold cross-validation strategy. The remaining 30 experiments are used as held-out test set. In all instances, test error refers to the model error as applied to samples not used during the training phase.

Table 5.1: Results for a simulation study in which EM-SDA is compared to NSC and SDA. In all analyses, the SDA results are generated by running the public code, available at http://www.imm.dtu.dk/projects/spasm/ [236] and the NSC results are generated by running the public R package PAMR [247]. For the missing data cases, the benchmark algorithms are combined with k-nearest neighbors imputation. The table contains the average for the five trials and standard deviation, in parenthesis.

	EM-SDA			NSC			SDA		
	0%	5%	15%	0%	5%	15%	0%	5%	15%
				Test AUC					
Full	0.95 (0.04)			0.80 (0.23)			0.97 (0.01)		
Random		0.94 (0.07)	0.92 (0.08)		0.79 (0.20)	0.79 (0.23)		0.96 (0.03)	0.98 (0.01)
Patterned		0.94 (0.05)	0.97 (0.09)		0.77 (0.22)	0.77 (0.17)		0.95 (0.03)	0.95 (0.04)
Censored		0.97 (0.05)	0.91 (0.08)		0.79 (0.24)	0.77 (0.27)		0.95 (0.04)	0.73 (0.30)
				Number of true dimensions found					
Full	6.8 (1.64)			6.0 (1.58)			5.4 (3.78)		
Random		6.4 (1.52)	5.8 (1.30)		4.6 (1.52)	4.6 (2.61)		5.2 (4.49)	5.4 (1.95)
Patterned		6.8 (1.10)	6.0 (2.55)		5.4 (2.88)	4.6 (1.82)		5.6 (2.70)	5.2 (3.96)
Censored		5.8 (1.64)	5.0 (1.73)		3.6 (2.30)	4.2 (2.49)		5.4 (2.79)	2.4 (0.89)
				Number of false dimensions found					
Full	2.6 (3.78)			1.6 (2.07)			4.2 (3.03)		
Random		2.2 (2.59)	2.8 (1.79)		0.8 (1.10)	2.8 (5.72)		2.0 (3.03)	6.0 (5.15)
Patterned		2.2 (3.35)	3.6 (2.88)		2.0 (2.35)	0.6 (0.89)		2.6 (1.82)	2.4 (3.36)
Censored		4.2 (4.49)	4.2 (4.76)		0.4 (0.55)	5.8 (6.26)		6.0 (5.79)	9.6 (8.96)

Table 5.1 compares the results to the NSC and SDA approaches combined with k-nearest neighbors (KNN) imputation. The area under the receiver operator curve (AUC) for the test data, the number of true discriminant variables, and the number of false discriminant variables selected by the model were chosen as the appropriate evaluation metrics. The best scores possible are 1, 20, and 0, respectively. In nearly all cases, the NSC method had the lowest AUC, the lowest number of true dimensions, and the fewest false dimensions. EM-SDA and SDA had similar AUC results, with EM-SDA having significantly better performance for censored data with high proportion of missing data. EM-SDA also found fewer spurious predictors than SDA in most cases.

5.3.2 Applications

To assess performance on real data, EM-SDA is applied to three publicly available biomedical datasets. The first, [85], is a landmark study classifying two types of leukemia, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) using microarray-based gene expression data. The dataset has 72 samples, which are pre-assigned as train (38 samples, 27 ALL) and test (34 samples, 20 ALL), and 7129 measurements per sample. The preprocessing methodology described by [63] was followed. The second dataset, Ramilo et al. [199], also utilizes gene expression microarrays, but for a different application: classifying patients with acute infections of different pathogens, specifically *E. coli* infection and *S. aureus* infection. The dataset contains 59 samples, each with 211 measurements. The data was split into training (20 samples, 10 *E. coli*) and testing (39 samples, 18 *E. coli*) and the preprocessing methodology described by [199] was followed. Finally, a third dataset, Higuera et al. [94] was chosen because it uses a different technology and reports missing data, unlike the first two. This dataset classifies rescued and failed learning in trisomic mice based on protein expression levels from reverse phase protein arrays. The dataset has 240 samples each with 77 protein measurements. The dataset was split into training (120 samples, 67 rescued learning) and testing (120 samples, 68 rescued learning) and the preprocessing methodology described by [94] was followed.

For the gene microarray datasets, missing data were artificially introduced. [253] cite many possible reasons for missing data in microarrays such as insufficient resolution, image corruption, or scratches and dust on the slide. All of the presented missingness mechanisms could be applicable to microarray datasets, and therefore all three were tested. The protein expression dataset has missing data due to technical artifacts [94]. 2.4% of the data is missing; however, of the 77 protein measurements, only 9 have missing data and therefore the dataset follows the patterned assumption and only the patterned mechanism was tested.

To fit the models, both the latent dimension and the value of the regularization parameter γ must be chosen. A 5-fold cross-validation strategy was used to determine the values for these hyper-parameters. The values were chosen by considering the negative log-likelihood of the validation set, the dimension of the final model, and the prediction error. Here, a strong preference is given towards sparsity. The Supplemental Information provides additional details on the cross validation procedure.

To compare with EM-SDA, both imputation and classification algorithms must be chosen. As in the simulation study, NSC and SDA were selected as the classification algorithms. Using the results of [27] which surveys the imputation literature for microarray data, the imputation benchmarking algorithms were chosen as KNN, Bayesian PCA (BPCA) [176], and local least squares (LLS) [121]. Mean imputation is also included as a baseline technique. For the majority of cases considered, complete case analysis is not reasonable and is not presented here.

Table 5.2: Results for the non-missing (full) cases for the leukemia and infection problems. The results of [85], [247], and [199] are directly from their publications.

Application	Method	Train Error	Test Error	Number of genes
Leukemia	[85]	3/38	4/34	50
	NSC	1/38	2/34	21
	SDA	0/38	2/34	5
	EM-SDA	1/38	1/34	4
Infection	[199]	1/20	6/39	30
	NSC	1/20	6/39	26
	SDA	0/20	11/39	9
	EM-SDA	0/20	5/39	7

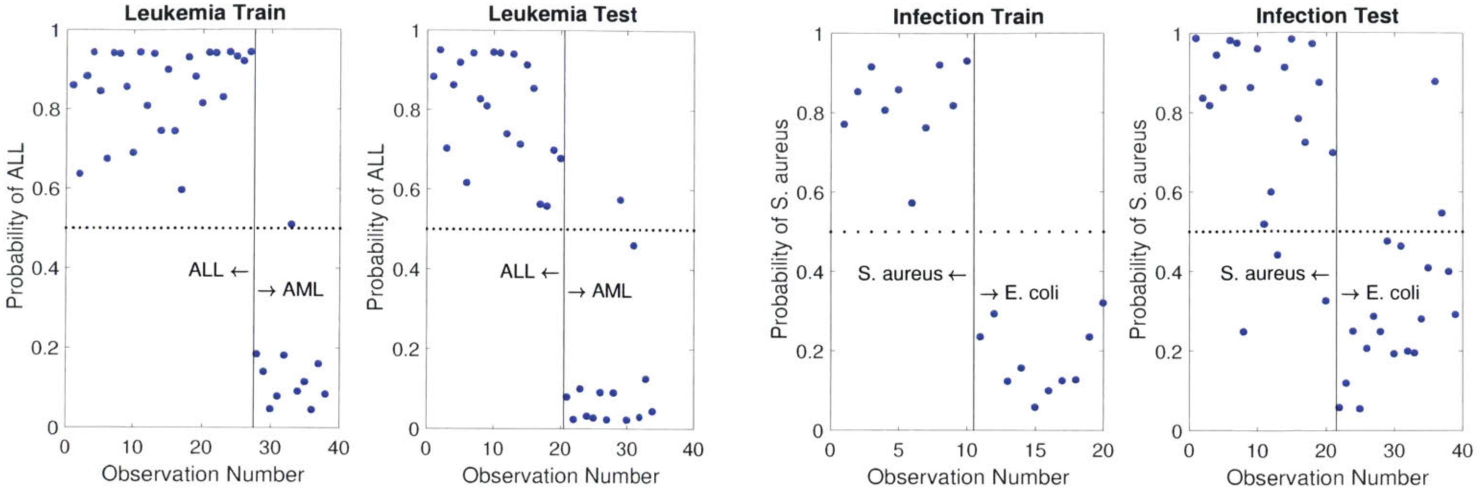


Figure 5-4: The EM-SDA model predictions for the full dataset cases. The dotted line shows the decision boundary at 50%. In the leukemia problem, there is one misclassified point in each of the training and testing datasets. In the infection problem, there are zero and five misclassified points in the training and testing datasets, respectively.

Table 5.3: Results for the trisomic mice classification with patterned missingness [94]. The 2.4% case is the original dataset and the 10% case demonstrates the effect of additional missing data. NSC and SDA each use KNN as the imputation technique.

	Patterned 2.4%			Patterned 10%		
Method	Train Error	Test Error	Number of proteins	Train Error	Test Error	Number of proteins
NSC	35/120	31/120	15	31/120	31/120	8
SDA	5/120	4/120	9	11/120	8/120	9
EM-SDA	2/120	3/120	14	6/120	5/120	12

The model is then applied to the test data. The results for the full datasets are shown in Fig. 5-4 and compared to the originally proposed model and the NSC and SDA approaches in Table 5.2.

For both leukemia and infection, EM-SDA improved the classification accuracy on the test data while using a smaller subset of genes.

The results for the missing data are shown in Tables 5.3, 5.4, and 5.5. EM-SDA provided about a factor of ten and a factor of two reduction in the sum of training and testing errors for patterned missingness for trisomic mice, compared to NSC and SDA, respectively (Table 5.3). EM-SDA also outperformed the other methods for patterned missingness for leukemia (Table 5.4). For the other missingness patterns,

Table 5.4: Results for the four missing data cases as compared to benchmark approaches for sparse classification for the leukemia classification problem [85].

	Random 1.5%			Random 15%			Patterned 18%			Censored 20%		
Method	Train Error	Test Error	Number of genes	Train Error	Test Error	Number of genes	Train Error	Test Error	Number of genes	Train Error	Test Error	Number of genes
NSC MI	2/38	3/34	7	1/38	3/34	6	2/38	2/34	4	2/38	6/34	9
NSC KNN	1/38	3/34	7	1/38	3/34	5	2/38	2/34	4	3/38	5/34	9
NSC LLS	2/38	2/34	7	4/38	3/34	6	1/38	2/34	5	11/38	14/34	11
NSC BPCA	1/38	2/34	8	1/38	1/34	8	2/38	5/34	5	1/38	6/34	11
SDA MI	0/38	2/34	7	0/38	4/34	7	0/38	2/34	4	19/38	21/34	5
SDA KNN	0/38	2/34	5	0/38	2/34	9	0/38	8/34	5	19/38	14/34	8
SDA LLS	0/38	2/34	7	2/38	2/34	2	2/38	2/34	2	11/38	14/34	9
SDA BPCA	0/38	2/34	6	0/38	2/34	3	0/38	7/34	4	1/38	9/34	3
EM-SDA	0/38	2/34	7	2/38	4/34	5	1/38	1/34	4	4/34	6/34	4

EM-SDA performed similarly to the best of the other methods, often with fewer genes. In the censored case, some methods such as LLS fail to generate reasonable imputations and cannot be used in the modeling phase.

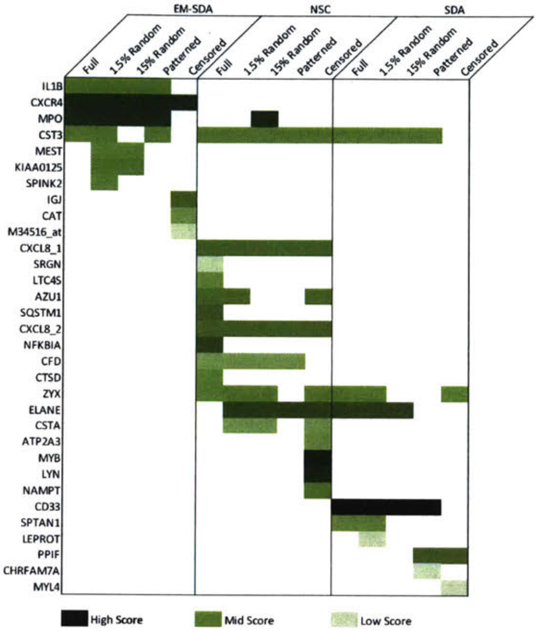


Figure 5-5: Overlap and biological significance of the genes that are selected for the various leukemia classification cases. NSC and SDA are combined with BPCA for imputation. Shaded cells indicate that a particular gene was selected and the intensity of the cell represents the leukemia-relevant score based on an independent literature review.

In addition to prediction accuracy, consistency and biological relevance are important to consider. Consistency is defined as the amount of gene overlap between the missing and non-missing cases for a given method. Specifically, biological significance of each classifier was assessed by a score derived from Pubmed search results. The score is the number of results for "gene/protein name" + "problem domain" squared divided by the total number of results for the gene/protein. The problem domain terms are: leukemia, infection, and Down syndrome/memantine/cognitive where "/" refers to OR statements. The score is then log-scaled. A "–" indicates that there were no results for that gene/protein.

Fig. 5-5 shows the genes that are selected and a relevance metric for the leukemia classification in EM-SDA, SDA with BPCA, and NSC with BPCA. Generally, SDA

Table 5.5: Results for the four missing data cases as compared to benchmark approaches for sparse classification for the infection classification problem [199].

	Random 1.5%			Random 15%			Patterned 15%			Censored 11%		
Method	Train Error	Test Error	Number of genes	Train Error	Test Error	Number of genes	Train Error	Test Error	Number of genes	Train Error	Test Error	Number of genes
NSC MI	2/20	14/39	2	1/20	9/39	5	1/20	7/39	25	1/20	10/39	13
NSC KNN	2/20	13/39	2	1/20	11/39	3	2/20	15/39	1	1/20	21/39	1
NSC LLS	2/20	12/39	3	1/20	10/39	7	1/20	7/39	24	10/20	18/39	3
NSC BPCA	2/20	15/39	1	1/20	8/39	9	1/20	10/39	15	1/20	8/39	34
SDA MI	0/20	12/39	12	1/20	10/39	12	0/20	14/39	17	0/20	11/39	13
SDA KNN	1/20	14/39	5	1/20	12/39	13	0/20	12/39	15	1/20	13/39	9
SDA LLS	0/20	11/39	12	0/20	9/39	13	2/20	16/39	6	–	–	–
SDA BPCA	1/20	11/39	12	0/20	13/39	5	0/20	13/39	12	0/20	21/39	11
EM-SDA	1/20	7/39	5	2/20	8/39	4	0/20	8/39	5	1/20	10/39	5

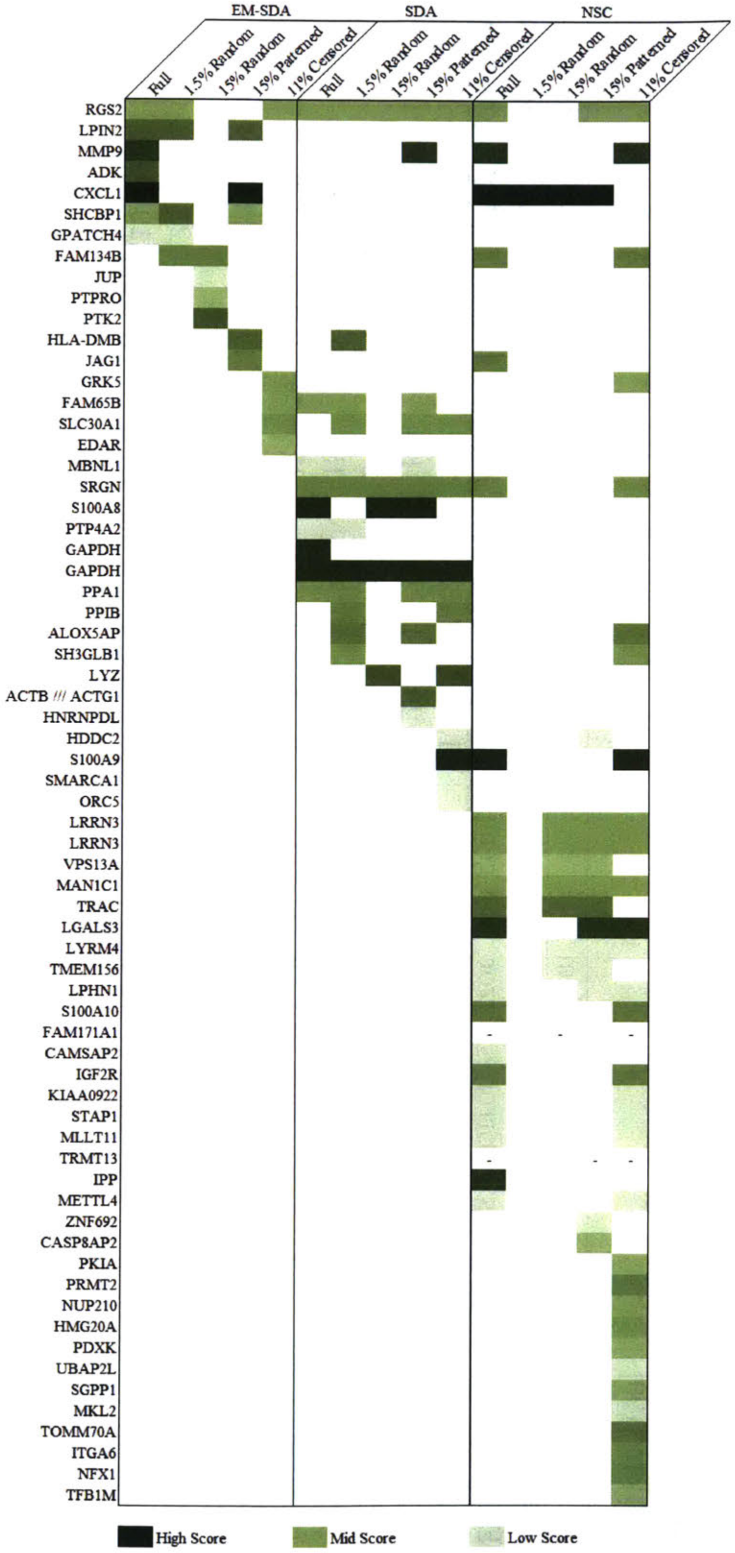


Figure 5-6: Genes that were selected for the various infection cases. NSC and SDA are combined with BPCA for imputation. Shaded cells indicate that a particular gene was selected and the intensity of the cell represents the infection-relevant score based on an independent literature review. The dashes represent genes that were selected but were not found during the literature review.

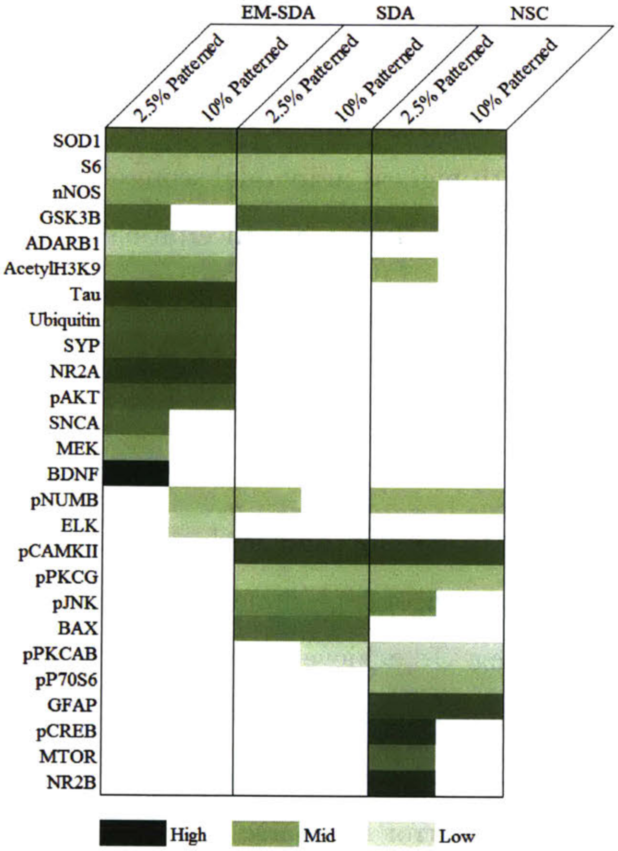


Figure 5-7: Proteins that were selected for the various trisomic mice cases. NSC and SDA are combined with KNN for imputation. Shaded cells indicate that a particular protein was selected and the intensity of the cell represents the trisomy-relevant score based on an independent literature review.

has the most trouble with consistency, although the results are improved when a more advanced imputation technique (BPCA) is used. NSC and EM-SDA have similar success for consistency in the random and patterned cases. For the censored case, the problem is much more challenging. NSC and KNN performs well for the leukemia dataset but fails for the infection dataset. In both censored cases, EM-SDA does well in terms of classification error but does not recover the same set of genes. EM-SDA identifies two genes of high biological relevance – CXCR4 and MPO – for nearly all levels and types of missingness that were missed by SDA and only identified by NSC in one case. EM-SDA also had the highest average score for biological relevance, but did not find the gene with highest individual score of all identified genes, CD33. Similar figures for the infection and trisomic mice problems are in Fig. 5-6 and 5-7, respectively.

5.4 Discussion

The goal of this study was to develop and evaluate a new method for simultaneous imputation and classification of high-dimensional, correlated data where some measurements may be missing. To achieve these goals, an expectation maximization framework was adopted. The resulting methodology, EM-SDA, was tested using both synthetic and real data for varying levels and mechanisms of missing data. EM-SDA demonstrated low classification error for sparse models in all settings and was shown to be particularly successful when the missingness is patterned.

Compared to the other methods, one advantage of EM-SDA is its ability to handle missing data. Another advantage seen in the case study is that EM-SDA found nearly the same models as if data were not missing. Its use of the structured covariance approximation avoids the nonphysical assumption that different measurements are independent. Because the model is generative, it can also be used when test cases have missing elements by imputing the maximum likelihood estimate. A limitation of EM-SDA is its computational cost. For the case where no data are missing, the computational cost per iteration is $\mathcal{O}(np^2)$ and the memory storage is $\mathcal{O}(na^2)$. When data are missing, the computational cost per iteration is $\mathcal{O}(n\tilde{m}^2a)$ and the memory storage is $\mathcal{O}(np^2)$ where $\tilde{m}$ is the maximum number of elements that are missing for any sample. The increase in memory for the missing data case is due to the need to store the expected value of the outer product of the missing data. Expectation maximization is known to be slow to converge. A possible way to speed up convergence would be to use adaptive overrelaxed EM (AEM) [215]. As the fraction of missing data increases, EM is known to take smaller steps, in which case AEM can lead to large speedups [215].

EM-SDA has been demonstrated to be successful for all of the types of missingness studied. EM-SDA is particularly recommended when the missingness is patterned or if missingness is likely to occur in test samples. EM-SDA is well suited to wide, correlated biological datasets, such as microarray data, RNA-Seq data, patient metadata, and proteomic data. As more of these datasets are generated and subjected to rigorous statistical analyses, new models that can both systematically handle missing data and yield simple, interpretable, and accurate results will become increasingly valuable.

Chapter 6

Survey of Industrial Process Monitoring

This work originally appeared as: Kristen Severson, Paphonwit Chaiwatanodom and Richard D. Braatz. Perspectives on process monitoring of industrial systems. Annual Reviews in Control, 42:190-200, 2016.

6.1 Introduction

Process monitoring is an important component in the long-term reliable operation of any automated controlled system. To distinguish between different types of disruptions on operations, this chapter adopts the definitions of [109]. A **disturbance** is an unknown and uncontrolled input acting on a system. A **fault** is an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard operating conditions. A **failure** is a permanent interruption of a system's ability to perform a required function under specified operating conditions. Traditional control systems are designed to return the system to normal operations in the presence of disturbances but not in the presence of faults or failures. *Fault-tolerant control* (FTC) systems refer to control systems that have been designed to explicitly account for some class of specified faults in the closed-loop system. FTC systems must act in the time between a fault and a system failure.

In chemical systems, a fault is an extreme event such as catalyst deactivation, valve blockage or compressor failure. Due to the increasing complexity of facilities, faults are inevitable and occur more often. Monitoring is complicated by recycle streams that cause bidirectional interactions as well as by control systems which can mask the effect of faults. Additionally faults will commonly occur together, known as multiple faults (see Fig. 6-1). However, even a relatively simple modern facility, in terms of its operations, will have a large sensor network which can be used for process monitoring (see Fig. 6-2). The key of fault detection and diagnosis (FDD) is how to use these sensors effectively to minimize the impact of faults.

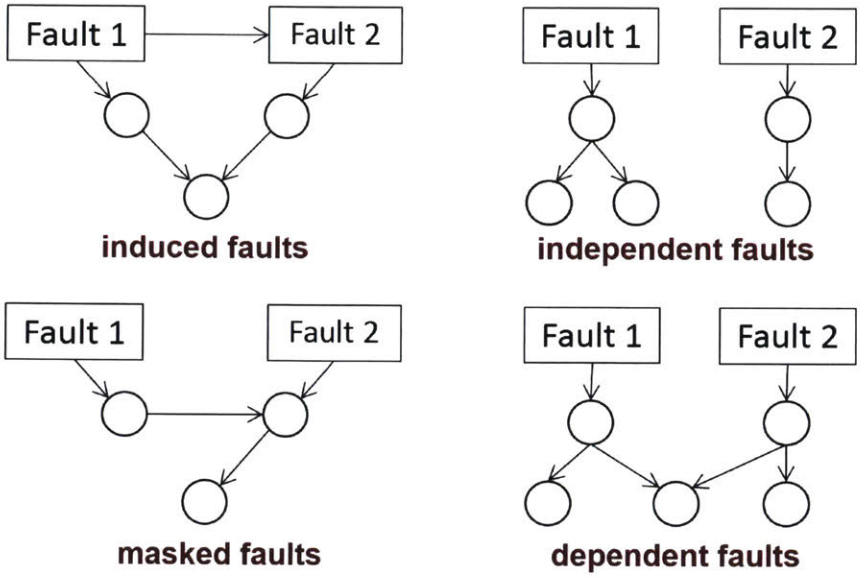


Figure 6-1: The four classes of multiple faults [45].

Many process monitoring systems are implemented in the form of a loop that consists of fault detection, fault isolation, fault identification, and process recovery (see Fig. 6-3). Sometimes the combined steps of fault isolation and identification are referred to as fault diagnosis. The steps are to progressively determine: (1) whether a fault occurred, (2) the location and time of the fault, (3) the magnitude the fault, and (4) how to reverse the effects of the fault [83].

Process monitoring has been a growing field for nearly a half century. Relevant works on process monitoring in the 1970s include the application by [159] of systems

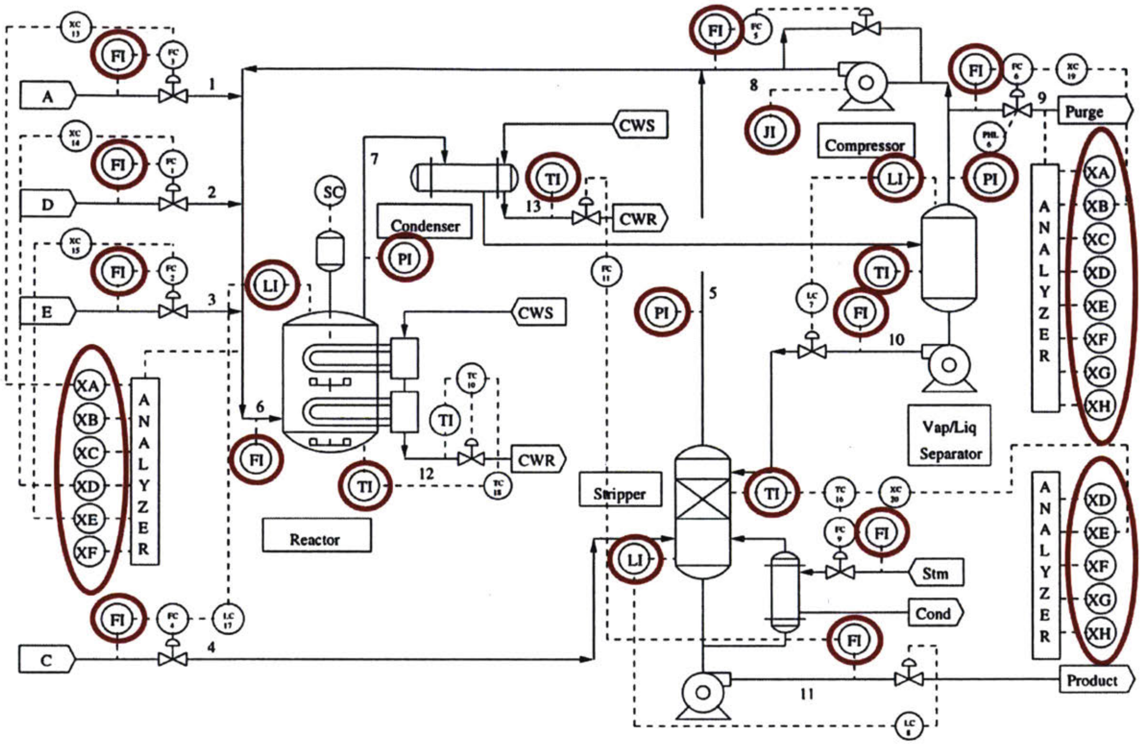


Figure 6-2: The process diagram for the Tennessee Eastman (TE) benchmark problem [58]. The process is a reactor/separator/recycle with two simultaneous gas-liquid exothermic reactions. The process has 12 valves for manipulation and 41 measurements for monitoring and control. The sensors are circled in red.

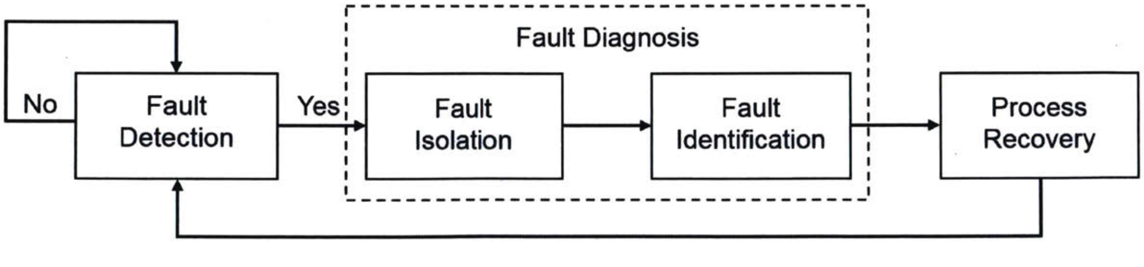


Figure 6-3: Process monitoring loop [109, 211].

and statistical decision theory to dynamic systems, the review paper by [280] on publications up to the mid 1970s, and the textbook by [95]. Over the years, much of the literature has been focused on particular applications including to aerospace, chemical, nuclear, and automotive systems [101]. The growing complexity and degree of integration in these systems has increased the possibility that faults occurring locally somewhere in a system can have their effects propagate to other parts of the system, and has made the consequences of designing a poor process monitoring system greater, therefore making the design of process monitoring systems more challenging. As such, many reviews have been published over the last twenty years, e.g. [4, 74, 101, 109, 108, 193, 211, 258, 259, 260, 290].

This article does not review the entire process monitoring field which, according to the Web of Science in March 2015, has had over 34,000 publications since the 1970s. This article provides some perspectives on the current state of process monitoring systems as well as current challenges and promising future directions for the field.

6.2 Process monitoring – background

Modern process monitoring systems are designed based on a model of some form that is developed using process data. The model allows process operators to make informed decisions about whether or not there is a fault. Different fault detection methods provide information of different quality and quantity to the fault diagnosis steps. In this section, each step in the process monitoring loop is presented.

6.2.1 Fault detection

The design of a fault detection system generally begins with the development of a model that characterizes the normal operating signature of a process. Faults are then typically defined as a deviation from this normal operation above a threshold. As such, the design of a fault detection system can be described as consisting of two steps: building a process model and choosing metrics to test for faults. Active fault detection and identification is an exception to this pattern and is discussed later in the section on process monitoring.

Many types of process models have been employed in fault detection. Principal component analysis (PCA) is one of the most commonly applied fault detection methods for industrial systems. PCA is a linear dimensionality reduction technique that produces lower dimensional representations of the original data that maximize the retained variance [99, 118]. In the absence of noise and disturbances, data from normal operating conditions operate in a much lower dimensional manifold due to physical, chemical, and biological constraints such as Euler's laws of motion, stoichiometry in chemical and/or metabolic reaction networks, and mass, energy, molar species, and fluid momentum balances. In the presence of noise and disturbances, the data from normal operating conditions will approximately lie within a lower dimensional manifold, and data-based dimensionality reduction techniques such as PCA attempt to construct the manifold purely from data.

Variance is a useful metric for fault detection, since it is often reasonable to assume that an outlier as compared to historical operation would indicate a fault. PCA calculates a set of orthogonal vectors, called *loading vectors*, ordered by the amount of variance explained in each loading vector direction using a singular value decomposition. This set of vectors is then truncated, retaining the columns corresponding to the largest singular values. New observations can then be projected into lower dimensional space using the reduced set of loading vectors. The aim of this dimensionality decrease is to keep systematic variations while removing random variations [281]. The technique can be extended to nonlinear systems by using kernel functions within the PCA formulation [48]. PCA has been applied in a variety of fields including (bio)pharmaceutical manufacturing [89, 123, 126, 255], the chemicals industry [174, 301], and semiconductor manufacturing [43].

Partial least squares (PLS, aka projection to latent structures) is another linear dimensionality reduction technique [286] widely applied for fault detection in industrial systems. PLS maximizes the covariance between the input and output data in the reduced space [81]. Unlike PCA, PLS does not have a closed-form solution but instead uses an iterative algorithm such as NIPALS [284]. PLS is widely applied in the chemicals, petrochemicals, and refining industries [211, 300] and in pharmaceutical and biologic drug manufacturing [123, 228]. The low cost of entry of chemometrics (PCA, PLS) methods and the lack of dynamic models for most plant operations are the main reasons for their dominance in these industries. Both their current heavy usage and the ever-increasing quantity of real-time data [205] suggests that chemometrics methods will continue to dominate those industries for the foreseeable future.

An alternative to fault detection methods that rely on dimensionality reduction are methods based on state-space models. The most commonly used model is the discrete-time linear stochastic state-space model

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + G\mathbf{u}_k + \mathbf{w}_k \tag{6.1}$$

$$\mathbf{y}_k = H\mathbf{x}_k + A\mathbf{u}_k + B\mathbf{w}_k + \mathbf{e}_k \tag{6.2}$$

where k is the sampling index; $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{y}$ are the system states, inputs, and outputs, respectively; and $\mathbf{w}$ and $\mathbf{e}$ denote the sensor and process noise of the system [130, 235]. Such models are typically constructed from subspace identification techniques, such as canonical variate analysis (CVA) [131], multivariable output-error state-space (MOESP) [261, 262, 263, 264, 265], and numerical algorithm subspace-based state-space system identification (N4SID) [257]. The subspace identification techniques most applied to industrial systems is CVA, which was pioneered by Akaike [2] and promoted and further developed by Larimore [131]. The objective of CVA is to identify a linear combination of past inputs and outputs that are most predictive of future outputs. CVA relies on minimizing the prediction error using a singular value decomposition of the covariance matrix for past inputs and outputs. CVA has been reported to produce near maximum-likelihood solutions [119]. Another type of identification technique uses fuzzy rule-based models. In this approach, fuzzy clustering techniques are used to partition the data into linear subsets [251]. This approach was originally proposed for modeling and control and then extended to fault diagnosis [233].

Another class of fault detection models relies on graphical models, which are typ-

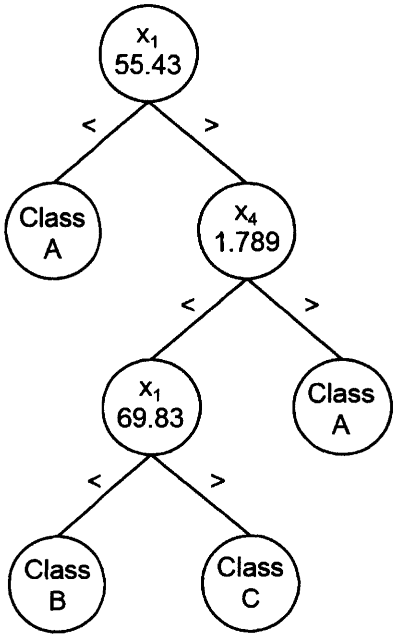


Figure 6-4: An example of a decision tree as applied to a process for maintaining octane number of a gasoline product adapted from [12].

ically directed and often lumped into the broader class of knowledge-based methods. These methods employ some form of expert knowledge in their construction. A decision tree is a type of graphical model developed via inductive learning that aims to map measured data to classes of operating conditions. These models are able to describe normal and abnormal operations during complicated startup, shutdown, and changeover procedures (such systems are often called *mixed continuous-discrete systems* or *hybrid systems*). Feature selection and extraction are important considerations for the success of decision trees and are facilitated by process understanding. A benefit of this approach is that a well-developed graphical model has an easily interpreted physical meaning (e.g., see Fig. 6-4), and that the same model can be used in fault identification and diagnosis [12].

Several other types of graphical models have also been applied in the field, with representative examples being causal maps, Petri nets, bond graphs, and neural networks. A *causal map* is a directed graph where the nodes represent process variables and the directed edges represent cause-and-effect relationships [47]. A model of this type has a clear physical interpretation, and can be constructed from a piping and instrumentation diagram or process flow diagram embedded in the distributed control system. A *Petri net* is a graphical model that is suitable for modeling transitions/events that may occur in the operation of the system and is most well-suited to graphs that have parallel or concurrent events [168]. The graph consists of transitions, places, and arcs in which nodes can be transitions or places (marked with different symbols, typically bars and circles) and arcs connect nodes of different type only. Petri nets were first introduced by [189] and conferences and several tutorials helped popularize the technique [168]. [269] was one of the first researchers to apply Petri nets to fault detection applications, which was followed by a large number of studies (e.g., see [26, 32, 240] and citations therein). Rather than focus on transitions, *a bond graph* is a graphical representation of a physical dynamical system that represents its energy flows [25]. Bond graphs were first introduced by [187], and examples of the application of bond graphs to fault detection include [71, 147]. A *neural network* is a graphical model that is characterized by input, output and hidden nodes. When applied to fault diagnosis problems, often the input nodes represent the measurement space, the hidden nodes represent the feature space, and the output nodes represent the decision space [258]. Examples of the application of neural networks can be found in [124, 167, 166].

Once the model has been determined, a metric for detecting faults is required. In PCA, PLS, and related models, faults are usually detected using the T^2 statistic, which is the Euclidean norm of the deviation of an observation vector from its mean in the reduced space, scaled by its variance. A fault is detected when the T^2 value exceeds a specified threshold. Alternatively, the Q statistic (also known as standard prediction error or SPE), which measures the total sum of variations in the residual space, can also be used to identify faults. In extensive simulations, the Q statistic has been observed to be usually more effective at detecting faults than the T^2 statistic [47]. The explanation for this observation is that most faults push the process operations outside of the normal linear relationships between variables rather than magnify the extent of operation within the normal linear relationships between variables. Some researchers have used a weighted combination of Q and T^2 statistic [294].

For knowledge-based models, faults are detected if the measured variables result in a prediction of a fault, based on the model.

If a state-space model is used, fault detection typically occurs via a similar residual generation, which compares model predictions and measurements (often referred to as *output estimation* approaches). Alternatively, the difference between nominal and estimated parameters has been used to detect faults (often referred to as *parameter estimation* approaches). In the particular case of CVA, a series of different statistics have been proposed for fault detection [44, 120]. In state-space models, fault detection and diagnosis are closely coupled, as discussed below.

6.2.2 Fault isolation

Once the fault has been detected, the next step is to determine the location of the fault. One fault isolation method widely used in industrial systems is the contribution chart. Contribution charts are typically used in concert with dimensionality reduction techniques, such as PCA and PLS. The contribution chart projects the data back into the higher dimensional observation space, which can be used by an operator to identify which process variables are deviating from their historical values. This approach exploits correlations between variables to reduce the effects of process and sensor noise on identifying which observation variables are most likely associated with the fault.

As an example, a contribution chart of the TE process is shown in Fig. 6-5, both in the form of a classic contribution chart at one time instance and in the form of a 2D contribution map with the contributions in each column as a function of time in the form of a color map. The 2D contribution map, introduced by [301], allows the operator to visualize the dynamic propagation of the effects of a fault on the observation variables through the facility. The 2D plot shows which deviations are suppressed by the various control systems and which deviations are persistent. The variables which have deviations can be compared to the process flow diagram or piping and instrumentation diagram to better track down the location of the root cause of the fault. The ability to visualize the data so that the fault can be located is a crucial element for the success of a method. Methods that allow for easy interpretation of the data are much more valuable in industrial application.

An alternative to the classic contributions chart, referred to as *reconstruction-based contribution chart*, has been proposed [3]. This method finds the contribution of each monitored variable to the fault detection metric, for example T^2. An example is given where the reconstruction-based contribution chart provides an accurate fault isolation while the classic contribution chart cannot.

6.2.3 Fault identification

Fault identification can be very challenging if fault detection and isolation have been carried out using PCA or PLS, as the quality of information that can be extracted from models constructed from normal operating data is limited. If the training data has been characterized from past experience into normal operating conditions and specific faulty conditions, then Fisher discriminant analysis (FDA) is a dimensionality reduction method that can be used for fault identification.

FDA maximizes the separation (aka scatter) among different classes while minimizing the scatter within each class [62]. The formulation of the problem is

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}\neq 0} \frac{\mathbf{v}^\top S_b \mathbf{v}}{\mathbf{v}^\top S_w \mathbf{v}} \tag{6.3}$$

where

$$S_b = \sum_{j=1}^{p} n_j (\overline{\mathbf{x}}_j - \overline{\mathbf{x}})(\overline{\mathbf{x}}_j - \overline{\mathbf{x}})^\top, \tag{6.4}$$

$$S_w = \sum_{j=1}^{p} \sum_{\mathbf{x}_i \in \mathcal{X}_j} (\mathbf{x}_i - \overline{\mathbf{x}}_j)(\mathbf{x}_i - \overline{\mathbf{x}}_j)^\top, \tag{6.5}$$

$\mathbf{x} \in \mathcal{R}^m$, $\overline{\mathbf{x}}$ is the total mean vector, $\overline{\mathbf{x}}_j$ is the mean vector for class j, n_j is the number of observations in class j, and p is the number of classes, and sufficient data have been collected that the matrix S_w is nonsingular. For FDA to be well-defined, at least two sets of characterized data are required (e.g., normal operating conditions and data

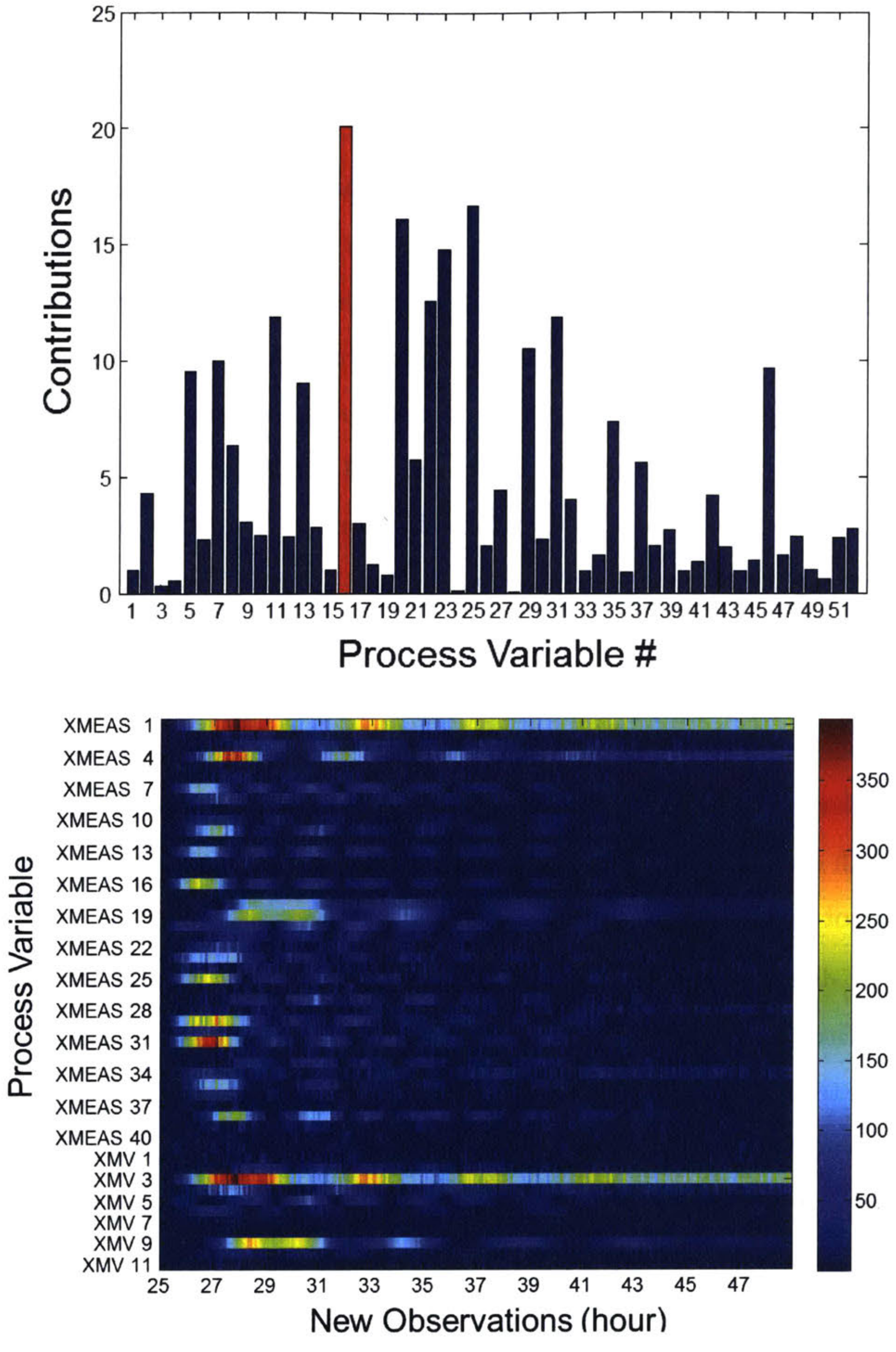


Figure 6-5: Classic contribution chart (top) and 2D contribution map (bottom). The contributions are for a fault in a simulated chemical manufacturing facility [301].

collected during one fault).

FDA models can be more specific about *which* fault is occurring, if they have been trained using multiple fault classes and that set is comprehensive. Often the amount of faulty data is limited in practice and each fault will require its own investigation once the fault isolation step using the contribution chart is complete.

Another data-based fault diagnosis technique that attempts to reduce the dimensionality of the problem is support vector machines (SVMs). SVM methods find a separating hyperplane which is specified by a number of support vectors (samples). These support vectors typically represent a small subset of the complete dataset used for analysis. The separating hyperplane is oriented in such a way as to maximize the distance, called the *margin*, between the plane and the nearest point of each class [16]. SVMs can be formulated using a kernel, which is amenable to feature selection. This technique has gained increased interest in the past 15 years due to efficient optimization formulations [192]. Since then, SVMs have been tested in mechanical engineering applications [279, 9] and semiconductor manufacturing [154]. Like FDA, SVM is typically trained with labeled target data and therefore requires data that have been collected during past faults and, for best results, specific faults must be associated with each data set. Both FDA and SVMs are ineffective for fault diagnosis if data have not been collected during past fault states.

Most fault diagnosis methods based on state-space models assume that the fault is either *additive* or *multiplicative* [41, 125]. An additive fault is assumed to be well represented in terms of a vector added to the fault-free state-space equations, whereas a multiplicative fault is assumed to be well represented by a deviation in a parameter in the state-space matrices; for this reason, multiplicative faults are also commonly referred to as *parametric faults.*

Multiplicative faults can be diagnosed by determining which online parameter estimates have the largest deviations from nominal values. This method is sufficiently general to be applicable to nonlinear dynamical systems. A weakness of this method is that it requires that the data are sufficiently rich in information to be able to accurately estimate parameters online. Fault diagnosis occurs via the link between the parameters and the physical system. If the model parameters are not tied to physical parameters, diagnosis abilities are limited. In particular, deviations in the elements of state-space models constructed from subspace identification methods are not tied to any physical parameters, so this approach provides little value for such models.

Observer-based methods are most commonly used for diagnosing additive faults. In observer-based methods, the residuals between estimated and measured outputs are used for detection and diagnosis. As an example, consider the full-order state estimator

$$\hat{\mathbf{x}}_{k+1} = A\hat{\mathbf{x}}_k + B\mathbf{u}_k + H(\mathbf{y}_k - \hat{\mathbf{y}}_k) \tag{6.6}$$

$$\hat{\mathbf{y}}_k = C\hat{\mathbf{x}}_k \tag{6.7}$$

where $\hat{\mathbf{x}}$ is the predicted state, $\hat{\mathbf{y}}$ is the predicted output, $\mathbf{y}$ is the measured output, and the observer gain H is chosen to satisfy design criteria such as stability, fault sensitivity, and robustness. For a linear process with additive faults, the residuals are

$$\begin{aligned}\Delta\mathbf{x}_{k+1} &= (A - HC)\Delta\mathbf{x}_k + (B_f - HD_f)\mathbf{f}_k \\ &+ (B_d - HD_d)\mathbf{d}_k\end{aligned} \tag{6.8}$$

$$\mathbf{r}_k = \Delta\mathbf{y}_k = C\Delta\mathbf{x}_k + D_f\mathbf{f}_k + D_d\mathbf{d}_k \tag{6.9}$$

where $\Delta\mathbf{x}_k$ is the state estimation error. The residuals are a function of both the faults and disturbances. In large-scale systems, disturbances can be significant, which motivates the use of transformed output errors as the residual,

$$\mathbf{r}_k = W\Delta\mathbf{y}_k. \tag{6.10}$$

where the matrix W is designed such that the residuals are insensitive to disturbances but sensitive to faults. One common method for designing both matrices H and W is the unknown input observer (UIO) method. This method attempts to design the observers such that the effects of disturbances approach zero asymptotically [234].

The isolation and identification steps then occur via a structured residual set, where *structured* implies that each residual is designed to be sensitive to only one particular fault [41].

The above approach generalizes directly to nonlinear dynamical systems and to models with explicit uncertainty descriptions—the latter known as *robust observer-based fault diagnosis* methods [83]. A challenge in applying the latter methods to industrial systems is that the requirement of having accurate models of the nominal system, the faults, the disturbances, process noise, and the structure of the model uncertainties.

6.2.4 Process recovery

The end goal of all process monitoring is process recovery, where the process is returned to its normal operation. Most FDD methods will require manual intervention once a fault has been diagnosed. Fault-tolerant control (FTC) refers to a control system that automatically performs process recovery, that is, without real-time human intervention [184, 282, 296].

The objective of FTC can be interpreted as treating faults as if they are disturbances, to return the system to acceptable operation either via retuning or restructuring the control system [152]. FTC can generally be divided into two methodologies: passive and active. In passive FTC, the process monitoring system observes the process data and decides if a fault has occurred, using methods as described in Section 2, with the fault classes known *a priori*. The control system is designed with redundancies so that it is not necessary to reparameterize or restructure the controller during faulty operation. If there is more than one system fault possible, this approach often leads to a conservative controller design with slow closed-loop performance [115].

In active FTC, depending on what conditions are detected, the controller is reconfigured for that scenario (see Fig. 6-6). A major challenge of active FTC is the coordination of the process monitoring and control systems [196]. Furthermore, most FTC design methods assume that faults are detected and isolated correctly and instantaneously, to allow for computational tractability [196].

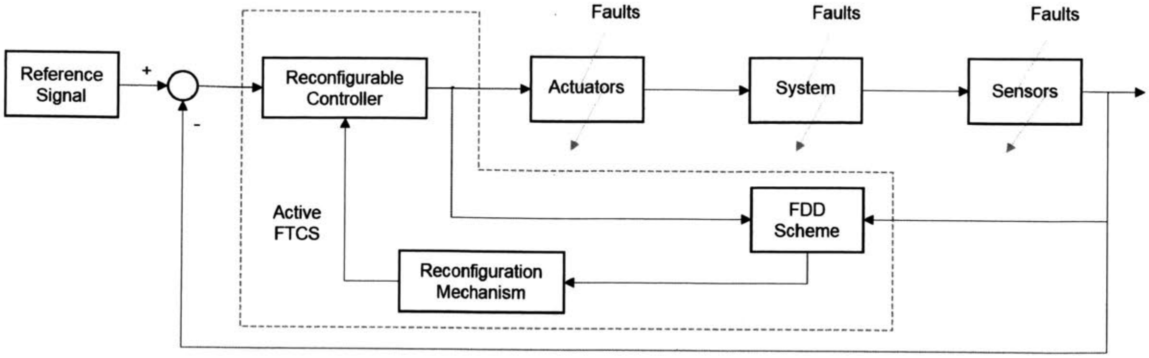


Figure 6-6: Architecture of an active FTC, adapted from [115].

Some recent algorithms combine active FDD with active FTC. Active FDD uses a test signal, called an *auxiliary input*, to generate data that enables more effective determination of whether a fault has occurred or, if a fault has been detected, which fault has occurred. This approach addresses one of the major issues of passive FDD, which can have difficulties identifying faulty conditions because the process can mask faults, particularly if the process is under control [172]. One method of active FDD is set based, which aims to find the separating inputs which guarantee fault diagnosis [172, 195, 222, 223]. This technique has been combined with model predictive control to guarantee diagnosability given input and state constraints for linear systems [196]. These methods are formulated for discrete-time models. Unlike the generalization of many results from discrete-time models to continuous-time models, the generalization of these results to continuous-time models would be challenging.

6.2.5 Comparisons of classical methods

Each process monitoring method has advantages and disadvantages. The data-based dimensionality reduction techniques of PCA and PLS are easy to implement for fault detection and isolation but of limited value for fault identification. Graphical models have the ability to incorporate expert knowledge, which is a positive if such information is available, but also require expert knowledge in their construction, which is a negative if such information is not available. State-space models require a lot of investment to develop and maintain for an industrial system, but have the potential for including very precise information on faults and disturbances in fault diagnosis procedures. The research area of process monitoring is still very active as researchers aim to tackle some of the drawbacks of various methods.

6.3 Challenges and opportunities

In the past twenty years, the quantity of data that can be collected and processed for industrial processes has greatly increased. The development of new tools such as smart and wireless sensors, the Internet of Things, smart devices, and smart manufacturing has allowed the amount of available data to grow exponentially [194]. Although FDD methods are often categorized as model-, data-, or knowledge-based, all FDD models require process data for validation and successfully utilizing this data is a key challenge and opportunity for the continued improvement of process monitoring. This section presents challenges in the field that could be addressed using this new data and methods tailored to such data.

Increasingly, these new datasets are referred to as *Big Data.* Big Data is characterized by four characteristics referred to as the 4 V's: velocity, volume, variety, and veracity [103]. These characteristics will be referenced throughout the section.

Although this section focuses on methods, it is useful to first comment about data infrastructure. Because the very large size of the data (volume), and the quick rate at which data are collected (velocity), new data systems are required. Data-centric architectures and distributed storage and processors need to be used for the value of Big Data to be realized [194]. In other words, the data are useless if the data cannot be accessed and processed reliably with reasonable computational cost. Waiting a longer time to access the data and compute a useful result from the data is not always an option, as the time available for making decisions based on the data is constrained by the time in which such decisions would be useful. This consideration is especially important in process monitoring, as faults need to be detected and diagnosed quickly enough that damage to the system is limited. A technology for improving access to

Big Data is Hadoop [178], which is a distributed file system and distributed computing framework specifically designed to handle Big Data. All modules in Haddop are designed to automatically handle any computer hardware failures, such as crashes of processors within computer clusters, with minimal disruption on the calculations applied to the data. More recently, Spark, an open-source processing engine developed at UC Berkeley, has been gaining popularity as an additional tool for Big Data analytics [55].

6.3.1 Utilizing new data sources

Beyond needing to handle a "black-box" of data as described above, new methods are required to handle new features (variety) of Big Data datasets. One of these features is high-dimensional data. In high-dimensional data, it is often the case that there are many more measurements per sample than samples, which can lead to ill-conditioning. Methods such as PCA address ill-conditioning by projecting the data into a lower dimensional space. However, with the increase in the new of measurements, there may be motivation to select a subset and not a subspace. A subset may allow for a decrease in the number of sensors which can be desirable to decrease maintenance and data storage costs. To find subsets, several avenues exist such as subset selection via optimization, penalty methods, and greedy methods. One approach is to use mutual information as the selection criteria for a greedy approach [267]. A drawback of the greedy approach is the lack of optimality guarantees. Mutual information is also not necessarily the best metric. Research is needed in this area to better understand tradeoffs between the number of sensors and the accuracy of the model. This issue is inherently intertwined with design of experiments for new process development. Experiments should be planned with process monitoring in mind such that the most valuable data can be extracted for the lowest cost while still considering standard operations. The issue of the connection of data-based monitoring and process design has not yet been solved.

Another feature of Big Data is the presence of higher-order tensors associated with new types of measurements such as real-time spectroscopic imaging or video. Instead of vectors or matrices, a single "measurement" can consist of third-, fourth-, or higher-order tensors. An example would be an inline imaging system used to characterize the shape properties of crystals in fluid flow (see Fig. 6-7), in which a single measurement at a time instance is a second-order tensor (aka matrix), with the two dimensions being space along horizontal and vertical axes, with each pixel being a grey-scale value between 0 and 255. Typically such data are collected at many frames per second at time scales much faster than the process time scales, with few particles per image. To obtain statistically reliable measurement, each measurement is treated as a video collected from seconds to minutes, which consists of many individual images (aka frames). This measurement constitutes a third-order tensor with the third dimension being the time axis over a short period of time. For color imaging systems, the order of the tensor increases by one, with the additional dimension being the color axis for red, green, and blue. The data are stored as a number between 0 and 255 for red, green, and blue at each pixel, for a two-dimensional array of pixels that make up an image. When the measurement is video over a short time period, a single measurement is a fourth-order tensor (that is, two physical dimensions, color, and time). Stacking the data into vectors and then applying PCA and PLS methods is suboptimal in practice, and such methods ignore the inherent correlations and internal structure that such datasets possess, such as that neighboring images in a video have dominant signals being shifted slightly in space as particles move. The quality of model predictions based on such data would be improved if higher order correlations and internal structure were explicitly exploited by the methods.

A related feature of Big Data is heterogeneity. New data sources are increasingly heterogeneous in terms of types and time scale. For instance, some data in the bioprocess industry are collected online, such as dissolved oxygen in a bioreactor as a function of time, while other data are collected offline, such as cell density [39]. Both sets of data provide valuable information about the status of the bioreactor, and new methods are needed for efficient integration. Some level of integration can be obtained via similarity scores and kernel transformations [39], but a lot of research is needed to generate optimal methods. Methods developed to apply to Big Data

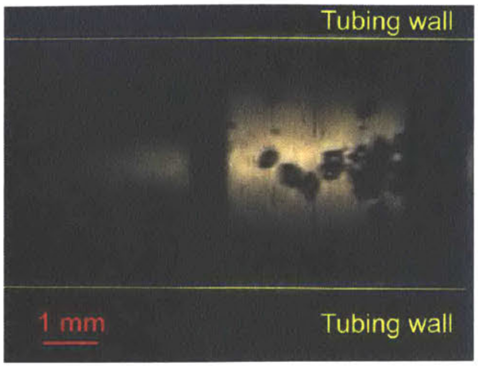


Figure 6-7: An in-line stereomicroscope image for the monitoring system of a crystallization process in which particles are in liquid slugs that flow down a tube [116]. Many such images are collected each second in real-time video. This type of data highlights the high-order structures occurring in modern datasets.

need to be able to handle rare-event data well. In fault detection, because the goal is often to find an anomaly, careful attention must also be given to data cleaning. Data cleaning is a process of removing faulty data while still retaining unexpected values. If an analysis does not take care in handling data cleaning, the behavior of interest can be overlooked.

6.3.2 Semi-supervised and online learning

Another challenge deals with using all available data. Here, specifically, the interest lies in using unlabeled data that is readily available from operations. Particularly in industrial applications, it is not reasonable, for safety or financial concerns, to purposely generate faulty data for training process monitoring algorithms. Therefore, datasets to be used for process monitoring are inherently unbalanced and methods attempt to characterize nominal operations without access to faults. In a best-case scenario, a small subset of the data is labeled as associated with some fault, but most data are not. In this setting, a state-space model using either parameter or prediction residuals may be successful, but such models are expensive to develop and maintain for complex industrial systems. Data-based methods such as PCA may be successful, but have limited capability for fault identification. Therefore semi-supervised and online learning methods should be a focus of future research.

Unsupervised learning refers to model building without knowledge of the true value of the output. Clustering and density estimation are common examples of unsupervised learning [16]. The opposite approach is supervised learning, where the targets are known. Supervising learning is ideal, but typically unreasonable in fault detection applications for the aforementioned reasons. Semi-supervised learning is in-between, where some but not all targets are known. In online learning, sometimes also referred to as *sequential learning*, the model is continually updated as additional data become available [16, 169]. These methods are more suited to the constraints of the fault detection problem. Some work in these areas is already being done. In [117], the set of features that characterize faults are calculated online as new data are streaming. The approach requires limited to no prior fault information. [117] apply the approach to the monitoring of stamping tonnage signal analysis and are able to detect faults related to shut height, which is a common process variable in these operations. Another example is [298], who also develop a technique that adapts over time. In their work, a small set of labeled data to train the model, i.e. semi-supervised. [79] also uses a semi-supervised approach, although for the goal of process modeling and not fault detection.

Semi-supervised, unsupervised, and online learning methods are gaining increased focus in the machine learning literature. The fault detection and diagnosis community would benefit from leveraging results from the machine learning community, by tailoring the methods to the specific needs of FDD problems. Some examples of methodologies for utilizing unlabeled data are support vector machines (SVM) [219, 287] and Parzen density estimates [183]. Many advances have been made more recently in deep learning [132] and the leveraging of such advances in FDD would be interesting.

6.3.3 Addressing process uncertainty

Another challenge is most closely related to data veracity. Reliable process monitoring can often be limited due to process uncertainties, which inhibit interpretation of process data [34]. Much of the past work has focused on deterministic bounded uncertainties, while some newer work has shifted that focus towards formulations that utilize probability distributions to characterize the uncertainties. For example, [299] consider uncertainty in the inputs and parameters of linear systems and propose reducing the robust fault detection problem to a standard H_∞ model-matching problem. The central concept of the work is to find a robust fault detection filter. As an example of handling probabilistic uncertainties, [162] proposed one such system that treats probabilistic uncertainties in the parameters and initial conditions of a nonlinear system, and utilizes polynomial chaos theory for uncertainty propagation. The input design is then performed using a constrained nonlinear optimization. Readers interested in robust process monitoring methods are encouraged to read the papers cited in the above publications.

6.4 Hybrid methods

The next generation of process monitoring systems need to meet a variety of needs including reliability, ability to handle uncertainty, and ability to utilize large quantities of data. An important technique for handling these demands is the use of hybrid methods that capture the strengths of different methods while minimizing their weaknesses. This section highlights some examples of hybrid models.

One example is the approached used by [44] as applied to the Tennessee Eastman benchmark problem. Their technique aimed to improve upon PCR/PLS which ignores information on process connectivity by instead using a causal map and a modified distance metric. A causal map is easily developed in many chemical applications using existing process flow or piping and instrumentation diagrams. This causal map is a type of graph that can then be combined with information theory and multivariate statistics to measure changes in the distributions of variables and in relationships between distributions of causally related variables. Furthermore, because the directed graph is directly related to the process, fault propagation could be visualized in real time (see Fig. 6-8 for an example of this visualization).

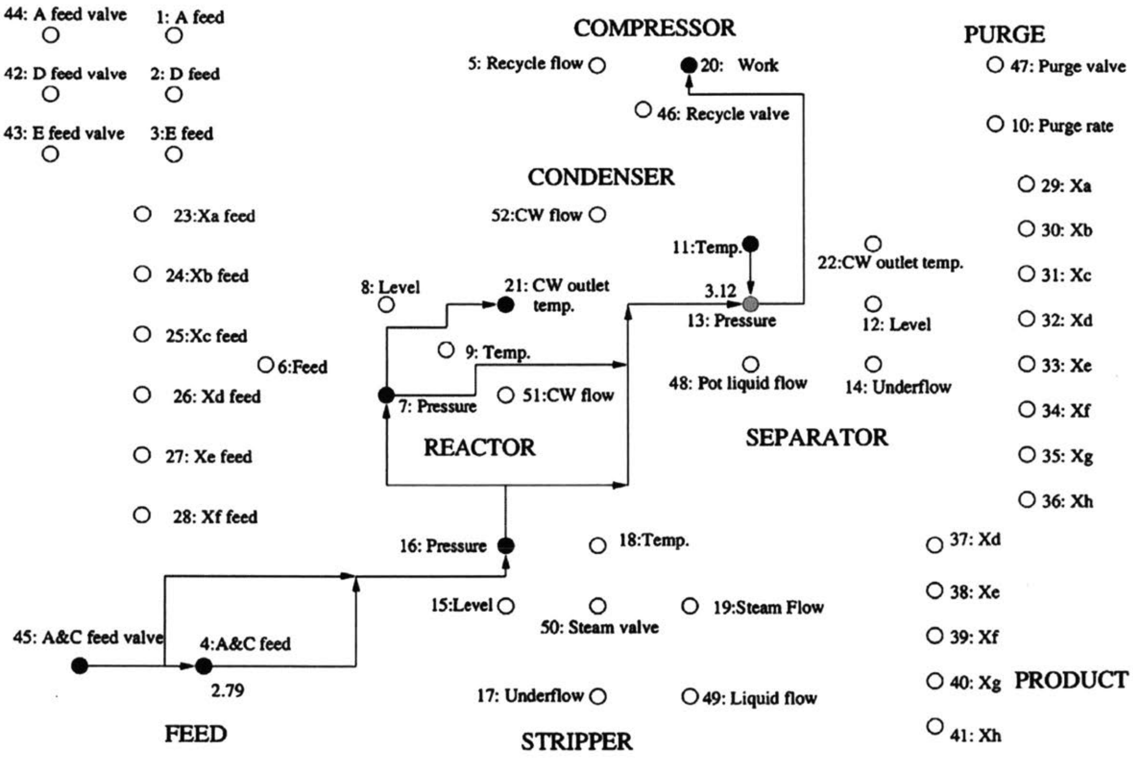


Figure 6-8: Visualization of a fault propagation using the Tennessee Eastman benchmark problem [44].

Another example of a hybrid method is the CVA-FDA method proposed by [114]. This method was implemented to tackle the challenge of fault identification and diagnosis in the presence of data overlap. This work was also applied to the Tennessee Eastman benchmark. Initially FDA was applied to the problem but it was determined that the data had too much serial correlation for FDA to provide good separation. Therefore, drawing from the state-space literature, the authors first applied CVA then FDA to handle the serial correlations and then perform fault diagnosis and identification. Using this technique decreased the misclassification rate by approximately 40% compared to using FDA alone [114].

A third example of the power of hybrid methods relates to active FDD. Active FDD methods are largely either stochastic or set-based. Stochastic methods provide convenient descriptions but do not provide guarantees, whereas set-based methods compute hard bounds but are often based on worst-case uncertainty. Hybrid methods were proposed by [224] and [157] to compromise between these two methodologies by using model uncertainties described by pdfs of finite support but also guaranteed correct diagnosis at a given time, N, while maximizing the probability of correct diagnosis at some earlier time (see Fig. 6-9). These approaches provide better flexibility compared to using purely stochastic or purely deterministic approaches.

Many other examples of hybrid approaches are described in the literature, e.g. [44, 45, 46, 112, 113, 155, 212]. The process monitoring field has been increasingly focused on complex and high-value processes over the past 40 years. Hybrid systems show the most promise for being able to handle the fault scenarios that arise in such systems.

6.5 Conclusions and future directions

This chapter provides an overview of process monitoring methods and introduces the major challenges facing the next generation of techniques. The article advocates for the use of hybrid methods to address these challenges in modern and complex facilities and provides some examples of how hybrid methods have been successful

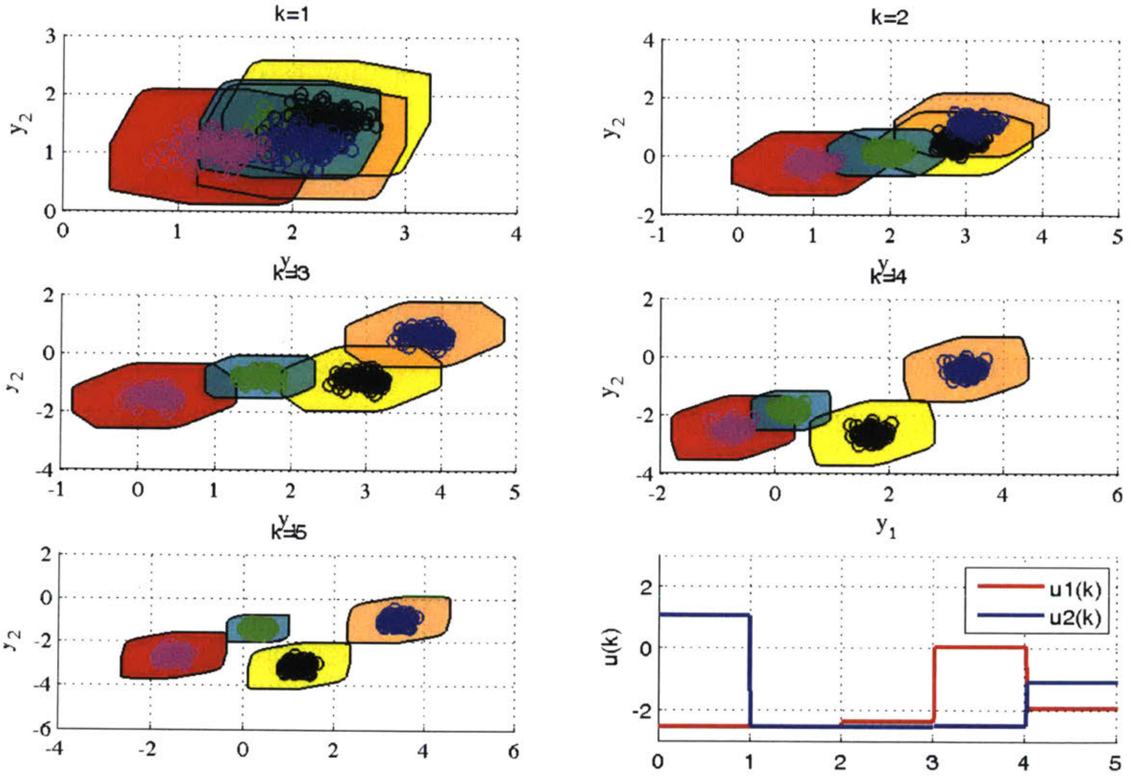


Figure 6-9: Reachable output sets of nominal and faulty models using the hybrid stochastic-deterministic approach implemented with the input $\tilde{u}_1$, which guarantees separation in five steps and approximately maximizes the probability of diagnosis in three steps [157].

in past studies. The process monitoring field would benefit from increased sharing of data for the comparative evaluation of process monitoring systems. The machine learning community has benefited greatly from the availability of public data sources, for example, the Wall Street Journal corpus used for speech recognition and natural language processing [185], the PASCAL challenge for image recognition [70], and the MNIST dataset for digit recognition [133]. The FDD community is also heavily dependent on data. Robust and implementable models need real process data for training and testing. The Tennessee Eastman chemical manufacturing facility meets this need in many ways [47]. However, the community would benefit from additional data, particularly real data or from a different manufacturing setting such as pharmaceutical manufacturing or oil well data. Progress in process monitoring systems would benefit from the availability of public datasets for comparative studies to focus on the most promising directions in algorithm development.

Chapter 7

Anomaly Detection and Diagnosis using Semi-supervised Models for Industrial Time-Series Data

7.1 Introduction

Anomaly detection commonly refers to the task of finding unexpected patterns or behaviors [37]. Anomaly detection is a challenging task because of the lack of recorded past anomalous events and their rare occurrences. Recently, in industrial systems, the amount of available data has increased, due to factors such as improved sensor technology, decreased storage costs, and the Internet of Things [194]. One potential use for these data is to improve anomaly detection systems.

Typically, the anomaly detection problem is performed using a two-step approach. First, training data, either known or assumed to be nominal, are used to develop a distance or probability metric. Second, a threshold is set for determining if a new point is sufficiently different so as to be labeled anomalous [241]. Because only nominal data is used, these approaches are referred to as one-class classification, or *unsupervised*. In the event known examples of anomalies are available, two-class, or *supervised*, approaches can be employed instead. When both labeled and unlabeled data are used in model training, the approach is call *semi-supervised* [38]. One sub-class of semi-supervised methods is positive and unlabeled (PUL) approaches. PUL describes the type of dataset, also called *presence-only* datasets, where information concerning the labels of one class is available for some data and the rest of the labels are unknown. The characterization of habitats for the presence or absence of a particular species is one example of PUL data, where it is easy and provable to label the presence of an animal but difficult to guarantee its absence [275]. A similar situation occurs in industrial systems where anomalous operations are rare and difficult to label and labeling requires an expert. Given these similarities, a framework that leverages results from the PUL literature was proposed for the anomaly detection problem.

In this article, a framework for anomaly detection and diagnosis (ADD) for industrial operational systems is presented. The ADD task is achieved using a Neyman-Pearson (NP) classification model built using a feature transformation of the raw time series data. An application of the approach is demonstrated using oil and gas well production data and is compared to the performance of one-class approaches.

7.2 Background

7.2.1 Semi-supervised models for anomaly detection

Blanchard *et al.* [18] proposed a novelty (anomaly) detection technique for applications where the dataset contains labeled examples from the nominal class as well as an unlabeled sample of potentially both nominal and anomalous examples. The resulting model uses a likelihood ratio test which is trained using the Neyman-Pearson criterion

$$\max_{\hat{H}(\cdot)} P_D \text{ subject to } P_F \leq \alpha \tag{7.1}$$

where P_D is the detection probability, P_F is the false detection probability, and α is a user-specified level. $\hat{H}(\cdot)$ is a decision rule based on a likelihood ratio test. Blanchard *et al.* [18] show that the nominal and unlabeled samples can be treated as two classes and the optimal test of size α is the same for the test of nominal vs. unlabeled and nominal vs. anomalous. To characterize the likelihoods, a kernel density estimate is proposed using a Gaussian kernel. The resulting model is of the form:

$$f(x) = \begin{cases} 1 & \text{if } \frac{m}{n} \frac{\sum_{i=m+1}^{m+n} \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \sigma_X^2 \mathbf{I}_d)}{\sum_{i=1}^{m} \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \sigma_0^2 \mathbf{I}_d)} > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{7.2}$$

where $\mathcal{N}(\cdot|\mu, \Sigma)$ is a multivariate normal distribution with mean μ and covariance matrix Σ, σ is the kernel bandwidth, $\mathbf{I}_d$ is an identity matrix of size d, and the first m samples are labeled nominal and the remaining n samples are unknown. The model requires choosing two bandwidths, σ_X for the unlabeled sample and σ_0 for the nominal sample, and α.

In the event that there are no anomalous samples in the unlabeled set, the proposed model performs no better than random guessing. To account for this possibility, a uniform sample can be appended to the unlabeled set, which will cause the model to perform similar to a level set estimation, i.e. one-class classification [18].

7.2.2 Previous work

Much of the industrial process monitoring literature has focused on the related problem of fault detection in a supervised setting, e.g. [46]. However there are some examples of unsupervised and semi-supervised approaches. One-class support vector machine (SVM) [219] and support vector data description (SVDD) [242, 243] are two techniques that have been applied in the process monitoring field e.g. [291, 230, 154]. In one-class SVMs, the objective is to learn a hyperplane that separates the nominal data from the origin with the maximum margin. In SVDD, the objective is to learn an enclosing boundary of the nominal dataset. Both approaches are amenable to kernelized feature space and can be solved as a quadratic program. Alternatively, the probability distribution of the dataset can be characterized, for instance via a Parzen density estimate [183]. To make predictions, a threshold is set and data points with probability below the threshold are labeled anomalous.

Semi-supervised approaches are less common in industrial process monitoring.

Monroy *et al.* [164], Yan *et al.* [288], and Ge *et al.* [80] all propose approaches that utilize unlabeled data to improve process monitoring performance, however, it is assumed that the labeled dataset contains nominal and faulty data. The case of positive and unlabeled data has been applied in other areas. As noted in the introduction, PUL has been applied to habitat modeling [275]. Other applications of PUL include document classification [68, 142], learning gene regulatory networks [36], and land cover classification [137].

7.3 Approach

Applying the semi-supervised anomaly detection method to time series data has three main steps: (1) transformation to feature space, (2) training the NP classification model, and (3) setting the threshold. Each of these steps is described in detail below.

7.3.1 Feature space transformation

Feature-based approaches to analysis with time series data have been previously proposed in the literature (e.g. [76, 77, 102, 80, 75]). A feature can utilize univariate data, $f : \mathbb{R}^t \rightarrow \mathbb{R}$, or multivariate data, $f : \mathbb{R}^{txn} \rightarrow \mathbb{R}$. Because time-series data are correlated, feature-based approaches can assist in approximating independent and identically distributed data, an assumption of many data-driven techniques. Feature-based approaches also provide a methodology for handling multivariate time series with different scales without pre-normalizing. In this work, an expert feature set using non-overlapping time windows is proposed for the oil and gas well monitoring task. The choice of feature space will depend on the application area. Once the features are computed they are scaled using robust sigmoid scaling

$$\hat{x} = \frac{1}{1 + \exp\frac{-(x-\bar{x})}{\frac{1}{1.35}\mathrm{IQR}_x}} \tag{7.3}$$

where $\bar{x}$ is the median and IQR_x is the interquartile range using only the training data. The scaling parameters are then applied to the testing data. Robust sigmoid scaling is chosen so that each feature has approximately the same range.

7.3.2 Semi-supervised anomaly detection

As described in the background section, Neyman-Pearson classification models use likelihood ratio tests as the decision rule where the threshold is determined based on a specified false positive rate. Here, the likelihood is estimated using a Parzen density estimate with a Gaussian kernel. To train the model, the bandwidths of the Gaussian kernel densities must be estimated for the nominal and unknown classes. The kernel density estimate is given by

$$\tilde{p}(\mathbf{x}) = \frac{1}{n\sigma^d} \sum_{i=1}^{n} k_\sigma\Big(\frac{\mathbf{x} - \mathbf{x}_i}{\sigma}\Big) \tag{7.4}$$

where d is the dimension of the vector $\mathbf{x}$, n is the number of samples, and k_σ is the kernel parameterized by σ [232]. The Gaussian kernel is

$$k_\sigma(\mathbf{u}) = (2\pi)^{\frac{d}{2}} \exp\Bigg(-\frac{1}{2}\mathbf{u}^\top \mathbf{u}\Bigg) \tag{7.5}$$

A cross-validation strategy is adopted to choose the bandwidths by maximizing the positive and unlabeled learning performance (PULP) metric [90]. PULP is defined as

$$PULP(S|n,t) = \frac{1}{n+1} \sum_{i=1}^{n} F(k_{\{S,i\}} - 1|n,t,i) \tag{7.6}$$

where F is the cumulative hypergeometric function, $S : \{s_1, s_2, ..., s_n\}$ is the sorted list of the classifier outputs, n is the total number of samples, t are the labeled samples, and $k_{\{S,i\}}$ is the number of hits, given that those items are predicted to be positive [90]. PULP is chosen over other metrics such as the area under the receiver operator curve because PULP has been shown to be more robust when applied to settings with non-random sampling, which is likely the scenario in process monitoring, where data are correlated in time. Additionally, in PUL settings, the labeled nominal samples are likely not chosen randomly, as some samples can be labeled more confidently than others [90].

7.3.3 Threshold setting

Using the results of Blanchard *et al.* [18], the threshold learned from the surrogate problem should be applicable to the underlying anomaly detection problem. However, because the application area is high-dimensional, the threshold-setting approach of Zhao *et al.* [297] is adopted here. m_2 examples of the labeled nominal data, S^0, are not used in the model training phase. The learned likelihood ratio test f is applied to this data. The threshold is set to be $f_{(k)}(S^0)$, which is defined as the k-th order statistic of $f(S^0)$. k is defined as

$$k(\alpha, \delta, m_2) = \min(\lceil (m_2 + 1)A_{\alpha,\delta}(m_2) \rceil, m_2) \tag{7.7}$$

where $\lceil x \rceil$ is the smallest integer greater than or equal to x and

$$A_{\alpha,\delta}(m_2) = \frac{1 + 2\delta\tilde{m}(1-\alpha) + \sqrt{1 + 4\delta(1-\alpha)\alpha\tilde{m}}}{2(\delta\tilde{m} + 1)} \tag{7.8}$$

where $\tilde{m} = m_2 + 2$, α is the user defined level of probability of false detection, and $\delta \in (0, 1)$ is user-defined. Zhao *et al.* [297] use $\delta = 0.05$ in their work and the same value is adopted here.

7.3.4 Model visualization

One limitation of non-parametric density estimates is their ability to be interpreted. Interpretability is important in ADD problems because an intervening action must be selected. Motivated by control charts [193, 301], an approximated visualization of the contributions of each feature can be found by applying a log transformation and

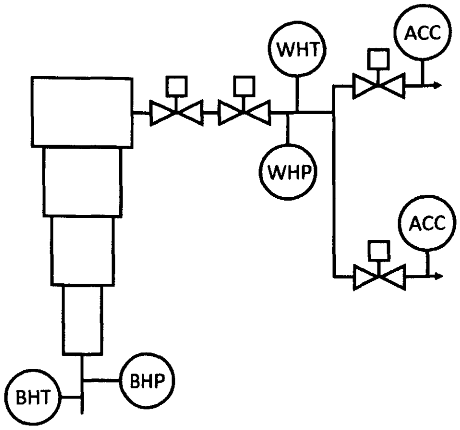


Figure 7-1: Schematic of the six sensors implemented on the case study well. Temperature and pressure are measured at the bottomhole and wellhead. Acoustic measurements are measured prior to the manifold. HP and LP are high pressure and low pressure, respectively.

Table 7.1: Features used in the PUL model as well as the one-class benchmarks. General features are applied to all sensors.

General Features	BHP-only Features
Standard deviation	Peak separation from uniform
Skewness	MSE of third order polynomial fit
Kurtosis	Range
Entropy	Amplitude of sine fit
Burstiness [84]	Period of sine fit
Coarsened rate of change [76]	Goodness of sine fit
ACC-only Features	Multivariate Features
Minimum	BHT-BHP Mutual Information
Maximum	BHP-WHP Mutual Information

Jensen's inequality to the model. The resulting equation is

$$\sum_{j=1}^{p}\left[\frac{1}{2m\sigma_0}\sum_{i=1}^{m}(x_j - x_{ij})^2 - \frac{1}{2n\sigma_x}\sum_{i=1}^{n}(x_j - x_{ij})^2\right] > \log\tilde{\lambda} \tag{7.9}$$

Because Eqn. 7.9 is an approximation, this equation is used only for visualization and not for prediction.

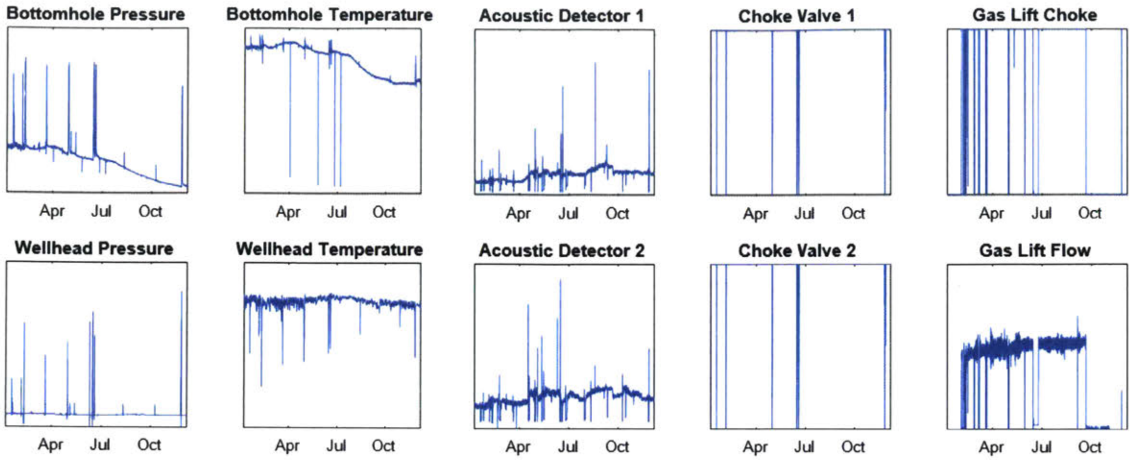


Figure 7-2: Data from each of the sensors for the well as well as the valve positions.

7.4 Case study

The anomaly detection approach is demonstrated using data from a production oil and gas well. The well scenario exhibits many of the characteristics of the PUL framework. The success of the oil and gas well depends on continuous operation and therefore it is critically important to identify anomalies. Modern wells are deployed with sensor systems that collect data continuously in time [54]. It is challenging to assess when the well is in a anomalous state but somewhat simpler to assess 'good' performance.

To implement the approach, early production data is assumed to have some number of unlabeled anomalies and is used in the training phase. To account for the possibility of no anomalies, a uniform sample is appended to the unlabeled set, following the hybrid approach recommended by Blanchard *et al.* [18]. 10% of the data is labeled nominal based on sensor measurements and operator logbook information. Because of the unique characteristics of each well, both in terms of operation strategy and sensor deployment, it is recommended that a different model is trained for each well utilizing early production data (approximately six months).

A schematic of the sensor system is shown in Fig. 7-1. Measurements are averaged over a fixed two-minute sampling interval for the bottomhole pressure (BHP) and temperature (BHT), the wellhead pressure (WHP) and temperature (WHT),

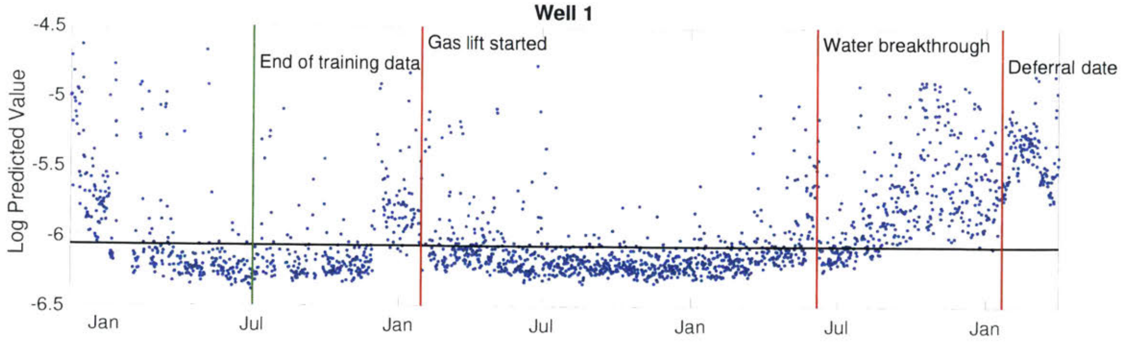


Figure 7-3: Prediction results for the oil and gas well using the proposed methodology. All points above the black threshold indicate anomaly predictions. Data before the first vertical line is used in training and all remaining data is for testing. Anomalies are predicted prior to all three of the major well events: the start of the gas lift, water breakthrough and the deferral date.

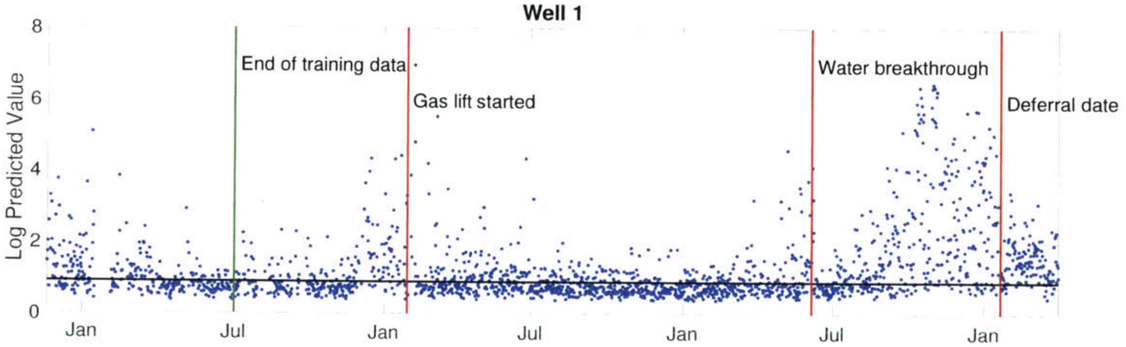


Figure 7-4: Prediction results for the oil and gas well with a Parzen density model. All points above the black threshold indicate anomaly predictions. Data before the first vertical line is used in training and all remaining data is for testing. Although the general trend is similar to the PUL model, this prediction has a greater number of false positives and fewer true positives.

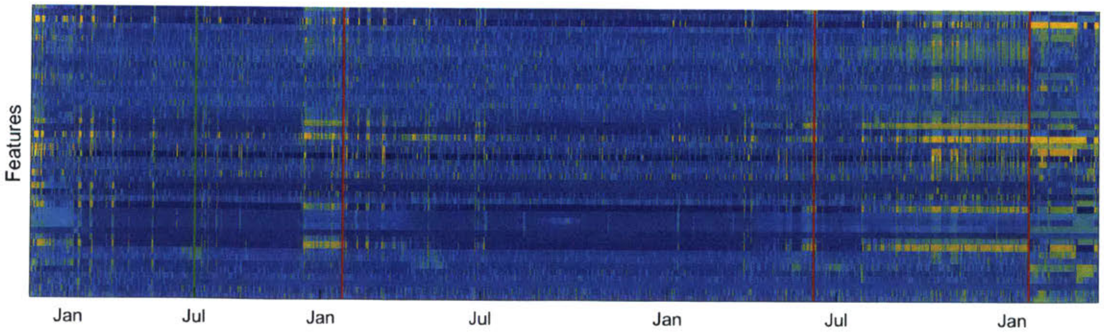


Figure 7-5: Heatmap of the feature values for each prediction day. As in Fig. 7-3, the vertical lines correspond to the end of the training data, the start of the gas lift, water breakthrough, and the deferral date, from left to right.

and acoustic sensors (ACC1 and ACC2). Sensor data from one year of production are shown in Fig. 7-2. The feature set is applied and scaled as described in section 3.1. The specific features used in this application are listed in Table 7.1. The two bandwidth parameters and threshold are trained using the first six months of operational data. 10% of the data is labeled: half is used to tune the bandwidths using the PULP metric and a 3-fold cross validation strategy and the other half is used to set the threshold as described in section 3.3 with $\alpha = 0.05$ and $\delta = 0.05$. The results of the model are shown in Fig. 7-3.

The model predicts anomalous behavior prior to all of the major well events, as shown in Fig. 7-3. Points above the black line indicate anomaly predictions. This particular well has three major events: start of the gas lift, water breakthrough, and deferral date. The gas lift alters the bottomhole pressure in an effort to improve well production. Gas lift startup indicates a reaction by the well operators to decreasing well performance. Water breakthrough occurs when water enters the well bore and is determined based on total water production. A deferral indicates the need to suspend production to allow for an intervention. It is therefore appropriate that the model would predict anomalous conditions prior to each of these events.

To compare performance, Parzen density models and support vector data descriptions (SVDDs) were applied to the feature data. Several variations of model training were considered: (1) the full training dataset with a specified percentage of outliers, (2) the principal components of the full training data set with a specified percentage of outliers, (3) only the labeled nominal examples in the training data set with no outliers, and (4) the principal components of the labeled nominal examples with no outliers. For cases (1) and (2), the percentage of outliers was required for training and defined as the percentage of data points where the well is not in the desired operational mode. Applying this rule to the dataset results in an outlier percentage of 40%. Note that only the percentage of outliers is need for training and particular data points do not need to be labeled. For cases (2) and (4), the number of principal components was required and was selected via parallel analysis [98], where the singular values of the data matrix are compared to the singular values of a random matrix of the same size. The number of principal components is equal to the floor of the crossing point.

Several evaluation metrics for the models are presented in Table 7.2. The true positive rate (TPR) is the number of correct anomaly predictions divided by the number of true anomalies and the false positive rate (FPR) is the number of incorrect anomaly predictions divided by the number of true nominal points. The F1 score is the harmonic mean of the precision and recall and is defined as

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{7.10}$$

where TP, FP, and FN are the number of true positives, false positives, and false negatives, respectively. Area under the receiver operator curve (AUC) is the integral of all possible TPR and FPR values if the threshold was varied over its domain. The maximum and desired value for AUC is 1 and random guessing is expected to result in an AUC of 0.5. To calculate the TPR only the 3 month period prior to the deferral date was considered to be anomalous. This is a conservative approach, as other well operation periods were likely anomalous however the deferral was the largest event the well experienced over the time period available. The FPR is based only on the testing period.

The proposed methodology has the highest AUC and TPR. Several of the estimators using the full feature space are also able to achieve the highest AUC however struggle to set an appropriate threshold which results in a high false positive rate. The SVDD with PCA feature transformation using only the nominal data has the lowest false positive rate, but also has a comparatively low true positive rate. The results of the nominal Parzen model using PCA as the features is shown in Fig. 7-4. While the shape is similar to the proposed methodology, the shifts are less dramatic, resulting in a lower AUC.

The results of the visualization technique described in Section 3.4 is shown in Fig. 7-5. The model contribution of each feature is presented as a heat map The diagnosis step considers which feature values have significant deviations and maps

Table 7.2: Comparison of metrics for the proposed PUL model and one-class classification models. The percentage indicates the target amount of data to be predicted as an anomaly during the training phase.

Methodology	AUC	TPR	FPR	F1
Proposed PUL Model	0.94	1	0.31	0.61
Parzen, 40%	0.94	1	1	0.32
Parzen + PCA, 40%	0.89	1	0.78	0.38
SVDD, 40%	0.88	0.96	0.59	0.44
SVDD + PCA, 40%	0.64	0.54	0.31	0.38
Parzen, 0%	0.94	1	1	0.32
Parzen + PCA, 0%	0.84	0.89	0.4	0.50
SVDD, 0%	0.94	1	0.58	0.45
SVDD + PCA, 0%	0.74	0.59	0.23	0.46

those deviations back to sensor measurements. This provides operators with a starting point for root cause analysis. As historical data on features grows, past experience combined with clustering of these anomaly signatures can assist in improved root cause and corrective action determination.

7.5 Conclusions

The proposed positive and unlabeled methodology has two main strengths: the ability to utilize unlabeled data during training and the ability of the feature space transformation to capture the status of the well. It is common in industrial applications to have historical data that is very difficult to label. Unlike some applications, expert analysis is required for labeling, which is time-consuming and error-prone. However, labeling a small number of nominal data points is much more reasonable. The feature space transformation allows the model to capture important characteristics of the data. The proposed feature set is determined based on input from oil and gas well operation experts, which can be limiting. Future work will consider automation of this task to make the approach even more flexible.

Finally, the model is very fast and easy to calculate. Training the model requires setting only four parameters: two bandwidths, α and δ, the latter two are related to the threshold calculation. The approach avoids assumptions on the distribution of anomalies. The visualization tool also provides users with a starting point for diagnosis, which is ultimately the goal in anomaly detection problems.

Chapter 8

Data-driven modeling enables quantitative cycle life prediction for lithium-ion batteries

This work has been submitted as: Kristen A. Severson, Peter M. Attia, Norman Jin, Zi Yang, Nicholas Perkins, Michael Chen, Muratahan Aykol, Patrick K. Herring, Stephen J. Harris, William C. Chueh, and Richard D. Braatz. Data-driven modeling enables quantitative cycle life prediction for lithium-ion batteries.

8.1 Introduction

Lithium-ion batteries are desirable for use in a variety of applications because of their high energy and power densities and long cycle lives [87, 69, 150, 218, 175]. However, long cycle life implies delayed feedback of battery performance during development and manufacture. Early prediction of cycle life would accelerate this feedback loop as well as enable accurate estimation of battery life expectancy for applications including consumer electronics and electric vehicles. However, the task of predicting capacity fade and/or cycle life for lithium-ion batteries is challenging. Lithium-ion batteries often have nonlinear aging profiles as a function of cycle number and wide variability, even when controlling for operating conditions [14, 91, 186, 220, 221]. Furthermore,

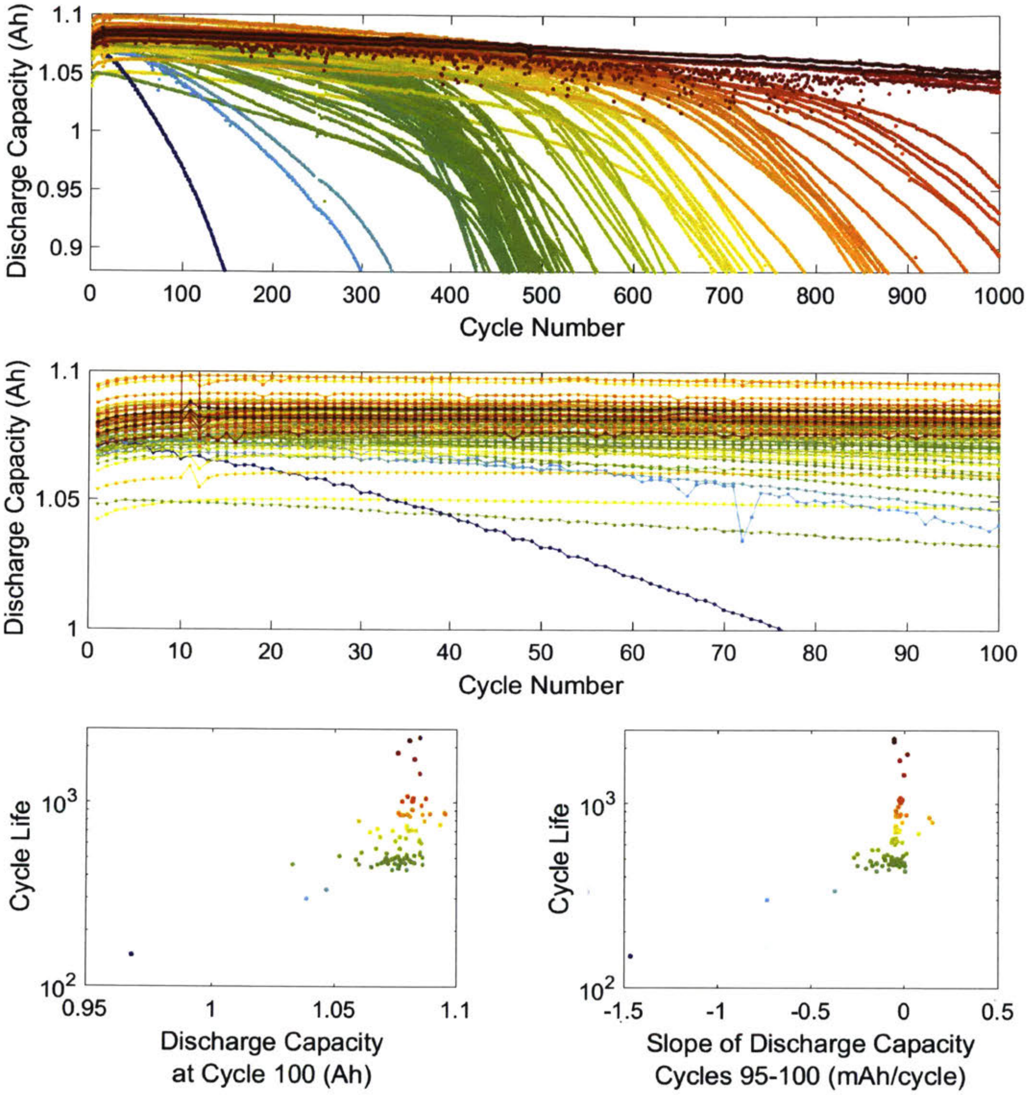


Figure 8-1: Top: Discharge capacity for the first 1000 cycles of the A123 M1A LFP|graphite cells used in analysis. The trajectories cross each other, indicating a nonlinear trend in capacity fade. The color of each curve is scaled based on the battery's cycle life, as in done throughout the manuscript. Middle: A detailed view of a, showing only the first 100 cycles. A clear ranking of cycle life has not emerged at this point. Bottom left: Cycle life as a function of discharge capacity at cycle 100. The correlation coefficient of capacity at cycle 100 and log cycle life is 0.54. Bottom right: Cycle life as a function of the slope of the discharge capacity curve for cycles 95 to 100. The correlation coefficient of this slope and log cycle life is 0.54.

many mechanisms can contribute to capacity fade in lithium-ion batteries, such as side reactions at the electrode/electrolyte interface, loss of active material, resistance increase, and issues related to the composite electrode such as changes to current collector, porosity, binder, etc. [180, 8, 268, 28]. These effects occur heterogeneously within a cell and may interact [92, 135, 11].

Many studies have considered the task of predicting cycle life in lithium-ion batteries using physics-driven models. Bloom *et al.* [21] and Broussely *et al.* [28] performed some of the first studies of modeling capacity fade in lithium-ion batteries by fitting empirical models to predict the percentage power and capacity loss. Since then, many semi-empirical studies have been conducted to capture different capacity fade phenomenon during cycling [197, 239, 50, 49, 295, 214, 272, 198, 191, 66, 53] and storage [139, 65, 238]. The key challenge in applying physics-based modeling to lithium-ion batteries is capturing the many degradation length scales, ranging from atoms and interfaces (Å-nm) [266, 82, 86, 289] to electrodes and cells (μm-cm) [271, 138, 73]. The complexity of battery degradation has hindered development of accurate models of cell lifetime.

Data-driven approaches are an alternate technique for cycle life prediction. Recently, advances in computational power and data generation have enabled machine learning techniques to accelerate progress in a variety of fields, including materials discovery for energy storage [111, 227]. Data-driven approaches to battery lifetime prediction have used data from standardized initial reference tests [14] and initial cycling [91, 143, 100] and storage [245] data. However, many studies are limited by small sample sizes (<10) and typically rely on the inherent cell-to-cell variation in cycle life for identical operating conditions. Furthermore, these analyses have focused on using capacity as a function of cycle number (Q(n)) to inform predictions, disregarding the voltage-capacity relation (V(Q)) within each cycle. Other measurements, such as high-precision coulombic efficiency [30, 31] and impedance spectroscopy [40, 252], have demonstrated predictive capability but require specialized characterization equipment.

In this study, we were particularly interested in the early prediction setting, where the goal is to use a small number of cycles to accurately predict the cycle life. Specifically, we set out to use information from the first 100 cycles to predict the number of cycles to 80% state of health (SOH) for cells with lifetimes of up to 2300 cycles. A dataset of 84 commercial cells cycled to failure was generated using a variety of fast-charging conditions. We then developed feature-based models resulting in prediction errors ranging from 5% to 15%. Our model performs well for cycle lives ranging from 150 to 2300 without incorporating diagnostic cycles. These results illustrate the utility of data-driven prediction models to accelerate the development of energy storage technologies. Similar data-driven methods are broadly applicable for complex systems that lack a full first-principles characterization.

8.2 Data generation

Because of the large number of capacity fade mechanisms and intrinsic variability of lithium-ion batteries, we expect the space that parameterizes capacity fade to be high dimensional. To probe this space, 84 commercial lithium-iron-phosphate (LFP)|graphite cells from A123 were tested in a temperature-controlled environmental chamber under a variety of charging conditions but identical discharging conditions. The study used commercial high-power $LiFePO_4$/graphite A123 APR18650M1A cells. These batteries have a nominal capacity of 1.1 Ah and a nominal voltage of 3.3 V. The manufacturer's recommended fast-charging protocol is 3.6C constant current - constant voltage (CC-CV), which charges to 80% in 13.3 minutes.

Data collection was performed using a 48-channel Arbin LBT battery testing potentiostat. The tests were performed at 30 °C in an environmental chamber (Amerex Instruments). Cell surface temperatures were recoded by stripping a small section of the plastic insulation and contacting a type T thermocouple to the bare metal casing using thermal epoxy (OMEGATHERM 201) and Kapton tape.

The charging protocols were varied amongst the cells but the discharge policy remained constant. Cells were charged from 0% to 80% SOC with various one- and two-step policies. The charging times ranged from 9 to 13.3 minutes. An internal resistance measurement was obtained at 80% SOC. All cells charged from 80% to 100% SOC with a uniform 1C CC-CV charging step to 3.6V and a current cutoff of C/50. All cells were discharged with a CC-CV discharge at 4C to 2.0V with a current cutoff of C/50.

By varying the charging protocol, the dataset captures a wide range of cycle lives, from approximately 150 to 2300 (average cycle life of 692 with a standard deviation of 362). Voltage, current, temperature, and internal resistance (IR) at 80% state of charge are measured. The dataset has approximately 58,000 cycles. Fig. 8-1 shows the discharge capacity as a function of cycle number for the first 1000 cycles, where the color denotes the cycle life. Immediately this dataset demonstrates the limitations of using only discharge capacity and cycle number for prediction. The correlation between the discharge capacity at the 100th cycle as well as the trend of capacity fade near cycle 100 have low correlation with final cycle life ($p = 0.54$). Given the limited predictive power of these models, we instead investigated a data-driven approach that leverages a larger set of cycling data which includes the full voltage-capacity relation, as well as additional measurements including internal resistance (IR) and temperature.

8.3 Machine learning approach to capture cell-to-cell variability and domain knowledge

To utilize the dataset, we proposed a feature-based approach and a linear model. In this paradigm, features, or transformations of the raw data, are generated and used in a regularization framework. The final model uses a linear combination of a subset of the proposed features to predict log cycle life. Ideally, the specific features in this linear model can be related back to underlying electrochemical phenomenon.

The modeling goals included in both model fitting, selection of the coefficient values, and model selection, selection of the model structure. To perform both of these tasks simultaneously, the elastic net approach was used [302]. While the lasso

[246] would also result in a sparse model, elastic net is preferred when there are high correlations between the features, which is the case in this application. To choose the value(s) of the hyper-parameter(s), cross-validation is recommended. Here 4-fold cross validation and Monte Carlo sampling is applied.

The dataset is divided into two equal pieces referred to as the training and testing data. The training data is used to choose the hyper-parameters α and λ and determine the values of the coefficients. The training data is further subdivided into calibration and validation sets for the cross-validation procedure. The testing data is then used as a measure of generalizability because it is data that has not been used to learn the model coefficients or form. To standardize the voltage-capacity data across cells and cycles, all 4C discharge curves were fit to a spline function and linearly interpolated. These uniformly-sized vectors enabled straightforward data manipulations.

Features were proposed based on domain knowledge of battery operation and degradation, e.g. a dependence on temperature, as well as an interest in capturing individual cell variability. To do this, several features were based on the difference of discharge capacity as a function of voltage between two cycles. Capacity as a function of voltage was of particular interest because it is a high-fidelity and rich data source and is an effective data source for degradation diagnosis [22, 20, 59, 237, 213, 60, 61, 7, 6]. Unlike the single capacity measurement, as in Fig. 8-1, the difference is a vector with units of capacity. Summary statistics were then applied to this difference, e.g. minimum, mean, and variance. An example is shown in Fig. 8-2 using the 100th and 10th cycles.

A clear linear trend emerges between the summary statistics of the difference of the voltage-capacity curves during discharge for 100th and 10th cycles (referred to as ΔQ(V) throughout this chapter). Because of the high predictive power of ΔQ(V) on its own, we investigate models using only this feature, features using only discharge information, and features using the entire dataset. This progression is used to consider the setting where measurements such as surface temperature are not available and quantify the improvement when such additional measurements are added. The complete set of 20 candidate features can be found in Table 8.1.

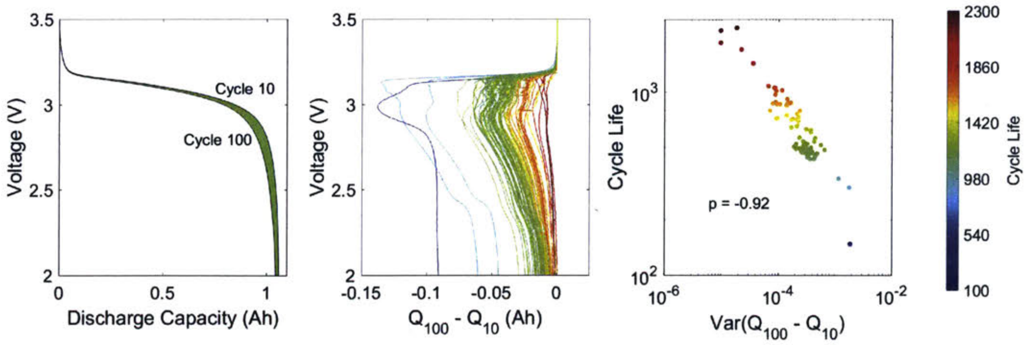


Figure 8-2: Discharge capacity curves for 100th and 10th cycles for a representative cell. b, Difference of the discharge capacity curves at the 100th and 10th cycles as a function of voltage for 84 cells. c, Cycle life plotted as a function of the variance of the difference on a loglog axis. In all plots, the colors are determined based on the final cycle lifetime. In c, the color is redundant with the y-axis.

8.4 Results

The complete dataset of 84 cells is divided into two subsets, referred to as training and testing. The training data (41 cells) is used to select the model form and set the values of the coefficients, and the testing data (43 cells) is used to evaluate the model performance. Many possible divisions of the data are possible; a 50-50 split is used to have a large amount of data to test the model's performance on unseen data. Two metrics are used to evaluate performance: root-mean-squared error (RMSE), units of cycles, and average percent error. RMSE is defined

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2} \tag{8.1}$$

where y_i is the observed cycle life, $\hat{y}_i$ is the predicted cycle life and n is the total number of samples. Average percent error is defined

$$\%err = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \times 100 \tag{8.2}$$

where all variables are defined as above. The error metrics for the various models are reported in Table 8.2.

We present three models to predict cycle life using increasing feature set sizes.

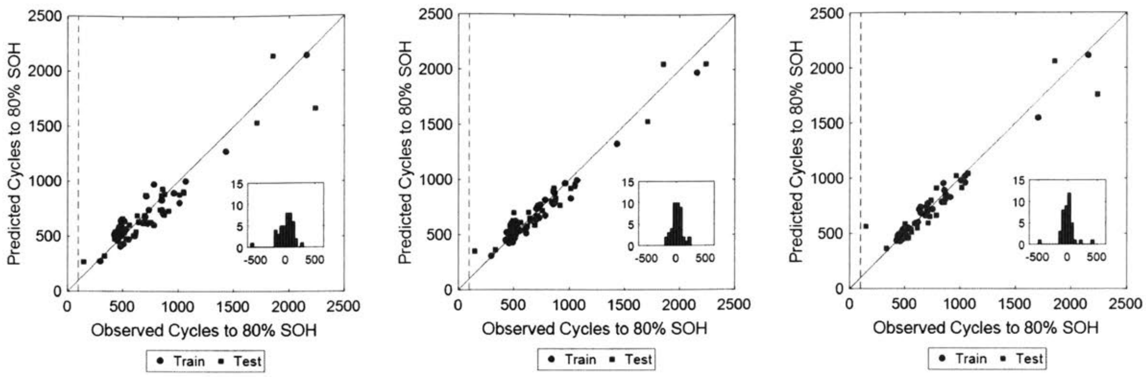


Figure 8-3: Observed and predicted cycles to 80% SOH for several implementations of the feature-based model. The training data are used to learn the model structure and coefficient values. The testing data are used to assess generalizability of the model. The vertical dotted line indicates when the prediction is made in relation to the observed cycle life. The in-lay shows the histogram of residuals (predicted – observed) for the test data. Left: model using only the log variance of ΔQ. Center: model using six features based only on discharge cycle information, described in Table 8.1. Right: model using the nine features, described in Table 8.1. Because some temperature probes did not maintain contact during experimentation, four cells are excluded from the full model analysis.

The first model, denoted var[ΔQ(V)], does not consider subset selection and uses only the log variance of ΔQ(V) for prediction. Surprisingly, using only this single feature results in a model that has less than 15% error. The second model only considers information derived from measurements of voltage and current during discharge (columns 1 and 2 of Table 8.1) and is denoted *discharge* model. The third model considers all available data (all columns of Table 8.1) and is denoted *full* model. These models select a subset of the available features using the elastic net. As expected, for increasing model complexity, the error decreases, see Table 8.2 and Fig. 8-3. The specific features and model coefficients used in the full model can be found in Fig. 8-4.

8.5 Discussion

We observe that ΔQ(V)-based features have high predictive performance, while models using only Q(n)-based features perform poorly. We rationalize this observation by investigating degradation modes that do not immediately result in capacity fade

Table 8.1: Features considered for the various model implementations. The simplest model uses only the log variance of ΔQ(V) and does not consider model selection. More complex models are considered using only discharge information (first two sections) as well as additional measurements (all sections).

	Features	var[ΔQ(V)]	Discharge	Full
ΔQ(V)	Minimum		✓	✓
	Mean			
	Variance	✓	✓	✓
	Skewness		✓	
	Kurtosis		✓	
	Value at 2V			
Q(n) features	Slope of Q(n), cycles			✓
	Intercept of the linear fit to Q(n), cycles 2-100			✓
	Slope of Q(n), cycles 91-100			
	Intercept of the linear fit to Q(n), cycles 91-100			
	Discharge capacity, cycle 2		✓	✓
	Max discharge capacity - discharge capacity, cycle 2		✓	
	Discharge capacity, cycle 100			
Other features	Average charge time, first 5 cycles			✓
	Maximum temperature, cycles 2-100			
	Minimum temperature, cycles 2-100			
	Integral of temperature over time, cycles 2-100			✓
	Internal resistance, cycle 2			
	Minimum internal resistance			✓
	Internal resistance, cycle 100-2			✓

Table 8.2: Model metrics for the results shown in Fig. 8-3. Train and test refer to the data used to learn the model and evaluate model performance, respectively. One battery in the test reaches 80% SOH rapidly and does not match other observed patterns. Therefore the parenthetical test results correspond to the exclusion of this battery.

	RMSE (cycles)		Mean Percent Error	
	Train	Test	Train	Test
ΔQ model	103	138 (138)	14.1%	14.7% (13.2%)
Q v. N model	76	91 (86)	9.8%	13.0% (10.1%)
Full model	51	118 (100)	5.6%	14.1% (7.5%)

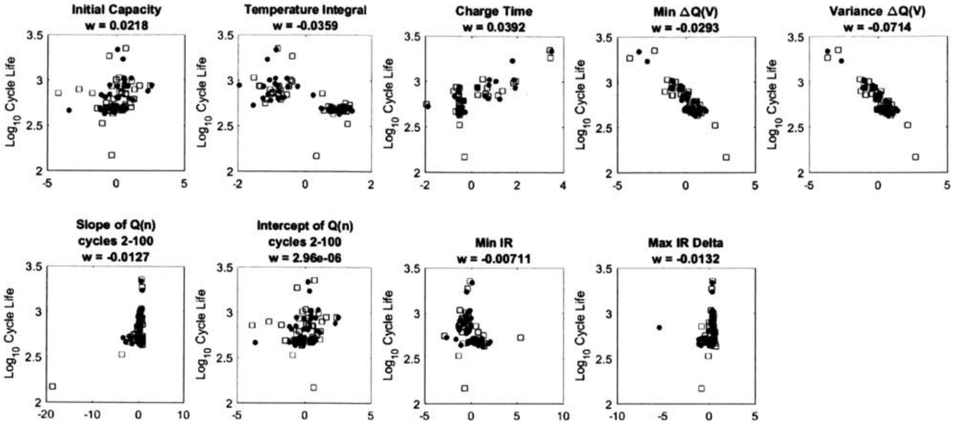


Figure 8-4: Nine features used in the predictive model described in Table 8.1. The coefficient value for the feature in the linear model is in the title of plot. The training and testing batteries are represented by circles and squares, respectively. Each of the features has been z-scored based on the training data.

yet still manifest in V(Q). The degradation mode(s) also results in the aggressive, nonlinear capacity fade at high cycle number.

Capacity fade mechanism determination has relied primarily on a popular electrochemical analysis technique which considers numerical derivatives of the voltage-capacity relation, dQ/dV = f(V) and dV/dQ = f(Q), and their evolution with cycling [22, 20, 237, 213, 60, 61, 7, 6]. Dubarry *et al.* map six degradation modes in LFP|graphite cells to their resultant shift in dQ/dV and dV/dQ for diagnostic cycles at C/20 [60]. Only one degradation mode – loss of active material of the delithiated negative electrode (LAMdeNE) – results in a shift in V(Q) with no change in capacity. This behavior is observed when the negative electrode is oversized relative to the positive electrode, as is typical in commercial lithium-ion batteries. Thus, a loss in delithiated negative electrode changes the potentials at which lithium ions are stored without changing the overall capacity [60, 7].

As LAMdeNE continues, the negative electrode capacity will eventually fall below the lithium-ion inventory remaining in the cell. At this point, the negative electrode will not have enough sites to accommodate lithium ions during charging, inducing lithium plating [7]. Since plating is largely irreversible, the capacity loss

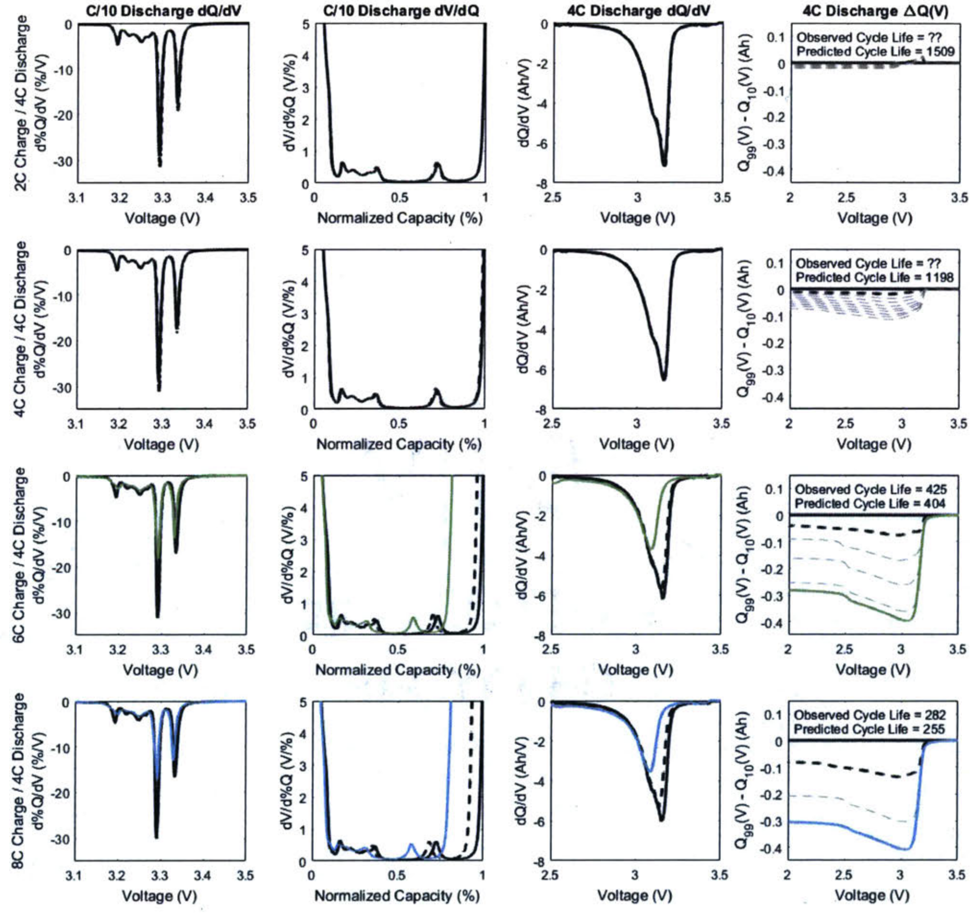


Figure 8-5: Results of four cells that were tested with periodic slow diagnostic cycles. From left to right, the plots are dQ/dV using slow cycling, dV/dQ using slow cycling, dQ/dV using fast cycling, and ΔQ(V) using fast cycling. The solid black line is the first cycle (cycle 10 for fast cycling), the dotted black line is cycle 101 or 100 (fast and slow, respectively), and the colored thick line is the end of life cycle (80% SOH). For ΔQ(V), a dotted grey line is added every 100 cycles. The patterns observed using slow cycling are consistent with LAMdeNE and LLI. The features are smeared during fast charging. The log variance ΔQ(V) model trained using the high-throughput dataset is able to predict lifetime within 10%. At the time of submission, only the 6C and 8C experiments had concluded.

will accelerate. Thus, LAMdeNE shifts V(Q) without affecting Q(n) and induces rapid capacity fade at high cycle number. Furthermore, LAMdeNE, in conjunction with loss of lithium inventory (LLI), are widely reported as the dominant degradation modes in commercial LFP|graphite cells operated under similar conditions [135, 213, 60, 7, 6, 145, 216].

To investigate if LAMdeNE is a contributing degradation mode in our experiments, additional experiments were performed with varied charging rates (2C, 4C, 6C, and 8C) and constant discharge rates (4C), incorporating slow cycling at cycles 1, 100, and end of life (80% SOH). dQ/dV and dV/dQ of diagnostic cycles (C/10) at n=1 and n=100 and $\Delta Q_{101-10}(V)$ during 4C discharge are compared. The results for four cells are displayed in Fig. 8-5. The shifts in dQ/dV and dV/dQ observed in diagnostic cycling are consistent with LAMdeNE and LLI operating concurrently [60, 7, 6]. This supports the hypothesis as to why models using V(Q) have lower errors than models using only Q(n). The var[ΔQ(V)] model developed above predicts cycle life of these batteries within 15%. Furthermore, we expect that this method is valuable for any degradation mode that does not immediately manifest in Q(n) but does impact the V(Q) relationship, such as high-voltage cathode materials undergoing voltage fade [82, 244, 278].

ΔQ(V) has many advantages for cycle life prediction over differential methods like dQ/dV and dV/dQ despite using the same V(Q) data. ΔQ(V) does not require periodic slow diagnostic cycles, instead using high rate discharge data. Slow diagnostic cycles are unrepresentative of batteries' use cases and induce a temporary capacity recovery that interrupts the trajectory of capacity fade (see Fig. 8-6) [200, 136]. dQ/dV applied to the high rate data shows only very small changes as compared to ΔQ(V) in the first 100 cycles and the individual features are difficult to discern. ΔQ(V) also avoids the use of numerical differentiation, which reduces the signal to noise ratio and thus the predictive power. Numerical derivatives also require selecting values for parameters such as step size, which can have significant impacts on the qualitative interpretation of patterns.

Additional analysis was performed to understand the impact of the cycle indices

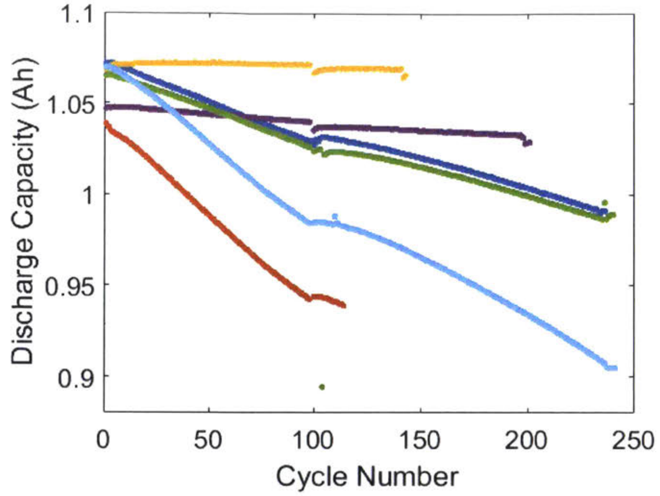
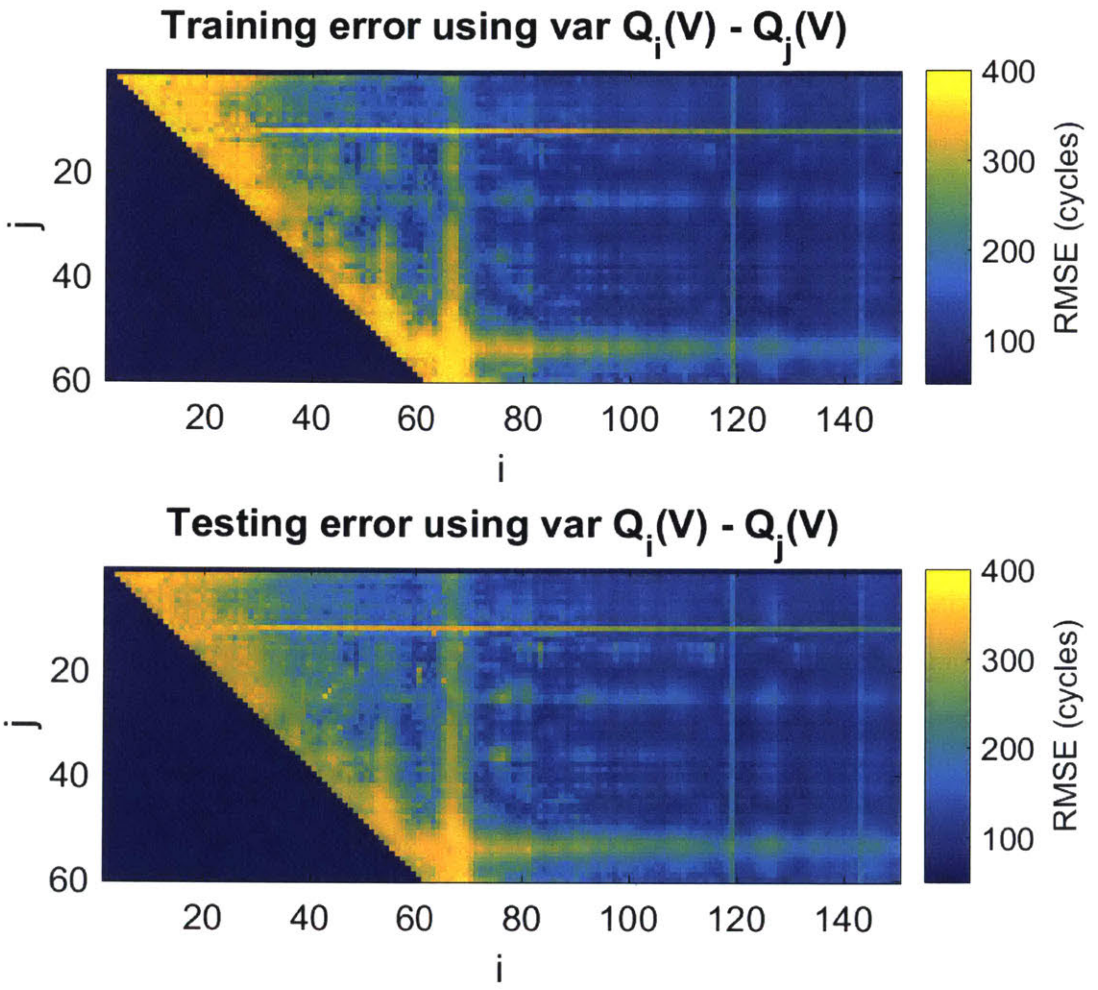


Figure 8-6: Discharge capacity curves for batteries with periodic slow charging. A slow charging protocol is employed at cycle 100, resulting in an increase in discharge capacity.

chosen for ΔQ(V). Linear models using only the variance of the difference Q_i(V) - Q_j(V) were investigated and displayed in Fig. 8-7. The model is relatively insensitive to the indexing scheme after cycle 80.

Relative indexing schemes based on cycles in which a specified capacity fade was achieved were also investigated. In the relative indexing paradigm, indices are chosen based on the relative capacity decrease. There are three primary choices for the baseline capacity: the nominal capacity of the cell reported by the manufacturer, the initial capacity of the cell, or the maximum capacity of the cell. The nominal capacity of the cells used in analysis is 1.1 Ah. Many cells never reached this level, meaning it is not a useful baseline. Most of the cells experience an initial increase in capacity, which if used for scaling, delays the point at which the first decrease is observed. This leave the maximum of the relative capacity benchmark as the best option.

Two possible indexing schemes using the capacity scaled to its maximum value were investigated. In the first scheme, a fixed number of cycles after the maximum is achieved was used. The results of this procedure are shown in Fig. 8-8. The errors of the resulting models do decrease in a similar pattern to Fig. 8-7, however the improvements in predictive power take longer to develop and do not go as low as observed in the fixed indexing scheme.



Training error using var $Q_i(V) - Q_j(V)$

Testing error using var $Q_i(V) - Q_j(V)$

Figure 8-7: RMSE error, in cycles, for training and testing datasets using only the log variance of $\Delta Q(V)$, where the discharge cycles that are used in analysis are varied. These errors are averaged over 20 random partitions of the data into equal training and testing datasets. The errors are relatively flat after cycle 80, suggesting a minimum of 80 cycles are needed before the $\Delta Q(V)$ features can be used for prediction.

The second indexing scheme considers choosing each of the indices based on when a particular relative capacity fade is achieved. Fig. 8-9 shows an example of the scaled capacity curves as well as the selection of the cycle corresponding to a relative capacity of 0.995. For the dataset, more than 250 cycles have passed before 99.5% capacity fade is reached for all of the cells. Therefore, immediately this type of indexing scheme delays when predictions can be made. The errors for the resulting models are shown in Fig. 8-10. The colorbar is set such that it matches Fig. 8-7. Immediately it can be seen that the relative indexing scheme models have higher error than the fixed indexing scheme models.

Initially, this result may seem surprising, however the relative indexing scheme has the effect of collapsing the trend that differentiates the cells by rescaling. It is therefore unsurprising that fixed indexing schemes are better suited to the prediction task. Relative indexing schemes did not result in improved predictions. Furthermore, because the discharge capacity initially increases, specified decreases in capacity take longer to develop in terms of cycles than fixed indexing.

As noted above, the model errors in Fig. 8-7 are relatively flat after cycle 80. We hypothesize that the insensitivity of the model to the indexing scheme implies linear degradation with respect to cycle number. This trend is further validated by the model coefficients shown in Fig. 8-11. Linear degradation with respect to cycle number is assumed for the LAM modes in the work of Dubarry *et al.* [60] and Anséan *et al.* [7].

There are some challenges to using $\Delta Q(V)$ with this dataset. The increases in error around cycles $j = 55$ and $i = 70$ are due to temperature fluctuations of the chamber (see Fig. 8-12 for information on experimental temperature). Temperature does have an effect on $\Delta Q(V)$, but the relationship is non-obvious. Occasional data handling issues, which appear as stripes in Fig. 8-7, also occur due to glitches in the potentiostat's database, but these types of concerns would apply to any data-driven method.

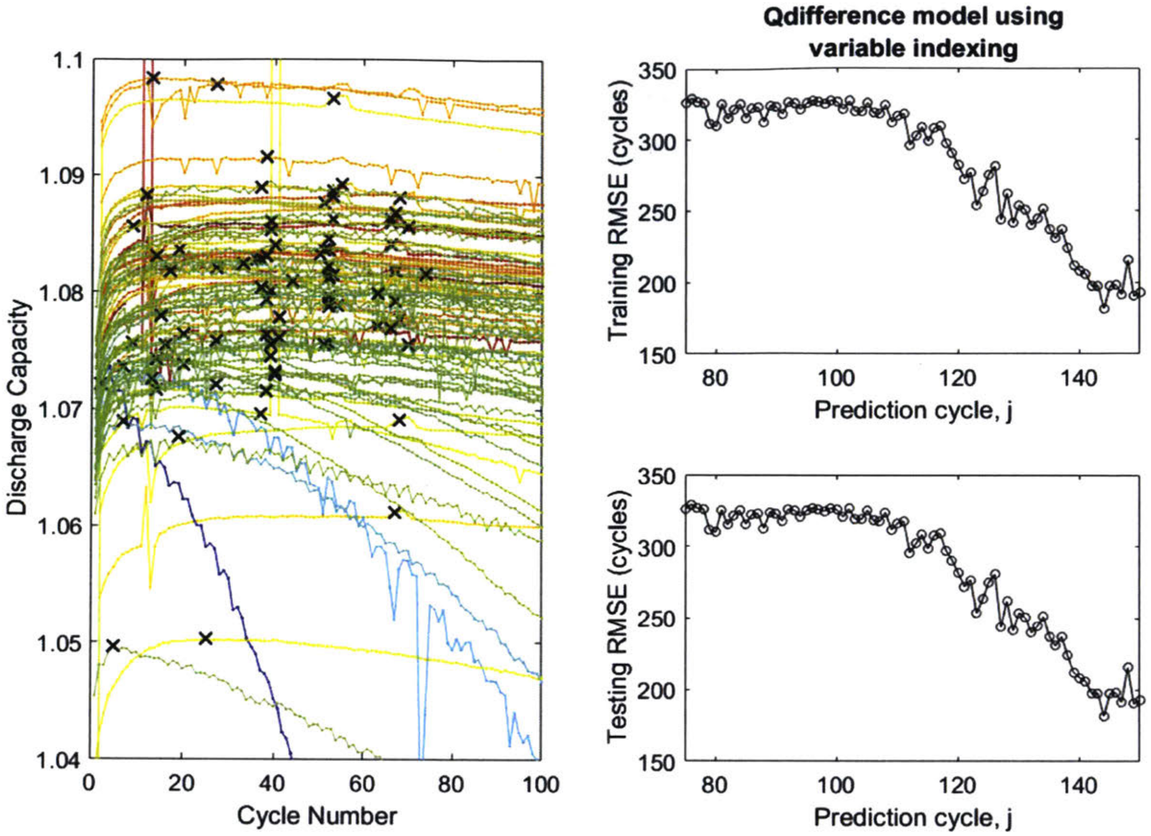


Figure 8-8: Results of an alternate indexing scheme for the ΔQ features. The early index is determined based on the maximum achieved capacity index, h_i, indicated for each battery, i by a black x in the left plot. The ΔQ is then calculated $Q_k - Q_{h_i}$ where $k = j + h_i - \max_i h_i$ such that each ΔQ uses the same number of elapsed cycles. The model uses the variance of ΔQ.

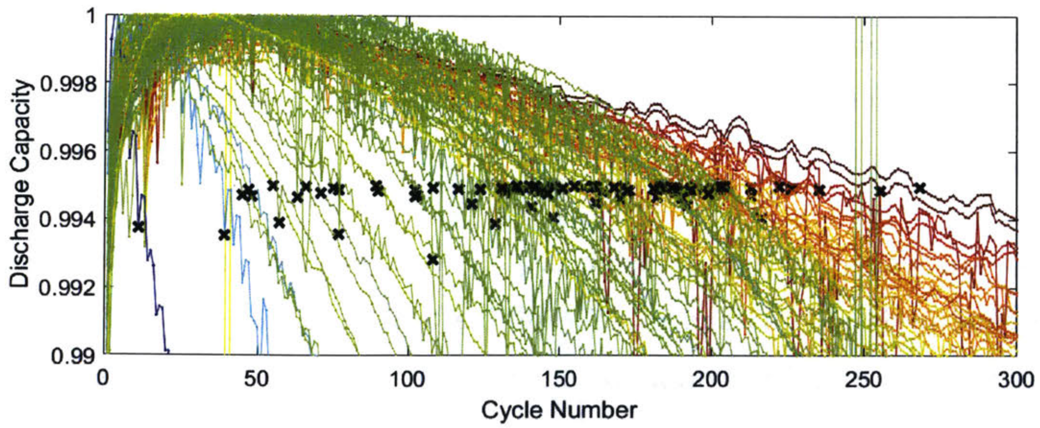


Figure 8-9: Example of the selection of indices for applying the ΔQ(V) features using the relative discharge capacity curves. Each discharge capacity is scaled by the maximum discharge capacity value (shown in Fig. 8-8). The xâĂŹs indicate the cycle corresponding to a relative discharge capacity of 0.995.

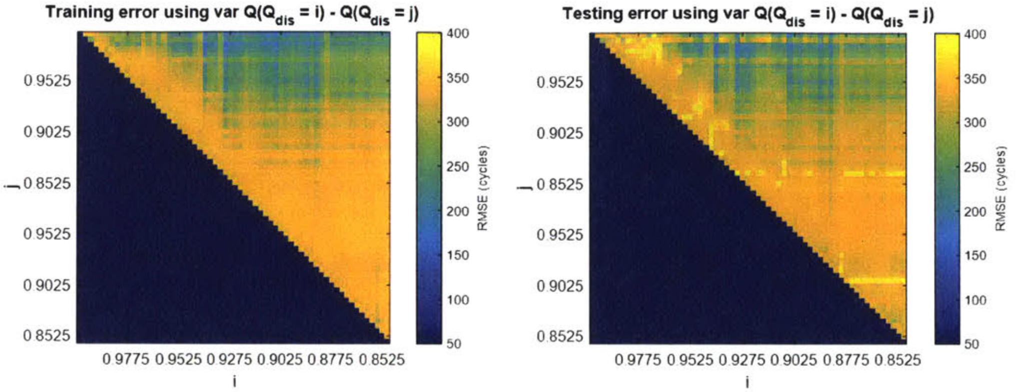


Figure 8-10: Results of an alternate indexing scheme for the ΔQ(V) features. The indices are based on when the discharge capacity reaches a relative capacity fade. Relative capacity is determined by dividing the discharge capacity trajectory by the maximum capacity achieved by the battery. The model uses the variance of ΔQ(V). RMSE values greater than 400 are thresholded to improve readability.

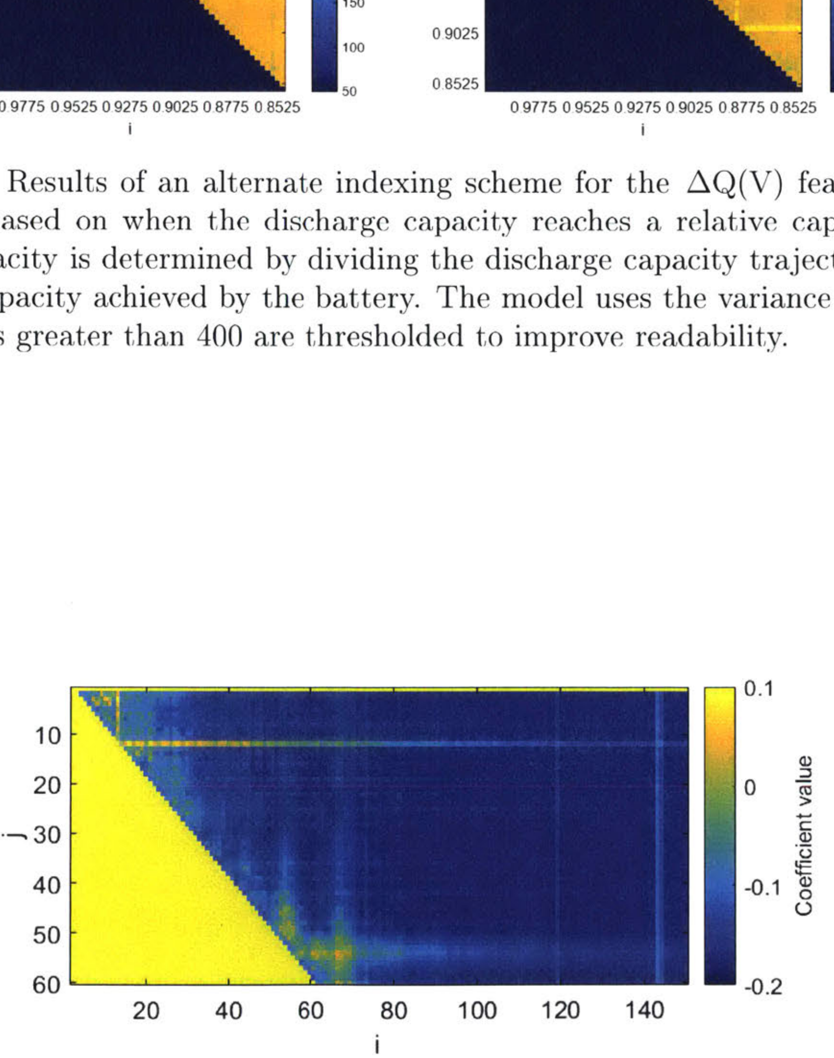


Figure 8-11: Value of the coefficients corresponding to the results in Fig. 8-7. The model is $\hat{y}_k = w \times x_k + b$ where $\hat{y}_k$ is the predicted cycle life for battery k, x_k is the ΔQ(V) feature for battery k, w is the coefficient and b is an offset term.

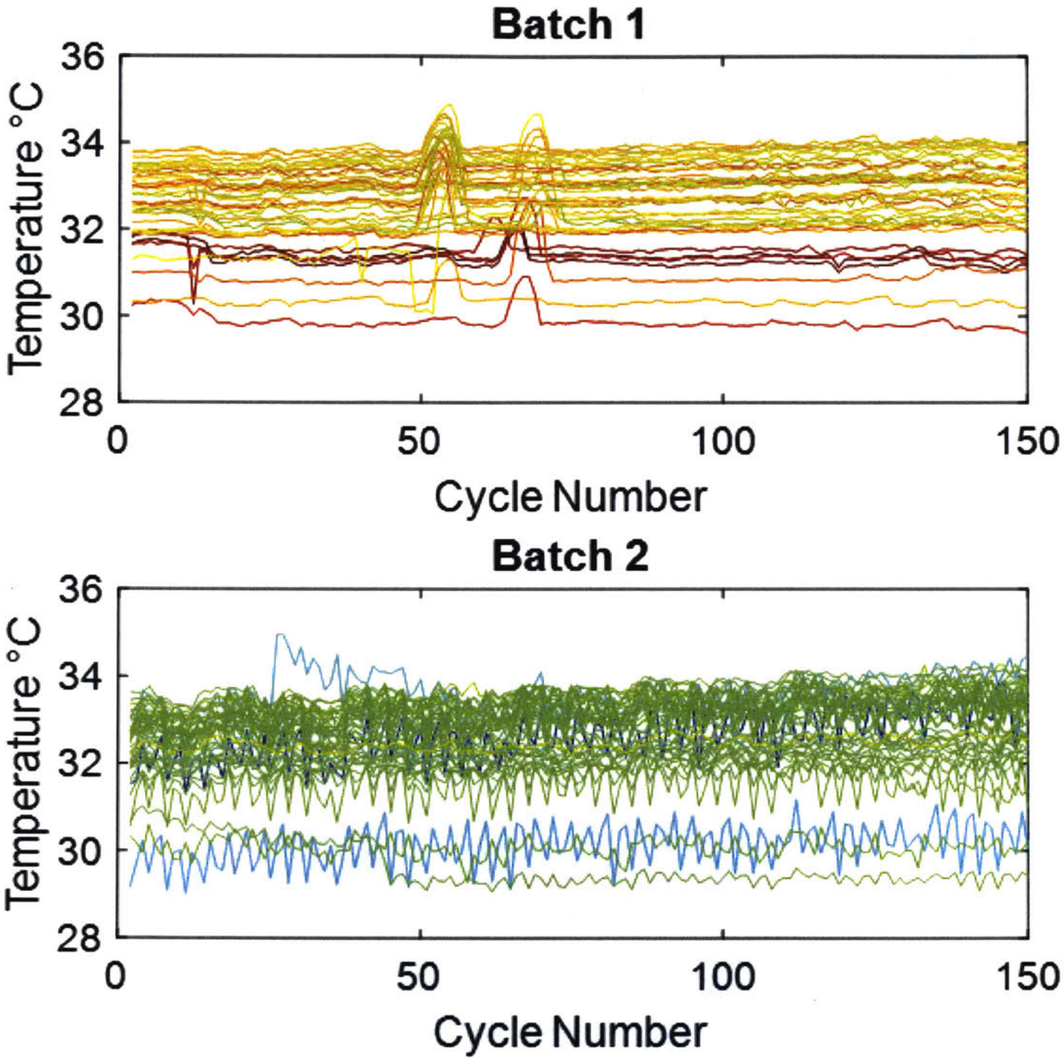


Figure 8-12: The average temperature for each of the batteries over the first 150 cycles. The spike in temperature observed in batch 1 corresponds to the decrease in performance observed in Fig. 8-7.

8.6 Conclusion

Data-driven modeling has an important role to play in diagnostics and prognostics of lithium-ion batteries. In this work, large, varied datasets enabled greater insight into the variety of observed behaviors. We presented early prediction models using fast (15 minute) discharge data for cycle life of commercial LFP|graphite batteries. The models have errors of 5-15% using data from the first 100 cycles for batteries with lifetimes ranging from 150 to 2300 cycles. This prediction is done without performing slow diagnostic cycles. The value of using the full discharge curve as opposed to only the capacity was demonstrated through the use of ΔQ(V) for quantitative prediction of cycle life. The success of the model is rationalized by demonstrating consistency with LAMdeNE and LLI as the main degradation modes, which are not reflected in trends of Q(n). This method is most valuable for any degradation mode that does not immediately manifest in Q(n) but does impact the V(Q) relationship. This work illustrates the potential of data-driven modeling for accelerating the development and deployment of lithium-ion batteries, a critical technology for renewable energy applications.

Chapter 9

Conclusions

This thesis presents methods for applying machine learning techniques to chemical and biological engineering applications. Particular challenges of these applications were addressed, namely, small datasets, an interest in an interpretable model, and incorporation of physical system knowledge. Four case studies as well as tutorials and surveys were reported.

Chapter 3 was an application to the biopharmaceutical industry. The challenges were a small dataset and an interest in an interpretable model. Often in the biopharmaceutical industry, PCA or PLS models are used, which leverage a combination of all input data. Here, elastic net models were used instead to choose a subset of variables, resulting in a simpler, and therefore more interpretable model. The issue of overfitting small datasets was also addressed. For a majority of the outputs, the elastic net models had lower errors.

Chapter 5 presented a new method for learning a sparse (i.e. interpretable) model using high-dimensional datasets where some data may be missing. Missing data is a common challenge when working with real data. It is argued that, instead of a two step approach where data are first imputed and then used for modeling, both tasks can be handled simultaneously by employing expectation-maximization (EM). The particular model assumptions were based on domain expertise. Gene and protein datasets were studied. Because of the expected correlations amongst genes and proteins that participate in the same pathway, it is argued that the structured covariance, i.e. a covariance that lies on a lower-dimensional linear manifold, is a reasonable assumption. Furthermore, it is argued that the differential expression of only a few genes/proteins differentiates the two classes, which motivates a subset. Using these assumptions, the EM steps are derived. The resulting procedure is applied to three datasets and shown to outperform competing techniques. In future work, it would be of interest to use the factor model instead of PPCA for applications with heterogeneous data.

Chapter 7 demonstrated the usefulness of semi-supervised approaches for anomaly detection. Anomaly detection problems are typically framed as unsupervised. Here, it is shown that superior performance can be achieved by labeling a small number of nominal samples. The approach is evaluated using production oil and gas well data. Because of the heterogeneity of the dataset and expected shifts over the course of the well's operation, a feature-based approach was proposed. This technique avoids the need to pre-normalize the data. The model uses these features in a Neyman-Pearson test and a kernel density estimate is used to estimate the distribution. Interpreting this class of models is not straightforward. Therefore, a model approximation is proposed to rewrite the model in terms of feature contributions. This tool supports diagnosis of the root cause. For the test case, the model predicts all three well events.

Chapter 8 presented a novel feature transformation to be used in the cycle life prediction for lithium-ion batteries. A major challenge in cycle life prediction is the observed variability, even when accounting for chemistry and operating conditions. The proposed feature subtracts discharge curves from early and late operation and is able to characterize the battery's performance over time. Models using this feature, as well as other features that incorporate temperature and charge time information, are demonstrated on real data from a high-throughput cycler.

Machine learning methods have an important role to play in complex engineering systems. Machine learning methods are particularly well suited to situations where there is either not enough information to build a first-principles model and/or a large amount of parametric uncertainty because of their ability to make limited assumptions and/or generate probabilistic predictions. This thesis provides insight on how to build and adapt methods for chemical and biological engineering.

Bibliography

[1] S. Aggarwal. What's fueling the biotech engine - 2012 to 2013. *Nature Biotechnology*, 32:32–39, 2014.

[2] H Akaike. Canonical correlations analysis of time series and the use of an information criterion. In *Advances and Case Studies in System Identification*, pages 27–96. Academic Press, New York, 1976.

[3] C. Alcala and S. J. Qin. Reconstruction-based contribution for process monitoring. In *Proceedings of the IFAC World Congress*, pages 7889–7894, 2008.

[4] C. F. Alcala and S. J. Qin. Analysis and generalization of fault diagnosis methods for process monitoring. *Journal of Process Control*, 21(3):322–330, 2011.

[5] D. T. Andrews and P. D. Wentzell. Applications of maximum likelihood principal component analysis. *Analytica Chimica Acta*, 350:341–352, 1997.

[6] D. Anseán, M. Dubarry, A. Devie, B. Y. Liaw, V. M. García, and J. C. Viera. Operando lithium plating quantification and early detection of a commercial LiFePO4 cell cycles under dynamic driving schedule. *Journal of Power Sources*, 356:36–46, 2017.

[7] D. Anseán, M. Dubarry, A. Devie, B. Y. Liaw, V. M. García, J. C. Viera, and M. González. Fast charging technique for high power LiFePO4 batteries: A mechanistic analysis of aging. *Journal of Power Sources*, 321:201–209, 2016.

[8] P. Arora, R. E. White, and M. Doyle. Capacity fade mechanisms and side reactions in lithium-ion batteries. *Journal of the Electrochemical Society*, 145:3647–3667, 1998.

[9] L. M. R. Baccarini, V. V. Rocha E Silva, B. R. De Menezes, and W. M. Caminhas. SVM practical industrial application for mechanical faults diagnostic. *Expert Systems with Applications*, 38(6):6980–6984, 2011.

[10] F. R. Bach. Bolasso: Model consistent lasso estimation through the bootstrap. In *International Conference on Machine Learning*, pages 33–40, 2008.

[11] T. C. Bach, S. F. Schuster, E. Fleder, J. Müller, M. J. Brand, H. Lorrmann, A. Jossen, and G. Sextl. Nonlinear aging of cylindrical lithium-ion cells linked to heterogeneous compression. *Journal of Energy Storage*, 5:212–223, 2016.

[12] B.R. Bakshi and G. Stephanopoulos. Representation of process trends – IV. Induction of real-time patterns from operating data for diagnosis and supervisory control. *Computers & Chemical Engineering*, 18(4):303–332, 1994.

[13] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occuring in the statistical analysis of probabalistic functions in Markov chains. *The Annals of Mathematical Statistics*, 41:164–171, 1970.

[14] T. Baumhöfer, M. Brühl, S. Rothgang, and D. U. Sauer. Production caused variation in capacity aging trend and correlation to initial cell performance. *Journal of Power Sources*, 247:332–338, 2014.

[15] C. M. Bishop. Variational principal components. In *International Conference on Artificial Neural Networks*, pages 509–514, 1999.

[16] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, New York, 2007.

[17] A. Björck and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27:579–594, 1973.

[18] G. Blanchard, G. Lee, and C Scott. Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11:2973–3009, 2010.

[19] J. Blanchet and M. Vignes. A model-based approach to gene clustering with missing observation reconstruction in a Markov random field framework. *Journal of Computational Biology*, 16:475–486, 2009.

[20] I. Bloom, J. Christophersen, and K. Gering. Differential voltage analyses of high-power lithium-ion cells: 2. Applications. *Journal of Power Sources*, 139:304–313, 2005.

[21] I. Bloom, B. W. Cole, J. J. Sohn, S. A. Jones, E. G. Polzin, V. S. Battaglia, G. L. Henriksen, C. Motloch, R. Richardson, T. Unkelhaeuser, D. Ingersoll, and H. L. Case. An accelerated calendar and cycle life study of Li-ion cells. *Journal of Power Sources*, 101:238–247, 2001.

[22] I. Bloom, A. N. Jansen, D. P. Abraham, J. Knuth, S. A. Jones, V. S. Battaglia, and G. L. Henriksen. Differential voltage analyses of high-power lithium-ion cells: 1. Technique and application. *Journal of Power Sources*, 139:295–303, 2005.

[23] T. H. Bø, B. Dysvik, and I. Jonassen. LSimpute: Accurate estimation of missing values in microarray data with least squares methods. *Nucleic Acids Research*, 32:e34, 2004.

[24] D. Bonné, M. A. Alvarez, and S. B. Jorgensen. Data driven modeling for monitoring and control of industrial fed-batch cultivations. *Industrial & Engineering Chemistry Research*, 53:7365–7381, 2013.

[25] M. Borutzky. Bond graph modelling and simulation of multidisciplinary systems - An introduction. *Simulation Modelling Practice and Theory*, pages 3–21, 2009.

[26] R. Boubour, C. Jard, A. Aghasaryan, E. Fabre, and A. Benveniste. A Petri net approach to fault detection and diagnosis in distributed systems. Part I: Application to telecommunication networks, motivations and modelling. In *Proceedings of the IEEE Conference on Decision and Control*, pages 720–725, 1997.

[27] G. N. Brock, J. R. Shaffer, R. E. Blakesley, M. J. Lotz, and G. C. Tseng. Which missing value imputation method to use in expression profiles: A comparative study and two selection schemes. *BMC Bioinformatics*, 9:12, 2008.

[28] M. Broussely, S. Herreyre, P. Biensan, P. Kasztejna, K. Nechev, and R. J. Staniewicz. Aging mechanism in Li ion cells and calendar life predictions. *Journal of Power Sources*, 97-98:13–21, 2001.

[29] A. J. Burnham, R. Viveros, and J. F. MacGregor. Frameworks for latent variable multivariate regression. *Journal of Chemometrics*, 10:31–45, 1996.

[30] J. C. Burns, G. Jain, A. J. Smith, K. W. Eberman, E. Scott, J. P. Gardner, and J. R. Dahn. Evaluation of effects of additives in wound Li-ion cells through high precision coulomtry. *Journal of the Electrochemical Society*, 158:A255–A261, 2011.

[31] J. C. Burns, A. Kassam, N. N. Sinha, L. E. Downie, L. Solnickova, B. M. Way, and J. R. Dahn. Predicting and extending the lifetime of Li-ion batteries. *Journal of the Electrochemical Society*, 160:A1451–A1456, 2013.

[32] M. P. Cabasino, A. Giua, and C. Seatzu. Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*, 46:1531–1539, 2010.

[33] J. F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal of Optimization*, 20:1956–1982, 2010.

[34] S. L. Campbell and R. Nikoukhah. *Auxiliary Signal Design for Failure Detection.* Princeton University Press, New Jersey, 2004.

[35] R. B. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1:245–276, 1966.

[36] L. Cerulo, C. Elkan, and M. Ceccarelli. Learning gene regulatory networks from only positive and unlabeled data. *BMC Bioinformatics*, 11:228, 2010.

[37] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41:15:1 – 15:58, 2009.

[38] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised Learning.* MIT Press, Cambridge, 2006.

[39] S. Charaniya, W. S. Hu, and G. Karypis. Mining bioprocess data: Opportunities and challenges. *Trends in Biotechnology*, 26(12):690–699, 2008.

[40] C. H. Chen, J. Liu, and K. Amine. Symmetric cell approach and impedance spectroscopy of high power lithium-ion batteries. *Journal of Power Sources*, 96:321–328, 2001.

[41] J. Chen and R. J. Patton. *Robust Model-based Fault Diagnosis for Dynamic Systems.* Springer, Boston, 1999.

[42] Z. Chen, D. Lovett, and J. Morris. Process analytical technologies and real time process control a review of some spectroscopic issues and challenges. *Journal of Process Control*, 21:1467–1482, 2011.

[43] G. A. Cherry and S. J. Qin. Multiblock principal component analysis based on a combined index for semiconductor fault detection and diagnosis. *IEEE Transactions on Semiconductor Manufacturing*, 19(2):159–172, 2006.

[44] L. H. Chiang and R. D. Braatz. Process monitoring using causal map and multivariate statistics: Fault detection and identification. *Chemometrics and Intelligent Laboratory Systems*, 65:159–178, 2003.

[45] L. H. Chiang, B. Jiang, X. Zhu, D. Huang, and R. D. Braatz. Diagnosis of multiple and unknown faults using the causal map and multivariate statistics. *Journal of Process Control*, 28:27–39, 2015.

[46] L. H. Chiang, E. L. Russell, and R. D. Braatz. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares and principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 50:240–252, 2000.

[47] L. H. Chiang, E. L. Russell, and R. D. Braatz. *Fault Detection and Diagnosis in Industrial Systems.* Springer, London, 2001.

[48] S. W. Choi, C. Lee, J. M. Lee, J. H. Park, and I. B. Lee. Fault detection and identification of nonlinear processes based on kernel PCA. *Chemometrics and Intelligent Laboratory Systems*, 75:55–67, 2005.

[49] J. Christensen and J. Newman. Effect of anode film resistance on the charge/discharge capacity of a lithium-ion battery. *Journal of the Electrochemical Society*, 150:A1416–A1420, 2003.

[50] J. Christensen and J. Newman. Cycable lithium and capacity loss in li-ion cells. *Journal of the Electrochemical Society*, 152:A818–A829, 2005.

[51] A. Christoffersson. *The One Component Model with Incomplete Data*. PhD thesis, Uppsala University, 1970.

[52] L. Clemmensen, T. Hastie, D. Witten, and B. Ersbøll. Sparse discriminant analysis. *Technometrics*, 53:406–413, 2011.

[53] Andrea Cordoba-Arenas, S. Onori, Y. Guezennec, and G. Rizzoni. Capacity and power fade cycle-life model for plug-in hybrid electric vehicle lithium-ion battery cells containing blended spinel and layered-oxide positive electrodes. *Journal of Power Sources*, 278:476–483, 2015.

[54] M. F. Da Silva, K. M. Muradov, and D. R. Davies. Review, analysis and comparison of intelligent well monitoring systems. In *SPE Intelligent Energy International*, pages 1–20, 2012.

[55] databricks. Apache Spark, 2016.

[56] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39:1–38, 1977.

[57] D. L. Donoho and M. Gavish. The optimal hard threshold for singular values is $4/\sqrt{3}$. Technical report, Stanford University, 2013.

[58] J.J. Downs and E. F. Vogel. A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17:245–255, 1993.

[59] M. Dubarry, V. Svoboda, R. Hwu, and B. Y. Liaw. Incremental capacity analysis and close-to-equilibrium OCV measurements to quantify capacity fade in commercial rechargeable lithium batteries. *Electrochemical and Solid-State Letters*, 9:A454–A457, 2006.

[60] M. Dubarry, C. Truchot, and B. Y. Liaw. Synthesize battery degredation modes via a diagnostic and prognostic model. *Journal of Power Sources*, 219:204–216, 2012.

[61] M. Dubarry, C. Truchot, and B. Y. Liaw. Cell degradation in commercial LiFePO4 cells with high-power and high-energy designs. *Journal of Power Sources*, 258:408–419, 2014.

[62] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973.

[63] S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97:77–87, 2002.

[64] R. Dunia, S. J. Qin, T. F. Edgar, and T. J. McAvoy. Identification of faulty sensors using principal component analysis. *AIChE Journal*, 42:2797–2812, 1996.

[65] M. Ecker, J. B. Gerschler, J. Vogel, S. Kabitz, F. Hust, P. Dechent, and D. U. Sauer. Development of a lifetime prediction model for lithium-ion batteries based on extended accerlated aging test data. *Journal of Power Sources*, 215:248–257, 2012.

[66] M. Ecker, N. Nieto, S. Käbitz, J. Schmalstieg, H. Blanke, A. Warnecke, and D. U. Sauer. Calendar and cycle life study of Li(NiMnCo)O2-based 18650 lithium-ion batteries. *Journal of Power Sources*, 248:839–851, 2014.

[67] B. Efron, T. Hastie, I. Johnstone, and Tibshirani R. Least angle regression. *The Annals of Statistics*, 32:407–451, 2004.

[68] C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 213–220, 2008.

[69] V. Etacheri, R. Maron, R. Elazari, G. Slitra, and D. Aurbach. Challenges in the development of advanced Li-ion batteries: A review. *Energy & Environmental Science*, 4:3243–3262, 2011.

[70] M. Everingham, L. Van Gool, C. Williams, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2005.

[71] P. J. Feebstra, P. J. Mosterman, G. Biswas, and P. C. Breedveld. Bond graph modeling procedures for fault detection and isolation of complex flow processes. *Simulation Series*, 33:77–84, 2001.

[72] M. A. T. Figueiredo. Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1150–1159, 2003.

[73] D. P. Finegan, E. Tudisco, M. Scheel, J. B. Robinson, O. O. Taiwo, D. S. Eastwood, P. D. Lee, M. Di Michiel, B. Bay, S. A. Hall, G. Hinds, D. J. L. Brett, and P. R. Shearing. Quantifying bulk electrode strain and material displacement within lithium batteries via high-speed operando tomography and digital volume correlation. *Advanced Science*, 3:150332, 2016.

[74] P. M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, 7(6):403–424, 1997.

[75] R. Fujimaki, T. Yairi, and K. Machida. An approach to spacecraft anomaly detection problem using kernel feature space. In *ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 401–410, 2005.

[76] B. D. Fulcher and N. S. Jones. Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26:3026–3037, 2014.

[77] B. D. Fulcher, M. A. Little, and N. S. Jones. Highly comparative time-series analysis: The empirical structure of time series and their methods. *Journal of the Royal Society Interface*, 10:20130048, 2013.

[78] P. J. García-Laencina, J. L. Sancho-Gómez, and A. R. Figueiras-Vidal. Pattern classification with missing data: A review. *Neural Computing and Applications*, 19:263–282, 2010.

[79] Z. Ge and Z. Song. Semisupervised Bayesian methods for soft sensor modeling with unlabeled data samples. *AIChE Journal*, 57:2109–2119, 2011.

[80] Z. Ge, S. Zhong, and Y. Zhang. Semisupervised kernel learning for FDA model and its application for fault classification in industrial processes. *IEEE Transactions on Industrial Informatics*, 12:1403–1411, 2016.

[81] P. Geladi and B. R. Kowalski. Partial least-squares regression: A tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.

[82] W. E. Gent, L. Kipil, Y. Liang, Q. Li, T. Barnes, S.-J. Ahn, K. H. Stone, M. McIntire, J. Hong, J. H. Song, Yiyang Li, A. Mehta, S. Ermon, T. Tyliszcak, D. Vine, J.-H. Park, S.-K. Doo, M. F. Toney, W. Yang, D. Prendergast, and W. C. Chueh. Couling between oxygen and redox and cation migration explains unusual electrochemistry in lithium-rich layered oxides. *Nature Communications*, 8:2091, 2017.

[83] J. J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Mercel Dekker, New York, 1998.

[84] K. Goh and A. Barabasi. Burstiness and memory in complex systems. *Europhysics Letters*, 81:1–5, 2008.

[85] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[86] J. B. Goodenough and Y. Kim. Challenges for rechargeable Li batteries. *Chemistry of Materials*, 22:587–603, 2010.

[87] J. B. Goodenough and K.-S. Park. The Li-ion rechargeable battery: A perspective. *Journal of the American Chemical Society*, 135:1167–1176, 2013.

[88] B. Grung and R. Manne. Missing values in principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 42:125–139, 1998.

[89] J. C. Gunther, J. S. Conner, and D. E. Seborg. Fault detection and diagnosis in an industrial fed-batch cell culture process. *Biotechnology Progress*, 23(4):851–857, 2007.

[90] S. Hajizadeh, Z. Li, R. P. B. J. Dollevoet, and D. M. J. Tax. Evaluating classification performance with only positive and unlabeled samples. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 233–242. Springer, Berlin, 2014.

[91] S. J. Harris, D. J. Harris, and C. Li. Failure statistics for commercial lithium ion batteries: A study of 24 pouch cells. *Journal of Power Sources*, 342:589–597, 2017.

[92] S. J. Harris and P. Lu. Effects of inhomogeneities - nanoscale to mesoscale - on the durability of Li-ion batteries. *The Journal of Physical Chemistry C*, 117:6481–6492, 2013.

[93] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2nd edition, 2013.

[94] C. Higuera, K. J. Gardiner, and K. J. Cios. Self-organizing feature maps identify proteins critical to learning in a mouse model of Down syndrome. *PLOS One*, 10:e0129126, 2015.

[95] D. M. Himmelblau. *Fault Detection and Diagnosis in Chemical and Petrochemical Processes.* Elsevier, New York, 1978.

[96] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[97] M. S. Hong, K. A. Severson, M. Jiang, A. E. Lu, J. C. Love, and R. D. Braatz. Challenges and opportunities in biopharmaceutical manufacturing and control. In *International Conference on Chemical Process Control*, 2017.

[98] J. L. Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30:179–185, 1965.

[99] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[100] C. Hu, G. Jain, P. Tamirisa, and T. Gorka. Method for estimating the capacity and predicting remaining useful of lithium-ion battery. *Applied Energy*, 126:182–189, 2014.

[101] I. Hwang, S. Kim, Y. Kim, and C. E. Seah. A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, 18(3):636–653, 2010.

[102] R. J. Hyndman, E. Wang, and N. Laptev. Large-scale unusual time series detection. In *IEEE International Conference on Data Mining Workshop*, pages 1616–1619, 2015.

[103] IBM. The four v's of big data, 2016.

[104] ICH. Pharmaceutical Development Q8(R2), 2009.

[105] A. Ilin and T. Raiko. Practical approaches to principal component analysis in the presence of missing values. *Journal of Machine Learning Reserach*, 11:1957–2000, 2010.

[106] Imarc. Global biopharmaceutical market report and forecast, 2012.

[107] S. A. Imtiaz and S. L. Shah. Treatment of missing values in process data analysis. *The Canadian Journal of Chemical Engineering*, 86:838–858, 2008.

[108] R. Isermann. Model-based fault-detection and diagnosis – Status and applications. *Annual Reviews in Control*, 29:71–85, 2005.

[109] R. Isermann and P. Ballé. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5:709–719, 1997.

[110] J. E. Jackson and G. S. Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21:341–349, 1979.

[111] A. Jain, K. A. Persson, and G. Ceder. Research update: The materials genome intiative: Data sharing and the impact of collaborative ab initio databases. *APL Materials*, 4:053102, 2016.

[112] B. Jiang, D. Huang, X. Zhu, F. Yang, and R. D. Braatz. Canonical variate analysis-based contributions for fault identification. *Journal of Process Control*, 26:17–25, 2015.

[113] B. Jiang, X. Zhu, D. Huang, and R. D. Braatz. Canonical variate analysis-based monitoring of process correlation structure using casual feature representation. *Journal of Process Control*, 52:109–116, 2015.

[114] B. Jiang, X. Zhu, D. Huang, J. A. Paulson, and R. D. Braatz. A combined canonical variate analysis and Fisher discriminant analysis (CVA–FDA) approach for fault diagnosis. *Computers & Chemical Engineering*, 77:1–9, 2015.

[115] J. Jiang and X. Yu. Fault-tolerant control systems: A comparative study between active and passive approaches. *Annual Reviews in Control*, 36(1):60–72, 2012.

[116] M. Jiang, Z. Zhu, E. Jimenez, C. D. Papageorgiou, J. Waetzig, A. Hardy, M. Langston, and R. D. Braatz. Continuous-flow tubular crystallization in slugs spontaneously induced by hydrodyanmics. *Crystal Growth & Design*, 14:851–860, 2014.

[117] J. Jin and J. Shi. Automatic feature extraction of waveform signals for in-process diagnositic performance improvement. *Journal of Intelligent Manufacturing*, 12:257–268, 2001.

[118] I. T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2nd edition, 2002.

[119] B. C. Juricek, D. E. Seborg, and W. E. Larimore. Identification of the Tennessee Eastman challenge process with subspace methods. *Control Engineering Practice*, 9(12):1337–1351, 2001.

[120] B. C. Juricek, D. E. Seborg, and W. E. Larimore. Fault detection using canonical variate analysis. *Industrial & Engineering Chemistry Research*, 43:458–474, 2004.

[121] H. Kim, G. H. Golub, and H. Park. Missing value estimation for DNA microarray gene expression data: Local least squares imputation. *Bioinformatics*, 21:187–198, 2004.

[122] H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23:1495–1502, 2007.

[123] A. O. Kirdar, J. S. Conner, J. Baclaski, and A. S. Rathore. Application of multivariate analysis toward biotech processes: Case study of a cell-culture unit operation. *Biotechnology Progress*, 23(1):61–67, 2007.

[124] J. Korbicz, J. M. Kośielny, Z. Kowalczuk, and W. Cholewa, editors. *Fault Diagnosis: Models, Artificial Intelligence, Applications*. Springer-Verlag, Berlin, 2004.

[125] J. M. Kościelny and Z. M. Łabęda-Grudziak. Double fault distinguishability in linear systems. *International Journal of Applied Mathematics and Computer Science*, 23:395–406, 2013.

[126] T. Kourti. The Process Analytical Technology initiative and multivariate process analysis, monitoring and control. *Analytical and Bioanalytical Chemistry*, 384:1043–1048, 2006.

[127] J. V. Kresta, J. F. MacGregor, and T. E. Marlin. Multivariate statistical process monitoring of process operating performance. *Canadian Journal of Chemical Engineering*, 69:35–47, 1991.

[128] W. Ku, R. H. Storer, and C. Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30:179–196, 1995.

[129] R. Lakerveld, B. Benyahia, R. D. Braatz, and P. I. Barton. Model-based design of a plant-wide control strategy for a continuous pharmaceutical plant. *AIChE Journal*, 59:3671–3685, 2013.

[130] W. E. Larimore. System identification, reduced-order filtering and modeling via canonical variate analysis. In *Proceedings of the American Control Conference*, pages 445–451, 1983.

[131] W.E. Larimore. Canonical variate analysis in identification, filtering, and adaptive control. In *Proceedings of the IEEE Conference on Decision and Control*, pages 596–604, 1990.

[132] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[133] Y. LeCun, C. Cortes, and C.J.C. Burges. The MNIST database of handwritten digits, 1998. Available for download at http://yann.lecun.com/exdb/mnist/, retrieved on May 22, 2015.

[134] J. H. Lee and A. W. Dorsey. Monitoring of batch processes through state-space models. *AIChE Journal*, 50:1198–1210, 2004.

[135] M. Lewerenz, A. Marongiu, A. Warnecke, and D. U. Sauer. Differential voltage analysis as a tool for analyzing inhomogeneous aging: A case study for LiFePO4|graphite cylindrical cells. *Journal of Power Sources*, 368:57–67, 2017.

[136] M. Lewerenz, J. Münnix, J. Schmalstieg, S. Käbitz, M. Knips, and D. U. Sauer. Systematic aging of commercial LiFePO4|graphite cylindrical cells including a theory explaining rise of capactiy during aging. *Journal of Power Sources*, 345:254–263, 2017.

[137] W. Li, Q. Guo, and C. Elkan. A positive and unlabeled learning algorithm for one-class classification of remote-sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 49:717–725, 2011.

[138] Y. Li, S. Meyer, J. Lim, S. C. Lee, W. E. Gent, S. Marchesini, H. Krishnan, T. Tyliszcak, D. Shapiro, A. L. D. Kilcoyne, and W. C. Chueh. Effects of particle size, electronic connectivity, and incoherent nanoscale domains on the sequence of lithiation in LiFePO4 porous electrodes. *Advanced Materials*, 27:6591–6597, 2015.

[139] B. Y. Liaw, R. G. Jungst, G. Nagasubramanian, H. L. Case, and D. H. Doughty. Modeling capacity fade in lithium-ion cells. *Journal of Power Sources*, 140:157–161, 2005.

[140] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low rank representation. In *Advances in Neural Information Processing Systems*, pages 612–620, 2011.

[141] R. J. A. Little and D. B. Rubin. *Statisical Analysis with Missing Data.* John Wiley & Sons, New Jersey, 2nd edition, 2002.

[142] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu. Building text classifiers using positive and unlabeled examples. In *ICDM Data Mining*, pages 179–186, 2003.

[143] D. Liu, J. Pang, J. Zhou, Y. Peng, and M. Pecht. Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process function regression. *Microelectronics Reliability*, 53:832–839, 2013.

[144] J. Liu. On-line soft sensor for polyethylene process with multiple production grades. *Control Engineering Practice*, 15:769–778, 2007.

[145] P. Liu, J. Wang, J. Hicks-Garner, E. Sherman, S. Soukiazian, M. Verbrugge, H. Tataria, J. Musser, and P. Finamore. Aging mechanisms of LiFePO4 batteries deduced by electrochemical and structural analyses. *Journal of the Electrochemical Society*, 157:A499–A507, 2010.

[146] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[147] C. B. Low, D. Wang, S. Arogeti, and M. Luo. Quantitative hybrid bond graph-based fault detection and isolation. *IEEE Transactions on Automation Science and Engineering*, 7:558–569, 2010.

[148] Scale-Up Systems Ltd. DynoChem, 2017.

[149] A. E. Lu, J. A. Paulson, N. J. Mozdzierz, A. Stockdale, F. N. Ford Versypt, K. R. Love, J. C. Love, and R. D. Braatz. Control systems technology in the advanced manufacturing of biologic drugs. In *Proceedings of the IEEE Conference on Control Applications*, pages 1505–1515, 2015.

[150] J. Lu, T. Wu, and K. Amine. State-of-the-art characterization techniques for advanced lithium-ion batteries. *Nature Energy*, 2:17011, 2017.

[151] P. R. Lyman and C. Georgakis. Plant-wide control of the Tennessee Eastman problem. *Computers & Chemical Engineering*, 19:321–331, 1995.

[152] J. MacGregor and A. Cinar. Monitoring, fault diagnosis, fault-tolerant control and optimization: Data driven methods. *Computers & Chemical Engineering*, 47:111–120, 2012.

[153] J. F. MacGregor and T. Kourti. Statistical process control of multivariate processes. *Control Engineering Practice*, 3:403–414, 1995.

[154] S. Mahadevan and S. L. Shah. Fault detection and diagnosis in process data using one-class support vector machines. *Journal of Process Control*, 19(10):1627–1639, 2009.

[155] M. Maki, J. Jiang, and K. Hagino. A stability guaranteed active fault-tolerant control system against actuator failures. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1893–1898, 2001.

[156] B. M. Marlin. *Missing Data Problems in Machine Learning.* PhD thesis, University of Toronto, 2008.

[157] G. R. Marseglia, J. K. Scott, L. Magni, R. D. Braatz, and D. M. Raimondo. A hybrid stochastic-deterministic approach for active fault diagnosis using scenario optimization. In *Proceedings of the IFAC World Congress*, pages 1102–1107, 2014.

[158] J. Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6:7–12, 1960.

[159] R.K. Mehra and J. Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7(5):637–640, 1971.

[160] N. Meinshausen. Relaxed lasso. *Computational Statistics & Data Analysis*, 52:374–393, 2007.

[161] S. M. Mercier, B. Diepenbroek, M. C. F. Dalm, and R. H. Wijffels. Multivariate data analysis as a PAT tool for early bioprocess development data. *Journal of Biotechnology*, 167:262–270, 2013.

[162] A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz. Active fault diagnosis for nonlinear systems with probabilistic uncertainties. In *Proceedings of the IFAC World Congress*, pages 7079–7084, 2014.

[163] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Assocation*, 44:335–341, 1949.

[164] I. Monroy, R. Benitez, G. Escudero, and M. Graells. A semi-supervised approach to fault diagnosis for chemical processes. *Computers & Chemical Engineering*, 34:631–642, 2010.

[165] D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers.* John Wiley & Sons, Hoboken, NJ, 5th edition, 2011.

[166] M. Mrugalski. An unscented Kalman filter in designing dynamic GMDH neural networks for robust fault detection. *International Journal of Applied Mathematics and Computer Science*, 23:157–169, 2013.

[167] M. Mrugalski. *Advanced Neural Network-Based Computational Schemes for Robust Fault Diagnosis.* Springer, Cham, 2014.

[168] T. Murata. Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE*, volume 77, pages 541–580, 1989.

[169] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, 2012.

[170] R. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Springer, Netherlands, 1998.

[171] P. R. C. Nelson, P. A. Taylor, and J. F. MacGregor. Missing data methods in PCA and PLS: Score calculations with incomplete observations. *Chemometrics and Intelligent Laboratory Systems*, 35:45–65, 1996.

[172] R. Nikoukhah. Guaranteed active failure detection and isolation for linear dynamical systems. *Automatica*, 34(11):1345–1358, 1998.

[173] P. Nomikos and J. F. MacGregor. Monitoring batch processes using multiway principal component analysis. *AIChE Journal*, 40:1361–1375, 1994.

[174] P. Nomikos and J. F. MacGregor. Multivariate SPC charts for monitoring batch processes. *Technometrics*, 37:41–59, 1995.

[175] B. Nykvist and M. Nilsson. Rapidly falling costs of battery packs for electric vehicles. *Nature Climate Change*, 5:329–332, 2015.

[176] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19:2088–2096, 2003.

[177] U. S. Department of Health & Human Services. Guidance for industry, process validation: General principles and practices, 2011.

[178] O. O'Malley et al. Apache Hadoop, 2016.

[179] M. Ouyang, W. J. Welsh, and P. Georgopoulos. Gaussian mixture clustering and imputation of microarray data. *Bioinformatics*, 20:917–923, 2004.

[180] M. R. Palacín and A. de Guibert. Why do batteries fail? *Science*, 351:1253292, 2016.

[181] S. Pampuri, A. Schirru, G. Fazio, and G. De Nicolao. Multilevel lasso applied to virtual metrology in semiconductor manufacturing. In *IEEE International Conference on Automation Science and Engineering*, pages 244–249, 2001.

[182] T. Park and G. Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103:681–686, 2008.

[183] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[184] R. J. Patton. Fault-tolerant control systems: The 1997 situation. *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, 3:1033–1054, 1997.

[185] D. B. Paul and J. M. Baker. The design for the Wall Street Journal-based CSR corpus. In *Proceedings of the Workshop on Speech and Natural Language*, pages 357–362, 1992.

[186] S. Paul, C. Diegelmann, H. Kabza, and T. Wener. Analysis of ageing inhomogeneities in lithium-ion battery systems. *Journal of Power Sources*, 239, 2013.

[187] H.M. Paynter. *Analysis and Design of Engineering Systems.* MIT Press, Cambridge, MA, 1961.

[188] K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, 2:559–572, 1901.

[189] C. A. Petri. *Kommunikation mit automaten.* PhD thesis, Bonn: Institut für Instrumentelle Mathematik, 1962.

[190] J. Pieracci, L. Perry, and L. Conley. Using partition designs to enhance purification process understanding. *Biotechnology and Bioengineering*, 107:814–824, 2010.

[191] M. B. Pinson and M. Z. Bazant. Theory of SEI formation in rechargeable batteries: Capacity fade, accelerated aging and lifetime prediction. *Journal of the ELectrochemical Society*, 160:A243–A250, 2013.

[192] J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, Cambridge, MA, 1998.

[193] S. J. Qin. Statistical process monitoring: Basics and beyond. *Journal of Chemometrics*, 17:480–502, 2003.

[194] S. J. Qin. Process data analytics in the era of Big Data. *AIChE Journal*, 60:3092–3100, 2014.

[195] D. M. Raimondo, R. D. Braatz, and J. K. Scott. Active fault diagnosis using moving horizon input. In *Proceedings of the European Control Conference*, pages 3131–3136, 2013.

[196] D. M. Raimondo, G. R. Marseglia, R. D. Braatz, and J. K. Scott. Fault-tolerant model predictive control with active fault isolation. In *Proceedings of Conference on Control and Fault-Tolerant Systems*, pages 444–449, 2013.

[197] P. Ramadass, B. Haran, R. E. White, and B. N. Popov. Mathematical modeling of the capacity fade of Li-ion cells. *Journal of Power Sources*, 123:230–240, 2003.

[198] V. Ramadesigan, K. Chen, N. A. Burns, V. Boovaragavan, R. D. Braatz, and V. R. Subramanian. Parameter estimation and capacity fade analysis of lithium-ion batteries using reformulated models. *Journal of the Electrochemical Society*, 158:A1048–A1054, 2011.

[199] O. Ramilo, W. Allman, W. Chung, A. Mejias, M. Ardura, C. Claser, K.M. Wittkowski, B. Piqueras, J. Banchereau, A.K. Palucka, and D. Chaussabel. Gene expression patterns in blood leukocytes discriminate patients with acute infections. *Blood*, 109:2066–2077, 2007.

[200] M. Rashid and A. Gupta. Effect of relaxation periods over cycling performance of a Li-ion battery. *Journal of the Electrochemical Society*, 162:A3145–A3153, 2015.

[201] M. Rasmussen and R. Bro. A tutorial on the lasso approach to sparse modeling. *Chemometrics and Intelligent Laboratory Systems*, 119:21–31, 2012.

[202] A. S. Rathore, N. Bhushan, and S. Hadpe. Chemometrics applications in biotech processes: A review. *Biotechnology Progress*, 27:307–315, 2011.

[203] E. Read, J. T. Park, R. Shah, B. S. Riley, K. A. Brorson, and A. S. Rathore. Process analytical technology (PAT) for biopharmaceutical products: Part II concepts and applications. *Biotechnology and Bioengineering*, 104:276–284, 2010.

[204] E. Read, R. Shah, B. S. Riley, J. Park, K. A. Brorson, and A. S. Rathore. Process analytical technology (PAT) for biopharmaceutical products: Part I concepts and applications. *Biotechnology and Bioengineering*, 105:285–295, 2010.

[205] M. S. Reis, R. D. Braatz, and L. H. Chiang. Big data challenges and future research directions. *Chemical Engineering Progress*, 112(3):46–50, 2016.

[206] M. S. Reis and P. M. Saraiva. Heteroscedastic latent variable modelling with applications to multivariate statistical process control. *Chemometrics and Intelligent Laboratory Systems*, 80:57–66, 2006.

[207] Y. Roggo, P. Chalus, L. Maurer, C. Lema-Martinez, A. Edmond, and N. Jent. A review of near infrared spectroscopy and chemometrics in pharmaceutical technologies. *Journal of Pharmaceutical and Biomedical Analysis*, 44:683–700, 2007.

[208] S. Roweis. EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems*, pages 626–632, 1998.

[209] D. B. Rubin. Inference and missing data. *Biometrika*, 63:581–592, 1976.

[210] E. L. Russell, L. H. Chiang, and R. D. Braatz. Tennessee Eastman Problem Simulation Data. http://web.mit.edu/braatzgroup/links.html, 1998. accessed on April 12 2017.

[211] E. L. Russell, L. H. Chiang, and R. D. Braatz. *Data-driven Techniques for Fault Detection and Diagnosis in Chemical Processes*. Springer, New York, 2000.

[212] E. L. Russell, L. H. Chiang, and R. D. Braatz. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 51:81–93, 2000.

[213] M. Safari and C. Delacourt. Aging of a commercial graphite/LiFePO4 cell. *Journal of the Electrochemical Society*, 158:A1123–A1135, 2011.

[214] M. Safari, M. Morcrette, A. Teyssot, and C. Delacourt. Life-prediction methods for lithium-ion batteries derived from a fatigue approach I. Introduction: Capactiy-loss prediction based on damage accumulation. *Journal of the Electrochemical Society*, 157:A713–A720, 2010.

[215] R. Salakhutdinov and S. Roweis. Adaptive overrelaxed bound optimization methods. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 664–671, 2003.

[216] E. Sarasketa-Zabala, F. Aguesse, I. Villarreal, L. M. Rodriguez-Martinez, C. M. López, and P. Kubiak. Understanding lithium inventory loss and sudden performance fade in cylindrical cells during cycling with deep-discharge steps. *The Journal of Physical Chemistry C*, 119:896–906, 2015.

[217] J. L. Schafer. Multiple imputation: A primer. *Statistical Methods in Medical Research*, 8:3–15, 1999.

[218] O. Schmidt, A. Hawkes, Gambhir A., and I. Staffell. The future cost of electrical energy storage based on experience rates. *Nature Energy*, 2:17110, 2017.

[219] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

[220] S. F. Schuster, T. Bach, E. Fleder, J. Müller, M. Brand, G. Sextl, and Jossen A. Nonlinear aging characteristics of lithium-ion cells under different operational conditions. *Journal of Energy Storage*, 1:44–53, 2015.

[221] S. F. Schuster, Martin J. Brand, P. Berg, M. Cleissenberger, and A. Jossen. Lithium-ion cell-to-cell variation during battery electric vehicle operation. *Journal of Power Sources*, 297:242–251, 2015.

[222] J. K. Scott, R. Findeisen, R. D. Braatz, and D. M. Raimondo. Design of active inputs for set-based fault diagnosis. In *Proceedings of American Control Conference*, pages 3561–3566, 2013.

[223] J. K. Scott, R. Findeisen, R. D. Braatz, and D. M. Raimondo. Input design for guaranteed fault diagnosis using zonotopes. *Automatica*, 50:1580–1589, 2014.

[224] J. K. Scott, G. R. Marseglia, L. Magni, R. D. Braatz, and D. M. Raimondo. A hybrid stochastic-deterministic input design method for active fault diagnosis. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5656–5661, 2013.

[225] S. R. Searle. *Matrix Algebra Useful for Statistics.* John Wiley & Sones, 1982.

[226] M. S. B. Sehgal, I. Gondal, and L. S. Dooley. Collateral missing value imputation: A new robust missing value estimation algorithm for microarry data. *Bioinformatics*, 21:2417–2423, 2005.

[227] A. D. Sendek, Q. Yang, E. D. Cubuk, K.-A. N. Duerloo, Y. Cui, and E. J. Reed. Holistic computational structure screening of more than 12000 candidates for solid lithium-ion conductor materials. *Energy & Environmental Science*, 10:306–320, 2017.

[228] K. Severson, J. G. VanAntwerp, V. Natarajan, C. Antoniou, J. Thömmes, and R. D. Braatz. Elastic net with Monte Carlo sampling for data-based modeling in biopharmaceutical manufacturing facilities. *Computers & Chemical Engineering*, 80:30–36, 2015.

[229] K. A. Severson and R. D. Braatz. The data analytics triangle. In *AIChE Spring National Meeting*, page 480093, 2017.

[230] H. J. Shin, D.H. Eom, and S.S. Kim. One-class support vector machines – An application in machine fault detection and classification. *Computers & Industrial Engineering*, 48:395–408, 2005.

[231] A. Shukla and J. Thömmes. Recent advances in large-scale production of monoclonal antibodies and related proteins. *Trends in Biotechnology*, 28:253–261, 2010.

[232] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, New York, 1986.

[233] S. Simani. Residual generator fuzzy identification for automotive diesel engine fault diagnosis. *International Journal of Applied Mathematics and Computer Science*, 23:419–438, 2013.

[234] S. Simani, C. F. Fantuzzi, and R. J. Patton. *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques.* Springer, New York, 2003.

[235] A. Simoglou, E. B. Martin, and A. J. Morris. Statistical performance monitoring of dynamic multivariate processes using state space modelling. *Computers & Chemical Engineering*, 26(6):909–920, 2002.

[236] S. Sjostrand, L. H. Clemmensen, R. Larsen, and B. Ersboll. SpaSM: A Matlab Toolbox for Sparse Statistical Modeling. http://www2.imm.dtu.dk/projects/spasm/, 2004–2016.

[237] A. J. Smith, J. C. Burns, and J. R. Dahn. High-precision differential capacity analysis of LiMn2O4/graphite cells. 14:A39–A41, 2011.

[238] A. J. Smith, N. N. Sinha, and J. R. Dahn. Narrow range cycling and storage of commercial Li ion cells. *Journal of the Electrochemical Society*, 160:A235–A242, 2013.

[239] R. Spotnitz. Simulation of capacity fade in lithium-ion batteries. *Journal of Power Sources*, 113:72–80, 2003.

[240] V. S. Srivinvasan and M. A. Jafari. Fault detection/monitoring using time Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 23:1155–1162, 1993.

[241] D. M. J. Tax. *One-class Classification.* PhD thesis, Delft University of Technology, 2001.

[242] D. M. J. Tax and R. P. W. Duin. Suport vector domain description. *Pattern Recognition Letters*, 20:1191–1199, 1999.

[243] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54:45–66, 2004.

[244] M. M. Thackeray, Sun-Ho Kang, C. S. Johnson, J. T. Vaughey, R. Benedek, and S. A. Hackney. Li2MnO3-stabilized LiMO2 (M = Mn, Ni, Co) electrodes for lithium-ion batteries. *Journal of Materials Chemistry*, 17:3112–3125, 2007.

[245] E. V. Thomas, I. Bloom, J. P. Christophersen, and V. S. Battaglia. Statistical methodology for predicting the life of lithium-ion cells via accelerated degradation testing. *Journal of Power Sources*, 184:312–317, 2008.

[246] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58:267–288, 1996.

[247] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99:6567–6572, 2002.

[248] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. Technical report, Aston University, 1997.

[249] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61:611–622, 1999.

[250] T. Togkalidou, R. D. Braatz, B. K. Johnson, O. Davidson, and A. Andrews. Experimental design and inferential modeling in pharmaceutical crystallization. *AIChE Journal*, 47:160–168, 2001.

[251] T. Tomohiro and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15:116–132, 1985.

[252] U. Tröltzsch, O. Kanoun, and H.-R. Tränkler. Characterizing aging effects of lithium-ion batteries by impedance spectroscopy. *Electrochimica Acta*, 51:1664–1672, 2006.

[253] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17:520–525, 2001.

[254] E. Tziampazis and A. Sambanis. Modeling of cell culture processes. *Cytotechnology*, 14:191–204, 1994.

[255] C. Undey, S. Ertunç, and A. Çinar. Online batch/fed-batch process performance monitoring, quality prediction, and variable-contribution analysis for diagnosis. *Industrial & Engineering Chemical Research*, 42:4645–4658, 2003.

[256] U.S. Department of Health and Human Services. Guidance for industry, process validation: General principles and practices, 2011.

[257] P. Van Overshcee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30:75–93, 1994.

[258] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri. A review of process fault detection and diagnosis. Part II: Qualitative models and search strategies. *Computers & Chemical Engineering*, 27:313–326, 2003.

[259] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin. A review of process fault detection and diagnosis. Part III: Process history based methods. *Computers & Chemical Engineering*, 27:327–346, 2003.

[260] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. Kavuri. A review of process fault detection and diagnosis. Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 27:293–311, 2003.

[261] M. Verhaegen. Identification of the deterministic part of MIMO state space models given in innovation form from input-output data. *Automatica*, 30:61–74, 1993.

[262] M. Verhaegen. Subspace model identification. Part III: Analysis of the ordinary output-error state space model identification algorithm. *International Journal of Control*, 58:555–586, 1993.

[263] M. Verhaegen. Application of a subspace model identification technique to identify LTI systems operating in closed loop. *International Journal of Control*, 29:1027–1040, 1994.

[264] M. Verhaegen and P. Dewilde. Subspace model identification. Part I: The output-error state space model identification class of algorithms. *International Journal of Control*, 56:1187–1210, 1992.

[265] M. Verhaegen and P. Dewilde. Subspace model identification. Part II: Analysis of the elementary output-error state space model identification algorithm. *International Journal of Control*, 56:1211–1241, 1992.

[266] P. Verma, P. Maire, and Novák. A review of the features and analyses of the solid electrolyte interface in Li-ion batteries. *Electrochimica Acta*, 55:6332–6341, 2010.

[267] S. Verron, T. Tiplica, and A. Kobi. Fault detection and identification with a new feature selection based on mutual information. *Journal of Process Control*, 18:479–490, 2008.

[268] J. Vetter, P. Novak, M. R. Wagner, C. Veit, K.-C. Moller, J. O. Besenhard, M. Winter, M. Wohlfahrt-Mehrens, C. Vogler, and A. Hammouche. Ageing mechanisms in lithium-ion batteries. *Journal of Power Sources*, 147:269–281, 2005.

[269] N. Viswanadham. Fault detection and diagnosis of automated manufacturing systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2301–2306, 1988.

[270] B. Walczak and D. Massart. Dealing with missing data: Part I. *Chemometrics and Intelligent Laboratory Systems*, 58:29–42, 2001.

[271] T. Waldmann, S. Gorse, T. Samtleben, G. Schneider, V. Knoblauch, and M. Wohlfahrt-Mehrens. A mechanical aging mechanism in lithium-ion batteries. *Journal of Electrochemical Society*, 161:A1742–A1747, 2014.

[272] J. Wang, P. Liu, J. Hicks-Garner, E. Sherman, S. Soukiazian, M. Verbrugge, H. Tataria, J. Musser, and P. Finamore. Cycle-life model for graphite-LiFePO4 cells. *Journal of Power Sources*, 196:3942–3948, 2011.

[273] S. Wang and J. Zhu. Improved centroids estimation for the nearest strunken centroid classifier. *Bioinformatics*, 23:972–979, 2007.

[274] X. Wang, A. Li, Z. Jiang, and H. Feng. Missing value estimation for DNA microarray gene expression data by support vector regression imputation and orthogonal coding scheme. *BMC Bioinformatics*, 7:32, 2006.

[275] G. Ward, T. Hastie, S. Barry, J. Elith, and J. R. Leathwick. Presence-only data and the EM algorithm. *Biometrics*, 65:554–563, 2009.

[276] P. A. Wedin. On angles between subspaces of a finite dimensional inner product space. In B. Kagstrom and A. Ruhe, editors, *Matrix Pencils, Lecture Notes in Mathematics 973*, pages 263–285. Springer, 1983.

[277] P.D. Wentzell, D. T. Andrews, D. C. Hamilton, K. Faber, and B. R. Kowalski. Maximum likelihood principal component analysis. *Journal of Chemometrics*, 11:339–366, 1997.

[278] M. S. Whittingham. Ultimate limits to intercalation reactions for lithium batteries. *Chemical Reivews*, 114:11414–11443, 2014.

[279] A. Widodo and B.-S. Yang. Support vector machines in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21:2560–2574, 2007.

[280] A. S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611, 1976.

[281] B. M. Wise, N. L. Ricker, D. F. Veltkamp, and B. R. Kowalski. A theoretical basis for the use of principal component models for monitoring multivariate processes. *Process Control and Quality*, 1:41–51, 1990.

[282] M. Witczak. *Fault Diagnosis and Fault-Tolerant Control Strategies for Non-Linear Systems: Analytical and Soft Computing Approaches*. Springer, Cham, 2014.

[283] D. M. Witten and R. Tibshirani. Penalized classification using Fisher's linear discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73:753–772, 2011.

[284] H. Wold. Path models with latent variables: The NIPALS approach. In H. M. Blalock, A. Aganbegian, F. M. Borodkin, R. Boudon, and V. Capecchi, editors, *Quantitative Sociology: International Perspectives on Mathematical and Statistical Modeling*, pages 307–357. Academic Press, New York, 1975.

[285] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 3:37–52, 1987.

[286] S. Wold, A. Ruhe, H. Wold, and W.J. Dunn III. The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3):735–743, 1984.

[287] L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. *Association for the Advancement of Artificial Intelligence*, 5:904–910, 2005.

[288] Z. Yan, C. Huang, and Y. Yao. Semi-supervised mixture discriminant monitoring for chemical batch processes. *Chemometrics and Intelligent Laboratory Systems*, 134:10–22, 2014.

[289] X.-G. Yang, Y. Leng, S. Ge, and C.-Y. Wang. Modeling of lithium plating induced aging of lithium-ion batteries: Transition from linear to nonlinear aging. *Journal of Power Sources*, 360:28–40, 2017.

[290] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *Journal of Process Control*, 22:1567–1581, 2012.

[291] A. Ypma, D. M. J. Tax, and R. P. W. Duin. Robust machine fault detection with independent component analysis and support vector data description. In *Neural Networks for Signal Processing*, pages 67–76, 1999.

[292] H. Yu and J. F. MacGregor. Multivariate image analysis and regression for prediction of coating content and distribution in the production of snack foods. *Chemometrics and Intelligent Laboratory Systems*, 67:125–144, 2003.

[293] L. Yu, R. R. Snapp, T. Ruiz, and M. Radermacher. Probabilistic principal component analysis with expectation maximization (PPCA-EM) facilitates volume classification and estimates the missing data. *Journal of Structural Biology*, 171:18–30, 2012.

[294] H. H. Yue and S. J. Qin. Reconstruction-based fault identification using a combined index. *Industrial & Engineering Chemistry Research*, 40:4403–4414, 2001.

[295] Q. Zhang and R. E. White. Capacity fade analysis of a lithium ion cell. *Journal of Power Sources*, 179:793–798, 2008.

[296] Y. Zhang and J. Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32:229–252, 2008.

[297] A. Zhao, Y. Feng, L. Wang, and X. Tong. Neyman-Pearson classification under high-dimensional settings. *Journal of Machine Learning Research*, 17:1–39, 2016.

[298] Y. Zhao, R. Ball, J. Mosesian, J.F. de Palma, and B. Lehman. Graph-based semi-supervised learning for fault detection and classification in solar photovoltaic arrays. *IEEE Transactions on Power Electronics*, 30:2848–2858, 2015.

[299] M. Zhong, S. X. Ding, J. Lam, and H. Wang. An LMI approach to design robust fault detection filter for uncertain LTI systems. *Automatica*, 39:543–550, 2003.

[300] D. Zhou, G. Li, and S. J. Qin. Total projection to latent structures for process monitoring. *AIChE Journal*, 56(1):168–178, 2010.

[301] X. Zhu and R. D. Braatz. 2D contribution map for fault detection. *IEEE Control Systems*, 33(4):72–77, 2014.

[302] H. Zou and T. Hastie. Regularizationn and variable selection via the elastic net. *Journal of Royal Statistical Society: Series B (Statistical Methodology)*, 67:301–320, 2005.