APPLICATIONS OF SUPPORT VECTOR MACHINES TO PROCESS ENGINEERING SYSTEMS

THESIS SUBMITTED TO THE

UNIVERSITY OF PUNE

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

CHEMICAL ENGINEERING

BY

Jade Avinash Madhusudanrao

Chemical Engineering and Process Development Division

National Chemical Laboratory

Dr Homi Bhabha Road

Pune 411008

India

December 2006

राष्ट्रीय रासायनिक प्रयोगशाला
(वैज्ञानिक तथा औद्योगिक अनुसंधान परिषद)
डॉ. होमी भाभा मार्ग पुणे – 411 008. भारत
NATIONAL CHEMICAL LABORATORY
(Council of Scientific & Industrial Research)
Dr. Homi Bhabha Road, Pune - 411 008. India.

CERTIFICATE

This is to certify that the work incorporated in the thesis, "***Applications of Support Vector Machines to Process Engineering Systems*** " submitted by Mr. Jade Avinash Madhusudanrao, for the Degree of Doctor of Philosophy, was carried out by the candidate under my supervision in the Chemical Engineering and Process Development Division, National Chemical Laboratory, Pune – 411 008, India. Such material as has been obtained from other sources has been duly acknowledged in the thesis.

Dr. B. D. Kulkarni

(Research Advisor)

DECLARATION

I hereby declare that the thesis "***Applications of Support Vector Machines to Process Engineering Systems"*** submitted for the degree of Doctor of Philosophy to the University of Pune has not been submitted by me for a degree to any other University.

Jade Avinash Madhusudanrao

Dedicated to

My Beloved Parents

Acknowledgements

I take this opportunity with deep sense of gratitude to record my sincere thanks to my research supervisor Dr. B. D. Kulkarni for introducing me to a fascinating and challenging frontier of artificial intelligence and machine learning that has brought a positive turning point in my career. I remain deeply indebted to him for his precious advice, his caring attention, for helping me out in most stressful situations, for giving me the edge to meet future challenges...

My heartfelt thanks are due to Dr. V. K. Jayaraman for his keen interest, valuable suggestions, personal care and for helping me in all possible ways to comprehend my work in its present form. I will be grateful to him also for giving me exposure to the exciting field of time series analysis.

Colleagues and Staff at National Chemical Laboratory (NCL), Pune were quite supportive throughout my stay. Particularly I am indebted to Prakash, Abhijeet, Nilesh, Sudheerkumar, Kaustubh, Kalpendra, Yogesh, Rakesh, Onkar for making my stay at NCL pleasant and memorable. Technical discussions with Abhijeet and Prakash were really useful.

I am thankful to the teachers at Bharat Vidyalaya and Adarsh Mahavidyalaya, Omerga (Dist. Osmanabad) for orienting my mind towards research during the schooldays itself. I am also grateful to Dr. T.N.S. Mathur, Dr. Bikas Mohanty, Dr. Surendra Kumar and Dr. I. M. Mishra at IIT, Roorkee for their valuable guidance and encouragement during my career development.

I would also like to express my sincere thanks to the numerous anonymous referees who have reviewed parts of this work prior to publication in journals and whose valuable comments have contributed to the clarification of many of the ideas presented in this thesis.

Whatever I am and whatever I will be in future is because of the goodwill and unstinted support that I have received from my parents. Their constant encouragement, sacrifice and support made me achieve the goal. I owe them a lot for which mere words are insufficient. Also I will be thankful to my loving wife and wonderful son for putting up with having a 'student' husband/dad.

I would also like to thank Council for Scientific and Industrial Research (CSIR), New Delhi for granting me Senior Research Fellowship.

Jade Avinash Madhusudanrao

Table of Contents

List of Figures

List of Tables

Abstract

The inherent non-linearity, complexity and uncertainty in chemical processes make it difficult to develop a compact mathematical model to represent the system over a wide range of governing parameters. This provides a continuing driving force to explore alternative means to achieve the objectives. Data-driven models, which do not require substantial understanding of the phenomenology involved and are more robust to presence of noise and relatively scarce measurements, are thus becoming more attractive.

In the last decade, there has been considerable growth in the development and application of artificial intelligence (AI) tools — in particular artificial neural networks — to build data-driven models for process engineering applications. Hybrid combinations of AI tools and newer algorithms are also being developed with a view to increasing robustness and prediction capabilities. In recent years, a new approach called support vector machines (SVM) has been proposed. SVM are universal feed-forward networks firmly based on the rigorous statistical learning theory developed by Vapnik. The simplicity of implementation, excellent generalization ability and remarkable performance on difficult tasks have made SVM one of the most popular tools in various disciplines. For binary classification problems, given a set of nonlinearly separable input vectors belonging to two distinct classes, SVM finds an optimal linear separating hyperplane in a high dimensional feature space. SVM use a convex quadratic optimization algorithm to find a unique globally optimal decision surface. This decision surface can be represented by a subset of training data lying on the margin. These data, known as support vectors, carry all the relevant information about the classification problem. The algorithm is rigorous, but very compact as the optimization problem and the decision surface depends only on the dot product between the training data in feature space. SVM handles the computational intractability arising out of high dimensionality of the feature space by computing the dot product of transformed data in the input space itself by employing a kernel trick. Inspired from SVM, a number of easy and elegant non-linear versions of classical linear algorithms have been developed by the use of kernel functions.

The general class of algorithms resulting from notion of implementing kernel trick is known as kernel methods or kernel machines. A family of kernel methods mainly includes support vector machines (SVM), kernel principal component analysis (kernel PCA), support vector regression (SVR), support vector domain distribution (SVDD) etc. The present thesis is devoted for the application of these kernel methods for solving various kinds of process engineering problems. In first chapter deal with the brief review on conventional methods for classification and regression and an introduction on kernel based learning algorithms.

In Chapter 2, detailed derivation of SVM for classification and regression has been explained. Chapter 2 also includes applications of SVM to fault diagnosis, a well-known pattern recognition problem in process engineering and to quantitative structure property relationships (QSPRs) problem.

In Chapter 3, kernel PCA methodology, an elegant nonlinear generalization of the linear PCA, is illustrated by considering the examples of (i) denoising chaotic time series and, (ii) prediction of properties of polymer nanocomposites. Kernel PCA captures the dominant nonlinear features of the original data by transforming it to a high dimensional feature space. An appropriately defined kernel function allows the computations to be performed in the original input space and facilitates extraction of substantially higher number of principal components enabling excellent denoising and feature extraction capabilities. In comparison to other nonlinear principal component analysis (PCA) techniques, kernel PCA requires only the solution of an eigenvalue problem and does not involve any nonlinear optimization. In addition, the number of principal components need not be specified prior to modeling. This makes the kernel PCA algorithm very attractive tool for modeling of nonlinear process engineering systems.

In chapter 4, a hybrid strategy of using (i) locally linear embedding (LLE) for nonlinear dimensionality reduction of high dimensional data and (ii) support vector domain distribution (SVDD) for classification of the resultant features, is proposed as a robust methodology for process monitoring. The method of online

abnormality detection in a process plant has been described with the two case studies viz. acetone-butanol fermentation and a benchmark semi-batch reactor problem. Illustrative examples substantiate the methodology vis-à-vis current practice.

A novel method for characterization of time series employing a unique combination of wavelet based local singularity analysis and support vector machines (SVM) classification is developed in chapter 5. The method is illustrated by considering the case example of flow regime identification in gas-liquid stirred tank equipped with Rushton turbine. Pressure fluctuations time series data obtained at different operating conditions were first analyzed to obtain the distribution of local Hölder exponents' estimates. The relevant features from this distribution were then used as input data to the SVM classifier. Employing this method we could classify flow regimes with 98% accuracy. The results highlight the fact that the local scaling behavior of a given regime follows a distinct pattern. Further, the singularity features can be employed by intelligent machine learning based algorithms like SVM for successful online regime identification. The method can be readily applied to the other multiphase systems like bubble column, fluidized bed etc.

Chapter 6 comprises a new method for selection of model parameters in prediction of time series. Apart from the conventional criterion of minimizing RMS error, the method also minimizes the error on the distribution of singularities, evaluated through the local Hölder estimates and its probability density spectrum. Predictions of two simulated and one real time series have been done using kernel principal component regression (KPCR) and model parameters of KPCR have been selected employing the proposed as well as the conventional method. Results obtained demonstrate that the proposed method takes into account the sharp changes in a time series and improves the generalization capability of the KPCR model in better prediction of the unseen test data.

In chapter 7, salient conclusions from results obtained for the case studies of chapters 2-7 are described.

Chapter 1
INTRODUCTION

1.1 Motivation

The inherent non-linearity, complexity and uncertainty in chemical processes make it difficult to develop a compact mathematical model to represent the system over a wide range of governing parameters. This provides a continuing driving force to explore alternative means to achieve the objectives. Data-driven models, which do not require substantial understanding of the phenomenology involved and are more robust to presence of noise and relatively scarce measurements, are thus becoming more attractive.

In the last decade, there has been considerable growth in the development and application of artificial intelligence (AI) tools — in particular artificial neural networks — to build data-driven models for process engineering applications. Neural networks with different architectures (such as feedforward (Hoskins et al. 1991), recurrent (Karjala & Himmelblau, 1994), and multiple network architectures (Watanabe et al. 1994)) and different basis functions (such as sigmoidal (Venkatasubramanian et al. 1990), radial (Leonard and Kramer, 1991), wavelet (Bhakshi and Stephanopoulos, 1993), and ellipsoidal basis functions (Girosi, 1992)) have been explored. These neural networks have been exploited for several types of applications including fault diagnosis (Venkatasubramanian et al 1990, Ungar et al. 1990, Hoskins et al. 1991), dimensionality reduction (Tan and Mavrovouniotes, 1995), modeling (Thompson and Kramer, 1994), data rectification (Karjala and . Himmelblau, 1994), dynamic optimization (Chen and Weigand, 1994), experimental design (Glassey et al. 1994). Hybrid combinations of AI tools and newer algorithms are also being developed with a view to increasing robustness and prediction capabilities. In recent years, a new approach called support vector machines (SVM) has been proposed. SVM are universal feed-forward networks firmly based on the rigorous statistical learning theory developed by Vapnik (1995, 1998). The simplicity of implementation, excellent generalization ability and remarkable performance on difficult tasks have made SVM one of the most popular tools in various disciplines (Burges, 1998; Christianini and Shawe-Taylor, 2000). For binary classification problems, given a

set of nonlinearly separable input vectors belonging to two distinct classes, SVM finds an optimal linear separating hyperplane in a high dimensional feature space. SVM use a convex quadratic optimization algorithm to find a unique globally optimal decision surface. This decision surface can be represented by a subset of training data lying on the margin. These data, known as support vectors, carry all the relevant information about the classification problem. The algorithm is rigorous, but very compact as the optimization problem and the decision surface depends only on the dot product between the training data in feature space. SVM handles the computational intractability arising out of high dimensionality of the feature space by computing the dot product of transformed data in the input space itself by employing a kernel trick. Inspired from SVM, a number of easy and elegant non-linear versions of classical linear algorithms have been developed by the use of kernel functions. (Muller et al. 2001).

The general class of algorithms resulting from notion of implementing kernel trick is known as kernel methods or kernel machines. A family of kernel methods mainly includes support vector machines (SVM), kernel principal component analysis (kernel PCA), support vector regression (SVR), support vector domain distribution (SVDD) etc. They utilize the techniques from optimization, statistics, and functional analysis to achieve the maximal flexibility, and performance, both in terms of generalization and in terms of computational cost. Support vector machines (SVM) and other kernel based methods differs from the conventional machine learning tools in following ways: i) They are explicitly based on a theoretical model of learning rather than on loose analogies with natural learning systems or other heuristics. ii) The formulation of these methods emerges with theoretical guarantees about their performance and has a modular design that makes it possible to separately implement and analyze its components. iii) They are not affected by the problem of local minima because their training leads to convex optimization. The simplicity of their implementation, remarkable performance on difficult tasks is attracting further attention. Hence, the present thesis is devoted for the application of these kernel methods for solving various kinds of process engineering problems. The applications considered will include fault detection/diagnosis, nonlinear modeling of chemical engineering systems, time series prediction, etc. These methods can be combined with wavelet-fractal theory for analysis, characterization and prediction of chaotic time series.

1.2 Organization of the Thesis

Subsequent sections of this chapter deals with brief review on conventional methods for classification and regression and an introduction on kernel based learning algorithms. In Chapter 2, detailed derivation of SVM for classification and regression has been explained. Chapter 2 also includes applications of SVM to fault diagnosis, a well-known pattern recognition problem in process engineering and to quantitative structure property relationships (QSPRs) problem. In Chapter 3, kernel PCA methodology, an elegant nonlinear generalization of the linear PCA, is illustrated by considering the examples of (i) denoising chaotic time series and, (ii) prediction of properties of polymer nanocomposites. In chapter 4, a hybrid strategy of using (i) locally linear embedding (LLE) for nonlinear dimensionality reduction of high dimensional data and (ii) support vector domain distribution (SVDD) for classification of the resultant features, is proposed as a robust methodology for process monitoring. The method of online abnormality detection in a process plant has been described with the two case studies viz. acetone-butanol fermentation and a benchmark semi-batch reactor problem. A novel method for characterization of time series employing a unique combination of wavelet based local singularity analysis and support vector machines (SVM) classification is developed in chapter 5. The method is illustrated by considering the case example of flow regime identification in gas-liquid stirred tank equipped with Rushton turbine. Chapter 6 comprises a new singularity distribution based method for selection of model parameters in prediction of time series. In chapter 7, salient conclusions are drawn.

1.3 Review of Conventional Classification and Regression Methods

Several conventional methods for classification and regression with their applications to process engineering are described below:

1.3.1 Classification methods

The task of pattern classification is to find a rule to assign an object to one of several classes using features of that object. Classification can be done in supervised as well as unsupervised manner. In the supervised classification the class labels of data are known, a priori, and the new object is assigned to one of

the several classes using the rules generating from the already available data (known as training data). In unsupervised classification the class labels are missing and problem is solved using clustering, density estimation and data description methods. Few important methods for supervised classification with their basic principles are described below and for details of these methods one may refer to the references cited therein.

1.3.1.1 Bayesian classifiers

The fundamental principle of a Bayesian classifier is a combination of Bayes' theorem and Bayes' rule. The practical use of Bayes' theorem is to turn probabilities that can be estimated from a training set into those required for classification. A naive Bayes classifier is a simple probabilistic classifier (Hand and Yu, 2001). It is based on probability models that incorporate strong independence assumptions which often have no bearing in reality, hence are naive. Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised manner. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood. In spite of their very simple design and apparently over-simplified assumptions, naive Bayes classifiers often work well in many complex real-world situations (Kim et al. 2000, Yamashita, 2000).

1.3.1.2 Discriminant analysis

Linear discriminant analysis (LDA) is a classification procedure in which the classes are considered to have normal distribution and equal dispersion (covariance matrix). It owes its success to its wide applicability in the most general classification problems, to its scaling invariance, and to the ready interpretability of the discriminant functions obtained. (Wu et al. 1996; Martin et al. 1999) Additionally, LDA is particularly robust and effective, even if the theoretical statistical requirements of multivariate normal distributions, equal class variance/ covariance matrices and large object/variable ratio are not fulfilled. However, discriminant functions obtained for LDA are not orthogonal to each other and their graphical projections are often unsatisfactory for checking an

effective separation of objects. Furthermore, for non-linearly separable data sets some limitation also occurs in the results of the classification.

In Quadratic discriminant analysis (QDA), it is assumed that each class has normal distribution and that dispersion is different for each class and that the hypersurfaces separating the class are therefore quadratic. (Wu et al. 1996) The use of quadratic discriminant analysis (QDA) is limited to cases where all the classes are well represented; otherwise the class covariance matrix is a near singular matrix. However, QDA does not improve LDA results, except in those cases where there is a significant departure from linear separability between the classes. A trade-off can be performed between LDA and QDA and a bias based on the training set can be properly introduced in regularized discriminant analysis (RDA) (Wu et al. 1996). However, RDA is a two parameter method which is not invariant to scaling, and the ready interpretability of the discriminant functions is lost.

1.3.1.3 Classification trees

Classification tree methods are a form of knowledge representation based on a decision tree. (Breiman et al. 1984; Mulholland et al. 1995) In a binary tree classifier a decision is made at each non-terminal node of the tree based upon the value of one of many possible attributes or features. If the feature value is less than some threshold then the left branch of the tree is taken, otherwise the right branch is taken. The leaves, or terminal nodes, of the tree represent the classes to be identified. CTs are very popular in machine learning applications because they provide a symbolic representation that lends itself to easy interpretation. The representation can also be extended or easily modified when a tree is translated into convenient If–Then rules.

1.3.1.4 Nearest neighbor

The *k*-nearest neighbor classifier is a conventional nonparametric classifier. The principle of the method is that the test object is assigned to the class according to the majority vote procedure, i.e. to the class which is most represented in the set of *k* nearest training objects (Tominaga, 1999, Wu and Massart, 1997). It can be shown that the *k*-nearest neighbor rule becomes the Bayes optimal decision rule as *k* goes to infinity. The simplest case of the k-NN method is 1NN classification.

The idea in 1NN is extremely simple: to classify **x** find its closest neighbor among the training points (say **x'**,) and assign to **x** the class of **x'**. The Euclidean distance or the Mahalanobis distance is commonly used as the measurement of similarity of two objects. It should be noted that this paradigm does not provide an explicit model of the data. Hence it is said that instead of an induction process, the nearest-neighbor method is based on a transduction process that avoids the specification of the model.

1.3.1.5 Neural networks

Artificial neural networks (ANNs) are connectionist models from the field of artificial intelligence applying non-linear analytical processes to solve pattern recognition problems. ANN consists of a network of interconnected processing units, the structure of which is based on the structure of the human brain. For classification purposes, the network builds a model based on a set of input objects (the training set) with known outputs, adjusting the weights associated with each connection so that output values as similar as possible to the real values are generated. These weights contain information about the relationships between the input variable set (inputs) and the categories studied (outputs). The most simple neural network, called perceptron, is a one-neuron classifier. By connecting perceptrons one can design a neural network structure called multilayer perceptron (MLP). In the MLP, training is achieved by minimising the square mean output error by backpropagation and using the generalised delta rule. Neural networks with different structures have been successfully applied to various fault detection and diagnosis problems in process engineering (Hoskins et al. 1991; Venkatasubramanian et al. 1990).

1.3.1.6 Logistic regression

Logistic regression function is analogous to linear regression; however, it classifies the input data into output categories, rather than generating the numeric outputs. A logistic regression model is a parametric model that specifies the probability of a dichotomous variable $Y(\{0,1\})$ to have the value 1 given the values of the features of an instance. (Hosmer & Lemeshow, 2000). The logistic model has the following form $p(Y=1\,|\,\mathbf{x}) = 1/[1+e^{\beta_0+\sum_{i=1}^{n}\beta_i x_i}]$ where **x** represents an

instance to be classified and β_i (i=0,1,…,n) denotes the coefficients of the n predictors. The logistic regression model is very simple to develop, and the independent variables do not need to be pre-processed, viz., standardized, normalized, etc. In order to avoid misinterpretations of the results it can also be used as a tool for screening.

1.3.2 Regression methods

In regression, the dependent variable has real value, instead of label as in case of classification. The task is to map the input (explanatory) variables to the output variables.

1.3.2.1 Multiple Linear Regression

The goal of multiple linear regression (MLR) is to establish a linear relationship between input (independent) and output variables as follows:

$$y = \sum_{i=1}^{n} b_i x_i + e \qquad (1)$$

the coefficients $b_{1,} \ldots \ldots, bn$ are least square estimates.

Ridge regression (RR), partial least squares (PLS), and principal components regression (PCR) are three of the more familiar alternatives to MLR when one is concerned with the problem of estimating the regression coefficients of the standard linear model in the presence of highly correlated predictor variables.

1.3.2.2 Principal component regression (PCR) and partial least squares (PLS)

Principal component analysis (PCA) and partial least squares (PLS) are the well-known chemometric techniques. PCA and PLS have been applied with a great success to a wide variety of problems in analytical chemistry, biological and medicinal chemistry and chemical engineering. These statistical methods help in denoising, dimensionality reduction, feature extraction and regression. PLS and principal component regression (PCR) found to be useful in situations when the collinearity among the variables exists. (Hoskuldsson, 1988, Wold et al. 2001). In PCR, principal components are solely determined from explanatory variables and

uncorrelated input variables are used in a regression model, whereas PLS creates orthogonal components by using the existing correlations between explanatory variables and corresponding outputs while also keeping most of the variance of explanatory variables. It is well known that PCR and PLS extracts the linear relations among the variables only. To capture the correct phase details of the data set and to identify the dominant features existing between variables, various nonlinear versions of PCA and PLS have been proposed. Thus for instance, Nonlinear PLS with inner spline and quadratic functions have been used by Wold (Wold et al., 1989; Wold, 1992). Kramer (1991) presented a method based on autoassociative neural network topology for nonlinear PCA. Dong and McAvoy (1996) combined the principal curves algorithm (Hastie & Stuezle, 1989) and the autoassociative network (Kramer, 1991) to provide an effective algorithm for nonlinear PCA. Nonlinear PCA and PLS based upon the concept of the input training network have been successfully employed for process engineering applications (Tan & Mavrouvouniotis (1995), Malthouse et al 1997).

1.3.2.3 Ridge regression

Ridge regression (RR) is a commonly used statistical technique (Hoerl and Kennard, 1970) that is based on correlations within the data rather than generating latent variables, as in the case of PLS. RR technique has a ability to overcome the type of 'ill-conditioned' situation that is when $X'X$ matrix tends to be very near to singular. The method takes its name from the fact that the procedure adds a value (θ) to the ridge or diagonal of the correlation matrix. In this way, the rank of the data matrix can be improved by exaggerating the orthoganality or unique features in the data. Thus, the coefficients are computed as follows:

$$B(\theta) = (X'X + \theta I)^{-1} X'Y \qquad (2)$$

Where θ is a positive number between 0 and 1 (the ridge constant) and I is the identity matrix. If $\theta = 0$, i.e. no value added, then the result will be the same as the least squares result (i.e. as in the case of MLR). The addition of the constant to the ridge has the effect of stabilising the coefficients obtained from regression by reducing the magnitude of the regression coefficients.

1.3.2.4 Neural networks

Neural networks as described in section 1.3.1.5 can also be used for nonlinear regression. Neural networks have been widely used for various nonlinear function approximation problems like identification, control etc. in process engineering (Thompson and Kramer, 1994).

After reviewing the classical methods for classification and regression, an introduction to statistical learning theory and kernel methods is given in the next section.

1.4 Introduction to Statistical Leaning Theory and Kernel Methods

1.4.1 Statistical learning theory

Let us start with a general idea of learning pattern classification problem. The task of classification is to find rule to assign an object to one of several classes using features of that object. In the simplest case there are two different classes. For this binary classification now the task is to estimate a function f using input-output training data pairs generated i.i.d. according to an unknown probability distribution $P(\mathbf{x},y)$

$$(\mathbf{x}_1, y_1),.....,(\mathbf{x}_m, y_m), \qquad \mathbf{x} \in \Re^n, y \in \{-1,+1\} \tag{3}$$

such that f will correctly classify the unseen examples $(\mathbf{x},y)$. An example is assigned to the class +1 if $f \geq 0$ and to the class -1 otherwise. The test data are assumed to be generated from the same distribution as training data. The best function f can be obtained by minimizing expected error (risk) (Vapnik 1995; Muller et al. 2001)

$$R[f] = \int l(f(\mathbf{x}), y) dP(\mathbf{x}, y) \tag{4}$$

where l denotes suitably chosen loss function. e.g. 0/1 loss function which is defined as

$$l(f(\mathbf{x}), y) = \begin{cases} 0 & if \quad y = f(\mathbf{x}) \\ 1 & if \quad y \neq f(\mathbf{x}) \end{cases} \tag{5}$$

The risk cannot be minimized directly as the underlying probability distribution is unknown. Therefore, one has to try to estimate a function that is close to the optimal based on the available information, i.e. the training data and properties of the function class F the solution f is chosen from.

A simple induction principle will approximate the minimum of the expected risk to the minimum of the empirical risk, (Vapnik 1995; Muller et al. 2001)

$$R_{emp}[f] = \frac{1}{m}\sum_{i=1}^{m} l(f(\mathbf{x}_i), y_i) \qquad (6)$$

The empirical risk will converge towards the expected risk asymptotically (as $n \to \infty$). However, for small sample sizes large deviations are possible and overfitting might occur. Then a small generelisation error cannot be obtained by simply minimizing the training error. One way to avoid overfitting problem is to restrict the complexity of the function class F that one chooses the function f from. Simple function that explains most of the data is preferable to the complex one. For instance, it is always possible to interpolate 5 points in the plane with a polynomial of, say, degree 25, but the resulting function may not have any predictive power. However, predictions of a linear function interpolating them are more reliable. Therefore, a regularization term can be added to limit complexity of the function class F from which the learning machine can choose the function. The problem of the selection of optimal model complexity of the function can be addressed by VC theory and structural risk minimization (SRM) principle. VC dimension h of function class F is a measure of complexity i.e. it measures how many training points can be shattered (separated) for all possible labelings using the functions of the class. The SRM principle proceeds as follows: Let $f_1,\ldots,f_k$ be the solutions of the empirical risk minimization in the function classes F_i , SRM chooses the function class F_i (and the function f_i) such that upper bound on the generalization error is minimized which can be computed as following: (Vapnik 1995; Muller et al. 2001)

$$R[f] \le \mathrm{Re}\, mp[f] + \sqrt{\frac{h(\ln(2m/h)+1) - \ln(\eta/4)}{n}} \qquad (7)$$

where η is a number lying between 0 and 1. For example if η = 0.95, the above error bound, known as the risk bound, holds with a probability of 95%. m denotes the number of training samples and h is the Vapnik-Chervonenkis dimension.

The above risk bound clearly brings out the trade-off between the structural complexity of the hypothesis space and the training error. A simple hypothesis space with a small VC dimension may lead to a high training error. On the other hand a structurally rich hypothesis function having a large VC dimension may work well on training phase but generalizes poorly on unseen test examples. The task, therefore, is to find the optimal hypothesis space with maximal generalizing capability that neither overfits nor leads to a high training error.

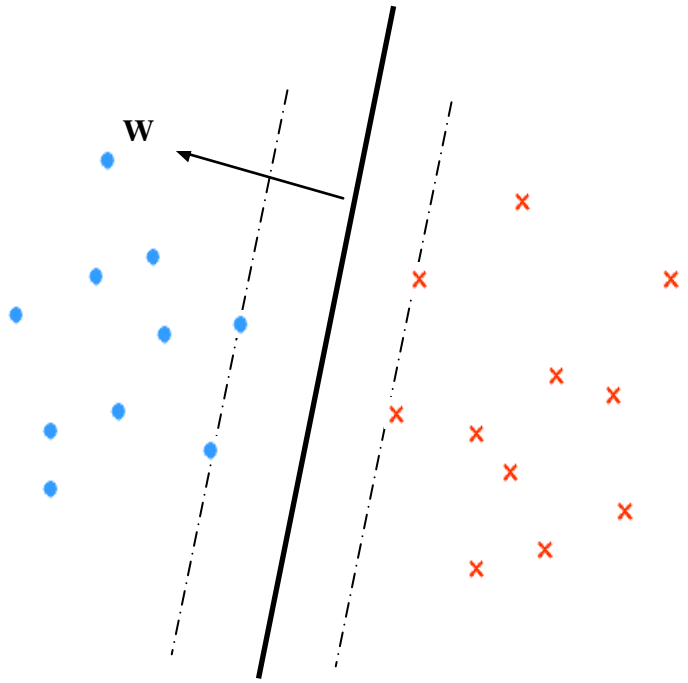


Figure 1.1: Margin of linear classifier: minimal distance between any training point to the hyperplane. (in this case, it is the distance between the dotted lines and solid line)

It was shown that for the class of hyperplanes of the form $f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b$, separating the training samples (Figure 1.1), the VC dimension itself can be bounded by another quantity known as margin. The margin is defined as minimal distance of a sample to the decision surface. The margin can be measured by the length of the weight vector **w.** The weight vector **w** and b can be rescaled such that the points closest to the hyperplanes satisfy $|(\mathbf{w} \cdot \mathbf{x}_i) + b| = 1$. Using the samples from different classes $(\mathbf{w} \cdot \mathbf{x}_1) + b = 1$ and $(\mathbf{w} \cdot \mathbf{x}_2) + b = -1$, the margin can be given by the distance between the two points, measured perpendicular to the hyperplane, i.e. $\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = \frac{2}{\|\mathbf{w}\|}$ (Vapnik 1995; Muller et al. 2001).

VC-dimension of the class of separating hyperplanes can be linked to the margin or the length of the weight vector **w** as follows:

$$h \leq R^2 A^2 + 1 \quad \text{and} \quad \|\mathbf{w}\|_2 \leq A \qquad (8)$$

where *R* is the radius of the smallest ball around the data. Thus, if we bound the margin of a function class from below, say by 2/A, its VC-dimension can be controlled. Support vector machines, which is described in detail in chapter 2, implements this principle.

As we are dealing with linear separating hyperplanes, this choice of linear functions may pose limitations i.e. we may likely underfit instead of overfitting. This problem can be tackled my mapping the data to a feature space which is nonlinearly related to the input space and then building the linear decision surface in the feature space. This is equivalent of building the nonlinear decision surface in the input space.

1.4.2 Building algorithms in feature space

Algorithms in feature space make use of the following idea: via a nonlinear mapping

$$\begin{aligned} \Phi : \mathbf{R}^n &\rightarrow F \\ \mathbf{x} &\rightarrow \Phi(\mathbf{x}) \end{aligned}$$

the data $\mathbf{x}_1, \ldots\ldots, \mathbf{x}_m \in R^N$ is mapped into potentially much richer feature space *F*.

Now, for a given learning problem one now considers the same algorithm in F instead of R^N, i.e. one works with sample

$$(\Phi(\mathbf{x}_1), y_1),.....,(\Phi(\mathbf{x}_m), y_m), \qquad \in F \times Y \qquad (9)$$

Statistical learning theory tells us that learning in feature space can be simplified by using a simple class of decision rules (e.g. linear classifiers). The variability and richness to the classifier can be accomplished by introducing the mapping Φ.

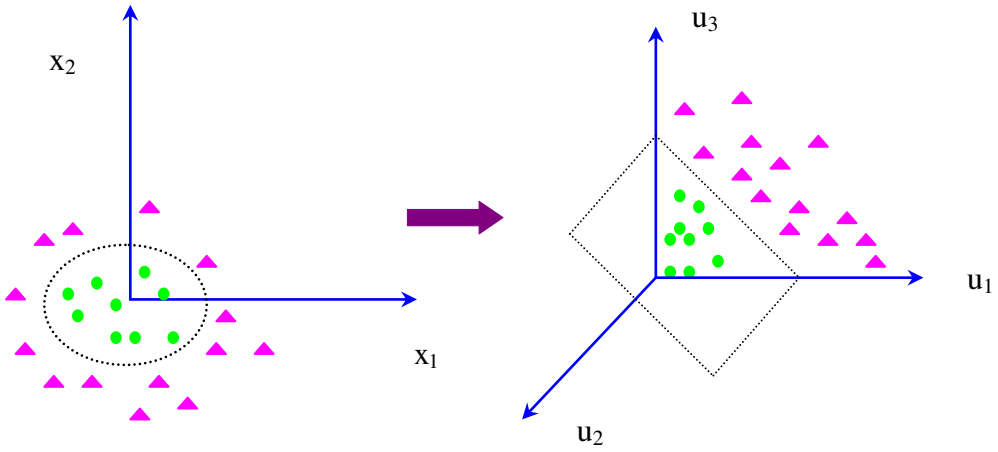


Figure 1.2: Mapping of data from low (2-d) dimension to high dimension (3-d) (Muller et al. 2001)

The idea can be explained through an example shown in Figure 1.2. In the two dimensional space a nonlinear decision surface is required to separate the data belonging to two classes, whereas projecting the data to three dimensional space with a mapping

$$\Phi : R^2 \rightarrow R^3$$

$$(x_1, x_2) \Rightarrow (u_1, u_2, u_3) := (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

The data now is linearly separable by a hyperplane. In this problem the algorithmic complexity is not much as we are dealing with only three dimensional feature space, but most of the real life problems need to be dealing with a very large dimensional feature space. This intractability can be handled by employing a *kernel trick* which is explained in subsequent section.

1.4.3 Kernel functions

For certain features spaces F and corresponding mappings Φ there is highly effective way of computing dot products in feature spaces using kernel functions. For instance for the above problem (Figure 1.2), the two feature space vectors, can be formulated in terms of the kernel function

$$\begin{aligned}(\Phi(\mathbf{x})\cdot\Phi(\mathbf{y})) &= (x_1{}^2, \sqrt{2}\,x_1x_2, x_2{}^2)(y_1{}^2, \sqrt{2}\,y_1y_2, y_2{}^2) \\ &= ((x_1, x_2)(y_1, y_2)^T)^2 \\ &= (\mathbf{x}\cdot\mathbf{y})^2 \\ &=: k(\mathbf{x},\mathbf{y})\end{aligned} \tag{10}$$

For any $d \in N$, kernel function can be generalized as

$$k(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y})^d \tag{11}$$

Equation (11) does not however, hold for all possible features spaces. In fact, one specifies the kernel k at priori, that satisfies Mercer's condition as well as possesses some desired property (e.g. useful measure of similarity or dissimilarity). A Mercer kernel is a function $k(\mathbf{x},\mathbf{y})$ which for all data sets $\{\mathbf{x}_i\}$ gives rise to a positive matrix K with elements $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. (Vapnik 1995; Muller et al. 2001). Most popular examples of Mercer kernel are the polynomial and radial basis function (RBF) kernel.

1.4.4 Brief description of some kernel based algorithms

A family of kernel methods includes support vector for classification and regression, kernel PCA, kernel PLS, Support vector domain distribution (SVDD). The methods are described here in brief and their detailed description with applications have been explained in subsequent chapters.

1.4.4.1 Support vector classification

Support vector machines like any other kernel method rely on pre-processing the data to represent patterns in a high dimension — typically much higher than the original feature space (Vapnik 1995, 1998). With an appropriate non-linear mapping to a sufficiently high dimension, data belonging two classes can always

be separated by a hyperplane. Defining the margin as any positive distance from the decision hyperplane, support vector machines finds the separating hyperplane with the largest margin anticipating the better the generalization of the classifier. The support vectors are the closest (transformed) training patterns to the hyperplane. The support vectors are the training samples that define the optimal separating hyperplane and are the most informative patterns for the classification task. By employing statistical learning theory, it can be shown that the optimal decision surface is the one, which minimizes Euclidean norm $\|\mathbf{w}\|^2$. The problem of minimizing the magnitude of the weight vector constrained by the separation can be reformulated into an unconstrained problem by the method of Lagrange undetermined multipliers. Using the Kuhn–Tucker construction, this optimization can be rewritten as a maximizing problem that can be solved using quadratic programming.

1.4.4.2 Support vector regression

SVMs have originally been developed for classification purposes but their principles can be extended easily to the task of regression. A generalization to regression estimation with $y \in R$, can be given in similar way to support vector classification. (Smola, A. and Schölkopf, 1998) and a quadratic programming problem in terms of kernels can be formulated.

1.4.4.3 Kernel PCA

Kernel PCA corresponds to linear PCA in a higher dimensional feature space, which is nonlinearly related to the input space.(Schölkopf, et al. 2001) The input data **x** are first mapped through some appropriate nonlinear function $\Phi(\mathbf{x})$. Then an *a priori* defined kernel function is used to deal with the possibly very high dimensional space. In other words, in kernel PCA the original problem, reformulated in the form of a dot product of the nonlinear function in the feature space, can be substituted by a kernel function. This simplifies the calculation procedure because the dot product can be computed in the input space itself.

In the instance of noisy data, linear PCA discards the finite variance due to noise by projection of data onto the main principal components. The same holds true for kernel PCA in feature space by using nonlinear principal

components. Kernel PCA, however, extracts a substantially larger number of nonlinear principal components and therefore allows spreading the information regarding the data structure more widely giving a better opportunity to discard some of the eigen directions where the noisy part of data resides. In comparison to other nonlinear principal component analysis (PCA) techniques, kernel PCA requires only the solution of an eigenvalue problem and does not involve any nonlinear optimization. In addition, the number of principal components need not be specified prior to modeling.

1.4.4.4 Kernel PLS

In kernel PLS, the original input data are nonlinearly mapped to a feature space *F* where a linear PLS model is created. (Rosipal and Trejo, 2001) Good generalization properties of the corresponding nonlinear PLS model are then achieved by appropriate estimation of regression coefficients in *F* and by the selection of an appropriate kernel function. Moreover, utilizing the kernel function corresponding to the canonical dot product in feature space allows us to avoid the nonlinear optimization, which is the characteristic of most of nonlinear PLS algorithms. In fact only linear algebra as simple as in a linear PLS regression is required.

1.4.4.5 Support vector domain distribution (SVDD)

For many real-world problems the task is not to classify but to detect novel or abnormal instances. The method of SVDD use the principle of SVM for novelty/abnormality detection. SVDD (Tax and Duin, 1999) avoids solving the harder density estimation problem and uses the simple task of finding the support vectors of the multivariate distribution. The objective of classification of data domain is that the given set of data in should be represented in a unique minimal volume spherical domain enclosing all or nearly all the training points a feature space. The effect of outliers is reduced by using slack variables to allow for data points outside the sphere and task is to minimize the volume of the sphere and number of data points outside the sphere. Having completed the training process a test point is declared as an outlier, if the distance of the point to the center of the sphere is larger than the radius.

In the subsequent chapters the applications of the above methods to solve process engineering have been described in detail.

References

Bhakshi, B., & Stephanopoulos, G., "Wavenet: a Multiresolution Hierarchical Neural Network with Localized Learning," AIChE J., 39, 57 (1993).

Breiman, L., Friedman, J.H., Olshen R.A. & Stone C.J., "Classification & Regression Trees", Wadsworth & Brooks/Cole, Monterey, CA (1984).

Burges, C. J. C., "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining & Knowledge Discovery, 2, 121–167 (1998).

Chen, Q., & Weigand, W. A., "Dynamic Optimization of Nonlinear Processes by Combining Neural Net Model with UDMC," AIChE J., 40, 1488 (1994).

Christianini, N. & Shawe-Taylor, J., "An Introduction to Support Vector Machines," Cambridge University Press, Cambridge, UK (2000).

Dong, D., & McAvoy, T. J., "Non-linear principal component analysis- based on principal curves & neural networks", Computers & Chemical Engineering, 20, 65-78, (1996).

Girosi, F., "Some Extensions of Radial Basis Functions & Their Applications in Artificial Intelligence," Comput. Math. Applic., 24, 61 (1992).

Glassey, J., Montague, G. A., Ward, A. C. & Kara, B. V., "Artificial Neural Network Based Experimental Design Procedures for Enhancing Fermentation Development," Biotechnol. & Bioeng., 44, 397 (1994).

Hand, D. J., & Yu, K. (2001). "Idiot's Bayes - not so stupid after all?" International Statistical Review. 69 (3), 385-399. ISSN 0306 7734

Hastie, T., & Stuezle, W., "Principal curves". Journal of American Statistical Association, 84(406), 502-516, (1989).

Hoerl A.E. and Kennard R.W. , "Ridge regression: Biased estimation for nonorthogonal problems", Technometrics, 12(3), 55-67, (1970).

Hoskins, J. C., Kaliyur, K. M. & Himmelblau, D. M. "Fault Diagnosis in Complex Chemical Plants Using Artificial Neural Networks," AIChE J., 37, 137, (1991).

Hoskuldsson, A., "PLS regression methods", J. Chemom. 211–228, (1988).

Hosmer, D. W. & Lemeshow S., "Applied logistic regression", New York; Chichester, Wiley, (2000).

Karjala, T. W., & Himmelblau, D. M. "Dynamic Data Rectification by Recurrent Neural Networks vs. Traditional Methods," AIChE J., 40, 1865 (1994).

Kim, M., Lee Y-H & Han, C., " Real-time classification of petroleum products using near-infrared spectra", Computers & Chemical Engineering, 24 (2-7), 513-517, (2000).

Kramer, M. A., "Nonlinear principal component analysis using autoassociative neural networks", A. I. Ch. E. Journal, 37(2), 233-243, (1991)

Leonard, J. A., & Kramer, M. A., "Radial Basis Function Networks for Classifying Process Faults," ZEEE Control. Sys. Mag., 11, 31 (1991).

Malthouse, E. C., Tamhane A. C. & Mah, R. S. H., "Nonlinear partial least squares", Computers chem. Engng Vol. 21, No. 8. pp. 875-890, 1997

Martín, Y. G., Pavón, J.L.P., Cordero B. M. & Pinto C. G., "Classification of vegetable oils by linear discriminant analysis of Electronic Nose data", Analytica Chimica Acta, 384(1), 83-94, (1999).

Mulholland, M. , Hibbert, D. B., Haddad P. R., & Sammut C., "Application of the C4.5 classifier to building an expert system for ion chromatography", Chemometrics & Intelligent Laboratory Systems, 27(1), 95-104, (1995).

Muller, K.R., Mika, S. G. , Tsuda, R., K. & Scholkopf B., " An Introduction to Kernel-Based Learning Algorithms", IEEE Transactions on Neural Networks, 12(2), 181–201, (2001).

Rosipal R., & Trejo, L. J., " Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space", Journal of Machine Learning Research 2, 97-123, (2001).

Schölkopf, B., Smola, A., & Müller, K. R. "Nonlinear component analysis as kernel eigenvalue problem", Neural Computation, 10(5), 1299-1319, (1998).

Smola, A. & Schölkopf, B.,"A Tutorial on Support Vector Regression," NeuroCOLT2 Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK (1998).

Tan, S., & Mavrovouniotes, M. L "Reducing Data Dimensionality through Optimizing Neural Network Inputs," AIChE I., 41, 1471 (1995).

Tax, D. M. J. & Duin, R. P.W., "Support Vector Domain Distribution", Pattern Recognition Letters, 20 (11-13), 1191-1199, (1999).

Thompson, M. L., & Kramer, M. A., "Modeling Chemical Processing Using Prior Knowledge & Neural Networks," AIChE J., 40, 1328 (1994).

Tominaga Y., "Comparative study of class data analysis with PCA-LDA, SIMCA, PLS, ANNs, & k-NN", Chemometrics & Intelligent Laboratory Systems, 49 (1),105-115, (1999).

Ungar, L. H., Powell, B. A. & Kamens, S. N., "Adaptive Networks for Fault Diagnosis & Process Control," Comput. & Chem. Eng., 14, 461 (1990).

Vapnik, V., "Statistical Learning Theory," Springer, New York (1998).

Vapnik, V., "The Nature of Statistical Learning Theory," Springer, New York (1995).

Venkatasubramanian, V., Vaidyanathan, R. & Yamamoto, Y. "An Analysis of the Learning, Recall, & Generalization Characteristics of Neural Networks for Process Fault Diagnosis," Comput. & Chem. Eng., 14, 699 (1990).

Watanabe, K., Hirota, S., Hou, L. & Himmelblau, D. M., "Diagnosis of Multiple Simultaneous Faults via Hierarchical Artificial Neural Networks," AIChE J., 40, 839 (1994).

Wold, S., Kettaneh-Wold, N., Skagerberg, B., "Nonlinear PLS modeling", Chemometr. Intell. Lab. Syst. 7, 53– 65, (1989).

Wold, S., "Nonlinear partial least squares modeling: II. Spline inner relation", Chemometr. Intell. Lab. Syst., 14, 71–84, (1992).

Wold, S., Sjostrom, M., & Eriksson L, "PLS-regression: a basic tool of chemometrics", Chemometrics & Intelligent Laboratory Systems, 58, 109–130, (2001).

Wu, W., Mallet, Y., Walczak, B., Penninckx, W., Massart, D. L., Heuerding S. & Erni F., "Comparison of regularized discriminant analysis, linear discriminant analysis & quadratic discriminant analysis applied to NIR data", Analytica Chimica Acta, 329(3), 257-265, (1996).

Wu W. & Massart D. L., "Regularised nearest neighbour classification method for pattern recognition of near infrared spectra", Analytica Chimica Acta, 349 (1-3), 253-261, (1997).

Yamashita, Y., “Supervised learning for the analysis of process operational data”, Computers & Chemical Engineering, 24(2-7), 471-474, (2000).

Chapter 2
SUPPORT VECTOR MACHINES AND ITS APPLICATIONS TO PROCESS ENGINEERING

2.1 Introduction

Support Vector machine (SVM), a recently developed tool, is increasingly gaining popularity as the preferred tool for classification, regression and novelty detection type of applications. Among the many attractive features, a rigorous basis on statistical learning theory, nominal increase in computational cost for nonlinear learning and excellent generalization performance are worthy of mentioning. In their present form, Support vector machines were first developed at AT&T Bell laboratories by Vapnik and co-workers (Vapnik, 1995, 1998; Cortes and Vapnik, 1995). Initially developed for optical character recognition, they have since been applied in a variety of fields for solving important classes of problems. Conventional tools like artificial neural networks are based on empirical risk minimization methodology that minimizes the mean square error over the training set (Lapedes and Farber, 1987; Wasserman, 1993). This hypothesis would perform poorly on unseen data unless some sort of capacity control is introduced. Thus in spite of being very accurate, an artificial neural network may suffer from overfitting the training data. Other difficulties with the use of neural networks concern the reproducibility of results due to the largely random initialization of the networks, convergence to local minimum and the lack of information regarding the classification produced. SVMs on the other hand are based on structural risk minimization principle that aims at minimizing a bound on the generalization error of a model.

The number of free parameters in SVMs does not depend explicitly on the input dimensionality, unlike other machine learning methods. This property is highly desirable and useful for problems with large dimensions. The basic idea of SVM to handle non-linearly separable data is to transform the input space into a higher dimensional feature space, nonlinearly related to the input space. This induces a computational problem of having to work with very large vectors. This problem can be tackled by using appropriate kernels, whereby all the

computations can be done in the input space itself, thus greatly simplifying the learning task. SVM has been applied to a variety of pattern recognition applications such as handwritten digit recognition, object recognition, speaker identification, face detection in image, text categorization, microarray data classification etc. (Cortes and Vapnik, 1995; Osuna et al. 1997; Schmidt, 1996). SVM has also been applied to a various regression estimation problems like time series prediction (Muller et al. 1997). In most of these applications generalization performance of SVM has been found to be as good or better than that of the competing methods.

The potential of SVMs for process engineering applications has not yet been utilized. In the present work we highlight the usefulness of SVMs in both classification and regression applications for problems important to process engineering. SVM methodology is derived for both classification and regression problems and one case study each for SVM classification and regression are then described. Finally, the significance of the performance of SVMs and conclusions derived from the present study are discussed.

2.2 Support Vector Classification

2.2.1 Classifier for linearly separable patterns

Consider separation of the set of training vectors belonging to two classes,

$$(y_1, \mathbf{x}_1), \ldots, (y_N, \mathbf{x}_N), \qquad \mathbf{x} \in \Re^n, y \in \{-1, +1\} \tag{1}$$

with a hyperplane defined by

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0 \tag{2}$$

For a linear classification problem, the optimal values of $\mathbf{w}$ and b are those for which the hyperplane separates both the classes perfectly and the distance between the nearest data point belonging to different classes is maximum (Burges,

1998; Christianini and Shawe-Taylor, 2001; Gunn, 1997; Osuna et al., 1997). Defining canonical hyperplanes to remove the redundancy in the above equation, the equation for canonical hyperplane is

$$\min_i \left|\mathbf{w}\cdot\mathbf{x}_i + b\right| = 1 \qquad (3)$$

So, for the correctly classified data set

$$\mathbf{w}\cdot\mathbf{x}_i + b \geq 1 \qquad \forall\ y_i = +1 \qquad (4)$$

and $\quad \mathbf{w}\cdot\mathbf{x}_i + b \leq -1 \qquad \forall\ y_i = -1$

The above constraint can also be written in a compact form as:

$$y_i\left[\left(\mathbf{w}\cdot\mathbf{x}_i\right)+b\right] \geq 1, \qquad i = 1,.....,N \qquad (5)$$

A set of hyperplanes satisfying the above constraints are known as *canonical hyperplanes.* The statistical learning theory by Vapnik (1995, 1998) shows that if all the points lie in the unit *n*-dimensional sphere, the set

$$\{f_{\mathbf{w},b} = sign(\mathbf{w}\cdot\mathbf{x}+b) \left| \|\mathbf{w}\| \leq A\right.\} \qquad (6)$$

has a VC dimension d that satisfies

$$d \leq \min\{\left\lfloor R^2A^2 \right\rfloor, n\} + 1 \qquad (7)$$

where R is the radius of the hypersphere enclosing all the data points. From the above equation, it is obvious that we can exert control over the VC dimension of the canonical hyperplanes independently of the number of data points by properly choosing the quantity A. Further, it can also be shown that the distance from a point **x** to the hyperplane associated with the pair $(\mathbf{w}, b)$ can be given by

$$D\left(\mathbf{x};\mathbf{w},b\right) = \frac{\left|\mathbf{w}\cdot\mathbf{x}+b\right|}{\|\mathbf{w}\|} \qquad (8)$$

and as per normalization given by equation (3), the distance between the canonical hyperplane and the closest point becomes $1/\mathbf{w}$ and the margin between the closest points of the two data sets of different classes is given by $2/\mathbf{w}$. Thus, the VC dimension and the complexity of the canonical hyperplane structures can be controlled by constraining $\|\mathbf{w}\|$ and the hyperplane that optimally separates the data is the one that minimizes $\frac{1}{2}\|\mathbf{w}\|^2$. If $\|\mathbf{w}\| < A$ then the distance of a canonical hyperplane to the closest data point, due to equation (8), has to be larger than $1/A$. Such a constraint reduces the set of possible canonical hyperplanes and thereby reducing the capacity of the classifier. Thus by maximizing the margin between the closest points belonging to the two classes the VC dimension can be controlled and hence the true error can be minimized. Thus SVM obtains the optimal hyperplane by minimizing the Euclidean norm, $\|\mathbf{w}\|^2$.

So, a linear support vector machine minimizes the function

$$g(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \tag{9}$$

subject to the constraints:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 \qquad \forall i \tag{10}$$

The solution of this problem is equivalent to determining the saddle point of the Lagrangian

$$L_p = \frac{1}{2}(\mathbf{w}\cdot\mathbf{w}) - \sum_{i=1}^{N}\alpha_i\{[(\mathbf{w}\cdot\mathbf{x}_i)+b]y_i - 1\} \tag{11}$$

with $L_p = L(\mathbf{w}, b, \alpha)$, and α_i s are the nonnegative Lagrangian multipliers.

At saddle point, the primal problem has minimum for $\mathbf{w} = \overline{\mathbf{w}}$ and $b = \bar{b}$. Differentiating the Lagrangian with respect to $\mathbf{w}$ and b and setting to zero, we get the following

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{N} y_i \alpha_i = 0 \tag{12}$$

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0$$

yielding

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{13}$$

The primal problem, formulated as above, deals with a convex cost function containing linear constraints. It would thus be possible to construct the Wolfe dual Lagrangian. For this, we first expand the primal formulation term by term

$$L_P = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{w} \cdot \mathbf{x}_i + \sum_{i=1}^{N} \alpha_i - b \sum_{i=1}^{N} \alpha_i y_i \tag{14}$$

The last term on the right hand side is zero due to the optimality condition of equation (12). Also, we know that

$$\mathbf{w} \cdot \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{w} \cdot \mathbf{x}_i = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{15}$$

With these simplifications, the dual problem can now be stated as

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{16}$$

Subject to $\sum_{i=1}^{N} y_i \alpha_i$

$$\alpha \geq 0$$

Or, in the typical format of a constrained quadratic optimization problem, as

Maximize $-\frac{1}{2}\alpha^T Q\alpha + \sum_{i=1}^{N}\alpha_i$

where Q is an $N \times N$ matrix such that

$$Q_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \qquad (17)$$

For the solution at the saddle point $\left(\overline{\mathbf{w}}, \overline{b}\right)$, it follows that

$$\overline{\mathbf{w}} = \sum_{i=1}^{N} \overline{\alpha}_i y_i \mathbf{x}_i \qquad (18)$$

$\overline{b}$ can be determined from $\overline{\alpha}$, which is a solution of the dual problem, and from the Kuhn-Tucker conditions, which state that for each Lagrangian multiplier, the product of the multiplier with its corresponding constraint vanishes:

$$\overline{\alpha}_i \left(y_i \left(\overline{\mathbf{w}} \cdot \mathbf{x}_i + \overline{b} \right) - 1 \right) = 0, \qquad i = 1,.....,N \qquad (19)$$

It must be noted that only those values of $\overline{\alpha_i}$ can be nonzero for which the constraints equation

$$y_i \left(\overline{\mathbf{w}} \cdot \mathbf{x}_i + \overline{b} \right) \geq 1, \qquad i = 1,2,.....,N \qquad (20)$$

is satisfied with equality sign. This restriction reduces the number of Lagrangian multipliers with finite values, meaning that the solution vector $\overline{\mathbf{w}}$ is a linear combination of a small percentage of the points $\mathbf{x}_i$. So, these are the points closest to the optimal separating hyperplane and are known as the support vectors. (Figure 2.1)

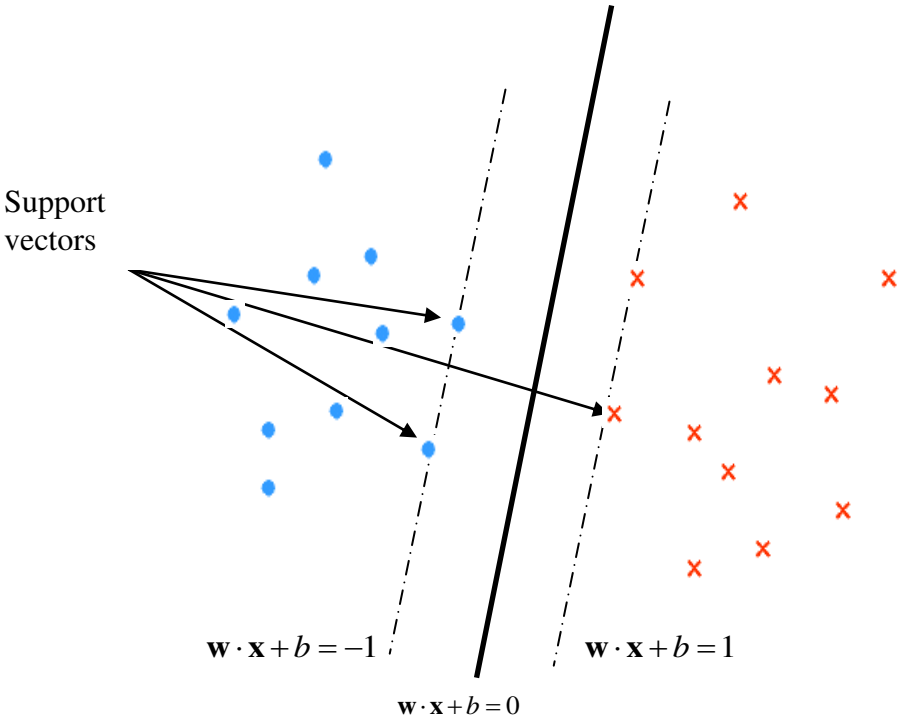


Figure 2.1: Diagram of linear SVM classifier showing Support Vectors

The problem of classifying a new data point $\mathbf{x}$ is now simply solved by looking at the

$$\text{sgn}\left(\overline{\mathbf{w}}\cdot\mathbf{x}+\overline{b}\right) \qquad (21)$$

2.2.2 Classifier for linearly non-separable patterns

For a linearly non-separable data set, it is not possible to construct a hyperplane without a certain amount of classification error. It would, however, be possible to find an optimal hyperplane that minimizes the probability of occurrence of classification errors, averaged over the training set. This is done by introducing N nonnegative slack variables such that

$$y_i(\mathbf{w}\cdot\mathbf{x}_i+b) \geq 1-\zeta_i \qquad i=1,2,.....,N \qquad (22)$$

where $\zeta_i \geq 0$. The generalized optimal separating hyperplane is determined by finding the vector $\mathbf{w}$, that minimizes the functional,

$$g\left(\mathbf{w},\zeta\right) = (1/2)\,\|\,\mathbf{w}\,\|^2 + C\sum_{i=1}^{N}\zeta_i \qquad (23)$$

(where, C, is a given value) subject to the constraints in equation (22).

The saddle point of the Lagrangian corresponds to the solution to the optimization problem of equation (23) under the constraints of equation (22). It can be shown by using methods described above that the dual solution can be obtained as:

$$\max_{\alpha} W(\alpha) = \max_{\alpha} -(1/2)\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\,\alpha_j\,y_i\,y_j\,(\mathbf{x}_i\cdot\mathbf{x}_j)+\sum_{i=1}^{N}\alpha_i \qquad (24)$$

with the constraints,

$$0\leq\alpha_i\leq C \qquad i=1,.....,N \qquad (25)$$

$$\sum_{i=1}^{N}\alpha_i\,y_i = 0 \qquad (26)$$

Thus the problem for the case of linearly non-separable patterns is again a problem of solution of a quadratic optimization problem exactly similar to that of the simple case of linearly separable patterns excepting that the constraints $\alpha_i \geq 0$ are now replaced by a new set of constraints $0 \leq \alpha_i \leq C$. The parameter C controls the tradeoff between complexity of the support vector machine and the number of non-separable points. This can be viewed as a *regularization* parameter while obtaining the optimal hyperplane and can be determined by experimental cross validation. Alternatively, it can be obtained by analytically estimating the VC dimension and then by using bounds on the generalization performance of the machine based on the VC dimension.

2.2.3 Non-linear support vector machines

The methods developed in the above sections are for linear classifiers and as such cannot deal with non-linearly separable data. SVM handles non-linearly separable data by mapping the data into a richer higher dimensional feature space, which is nonlinearly related to the input space and by subsequently using a linear classifier. The mapping of the input data $\mathbf{x}$ in the feature space $\mathbf{x} \rightarrow \Phi(\mathbf{x})$ where they are linearly separable is shown schematically in Figure 2.2. Working in higher dimensional feature space induces an intractable computational problem of having to deal with very large vectors. This problem can be solved by introduction of implicit mapping by kernels. Any function that returns the value of the dot product between the images of two arguments can be used as a kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{27}$$

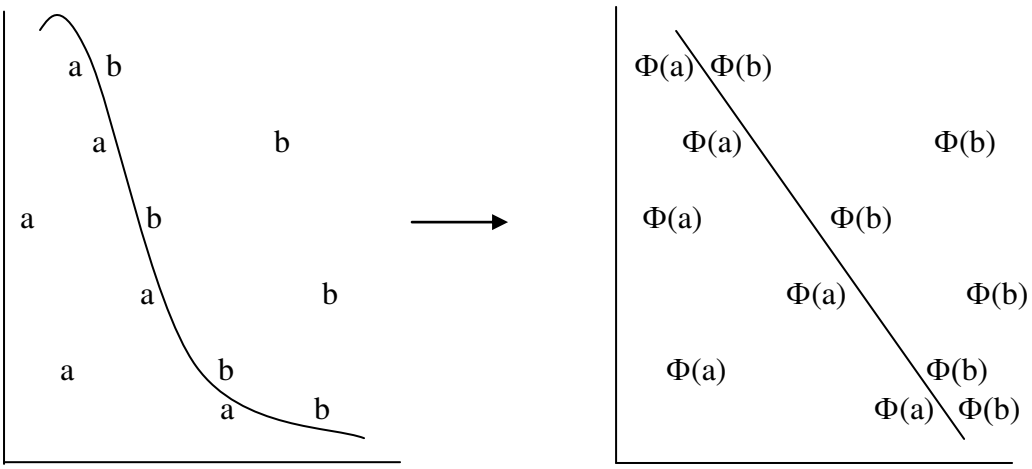


Figure 2.2: Mapping of data into feature space where it is linearly separable

The idea of kernel functions is to perform operations in input space rather than the very high dimensional feature space. In other words, an inner product in the feature space has an equivalent kernel in the input space. A kernel function can be selected by using the Mercer's theorem. The kernel matrix contains all the necessary information for the support vector machine learning algorithm and is generally known as the *information bottleneck.* A kernel matrix is a symmetric positive definite matrix. A list of popular kernels (Gunn, 1997) is shown in Table 2.1.

Table 2.1: A list of some popular kernel functions

Sr.No.	Name of the Kernel	Expression
1	Polynomial	$K(\mathbf{x}_i, \mathbf{x}_j) = ((\mathbf{x}_i \cdot \mathbf{x}_j) + 1)^p \quad p = 1,2,....$
2	Gaussian Radial Basis Function	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{\left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2}{-2\sigma^2}\right)$
3	Exponential Radial Basis Function	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{\left\| \mathbf{x}_i - \mathbf{x}_j \right\|}{-2\sigma^2}\right)$
4	Multi-layer Perceptron	$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(b(\mathbf{x}_i \cdot \mathbf{x}_j) - c)$

Thus equation (24) can be written in the form of kernel functions in the low dimensional input space itself as:

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \sum_{i=1}^{N} \alpha_i - (1/2) \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \qquad (28)$$

subject to the constraints

$$0 \le \alpha_i \le C \qquad i = 1,....., N \qquad (29)$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \qquad (30)$$

After the optimal values of α_i have been found, the decision function is based on the sign of:

$$f(\mathbf{x}) = \sum_{i=1}^{m} y_i \overline{\alpha}_i K(\mathbf{x}, \mathbf{x}_i) + \overline{b} \qquad (31)$$

The bias can be found from the primal constraints:

$$\overline{b} = -\frac{1}{2} \overline{\mathbf{w}} \cdot [\Phi(\mathbf{x}_r) + \Phi(\mathbf{x}_s)] \qquad (32)$$

where $\mathbf{x}_r$ and $\mathbf{x}_s$ are any support vectors from each class.

Thus SVM can transform the problem of finding an optimal hyperplane to deal with nonlinearly separable data into a compact QP formalism having a unique solution that can be solved with standard QP solvers.

2.3 Support Vector Regression

Similar to classification problems many of the real life problems require a nonlinear model to adequately regress the data. The methodology described in the previous sections can be easily extended to employ SVMs to handle nonlinear regression (Drucker et al., 1997, Schölkopf et al. 1999, Smola and Schölkopf, 1998). A nonlinear mapping can be used in a similar fashion to map the data into a high dimensional feature space and perform the linear regression. Kernel functions can again be used to do this linear regression in the input space.

2.3.1 Linear Regression

Consider the problem of approximating the training data-set,

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N), \qquad \mathbf{x} \in \Re^n, y \in \Re \tag{33}$$

with a linear function,

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \tag{34}$$

The minimum of the functional,

$$g(\mathbf{w}, \xi) = \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{N} \left(\xi_i^- + \xi_i^+ \right) \tag{35}$$

gives the optimal regression function. In equation (35) C is a pre-specified value and ξ^-, ξ^+ are slack variables representing upper and lower constraints on the outputs of the system.

We can use ε-insensitive loss function in the form,

$$\begin{aligned} L_\varepsilon &= 0 && \text{for } |f(\mathbf{x}) - y| < \varepsilon \\ &= |f(\mathbf{x}) - y| - \varepsilon && \text{otherwise} \end{aligned} \tag{36}$$

So, the solution is given by,

$$\max_{\alpha,\alpha^*} W(\alpha,\alpha^*) = \max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^{N}\alpha_i(y_i - \varepsilon) - \alpha_i^*(y_i + \varepsilon) \tag{37}$$

or alternatively,

$$\bar{\alpha}, \bar{\alpha}^* = \arg\min_{\alpha,\alpha^*} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)y_i + \sum_{i=1}^{N}(\alpha_i + \alpha_i^*)\varepsilon \tag{38}$$

with constraints,

$$\begin{aligned} &0 \le \alpha_i, \alpha_i^* \le C, \qquad i = 1, ..., N \\ &\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \end{aligned} \tag{39}$$

Solving equation (37) with constraints from equation (39) determines the Lagrange multipliers, α_i, α_i^*, and the regression function is given by equation (34), where,

$$\begin{aligned} \bar{\mathbf{w}} &= \sum_{i=1}^{N}(\bar{\alpha}_i - \bar{\alpha}_i^*)\mathbf{x}_i \\ \bar{b} &= -\frac{1}{2}\bar{\mathbf{w}} \cdot (\mathbf{x}_r + \mathbf{x}_s) \end{aligned} \tag{40}$$

The Karush-Kuhn-Tucker (KKT) conditions that are satisfied by the solution are

$$\overline{\alpha}_i \overline{\alpha}_i^* = 0, \qquad i=1,.....,l \tag{41}$$

Therefore, the support vectors are points where exactly one of the Lagrange multipliers is greater than zero.

2.3.2 Non-linear Regression

The idea here again is to employ a non-linear mapping to map the data into a higher dimensional feature space, where linear regression is performed. The kernel trick as explained in the classification section comes to the rescue to address this 'curse of dimensionality'. The dot product is replaced by a suitable kernel function K Thus, the optimization problem becomes,

$$\max_{\alpha,\alpha^*} W(\alpha,\alpha^*) = \max_{\alpha,\alpha^*} \sum_{i=1}^{N} \alpha_i^* (y_i - \varepsilon) - \alpha_i (y_i + \varepsilon) - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j) \tag{42}$$

with constraints

$$\begin{aligned} &0 \le \alpha_i, \alpha_i^* \le C, \qquad i=1,.....,N \\ &\sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0 \end{aligned} \tag{43}$$

Solving equation (42) with constraints equation (43) determines the Lagrange multipliers, α_i, α_i^*, and the regression function is given by,

$$f(\mathbf{x}) = \sum_{\text{SVs}} (\overline{\alpha}_i - \overline{\alpha}_i^*) K(\mathbf{x}_i, \mathbf{x}) + \overline{b} \tag{44}$$

where,

$$\overline{\mathbf{w}} \cdot \mathbf{x} = \sum_{SVs} (\overline{\alpha}_i - \overline{\alpha}_i^*) K(\mathbf{x}_i, \mathbf{x}_j)$$
$$\overline{b} = -\frac{1}{2} \sum_{SVs} (\overline{\alpha}_i - \overline{\alpha}_i^*) (K(\mathbf{x}_i, \mathbf{x}_r) + K(\mathbf{x}_i, \mathbf{x}_s)) \quad (45)$$

SVM methodology described above is applied to fault diagnosis and QSPR problem in next subsequent sections.

2.4 Application of SVM to Fault Diagnosis

The case study is the reactor example given in Luyben (1990) and worked out by Venkatsubramanian, *et. al.* (1990) by using back-propagation neural networks. The faults (F_1 through F_6) occur due to malfunctions in: high inlet flowrate, F_1; low inlet flowrate, F_2; high inlet reactant concentration, F_3; low inlet reactant concentration, F_4; high inlet temperature of reactant stream, F_5; and low inlet temperature of reactant stream, F_6. Table 2.2 shows the twelve measured patterns of the reactor output data corresponding to faults F_1–F_6. The data belonging to normal operation is also shown in the table. Table 2.3 shows twelve measured output data corresponding to double faults. All the double-faults involved the simultaneous occurrence of faults due to high inlet temperature of the reactor input stream along with other faults; for example, the first data in the table shows the case of simultaneous occurrence of malfunctioning due to high inlet flowrate and high temperature of the inlet stream.

Table 2.2: Single-fault training data

No.	Fault	C	T	V	F	T_j	F_j
1	F_1 (+15%)	0.2575	600.66	48.6	46.0	595.0	52.5
2	F_1 (+5%)	0.2494	600.24	48.2	42.0	594.8	50.8
3	F_2 (-15%)	0.2307	599.17	47.4	34.0	594.2	46.6
4	F_1 (-5%)	0.2405	599.73	47.8	38.0	594.5	48.8
5	F_3 (+15%)	0.2520	602.82	47.6	36.3	596.1	61.2
6	F_3 (+5%)	0.2480	600.99	47.9	38.8	595.5	53.9
7	F_4 (-15%)	0.2315	596.74	48.4	43.7	592.9	36.9
8	F_4 (-5%)	0.2414	598.94	48.1	41.2	594.1	45.7
9	F_5 (+15%)	0.2020	608.44	48.0	40.0	598.9	83.7
10	F_5 (+5%)	0.2296	602.96	48.0	40.0	596.2	61.8
11	F_6 (-15%)	0.2991	589.73	48.0	40.0	588.9	8.8
12	F_6 (-5%)_	0.2617	596.83	48.0	40.0	592.9	37.2

For classifying these patterns we adopted a procedure somewhat different from the one adopted in the earlier studies by Venkatasubramanian, *et al.* (1990). First, we used a classifier to separate the single and double-faults. This involves combining all the data in Table 2.2 comprising of single faults in one group and data in Table 2.3 comprising of double faults in another group and using SVM to classify them into two different classes. An SVM with an RBF kernel was able to classify the single faults and double faults into separate classes with 100% success rate. Test-data shown in Table 2.4 were also employed to test the robustness of the machine in generalizing the machine capability for classifying untrained data.

Further, we used multiclass SVMs to identify and sub-classify different types of faults in both Table 2.2 and Table 2.3. For both the single-faults and double-faults cases, our aim was to train the SVM to correctly classify the faults belonging to a particular fault, irrespective of the percentage deviation. Thus for the single fault patterns, the measurement patterns with 15% increase and 5% increase in inlet flowrate were put into class 1 (*i.e.*, the F_1 Class), 15% and 5% decrease was put in class 2 and so on. There are various algorithms in vogue for multi-class classification. We have used the simplest method, viz. the one-against-all (Weston & Watkin, 1999) method to classify the different classes of faults. In this method the k class problem is converted into a problem of solving k binary classifiers problems. The k^{th} classifier constructs a hyperplane between class n and k–1 other classes. A majority vote across the classifiers is applied to classify the new test point. Newer methods for multi-class pattern recognition problem solving have recently been described, but these methods do not out-perform the one-against-all method as described above. The multiclass pattern recognition problem of separating the data in Table 2.2 into six separate classes can thus be solved by considering it as a collection of binary classification problems.

Table 2.3: Double faults training data

No.	FAULT	C	T	V	F	Tj	Fj
1	F_1 (+10%) F_5 (+10%)	0.2201	606.9540	48.4	44	598.5400	77.7193
2	F_1 (5%) F_5 (10%)	0.2178	606.3732	48.2	42	597.8720	75.3920
3	F_1 (10%) F_5 (5%)	0.2361	603.8100	48.4	44	596.6025	65.1414
4	F_2 (-10%) F_5 (+10%)	0.2095	604.5757	47.6	36	596.9857	68.2029
5	F_2 (-5%) F_5 (10%)	0.2125	605.1800	47.8	38	597.2882	70.6393
6	F_2 (-10%) F_5 (5%)	0.2221	602.0900	47.6	36	595.7284	58.2613
7	F_2 (-5%) F_5 (5%)	0.226	602.5300	47.8	38	595.9500	60.0570
8	F_3 (10%) F_5 (10%)	0.2237	608.1700	48.0	40	598.7400	82.6100
9	F_3 (15%) F_5 (10%)	0.2269	609.4000	48.0	40	599.3200	87.5333
10	F_3 (15%) F_5 (5%)	0.2427	606.6200	48.0	40	597.9900	76.3800
11	F_4 (-10%) F_5 (10%)	0.2043	603.4797	48.0	40	596.4350	63.8189
12	F_4 (-5%) F_5 (10%)	0.2101	604.6200	48.0	40	597.0070	68.3792
13	F_4 (-10%) F_5 (5%)	0.2174	600.6700	48.0	40	594.9930	52.5800
14	F_4 (-5%) F_5 (5%)	0.2238	601.8100	48.0	40	595.5830	57.1370

The above-mentioned problem of separating the single fault data into six different classes was solved as an SVM multi-class problem employing one-against- all method. Both polynomial and RBF kernels were able to classify the data into seven different classes without any misclassifications. We used a similar methodology for sub-classifying the double faults data into four different classes, *i.e.*, data belonging to F_1 and F_5 class; F_2 and F_5 class; F_3 and F_5 class and F_4 and F_5 class. We then expanded the dataset in Tables 2.2 and 2.3 to include errors ranging between –25 and +25%. Again, SVM successfully classified the faults belonging to separate classes perfectly. The trained classifiers were also tested with test data. All the fault-patterns of these test-data were again successfully classified by SVM without any errors. Finally, we included data in which three faults occur simultaneously. The data are shown in the Table 2.5.

Table 2.4: Single and double faults test data

Class	No.	Fault	C	T	V	F	Tj	Fj
Single	1	F_1(+10%)	0.2535	600.53	48.4	44	594.53	51.9141
Faults	2	F_1(20%)	0.2613	600.21	48.8	48	595.41	53.1704
Double Faults	1	F_1(5%) F_5(5%)	0.2329	603.39	48.2	42	596.394	63.4925
	2	F_3(10%) F_5(5%)	0.239	605.38	48.0	40	597.38	71.425

Table 2.5: Triple faults training data

No.	Fault	C	T	V	F	T_j	F_j
1	F_1 (+10%) F_2 (+10%) F_3 (+10%)	0.2450	600.000	48.40	44.0	594.590	49.9
2	F_1 (+5%) F_2 (+5%) F_3 (+5%)	0.2379	604.640	48.20	42.0	597.080	68.46
3	F_1 (+15%) F_2 (+15%) F_3 (+15%)	0.2147	614.770	48.60	46.0	602.040	108.98
4	F_2 (-5%) F_4 (-5%) F_6 (-5%)	0.2488	595.720	47.80	38.0	591.907	32.78
5	F_1 (+8%) F_2 (+8%) F_3 (+8%)	0.2321	607.600	48.32	43.2	598.240	80.30
6	F_1 (12%) F_2 (12%) F_3 (12%)	0.2227	611.675	48.48	44.8	600.110	96.60

All these data represent cases in which there is a simultaneous malfunctioning in the inlet flowrate, inlet concentration and the inlet temperature. We employed an SVM with a RBF kernel to separate data belonging to single fault, double fault and triple faults into different groups. The three-class classifier classified all the data into different classes without any errors.

2.5 Application of SVM to Quantitative Structure Property Relationships (QSPR)

In the case of regression, we have taken a case study of development of *Quantitative Structure Property Relationships* (QSPRs) for the correlation and estimation of physical properties of organic compounds. The fact that the physico-chemical properties can be successfully correlated with molecular-structural characteristics expressed in terms of appropriate molecular descriptors has been amply revealed in various recent studies. Different investigators have demonstrated the usefulness of AI (Artificial Intelligence) methods as effective tools for the development of QSPRs. The main advantage of AI techniques like neural networks and SVM is that the QSPRs can be developed directly from the input-output data without *a priori* specification of the analytical form of the particular correlation model. Recently, Cohen and coworkers have developed neural network based QSPRs for predicting boiling points (Espinosa et al., 2000), aqueous solubility (Yaffe et al., 2001b) and vapor pressure (Yaffe et al., 2001a) of different hydrocarbons. They have employed the Back-propagation neural networks architecture. The QSPRs were obtained from the knowledge of four valance molecular connectivity indices, a second order Kappa shape index, the dipole moment and the molecular weight (Espinosa et al., 2000). In all, seven input parameters were used to predict the output, viz. the boiling point of the alkenes.

2.5.1 Prediction of boiling points of alkenes

In the present study the support vector machine with a radial basis function kernel was trained to predict the boiling points of various alkenes. The total data set for alkenes (144 examples) was split into two sets, 26 for testing, and the remaining 118 were further split into 97 training examples and 21 validation examples. The SVM was trained with the 97 examples. Several combinations of the parameters C, σ and ε were used and the parameter combination that gave the lowest error on the validation set of 21 alkenes was chosen. The corresponding error on the test set of 26 alkenes was then found out. The accuracy of the results was the best for $\varepsilon = 0.1$, $\sigma = 0.54$ and $C = 2985$. The errors obtained using SVM are compared with

the errors obtained by back propagation neural network in Table 2.6. It can be seen from the table that SVM obtains a regression function having lower error in all the three phases, viz., testing, validation and training.

Table 2.6: Results for Boiling point prediction

Error	BPNN	SVR
Training	0.93 %	0.68 %
Testing	1.96 %	1.78 %
Validation	1.83 %	1.46 %

2.6 Summary

The working of support vector machines based on structural risk minimization principle in learning tasks involving linear and nonlinear classification and regression has been highlighted. For binary classification tasks, the support vector machine obtains the optimal decision surface by maximizing the margin between the closet data points belonging to the two classes. Such a large margin classifier minimizes both the VC dimension and the training error. This facilitates in controlling the capacity of the classifier and in exhibiting good generalization capabilities. SVMs first transform the input space into a higher dimensional feature space, nonlinearly related to the input space. Further, by using the kernels functions it enables computations to be done in the input space itself. The convex quadratic optimization of the learning problem enables us to use standard QP solvers and get the unique global minimum. There are relatively few free parameters and cross validation and generalization becomes easy and simple. The SVM methodology was applied to classification and regression problems. The case studies considered are the fault detection in CSTR and quantitative structure property relations (QSPR) problem dealing with prediction of boiling points of aliphatic hydrocarbons from molecular descriptors data. SVM successfully classifies and sub-classifies various types of faults occurring in simulated CSTR using one against all multi-class strategy. For the QSPR problem, SVM obtains smaller errors for training, validating and testing sets than the ones obtained by using back propagation networks. The examples clearly demonstrate the ease, elegance and superiority of this new tool over the other conventional tools and should prove useful in a number of other process engineering applications.

References

Burges, C. J. C., "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining & Knowledge Discovery, 2, 121-167, (1998).

Christianini, N., & Shawe-Taylor, J., "An Introduction to Support Vector Machines", Cambridge University Press, (2001).

Cortes, C., & Vapnik, V., "Support Vector Networks", Machine Learning 20, 273-297, (1995).

Drucker, H., Burges, C., Kaufman, L., Smola, A. & Vapnik, V. "Support Vector Regression Machines" in "Neural Information Processing Systems 9", M. Mozer, M. Jordan, & T. Petsche, Eds., MIT Press, Cambridge, MA.(1997).

Espinosa, G., Yaffe, D. , Cohen, Y., Arenas, A. & Giralt, F., "Neural Network Based Quantitative Structure-Property Relationships (QSPRs) for Predicting Boiling Points of Aliphatic Hydrocarbons", J. Chem. Inf. Comput. Sci. 40, 859-879, (2000).

Gunn, S., "Support Vector Machines for Classification & Regression", ISIS Technical Report, (1997).

Lapedes, A. & Farber, R., "How Neural Nets Work," in "Neural Information Processing Systems", D. Z. &erson, Eds., American Institute of Physics, New York, (1988), pp. 442-456.

Luyben, W. L., "Process Modeling, Simulation & Control for Chemical Engineers" (2nd ed.) McGraw-Hill, New York (1990).

Muller K.-R., Smola A., Ratsch G., Scholkopf B., Kohlmorgen J., & Vapnik. V., "Predicting time series with support vector machines", In Proceedings, International Conference on Artificial Neural Networks, page 999, Springer Lecture notes in Computer Science. (1997).

Osuna E., Freund R. & Girosi F., "Training support vector machines: an application to face detection", In IEEE Conference on Computer Vision and Pattern Recognition, 130-136, (1997).

Schmidt M. "Identifying speaker with support vector networks", In Interface' 96 Proceedings, Sydney. (1996).

Schölkopf, B., Bartlett, P., Smola, A. & Williamson, R., "Shrinking the Tube: A New Support Vector Regression Algorithm," in "Advances in Neural Information Processing Systems 11", M. Kearns, S. Solla, & D. Kohn, Eds., The MIT Press, Cambridge, MA. (1999).

Smola, A., & Schölkopf, B., "A Tutorial on Support Vector Regression", NeuroCOLT2 Technical Report Series, (1998).

Vapnik, V., "Statistical Learning Theory", Springer, (1998).

Vapnik, V., "The Nature of Statistical Learning Theory". Springer, (1995).

Venkatasubramanian, V., Vaidyanathan R. & Yamamoto, Y., "Process Fault Detection & Diagnosis Using Neural Networks-1: Steady State Processes", Comput. Chem. Eng. 14(7), 699-712, (1990).

Wasserman, P. D., "Advanced Methods in Neural Computing", Van Nostr& Reinhold, New York, (1993)

Weston, J., & Watkin, C., "Support Vector Machines for Multi-class Pattern Recognition" in "Proc.Seventh European Symposium on Artificial Neural Networks", (1999).

Yaffe, D., & Cohen, Y., "Neural Network Based Temperature Dependent Quantitative Structure-Property Relationships (QSPRs) for Predicting Vapour Pressure of Hydrocarbons", J. Chem. Inf. Comput. Sci. 41, 463-477, (2001a).

Yaffe, D., Cohen, Y., Espinosa, G., Arenas, A. & Giralt, F., "A Fuzzy ARTMAP Based on Quantitative Structure-Property Relationships (QSPRs) for Predicting Aqueous Solubility of Organic Compounds", J. Chem. Inf. Comput. Sci. 41, 1177-1207 (2001b).

Notation

b = bias
$\overline{b}$ = optimal bias
C = Upper bound for Lagrange multipliers
d = Vapnik-Chervonenkis dimension
D = Distance of a point from the hyperplane
$f(x_k)$= functional relationship describing the dynamics of the time series
$\hat{f}(x_k)$=estimate of $f(x_k)$
$f(\mathbf{z})$= decision function for the vector $\mathbf{z}$
$g(\mathbf{w})$= objective function
k = number of classes in multiclass problem
K = kernel matrix
$K(\mathbf{x}_i, \mathbf{x}_j)$ = kernel function
L_P, L_D = Lagrangian functions for primal and dual formulations respectively
m = number of support vectors
N = Number of training set
R = radius of hypersphere
r = desired response in regression
SVs = support vectors
t =time
$\mathbf{w}$ = weight vector
$\overline{\mathbf{w}}$ = optimal weight vector
$\mathbf{x}_i$ = i$^{\text{th}}$ vector of input pattern
$\mathbf{x}_r, \mathbf{x}_s$ = Support vectors
$\mathbf{x}^T$ = transpose of matrix $\mathbf{x}$
y_i = Target output corresponding to the i^{th} vector

Greek symbols

α, α^* = Lagrange multipliers
$\overline{\alpha}, \overline{\alpha}^*$ = Optimal Lagrange multipliers
$\beta = \alpha - \alpha^*$
$\overline{\beta} = \overline{\alpha} - \overline{\alpha}^*$
ε = insensitivity
$\Phi(\mathbf{x}_i)$ = feature space for the i$^{\text{th}}$ input vector

σ = width of the Gaussian RBF kernel
ζ, ξ_i^- , ξ_i^+ = scalar slack variables

Chapter 3

KERNEL PCA FOR FEATURE EXTRACTION AND DENOISING

3.1 Introduction

Principal component analysis, or PCA (Geladi et al., 1989), is currently used in many areas for several different applications related to detecting the underlying structure in a given data set. It provides analysis of correlations between the variables and data dimension reduction that has important benefits. First, the computational overhead of the subsequent processing stages is reduced. Second, the superimposed noise can be reduced, as the data in the last few components may be mostly due to noise. Third, a projection onto a subspace of a very low dimension is useful for easy visualization of the data. PCA has also been used in process engineering applications for data reduction, validation and visualization, fault identification, outlier detection and quality control (Kramer, 1991; Dong & McAvoy, 1996; Hiden et al., 1999; Jia et al, 2000; Nomikos & MacGregor, 1994; Wise & Gallagher, 1996).

It is well known that linear PCA utilizes second order statistics (Doymaz et al., 2001) and can extract only the linear features of the data. To capture the correct phase details of the data set and to identify the dominant features existing between variables, a nonlinear version of the PCA should be employed (Kramer, 1991). A number of methods for nonlinear generalization of linear PCA are reported in literature. Thus for instance, Kramer (1991) presented a method based on autoassociative neural network topology. Dong and McAvoy (1996) judiciously combined the principal curves algorithm (Hastie & Stuezle, 1989) and the autoassociative network (Kramer, 1991) to provide an effective algorithm for nonlinear PCA. Tan and Mavrouvouniotis (1995) proposed a nonlinear PCA method based upon the concept of the input training network. In the work of Hiden et al. (1999) genetic programming technique was used for feature extraction. Recently a novel method of performing nonlinear form of principal component analysis using integral operator kernel functions has been described (Rosipal et al., 2001; Schölkopf, et al., 1998).

Kernel PCA corresponds to linear PCA in a higher dimensional feature space, which is nonlinearly related to the input space. The input data **x** are first

mapped through some appropriate nonlinear function $\Phi(\mathbf{x})$. Then an *a priori* defined kernel function is used to deal with the possibly very high dimensional space. In other words, in kernel PCA the original problem, reformulated in the form of a dot product of the nonlinear function in the feature space, can be substituted by a kernel function, $\mathbf{K}(\mathbf{x},\mathbf{x}')=\Phi(\mathbf{x}).\ \Phi(\mathbf{x}')$. This simplifies the calculation procedure because the dot product can be computed in the input space itself. The general question of choice of different kernel functions has been discussed by Vapnik (1998). In particular, Mercer's theorem of functional analysis can be used for finding whether a function is indeed a kernel function or not. (Schölkopf, et al., 1998). Different researchers have used different kernels, the most common ones being polynomial kernels of different orders, Gaussian kernel and spline kernel. The kernel trick allows employment of algorithms such as support vector regression, kernel ridge regression etc. that can handle the dot products in the feature space for establishing the input output mapping.

In the instance of noisy data, linear PCA discards the finite variance due to noise by projection of data onto the main principal components. The same holds true for kernel PCA in feature space by using nonlinear principal components. Kernel PCA, however, extracts a substantially larger number of nonlinear principal components and therefore allows spreading the information regarding the data structure more widely giving a better opportunity to discard some of the eigen directions where the noisy part of data resides. The kernel-based treatment thus provides an attractive alternative for feature extraction and denoising. In the present work we shall illustrate these features by considering two case studies, namely (i) model noisy time series data generated using the well known Rössler and Lorenz models and (ii) prediction of dynamic mechanical properties of polymer nanocomposites. The chapter is organized as follows. Section 3.2 provides the basic framework for extracting nonlinear principal components and denoising the data set. The extracted features are then correlated to the outputs using kernel principal component regression in section 3.3. Finally section 3.4 describes the results for the case studies with comparative advantages of this methodology vis-à-vis other techniques.

3.2 Kernel Principal Component Analysis

Let us first start with a set of *M* centered data, $\mathbf{x}_k$, in the input space, *k*=1,....,*M*, $\mathbf{x}_k \in \mathbf{R}^N$. Linear principal component analysis requires the diagonalization of *M*-sample estimate of the covariance matrix

$$\mathbf{C} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_i \mathbf{x}_i^T \qquad (1)$$

with an intent to find eigenvalues $(\lambda \geq 0)$ and the associated eigenvectors **v** satisfying,

$$\lambda \mathbf{v} = \mathbf{C}\mathbf{v} \qquad (2)$$

Linear PCA is well known and the details can be found in standard works (Anderson, 1984; Jolliffe, 1986; Wold et al., 1987). It would be useful to note that for non-negative eigenvalues, all solutions must lie in the span of the input data. Thus the eigenvalue equation can be written as:

$$\lambda(\mathbf{x}_k . \mathbf{v}) = (\mathbf{x}_k . \mathbf{C}\mathbf{v}) \quad \text{for all } k=1,\ldots\ldots,M. \qquad (3)$$

For the kernelized version of PCA we first define a nonlinear mapping of the centered input data in the feature space as:

$\Phi : \mathbf{R}^N \rightarrow F$,

The problem can now be formulated as the diagonalization of the *M*- sample estimate of the covariance matrix in the high dimensional feature spaces:

$$\hat{\mathbf{C}} = \frac{1}{M}\sum_{i=1}^{M} \Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^T \qquad (4)$$

where $\Phi(\mathbf{x}_i)$ are centered nonlinear mapping of the input variables.

Here, we need to find the nonnegative eigenvalues λ and eigenvectors $\mathbf{V}$, satisfying the equation,

$$\lambda \mathbf{V} = \hat{\mathbf{C}}\mathbf{V} \qquad (5)$$

Noting that all the eigenvalues lie in the span of the transformed data in the high dimensional space, the equivalent relation can be written as:

$$\lambda\left(\Phi(\mathbf{x}_k).\mathbf{V}\right) = \left(\Phi(\mathbf{x}_k).\hat{\mathbf{C}}\mathbf{V}\right) \text{ for all } k=1,\ldots..,M. \qquad (6)$$

Also, the coefficients α's can be related to $\mathbf{V}$ as:

$$\mathbf{V} = \sum_{i=1}^{M} \alpha_i \Phi(\mathbf{x}_i) \qquad (7)$$

Combination of equations (4), (6) and (7) yields

$$\lambda \sum_{i=1}^{M} \alpha_i \left(\Phi(\mathbf{x}_k).\Phi(\mathbf{x}_i)\right) = \frac{1}{M}\sum_{i=1}^{M} \alpha_i \left(\Phi(\mathbf{x}_k).\sum_{j=1}^{M} \Phi(\mathbf{x}_j) \right)\left(\Phi(\mathbf{x}_j).\Phi(\mathbf{x}_i)\right) \quad \forall\ k=1,\ldots..,M. \qquad (8)$$

Further we define an $M \times M$ kernel matrix $\mathbf{K}$ such that

$$K_{ij} := \left(\Phi(\mathbf{x}_i).\Phi(\mathbf{x}_j).\right) \qquad (9)$$

The idea of introduction of kernel functions is now clear. It makes use of the fact that an inner product in the feature space has an equivalent kernel in the input space (Vapnik, 1998). Thus it is neither necessary to know the form of the function, $\Phi(\mathbf{x})$ nor we need to calculate the dot product in the (possibly) very high dimensional space. We can thus employ appropriate kernels to evaluate this in the input space itself. Equation (8) can now be expressed as

$$M\lambda \mathbf{K}\boldsymbol{a} = \mathbf{K}^2\boldsymbol{a} \qquad (10)$$

By definition **K** is a symmetric matrix and thus it has a set of eigenvectors which span the whole space and thus

$$M\lambda \boldsymbol{a} = \mathbf{K}\boldsymbol{a} \qquad (11)$$

By definition **K** is positive semi definite, with nonnegative eigenvalues. We therefore only need to diagonalise **K**. Let $\lambda_1 \le \lambda_2 \le ..... \le \lambda_M$ denote the eigenvalues, and $\boldsymbol{a}^1, \boldsymbol{a}^2, ...\boldsymbol{a}^M$ the corresponding complete set of eigenvectors, with λ_p being the first nonzero eigenvalue. We normalize $\boldsymbol{\alpha}^p, ..., \boldsymbol{\alpha}^M$ by requiring that the corresponding vectors in *F* be normalized, i.e.

$$\left(\mathbf{V}^k . \mathbf{V}^k\right) = 1 \quad \forall k = p, ..., M$$

From equations (7) and (11), we have the normalization condition for $\boldsymbol{a}^p, ..., \boldsymbol{a}^M$:

$$\begin{aligned} 1 &= \sum_{i,j}^{M} \alpha_i^k \alpha_j^k \left(\Phi(\mathbf{x}_i).\Phi(\mathbf{x}_j)\right) \\ &= \sum_{i,j}^{M} \alpha_i^k \alpha_j^k \left(K_{i,j}\right) \\ &= \left(\boldsymbol{a}^k . \mathbf{K}\boldsymbol{a}^k\right) \\ &= \lambda_k \left(\boldsymbol{a}^k . \boldsymbol{a}^k\right) \end{aligned}$$

For the purpose of principal component extraction, we need to compute the projections on the eigenvectors $\mathbf{V}^k$ in *F (k=p,….,M)*:

$$\beta(\mathbf{x})_k = (\mathbf{V}^k . \Phi(\mathbf{x})) = \sum_{i=1}^{M} \alpha_i^k K(\mathbf{x}_i, \mathbf{x}) \qquad (12)$$

It is evident from equation (12) that by using the kernel trick the eigenvalues are now characterized by the corresponding α vectors, which can be

used to find the principal components. We then select the first $p<M$ nonlinear principal components, e.g. the directions that describe a desired percentage of data variance, and thus work in the p-dimensional sub-space of feature space. An inspection of equation (12) further reveals that in kernel PCA the number of principal components extracted can exceed the input dimensionality. Thus for a system with M data observations and N input dimensionality (N variables), the kernel PCA can find up to M nonzero eigenvalues. This is in contrast to linear PCA in which we can extract only N nonzero eigenvalues.

It must be mentioned here that in linear PCA it is possible to recover original input data from complete set of extracted principal components. In kernel PCA this may not be possible because the eigenvectors V in feature space do not have a preimage in input space. So for the purpose of data reconstruction we may have to perform a regression connecting the projected data in the feature space to the vector of M observations y. This can be done by various methods like SVM regression, kernel ridge regression and kernel principal component regression (Rosipal, et al 2001; Schölkopf, Smola, & Müller, 1998). For the purpose of denoising the time series in this work we have employed the kernel principal component regression.

The key difference between kernel PCA and linear PCA is in the extraction of principal components. For a data consisting of M test examples with input dimensionality N the linear PCA can extract a maximum of N principal components while the kernel PCA can extract up to M principal components. This may indeed be a limitation for some dimensionality reduction problems. On the other hand in certain classification problems it has also been found that after the initial kernel PCA preprocessing even a computationally cheap linear classifier could work quite efficiently (Schölkopf, Smola, & Müller, 1998). Kernel PCA can also have a definite advantage in dealing with multi-collinearity and noise. While dealing with multi-collinearity kernel PCA also allows us more flexibility in retaining principal components to capture the underlying nonlinear features.

Finally, a word about the computational complexity involved is in order. It is clear that the search for the principal components in the high dimensional feature space may not lead to computational problems because we do not need to

look for eigenvectors in the full space, but only in the subspace spanned by the images of observations. Also, the computation of dot products can be performed in the input space using kernel function. Thus in practice the computational loads for kernel PCA and linear PCA are of the same order of magnitude. For extracting principal components, however, we need to evaluate kernel function *M* times over each extracted principal component. The kernel principal component extraction is thus computationally more involved than its linear counterpart, but the efforts could be compensated, as even in nonlinear cases, construction of simple linear regression may be sufficient. Also, the complexity involved while dealing with larger dimensions can be solved by introducing different sparse approximations (Schölkopf, Smola, & Müller, 1998).

3.3 Kernel Principal Component Regression

Consider the standard regression model in feature space *F*,

$$\mathbf{y} = \Phi\xi + \varepsilon, \qquad (13)$$

where ***y*** is a vector of *M* observations of the dependent variable, Φ is an $M \times n$ matrix ($n \leq \infty$) of regressors whose i^{th} row is the vector $\Phi(\mathbf{x}_i)$ of the mapped $\mathbf{x}_i$ observation into the high dimensional feature space *F* and , ξ is a vector of regression coefficients and $\boldsymbol{\varepsilon}$ is the vector of error terms. The fact that $\Phi^T\Phi$ is proportional to the sample covariance matrix can be exploited to extract upto *n* eigenvalues $\{\lambda_j\}_{j=1}^{n}$ and corresponding eigenvectors $\{\mathbf{V}^J\}_{j=1}^{n}$. The projection $\Phi(\mathbf{x})$ onto the *k*-th nonlinear principal component was given in equation 12. Projection of all regressors on to the principal components yield

$$\mathbf{y} = \mathbf{B}\mathbf{w} + \boldsymbol{\varepsilon}, \qquad (14)$$

where $\mathbf{B}=\Phi\mathbf{V}$ is now a matrix of transformed regressors and $\mathbf{V}$ is an $(n \times n)$ matrix whose *k*-th column is the eigenvector $\mathbf{V}^k$. The columns of the matrix are orthogonal and least squares estimate of the coefficients **w** becomes

$$\mathbf{w} = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\mathbf{y} \qquad (15)$$

The kernel matrix should be centralized before it is used to find the principal components:

$$\mathbf{K} = \left(\mathbf{I} - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T\right)\mathbf{K}\left(\mathbf{I} - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T\right) \qquad (16)$$

$$\mathbf{K}_t = \left(\mathbf{K}_t - \frac{1}{M}\mathbf{1}_{M_t}\mathbf{1}_M^T\mathbf{K}\right)\left(\mathbf{I} - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T\right) \qquad (17)$$

where $\mathbf{K}$ is the kernel of the input data (training) matrix, $\mathbf{K}_t$ is the kernel of the input data (testing) matrix, $\mathbf{I}$ the identity matrix of order M, and $\mathbf{1}_M$, $\mathbf{1}_{M\,t}$ are the vectors whose elements are the ones with length M, M_t respectively. To avoid the problem of multi-collinearity we employ only the first p principal components. The linear kernel principal component regression model using the first p ($p \le M$) nonlinear principal components in terms of the kernel matrix can finally be expressed as :

$$f(\mathbf{x},\mathbf{a}) = \sum_{i=1}^{p} w_k \beta_k(\mathbf{x}) + b = \sum_{k=1}^{p} w_k \sum_{i=1}^{M} \alpha_i^k K(\mathbf{x}_i,\mathbf{x}) + b = \sum_{i=1}^{M} a_i K(\mathbf{x}_i,\mathbf{x}) + b \qquad (18)$$

where $\left\{a_i = \sum_{k=1}^{p} w_k \alpha_i^k\right\}_{i=1}^{M}$ and b is bias term.. For the centralized regression model bias is zero.

3.4 Case Studies

The denoising and prediction capabilities of the kernel PCA algorithm have been tested by considering two different examples. In one case study chaotic time series data (Rössler and Lorenz model) has been used for testing denoising performance. The other case study deals with the development of a data driven model connecting the important input variables to the mechanical properties of polymeric nanocomposites developed in our laboratory. More details about the case studies

are given in the following subsections. For both case studies we have used Gaussian kernel, which is defined as $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{L}\right)$ where L is the width of the Gaussian function.

3.4.1 Denoising of chaotic Time series

In our simulations we have chosen two important benchmarking time series examples, viz., the Rössler system (Killory et al., 1986, 1987) and the Lorenz system governed by the following set of equations.

Rössler system

$$\frac{dx}{dt} = -z - y$$
$$\frac{dy}{dt} = x + ay$$
$$\frac{dz}{dt} = b + z(x - c)$$

with $a = 0.15;\ b = 0.2;\ c = 10$.

Lorenz system

$$\frac{dx}{dt} = -\sigma x + \sigma y$$
$$\frac{dy}{dt} = R x - y - x z$$
$$\frac{dz}{dt} = -b z + x y$$

with $\sigma = 10, R = 28, b = 8/3$.

The time series data was generated by integrating both sets of equations using a standard Runge-Kutta routine with a step size of 0.01. The training set consisted of 450 delay vectors, formed by using an embedding dimension of 3 and a time delay of 23 for Rössler series (Casdagli,1989 ; Fraser & Swinney ,1986) and 16 for Lorenz series. The test set consisted of similarly embedded 200 sets.

Gaussian noise with three different noise to signal ratio was added to the samples of Rössler and Lorenz series. Kernel PCA preprocessing followed by

kernel principal component regression for data reconstruction was carried out with a view to minimizing the RMS error in retrieving the clean data. The results for the two cases are shown in Tables 3.1a and Table 3.1b. It can be seen that for a noise to signal ratio of 5% the best results were obtained by retaining the first 152 principal components and the optimal width of Gaussian function was 1.84 for Rössler series where as for Lorenz series best results were obtained by retaining the first 65 principal components and the optimal width of Gaussian function was 1.1. The optimal number of principal components to be retained reduced with increase in noise as observed for both the cases. It can also be observed that there is a gradual increase in both the test and training errors with the increase in noise. The influence of retaining different number of principal components is shown in Figure 3.1 for a noise to signal ratio of 10% for Rössler series. Similar trend was observed for the other cases. Figure 3.1 indicates that there exist an optimal number of principal components to be retained. If more number of components is retained than the optimal number the noise elimination capability is reduced. On the other hand retaining lesser number of components leads to loss of information.

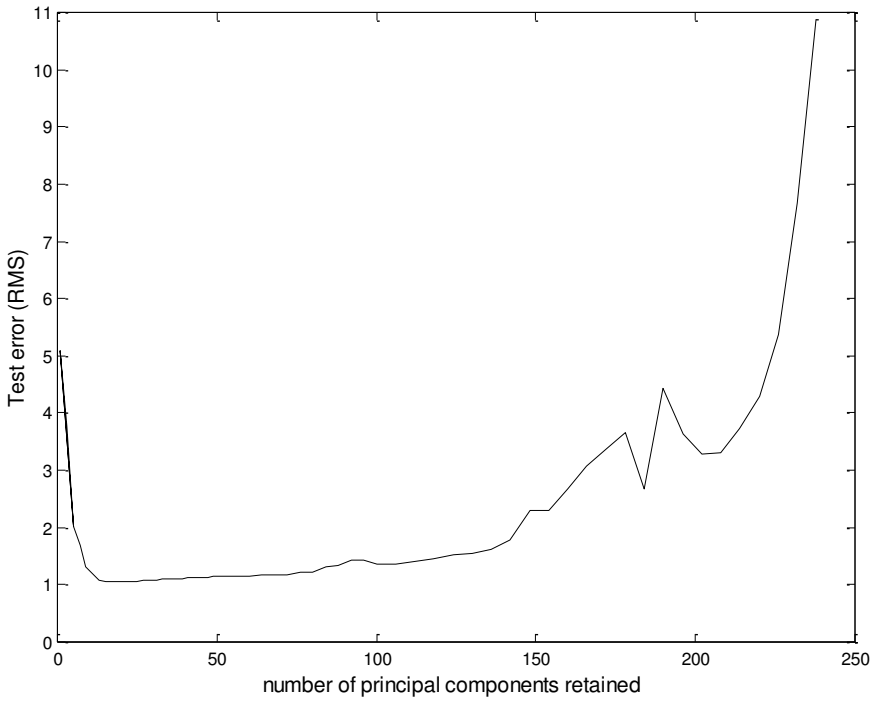


Figure 3.1: Effect of number of principal components on RMS test error for case study 1 (n/s = 10 %,σ =2.0).

For the sake of comparison, denoising was also carried out using linear PCA, wavelets (Daubechies, 1992) and moving median filter denoising techniques. The Daubechies wavelet technique is extensively used in engineering applications. (Doymaz et al., 2001). In the moving median (MM) filter technique the median of a window containing odd number of observations is found by sliding the window over the entire one-dimensional signal. The details of this method can be found in (Turkey, 1970; Davies, 1992; Doymaz et al., 2001). Linear PCA did not show any improvement with reduction in number of principal components. The results using wavelets (db4) and MM filter are marginally better than kernel PCA for Rössler time series (Table 3.2a), while for Lorenz time series (Table 3.2b) kernel PCA outperforms the other methods for low noise to signal ratio. For higher noise to signal ratio, the results of wavelet and kernel PCA are almost the same. For this time series kernel PCA does better than MM filters for both high and low noise to signal ratios.

Table 3.1a: Best results for Rössler time series

Level of noise	Width of Gaussian function	Number of Principal components retained	Training error (RMS)	Testing error (RMS)
n/s=0% (Clean time series)	1.0	136	0.001396	0.002317
n/s = 5%	1.3	50	0.553745	0.586854
n/s = 10%	2.0	23	1.063071	1.057948
n/s=15%	5.0	19	1.497384	1.581846

Table 3.1b: Best results for Lorenz time series

Level of noise	Width of Gaussian function	Number of Principal components retained	Training error (RMS)	Testing error (RMS)
n/s=0% (Clean time series)	1.0	164	0.019843	0.054608
n/s = 5%	1.1	65	0.696934	0.866639
n/s = 10%	1.2	31	1.393587	1.542977
n/s=15%	2.0	30	1.867302	1.865162

Kernel PCA's ability to extract up to M principal components indeed offers greater maneuverability in optimizing the denoising capability. Although a direct comparison with neural network based PCA is not possible it can be said that the neural networks based nonlinear PCA requires solution of a nonlinear optimization problem and can possibly get stuck in a local optima. Kernel PCA on the other hand is essentially a simple linear PCA in high dimensional feature space and does not require solution of a nonlinear optimization problem. In neural networks based PCA additional efforts have to be taken to orthogonalize the principal components (Doymaz et al., 2001), whereas kernel PCA is an orthogonal basis transformation in high dimensional feature space and the first q principal components carry more variance and more information content than any other q orthogonal directions (Rosipal, et al 2001; Schölkopf, Smola, & Müller, 1998).

Table 3.2a: Comparison with other denoising methods: Rössler series

Noise/signal ratio	*RMS error*		
	Kernel PCA	Wavelets (db4)	MM filter
5 %	0.586854	0.3439	0.3788
10 %	1.051539	0.5873	0.6319
15 %	1.577870	0.8812	0.8835

Table 3.2b: Comparison with other denoising methods: Lorenz series

Noise/signal ratio	*RMS error*		
	Kernel PCA	Wavelets (db4)	MM filter
5%	0.8623	1.426158	2.5833
10%	1.542977	1.44277	2.6614
15%	1.865162	1.723761	2.7159

3.4.2 Prediction of dynamic mechanical properties of polyvinyledene fluoride (PVDF)/Clay Nanocomposites

Polymer/clay nanocomposites represent a new class of materials with many desirable attributes and enhanced performance such as higher modulus, improved dimensional stability, decreased thermal expansion coefficient, increased solvent resistance, enhance ionic conductivity, and reduced gas permeability. In particular (PVDF)/clay nanocomposites prepared by melt intercalation with organophilic clay exhibits highly improved thermo-mechanical properties. Dynamic Mechanical properties such as storage modulus, loss modulus, and tan δ of organophilic clays with different compositions and particle sizes were measured at constant frequency over a wide spectrum of temperatures. A data driven model connecting the various inputs to the mechanical properties would be very useful in selecting the composites for various end uses. It is with this aim kernel PCA was employed to suitably arrive at a highly accurate input output mapping. The experimental procedure is as follows:

PVDF grade SOLEF 1008 supplied by solvay (Belgium) (M_w=100x10^3 g/mol and M_w/M_n = 2.5) was used. Organically modified clay, Cloisite 20A (Ditallowdimethylammonium salts with Bentonite), was generously supplied by Southern Clay Products, Texas. The clay was dried in an air circulatory oven at 60^0C. Four compositions containing 1.5, 3, 5, and 7% (wt/wt) of clay were prepared using melt mixing in Brabender Plasicorder mixer at 200^0C and 60 rpm. The corresponding particle size for the above compositions are measured experimentally and found to be 10.5, 12.31, 13.52 and 12.31 nm. The films used for mechanical analysis were prepared by compression molding at a temperature of 200^0C using Caver press model F-15181. The structure of the polymer/clay composite was evaluated using Rigaku model Dmax 2500 X-ray Diffractometer with Cu-Kα radiation of wavelength 0.1514 nm. The basel spacing of the clay was estimated from the d(001) peaks in the XRD pattern. Particle size measurement is carried out for the polymer clay nanocomposites using Scherrer equation.

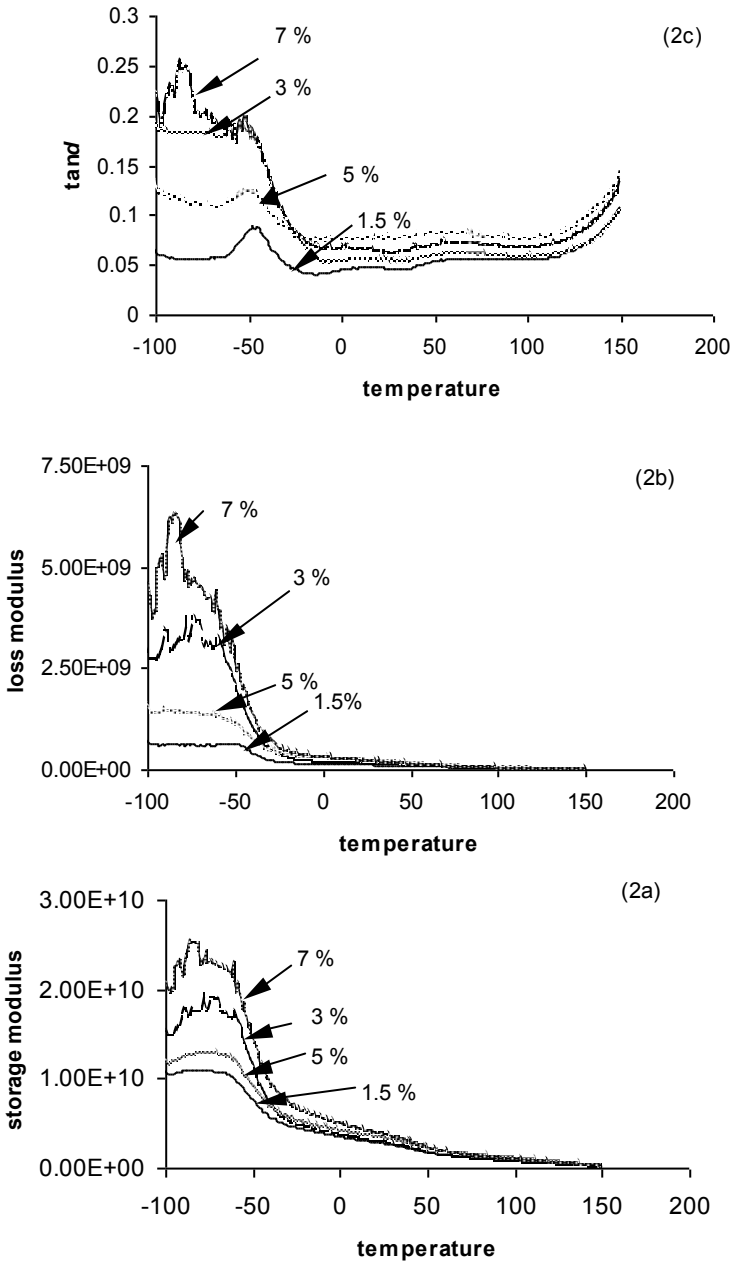


Figure 3.2: Variation of mechanical properties of polymer nanocomposites with temperature and composition.

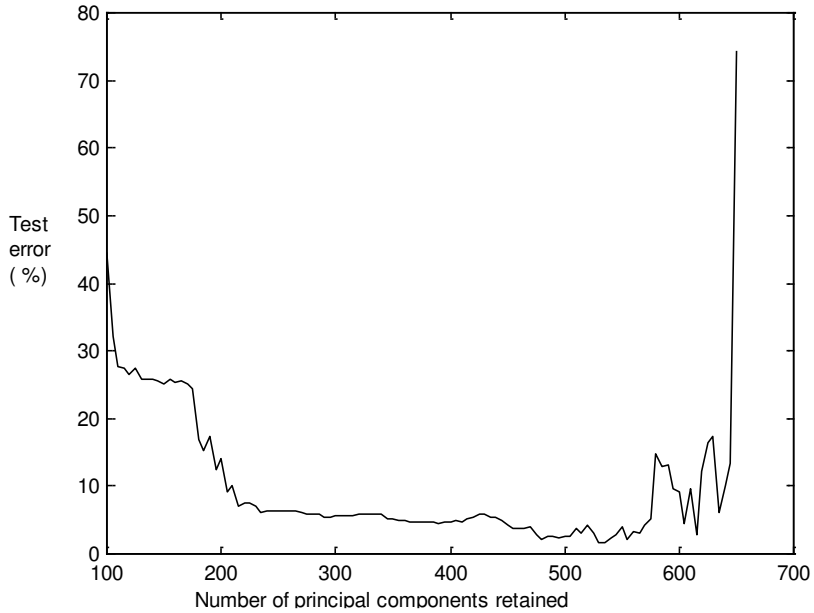


Figure 3.3: Effect of number of principal components retained on test error in predicting storage modulus of polymer nanocomposites (σ = 2.75)

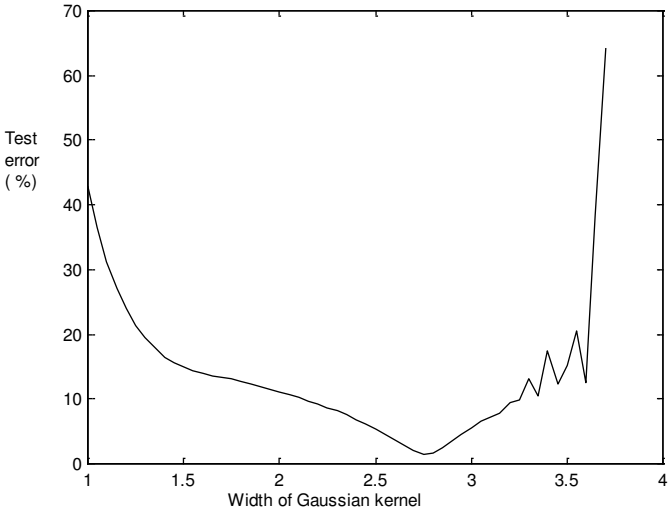


Figure 3.4: Effect of width of Gaussian function on test error in predicting storage modulus of polymer nanocomposites (number of principal components retained = 535)

The dynamic Mechanical properties of the samples were studied on compression-molded films using a dynamic mechanical analyzer Rheometrics model DMTA IIIE in the tensile mode. The samples were analyzed from –100 to 150^0C at a heating rate of 5^0C/min and frequency of 10 rad/s. The temperature sweep was carried out at 0.1 % strain. The storage modulus, loss modulus, and tan

δ were measured at constant frequency over the entire temperature range and the experimental data are presented in Figure 3.2.

The organically modified clays exhibit significant variations in their surface properties, which in turn affect the size, size distribution, and mechanical properties of the nanocomposites. In this case study, the composition of clay, size of clay particles and the temperature at which the mechanical tests were conducted provides the input variables space. The output variables are the storage modulus, loss modulus, and tan δ, representing the ratio of the two modulli. Only the first and third variables were therefore considered for the simulations. The input-output mapping was done separately with each of the two outputs. Out of a total number of 1167 data points 900 were taken for training and the remaining were employed for testing. The highly nonlinear interactions coupled with inevitable instrumental measurement errors make the data very difficult to analyze using linear methods. Preliminary analysis using linear PCA show the training and test errors to be in the range of 30-35 % for predicting tan δ and more than 130% for predicting storage modulus. The detailed results using linear principal component regression are shown in Table 3.3.

Table 3.3: Results for predicting properties of polymer nanocomposites using principal component regression

Property of polymer nanocomposite	Principal components retained	Test error (%)	Training error (%)
Storage modulus	1	254.681358	237.988341
	2	123.155203	128.387774
	3	127.638248	131.061781
$\tan\delta$	1	30.758271	29.124107
	2	30.833133	32.658523
	3	30.906240	32.404219

Table 3.4: Best results for predicting properties of polymer nanocomposites using kernel principal component regression

Property of polymer nanocomposite	Width of Guassian kernel	Number of Principal components retained	Training error	Testing error
Storage modulus	2.75	535	0.366 %	1.338 %
tan δ	1.63	801	0.119 %	2.330 %

Extensive simulations were carried out for various values of the two parameters, viz., number of principal components to be retained and the Gaussian function width. The best results obtained are shown in Table 3.4. For mapping the input variables with the storage modulus we found that optimal results were obtained by retaining the first 535 principal components and employing a value of 2.75 for the kernel width. For relating the tan δ with the input variables the optimal kernel width and the number of principal components were found to be 1.63 and 801 respectively. The effect of number of principal components retained on test error in predicting storage modulus is shown in Figure 3.3. The effect of Gaussian function width on the test error is depicted in Figure 3.4. The results indicate similar trends with those exhibited in previous case study.

3.5 Summary

Kernel PCA, a new method for performing nonlinear principal component analysis has been illustrated by considering the examples of (i) denoising of chaotic time series and, (ii) development of an input-output model for the case of polymer nanocomposites. In this method the original problem is first nonlinearly transformed to a higher dimensional space. The kernel function simplifies the computational complexities by performing the dot product of the transformed data in the input space itself. The capability of the method to extract a large number of principal components is very useful for feature extraction and denoising. For the chaotic time series the kernel PCA successfully denoises and recovers the original data with substantial accuracy. Similarly for the polymer nanocomposite example

the kernel PCA preprocessing followed by kernel regression is able to extract the dominant features and map the input output data very well. The fact that the method does not require solution of any hard nonlinear optimization problems makes the method very attractive for use in various process engineering applications.

References

Anderson, T. W. , "Introduction to multivariate statistical analysis", 2nd ed., New York: Wiley. (1984).

Casdagli, M., "Nonlinear prediction of chaotic time-series", Physica D 35, 335-356, (1989).

Daubechies, "Ten lectures on wavelets", Philadelphia: SIAM, (1992).

Davies, E. R., " The relative effects of median & mean filters on noisy signals", Journal of Modern Optics 39, 103-113, (1992).

Dong, D., & McAvoy, T. J. , "Non-linear principal component analysis- based on principal curves & neural networks", Computers & Chemical Engineering, 20, 65-78, (1996).

Doymaz, F., Chen, J., Romagnoli, J. A., & Palazoglu, A., "A robust strategy for real time process monitoring", Journal of Process Control, 11(4), 343-359, (2001).

Fraser, A. M., & Swinney, H. L., "Independent coordinates for strange attractors from mutual information", Physical Review A 33, 1134-1140, (1986).

Geladi, P., Isaksson, H., Lindqvist, L., Wold, S., & Esbensen, K., "Principal components analysis of multivariate images", Chemometrics & Intelligent Laboratory Systems, 5(3), 209-220, (1989).

Hastie, T., & Stuezle, W. , “Principal curves”, Journal of American Statistical Association, 84(406), 502-516, (1989).

Hiden, H. G., Willis, M. J., Tham, M. T., & Montague, G. A., “Non-linear principal components analysis using genetic programming”, Computers & Chemical Engineering, 23, 413-425, (1999).

Jia, F., Martin, E. B., & Morris, A. J. , “Non-linear principal component analysis with application to process fault detection”, International Journal of System science, 31(11), 1473-1487, (2000).

Jolliffe, I. T., “Principal component analysis”, New York: Springer-Verlag, (1986).

Killory, H., Hudson, J., & Rössler, O., “Chaos in a four-variable chemical reaction system”, Chemical Engineering Communications 46, 159, (1986).

Killory, H., Hudson, J., & Rössler, O. , “Higher chaos in a four-variable chemical reaction model”, Physics Letters A 122, 341, (1987).

Kramer, M. A. , “Nonlinear principal component analysis using autoassociative neural networks”, AIChE. Journal, 37(2), 233-243, (1991).

Nomikos, P., & MacGregor, J. F., “ Monitoring batch processes using multiway principal component analysis”, AIChE. Journal, 40(8), 1361-1373, (1994).

Rosipal, R., Girolami, M., Trejo, L. J., & Cichocki, A., “Kernel PCA for feature extraction & de-noising in non-linear regression”, Neural Computing & Applications, 10(3), 231-243, (2001).

Schölkopf, B., Smola, A., & Müller, K. R. , „Nonlinear component analysis as kernel eigenvalue problem”, Neural Computation, 10(5), 1299-1319, (1998).

Tan, S., & Mavrovouniotis, M. L., "Reducing data dimensionality through optimizing neural network inputs", AIChE Journal, 40, 1471-1480, (1995).

Turkey, J. W., "Exploratory data analysis", Reading, MA: Addison-Wesley, (1970).

Vapnik, V. , "Statistical learning theory", New York: Wiley. (1998).

Wise, B. M., & Gallagher, N. B. , "The process chemometrics approach to process monitoring & fault detection", Journal of Process Control, 6, 329-428, (1996).

Wold, S., Esbensen, K., & Geladi, P. , "Principal component analysis", Chemometrics Intelligent Laboratory Systems, 2, 37-52, (1987).

Chapter 4

A METHODOLOGY FOR PROCESS MONITORING USING LLE-SVDD

4. 1. Introduction

Last few years have experienced an explosive growth in the amount of data collected on different experimental systems due to availability of sophisticated instrumentation. New applications that require the storage and retrieval of huge amounts of data are emerging. They include examples such as protein matching in biomedical applications, fingerprint recognition, meteorological predictions, satellite image repositories, genomic data, text categorization etc. Most problems of interest in practice involve data with a large number of measurements (or dimensions). For example in many chemical process plants, sensors provide a large amount of measurements (features). This information overload can be a significant problem for plant operators responsible for insuring the safety and economic operation of the plant, particularly during abnormal situations resulting from disturbances, faults (sensor, equipment failure etc), human error, and/or unanticipated operating conditions. Thus abnormality detection in process plant constitutes a very vital aspect of safe and optimal operation of complex chemical plants. A number of methods have been proposed for batch process monitoring.

The early work, based on multiway principal component analysis (MPCA) (Wold et. al., 1987), was developed for batch process monitoring by Nomikos and MacGregor (1994). Multiway partial least square (MPLS) was then developed to correlate the process data and the product quality data (Nomikos & MacGregor, 1995). Various researchers have proposed several variants to the original methodology based on MPCA (e.g. Rannar et al. 1998). Louwerse and Smilde (2000) used PARAllel FACtor analysis (PARAFAC) and Tucker three-way models for monitoring batch processes. Boque and Smilde (1999) used multivariate statistical procedures based on multiway covariates regression models. Nonlinear principal component analysis (NLPCA) based on principal curves and neural networks produced independent principal components to unfold batch process data and get the nonlinear batch trajectory (Dong & McAvoy, 1996b). Martin and Morris (1996) and Martin et al. (1996) introduced a control

chart based on a non-parametric method. Wise et al. (1999), Westerhuis et al. (1999) and Dahl et al. (1999) applied and compared several alternatives for multivariate statistical analysis of batch process data. Combination of the orthonormal function approximation and the MPCA is proposed to analyze and monitor batch processes at the different operating time (Chen & Liu, 2000). Dynamic PCA and dynamic PLS models have also been used for on-line batch process monitoring (Chen & Liu, 2002). The performance of statistical process monitoring of batch processes can be enhanced by incorporating external information in model development (Ramaker et al., 2002). Some of these approaches have been evaluated by Van Sprang et al. (2002). Also there are number of approaches, which has been used for fault detection and diagnosis of the batch as well as continuous process using artificial neural networks (ANN) and combination of ANN with fuzzy and wavelets (Chen et. al. 1999; Dong & McAvoy, 1996a; Dong & McAvoy, 1996b; Fan et. al., 1993; Farell & Roat, 1994; Hoskins et. al. 1991; Kavuri & Venkatasubramanian ,1993, 1994; Rengaswamy & Venkatasubramanian, 2000; Ruiz et. al. 2000,2001; Scenna, 2000; Ungar et. al. 1990; Venkatasubramanian & Chan, 1989; Venkatasubramanian, et. al., 1990; Wang et. al., 1999; Watanabe et. al. 1989; Zhao et. al., 1997).

Many of the techniques elaborated above are based on the use of clustering and dimensionality reduction valid for linear structures (Devijver and Kittler, 1982; Duda et al., 2001; Jain et al. 2000; Mardia et al. 1979). Many real life data sets however contain essential nonlinear structures that are imperceptible to linear methods (Bailer-Jones et al. 1998; McClurkin et al. 1991; Murase & Nayar, 1995). A number of techniques to perform nonlinear mappings have been proposed in literature. They include: non-linear PCA (Malthouse, 1998), multi-dimensional scaling (MDS) (Borg and Groenen,1997), Sammon mapping (Sammon, 1969), singular value decomposition (SVD), self-organizing map (SOM) (Kohonen, 1995), generative topographic mapping (Bishop et al.,1998), principal curves and surfaces (Hastie and Stuetzle, 1989), auto-encoder neural networks (DeMers & Cottrell, 1993), mixtures of linear models (Tipping & Bishop 1999) etc. All of these methods while extremely useful in general, have some or the other specific limitation. Thus for instance: there is no single and unique solution to nonlinear PCA while MDS and Sammon mapping give a point-

to-point mapping but cannot provide the underlying mapping function. Consequently they cannot accommodate new data points (Sammon, 1969; Mao & Jain, 1995) and the entire procedure has to be repeated from start using all data points. Multi-dimensional scaling and neural networks are hard to train and time-consuming, as are principal curves and surfaces. The latter, as well as the generative topographic mapping, need large data sets to estimate their many parameters. Mixtures of localized linear models require the user to set a number of parameters, which are highly specific to each data set and determine how well the model fits the data.

Recently, several entirely new approaches have been devised to address these problems. These methods combine the advantages of PCA and MDS viz. computational efficiency; few free parameters; non-iterative global optimization of a natural cost function—with the ability to recover the intrinsic geometric structure of a broad class of nonlinear data manifolds. These algorithms can be local or global. Local approaches such as locally linear embedding (LLE) (Roweis & Saul, 2000), Laplacian eigenmaps (Belkin & Niyogi, 2002) attempt to preserve the local geometry of the data; essentially, they seek to map nearby points on the manifold to nearby points in the low-dimensional representation. Global approaches such as Isomap (Tenenbaum et al., 2000) attempts to preserve geometry at all scales, mapping nearby points on the manifold to nearby points in low-dimensional space, and faraway points to faraway points. Thus isomap preserves the neighborhood of each object, as well as the 'geodesic' distances between all pairs of objects. The global approach tends to give a more faithful representation of the data's global structure, and its metric-preserving properties are better understood theoretically. The local approaches have two principal advantages: (1) computational efficiency: they involve only sparse matrix computations which may yield a polynomial speedup; (2) representational capacity: they may give useful results on a broader range of manifolds, whose local geometry is close to Euclidean, but whose global geometry may not be.

LLE recovers global nonlinear structure from locally linear fits. Unlike clustering methods for local dimensionality reduction, LLE maps its inputs into a single global coordinate system of lower dimensionality, and its optimizations

does not involve local minima. LLE is based upon reconstruction of data, preserving local neighborhoods, and thus also the clusters which may be present in the database. Therefore algorithms such as Support Vector Domain Distribution (SVDD) (Tax and Duin, 1999) should show superior performance for LLE data representation than other representations like PCA. In the present work, we illustrate these advantages of LLE combined with SVDD to make the abnormality detection scheme more robust.

4.2. Locally Linear Embedding (LLE) Algorithm

The LLE algorithm is based on simple geometry. Consider the data set $\{\overset{\rho}{X}\}_{i=1,2.....P} \in \Re^D$, sampled from some smooth underlying manifold. For a well sampled (i.e. there is enough data) manifold, we expect each data point and its neighbors to lie on or close to a locally linear patch of the manifold. We can thus approximate the non-linear manifold in the vicinity of $\overset{\nu}{X}_i$ by a linear hyperplane passing through its nearest neighbors. In the simplest formulation of LLE, one identifies *N* nearest neighbors for every data point, as measured by Euclidean distance (Other notions of "closeness" are also possible, such as all points within a certain radius, or by using more sophisticated rules based on local metrics.) and then minimize the reconstruction error as measured by a cost function

$$\varepsilon(W) = \sum_i \left| \overset{\rho}{X}_i - \sum_j W_{ij} \overset{\rho}{X}_j \right|^2 \qquad (1)$$

subject to two constraints:

a) Each data point $\overset{\nu}{X}_i$ is reconstructed only from its neighbors, enforcing $W_{ij} = 0$ if $\overset{\nu}{X}_j$ does not belong to this set and

b) $\sum_j W_{ij} = 1$ for every i.

The weights W_{ij} signify the contribution of the j^{th} data point to the i^{th} reconstruction. The optimal weights W_{ij} subject to these constraints are found by solving a least squares problem. The constrained weights that minimize these reconstruction errors characterize intrinsic geometric properties of each

neighborhood, as opposed to properties that depend on a particular frame of reference. This is due to the fact that for any particular data point, the weights are invariant to rotations, rescalings, and translations of that data point and its neighbors. The invariance to rotations and rescalings results from the form of Equation (1); the invariance to translations is imposed by the sum-to-one constraint (b). Since the data lie on or near a smooth nonlinear manifold of dimensionality d<<D, there exists a linear mapping— comprising a translation, rotation, and rescaling—that maps the high dimensional coordinates of each neighborhood to global internal coordinates on the manifold. Thus reconstruction weights W_{ij}, invariant to such transformations, should characterize the local geometry, both in the original data space and local patches on the manifold. In particular, the same weights W_{ij} that reconstruct the i^{th} data point in D dimensions should also reconstruct its embedded manifold coordinates in d dimensions. Based on this idea each high dimensional observation $\vec{X}_i$ is mapped to a low dimensional vector $\vec{Y}_i$ representing global internal coordinates on the manifold. This is accomplished by choosing d dimensional coordinates $\vec{Y}_i$ to minimize the reconstruction errors as measured by embedding cost function:

$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2 \tag{2}$$

The embedding cost defines a quadratic form in the vectors $\vec{Y}_i$.

$$\Phi(Y) = \sum_{ij} M_{ij} (\vec{Y}_i . \vec{Y}_j) \tag{3}$$

Here M is $P \times P$ matrix:

$$M_{ij} = \delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj} \tag{4}$$

where δ_{ij} is 1 if $i = j$ and 0 otherwise.

To ensure the uniqueness of the solution the following two constraints are imposed: translation invariance by requiring the coordinates to be centered on the origin i.e. $\sum_i \vec{Y}_i = 0$ and we constrain the embedding vectors to have unit covariance,

$$\frac{1}{P}\sum_i \vec{Y}_i.\vec{Y}_i^T = \mathrm{I} \qquad (5)$$

where I is the $d \times d$ identity matrix.

These constraints do not affect the generality of the solutions as $\Phi(Y)$ is invariant to translation, rotations and homogeneous rescalings. The additional constraint that the covariance is equal to the identity matrix expresses an assumption that reconstruction errors for different coordinates in the embedding space should be measured on the same scale.

The optimal embedding $\vec{Y}_{i=1,2.....P} \in R^d$ is given by eigenvectors associated with the smallest d+1 eigenvalues of the matrix M (Horn & Johnson, 1990). The bottom eigenvector of this matrix is discarded, as it is a vector composed of all ones, with zero as eigenvalue. Discarding this eigenvector enforces the constraint that the embeddings have zero mean, as the components of other eigenvectors must sum to zero, by virtue of orthogonality.

The bottom d+1 eigenvectors (corresponding to smallest d+1 eigenvalues) of the matrix M can be determined without performing a full matrix diagonalization (Bai et al., 2000). Moreover, the matrix M can be stored and manipulated as the sparse symmetric matrix

$$M = (\mathrm{I} - W)^T (\mathrm{I} - W) \qquad (6)$$

giving substantial computational savings for large values of P.

Although the reconstruction weights for each data point are computed from its local neighborhood independently, the embedding coordinates are computed by an $P \times P$ eigensolver, a global operation that couples all data points in connected components of the graph defined by the weight matrix. The different dimensions in the embedding space can be computed successively; this is done simply by computing the bottom eigenvectors from Equation (2) one at a time.

The nearest neighbor parameter N is a measure of the "quality" of input-output mapping (i.e. how well the high-dimensional structure is represented in the

embedded space). If N is set too small, the mapping will not reflect any global properties; if it is too high, the mapping will lose its nonlinear character and behave like traditional PCA, as the entire data set is seen as local neighborhood. N is selected based on the residual variance (Kouropteva et al., 2002). It is defined as $1-\rho^2_{E_xE_y}$ where ρ is the standard linear correlation coefficient, computed over all entries of E_x and E_y; E_x and E_Y are the matrices of Euclidean distances (between pairs of points) in X and Y (as computed above), respectively. The lower the residual variance is, the better the high-dimensional data are represented in the embedded space. Hence, the optimal value for N, N_{opt} can be determined as

$$N_{opt} = \arg\min_{N}(1-\rho^2_{E_xE_y}) \qquad (7)$$

A few techniques like linear interpolations and training a neural network or RBF network (Vlachos et al., 2002) are available for mapping a new (previously unseen) sample. In the present work we have however preferred a simple strategy of concatenating the new sample with given samples and repeating the whole LLE procedure for on-line implementation. This preference is based on our observation that the LLE algorithm takes only few seconds of time to run (as LLE only involves sparse matrix computations), retaining non-linear mapping even for query point. Approaches like Neural or RBF networks are hard to train and linear interpolations may lose the non- linearity of data.

The procedure as described above leads to nonlinear dimensionality reduction of data. We shall now briefly describe the classification using SVDD.

4.3. Support Vector Domain Distribution

Support vector domain distribution (Tax and Duin, 1999) avoids solving the harder density estimation problem and uses the simple task of finding the support vectors of the multivariate distribution. The objective of classification of data domain is that the given set of data should be represented in a unique minimal volume spherical domain enclosing all or nearly all the training points. The effect of outliers is reduced by using slack variables ξ_i to allow for data points outside

the sphere and task is to minimize the volume of the sphere and number of data points outside the sphere.

$$F(R,\vec{a},\xi_i) = R^2 + C\sum_i \xi_i \qquad i = 1,2.....P \tag{8}$$

with constraints

$$(\vec{x}_i - \vec{a})^T(\vec{x}_i - \vec{a}) \le R^2 + \xi_i \qquad \forall \xi_i \ge 0 \tag{9}$$

P is the number of objects in training set and $\vec{a}$ is the center of the sphere. The parameter C characterizes the trade off between the volume of sphere and number of data points that lie outside.

Combining (7) & (8), we formulate the Lagrangian as,

$$L(R,\vec{a},\alpha_i,\xi_i) = R^2 + C\sum_i \xi_i - \sum_i \alpha_i(R^2 + \xi_i - (\vec{x}_i \cdot \vec{x}_i - 2\vec{a}\cdot\vec{x}_i + \vec{a}\cdot\vec{a})) - \sum_i \gamma_i \xi_i \tag{10}$$

with Lagrange multipliers $\alpha_i \ge 0$ & $\gamma_i \ge 0$.

After replacing dot products by kernel, the dual formulation amounts to the maximization of

$$L = \sum_i \alpha_i K(\vec{x}_i,\vec{x}_i) - \sum_{i,j} \alpha_i \alpha_j K(\vec{x}_i,\vec{x}_j) \tag{11}$$

with constraints

$$0 \le \alpha_i \le C \tag{12}$$

$$\sum_i \alpha_i = 1 \tag{13}$$

Only for some set of points the equality in Equation (9) is satisfied. These points lie on the boundary of sphere and are called as support vectors for which the coefficients α_i are non-zero. These points completely describe the sphere. The radius of the sphere is calculated as the distance of support vector for which $\alpha_i < C$ from the center of the sphere. The outliers or abnormal points are the bound objects for which $\alpha_i = C$. Having completed the training process a test point $\vec{z}$ is

declared as an outlier, if the distance of the point to the center of the sphere is larger than the radius:

$$K(\underline{z},\underline{z}) - 2\sum_i \alpha_i K(\underline{z},\underline{x}_i) + \sum_{i,j} \alpha_i \alpha_j K(\underline{x}_i,\underline{x}_j) > R^2 \qquad (14)$$

Different kernel functions can be used to get different domain description boundaries. The most popular kernels are polynomial kernel and Gaussian RBF kernel.

Polynomial kernel $$K(\underline{x}_i,\underline{x}_j) = (1 + \underline{x}_i \cdot \underline{x}_j)^n \qquad (15)$$

Gaussian RBF kernel $$K(\underline{x}_i,\underline{x}_j) = \exp\left(\frac{-\left\|\underline{x}_i - \underline{x}_j\right\|^2}{2\sigma^2}\right) \qquad (16)$$

4.4. Case Studies

We have illustrated the method of online abnormality detection in a process plant using LLE-SVDD method in the following sub-sections with the two case studies viz. acetone-butanol fermentation and a benchmark semi-batch reactor problem.

4.4.1 Case study 1: Batch Acetone-Butanol Fermentation

Acetone-butanol fermentation is considered here as a case study. The mathematical model is taken from Vortruba et al. (1986). The model for the batch culture of *clostridium acetobutylicum* has been formulated using experimental data for anaerobic solvent production. The model takes into account biochemical as well as physiological aspects of growth and metabolite synthesis. The same example has been considered by Singhal (2002) for evaluating different pattern matching techniques. In this example we use the model

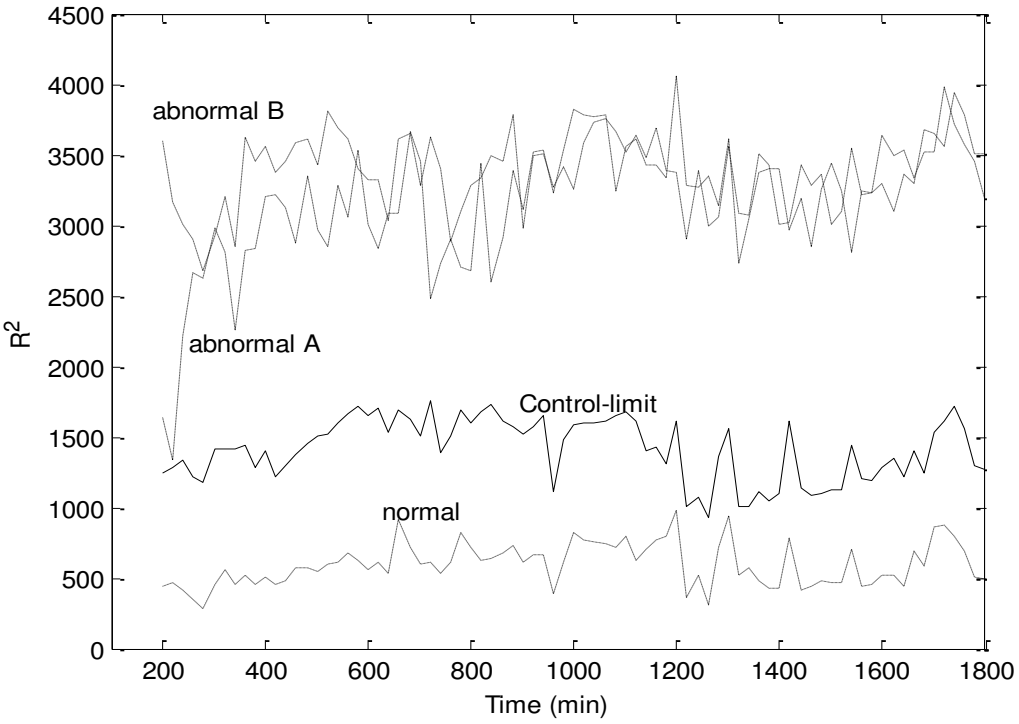


Figure 4.1: Abnormality detection in acetone-butanol fermentation using LLE-SVDD (showing abnormal batches A &B)

for online abnormality detection using SVDD along with nonlinear dimensionality reduction technique, LLE. The model consists of ten nonlinear differential equations. The parameters are the same as in Vortruba et al. (1986) and Singhal (2002). 55 normal batches were simulated by giving some variations in reactor cell concentration, substrate concentration and dimensionless cellular RNA concentration. This forms the historical database for the training of LLE-SVDD. In addition to this, two normal and six abnormal batches were simulated as test batches. Out of the six abnormal batches the first two correspond to abnormality due to slow substrate utilization, the next two correspond to abnormality due to dead inoculum and the remaining two correspond to abnormality due to increased cell sensitivity to butanol. The nine process variables used for monitoring are: reactor cell concentration, substrate concentration, butyric acid concentration, acetic acid concentration, butanol concentration, acetone concentration, ethanol concentration, CO_2 concentration and H_2 concentration. All the variables are measured after every 10 minutes. The total time required for a single batch is 30 hr.

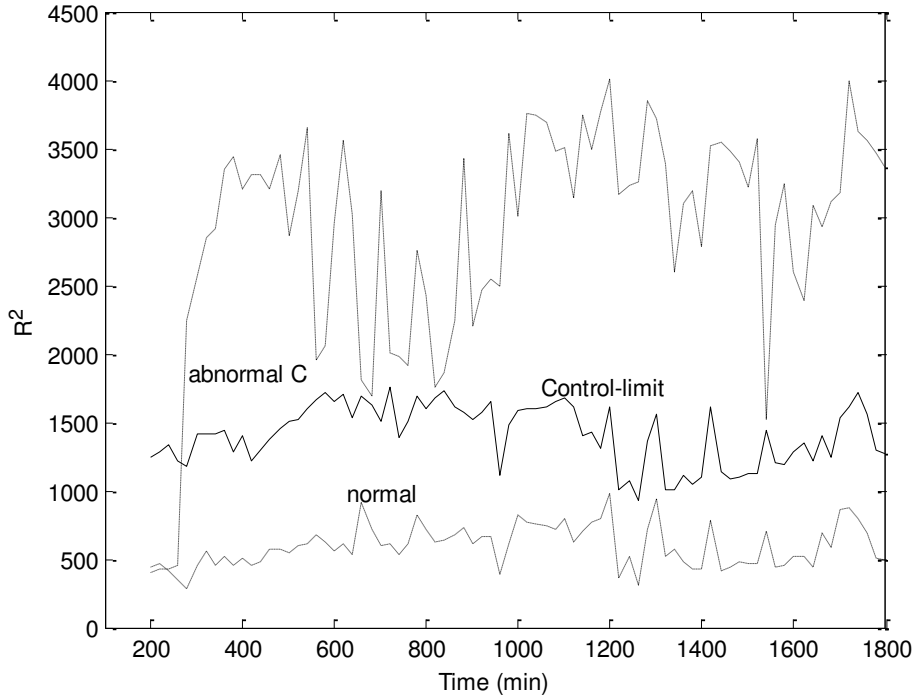


Figure 4.2: Abnormality detection in acetone-butanol fermentation using LLE-SVDD (showing abnormal batch C)

For this analysis, each nominal batch is unfolded, and can be represented as a data vector. For instance, at time τ, a batch can be written as,

$$[\underset{\sim}{Q}^T(\tau) \quad \underset{\sim}{Q}^T(\tau-\Delta t) \quad \underset{\sim}{Q}^T(\tau-2\Delta t) \quad .... \quad \underset{\sim}{Q}^T(\tau-(w-1)\Delta t)]$$

where

$$\underset{\sim}{Q}(\tau) = [q_{1,}\ q_{2,}\ \ .... \ q_J]^T$$

is the J-dimensional variables vector at time point τ. For this problem the number of monitoring variables J=9 and the length of moving window is *w*=20.

The training data for LLE-SVDD analysis consists of 55 normal batches with values of all the 9 selected variables at 20 consecutive sampling times. Thus data for LLE is a matrix of 55×180. We fix the dimensionality of reduced space (*d*) into which LLE is projecting to 15. The nearest neighbor parameter *N* (as obtained from the residual variance criterion) changes very slightly while moving from one window to another; thus for sake of computational simplicity the value of *N* was fixed at an average value of 12. The reduced matrix with dimension of

55×15 is then used to train SVDD for novelty detection. The testing data with reduced dimension is obtained using on-line LLE as described in section 2.

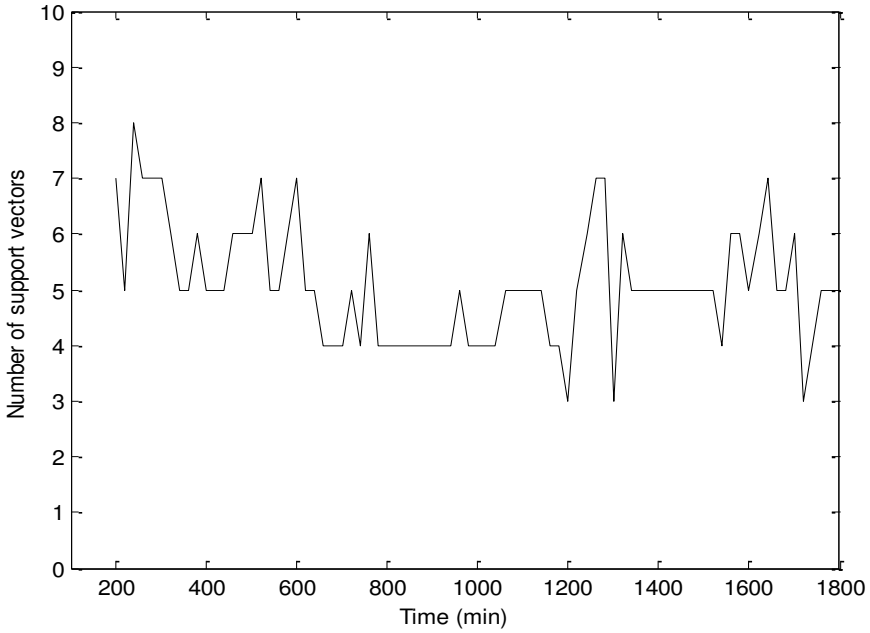


Figure 4.3: Number of support vectors obtained for Acetone-butanol fermentation problem

SVDD is carried out on this dataset by selecting an appropriate kernel, which is able to give the best description of the data domain. This classification is done by the following steps: (i) The quadratic optimization problem represented by Equation (11) is solved to get the support vectors and the corresponding Lagrange multipliers, (ii) These values are used to calculate the value of R^2 using Equation (14), (iii) The window is moved further by 10 minutes interval and the SVDD as described above is again carried out for the set of data vectors belonging to the new window. The procedure is repeated until the analysis covers the variables at the final time. Value of R^2 for support vectors act as the control limit for online testing of a new batch. The R^2 values are unique for each window and thus there exists a profile of R^2 for the nominal batches. As long as the R^2 value of the test batch lies below the SV line (solid line shown in Figure 4.1 & 4.2), the batch is normal. LLE-SVDD identifies all the six abnormal test batches. In Figures 4.1 and 4.2, one test batch belonging to each case i.e. slow substrate utilization (abnormal batch A), dead inoculum (abnormal batch B), increased cell sensitivity to butanol

(abnormal batch C) is shown. Test batches A & B are detected as abnormal batch from the beginning of the batch, whereas batch C is detected as abnormal 240 minutes after the start of the batch. The polynomial kernel of order 2 is used with parameter, C= 1. SVDD parameters were obtained heuristically. The number of support vectors obtained throughout the batch duration is shown in Figure 4.3 and on an average constitutes 9.24 % of total data. The number of outliers obtained for each window is zero. Computational requirement for SVDD algorithm is very low as it works on a fraction of data (support vectors) in training set. Dimensionality reduction with LLE requires more computations than that required by the conventional methods like PCA, but with an advantage of keeping the nonlinear features of the data intact.

An alternative way is to consider the data from the initial time point to current time point. Although this increases the computational load on SVDD, it is still manageable due to dimensionality reduction ability of LLE. The analysis was carried out by considering the following matrix for a batch at any given time instant τ :

$$[\overset{\cup}{Q}^T(\tau) \quad \overset{\cup}{Q}^T(\tau-\Delta t) \quad \overset{\cup}{Q}^T(\tau-2\Delta t) \quad .... \quad \overset{\cup}{Q}^T(0)\,]$$

The data fed to LLE consists of variables from initial time point to the current time point. The data was first reduced by LLE before being processed by the classification algorithm. SVDD with polynomial kernel of order 2 along with parameter C=1 successfully identifies the normal and abnormal test batches as shown in Figure 4.4. (To avoid complexity, only two of the abnormal batches are shown in the figure).

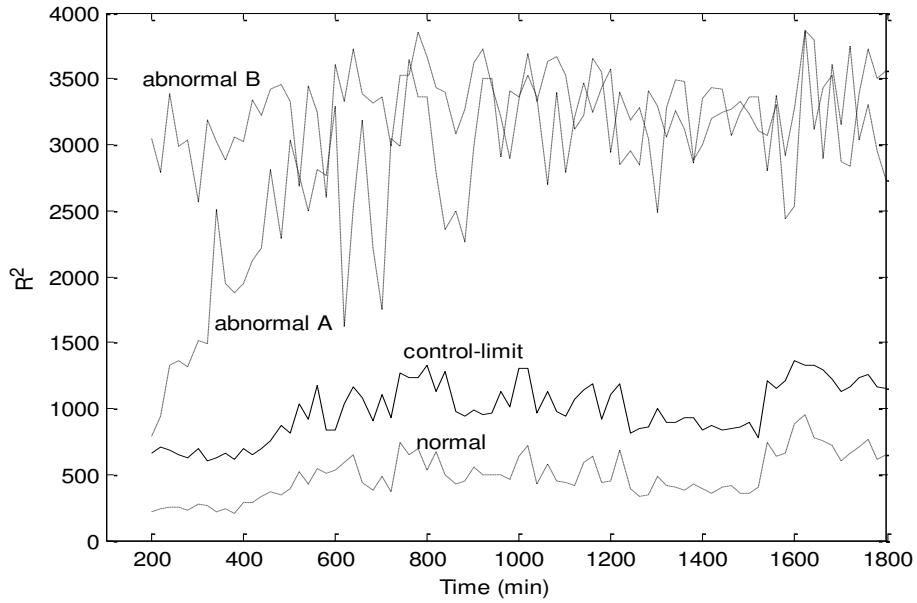


Figure 4.4: Abnormality detection in acetone-butanol fermentation using LLE- SVDD (Considering data from initial time point to the current one)

For testing the efficacy of the SVDD abnormality detection algorithm, we have compared the results with the recently proposed dynamic PCA (Chen and Liu, 2002). The criterion used for comparison of performance is time required by the method to detect the fault after it occurs. *Q*-chart for dynamic PCA with three principal components (capturing 97% variance) with 95% confidence limit with two time lag windows is shown in Figure 4.5. As shown in the figure the dynamic PCA identifies the normal batch. But it detects the abnormality of the batches (abnormal A, B & C) at much latter stages of the process, whereas LLE-SVDD identifies the abnormality of the batches from the beginning of the process (Figures 4.1 & 4.2).

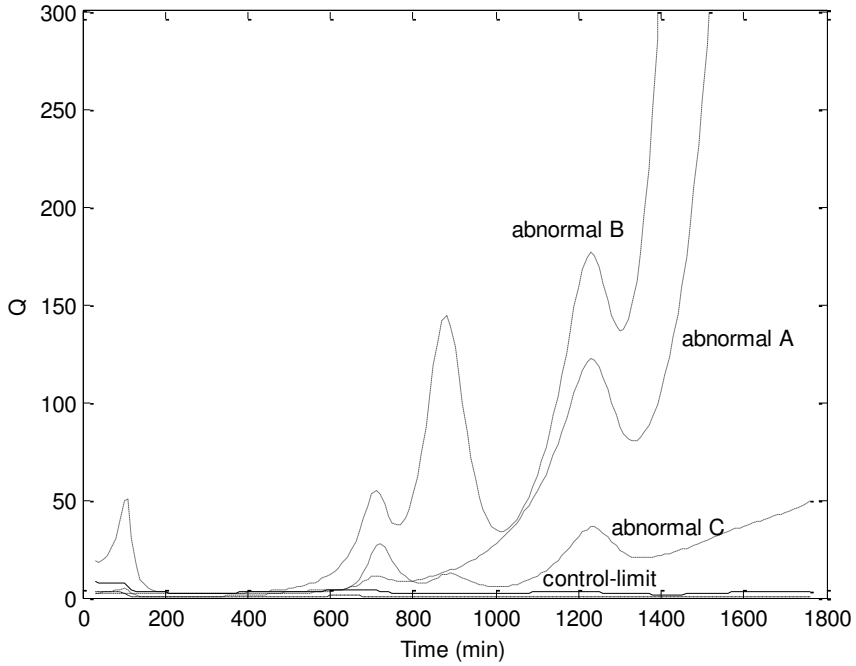


Figure 4.5: Q-chart for abnormality detection in acetone-butanol fermentation using dynamic PCA

4.4.2 Case Study 2: Semi-Batch Reactor for SBR production

This example is a simulated study of a semi batch reactor for the production of styrene-butadiene rubber (SBR) (Nomikos & MacGregor, 1994). This problem has been used as benchmark for evaluating the performance of various process monitoring methods (Nomikos & MacGregor, 1994; Chen & Liu, 2002). The reference data set contains 50 normal batches with some variations in the base conditions like impurities in the initial charge of organic phase and in the butadiene monomer feed to the reactor. The batch is divided into 200 time intervals and nine different variables were chosen for the purpose of monitoring. Apart from this reference set, three test batches were simulated: first is the normal batch (test batch A), second an abnormal batch with an initial organic impurity contamination in the butadiene feed, 30% above that of the base case (test batch B); and the third with the same problem, but this batch having contamination, 50 % above the normal level, started halfway through the operation (test batch C). The numerical data sets for all the variable measurements for the 50 nominal and three test batches (one normal and two abnormal) were obtained from Nomikos and MacGregor (1994). The three test batches are presented in the form of figures

in their paper. It can be clearly seen from these trajectories that it would be difficult to differentiate normal and abnormal batches by visual observations alone, requiring a rational abnormality detection methodology.

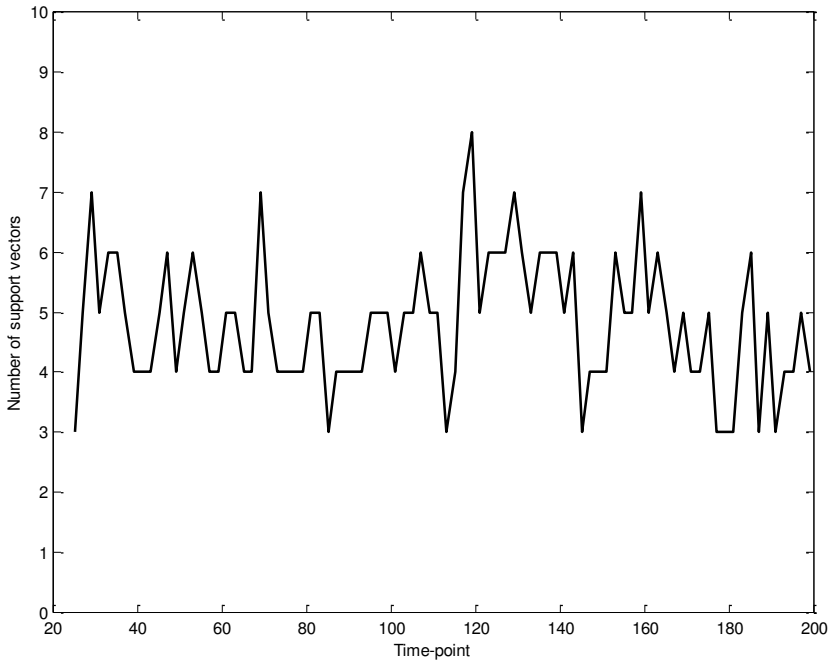


Figure 4.6: Number of support vectors obtained for Semi-batch reactor

LLE-SVDD analysis is done for SBR data using moving window of length 25. Thus training data for LLE-SVDD analysis consists of 50 normal batches with values of all the 9 selected variables at 25 consecutive sampling times. LLE reduces dimensionality of training data from 225 to 15 using the nearest neighbor parameter N equal to 12. The reduced matrix with dimension of 50×15 is then used to train SVDD. Thus LLE-SVDD analysis is done for each time-point and the window is moved after each time-point till the completion of the batch. Polynomial kernel of order 7 was used with parameter, C=0.5. The number of support vectors along the batch duration is shown in Figure 4.6 and on an average constitutes 9.63 % of total data. For this problem too, we obtained zero outlier for each window during the complete batch duration. The support vectors along with their Lagrange multipliers are used to calculate the value of R^2 for the test batch. The R^2 values for support vectors on boundary of the hypersphere is calculated and shown in Figure 4.7 as solid line, which acts as the control limit for the online test batch. It is clear from the figure, LLE-SVDD hybridization works well for all

the test batches i.e. it successfully identifies the normality and abnormality of the test batches for online monitoring. For instance, R^2 value of test batch A lies below the control-limit throughout the batch, hence it is a normal batch (Figure 4.7). The test batch B on the other hand crosses the control-limit at the 29th time point and is classified as abnormal from 30th time point to the end of batch. Similarly from figure 4.7, it is clear that the test batch C is normal up to 108th time point, but at the 109th time point it shows abnormality as it crosses the control limit and remains abnormal up to the end of the batch. The example clearly brings out the simplicity and usefulness of the method. This example has been studied by various methods in the literature and the performance of the proposed hybrid method for the SBR data is compared to that of benchmark MPCA method (Nomikos and MacGregor, 1994). Again the criterion used for comparison of performance is time required by the method to detect the fault after it occurs. MPCA method detect abnormality of test batch A within first 15 time points, while it detect the test batch C as abnormal before 110th time point (Nomikos and MacGregor, 1994). The results of both the methods are found to be comparable i.e. there is not significant difference between times taken by the hybrid method and the conventional MPCA method to detect the fault after its occurrence (Nomikos and MacGregor, 1994).

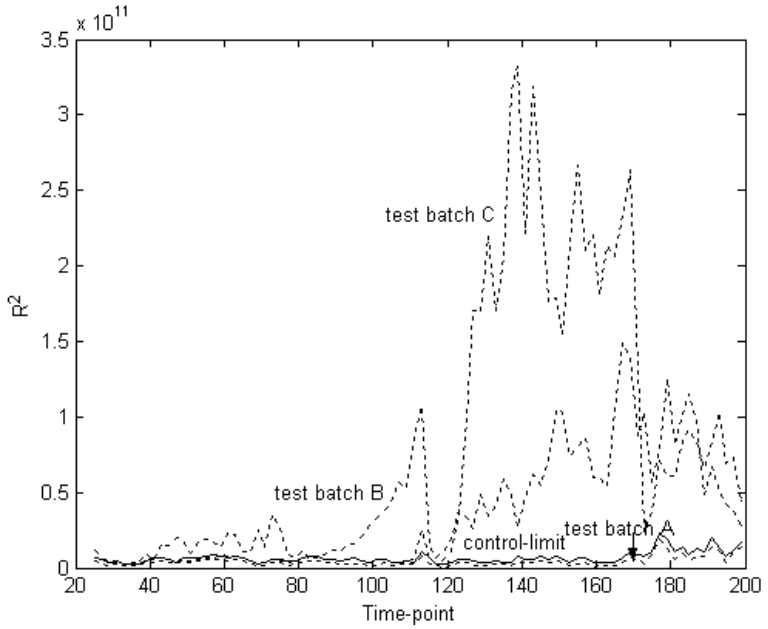


Figure 4.7: Abnormality detection in semi-batch reactor using LLE-SVDD

In working plants it may be very difficult to obtain large number of data pertaining to abnormal process conditions. As SVDD-LLE methodology requires only data belonging to normal conditions, the SVDD-LLE methodology is particularly advantageous as compared to many other existing techniques requiring large number of abnormal data. The trained SVDD algorithm can be completely characterized by a very small fraction (less than 10 %) of the total training data (i.e. support vectors) to define the distribution. This greatly reduces computational load during online testing. Another desirable feature of SVDD is that it requires solution of a quadratic optimization problem always leading to the unique global solution. On the other hand, some AI based methodologies solve a hard nonconvex optimization problem with the possibility of converging to one of the local minima. Additionally, the number of free parameters in SVDD does not depend explicitly on the input dimensionality, unlike other machine learning methods. The LLE part of the algorithm retains the relevant nonlinear features while reducing the input dimension rendering the hybrid methodology very attractive compared to the existing methods.

4. 5. Summary

The hybrid method using LLE and SVDD is illustrated with two case studies of acetone butanol fermentation and a benchmark SBR problem. The results show that LLE along with SVDD can be a very powerful tool for online process monitoring. As most of the industrial processes are nonlinear in nature, nonlinear dimensionality reduction using LLE can be very useful in reducing the features of the data, which in turn reduces the time for abnormality detection technique like SVDD.

References

Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., & Van Der Vorst, H., "Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide", Society for Industrial & Applied Mathematics, Philadelphia, (2000).

Bailer-Jones, C.A.L., Irwin, M., & Von Hippel, T., "Automated Classification of Stellar Spectra. II: Two-Dimensional Classification with Neural Networks & Principal Components Analysis", Monthly Notices of the Royal Astronomical Society, 298, 361-377, (1998).

Belkin, M., & Niyogi, P., "Laplacian Eigenmaps & Spectral Techniques for Embedding & Clustering" in "Advances in Neural Information Processing Systems", T.G. Dietterich, S. Becker & Z. Ghahramani Eds., MIT Press, Cambridge, Vol. 14, 585-591 (2002).

Bishop, C.M., Svensén, M., & Williams., C.K.I., "GTM: The Generative Topographic Mapping", Neural Computation, 10 (1), 215-234, (1998).

Boque, R., & Smilde, A. K., "Monitoring & Diagnosing Batch Processes with Multiway Covariates Regression Models", AIChE J., 45, 1503-1520, (1999).

Borg, I., & Groenen, P., "Modern Multidimensional Scaling: Theory & Applications", Springer-Verlag, Berlin, (1997).

Chen, B. H., Wang, X. Z., Yang, S. H., & Mcgreavy, C., "Application of wavelets & neural networks to diagnostic system development. I. Feature extraction", Comput. Chem. Engng. 23, 899-906, (1999).

Chen, J, & Liu, J., "Post Analysis on Different Operating Time Processes using Orthonormal Function Approximation & Multiway Principal Component Analysis", J. Process Control, 10, 411- 418, (2000).

Chen, J., & Liu, K.-C., "On-line Batch Process Monitoring using Dynamic PCA & Dynamic PLS Models", Chem. Eng. Sci., 57, 63-75, (2002).

Dahl, S. K., Piovoso, M. J., & Kosanovich, K. A., "Translating Third-order Data Analysis Methods to Chemical Batch Processes", Chemometr. lntell. Lab. Syst., 46, 161-180, (1999).

DeMers, D., & Cottrell, G.W., "Nonlinear Dimensionality Reduction", in "Advances in Neural Information Processing Systems", C.L. Giles, S.J. Hanson & J.D. Cowan Eds., Morgan Kaufmann, San Mateo, CA, Vol. 5, 580-587 (1993).

Devijver, P.A., & Kittler, J., "Pattern Recognition, A Statistical Approach", Prentice-Hall, London (1982).

Dong, D., & McAvoy, T. J., "Batch tracking via nonlinear principal component analysis", AIChE J., 42, 2199-2208, (1996b).

Dong, D., & McAvoy, T. J., "Nonlinear Principal Component Analysis - based on Principal Curves & Neural Networks", Comput. Chem. Engng., 20, 65-78, (1996).

Duda, R.O., Hart, P.E., & Stork, D.G., "Pattern Classification", John Wiley & Sons, New York, NY, 2nd edition, (2001).

Fan, J. Y., Nikolaou, M., & White, R. E., "An approach to fault diagnosis of chemical processes via neural networks", AIChE J., 39, 82-88, (1993).

Farell, A. E., & Roat, S. D., "Framework for enhancing fault diagnosis capabilities of artificial neural networks", Comput. Chem. Engng., 18, 613-635, (1994).

Hastie, T., & Stuetzle, W., "Principal curves", J. Amer. Statist. Assoc. , 84 (406), 502-516, (1989).

Horn, R. A., & Johnson, C. R., "Matrix Analysis", Cambridge University Press, Cambridge, 1990.

Hoskins, J. C., Kaliyur, K. M., & Himmelblau, D. M., "Fault diagnosis in complex chemical plants using artificial neural networks", AIChE J., 37, 137-141, (1991).

Jain, A.K., Duin, R.P.W., & Mao, J., "Statistical Pattern Recognition: a Review", IEEE Transactions on Pattern Analysis & Machine Intelligence, 22, ,4-37, (2000).

Kavuri, S. N., & Venkatasubramanian, V.,"Neural network decomposition strategies for large scale fault diagnosis", International Journal of Control, 59, 767-792, (1994).

Kavuri, S. N., & Venkatasubramanian, V.,"Using fuzzy clustering with ellipsoidal units in neural networks for robust fault classification", Comput. Chem. Engng., 17, 765-784, (1993).

Kohonen, T., "Self-organizing Maps", Springer Series in Information Sciences. Springer (1995).

Kouropteva, O., Okun, O., & Pietikainen, M., "Selection of the Optimal Parameter Value for the Local Linear Embedding Algorithm", Proceedings of the First International Conference on Fuzzy Systems & Knowledge Discovery, Singapore, 359-363, (2002).

Louwerse, D. J., & Smilde, A. K., "Multivariate Statistical Process Control of Batch Processes based on Three-Way Models", Chem. Eng. Sci., 55, 1225-1235, (2000).

Malthouse, E. C., "Limitations of Nonlinear PCA as Performed with Generic Neural Networks", IEEE Trans. Neural Networks, 9(1), 165-173, (1998).

Mardia, K.V., Kent, J.T., & Bibby, J.M., "Multivariate Analysis", Academic Press, London, (1979).

Martin, E. B., & Morris, A. J., "Non-Parametric Confidence Bounds for Process Performance Monitoring Charts", J. Process Control, 6, 349-358, (1996).

Martin, E. B., Morris, A. J., Papazoglou, M. C., & Kiparisassides, C.,"Batch Process Monitoring for Consistent Production", Comput. Chem. Engng., 20,S1, 599-604, (1996).

Mao, J., & Jain, A. K., "Artificial Neural Networks for Feature Extraction & Multivariate Data Projection", IEEE Trans. Neural Networks, 6, 296-317, (1995).

McClurkin, J.W., Optican, L.M., Richmond, B.J., & Gawne, T.J., "Concurrent Processing & Complexity of Temporally Encoded Neuronal Messages in Visual Perception", Science, 253, 675-677, (1991).

Murase, H., & Nayar, S. K., "Visual Learning & Recognition of 3-D Objects from Appearance", Int. J. Comp. Vision, 14, 5-24, (1995).

Nomikos, P., & MacGregor, J. F., "Multi-Way Partial Least Square in Monitoring Batch Processes", Chemometr. lntell. Lab. Syst., 30, 97-108, (1995).

Nomikos, P., & MacGregor, J. F., "Monitoring Batch Processes using Multiway Principal Component Analysis", AIChE J., 40, 1361-1375, (1994).

Ramaker, H.J., Van Sprang E.N.M., Gurden S.P., Westerhuis J.A., & Smilde A.K., "Improved Monitoring of Batch Processes by Incorporating External Information", J. Process Control, 12, 569-576, (2002).

Rannar, S., MacGregor, J. F., & Wold, S., "Adaptive Batch Monitoring using Hierarchical PCA", Chemometr. lntell. Lab. Syst., 41, 73-81, (1998).

Rengaswamy, R., & Venkatasubramanian, V.,"A fast training neural network & its updation for incipient fault detection & diagnosis", Comput. Chem. Engng., 24, 431-437, (2000).

Roweis, S.T., & Saul, L.K., "Nonlinear Dimensionality Reduction by Locally Linear Embedding", Science Vol. 290, 2323-2326 (2000).

Ruiz, D., Cantón J., Nougués, J. M., Espuña A., & Puigjaner, L., "On-line fault diagnosis system support for reactive scheduling in multipurpose batch chemical plants", Comput. Chem. Engng., 25, 829-837, (2001).

Ruiz, D., Nougués, J., Calderón, Z., Espuña A., & Puigjaner L., "Neural Network Based Framework for Fault Diagnosis in Batch Chemical Plants", Comput. Chem. Engng., 24, 777-784, (2000).

Sammon, J. W., "A Nonlinear Mapping for Data Structure Analysis", IEEE Transactions on Computer, 18 (5), 401-409, (1969).

Scenna N. J.,"Some aspects of fault diagnosis in batch processes", Reliability Engineering & System Safety, 70, 95-110, (2000).

Singhal, A., "Pattern Matching in Multivariate Time Series Data", PhD Thesis, Univ. of California, Santa Barbara (2002).

Tax, D. M. J., & Duin, R. P.W., "Support Vector Domain Distribution", Pattern Recognition Letters, 20 (11-13), 1191-1199, (1999).

Tenenbaum, J. B., de Silva, V., & Langford, J. C., "A Global Geometric Framework for Nonlinear Dimensionality Reduction", Science, 290, 2319-2323, (2000).

Tipping, M.E., & Bishop, C.M., "Mixtures of Probabilistic Principal Component Analyzers", Neural Computation, 11, 443-482, (1999).

Ungar, L. H., Powell, B. A., & Kamens, S. N.,"Adaptive networks for fault diagnosis & process control", Comput. Chem. Engng., 14, 561-572, (1990).

Van Sprang, E. N. M., Ramaker, H.-J., Westerhuis, J. A., Gurden, S. P., & Smilde, A. K., "Critical Evaluation of Approaches for on-line Batch Process Monitoring", Chem. Eng. Sci. 57, 3979-3991, (2002).

Venkatasubramanian, V., & Chan, K., "A neural network methodology for process fault diagnosis", AIChE J., 35, 1993-2002, (1989).

Venkatasubramanian, V., Vaidyanathan, R., & Yamamoto, Y.,"Process fault detection & diagnosis using neural networks I: steady state processes", Comput. Chem. Engng. 14, 699-712, (1990).

Vlachos, M., Domeniconi, C., Gunopulos, D., Kollios, G., & Koudas, N., "Non-Linear Dimensionality Reduction Techniques for Classification & Visualization", in "Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining", (2002).

Vortruba, J., Volesky, B., & Yerushalmi, L., "Mathematical Model of a Batch Acetone-Butanol Fermentation", Biotechnol. Bioeng., 28, 247-255, (1986).

Wang, X. Z., Chen, B. H., Yang, S. H., & Mcgreavy, C., "Application of wavelets & neural networks to diagnostic system development. 2. An integrated framework & its application", Comput. Chem. Engng., 23, 945-954, (1999).

Watanabe, K., Matsura, I., Abe, M., Kubota, M., & Himmelblau, D. M. , "Incipient fault diagnosis of chemical processes via artificial neural networks", AIChE J., 35, 1803-1812, (1989).

Westerhuis, J. A., Kourti, T., & MacGregor, J. F., "Comparing Alternative Approaches for Multivariate Statistical Analysis of Batch Process Data", J. Chemometrics, 13, 397- 413, (1999).

Wise, B. M., Gallagher, N. B., Butler, S. W., White Jr., D. D., & Barna, G. G., "A Comparison of Principal Component Analysis, Multiway Principal Component Analysis, Trilinear Decomposition & Parallel Factor Analysis for Fault Detection in a Semiconductor Etch Process", J. Chemometrics, 13, 379-396, (1999).

Wold, S., Geladi, P., Esbensen, K., & Ohman, J., "Multi-way Principal Components & PLS Analysis", J. Chemometrics, 1, 41-56, (1987).

Zhao, J., Chen, B., & Shen, J., "A hybrid ANN-ES system for dynamic fault diagnosis of hydrocracking process", Comput. Chem. Engng., 21, S929-S933, (1997).

Notation

$\vec{a}$	center of hypersphere
C	regularization parameter in SVM
C_{jk}	local covariance matrix
D	original dimension
d	reduced dimension
E_X, E_Y	matrix of Euclidean distances in X, Y
F	optimization function in single class SVM
I	identity matrix
J	number of monitoring variables
K	kernel function
L	Lagrangian function
M	sparse matrix
N	number of nearest neighbors
n	order of polynomial kernel
P	number of data points
$\vec{Q}$	vector of monitoring variables
q	monitoring variable
R	radius of the hypersphere
t	time
w	length of moving window
W	weights for reconstruction
$\vec{X}$	original dataset
$\vec{x}$	datapoint
$\vec{Y}$	reduced dataset
$\vec{z}$	test point

Greek letters

α, γ	Lagarange multipliers
δ	parameter in Equation (4)
ε	reconstruction error
σ	width of Gaussian RBF kernel
Φ	embedding cost function
ω	reconstruction weight
ψ	nearest neighbors
ρ	standard linear correlation coefficient
ξ	slack variables
τ	time point

Superscripts

T	transpose of matrix
-1	inverse of matrix

Subscripts

opt	optimal

Chapter 5

A NOVEL SINGULARITY BASED METHOD FOR TIME SERIES CHARACTERIZATION: AN APPLICATION TO FLOW REGIME IDENTIFICATION IN STIRRED REACTOR

5. 1 Introduction

Gas-liquid flows in stirred reactor depend on the operating conditions and the impeller design and can be classified into different regimes. These flow regimes in turn manifest different fluid dynamic characteristics (see Figure 5.1) and demonstrate complex interaction of transport and mixing processes. Significant research efforts have been undertaken in the recent past for developing regime maps and the corresponding design correlations (see the excellent review of Nienow, 1998 and the references cited therein). However, the universal applicability of the regime maps and the correlations to design, scale-up and for setting up of operating protocols for industrial systems is not yet well established. Therefore, the need to develop a new robust experimental methodology based on a simple and non-intrusive measurement technique continues to exist.

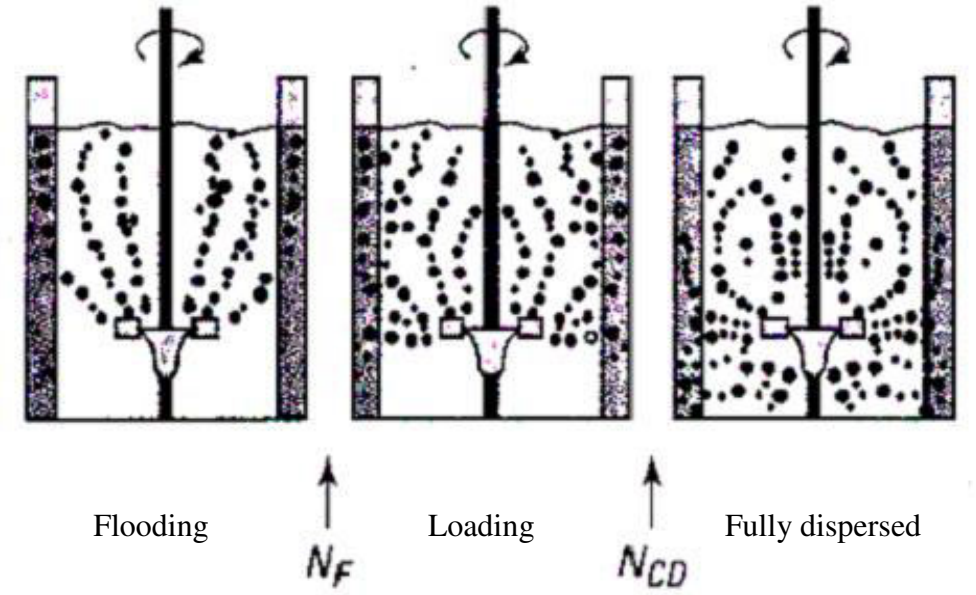


Figure 5.1: Different flow regimes in stirred reactor equipped with Rushton turbine (Nienow et al., 1985)

Warmoeskerken and Smith (1985); Sutter et al., (1987) and Bombac et al. (1997) used intrusive techniques such as micro-impeller, hydrophones and

resistivity probes respectively to extract the information of cavity structure present behind the impeller blades and develop flow regime map. All the techniques so far suggested are reliable for laboratory scale reactor and are difficult to use with industrial reactors. In order to overcome some of these limitations, Pagalianti et al. (2000) made an attempt to characterize the gas-liquid flows in stirred vessel by means of statistical methods such as nonlinear time series analysis from the output signal of the non-intrusive probes. Pagalianti et al. (2000) identified the flooding/loading transition by using time series analysis of the measured impedance. The proposed technique was limited only to identify flooding/loading transition, which is clearer and sharper than the other regime transitions.

Various authors have also identified regimes of operation by analyzing the extracted nonlinear dynamical, fractal and statistical features from pressure fluctuation measurements. (Bai et al. 1996; Bai et al. 1997 ; Johnsson et al. 2000; Letzel et al. 1997; Lin et al. 2001; Wu et al. 2001; Xie et al. 2003; Xie et al. 2004). These studies were mainly restricted to fluidized bed and bubble column. Recently, Khopkar et al. (2005) characterized the gas-liquid flows in stirred reactor employing wall pressure and torque fluctuations and used non-linear time series analysis to set up robust criteria for the identification of the prevailing flow regimes. They differentiated the flow regimes based on the cavity structure present behind the impeller blades and also estimated the key time scale of the fluid dynamics. In the present study, we have proposed a novel methodology for characterization of time series based on the combination of wavelet based local singularity distribution analysis and support vector machines (SVM), a newly developed pattern classification method. The method developed is subsequently applied for characterization of flow regimes in stirred tank vessel with Rushton turbine. While wavelet techniques have been extensively used in several engineering applications including chemical engineering (Chen et al. 2004; Ellis et al. 2003; Kulkarni et al. 2001; Park, et al. 2001; Roy et al. 1999; Zhao & Yang 2003), the use of local singularity distribution analysis is relatively new and finds recent applications in biomedical engineering; stock market etc. for analyzing and charactering time series. (Scafetta et al. 2003; Struzik & Siebes, 2002; West et al. 2004). Support vector machines (SVM), a novel tool for classification, is firmly based on rigorous statistical learning theory (Burges, 1998; Vapnik, 1995, 1998).

SVM also has found wide spread use including applications in process engineering (Agarwal et al. 2003; Chiang et al. 2004; Kulkarni et al. 2004).

In the present work, wall pressure fluctuations were measured in a gas-liquid stirred reactor equipped with Rushton turbine. The time series of the pressure fluctuations were first subjected to singularity analysis based on wavelet transform modulus maxima (WTMM) method. The relevant features extracted from this analysis were employed as input data by SVM for identifying the operating regimes. The remaining part of the chapter is organized as follows: section 5.2 provides a detailed description of the proposed method for time series characterization, Section 5.3 provides a brief description of the experimental set up of stirred vessel and in section 5.4 we discuss the results of flow regime identification in a stirred vessel. The salient conclusions are highlighted in section 5.5.

5.2. Time Series Characterization using Singularity Distribution and SVM

The methodology proposed for characterization of time series is a novel combination of singularity analysis and SVM classification. The time series under consideration is first subjected to wavelet transform modulus maxima (WTMM) method and the most informative features from the singularity distribution are extracted. These features are then used as input to SVM for intelligent discrimination of the time series. SVM being a supervised learning method, data is divided into training and test sets. The model is built using the features extracted from the training set of time series. The trained model can then be readily employed for online characterization and identification of unseen test data. The algorithmic steps involved in the proposed methodology are shown Figure 5.2, while the details of method are explained in the subsequent sections.

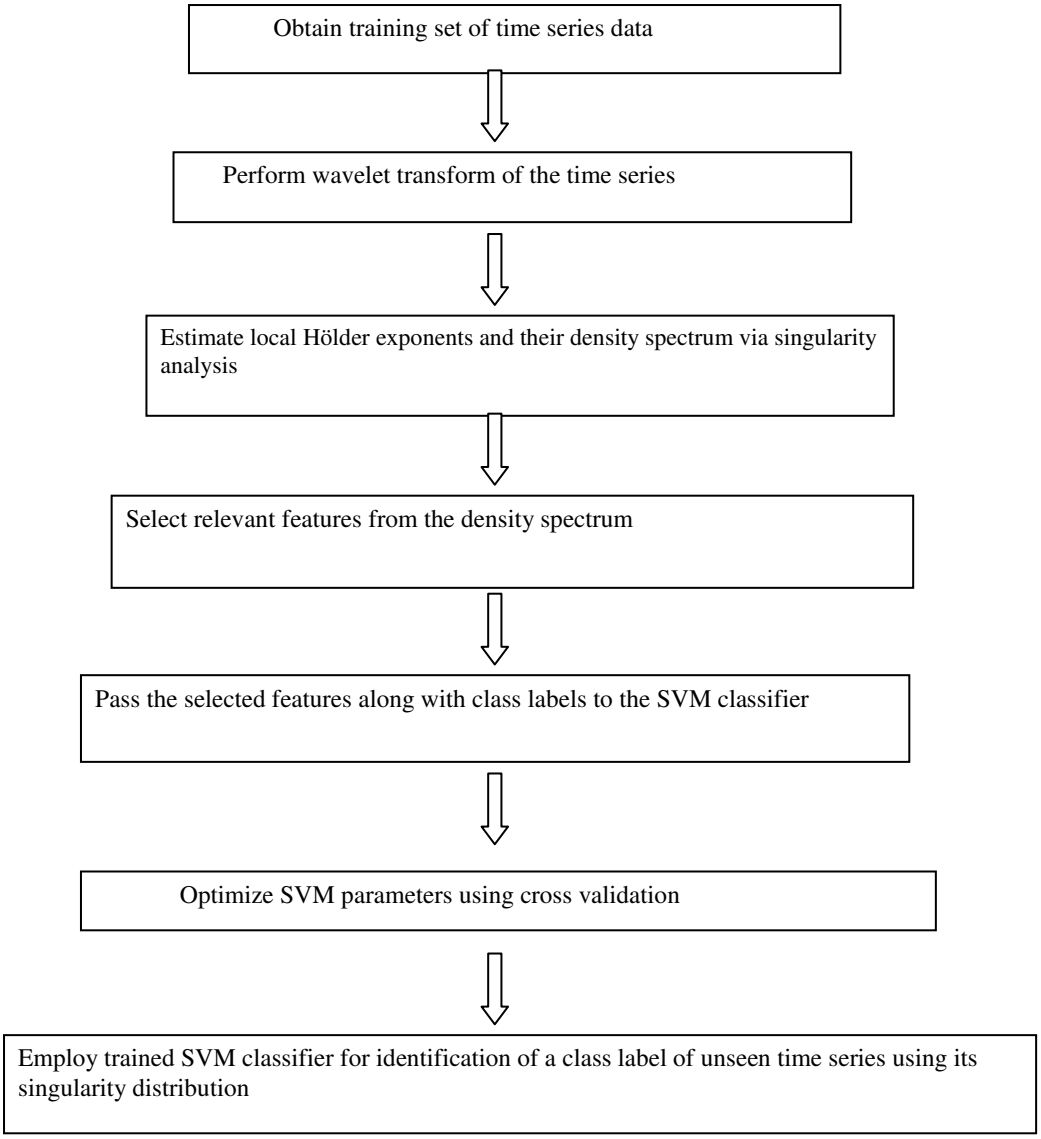


Figure 5.2: Proposed methodology for time series characterization

5.2.1 Characterization of Singularities

Many experimental or empirical time series have fractal features i.e., for some instances, the series $f(x)$ displays singular behavior. By this, we mean that at those instances, the signal can not be described solely by Taylor series and has components with non-integer powers of time which appear as step-like or cusp-like features, the so-called singularities, in the signal. (Figure 5.3) Such signals need to be represented as:

$$f(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + a_3(x - x_0)^3 + \ldots\ldots\ldots + a_h(x - x_0)^h \qquad (1)$$

If a polynomial, $P_n(x)$ of the degree n corresponding to the Taylor series expansion exists such that:

$$\left|f(x) - P_n(x - x_0)\right| \leq C_1 \left|x - x_0\right|^h \qquad (2)$$

then h is said be the local Hölder exponent of the function and it characterizes the scaling of the function locally and h lies within the bounds $n < h < n+1$.

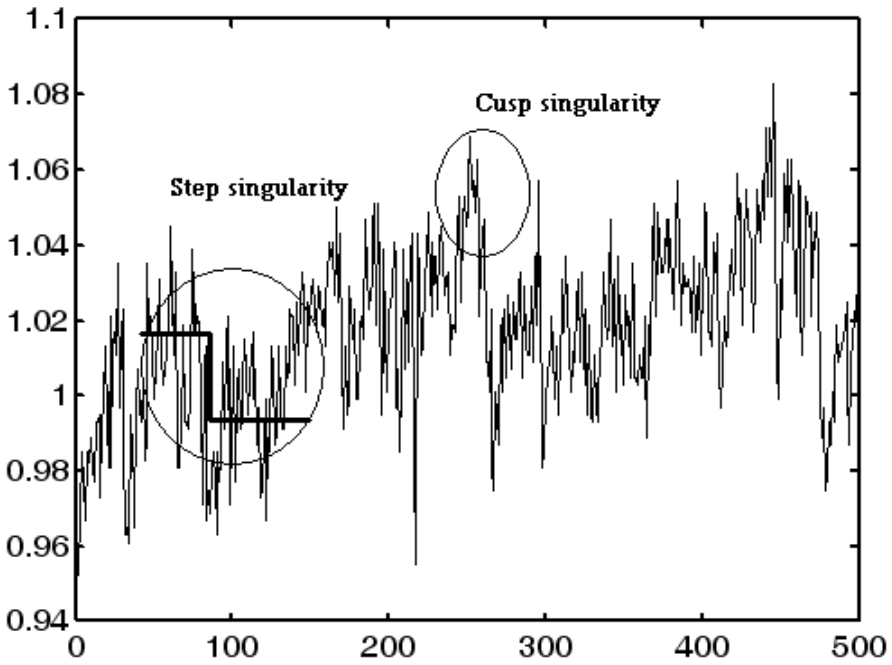


Figure 5.3: A real life human gait time series (available on public archive at www.physionet.org)

It must now be emphasized that if $h(x_0)$ is equal to a positive integer n, the function f is n times continuously differentiable in x_0. If on the other hand $n < h(x_0) < n+1$ the function f is continuous and singular in x_0. In this case f is n times differentiable, but its n^{th} derivative is singular in x_0 and the exponent h characterizes this singularity and hence the regularity of the function f in x_0. The higher the h, the more regular is the local behaviour of the function f.

The distinct behavior of dynamical systems can be rigorously characterized by singularity analysis. The distribution of local singularities of a

fractal time series can serve as its unique signature. Hence, the features of the singularity distribution of different time series can be employed for identifying the class of a time series. Recently, Muzy et al. (1991, 1993, and 1994) have introduced a novel wavelet transform based approach for direct determination of the singularity spectrum. Wavelet transform fundamentally differs from global transforms like the Fourier transformation in a way that in addition to locality, it possesses very desirable ability of filtering the polynomial behavior to some predefined degree. Therefore, rigorous characterization of time series is possible, in particular in the presence of non-stationarities. The ability of wavelet transform to reveal the hierarchy of singular features is particularly advantageous to tackle the problem at hand. Mathematically, the wavelet transform (WT) is a convolution product of the time series with a characteristic wavelet. The wavelet transform can be formally written as:

$$T_{\psi}[f](x_0, s) = \frac{1}{s}\int_{-\infty}^{\infty} f(x)\Psi\left(\frac{(x - x_0)}{s}\right)dx \qquad (3)$$

The scale parameter "*s*" modulates the width of the wavelet kernel to the desired level of resolution and the parameter x_0 determines the location of the governing wavelet.

By virtue of the scale parameter, wavelet transform can reveal even the weaker singularities within the time series, which facilitates the complete spectrum available for rigorous analysis. The wavelet function $\psi(x)$ is chosen to be well localized both in space and frequency. Usually, ψ is only required to be of zero mean but for the purpose of singularity tracking ψ is further required to be orthogonal to some low-order polynomials. (Arneodo et al. 1995 ; Muzy et al. 1991, 1993)

$$\int_{-\infty}^{\infty} x^m \Psi(x)dx = 0 \qquad \forall m, \quad 0 \le m < n_{\Psi} \qquad (4)$$

Wavelets given by the successive derivatives of the Gaussian function satisfy the above condition:

$$\Psi^M(x) = \frac{d^M}{dx^M} \exp^{-x^2/2} \quad , \quad \text{where } n_\Psi = M \; . \tag{5}$$

In our work, we have chosen the 2nd derivative of Gaussian function, i.e. Mexican hat wavelet as the analyzing wavelet function.

$$\Psi(x) = (1 - x^2)\exp(-x^2/2) \tag{6}$$

A wavelet that has the number of vanishing moments greater than or equal to the degree of polynomial *f*(*x*), will filter out the polynomial trends and focus only on the singularities in the time series. It can be proven that a local singular behavior of $f(x)$ around $x = x_0$ can be characterized by $h(x_0)$, (Arneodo et al. 1995 ; Muzy et al. 1991, 1993)

$$T_\psi[f](x_0, s) \sim s^{h(x_0)} \qquad , s \to 0^+ \tag{7}$$

The distribution of singularities and the singularity spectrum can be obtained from a partition function based multifractal formalism. Continuous wavelet transform (CWT) in its original form is an extremely redundant representation; however, Mallat and Hwang (1992) have shown that a representation consisting of only (the modulus of) the maxima lines of the CWT, the wavelet transform modulus maxima (WTMM) can detect all the singularities of a large class of signals. Thus the hierarchical distribution of singularities in the time series can be computed by employing the space-scale partitioning provided by the maxima representation. Partition function, $Z(s,q)$ can be calculated as the sum of the q^{th} powers of the local maxima of $\left|T_\psi[f](x_0, s)\right|$ at the scale s. The partition function $Z(s,q)$ reflects the large fluctuations and strong singularities in time series for positive *q* and emphasizes small fluctuations and weak singularities for negative *q*.

At small values of *s* partition function will follow the power law behavior (Muzy et al. 1991, 1993, 1994, Arneodo et al. 1995),

$$Z(s,q) \sim s^{\tau(q)} \tag{8}$$

where, scaling exponents, $\tau(q)$, can be numerically estimated from a plot of $\log(Z(s,q))$ against $\log(s)$ for any real number *q*. The usefulness of the above approach is that local maxima of the wavelet coefficients alone carry all the

information content. After computing the scaling exponents the singularity strength can be obtained by using the formula, $h = d\tau / dq$. The singularity spectrum $D(h)$ can be estimated from the Legendre transform (Arneodo et al. 1995; Muzy et al.1991, 1993, 1994).

$$D(h) = qh(q) - \tau(q) \qquad (9)$$

5.2.2 Estimation of Local Hölder exponents

The Wavelet transform modulus maxima (WTMM) based formalism developed by Muzy et al. (1991, 1993, and 1994) as described above provides global estimates of scaling properties of the time series. Recently it has been found that though the global estimates of scaling is often a required property, the estimation of the singularity spectrum poses certain problems. It is now well known that the $D(h)$ spectrum can be corrupted by divergences of negative moments of the partition function (Struzik, 1998). Various methods have been proposed in the literature to overcome the difficulties (Mallat, 1999; Struzik, 1998) Recently, several problems have been solved taking advantage of the fact that the local information about scaling provides more relevant information than the global spectrum.(Struzik, 2000). In a traditional form, the estimation of local singularity strengths and their spectra may not be possible due to instability and may lead to gross numerical errors. This is due to the fact that in real life data the singularities are not isolated but densely packed. This causes the logarithmic rate of increase or decrease of the corresponding wavelet transform maximum line to fluctuate. Very recently, Struzik (2000) has provided a stable method for evaluating the local Hölder exponents. In his methodology for estimating the local exponents, he has modeled the singularities as if they were created through a multiplicative cascading process. This method has been successfully applied to localize outliers (Struzik & Siebes, 2002), for classification of human gait (Scafetta et al., 2003) and to study the influence of progressive central hypovolemia on cardiac interbeat intervals (West et al., 2004). We describe here the method in brief and more details can be found in Scafetta et al.(2003), Struzik (2000).

The mean Hölder exponent $\bar{h}$ can be estimated as a linear fit of the following equation,

$$\log[M(s)] = \bar{h}\log(s) + c \tag{10}$$

where function $M(s)$ is obtained from the partition functions,

$$M(s) = \sqrt{\frac{Z(s,2)}{Z(s,0)}}\,. \tag{11}$$

where partition function $Z(s,2)$ can be calculated as the sum of squares of the maxima of $\left|T_{\psi}[f](x_0,s)\right|$ at the scale s and $Z(s,0)$ is the number of maxima at scale s.

It can be shown that by employing the multiplicative cascade model the estimate of local Hölder exponent, $\hat{h}(x_0,s)$ at the singularity x_0 and scale s can be determined as (Scafetta et al.2003; Struzik, 2000)

$$\hat{h}(x_0,s) = \frac{\log\left(\left|T_{\psi}[f](x_0,s)\right|\right) - (\bar{h}\log(s) + c)}{\log(s) - \log(s_N)} \tag{12}$$

where s_N is the maximum available scale that coincides with the sample length and $T_{\psi}[f](x_0,s)$ is the maxima at location x_0 and at scale s.

Thus the methodology as explained above can be employed to obtain the profiles of local Hölder exponents for any given time series. Our contention is that the density spectrum of these profiles in conjunction with SVM classification can be used to characterize the time series.

5.2.3 Support Vector Machines

Support vector machines (SVM), a machine learning algorithm based rigorously on statistical learning theory, was originally developed by Vapnik (1995) for solving pattern recognition problems. The simplicity of implementation, excellent

generalization ability and remarkable performance on difficult tasks have made SVM one of the most popular tools in various disciplines including process engineering (Agarwal et al. 2003; Chiang et al. 2004; Kulkarni et al. 2004). For binary classification problems, given a set of nonlinearly separable input vectors belonging to two distinct classes, SVM finds an optimal linear separating hyperplane in a high dimensional feature space. SVM handles the computational intractability issues arising out of high dimensionality of the feature space by defining an equivalent kernel function in the input space itself. We provide a detailed account of binary SVM classification methodology below*:*

5.2.3.1 SVM Classification

Starting with the binary classification problem with *N* instances (An instance corresponds to the vector of features extracted from a given time series)

$$(y_1, \underset{\rho}{x}_1), \ldots, (y_N, \underset{\rho}{x}_N), \quad , \quad \underset{\rho}{x} \in \Re^n \quad y \in \{-1,+1\} \tag{13}$$

where $\underset{\rho}{x}_i$ is a vector of input features of the i^{th} instance and y_i corresponds to the target class to which the i^{th} instance belongs to.

For locating the linear separating hyperplane, $(\underset{\rho}{w} \cdot \underset{\rho}{x}) + b$, SVM maximizes the distance (margin) between the closest instances belonging to the two classes. It can be proven that such a maximal margin hyperplane can be obtained by minimizing the norm of the weight vectors. Further, the classification problem can be formulated as the following optimization problem (Vapnik, 1995, 1998; Burges, 1998):

Minimize the function

$$g(\underset{\rho}{w}) = \frac{1}{2} \|\underset{\rho}{w}\|^2 \tag{14}$$

subject to the constraints:

$$y_i (\underset{\rho}{w} \cdot \underset{\rho}{x}_i + b) \geq 1 \tag{15}$$

In general, it may not possible to construct a hyperplane without a certain amount of classification error. It would, however, be possible to find an optimal hyperplane that minimizes the probability of occurrence of classification errors, averaged over the training set. This is done by introducing N nonnegative slack variables such that

$$y_i(\vec{w}\cdot\vec{x}_i+b) \geq 1-\zeta_i \qquad i=1,2,.....,N \tag{16}$$

where $\zeta_i \geq 0$. The generalized optimal separating hyperplane is now determined by finding the vector $\vec{w}$, that minimizes the function,

$$g(\vec{w},\zeta) = (1/2)\|\vec{w}\|^2 + C\sum_{i=1}^{N}\zeta_i \tag{17}$$

(where, C, is a given value) subject to the constraints in Eq. (16).

It can be shown that the above equations can be formulated in terms of the following quadratic optimization problem (Burges, 1998)

$$\max_{\alpha} \quad -(1/2)\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\,\alpha_j\,y_i\,y_j\,(\vec{x}_i\cdot\vec{x}_j)+\sum_{i=1}^{N}\alpha_i \tag{18}$$

with the constraints,

$$0\leq\alpha_i\leq C \qquad i=1,.....,N \tag{19}$$

$$\sum_{i=1}^{N}\alpha_i\,y_i = 0 \tag{20}$$

Real life problems are too complex to be separated by a simple linear hyperplane. SVM handles such non-linearly separable data by mapping the data into a richer higher dimensional feature space and by subsequently employing a linear

classifier. The mapping of the input data $\vec{x}$ in the feature space $\vec{x} \rightarrow \Phi(\vec{x})$ leads to the optimization problem:

$$\max_{\alpha} \quad -(1/2)\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i \alpha_j y_i y_j (\Phi(\vec{x}_i)\cdot\Phi(\vec{x}_j)) + \sum_{i=1}^{N}\alpha_i \qquad (21)$$

Working in higher dimensional feature space induces an intractable computational problem of having to deal with very large vectors. This problem can be tackled by introduction of so-called kernel trick. The idea is to replace the dot product in feature space by appropriate kernel functions in the original input space:

$$K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i)\cdot\Phi(\vec{x}_j) \qquad (22)$$

A kernel function can be selected by using the Mercer's theorem (Vapnik, 1995, 1998). In our studies we have used the most popular kernel based on the Gaussian function which is defined as $K(\vec{x}, \vec{y}) = \exp(-\|\vec{x} - \vec{y}\|^2 / (2\sigma^2))$. With the introduction of the kernel function in place of the dot product (Eq.(22)) the optimization problem (Eq. (21)) can now be written in terms of features in the low dimensional input space itself:

Maximize

$$\sum_{i=1}^{N}\alpha_i - (1/2)\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j) \qquad (23)$$

Subject to the constraints

$$0 \le \alpha_i \le C \qquad i = 1, ..., N \qquad (24)$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \qquad (25)$$

It can be seen that the set of Eqs. (23-25) represents a convex quadratic optimization problem. It is this QP formulation having a unique global minimum

has attracted several applications of SVM in diverse fields. This is due to the fact that the QP problem can be solved by standard methodologies.

It can be shown that the discriminating hyperplane can be represented by:

$$\begin{aligned} f(\underline{x}) &= \sum_{i=1}^{N} \alpha_i y_i \Phi(\underline{x}_i) \cdot \Phi(\underline{x}) + b. \\ &= \sum_{i=1}^{N} \alpha_i y_i K(\underline{x}_i, \underline{x}) + b \end{aligned} \tag{26}$$

where b is the bias term and can be found as,

$$\begin{aligned} b &= -\frac{1}{2} \underline{w} \cdot [\Phi(\underline{x}_r) + \Phi(\underline{x}_s)] \\ &= -\frac{1}{2} \sum_{i=1}^{N} \alpha_i y_i [K(\underline{x}_i, \underline{x}_r) + K(\underline{x}_i, \underline{x}_s)] \end{aligned} \tag{27}$$

where $\underline{x}_r$, $\underline{x}_s$ are any support vectors (which have been discussed later in this section) from each class.

The class affiliation of any instance can be identified by the sign of the above function with positive values and negative values indicating class 1 and 2 respectively. The parameter C controls the tradeoff between complexity of the SVM and the number of non-separable points. It can also be shown that only those instances which have non- zero α values are the support vectors. Thus the trained classifier can therefore be represented by a few support vectors enabling online computations very fast. The overall SVM classification algorithm can be compactly written in terms of the following steps:

a) Maximize the margin of the linear hyperplane for simultaneous optimization of training and test accuracy.

b) Transform the input data into a higher dimensional feature space to enable linear classification

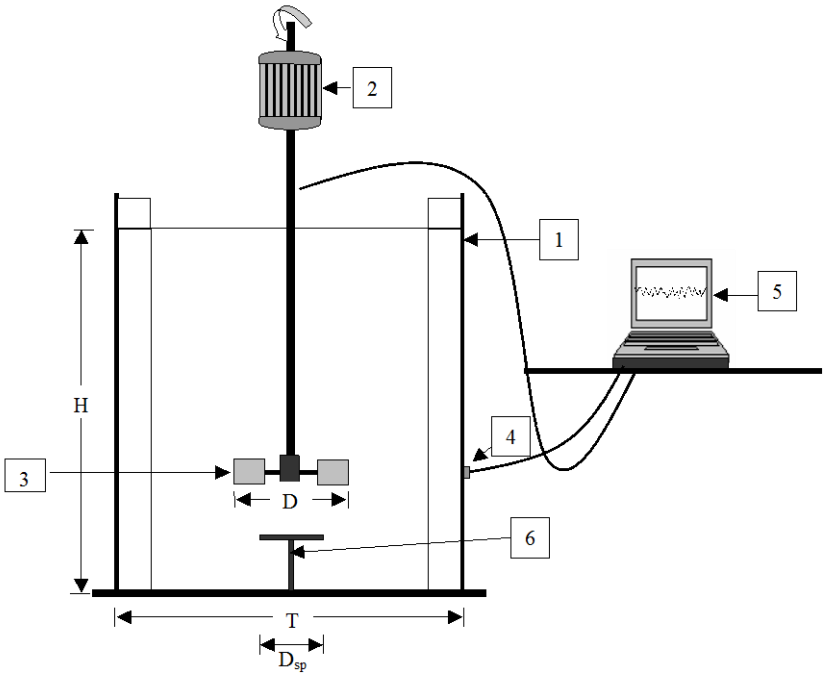
c) Solve the computational problem by defining an appropriate kernel in the input space in place of the dot product in the high dimensional feature space

d) Solve the dual formulation of the convex quadratic programming problem to obtain the unique global solution for the classifier.

The method presented for characterization of time series will now be applied to a case study of identification of flow regimes in stirred vessel with Rushton turbine. The experimental setup is described briefly in the following section.

5.3. Experimental Setup

The measurements were carried out in a fully baffled, flat bottom acrylic vessel (of diameter, T = 0.7 m and liquid height, H = 0.7 m). A schematic diagram of the experimental setup is shown in Figure 5.4. Four baffles of width T/10 were mounted diametrically opposite and perpendicular to the vessel wall. The shaft of the impeller (d_s = 0.032m) was concentric with the axis of the vessel and was extended till the impeller off-bottom clearance. Rushton turbine (of diameter, D = 0.2 m; impeller blade width, W = D/4 and impeller blade height, B = D/5) was used during experiments. The impeller off-bottom clearance was (C = T/3) measured from the bottom of the vessel to the center
of the impeller disc for Rushton turbine. The gas was introduced in the vessel through a ring sparger (of diameter, D_{sp} = 0.16 m, 12 holes with 1 mm diameter) and it was located at 0.16 m from the bottom of the vessel. The working fluids were water and compressed air in all the experiments. The measurements were carried out for three values of Froude number 0.6, 0.85 and 1.03 for Rushton turbine (power consumption varied between 0.96 kW/m^3 to 2.2 kW/m^3 for single-phase flow) and the gas flow number was varied between 0 and 0.37 for Rushton turbine. The operating conditions were selected in such a way so as to adequately represent all the key flow regimes occurring in gas-liquid stirred vessels.



1. Vessel, T = H = 0.7 m
2. Motor
3. Rushton turbine, D = 0.2 m
4. Pressure transducer
5. Data acquisition system with personal computer
6. Ring sparger, D_{sp} = 0.16 m, hole dia. = 1 mm

Figure 5.4: Schematic diagram of the experimental set-up

Pressure transducer with range of ± 34.46 kPa, resolution of 0.000482 kPa and sensitivity of 72.54 mV/kPa was used (of PCB Piezoelectronics Inc., USA, Model 106B50) to measure the wall pressure fluctuations. The pressure sensor was flush mounted on vessel wall at a height of impeller off-bottom clearance. The transducer was powered by ICP battery unit (PCB Piezoelectronics Inc., USA, Model 480E06), which also acted as an amplifier. The pressure fluctuations were acquired with a sampling frequency of 400 Hz (around 10 data points for blade passage) and the signal was acquired for 25 seconds. In the present study the low pass filter was used as per described in Khopkar et al.(2005) for filtering the experimentally measured time series.

Measurements were always performed in the same manner, starting from low to high impeller speeds with stepwise increase in the gas flow rate at a

constant impeller speed. The amplified signal from pressure sensor was acquired using a laptop computer with 16-bit A/D PCMCIA converter card and data acquisition software 'dAtagate' (of nCode, UK). The method proposed for characterization of time series as described in section 5.2 will be applied for analyzing the signals of pressure fluctuations in stirred vessel for identification of the flow regime.

5.4. Results and Discussion

5.4.1 Singularity distribution analysis of flow data

Pressure time series data were recorded with frequency of 400 Hz and the total acquisition time duration was 25 sec. Thus each time series consisted of 10,000 uniformly spaced points. The record consists of a total of 272 time series, which includes 152, 32 and 88 belonging to loading, flooding and fully dispersed regimes respectively. For each time series we performed the analysis as explained in section 5.2.2 to estimate the local Hölder exponents and their probability densities using kernel density estimation. The density spectra obtained for one illustrative time series data from each regime are shown in Figure 5.5. In the figure solid line shows the plot of Hölder exponent h v/s the probability density estimates for the fully dispersed regime, whereas the dotted and dashed lines highlight the same for loading and flooding regime respectively. We first extracted the density estimates corresponding to Hölder exponents in the range {-0.05, 0.45} with an interval of 0.05. This has resulted in eleven singularity features for each time series which were subsequently fed to the SVM classifier as inputs.

Along with these singularity distribution features we have also estimated statistical properties of the time series as features (Xie et al., 2003). The selected statistical features include standard deviation, coefficient of skewness, coefficient of kurtosis and second order correlation terms $Co(2)$, $Co(5)$, $Co(10)$, $Co(20)$, $Co(50)$, $Co(100)$ and $Co(200)$ of the normalized pressure signals. These correlation terms of the normalized pressure signal can be defined as,

$$Co(d) = \overline{p^*(t)p^*(t+k)} \tag{28}$$

where parameter k is the time shift and defined $k=d/R$, where the R is the rate at which the signal is measured.(for our experiments R=400 Hz). Normalized pressure fluctuation, p^* is defined as

$$p^* = (p - \overline{p}) / \sqrt{\overline{(p-\overline{p})^2}} \tag{29}$$

These statistical features were separately employed as inputs to another SVM classifier for the purpose of comparison of performance.

5.4.2 Characterization of flow data with SVM

We have extracted eleven singularity and ten statistical features for each time series. First the singularity features were employed as input data to the SVM multiclass classifier. The output of the trained SVM classifier would readily identify the flow regime of any time-series of the pressure fluctuation data. SVM identification of the three flow regimes is essentially a multiclass classification problem. This can be solved by one against one method.(Kreßel,1999) This method considers the problem as a collection of multiple binary classification problems. In general $m(m-1)/2$ classifiers are needed to solve the m class problem. Identifying the three classes corresponding to fully dispersed, loading and flooding regimes as I, II and III respectively, we need to build three binary classifiers. The first one classifies I vs II, the second one classifies I vs III and the third one classifies II vs III. Finally, the decision for class affiliation is made through a majority vote across the classifiers.

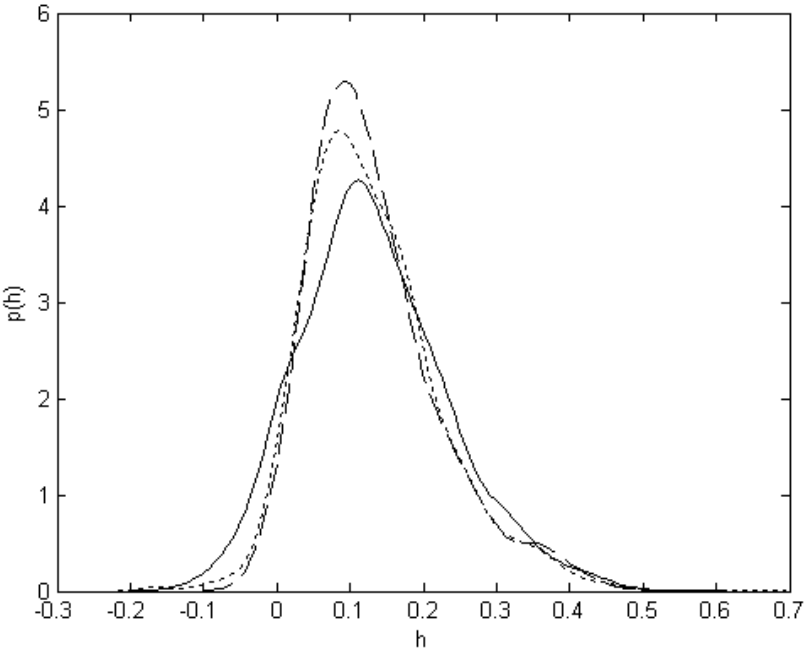


Figure 5.5: Local Hölder estimation and its probability densities: fully dispersed regime, loading and flooding are denoted by solid line, dotted line and dashed line respectively.

SVM being a supervisory method, the classifier is trained with data whose class labels are known a priori. In other words for the regime identification problem, we must know the corresponding regimes of the pressure fluctuation data employed for training the SVM classifier. In our case study the regime map data obtained earlier by visual and power spectrum analysis was used. Out of a total of 272 time series data available from existing regime map, a set of randomly chosen 222 time series were treated as the training set and the remaining 50 were used as a test set. As explained earlier, the three class regime identification problem was converted into three equivalent binary classifier problems. Extensive simulations were conducted to train these classifiers with a view to obtain maximum discriminatory power. Conventional five fold cross validation methodology was used to obtain optimal SVM model parameters. In this methodology, the entire training set was partitioned into five parts. Four parts were used as the training set, and the remaining part was used as the validation set. This process was repeated until each of the partitioned parts was used as the validation set. The model parameters (regularization parameter *C* and the spread

parameter in Gaussian kernel, σ) that result in the least average error were chosen as the optimal parameters. Employing the multiclass classifier trained as per the above methodology, we could correctly classify 49 data-points out of 50 from the test data (98% accuracy). The misclassified point is shown in the regime map (Figure 5.6). As one can see from the figure, the misclassified point belongs to the flow-transition region. The data was also classified into different regimes using the statistical features independently and gave the classification accuracy of 92%. Thus the performance of the hybrid combination of WTMM-singularity distribution density estimation-SVM classification methodology is superior and provides better flow regime identification.

The techniques presented here, although not illustrated for industrial data, could be used with commercially available industrial pressure sensors. The analysis and detection of flow regimes is fairly straightforward and unambiguous and looks promising for applications to industrial gas-liquid stirred vessels. The proposed methodology can also be readily applied to other multiphase systems like bubble column, fluidized bed etc.

5.5. Summary

A novel method for analysis and characterization of time series is proposed. This method is a unique combination of wavelet based singularity analysis and support vector machines classification. Proposed methodology was applied to a case study of flow regime identification in gas-liquid stirred tank equipped with Rushton turbine. Employing our method we could classify flow regimes with 98% accuracy. Also from the regime map it is clear that the misclassified data-point belongs to the regime transition zone. This proves the effectiveness of this method for the identification of the flow regime in gas-liquid stirred tank. The excellent classification accuracy brings out the fact that the local scaling behavior of a given regime follows a distinct pattern. Further, the singularity measures can be employed by intelligent machine learning based algorithms like SVM for online regime identification. The method is simple and can be generalized to the other multiphase systems like bubble column, fluidized bed etc.

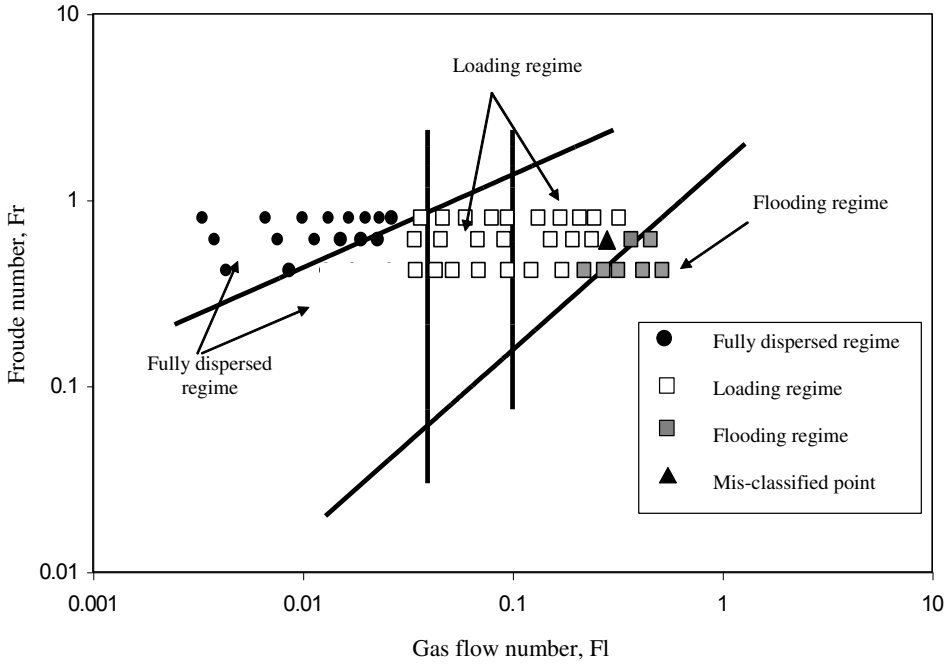


Figure 5.6: Flow regime map for stirred vessel equipped with Rushton turbine

References

Agarwal, M., Jade, A. M., Jayaraman, V. K., & Kulkarni, B. D., , "Support vector machines: a useful tool for process engineering applications", Chemical Engineering Progress, 98(1), 57-62, (2003).

Arneodo, A., Bacry E., & Muzy, J.F., "The Thermodynamics of Fractals Revisited with Wavelets", Physica A, 213, 232-275, (1995).

Bai, D., Shibuya, E., Nakagawa N., & Kato, K., "Characterization of gas fluidization regimes using pressure fluctuations", Powder Technology, 87(2), 105-111, (1996).

Bai, D., Shibuya, E., Nakagawa N., & Kato, K., "Fractal characteristics of gas-solids flow in a circulating fluidized bed", Powder Technology, 90(3), 205-212, (1997).

Bombac, A., Zun, I., Filipic, B., & Zumer, M., "Gas-filled cavity structure & local void fraction distribution in aerated stirred vessel", AIChE Journal, 43 (11), 2921- 2931, (1997).

Burges, C.J.C., "A tutorial on support vector machines for pattern recognition", Data Mining & Knowledge Discovery, 2(2), 121–167, (1998).

Chen, Y.-G, Tian, Z.-P., & Miao Z.-Q, "Detection of singularities in the pressure fluctuations of circulating fluidized beds based on wavelet modulus maximum method", Chemical Engineering Science 59(17), 3569 – 3575, (2004).

Chiang, L. H. , Kotanchek M. E., & Kordon, A. K., "Fault diagnosis based on Fisher discriminant analysis & support vector machines", Computers & Chemical Engineering, 28(8),1389-1401, (2004).

Ellis, N., Briens, L.A., Grace, J.R., Bi, H.T., & Lim, C.J., "Characterization of dynamic behaviour in gas–solid turbulent fluidized bed using chaos & wavelet analyses",
Chemical Engineering Journal, 96(1-3), 105–116, (2003).

Johnsson, F., Zijerveld, R.C., Schouten, J.C., van den Bleek, C.M., & Leckner, B., "Characterization of Fluidization regimes by time-series analysis of pressure fluctuations", International Journal of Multiphase Flow 26(4), 663-715, (2000).

Khopkar A. R, Panaskar , S. S., Pandit A. B., & Ranade V. V. , "Characterization of gas-liquid flows in stirred vessels using pressure & torque fluctuations", Industrial Engineering Chemistry Research (in press).

Kreßel. U.H.-G. "Pairwise classification & support vector machines", In B. Scholkopf, C. J. C. Burges, & A. J. Smola, editors, Advances in Kernel Methods: Support Vector Learning, pages 255–268. MIT Press, Cambridge, MA. (1999).

Kulkarni, A., Jayaraman, V. K., & Kulkarni, B. D., „Support vector classification with parameter tuning assisted by agent-based technique", Computers & Chemical Engineering, 28(3), 311-318, (2004).

Kulkarni A.A., Joshi J.B., Ravi Kumar V., & Kulkarni B.D., "Identification of the principal time scales in the bubble column by wavelet analysis", Chemical Engineering Science,56(20),5739-5747, (2001).

Lin, T.J., Juang, R.-C., & Chen, C.-C., "Characterizations of flow regime transitions in a high-pressure bubble column by chaotic time series analysis of pressure fluctuation signals", Chemical Engineering Science, 56(21-22), 6241–6247, (2001).

Letzel, H.M., Schouten, J.C., Krishna, R., & van den Bleek, C.M., „Characterization of regimes & regime transitions in bubble columns by chaos analysis of pressure signals", Chemical Engineering Science, 52(24), 4447-4459, (1997).

Mallat S., & Hwang, W. L., "Singularity detection & processing with wavelets", IEEE Transactions on Information Theory, 38(2), 617-643, (1992).

Mallat, S.G., "A wavelet tour of signal processing". 2nd Edition, Academic Press, Cambridge. (1999).

Muzy, J.F., Bacry, E., & Arneodo, A., "Wavelets & multifractal formalism for singular signals: application to turbulence data", Physics Review Letters. 67, 3515, (1991).

Muzy, J.F., Bacry, E., & Arneodo, A., "Multifractal formalism for fractal signals: The structure-function approach versus the wavelet-transform modulus-maxima method", Physical. Review E, 47, 875, (1993).

Muzy, J.F., Bacry E., & Arneodo, A., "The multifractal formalism revisited with wavelets", International Journal of Bifurcation & Chaos, 4(2), 245-302, (1994).

Nienow, A.W. “Hydrodynamics of stirred bioreactors”, Applied Mechanics Review, 51, 3-32, (1998),

Nienow, A.W., Warmoeskerken, M.M.C.G., Smith, J.M., & Konno, M. “On the flooding/ loading transition & the complete dispersal condition in aerated vessels agitated by a Rushton turbine”, Proceedings of the European Conference on Mixing, Wurzburg, Paper 15, (1985).

Paglianti, A., Pintus, S., & Giona, M., “Time-series analysis approach for the identification of flooding/loading transition in gas-liquid stirred tank reactors”, Chemical Engineering Science, 55(23), 5793-5802, (2000).

Park, S. H, Kang Y., & Kim, S.D., “Wavelet transform analysis of pressure fluctuation signals in a pressurized bubble column”, Chemical Engineering Science, 56, (21-22), 6259-6265, (2001).

Roy, M., Ravi Kumar, V., Kulkarni, B. D., Sanderson, J., Rhodes M., & vander Stappen M. “Simple Denoising Algorithm Using Wavelet Transform”, AIChE Journal, 45 (11), 2461-2466, (1999).

Scafetta, N., Griffin, L., & West B.J., “Hölder exponent spectra for human gait”, Physica A 328, 561-583, (2003).

Struzik, Z.R., “Removing divergences in the negative moments of the multi-fractal
partition function with the wavelet transformation”, CWI Report, INS-R9803. (1998)

Struzik, Z.R., “Determining local singularity strengths & their spectra with the wavelet transform”, Fractals, 8(2), 163–179, (2000),

Struzik Z.R.,& Siebes A.P.J. M., “Wavelet transform based multifractal formalism in outlier detection & localisation for financial time series”, Physica A: Statistical Mechanics & its Applications, 309(3-4), 388-402, (2002).

Sutter, T.A., Morrison, G.L., & Tatterson, G.B., “Sound spectra in an aerated agitated tank”, AIChE Jounal, 33 (4), 668-671, (1987).

Vapnik, V. , “The Nature of Statistical Learning Theory”, Springer, New York. (1995)

Vapnik, V. , “Statistical learning theory”. New York: Wiley (1998).

Warmoeskerken, M.M.C.G., & Smith, J.M. , “Flooding of disc turbines in gas-liquid
dispersions: A new description of the phenomenon”, Chemical Engineering Science, 40 (11), 2063-2071, (1985).

West B.J., Scafetta N, Cooke WH, & Balocchi R , “Influence of progressive central hypovolemia on Hölder exponent distributions of cardiac intervals”. Annals of Biomedical Engineering 32(8), 1077-87, (2004).

Wu, H., Zhou F., & Wu, Y., “Intelligent identification system of flow regime of oil-gas-water multiphase flow”, International Journal of Multiphase Flow 27(3) 459-475, (2001)

Xie, T., Ghiaasiaan, S. M., & Karrila, S., “Flow Regime Identification in Gas/Liquid/Pulp Fiber Slurry Flows Based on Pressure Fluctuations Using Artificial Neural Networks”, Industrial Engineering Chemistry Research 42, 7017-7024, (2003).

Xie, T., Ghiaasiaan, S.M., & Karrila, S. , “Artificial neural network approach for flow regime classification in gas–liquid–fiber flows based on frequency domain analysis of pressure signals”, Chemical Engineering Science, 59, 2241 – 2251, (2004).

Zhao G-B., & Yang Y-R, "Multiscale resolution of fluidized-bed pressure fluctuations". AIChE Journal 49(4), 869-882, (2003).

Notation

b bias term used in SVM classification

C Regularization parameter used in SVM classification

c , C_1 constant

Co correlation terms

$D(h)$ Singularity spectrum

h local Hölder exponent

$\bar{h}$ mean Hölder exponent

$\hat{h}(x_0, s)$ estimate of local Hölder exponent at scale s

$K(\overset{\rightharpoonup}{x}_i, \overset{\rightharpoonup}{x}_j)$ kernel function for the vectors $\overset{\rightharpoonup}{x}_i, \overset{\rightharpoonup}{x}_j$

M any integer

$M(s)$ function used to evaluate mean Hölder exponent

N total instances used in SVM classification.

N_F Critical impeller speed for flooding to loading transition

N_{CD} Critical impeller speed for loading to fully dispersed regime transition

p pressure fluctuation

p^* Normalized pressure fluctuation

$\bar{p}$ mean pressure fluctuation

Pn Polynomial of degree n

q any real number

s scale used in wavelet transform.

s_N length of the time series

$T_\psi[f](x_0, s)$ Wavelet transform at location x_0 and scale s.

$\overset{\rightharpoonup}{w}$ weight vector in SVM classification

$\overset{\rightharpoonup}{x}_i$ is a vector of input features of the i^{th} instance in SVM classification

x_r , x_s support vectors in SVM classification

y_i target class to which the i^{th} instance belongs to.

Z Partition function in WTMM method

Greek letters and miscellaneous

$\Phi(x_i)$ mapping of the input data $\vec{x}_i$ in the feature space in SVM classification

α Lagrange multipliers in SVM classification

$\tau(q)$ scaling exponents

ψ wavelet mother function

ζ slack variable used SVM classification

Chapter 6
IMPROVED TIME SERIES PREDICTION USING KERNEL METHODS WITH A NEW METHOD FOR SELECTION OF MODEL PARAMETERS

6.1. Introduction

Several methods have been proposed in literature for prediction of time series data (Casdagli, 1989, Farmer & Sidorowich, 1987; Belomestny et al. 2003; Navone & Ceccatto, 1995; Ho & Xie, 1998; Zhang et al., 1998; Connor et al., 1994; Rosen-Zvi et al., 2003; Elsner, 1992; Small & Tse, 2002 ; Freking et al., 2002 ; Müller et al., 1997; Mukherjee et al., 1997; Rosipal et al., 2001; Jade et al., 2003). The most familiar approaches include the linear methods such as ARX, ARMA, etc. and the nonlinear methods such as algorithms based on artificial neural networks (Zhang et al., 1998; Connor et al., 1994; Rosen-Zvi et al., 2003 ; Elsner, 1992; Small & Tse, 2002 ; Freking et al., 2002). Recently kernel based machine learning tools like support vector regression (SVR) and kernel principal component regression (KPCR) have become very popular because of their state-of-the-art performance (Müller et al., 1997; Mukherjee et al., 1997; Rosipal et al., 2001; Jade et al., 2003). All these methods split the data into three disjoint sets, viz., training, validation and test sets. Subsequently, the model parameters in the algorithm are optimized by minimizing the root mean square error of the predicted validation set. Finally, the performance is gauged by the test error. For time series data possessing sharp changes, selection of model parameters based only on the criterion of RMS error may produce higher generalization errors. There is therefore a need for a robust measure, which will take account of these sharp changes or singularities occurring in a time series. A methodology that picks up the local scaling behavior of the time series would be able to readily reveal such singularities.

In this chapter we present wavelet transform modulus maxima (WTMM) based method for characterizing and quantifying the singularities in a chaotic time series (Struzik, 2000). The method provides the density estimates of the local Hölder exponents that characterize the regular/ irregular local behavior of time series. Higher the value of the local Hölder exponent, more regular is the local behavior of time series and vice versa. The density estimates of the local Hölder

exponents represent the most informative features regarding the singularities in the time series. Thus an estimate of the error in the density spectrum of the predicted validation set, which we will henceforth denote as, HRSME, can be very useful for tuning the model parameters. For certain time series data it may be possible that both RMSE and HRMSE information would be useful for obtaining optimal performance. In this work, errors in the density estimates (HRMSE) (along with regular RMSE) of the validation set have been employed as an additional criterion for selection of optimal model parameters. Thus the problem of model selection is formulated in terms of a multiobjective optimization i.e. the selection of model parameters has been done by minimizing both criteria viz. RMS errors based on the original time series as well as based on the error in the singularity distribution. In this problem, we have to find the decision vector (parameters of model used for time series prediction), which will minimize RMSE as well as HRMSE. Multi-objective optimization, however, gives rise to a set of optimal solutions, instead of a single one (Zitzler et al.). These optimal solutions are called as Pareto-optimal solutions. This concept can be illustrated employing Figure 6.1, which depicts a plot of RMSE vs. HRMSE. The point 'P' represents the solution with minimum RMSE but has higher HRMSE and point 'Q' represents least HRMSE but high RMSE. Since both objectives are equally important one cannot say that solution 'P' is better than 'Q' or vice versa. All such solutions (marked by the dash line) are Pareto-optimal solutions. Also in the figure, there are few points (e.g. 'S') which are not members of the Pareto set. It can be seen that solution 'R' in the decision space has lower RMS and HRMS errors and hence is better than solution 'S' considering both the objectives. Thus solutions like 'S' are known as dominated or inferior solutions and solutions like 'R' belonging to the Pareto-optimal set are often called as non-dominated solutions. Also it is clear that no solution in the Pareto-optimal set is better or worse than the other considering both the objectives (RMSE and HRMSE). In the present study we have used this concept of non-dominated Pareto-optimal solutions for finding the optimal parameters of kernel principal component regression (KPCR) model to improve the generalization capability of the model. Kernel PCA, a nonlinear version of PCA, has recently been extensively used because of its computational simplicity and nonlinear feature extraction and denoising capabilities (Rosipal et al., 2001 ; Jade et al., 2003; Schölkopf et al.

1998). We have chosen KPCR because of its excellent performance on time series prediction problems (Rosipal et al., 2001 ; Jade et al., 2003). Moreover, only two parameters are needed to be tuned for model selection of KPCR. The efficacy of the proposed method has been tested on two simulated chaotic time series and one time series based on real observations.

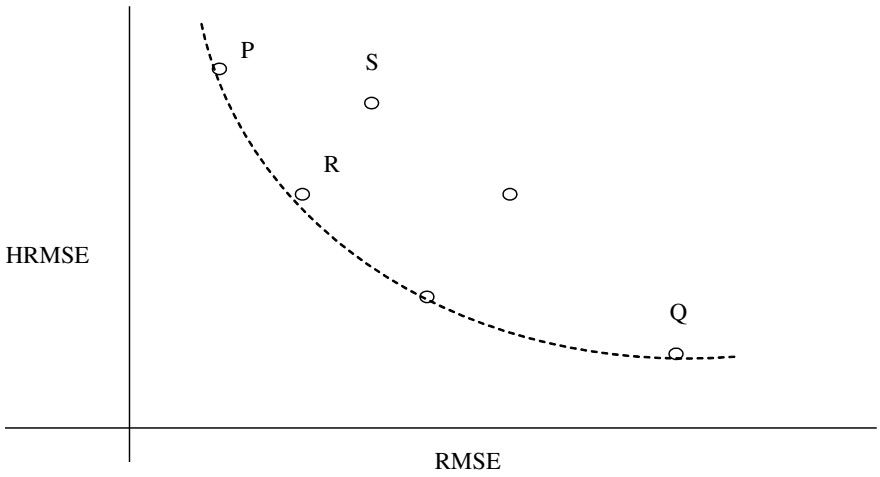


Figure 6.1: Concept of Pareto-optimal solutions in Multi-objective optimization.

The chapter is organized as follows: in the next section (section 6.2), we have described the key steps involved in the proposed algorithm. Section 6.3 includes the discussion on characterizations of the singularities and their analysis based on wavelet transform modulus maxima (WTMM) method. In section 6.4, we have illustrated KPCR for time series prediction. Section 6.5 includes the case studies used for time series prediction. Section 6.6 comprises the results and discussions and section 6.7 provides salient conclusions of the work.

6.2. Proposed Algorithm for KPCR Model Selection

The key steps involved in the proposed algorithm are described below

i) Divide the available time series data into three segments namely training, validation and test.

ii) For various model parameters.

   a) Build up the KPCR model using the training data as described in section 6.4.

   b) Predict the validation time series using the model.

c) Estimate the RMSE and HRMSE for validation time series employing the method described in section 6.3.

iii) Get the Pareto-optimal solutions using RMSE and HRMSE criteria

iv) Estimate the test error using the Pareto-optimal model parameter set.

6.3. Singularity Analysis using WTMM

The distribution of singularities and the singularity spectrum can be obtained from the well-known wavelet transform modulus maxima (WTMM) based multifractal formalism (Muzy et al. 1991; Muzy et al. 1993 ; Muzy et al. 1994). The method offers global estimates of scaling properties for characterization of a multifractal time series. The spectrum of the singularities as described in (Muzy et al. 1991; Muzy et al. 1993 ; Muzy et al. 1994) poses certain problems of stability when applied to observational data (Struzik, 1998). Recently, Struzik (2000) has presented a stable method for evaluating the estimation of local singularity strengths. In his methodology for estimating the local Hölder exponents, he has modeled the singularities as if they were created through a multiplicative cascading process. The method has been described in brief here and for more details readers may refer to (Struzik, 1998 ; Struzik, 2000; Struzik & Siebes, 2002; Scafetta et al. 2003 ; West et al. 2004; Jade et al. 2006)

It can be shown that by employing the multiplicative cascade model the estimate of local Hölder exponent, $\hat{h}(x_0, s)$ at the singularity x_0 and scale s can be evaluated as (Struzik, 2000; Struzik & Siebes, 2002)

$$\hat{h}(x_0, s) = \frac{\log\left(\left|T_\psi[f](x_0, s)\right|\right) - (\bar{h}\log(s) + c)}{\log(s) - \log(s_N)} \qquad (1)$$

where s_N is the maximum available scale and $T_\psi[f](x_0, s)$ is the maxima of wavelet coefficient at location x_0 and at scale s. In our work we have used Mexican Hat wavelet, which is the second derivative of the Gaussian function. The mean Hölder exponent $\bar{h}$ in Eq.(1) can be estimated as a linear fit of the following equation,

$$\log[M(s)] = \bar{h}\log(s) + c \tag{2}$$

where function M(s) is obtained from the partition functions,

$$M(s) = \sqrt{\frac{Z(s,2)}{Z(s,0)}}. \tag{3}$$

Partition function $Z(s,2)$ can be calculated as the sum of squares of the maxima of $\left|T_\psi[f](x_0, s)\right|$ at the scale s and $Z(s,0)$ is the number of maxima at scale s. The local Hölder estimates (from Eq. (1)) and their density spectrum can be estimated for original and predicted time series. The error on distribution of these estimates can be used as criterion for model selection of time series prediction problems using KPCR.

6.4. Kernel Principal Component Regression

KPCR has been chosen as a nonlinear regression method because of its successful applications in time series prediction. (Rosipal et al., 2001; Jade et al., 2003). Moreover, KPCR requires only two parameters to be tuned for its model selection. Kernel principal component analysis (kernel PCA) corresponds to linear PCA in a higher dimensional feature space, which is nonlinearly related to the input space. The input data x are first mapped through some appropriate nonlinear function Φ(x). The problem formulation is in terms of dot product of the input data in the feature space that can be substituted by a kernel function. Thus a priori defined kernel function is used to deal with the very high dimensional space and the dot product of transformed input vectors can be computed in the input space itself (Vapnik , 1998). Kernel PCA and PCR has been extensively used for the purpose

of nonlinear feature extraction and denoising. (Rosipal et al., 2001 ; Jade et al., 2003; Schölkopf et al. 1998). KPCR has a definite advantage in dealing with multi-collinearity and noise and allows us more flexibility in retaining principal components to capture the underlying nonlinear features where data observations are more than the dimensionality (Rosipal et al., 2001). The method has been described in brief here and for details readers may refer to (Rosipal et al., 2001; Jade et al., 2003; Schölkopf et al. 1998).

Consider a set of M centered input variables (regressors), $\{\mathbf{x}_k\}_1^M \in \mathbf{R}^N$ and output (response) variables $\{\mathbf{y}_k\}_1^M \in \mathbf{R}^l$.

The kernel principal component regression model for the prediction of response variable from any input vector, x can be expressed as (Rosipal et al., 2001)

$$f(\mathbf{x}) = \sum_{k=1}^{p} w_k \sum_{i=1}^{M} \alpha_i^k K(\mathbf{x}_i, \mathbf{x}) + b \qquad (4)$$

where p is the number of principal components retained in KPCR model ($p \le M$) and $K(\mathbf{x}_i, \mathbf{x})$ can be estimated using the kernel function. The variables α_i^k are computed by diagnolization of kernel matrix of the input variables and w_k are the least squares estimates of regression coefficients. For the centralized regression model bias, b is zero. In our study we have employed Gaussian kernel, which is defined as,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{L}\right)$$

where L is the width of the Gaussian function. The number of principal components retained, p and the width parameter, L are the two parameters that need to be tuned in fixing KPCR model.

6.5. Case Studies

In our analysis we have considered the three time series, two simulated and one based on real observations.

6.5.1 Simulated time series

In our simulations we have chosen two important benchmarking time series examples, viz., Lorenz system (Lorenz, 1963) and Mackey-Glass (Mackey & Glass, 1977) governed by the following set of equations.

6.5.1.1.Lorenz system

$$\frac{dx}{dt} = -\sigma x + \sigma y$$
$$\frac{dy}{dt} = R x - y - x z$$
$$\frac{dz}{dt} = -b z + x y$$

with $\sigma = 10, R = 28, b = 8/3$.

The time series data has been generated by integrating sets of equations using a standard Runge-Kutta routine with a step size of 0.01. The training set consisted of 500 delay vectors, formed by using an embedding dimension of 3 and a time delay of 16 for Lorenz series. The validation and test set consisted of similarly embedded 300 and 2000 respectively.

6.5.1.2 Mackey-Glass

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1 + x^{10}(t-\tau)} - bx(t)$$

where a=0.2, b=0.1, τ=17. We predict the x(t+6) using the input variables x(t), x(t-6), x(t-12) and x(t-18), respectively (Chen et al. 2006). We have used training, validation and test set of size 500 each.

6.5.2 Laser data

We have studied the prediction of real infrared NH_3 laser data (Hübner et al. 1989), contributed by U. Hübner to the Sante-Fe Institute prediction contest. This data set contains two sets, consisting 1000 and 10000 points. (http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html). We have used embedding dimension of 4, and chosen the delay of 2 (Bollt, 2000). First 1000 points of first set are used for training and the first 400 points of second set are

used for validation and the following 8000 points of the second set are used for test.

6.6. Results and Discussions

We have used KPCR for time series prediction of the case studies described in section 6.4. Different models can be generated using various free parameters of KPCR algorithm. For KPCR we need to tune the two parameters viz.: the number of principal components retained and width parameter in Gaussian kernel. For studying the effect of model selection criteria on the performance of time series prediction, we have divided the given time series into three parts viz.: training, validation and test. Simulations can be performed with different model parameters employing the training data and the set of parameters that predicts the least error on validation data will be selected as the optimal set and further can be employed for the prediction of the unseen test data. Generally, the RMS error on the validation data is used as an objective for selecting optimal model parameters. In other words, the parameters that yield the least RMS error on validation data are used for predicting unseen time series. In this study, we have proposed one additional criterion for selection of optimal set of parameters based on the distribution of the local Hölder estimates. The method of evaluating the proposed criterion is as follows: i) For each set of model parameters, estimates of the local Hölder exponents are evaluated for both original time series and the predicted time series of validation data using the algorithm described in section 6.3. ii) The probability density spectrums were then obtained and the root mean square error between the densities of the Holder exponents (HRMS error) is found. For the least HRMS error, the singularity distribution of predicted time series is closest to the singularity distribution of actual time series. Thus HRMS error can be used as an objective for optimizing the model parameters of time series with sharp changes. In our study, we have used both criteria of RMS and HRMS error and solved the problem of obtaining the optimal parameters as a multiobjective optimization. We have varied the two parameters of KPCR viz.: parameter (L) in Gaussian kernel and p (the number of principal components retained). Simulations were conducted with more than 10000 sets of the model parameters. From the solutions thus obtained, we get a set of Pareto-optimal solutions as

described in first section (6.1). After getting Pareto set, test error for each set of parameters in this set is estimated.

Pareto sets obtained for Lorenz, Mackey-Glass and laser time series are shown in Tables 6.1a, 6.1b and 6.1c respectively. In tables the results marked by bold letters are obtained by using one of the objectives (either RMSE or HRMSE). As can be seen from Table 1a, the model parameters (p=193 and L =1.08) has produced the least RMS error of 0.0342 on validation data. Using only RMS error as a criterion, one may choose these parameters as optimal parameters and directly employ it for predicting test data. These parameters lead to a test error of 0.0835. For Lorenz series minimum HRMS error for validation data is obtained for the set of parameters (p=170 and L =1.32), which results a test error of 0.0614. Thus use of HRMS error as a criterion has lead to improvement of 26.47% in predicting the unseen test data. A similar trend is observed in case of laser and Mackey-Glass time series. Employing the criterion of RMS error the set of parameters (p=330 and L = 0.2100) has been selected for laser time series which yields test error of 10.7952, whereas optimal parameters (p=281 and L = 0.4100) selected on the basis of new criterion results in test error of 9.6211. Thus application of HRMS error criterion has shown improvement of 10.88% over the conventional criterion of RMS error in predicting the test data of laser time series. Similarly an improvement of 5.78% is obtained by the application of the proposed criterion in predicting unseen Mackey-Glass time series.

The reason for the superior performance can be better understood from Figure 6.2. We have shown in this figure, the distribution of local Hölder exponents for the predicted and actual laser time series (validation data) for KPCR models selected by conventional and singularity based criterion. As can be seen from the figure, the distribution of Hölder exponents for the predicted time series is closer to the distribution of the same for the actual time series in case of the model selected by the singularity based criterion than the one corresponding to the conventional criterion. The superior performance of the proposed method can be attributed to the fact that the model selected by minimum HRMS criterion perfectly captures local singular behaviour of the time series and thus helps in improving the generalization capability of the model.

Table 6.1a: Non-dominated solutions for Lorenz time series

Num of Prin. Comp.	Kernel parameter	Validation RMSE	Validation HRMSE	Test error
180	*1.0000*	*0.0414*	*0.0237*	*0.0696*
193	1.0400	0.0357	0.0277	0.0782
193	**1.0800**	**0.0342**	**0.0282**	**0.0835**
193	1.1000	0.0343	0.0279	0.0852
170	**1.3200**	**0.0439**	**0.0138**	**0.0614**
170	1.3400	0.0439	0.0192	0.0623

Though employing HRMSE criterion has lead to better results than the conventional criterion of RMSE in all of the present case studies, there is a risk involved in choosing the model parameters based only on one of the criteria for prediction of any real time series at hand. If the range for the validation errors (both RMSE and HRMSE) for the Pareto-optimal set is narrow, then selection of parameters using either of the criteria will not make any significant difference.

Table 6.1b: Non-dominated solutions for Mackey-Glass time series

Num of Prin.Comp.	Kernel parameter	Validation RMSE	Validation HRMSE	Test error
343	**0.94**	**0.000473**	**0.065422**	**0.00019**
379	0.94	0.000517	0.039667	0.000185
343	0.96	0.000474	0.062189	0.000188
378	0.98	0.000532	0.039406	0.000187
379	1.0	0.000535	0.038293	0.000181
381	1.0	0.000538	0.032960	0.000182
381	**1.02**	**0.000538**	**0.031127**	**0.000179**
347	*1.10*	*0.000495*	*0.050874*	*0.000173*

But when the range for one or both of the errors is wide, one should choose the model parameters, which are performing better in both criteria. Thus for instance for Lorenz series we will choose the parameters p=180 and L=1 as these parameters produce lesser RMSE as well as HRMSE. The point with the optimal parameters is marked by the circle in Figure 6.3. By using these parameters we have obtained the test error of 0.0696 which is much lesser than the error given by the parameters using conventional criterion. In a similar way the optimal parameters performing well in both criteria are found for other case studies and are shown in Italic letters in the Tables 1a, 1b and 1c. One can conclude from these tables that the selection of parameters employing both criteria has resulted in lesser error on unseen test data than the errors produced by the parameters based on conventional criterion alone.

Table 6.1c :Non-dominated solutions for Laser time series

Num of Prin. Comp.	Kernel parameter	Validation RMSE	Validation HRMSE	Test error
330	**0.21**	**8.4848**	**0.3002**	**10.7952**
331	0.21	8.4856	0.2891	10.7954
320	0.25	8.4881	0.2523	10.3709
316	0.27	8.4912	0.1098	10.1842
317	0.27	8.4992	0.1025	10.1901
318	*0.27*	*8.4982*	*0.1030*	*10.1890*
261	0.31	8.5920	0.0997	10.0809
262	0.31	8.6037	0.0758	10.0494
262	0.33	8.5996	0.0805	9.9509
263	0.33	8.5962	0.0821	9.9542
263	0.35	8.5952	0.0840	9.8467
264	0.35	8.5951	0.0884	9.8450
281	**0.41**	**8.6717**	**0.0753**	**9.6211**

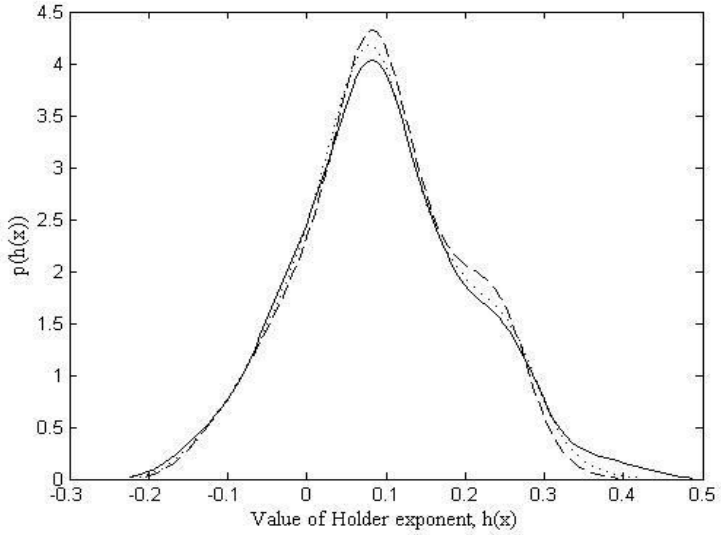


Figure 6.2: The distribution of the local Hölder exponents for the predicted and actual laser time series (validation set) using KPCR models. Distribution of the Hölder exponents of the actual time series is denoted by solid line and the distribution of predicted time series with a model selected by conventional criterion and singularity based criterion alone are denoted by dashed and dotted line respectively.

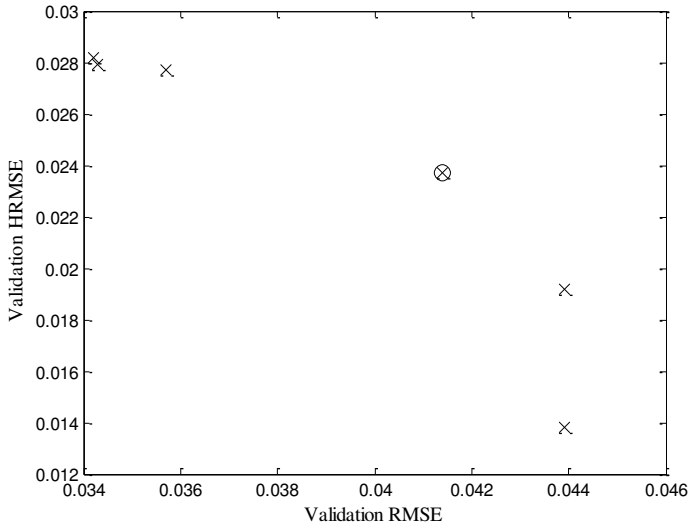


Figure 6.3: Non-dominated solutions for Lorenz time series (optimal point considering both objectives is marked by circle)

6.7 Summary

A new method for model selection in prediction of time series is proposed. Generally, the model parameters are selected for which the minimum RMS error is obtained on the predicted time series. This conventional criterion based only on RMS error may not take into account of the sharp changes or singularities in a time series and may fail in a case where this contribution of singularities is significant. Here, we have proposed an additional criterion based on error on the distribution of singularities in the predicted and actual time series. The distribution of singularities is evaluated through the local Hölder estimates and its probability density spectrum. Thus, the problem of model selection is solved by simultaneously minimizing both criteria, viz., RMS error based on the original time series as well as based on the error in the singularity distribution. The method is tested on three time series: two simulated and one based on real observations. Predictions of these time series have been done using kernel principal component regression (KPCR) and model parameters of KPCR have been selected employing the proposed as well as the conventional method. The problem now being a multiobjective optimization problem, we get a set of Pareto optimal solutions. Results obtained demonstrate that the proposed method helps in better prediction of the unseen test data and improves the generalization capability of the KPCR model. Model selection has produced the results which are better than the results yielded by the conventional method in all of the cases of the simulated and real time series. In general, we conclude that new method can be very useful in prediction of time series data having sharp singularities.

References

Belomestny D, Jentsch V & Schreckenberg M., " Completion and continuation of nonlinear traffic time series: a probabilistic approach", J. Phys. A: Math. Gen., 36, 11369-11383, (2003).

Bollt E M , "Model Selection, Confidence, and Scaling in Predicting Chaotic Time-Series", Int. J. Bifurcation & Chaos, 10, 1407-1422, (2000).

Casdagli M , "Nonlinear prediction of chaotic time series", Physica D, 35, 335-356, (1989)

Chen Y, Yang B & Dong J, "Time-series prediction using a local linear wavelet neural network", Neurocomputing, 69, 449-465, (2006).

Connor J T, Martin R D & Atlas L E, "Recurrent neural networks and robust time series prediction", IEEE Trans. Neural Networks 5 240-254, (1994).

Elsner J B., "Predicting time series using a neural network as a method of distinguishing chaos from noise", J. Phys. A: Math. Gen., 25, 843-850, (1992).

Farmer J D & Sidorowich J J., "Predicting chaotic time series", Phys. Rev. Lett., 59, 845-848, (1987).

Freking A, Kinzel W & Kanter I, "Learning and predicting time series by neural networks", Phys. Rev. E, 65, 050903, (2002).

Ho S L & Xie M., "The use of ARIMA models for reliability forecasting and analysis", Computers & Industrial Engineering 35 213-216, (1998).

Hübner U, Abraham N B & Weiss C O, "Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared NH3 laser", Phys. Rev. A 40 6354-6365, (1989).

Jade A M, Srikanth B, Jayaraman V K, Kulkarni B D, Jog J P & Priya L. "Feature extraction and denoising using kernel PCA", Chem. Eng. Sci., 58, 4441-4448, (2003)

Jade A M, Jayaraman V K, Kulkarni B D, Khopkar A R, Ranade V V & Sharma A., "A novel local singularity distribution based method for flow regime identification: Gas-liquid stirred vessel with Rushton turbine", Chem. Eng. Sci., 61, 688-697, (2006).

Lorenz E N, "Deterministic non-periodic flows", J. Atm. Sci., 20, 130-141, (1963).

Mackey M C & Glass L, "Oscillation and chaos in physiological control systems", Science, 197, 287-289, (1977).

Mukherjee S, Osuna E & Girosi F, " Nonlinear Prediction of Chaotic Time Series using Support Vector Machines" Proc. of IEEE NNSP'97, Amelia Isl&, FL, 24-26 Sep., 1997, (1997).

Müller K-R, Smola A J, Rätsch G, Schölkopf B, Kohlmorgen J & Vapnik V 1997 "Predicting time series with support vector machines", In Articial Neural Networks - ICANN'97, eds W. Gerstner, A. Germond, M. Hasler, & J.-D. Nicoud, pages 999-1004. Springer.

Muzy J F, Bacry E & Arneodo A, "Multifractal formalism for fractal signals: The structure-function approach versus the wavelet-transform modulus-maxima method", Phys. Rev E, 47, 875, (1993).

Muzy J F, Bacry E & Arneodo A, "The multifractal formalism revisited with wavelets", Int. J. Bifurcation & Chaos, 4, 245-302, (1994).

Muzy J F, Bacry E & Arneodo A. "Wavelets and multifractal formalism for singular signals: application to turbulence data", Phys. Rev. Lett., 67, 3515, (1991).

Navone H D & Ceccatto H A , "Forecasting chaos from small data sets: a comparison of different nonlinear algorithms ",J. Phys. A: Math. Gen., 28, 3381-3388, (1995)

Rosen-Zvi M, Kanter I & Kinzel W., "Time series prediction by feedforward neural networks-is it difficult?", J. Phys. A: Math. Gen., 36, 4543-4550, (2003).

Rosipal R , Girolami M, Trejo L J & Cichocki A, " Kernel PCA for feature extraction and de-noising in non-linear regression", Neural Comput & Applic, 10, 231-243, (2001).

Scafetta N, Griffin L & West B J, "Hölder exponent spectra for human gait", Physica A 328, 561 -583, (2003).

Schölkopf B, Smola A J & Müller K R., "Nonlinear component analysis as kernel eigenvalue problem", Neural Computation, 10, 1299-1319, (1998).

Small M & Tse C K, "Minimum description length neural networks for time series prediction", Phys. Rev. E, 66, 066701, (2002).

Struzik Z R & Siebes A P J M , "Wavelet transform based multifractal formalism in outlier detection and localisation for financial time series", Physica A 309 388-402, (2002).

Struzik Z R , "Determining local singularity strengths and their spectra with the wavelet transform", Fractals, 8, 163-179, (2000).

Struzik Z R, "Removing divergences in the negative moments of the multi-fractal partition function with the wavelet transformation", CWI Report, INS-R9803, ISSN 1386-3681, (1998).

Vapnik V , " Statistical learning theory", New York: Wiley, (1998).

West B J, Scafetta N, Cooke W H & Balocchi R , "Influence of progressive central hypovolemia on exponent distributions of cardiac intervals", Annals of Biomedical Engineering 32, 1077-87, (2004).

Zhang G, Patuwo E B & Hu M Y, "Forecasting with artificial neural networks: the state of the art", Int. J. Forecasting, 14, 35-62, (1998).

Zitzler E, Laumanns M & Bleuler S. , "A tutorial on evolutionary multiobjective optimization, In Metaheuristics for Multiobjective Optimisation", eds X. G&ibleux, M. Sevaux, K. Sörensen & V. T'kindt pages. 3-37, Springer. Lecture Notes in Economics & Mathematical Systems Vol. 535, Berlin.(2004)

Chapter 7

CONCLUSIONS

Nonlinear static and dynamic process modeling, fault detection and diagnosis, employing artificial intelligence tools like neural networks and fuzzy logic, have received considerable importance in recent years. Hybrid combinations of these algorithms and newer machine learning tools are also being developed with a view to increasing robustness and prediction capabilities. In the past few years, kernel methods like support vector machines (SVM) have become one of the most popular approaches within the machine learning community due to the possibility of building non-linear versions of classical linear algorithms in an easy and elegant way. The basic idea in kernel methods is to map data in the input space to higher dimensional feature space using some nonlinear mapping and then apply linear algorithm in that space. The computational difficulty arising out of high dimensionality of the feature space is handled by defining an equivalent *kernel function* in the input space itself. A family of kernel methods mainly includes support vector machines (SVM), kernel principal component analysis (kernel PCA), support vector regression (SVR) and support vector domain distribution (SVDD). The main objectives of the work were to apply these kernel based machine learning tools to solve process engineering problems. The chapters 2 to 4 dealt with applications of kernel methods for fault detection/ diagnosis and nonlinear modeling of chemical engineering systems, while in chapter 5 and 6 these tools were combined with wavelet-fractal theory for analysis, characterization and prediction of chaotic time series.

The working of support vector machines based on structural risk minimization principle in learning tasks involving linear and nonlinear classification and regression has been highlighted in chapter 2. The applications of SVM methodology were illustrated by considering the case studies of fault detection in CSTR and quantitative structure property relations (QSPR) problem dealing with prediction of boiling points of aliphatic hydrocarbons from molecular descriptors data. SVM successfully classifies and sub-classifies various types of faults occurring in simulated CSTR using one against all multi-class strategy. For

the QSPR problem, SVM obtains smaller errors for training, validating and testing sets than the ones obtained by using back propagation networks. The examples clearly demonstrate the ease, elegance and superiority of this new tool over the other conventional tools and should prove useful in a number of other process engineering applications.

In chapter 3, kernel PCA, a new method for performing nonlinear principal component analysis has been illustrated by considering the examples of (i) denoising of chaotic time series and, (ii) development of an input-output model for the case of polymer nanocomposites. In this method the original problem is first nonlinearly transformed to a higher dimensional space. The kernel function simplifies the computational complexities by performing the dot product of the transformed data in the input space itself. The capability of the method to extract a large number of principal components is very useful for feature extraction and denoising. For the chaotic time series the kernel PCA successfully denoises and recovers the original data with substantial accuracy. Similarly for the polymer nanocomposite example the kernel PCA preprocessing followed by kernel regression is able to extract the dominant features and map the input output data very well. The fact that the method does not require solution of any hard nonlinear optimization problems makes the method very attractive for use in various process engineering applications.

A hybrid method using locally linear embedding (LLE) and SVDD was developed in chapter 4 and applied to the case studies of acetone-butanol fermentation and a benchmark SBR problem. The results show that LLE along with SVDD can be a very powerful tool for online process monitoring. As most of the industrial processes are nonlinear in nature, nonlinear dimensionality reduction using LLE can be very useful in reducing the features of the data, which in turn reduces the time for abnormality detection technique like SVDD.

A novel method for analysis and characterization of time series was proposed in chapter 5. This method is a unique combination of wavelet based singularity analysis and support vector machines classification. Proposed methodology was applied to a case study of flow regime identification in gas-

liquid stirred tank equipped with Rushton turbine. Employing our method we could classify flow regimes with 98% accuracy. Also from the regime map it is clear that the misclassified data-point belongs to the regime transition zone. This proves the effectiveness of this method for the identification of the flow regime in gas-liquid stirred tank. The excellent classification accuracy brings out the fact that the local scaling behavior of a given regime follows a distinct pattern. Further, the singularity measures can be employed by intelligent machine learning based algorithms like SVM for online regime identification. The method is simple and can be generalized to the other multiphase systems like bubble column, fluidized bed etc.

A new method for model selection in prediction of time series was proposed in chapter 6. Generally, the model parameters are selected for which the minimum RMS error is obtained on the predicted time series. This conventional criterion based only on RMS error may not take into account of the sharp changes or singularities in a time series and may fail in a case where this contribution of singularities is significant. Here, we have proposed an additional criterion based on error on the distribution of singularities in the predicted and actual time series. The distribution of singularities is evaluated through the local Hölder estimates and its probability density spectrum. Thus, the problem of model selection is solved by simultaneously minimizing both criteria, viz., RMS error based on the original time series as well as based on the error in the singularity distribution. The method is tested on three time series: two simulated and one based on real observations. Predictions of these time series have been done using kernel principal component regression (KPCR) and model parameters of KPCR have been selected employing the proposed as well as the conventional method. The problem now being a multiobjective optimization problem, we get a set of Pareto optimal solutions. Results obtained demonstrate that the proposed method helps in better prediction of the unseen test data and improves the generalization capability of the KPCR model. Model selection has produced the results which are better than the results yielded by the conventional method in all of the cases of the simulated and real time series. In general, we conclude that new method can be very useful in prediction of time series data having sharp singularities.

The results presented and discussed in chapters 2-6 disclose the fact that these kernel based machine learning tools viz. SVM, kernel PCA, SVDD are really promising for building data driven models of process engineering systems. The applications also reveal that these methods are really flexible and simple in their implementation and can be combined with other conventional tools to suit the necessities of the problem at hand.

List of Publications

Research papers published in International Journals

A.M. Jade, B.Srikanth, V.K. Jayaraman, B.D. Kulkarni, J.P. Jog and L. Priya, "Feature Extraction and Denoising Using Kernel PCA", *Chemical Engineering Science* 58, 2003, 4441-4448.

M. Agarwal, A. M. Jade, V. K. Jayaraman, B. D. Kulkarni, "Support Vector Machines: a Useful Tool for Process Engineering Applications", *Chemical Engineering Progress*, 98(1) 2003, 57-62

Rakesh Kumar, Avinash M. Jade, Valadi K. Jayaraman, and Bhaskar D. Kulkarni (2004) "A Hybrid Methodology For On-Line Process Monitoring," *International Journal of Chemical Reactor Engineering,* Vol. 2: A14.
http://www.bepress.com/ijcre/vol2/A14

A.M. Jade, V.K. Jayaraman, B.D. Kulkarni, A. R. Khopkar, V. V. Ranade, Ashutosh (2006). "A Novel Local Singularity Distribution Based Method for Flow Regime Identification: Gas-Liquid Stirred Vessel With Rushton Turbine" *Chemical Engineering Science* , 61, 688-697.

A. M. Jade, V.K. Jayaraman, B.D. Kulkarni, "Improved Time Series Prediction with a New Method for Selection of Model Parameters", .*Journal of Physics A*, 2006, 39, 2006, L483–L491.

Papers/posters presented in National Conferences

A.M. Jade, V.K. Jayaraman, B.D. Kulkarni, "Classification of Control Profiles using Support Vector Machines", paper presented in CHEMCON-2003, Bhubaneshwar, India.

A.M. Jade, V.K. Jayaraman, B.D. Kulkarni, "Nonlinear Modeling Of Chemical Engineering Systems using Kernel PCA", poster presented at annual Research Scholar Meet (RSM) at NCL, Pune 28th Feb, 2004.

A.M. Jade, V.K. Jayaraman, B.D. Kulkarni, " A New Method for Process Monitoring using Single Class SVM", poster presented at annual Research Scholar Meet (RSM) at NCL, Pune, 28th Feb, 2005