The Detection of Reliability Prediction Cues in Manufacturing Data from Statistically Controlled Processes

by

James Mosley

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2011 by the
Graduate Supervisory Committee:

Darryl Morrell, Chair
Andreas Spanias
Antonia Papandreou-Suppappola
Douglas Cochran
Chell Roberts

ARIZONA STATE UNIVERSITY

August 2011

ABSTRACT

Many products undergo several stages of testing ranging from tests on individual components to end-item tests. Additionally, these products may be further "tested" via customer or field use. The later failure of a delivered product may in some cases be due to circumstances that have no correlation with the product's inherent quality. However, at times, there may be cues in the upstream test data that, if detected, could serve to predict the likelihood of downstream failure or performance degradation induced by product use or environmental stresses. This study explores the use of downstream factory test data or product field reliability data to infer data mining or pattern recognition criteria onto manufacturing process or upstream test data by means of support vector machines (SVM) in order to provide reliability prediction models. In concert with a risk/benefit analysis, these models can be utilized to drive improvement of the product or, at least, via screening to improve the reliability of the product delivered to the customer. Such models can be used to aid in reliability risk assessment based on detectable correlations between the product test performance and the sources of supply, test stands, or other factors related to product manufacture. As an enhancement to the usefulness of the SVM or hyperplane classifier within this context, L-moments and the Western Electric Company (WECO) Rules are used to augment or replace the native process or test data used as inputs to the classifier.

As part of this research, a generalizable binary classification methodology was developed that can be used to design and implement predictors of end-item field failure or downstream product performance based on upstream test data that may be composed of single-parameter, time-series, or multivariate real-valued data. Additionally, the methodology provides input parameter weighting factors that have proved useful in failure analysis and root cause investigations as indicators of which of several upstream product parameters have the greater influence on the downstream failure outcomes.

DEDICATION

To my wife, Norma

ACKNOWLEDGEMENTS

Gratitude is expressed to Dr. Darryl Morrell for his advice, reviews, and direction throughout the development of this Dissertation. Thanks to Dr. Douglas Cochran for his thorough review of the Dissertation draft and key guidance offered. Also, appreciation is expressed to the other members of the supervisory committee for their review comments on the initial proposal and during the final oral defense of this research. Dr. Joseph Palais encouraged my entry into the Ph.D. program after review of my application for the M.S. Program and has on numerous occassions assisted my continuation within the program to completion. Academic Advisors Darleen Mandt and Esther Korner have both been extremely helpful to me over the years with respect to awareness of and adherence to University policies and procedures.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

In the testing or quality-control phase of a manufacturing process, data is collected and analyzed in order to ensure that the manufactured products meet some acceptance criteria. This data may include selected process data and product subcomponent data as well as product performance data. Often this data is used not only to grade the product (or service) but also as a means of identifying the control state of the manufacturing process.

In statistical process control, the statistics of one or more parameters are used to develop a set of control limits. For example, the average measured value ($\overline{X}$) of a parameter over a defined subgroup (or "lot") of assemblies might be tracked across an increasing set of such subgroups as a process statistic. Using historical data a sample mean of $\overline{X}$ ($\overline{\overline{X}}$) and a sample standard deviation (σ) are determined. In general, $\overline{X}$ is assumed to be normally distributed.[1] Upper and lower control limits are then typically determined as this sample mean $\pm 3\sigma$. The units which perform outside the control limits are considered deviations from the controlled process (outliers) or as indications that the process has gone out of statistical control. In either case, in an SPC system, a process alarm (or signal) is set. Of course, if the distribution of a test parameter is indeed Gaussian, the probability of false alarm is non-zero. The classical "3-sigma" control limits assume a normal distribution of the process parameter or variable. In situations where the process variable is not normally distributed, control limits (or control regions) may be set based on a chosen probability of false alarm (Type I error)

[1]Based on the Central Limit Theorem, this assumption is increasingly justified as the fixed number N of independent (or partially-correlated) elements included in each subgroup or lot is increased. Typically, for ease of implementation, the number N is fixed across lots, but for particular applications N may be variable if associated adjustments are made for the calculation of the standard deviation of $\overline{X}$ across multiple subgroups.

or risk of false rejection of alarm (Type II error).

Even for production units which perform within the control limits, certain additional limitations (such as the "Western Electric Rules") may be imposed to identify possible process or measurement abnormalities known as runs[1][2, p.25]. Runs consist of a series of successive readings whose low joint probability of occurrence can be used to signal a process problem. For example, Western Electric "Rule 4" indicates an alarm condition if fifteen consecutive points (readings) in a row all fall within a one-sigma region on one side of the mean.

Products whose performance is consistent with a controlled process, exhibit no abnormalities, and meet product specifications are deemed good. However, even good product may have latent defects or environmental susceptibilities that may impinge upon the product's life or reliability. Some of these reliability characteristics may be detectable but unidentified given the available process data. Other characteristics may have no corresponding cues contained in the implemented process data set. If reliability history is available, this history might be used in concert with historical process data to develop predictive criteria. Since some failures may be induced by environmental events apart from inherent product quality deviations, some means of specifying the confidence of a prediction must be provided. The question "could this failure have been predicted as a function of the process data?" might possibly have the answer "no." Given a limited population of returns due to failure, it might be possible to develop a predictor function (or machine) on the prior process data that would be consistent with respect to that population. However, if with high probability, the returned population could represent simply an unbiased random sampling of available fielded units, then the predictor machine may be overfit and not generalize well for other test samples. To empirically determine the generalization ability of the predictor, one would check the classification accuracy of the initial predictor operating on a representative population of both failed and survived units that were not included as

samples in the training or design of the predictor.[2]

During this research, a binary classification methodology was developed that can be used to design and implement predictors of end-item field failure/survival or downstream product test pass/fail performance based on upstream test data that may be composed of single-parameter, time-series, or multivariate real-valued data. Additionally, the methodology has proved useful as a forensic tool in failure analysis investigations as it provides indicators of which of several upstream product parameters have the greater influence on the downstream failure outcomes. While the data analysis or design portion of this generalizable methodology requires several input data processing and transformation steps, the implementation form (synthesis) of the prediction machine is relatively simple, only requiring taking the inner product of a derived weight vector with the upstream input data for a particular component or end-item, adding a derived offset, and then basing the classification decision on the sign of the result. Once designed for a specific dataset, the prediction machine can enable effective screening out of suspect components or end-items, especially in cases where the methodology has identified high correlation between one or more parameter elements of the upstream data and the downstream failure mode. As a interim output, the methodology also provides a normalized weight vector whose elements are weighting values that can serve to indicate which of the elements of a parameter input (or time-series) vector is of more key importance to the classification decision. This interim weight vector has proved useful as a forensic tool in determining likely contributing factors to low downstream test yields or failure modes. In real-world scenarios, the correlation between the downstream failure and cues in the upstream

[2]Depending on the specific fault mode and period of performance on which "survival" is defined, the current set of "survived" units may or may not contain potential future failures. For example, if survival is defined as not exhibiting a particular failure mode prior to some fixed number of years, then data (if available) from non-overlapping sets of failed units and survived units could be used to test the predictor. In other scenarios, the possibility of potential future failures among the current survivals should inform the interpretation of this classification accuracy testing.

manufacturing data may not, if they exist at all, be perfectly correlated to the failure mode under review. If the correlation is only partial, the predictor generated by this methodology also has a non-zero probability of either screening out units that would not fail or allowing units that would fail to escape the screen. Hence, in practice, there tends to be a trade-off between the detection rate (or, the ability to positively identify "bad" units) and the false positive rate (or, the fraction of "good" units falsely rejected). There are cases where the upstream test data would not be expected to provide true cues since the downstream failure mode is related to a latent defect that evidences no performance change in the affected product until the defect (such as, for example, a broken structural support or ruptured vapor barrier) actually occurs. In such cases, the prediction machine developed under this methodology would be expected to not perform well (as a predictor) on input test data not included in the design analysis even if the training data were to be classified with little or no error. In such cases, as will be demonstrated in one of the case studies explored in this dissertation, the resultant prediction may have a false positive rate rivaling or even exceeding the detection rate.

In this dissertation, we explore the use of downstream factory test data or product field reliability data to infer data mining or pattern recognition criteria onto manufacturing process or test data by means of support vector machines (SVM's) in order to provide reliability prediction models. In concert with a risk/benefit analysis, these models can be utilized to drive reliability improvement of the product or, at least, through screening to improve the reliability of the product delivered to the customer. Additionally, such models can be used to aid in reliability risk assessment based on detectable correlations between the product test performance and the sources of supply, test stands, or other factors related to product manufacture.

This work provides the following contributions:

- Algorithmic details of a modified SVM classifier that can be trained on labeled subsets of data from a statistically controlled process along with performance analysis of the classifier on several sets of actual manufacturing test data. The classifier so trained could then be used as a predictor function on the members of the overall dataset with respect to inclusion in the classes represented by the training data.

- The use of L-moment vectors and/or L-moment extensions to the input data vectors as means of increasing the discrimination power of the SVM upon the data streams from a statistically controlled process or upon multi-parameter vectors that may have correlation between elements.

- Algorithmic details and performance analysis of a modified SVM classifier that uses specific functions of order statistics of input vectors in order to embed discriminant information into the classifier equivalent to that required in the implementation of the classical 3-sigma process limits and Western Electric Rules.

The general classifier design methodology involves variations of the following top-level plan:

1. Begin with real-valued data from a statistically controlled process, with all data falling within some defined sigma level (say, 3 to 6 standard deviations from the process mean)

2. Ensure the data are organized into a set of vectors that each have the same number of elements.

3. On an element-by-element basis statistically normalize the data using the ensemble means and standard deviations calculated over the available dataset or subset of interest.

4. Extend or replace the input data vectors with elements representing problem-specific functions on or transformations of the input data vectors..

5. Depending on the specific dataset or problem, statistically normalize the extension or replacement element (recommended if the classifier weight vector is to be later utilized to determine the relative influence of the input elements).

6. Use a portion of the dataset to train the binary classifier. (This assumes that samples from both classes are available.)

7. Review the resultant weight vector to determine which input data or extended input data vector elements are the most significant.

8. If desired, reduce the number of vector elements (or, alternately, set those elements to zero in the weight vector) and retrain the classifier.

9. Use the classification parameters to implement (synthesize) a classifier or predictor specific to the dataset.

10. If desired, transform the classifier parameters so that the classifier can be used directly on the input data in its native form.

11. Test the classifier on new data or a portion of the original input data (statistically normalized, of course) not used in the design (analysis) or training of the predictor.

In practice, it may be necessary to iteratively improve the classifier by varying the training set in order to enhance the performance of the classifier over the test set (i.e. a set of data not included in the training itself).

As part of this research, a modified SVM implementation has been applied to real-world product test data from several statistically controlled processes in an aerospace manufacturing environment. In each of three case studies, SVM's were

trained using measurement and/or error data vectors from two labeled classes. The generalization ability of the resultant SVM's was explored by using the SVM's to classify end items using transformed versions of the actual test data. These experiments are detailed in Chapter 5. Each sample vector for these experiments consists of sets of elements representing the values of several different measurement parameters. In Chapter 8, a fourth case study using SVM's is explored in which the sample vectors consist of set of instantiations of the same measurement parameter.

Feature selection continues to be a viable area of research in the SVM field and is often dependent on the particular dataset and data usage under consideration. Along with completion of the intended contributions outlined above, research objectives accomplished as part of this effort include exploration of the application of the Structural Risk Minimization approach to normalization of feature-vectors, reduction of feature vector length, and effects of using varying numbers of training vectors for particular sets of measurement and measurement error data[3] derived from aerospace sensor manufacturing processes.

Following this introduction, Chapter 2 provides a review of the literature and background material in four areas: statistical process control, support vector machines, divergence estimation using minimum spanning trees, and order statistics (especially L-moments). These provide context for subsequent discussion about the application of support vector machines in the analysis of data from statistically controlled processes. Chapter 3 provides a detailed development of the hyperplane classifier and its use in a modified Support Vector Machine (SVM) which relaxes the requirement to necessarily locate the optimal hyperplane. Chapter 4 examines several data-dependent limitations of this modified SVM and how these can be mitigated.

[3]Due to the potentially confidential nature of the real-world data used for this research, affine transformations of the data are performed whenever the need arises to use specific data in providing application examples.

Three case studies using statistically normalized versions of real-world data are used in Chapter 5 to demonstrate the application of the SVM to statistically controlled datasets. Chapter 6 provides a brief exposition of kernel theory and its application to the hyperplane classifer. Chapter 7 introduces L-moment kernels and the application of L-moment kernels in SVM's. Chapter 8 describes methods of adding discriminant information equivalent to the Western Electric Company (WECO) to the SVM and provides a fourth case study utilizing both L-moments and WECO information in various SVM implementations. Chapter 9 provides a summary of results and observations along with suggestions for further research.

During this research, a binary classification methodology was developed that can be used to design and implement predictors of end-item field failure/survival or downstream product test pass/fail performance based on upstream test data that may be composed of single-parameter, time-series, or multivariate real-valued data. Additionally, the methodology has proved useful a forensic tool in failure analysis investigations as it provides indicators of which of several upstream product parameters have the greater influence on the downstream failure outcomes. While the data analysis or design portion of this generalizable methodology requires several input data processing and transformation steps, the implementation form (synthesis) of the prediction machine is relatively simple, only requiring taking the inner product of a derived weight vector with the upstream input data for a particular component or end-item, adding a derived offset, and then basing the classification decision on the sign of the result. Once designed for a specific dataset, the prediction machine can enable effective screening out of suspect components or end-items, especially in cases where the methodology has identified high correlation between one or more parameter elements of the upstream data and the downstream failure mode. As a interim output, the methodology also provides a normalized weight vector whose relative weights serve to indicate which of the input elements of a parameter input (or time-series)

vector is of more key importance to the classification decision. This interim weight vector has proved useful as a forensic tool in determining likely contributing factors to low downstream test yields or failure modes.

Chapter 2

BACKGROUND AND LITERATURE REVIEW

2.1 Statistical Process Control

We begin with a brief background study and literature review of the area of statistical process control as applied in the manufacturing arena. Mass production assembly line processes, as introduced by Henry Ford and others in the early 1900's, required that the form, fit, and function of assemblies (or subassemblies) made by different individuals or machines be identical within allowable tolerances. One means of monitoring the quality (i.e. uniformity) of manufacture is to inspect each individual assembly using some means of measurement to ensure that the product meets predetermined specifications. Products that fall outside of specification limits are rejected or reworked. The fallout or rejection rate may be used as indicators of the need for a process to be improved or corrected. However, this 100% inspection of the outcome of each subprocess may be both unnecessary and uneconomical. In the 1920's, H.F. Dodge and H.G Romig developed the use of statistical sampling as a means of reducing this inspection burden [3, p.10]. Human errors, measurement system errors, unobservable defects, and random process variation all pose limitations to the benefit of relying primarily on inspection as a means of quality control.

In about 1924, Walter A. Shewhart and others at Western Electric's Bell Telephone Laboratories began work on the application of statistics to the control of production processes. This work formed the basis of what is now known as Statistical Process Control (SPC). A process is said to be in a state of statistical control with respect to a particular quality variable when the variation of that variable can be approximately described by a fixed probability distribution [4, pp.30-31]. This quality variable may be a direct measurement variable, a derived variable (such as the mean or range of a subgroup), or a vector of variables. Shewhart introduced process control

charts ("Shewhart Charts") for utilization in tracking the control state of manufacturing processes [4, p.2][5, p.xiii]. If the control charts indicated that a process output exceeded control limits or had changed control states (such as a significant shift of the mean), some action might be taken to address the "special" cause of the variation or to stabilize the process. Other variations of process outputs within the control limits are considered to be "common" cause variations resulting from the operation of a stable process. The use of the quality control concepts described by Shewhart in his book *Economic Control of Quality of Manufactured Product* (1931) [6] grew in the U.S.A. until World War II, but declined thereafter. However, during the 1950's, W. Edwards Deming and J.M Juran successfully promoted the use of statistical process control in Japan. The success of the Total Quality Control (TQC) movement in Japan would later prove an important influence on the resurgence of interest in the use of statistical process control in the U.S. manufacturing sector.[1]

By the mid-1950's, it had been recognized that the Shewhart-type chart was insensitive to some process abnormalities (small "shifts") that may occur with no points falling outside of the process control limits [7]. As a result, in 1956, the Western Electric Company introduced five rules (known as the "Western Electric Rules") for guidance in determining alarm conditions from the classical control chart. The first of these rules is simply a restatement of the rule already used with the Shewhart chart, namely, to signal an alarm when a point lies beyond 3-sigma of the mean of the estimation parameter. The remaining rules indicate alarm conditions for some unlikely (i.e. low-probability) runs of successive points within the control limits. Other enhancements to the control chart have been developed to (1) detect smaller

[1]However, as noted in both [4, p.6] and [5, pp.xix-xxi], there have been historical differences between Japan and the U.S.A. in emphases and philosophical approaches to the use of statistical methods with respect to quality control. A good discussion of these issues can be found in [5, pp.1-6].

process changes than the Shewhart chart, (2) account for autocorrelated data, and (3) provide for multivariate detection of changes. Among these are the cumulative sum (CUSUM) charts [8, p.12][3, pp.127-135] and the Exponentially Weighted Moving Average (EWMA) charts [3, pp.135-136]. Multivariate control charts enable consideration of interactive effects among multiple process variables in establishing control limits or rules [8, p.12]. In the 1940's, Hotelling developed the T-squared (T^2) control charts for detection of shifts in a multivariate process [3, p.22]. Principal component analysis (PCA) has been applied as a means of transforming correlated variables into a set of uncorrelated variables upon which the traditional univariate control chart methods can then be applied [9, p.147].

2.2 Support Vector Machines

The systematic study of the problem of inferring statistical relations in data began in about the 1920's as extensions of the work of Fisher [10, p.2] for parametric approaches (i.e. parameter estimation based on maximum likelihood) and of the work of Glivenko, Cantelli, and Kolmogorov for general or non-parametric (inductive) methods [10, pp.2-3]. The development and utilization of parametric methods proceeded rapidly through the 1930's and into the 1960's. It was not until the expanded availability of computers to researchers in the 1950's and 60's (which enabled extensive analysis of inference models on "real-life" datasets) that some practical shortcomings of classical parametric statistical methods were formally revealed to the statistical research community [10, pp.2-7][11, pp.ix - x]. The classical methods, as framed at that time, demonstrated limited utility in cases where the real-world datasets

1. were multivariate (the so-called "curse of dimensionality"),
2. had densities that could not be approximated by classical closed-form, parametric density functions,

3. were weighted sums of two or more normal distributions, or

4. had low cardinality (i.e. small sample sizes).

Research into the extension of classical methods to address these issues did continue, but awareness of the aforementioned issues served to motivate parallel research into methods that could be used to infer or "learn" patterns (relations or structure) directly from the data (i.e. inductively) rather than predetermining a parametric structure and using the data to determine the best fit or discriminator using maximum likelihood or expectation maximization. Frank Rosenblatt is credited with the introduction of the first supervised learning machine[2], the Perceptron [12, pp.62-68][13, pp.11-19]. The Perceptron essentially extends the McCulloch-Pitts neuron model (introduced in 1943 by Warren McCulloch and Walter Pitts) [12, pp.62-63] by feeding back the comparison of the present neuron output with the correct output as a means of adjusting the values of the neuron's internal weighting factors that operate upon the input data. Constructed to address a two-class pattern recognition problem, the Perceptron was demonstrated to be able to determine a hyperplane that correctly segmented the training data into two classes if the training data are linearly separable. Under the assumption that the training data are a representative sampling of the two fixed-distribution classes, the ability of this hyperplane classifier to generalize (i.e. to correctly classify subsequent test data) is related to the margin of separation between the two classes of training data.

While the study of learning machines based on neural networks progressed, learning machines (including general adaptive filters) not necessarily based on neurobiological models also demonstrated the ability to learn patterns or generalize based on training data. A common general principle uniting various learning

[2]While, as Vapnik points out [11, p.1], Fisher had considered the separation of two sets of vectors using their set probability distributions, Fisher had not used data or "examples" directly to infer the classification relation of the two sets of vectors.

approaches is the strategy of empirical risk minimization (ERM) [10, p.7]. In this strategy one chooses, from a given set of decision rules or functions, the function that minimizes the risk of training error (empirical risk). In the 1960's, this induction principle from the statistical sciences was applied to the pattern recognition problem using indicator functions (i.e. functions whose range is the discrete set $\{0, 1\}$). By the end of the 1970's, ERM theory was expanded to include real-valued functions in solution of regression and density estimation problems [10, p.8] For any set of indicator functions with finite VC dimension[3], the ERM induction process is a consistent method — that is, it converges in probability to a solution with minimum expected risk among the candidate functions as the number of training samples (or observations) increases[4]. However, if the set of functions is chosen such that for any possible finite set of training vectors and classifications assignments, training will be error-free, then generalization may not be possible due to overfitting. Stated another way, there exists an inherent trade-off between the classification power or capacity of a learning machine (family of functions) and its ability to generalize from the training data to new test samples.

Capacity (or VC dimension) control is a key feature of the statistical learning theory from which support vector machines were eventually developed. Based on bounds for the non-asymptotic (i.e. limited sample set) rate of convergence of the ERM learning principle and related bounds on the probability of test error of a learning machine, an induction approach known as "Structural Risk Minimization" (SRM) was developed [10, pp.55-57][10, p.10]. Given a nested structure of admissible

[3]The VC (Vapnik-Chervonenkis) dimension for this case is defined by Vapnik as the greatest number h of data vectors that can be "separated into two different classes in all 2^h possible ways using this set of functions (i.e. the VC dimension is the maximum number of vectors that can be *shattered by the set of functions*)." [10, p.147] Thus VC dimension is a measure of the binary classification capacity of a the learning machine (set of functions).

[4]See proofs in [10, pp.121-137].

machines[5] and a predetermined confidence interval, the SRM induction principle recommends selection of the machine for which minimizing the training error (empirical or sample-based risk) yields the lowest bound on the probability of test error (actual or global risk) [11, pp.93-96]. This bound is related to the VC dimension, the number of training errors, and the number of training samples [14, pp.123-124]. It should be noted that the error convergence "bounds" on the learning machines that we have been discussing are not absolute bounds in the sense that, as the number of training samples increases, the generalization or expected test error cannot exceed some given $\delta > 0$, but rather that with probability $1 - \eta$, where $0 < \eta < 1$, the learning error will not exceed that δ. For this reason, this statistical approach or learning model is generally known in the computer science community as the "Probably Approximately Correct" (or *pac*) model [13, pp.52-54].

Application of SRM to high dimensional linear learning problems proved to have accuracy and generalization results that rivaled those of neural networks including multilayer perceptrons. In combination with the use of kernels which can be used to map non-linear inputs into linear feature spaces, learning algorithms using the learning bias suggested by the SRM approach and well known Lagrange multiplier optimisation and dual theory lead to the development in the early 1990's of what is now known as Support Vector Machines (SVM's) [13, p.7]. Support vector machines use the Gram matrix relationships between functions of the training input vectors to train a learning machine that, in many cases, turns out finally to be a function of only a subset of the input vectors. These vectors are therefore called the support vectors since the training machine that is to be used to test new inputs is independent of the other (non-supporting) input vectors.

[5] A machine (set of indexed functions, $Q_a(x), a \in \Lambda$, where Λ is the set of indices) is admissible if it has finite VC dimension and the set is totally bounded or, at least, $||Q_a||_N / ||Q_a||_1$ is bounded for all $a \in \Lambda$ for some integer $N > 2$ [11, pp.94-95].

Since support vector machines (or kernel learning theory) draw upon several research disciplines, their conceptual roots encompass a multi-threaded and extensive background. Some additional background details can be accessed from [10, pp.1-15], [11, pp.7-15], [13, p.8], [15, pp.xvii - xix], and [16, pp.1-2]. Vapnik also provides a summary timeline of developments from the Perceptron to SVM's over the period 1958 to 1995 [10, pp.301-302].

Several authors, including Vapnik [10, pp.156-163], have noted the similarity of the SVM algorithm to approaches based on kernel logistic regression (KLR) which use regularized functions in reproducing kernel Hilbert spaces (RKHS). Approaches such as the import vector machine (IVM), which selects a submodel approximation of a fitted KLR model, have demonstrated performance similar to that of support vector machines even though the loss function differs from that used for the SVM [17, p.201]. The success of SVM-like machines in avoiding overfitting has been partly attributed to the fact that the regularization terms included in the model definitions act to control the complexity (or capacity) of the model space even in cases where the associated loss functions are not margin-maximizing loss functions [17, pp.200-201]. Extensions of SVM's to the problem of learning hidden information are discussed in some more recent material by Vapnik [18, pp.438-446].

2.3 Divergence Estimation Using Minimum Spanning Trees

In order to assess the performance of an SVM-classifier, given a particular set of sample data (embedded in $\mathbb{R}^n$) from two classes, it would be helpful to have a prior estimate of the separability or divergence of the two class distributions. Given representative training vectors from each distribution, the Henze-Penrose divergence estimate [19] ($\widehat{HP}$) ranges between 0 and 1. Let M represent the total number of points (or vectors) in the training set which contains labeled members from each class. Let m_0 and m_1 represent the number of samples in each of the two respective classes, so

that $M = m_0 + m_1$. Then as $m_0, m_1 \to \infty$, $\widehat{HP} \to HP$, the Henze-Penrose divergence between the two distributions [19]. $\widehat{HP}$ approaches 0 as M approaches infinity for a structured intermixing of the two class samples in which each class member is closer to a member of the opposite class than to any member of its own class. Conversely, $\widehat{HP}$ approaches 1 as M approaches infinity if each class member is closer to a member of its own class than to any member of the opposite class. If the two sets of training vectors have approximately the same spatial distribution (with arbitrary intermixing), then $\widehat{HP}$ tends toward 0.5 as M approaches infinity if the sets have the same number of members. If these sets do not necessarily have the same number of members, but the ratio $m_0 : m_1$ is fixed, then as M approaches infinity, $\widehat{HP}$ tends toward the quantity

$$\left(1 + \frac{\min(m_0, m_1)}{\max(m_0, m_1)}\right)^{-1}.$$

$\widehat{HP}$ may be determined through the use of a minimum spanning tree (MST) graph, embedded in $\mathbb{R}^n$, connecting the points (or vectors) of the training set [19]. Construction of the MST may be initiated by selecting any of the training points as the first member of the spanning tree. Next, find the sample point not in the tree that, among all non-tree sample points, is the shortest distance away from any points within the tree. This non-tree sample point is then connected with a new edge to the point in the tree to which it is the closest. This new point is then considered a point in the tree and the process of augmenting the tree with new edges and non-tree points continues until all points are in the tree. The Friedman-Rasky test statistic on this MST is the number N of edges in the tree that connect two points in opposite classes. Given N,

$$\widehat{HP} = 1 - N/M.$$

2.4 Order Statistics and L-moments

Linear (or "L") moments derive their name from the fact that they can be estimated using linear combinations of the expectations of order statistics. A unified approach to

univariate descriptive statistics based on L-moments was described by Hosking in 1990 [20]. This approach, built primarily upon several authors' contributions dating as far back as 1912 [20, p.106], provides an alternative to classical moment-based approaches for the estimation of parametric distributions from data samples. It has been shown to be able to provide more robust and accurate sample-based estimations of underlying data distributions when only a limited number of samples is available. L-moments are now widely used in the hydrology and flood-management fields [21, p.194][22, pp.18-41] where the reliability of inferences (i.e. predictions) based on classical methods in many cases was limited due to the availability of only a small number of samples and the inclusion of outlier data [23, p.6].

The following definitions are adapted from Hosking[6] [20, pp.106-107][21, pp.193-194]. Suppose $\{X_i\}_{i=1}^{n} \rightarrow \{X_{k:n} : X_{j:n} \leq X_{k:n} \text{ whenever } j < k\}$ is a real-valued set of n iid samples of a random variable X, which has a cumulative distribution function (CDF) F and an associated inverse CDF or quantile[7] function Q_F:

$$\begin{gathered} F : \mathbb{R} \rightarrow [0,1] \\ x \mapsto p := F(x), \text{ where } F(x) \equiv \Pr(X \leq x) \end{gathered} \tag{2.1}$$

$$\begin{gathered} Q_F : [0,1] \rightarrow \mathbb{R} \\ p \mapsto x := Q_F(p), \text{ where } Q_F(p) \equiv \inf\{x : F(x) \geq p\} \end{gathered} \tag{2.2}$$

[6]Some changes in notation have been made for the sake of heuristic clarity. For example, in his 1990 article [20], Hosking uses F to denote, at different times, both the cumulative distribution function (CDF) of the random variable X and the input to the inverse CDF. Likewise, x is used to denote both the inverse CDF itself and the input to the CDF. Hence, $F(x) \equiv \Pr(X \leq x)$ and $x(F) \equiv \inf\{x : F(x) \geq F\}$. While the expression $x(F)$ for the inverse CDF is technically correct and notationally convenient, we have chosen instead to express the inverse CDF (i.e. $x(y) \equiv F^{-1}(y)$) as $Q_F(y)$, where $y \in [0,1]$.

[7]The p-quantile is the infimum of the set $\{x : F(x) = p\}$ [23, p.14][24, p.5][25]. That is, with probability p, the random variable X does not exceed the p-quantile value.

Let the samples be sorted in ascending order and relabeled as $X_{k:n,}, (k \in [1,...,n])$,such that $X_{j:n} \leq X_{k:n}$whenever $j < k$. Then the sample $X_{k:n}$is called the k^{th} order statistic. The expected value of the k^{th} order statistic given a sample size of n can be expressed as [20, p.106][24, p.34]:

$$EX_{k:n} = \frac{n!}{(k-1)!\,(n-k)!} \int_{-\infty}^{\infty} x\{F(x)\}^{k-1}\{1-F(x)\}^{n-k} \mathrm{d}F(x) \tag{2.3}$$

or, in terms of the quantile function

$$EX_{k:n} = \frac{n!}{(k-1)!\,(n-k)!} \int_{0}^{1} Q_F(p)\, p^{k-1}\,(1-p)^{n-k}\,\mathrm{d}p. \tag{2.4}$$

The r^{th} L-moment, λ_r, is defined as [20, p.106]:

$$\lambda_r \equiv \frac{1}{r}\sum_{k=0}^{r-1}(-1)^k \begin{pmatrix} r-1 \\ k \end{pmatrix} EX_{r-k:r}, \quad r = 1,2,... \tag{2.5}$$

where $\begin{pmatrix} n \\ k \end{pmatrix} \equiv \frac{n!}{k!\,(n-k)!}$ and $0! \equiv 1$. By using equation 2.4 to expand the right side of equation 2.5 and combining polynomial terms of the same order, λ_r can be expressed in terms of the product of $Q_F(p)$and polynomials in p integrated on the interval [0,1] [20, pp.106-107]:

$$\begin{aligned}
\lambda_r &= \frac{1}{r}\int_0^1 Q_F(p) \left(\sum_{k=0}^{r-1} (-1)^k \frac{(r-1)!\,r!}{(k!\,(r-k-1)!)^2}\, p^{r-k-1}(1-p)^k \right) \mathrm{d}p \\
&= \int_0^1 Q_F(p) \left(\sum_{k=0}^{r-1} \left[\begin{pmatrix} r-1 \\ k \end{pmatrix} \right]^2 p^{r-k-1}(p-1)^k \right) \mathrm{d}p \\
&= \int_0^1 Q_F(p) \left(\sum_{k=0}^{r-1} (-1)^{r-k-1} \begin{pmatrix} r-1 \\ k \end{pmatrix} \begin{pmatrix} r+k-1 \\ k \end{pmatrix} p^k \right) \mathrm{d}p
\end{aligned} \tag{2.6}$$

Hence, the first five L-moments are[8]:

$$\begin{aligned}\lambda_1 &= E(X_{1:1}) \\ &= \int_0^1 Q_F(p)\,\mathrm{d}p\end{aligned}$$

$$\begin{aligned}\lambda_2 &= \tfrac{1}{2}E(X_{2:2}-X_{1:2}) \\ &= \int_0^1 Q_F(p)\,(2p-1)\,\mathrm{d}p\end{aligned}$$

$$\begin{aligned}\lambda_3 &= \tfrac{1}{3}E(X_{3:3}-2X_{2:3}+X_{1:3}) \\ &= \int_0^1 Q_F(p)\,\left(6p^2-6p+1\right)\mathrm{d}p\end{aligned}$$

$$\begin{aligned}\lambda_4 &= \tfrac{1}{4}E(X_{4:4}-3X_{3:4}+3X_{2:4}-X_{1:4}) \\ &= \int_0^1 Q_F(p)\,\left(20p^3-30p^2+12p-1\right)\mathrm{d}p\end{aligned}$$

$$\begin{aligned}\lambda_5 &= \tfrac{1}{5}E(X_{5:5}-4X_{4:5}+6X_{3:5}-4X_{2:5}+X_{1:5}) \\ &= \int_0^1 Q_F(p)\,\left(630p^4-560p^3+210p^2-30p+1\right)\mathrm{d}p\end{aligned}$$

Other quantities defined by Hosking are the *L-coefficient of variation* (L-CV), $\tau \equiv \lambda_2/\lambda_1$ and the *L-moment ratios* τ_r (where $\tau_r \equiv \lambda_r/\lambda_2, \quad r \geq 3$).

Some key properties of L-moments and L-moment ratios are

1. The existence of the L-moments of a real-valued random variable X requires as its only condition that the mean of X be finite [20, p.107].

2. Any distribution with a finite mean is determined by its L-moments [21, pp.194-195].

3. If X is non-degenerate and is non-negative almost surely, then the absolute values of its L-moment ratios have an upper bound of one. In particular [20,

[8]The first four of these were also expanded (with slightly different notation) in [20, p.107][23, p.22].

p.108],

$$0 < \tau < 1\,,$$

and

$$|\tau_r| < 1, \quad r \geq 3\,.$$

4. L-moment estimators can be used as to approximate a wider range of distributions than conventional moment estimation methods, typically converging to their asymptotic distribution more rapidly than classical parameter estimators when sample sizes are small [20, p.105][9].

5. Estimations of L-moments and L-moment ratios can be used as summary statistics for a data stream or data vector and provide a simultaneous means of "detecting" the underlying parametic distribution of the data if such a distribution exists. λ_1 is simply the distribution mean. $\lambda_2, \tau_3, \tau_4$, and τ_5 are somewhat related to the conventional measures of variance, skew, kurtosis, and bimodal tendency, respectively [20, pp.109-111][10].

L-moments can be expressed as linear combinations of probability weighted moments (PWM's) and vice versa [23, p.xii][20, p.108]. However, PWM's, which were introduced by Greenwood and others in the late 1970's [26], are not as easily interpretable (compared to L-moments) as measures of the variance, skew, kurtosis, etc. of probability distributions [20, p.109].

In many practical applications, the underlying distribution of X is unknown. For such cases, L-moments must be estimated from samples by averaging sample order statistics over subgroups of available data. These estimates of L-moments (and

[9] Hosking states that "L-moments sometimes yield more efficient parameter estimates than the maximum likelihood estimates" [20, p.105].

[10] $\lambda_1, \lambda_2, \tau_3$, and τ_4 are also known as the *L-location, L-scale, L-skewness, and L-kurtosis,* respectively [23, p.24][20, p.110].

L-statistics) based on sample data are known as *U-statistics*, which have been used in non-parametric (i.e. sample-based) statistical settings [20, p.113-114].

While, in theory, the characterization of some distributions by L-moments may require an infinite set of L-moments to fully describe the distributions, many distributions can be well-described or at least well-approximated by the first few lower order L-moments [20, p.110]. Given several instantiations of a random vector stream of size n generated from the same unknown distribution, one could conceivably control the generalization ability (i.e. avoid overfitting) of an L-moment estimation machine by limiting the order of L-moment terms included in the L-moment estimator. In this light, a finite-order L-moment estimator may be viewed as a limited capacity machine (set of functions) that transforms random data vectors into a multidimensional linear feature space. Hence, we propose to explore the use of L-moments in association with SVM's as a means of separating data vectors sourced from generators with different distributions or as a means of detecting a subdistribution within mixed-source data.

Chapter 3

HYPERPLANE CLASSIFIERS AND THE SUPPORT VECTOR MACHINE

3.1 Binary Classification Using Hyperplanes

Let Z be a nonempty set of inputs, parameter vectors, patterns, or objects and Y be a non-empty, countable set of outputs or class labels.[1] When presented with $z \in Z$, a decision agent or process selects $y \in Y$ in accordance with some fixed conditional probability distribution $P(y|z)$. Suppose the set $\{P(y|z) : y \in Y, z \in Z\}$ are unknown (or hidden), but a representative training set of m input/output observations $(z,y) \in Z \times Y$ is available. We hypothesize that the agent or process can be modeled as a capacity-limited set of functions $\{f_\lambda : Z \to Y\}$ and a probability distribution $P(\lambda)$ such that $P(y|z) = \sum_{\lambda \in \Lambda} P(\lambda) I(y, f_\lambda(z))$, where I is a comparator function that returns "1" if its two inputs are equal and "0" otherwise. Λ is a set of indices or parameter values that are used to identify particular members of this set of functions relating Z and Y. Under this framework, the classification learning problem is to infer (from the training set) estimates of $f \in \{f_\lambda\}$ or $P(y|z)$ that can be used to predict the class $y_t \in Y$ the decision agent will choose, given test input $z_t \in Z$. However, if z_t has no corresponding observation (z_t, y) included in the training set, we are at a loss to predict y_t unless some measure of similarity exists or can be imposed on the elements (z,y) [27, p.2] .

One way to impose (or redefine) a similarity measure on Z is to map (or embed) the inputs or patterns z into a feature space $\mathscr{H}$ (where $\mathscr{H}$ is a separable real Hilbert space and therefore has a countable orthonormal basis [28, p.168]) by a map [27, p.3]:

[1] Here we are restricting the discussion to the classification problem. For the general pattern recognition problem, which includes probability density estimation and regression analysis, Y need not be restricted to a countable set.

$$\Phi : Z \;\; \rightarrow \;\; \mathscr{H}$$

$$z \mapsto h := \Phi(z).$$

A real-valued similarity (or dissimilarity) measure k may be defined on Z by:

$$k(z,z') = \left\langle \Phi(z), \Phi(z') \right\rangle$$

where $\langle \cdot, \cdot \rangle$ is the inner product in $\mathscr{H}$. The similarity measure for Y in this classification learning context can be taken to be an indicator or comparator function:

$$I : Y \times Y \rightarrow \{0,1\}$$

$$y \times y' \mapsto r := I(y,y') \, .$$

For the binary classification problem, Y has only two elements or labels. Suppose $Y = \{-1,1\}$ and that X is the $n-$dimensional Hilbert space $\mathbb{R}^n$. Further assume that each $x \in X$ maps to only one class $y \in Y$ (i.e. $P(y|x) \in \{0,1\}$) and that the m-element training set $T_m = \{(x_1,y_1),...,(x_m,y_m)\}$ contains labeled members of both classes.[2] For this type of feature space one of the most elementary classifiers is the hyperplane classifier [27, pp.11-15]. For the binary hyperplane classifier, the decided (or estimated) classification for any test input vector x is determined by the side of a chosen hyperplane on which that input vector lies. For any fixed $w \in X$ and $b \in \mathbb{R}$, an

[2]This corresponds to "noiseless" observations. For noisy observations, in which $0 < P(y = -1|x) < 1$, some learning machine implementations remain robust in that they tend to weight conflicting class label assignments for a given input value x in keeping with their frequency of occurrence in the training set.

associated hyperplane in X is the set

$$h_{w,b} = \{x : \langle w,x\rangle + b = 0\}, \tag{3.1}$$

where w is, by construction, a normal vector of the hyperplane and $\frac{||b||}{||w||}$ is the distance of the hyperplane from the origin (zero vector) in X. Given a hyperplane $h_{w,b}$, the classification of a test input vector x_t is then determined by the decision function:

$$y_t = f(x_t) = \text{sgn}(\langle w,x_t\rangle + b)\,, \tag{3.2}$$

where

$$\text{sgn}(r) \equiv \begin{cases} 1, & \text{if } r \geq 0 \\ -1, & \text{if } r < 0 \end{cases}, \quad r \in \mathbb{R}\,.$$

Let $\mathscr{A} \subset X \times \mathbb{R}$ be the set of all possible hyperplane parameters $\alpha \doteq (w,b)$ for hyperplanes in X. Then $\{f_\alpha\}$, with f as defined in equation 3.2, is a set of decision functions each based on an associated hyperplane $h_\alpha \subset X$. Since $X \subset \mathbb{R}^{\text{n}}$, the decision function set $\{f_\alpha\}$ has a VC dimension $d = n+1$ [14, p.125]. In training the binary classifier or learning machine, the hyperplane parameters (w,b) are chosen to minimize the empirical risk (or average training error rate [27, p.8][14, p.123]):

$$R_{\text{emp}}\,[f_\alpha] = \frac{1}{m}\sum_{i=1}^{m} 0.5\,|y_i - f_\alpha(x_i)|\,. \tag{3.3}$$

We note that for the training set T_m, the subset of $\{f_\alpha\}$ that minimizes the empirical risk is non-singular. That is, for the Hilbert feature space and a separable training set consisting of a finite number of entries, there are always multiple members of $\{f_\alpha\}$ that result in the minimal value of empirical risk. Given any two parallel hyperplanes h_{w,b_1} and h_{w,b_2} (with $b_1 < b_2$) whose associated decision functions result in the same binary classifications of the training inputs, any member of the compact set of hyperplanes $\{h_{w,b} : b_1 \leq b \leq b_2\}$ also has an associated decision function f_α, where $\alpha = (w,b)$, that results in the same classifications. Indeed, if the number of training

inputs is finite, the solution of equation 3.3 for all decision functions f_α that minimize empirical risk, results in selection of a subset of $\{f_\alpha\}$ in which the members of the subset are associated with a family of non-singular compact sets of hyperplanes in X. If the training data are linearly separable, this set of decision functions can be chosen such that $R_{\text{emp}}[f_\alpha] = 0$.

3.2 The Optimal Hyperplane

As discussed in section 2.2, when the training set is linearly separable, the capacity of the hyperplane classifier decreases (and generalization improves) with increasing margin. The margin of a separating hyperplane is defined as the minimum distance between the separating hyperplane and the nearest training vector. Among the hyperplanes that correctly separate the training data, there is an optimal one that provides the maximum margin [11, pp.131-132][29, p.6]:

$$\max_{w,b} \min_{i} \{||x - x_i|| : x \in X, \langle w,x\rangle + b = 0, (w,b) \in \mathscr{A}, i = 1,...,m\} \tag{3.4}$$

As in the previous section, let $h_{w,b} = \{x : \langle w,x\rangle + b = 0, (w,b) \in \mathscr{A}, ||w|| > 0\}$ represent a hyperplane selected from $H \subset X$. Based on equation 3.2 and the density of H in X, correct classification of a linearly separable training set $T_m = \{(x_1,y_1),...,(x_m,y_m)\}$ by a hyperplane $h_{w,b}$, where $\{x_i\}_{i=1}^m \cap h_{w,b} = \emptyset$, implies that, for some $\delta > 0$,

$$y_i(\langle w,x_i\rangle + b) \geq \delta, \quad \forall i = 1,...,m \tag{3.5}$$

If, for at least one of the training vectors, equality is attained in this equation, then δ is known as the *functional margin* of the training inputs (x_i,y_i) with respect to $h_{w,b}$ [13, p.11]. Note that if $X = \mathbb{R}^n$, then given any real number $k > 0$, $h_{w,b}$ and $h_{kw,kb}$ are equivalent designations of the same oriented hyperplane:

$$\begin{aligned} h_{w,b} &= \{x : \langle w,x\rangle + b = 0\} \\ &= \{x : \langle kw,x\rangle + kb = 0\} \\ &= h_{kw,kb}\,. \end{aligned} \tag{3.6}$$

Now let the feature space $X = \mathbb{R}^n$. Given the separable training set T_m, we can then set $\delta = 1$ in equation 3.5 and find w and b such that [13, p.95][27, p.196][14, p.129]:

$$y_i\left(\langle w,x_i\rangle + b\right) \geq 1, \quad \forall i = 1,...,m \tag{3.7}$$

The *geometric margin* of (x_i, y_i) is the Euclidean distance between x_i and the hyperplane $h_{w,b}$ [13, p.12]. Its value can be deduced by normalizing the normal vector w in equation 3.7 (for the case of equality) to obtain:

$$y_i\left(\left\langle \frac{w}{||w||}, x_i \right\rangle + \frac{b}{||w||}\right) = \frac{1}{||w||}$$

To find the optimal hyperplane (equation 3.4), one minimizes $||w||$ or $||w||^2$ (i.e. maximizes the margin $\frac{1}{||w||}$) subject to the condition of equation 3.7. This constrained optimization problem can be solved using the method of Lagrange multipliers [27, pp.13-15][13, pp.94-100][29, pp.6-8]. Define the *objective function* $\phi(w)$ as

$$\phi(w) \equiv \frac{1}{2}||w||^2 = \frac{1}{2}\langle w,w\rangle \tag{3.8}$$

This function is to be minimized with respect to w subject to the constraints

$$y_i\left(\langle w,x_i\rangle + b\right) \geq 1, \quad \forall i = 1,...,m$$

Expressed in the conventional form of optimization inequality constraints [30, p.4][13, p.80], the constraints are

$$-\left[y_i\left(\langle w,x_i\rangle + b\right) - 1\right] \leq 0, \quad \forall i = 1,...,m$$

Using non-negative real *dual* variables a_i (where at least two of the variables are non-zero), the primal Lagrangian may then be expressed as [14, p.130][29, p.6]:

$$L_P(w,b,A) \equiv \frac{1}{2}||w||^2 - \sum_{i=1}^{m} a_i \left[y_i \left(\langle w, x_i \rangle + b \right) - 1 \right] \tag{3.9}$$

where where A is a vector or set of the coefficients a_i for $i = 1, \ldots, m$.

At a constrained minimum value of the objective function $\phi(w)$, we require by classical Lagrangian optimization theory that the partial derivatives of L_p with respect to the primal variables w and b be zero:

$$\frac{\partial L_P}{\partial w} = w - \sum_{i=1}^{m} a_i y_i x_i = 0$$

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^{m} a_i y_i = 0$$

This results in the relations [29, p.7][13, p.95]:

$$w = \sum_{i=1}^{m} a_i y_i x_i \tag{3.10}$$

$$\sum_{i=1}^{m} a_i y_i = 0 \tag{3.11}$$

Substituting these relations into 3.9 yields the following dual representation in terms of the coefficients a_i [13, p.96]:

$$\begin{aligned} W(A) &= \tfrac{1}{2} \left\langle \textstyle\sum_{i=1}^{m} a_i y_i x_i \, , \sum_{j=1}^{m} a_j y_j x_j \right\rangle - \sum_{i=1}^{m} a_i \left[y_i \left(\left\langle \textstyle\sum_{j=1}^{m} a_j y_j x_j \, , x_i \right\rangle \right) \right] \\ &\quad -b \textstyle\sum_{i=1}^{m} a_i y_i + \sum_{i=1}^{m} a_i \\ &= -\tfrac{1}{2} \textstyle\sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j y_i y_j \left\langle x_i, x_j \right\rangle + \sum_{i=1}^{m} a_i \end{aligned} \tag{3.12}$$

The *Wolfe dual* optimization problem [29, p.8] for this representation is to maximize $W(A)$ with respect to the dual variables a_i subject to the constraints

$$\sum_{i=1}^{m} a_i y_i = 0 \tag{3.13}$$

$$a_i \geq 0, \quad i = 1, \ldots, m \tag{3.14}$$

We note that the primal objective function ϕ is convex since for $\lambda \in [0,1]$ and $w_{1,}w_2 \in X$, we have, by the triangle inequality (recalling that $\phi(w) \equiv \frac{1}{2}||w||^2$),

$$\begin{aligned} \phi(\lambda w_1 + (1-\lambda) w_2) &\leq \lambda^2 \phi(w_1) + (1-\lambda)^2 \phi(w_2) \\ &\leq \lambda \phi(w_1) + (1-\lambda) \phi(w_2). \end{aligned}$$

We also note that the primal constraints are affine functions of w. Under these conditions, given that X is convex, the *Strong duality theorem* [13, p.86] implies that the optimal solution value of the triplet[3] (w,b,A) is optimal for both the primal and dual optimization problems (i.e. there is no *duality gap*). That is, there exists an optimal set or Lagrangian function *saddle point* $(w^{opt}, b^{opt}, A^{opt})$ such that for $w^{opt} \in X$; $b^{opt} \in \mathbb{R}$; and A^{opt} $\left(\text{where } 0 \leq a_i^{opt} \in A^{opt}\right)$:

$$L_p\left(w^{opt}, b^{opt}, A\right) \leq L_p\left(w^{opt}, b^{opt}, A^{opt}\right) \leq L_p\left(w, b, A^{opt}\right) \tag{3.15}$$

for all $w \in X$; $b \in \mathbb{R}$; and all sets A (with elements $a_i \geq 0, \quad i = 1,...,m$). Furthermore, the *Karush-Kuhn-Tucker (KKT) conditions* of optimization theory [30, pp.95-96][13, p.87] imply the existence at this optimal solution value of a set of non-negative dual variables or coefficients a_i^{opt} such that the following *KKT complementarity condition* holds:

$$a_i^{opt}\left[y_i\left(\left\langle w^{opt}, x_i\right\rangle + b^{opt}\right) - 1\right] = 0, \quad \forall i = 1,...,m. \tag{3.16}$$

For such a set of coefficients, it can be deduced from equation 3.16 that only input vectors x_i for which the inequality constraints are met with equality (i.e. where $y_i(\langle w, x_i\rangle + b) - 1 = 0$) can have corresponding non-zero coefficients a_i^{opt}[4]. Under the

[3]Note that this is not identical to the triplet described in [13, p.86]. There, β represents a vector of equality constraint coefficients and **w** represents a solution vector in the domain of the primal objective function. In our case, the primal variables consist of both the objective function domain variable w and the constraint condition offset variable b.

[4]In practice, due to numerical limitations of computers, equation 3.16 may be modified as $a_i^{opt}[y_i(\langle w^{opt}, x_i\rangle + b^{opt}) - 1] \leq \varepsilon, \quad \forall i = 1,...,m$ where $\varepsilon > 0$. The value of ε is chosen to be sufficiently large to compensate for limited numerical precision and round-off error [31, p.8].

KKT complementarity condition, the remaining coefficients must be zero and their corresponding input vectors therefore would not contribute to a constrained optimal solution of w via the relation $w^{opt} = \sum_{i=1}^{m} a_i^{opt} y_i x_i$. Of course, this does not in general negate the possibility that there exists a set A, not meeting the KKT complementarity condition, such that $w^{opt} = \sum_{i=1}^{m} a_i y_i x_i$[5]. However, if an optimal normal vector (or point) w exists, then the KKT conditions guarantee the existence of a set A^{opt}, associated with this optimal vector, for which equation 3.16 does hold. Thus to solve the primal optimization problem we can find the set of coefficients A that solve the Wolfe dual optimization problem, use equation 3.10 to solve for an optimal normal vector w, and then apply the KKT complementarity condition to find a corresponding coefficient set A^{opt}.

Introducing the Lagrangian multipliers γ and η_i for the constraints (equations 3.13 and 3.14) associated with $W(A)$, we obtain for the Wolfe dual optimization problem the dual Lagrangian L_D, where[6]

$$L_D(A) \equiv -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} a_i a_j y_i y_j \langle x_i, x_j \rangle + \sum_{i=1}^{m} a_i - \gamma \sum_{i=1}^{m} a_i y_i + \sum_{i=1}^{m} \eta_i a_i \tag{3.17}$$

Applying the KKT complementarity condition to the inequality constraint 3.14 implies that

$$\eta_i a_i = 0, \quad \forall\, i = 1, ..., m$$

Hence, η_i is zero unless $a_i = 0$. But if a_i is zero, then the associated input x_i is not a support vector and can be omitted from the current training set. Note that if η_i is chosen to be to zero for all i, then L_D does not impose the inequality conditions

[5]The existence of a set of non-negative coefficients a_i associated with the constrained optimal solution of w does not necessarily negate the possibility of the existence of another non-KKT set of coefficients a_i that can also be associated with this optimal solution. In such a scenario, some of the *non-support vectors* might be included with non-zero coefficients in equation 3.10.

[6]Our use here of the label L_D differs from Burges [14, p.130] who uses L_D to represent the quantity that we have labeled $W(A)$ in close similarity with the nomenclature of Scholkopf [29, p.8] and Cristianini [13, p.96].

$(a_i \geq 0)$ that are associated with $W(A)$, so the maximization of L_D can result in values of $a_i < 0$. In this case, the non-negativity constraint on a_i can be enforced by iteratively maximizing L_D, setting to zero the a_i that is most negative, and remaximizing L_D without the associated input vector x_i. The iteration stops when the obtained solution contains only non-negative values of a_i. At this point, we obtain w by application of equation 3.10. The solution set (w,b,A) thus obtained can then be checked against the KKT complementarity condition for the primal Lagrangian L_P (equation 3.16) and modified as necessary by setting to zero the violating a_i (if any) associated with the largest magnitude for the quantity

$$y_i\left(\langle w,x_i\rangle + b\right)$$

L_D is then remaximized and the primal KKT complementarity condition (equation 3.16) is rechecked. This iteration stops when the primal KKT complementarity condition is met (or approximately met, given numerical limitations of the computing device). In summary, if the Langrangian dual L_D is concave (i.e. $-L_D$ is convex) with respect to A, then by assuming η_i to be identically zero, the solution of the dual optimization problem (and a subsequent reduction of the obtained coefficient set) can be accomplished with an iterative approach to the optimization constraints imposed on the coefficients a_i. Note that if the Langrangian dual L_D is not uniformly concave with respect to A, then the solution obtained may not be the optimal one. For now, however, we will continue to develop the solution to equation 3.17 without assuming that η_i is identically zero.

At a saddle point of L_D, we have

$$\frac{\partial L_D}{\partial a_i} = -\frac{1}{2}\sum_{j\neq i} a_j y_i y_j \left\langle x_i, x_j\right\rangle - a_i y_i^2 \left\langle x_i, x_i\right\rangle + 1 - \gamma y_i + \eta_i = 0 \quad \forall i = 1,\ldots,m. \quad (3.18)$$

With explicit inclusion of the equality constraint, $\sum_{i=1}^{m} a_i y_i = 0$, this system of

equations can be expressed in matrix form as

$$\begin{bmatrix} y_1^2\langle x_1,x_1\rangle & \cdots & \frac{1}{2}y_1y_m\langle x_1,x_m\rangle & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ \frac{1}{2}y_my_1\langle x_m,x_1\rangle & \cdots & y_m^2\langle x_m,x_m\rangle & y_m \\ y_1 & \cdots & y_m & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \\ \gamma \end{bmatrix} = \begin{bmatrix} 1+\eta_1 \\ \vdots \\ 1+\eta_m \\ 0 \end{bmatrix} \tag{3.19}$$

Let G represent the leftmost matrix in equation 3.19, so that

$$G = \begin{bmatrix} y_1^2\langle x_1,x_1\rangle & \cdots & \frac{1}{2}y_1y_m\langle x_1,x_m\rangle & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ \frac{1}{2}y_my_1\langle x_m,x_1\rangle & \cdots & y_m^2\langle x_m,x_m\rangle & y_m \\ y_1 & \cdots & y_m & 0 \end{bmatrix}$$

Then, if G is invertible,

$$\begin{bmatrix} a_1 \\ \vdots \\ a_m \\ \gamma \end{bmatrix} = G^{-1} \begin{bmatrix} 1+\eta_1 \\ \vdots \\ 1+\eta_m \\ 0 \end{bmatrix} \tag{3.20}$$

The variables η_i may be set to zero, employing iterative solution methods as earlier discussed. However, depending on the training data set, the solution obtained in this manner may converge to a local minimum (that fails to meet the primal KKT complementarity condition) rather than the global minimum, particularly under conditions of data sparsity or extreme variance within classes. Yet even this non-optimal solution can be useful in determining classification boundaries for statistically controlled datasets as we plan to later explore. Certainly, iterative solution search, gradient descent, and other methods can be employed to solve for the maximin margin hyperplane directly from equation 3.4 or from the primal and dual Langrangians (ensuring that the KKT optimality conditions are met). But such approaches are often more computationally expensive compared with the use of equation 3.20, whose dependence on training data beyond the maximin margin is

actually advantageous in the classification of statistically controlled datasets. Alternately, relaxing the requirement to solve for the optimal hyperplane, the variables η_i may be set to values that result in selection of a satisfactory separating hyperplane which may or may not be the so-called "optimal" hyperplane (which is the hyperplane that maximizes the minimum margin between itself and all of the training vectors). One such useful definition (which is to be further explored with reference to case studies of data analysis) is

$$\eta_i = -\left(\frac{\|x_{max}\| - \|x_i\|}{\|x_{max}\|}\right)^p \tag{3.21}$$

where

$$x_{max} = x_n : \|x_n\| \geq \|x_i\| \quad \forall\, i = 1,...,m;\, n \in \{1,...,m\}$$

and the choice of p ($p \geq 0$) may be dataset or problem dependent.[7] If $0^0 \triangleq 1$, then as p ranges from 0 to ∞, η_i ranges from -1 to 0. After setting or calculating values for η_i, the resultant set of coefficients a_i are used in equation 3.10 to determine w. In this case, while there may be non-zero coefficients associated with training vectors that are not on the maximin margin, the coefficients a_i are larger for training vectors on or near that margin than for the more distant training vectors thus placing more *weight* on training vectors on or close to the maximin margin.

Next, having determined w, we need to solve for b. Let x_j and x_k be training vectors from opposite classes. The functional margin, δ, between these vectors is

$$\delta_{j,k} = \left|\langle w, x_j\rangle - \langle w, x_k\rangle\right|$$

Suppose x_j and x_k are chosen such that they minimize the functional margin among all possible pairs of opposite-class training vectors. Then b may be calculated by taking the inner product of each of these two vectors with the hyperplane normal vector w

[7] The value of p can be adjusted to obtain a maximum margin hyperplane that, depending on the particular training set, approximates or actually matches the optimal hyperplane as defined in equation 3.4.

and then negating the average:

$$b = -0.5\left(\left\langle w, x_j \right\rangle + \langle w, x_k \rangle\right).$$

Equivalently (see[13, p.96]),

$$b = -0.5\left(\max_{y_i=-1} \langle w, x_i \rangle + \min_{y_i=1} \langle w, x_i \rangle\right).$$

Chapter 4

CUE DETECTION FOR STATISTICALLY CONTROLLED DATASETS

In this chapter, we explore the supervised classification behavior of a modified support vector machine (SVM), based on equation 3.20, operating on data vectors whose elements are statiscally stable with a controlled subgroup mean and controlled subgroup range or variance. In the manufacturing setting, the data vectors may be a set of direct real-valued measurements (or measurement errors) or a set of real-valued transformations of the underlying measurement or process data. For example, one may extend a set of vectors consisting of the measurements of the twelve edges of a cube by including derived values of the six face surface areas, the volume of that cube, differences between edges, etc. In some cases, a transformed set of vectors may provide classification cues not distinguishable by an SVM operating directly on the underlying measurement or error data. Alternately, this transformation can be embedded in the SVM by a suitable choice of the kernel function used as a similarity measure (see section 3.1) for the underlying measurement or error data.

4.1 Data-Dependent Anomalies of SVM's

In practical implementations of SVM's, data dependent errors due to finite-precision calculations can sometimes accumulate in ways that cause divergence from an existing clasification solution even in the separable case. Additionally, the generalizing ability of a particular SVM implementation may result in misclassification of data in a training set even when that set is separable by a maximin margin hyperplane due to spatial colinearity resolution and vector element scaling issues.

Colinearity Resolution

If a subgroup of the training set contains vectors from both classes and all vectors of the subgroup are each closer to a particular separating hyperplane than to any other

vector in its own class, the SVM may treat the vectors as if they lie on that hyperplane with respect to the separability of the vectors. For example, suppose we are given the following training set T in two-dimensional space (as illustrated in figure 4.1):

$$\begin{aligned} T &= \{(x_1, y_1), ..., (x_4, y_4)\} \\ &= \left\{ \left(\begin{bmatrix} 2 \\ 1 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 10 \\ 9 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 4 \\ 5 \end{bmatrix}, -1 \right), \left(\begin{bmatrix} 9 \\ 10 \end{bmatrix}, -1 \right) \right\} \end{aligned}$$

The optimal separating hyperplane $h_{w,b}$ for this training set can be determined by applying the definition of equation 3.4. Its parameters, for a functional margin of $\delta = 1$, are

$$w = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$
$$b = 0$$

Solving for the separating hyperplane using equations 3.19 and 3.20, with $\eta_i = 0$, results in misclassification of two vectors ($[10,9]$ and $[4,5]$) due to the proximity of the training vectors to the separating hyperplane in concert with the relative proximity of each misclassified vector to one of the opposite class. The calculated hyperplane solution is

$$w = \begin{bmatrix} -0.107945857 \\ -0.338081215 \end{bmatrix}$$
$$b = 3.122189592,$$

where w and b are normalized so that the minimum functional margin is 1. Perturbing any single vector element by 5% or less results in the same classification solution (with slightly differing separating hyperplanes). The vectors are almost colinear with

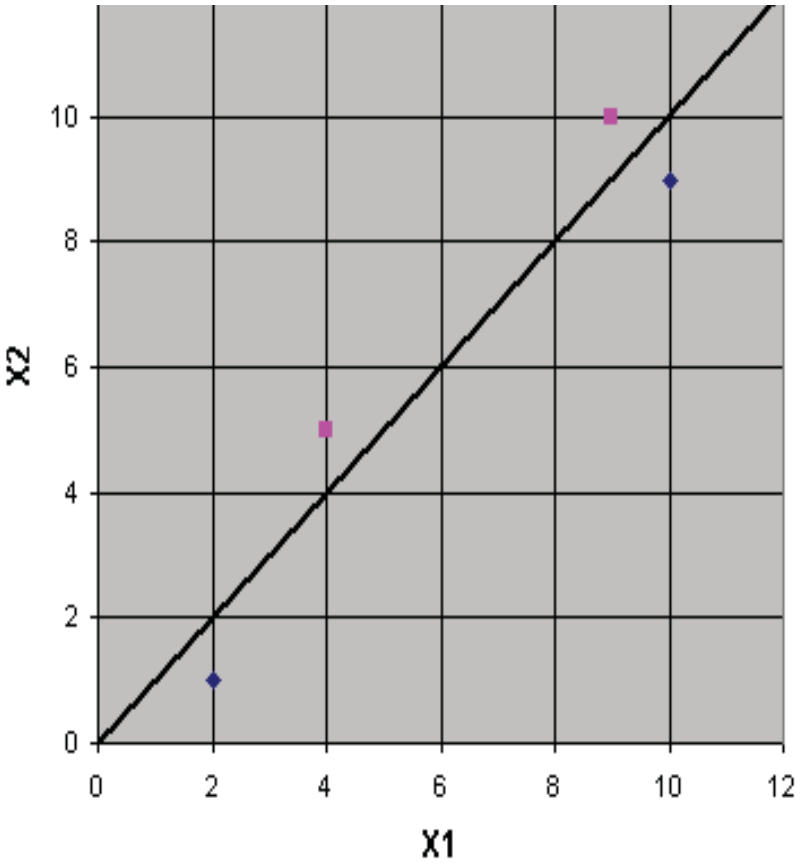

Figure 4.1: Colinearity Resolution Example

the optimal hyperplane relative to the distance between vectors of the same class and "appear" to this SVM to be non-separable.

Using equation 3.21 to determine η_i and setting p to approximately 0.017594729, we obtain the so-called "optimal" hyperplane, which does correctly classify all four input vectors. This value of p can be found by adjusting p to maximize the minimum geometric margin such that the separating hyperplane correctly classifies training vector. Again, for this hyperplane

$$w = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$b = 0$$

However, this solution is "unstable" in the sense that a very small perturbation of either the [10,9] or [9,10] vector results in a large change of hyperplane orientation and misclassification of two vectors. For example, changing the [9, 10] vector to [9, 10.000001] results in a calculated hyperplane solution of

$$w = \begin{bmatrix} -0.166951429 \\ -0.249572857 \end{bmatrix}$$

$$b = 2.91567$$

As in the case of $\eta_i = 0$, this hyperplane solution results in misclassification of the two vectors $[10,9]$ and $[4,5]$.

Vector Element Scaling

If a separable training set is composed of $N-$element vectors, where N is a positive integer greater than 1, then the relative scaling or range between vector elements may effect the ability of the SVM to locate or identify a separating hyperplane. As an example, consider the following training set T (see Figure 4.2):

$$\begin{aligned} T &= \{(x_1,y_1),...,(x_4,y_4)\} \\ &= \left\{ \left(\begin{bmatrix} 0.1 \\ 100 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 0.2 \\ 400 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 0.4 \\ 300 \end{bmatrix}, -1 \right), \left(\begin{bmatrix} 0.5 \\ 200 \end{bmatrix}, -1 \right) \right\} \end{aligned}$$

Let $x_{m,n}$ represent the n^{th} element of the m^{th} training vector. Note that for the given example the scaling, range, and range of variation (variance or sigma level) are several times greater for the elements $x_{m,2}$ than for the elements $x_{m,1}$. Assuming a functional margin of $\delta = 1$, the parameters of the optimal separating hyperplane $h_{w,b}$ for this training set are

$$w = \begin{bmatrix} -\frac{60}{7} \\ \frac{1}{350} \end{bmatrix}$$

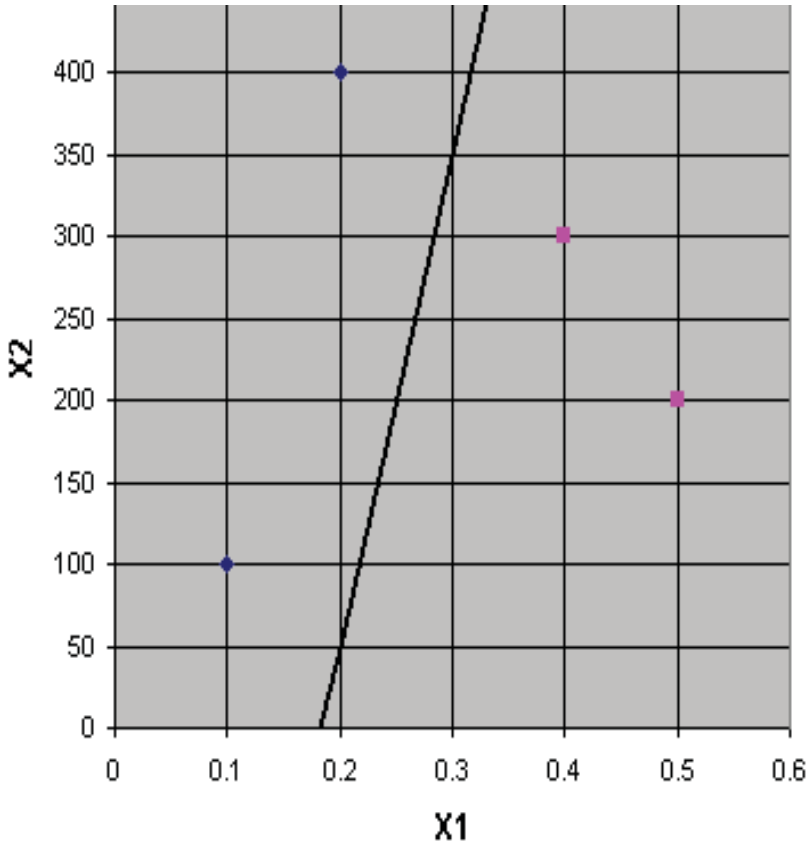

Figure 4.2: Vector Element Scaling Example

$$b = \frac{11}{7}$$

Solving for the separating hyperplane using equations 3.19 and 3.20, with $\eta_i = 0$, results in misclassification of two vectors ($[0.2, 400]$ and $[0.5, 200]$) even though the training vectors are separable on the basis of the first elements ($x_{i,1}$) alone. The calculated hyperplane solution is

$$w = \begin{bmatrix} -20 \\ -3952.573643 \\ 39 \end{bmatrix}$$

$$b = 1185779.093,$$

where w and b are normalized so that the minimum functional margin is 1. This hyperplane is a nearly horizontal line passing near the $[0.4, 300]$ vector.

This solution has apparently been dominated by the second vector element $x_{i,2}$, which (considered apart from the first element) is not separable. Why has this effect occurred? If the second vector elements of this training set had all been identical, regardless of magnitude, the SVM would have calculated the hyperplane solution parameters as

$$w = \begin{bmatrix} -10 \\ 0 \end{bmatrix}$$
$$b = 3.$$

This results defines a vertical line through the point $[0.3, 0]$ that correctly separates the training vectors. In experimenting with various values for the second vector element, it was found that the variance (or range of difference) among the input samples of this element (in comparison with the sample variance of the first vector element) rather than the absolute magnitude of the second vector element is the significant factor affecting the SVM's ability to detect the separability offered in this case by the first element.

Returning to the training set T, using equation 3.21 to determine η_i, and setting p to approximately 0.409290302, we obtain the optimal hyperplane solution mentioned earlier. However, similar to the colinearity resolution example, this SVM solution is sensitive to small ($< 0.01\%$) changes in the values of the second elements. For example, changing the [0.1, 100] vector to [0.1, 99.995] results in a calculated hyperplane solution of

$$w = \begin{bmatrix} -20 \\ -0.087089808 \end{bmatrix}$$
$$b = 33.12694253$$

As in the case of $\eta_i = 0$, this hyperplane solution results in misclassification of the two vectors $[0.2, 400]$ and $[0.5, 200]$.

4.2 Statistical Normalization

One way to reduce the effects of colinearity resolution and inter-element variance differences is to use the element sample means and element sample standard deviation to normalize the training vectors and subsequent test vectors on an element-by-element basis. These ensemble statistics (i.e. the element sample means and element sample standard deviations) may be derived from the training vectors or from a larger sample pool including the training vectors. Each input element is first offset by the element sample mean and then divided by the element sample standard deviation[1]. This has the effect of mapping or transforming the input vectors to a statistically normalized domain without affecting the relative order of the data among the input samples for an element. Hence, the separability (or non-separability) of the training vectors is unaffected by this normalization process.

Applying this statistical normalization to the two example cases above resulted in correct vector classification as follows. For the vector scaling problem, correct classification was achieved with $\eta_i = 0$, as well as with η_i determined by equation 3.21 with $p = 1$. For the colinearity problem a "stable" and correct classification result was achieved using statistical normalization along with using equation 3.21 to determine η_i with $p = 1$. However, using statistical normalization alone (i.e. with $\eta_i = 0$) for the colinearity problem still resulted in misclassification of two vectors (this time $[2, 1]$ and $[9, 10]$).

[1]If this sample standard deviation is zero, the associated input element may be set to zero without loss of generality since a sample standard deviation of zero for an element indicates that the element values (including those of the training vectors) that were used to calculate it were identical. Hence, the input vectors cannot be separated on the basis of that vector element.

Chapter 5

CASE STUDIES

5.1 Case A: Lot-Dependency of Statistically Controlled Datastream

To achieve a high level of measurement repeatability, the manufacture of a high accuracy sensor involved the bonding of highly specialized glass, metal, and silicon materials to produce subassemblies that were later incorporated at another facility into computerized measurement modules. To ensure consistency and proper operation of the sensor subassemblies, each sensor was preconditioned and required to undergo a suite of tests involving at least 44 test parameters. The pass/fail or statistical control limits established for these and other manufacturing process parameters still allowed more measurement response variation between sensors than was allowable at the end-item or module level. However, the highly consistent repeatibility of each sensor enabled its measurement response to be characterized under various conditions. Utilizing customizable parameters in the end-item device, a high degree of measurement accuracy was achieved. To validate the characterization, static and dynamic measurement error tests were performed at the end-item level.

During one period, even though the sensors from a given population were statistically “in control” with respect to the 44 test parameters used by the manufacturer, these sensors began failing a characterization error test (curve fit test) at a greater than typical rate at the end-item level prior to shipment to the customer. These failures were highly correlated with a particular manufacturing lot among the several lots that comprised this population. In the course of the investigation, the sensor manufacturer forwarded historical test data for several hundred sensors, including data for additional sensors from the suspected lot. The population failure rate for the curve fit test, excluding the suspect lot, was less than 5% while the failure rate for a tested subset of 36 units from the suspect lot was greater than 94%.

However, all 44 test parameters for this tested subset had values within the upper and lower 3-sigma control limits determined by the larger population.

Since the data used in the following case study is taken from actual process data taken from a company's data stores, an adjustment (element-by-element normalization) was performed to obscure the amplitudes and ranges of the original data. The ensemble population means and standard deviations were calculated on an element-by-element basis using the "in control" population just mentioned. Then each vector was transformed by subtracting the population mean and dividing by the population standard deviations (which were all non-zero). This transformation maintained the order relation among the vectors on an element-by-element basis and also served to reduce the influence of scale or range differences among the vector elements upon the solution of the classification hyperplane (or weight vector w_{adj}). As discussed in the previous chapter, this mapping maintains the relative separability relations among the input vectors.

The total number of samples in this case study set is 861. Let "Group A" designate the group of units with the lot code that based on a tested subset is highly correlated to characterization test failure and "Group B" the remainder of these samples (which, based on population test history, are less likely to contain characterization test failures). Group A is comprised of 160 samples, leaving 701 samples in Group B. The entire set (861 samples) was used to calculate a 44-element vector of ensemble means and a 44-element vector of ensemble standard deviations. For each data vector element, all of the samples for this study population were within 4.5 standard deviations of this ensemble mean vector. Additionally, 98.75% (158) of the lot A vectors and 84.45% (592) of the lot B vectors were also within 3 standard deviations of the mean (again considered on an element-by-element basis).

As a baseline measure of class separability, the Henze-Penrose divergence of the study set was estimated through use of minimum spanning trees on groups of 200

vectors (100 from each class) to determine Friedman-Rafsky (FR) statistics. The Henze-Penrose divergence estimate ($\widehat{HP}$) for each such group is 1 minus the FR statistic normalized by the total number of vectors (200). The range of fourteen (14) such estimates, encompassing the entire study set, was from 0.815 to 0.985, averaging 0.884 with a standard deviation of 0.053. The expected value for $\widehat{HP}$ is 0.5 if the two classes were random samples drawn from the same distribution, and 1 if the two classes are drawn from fully separable distributions. The estimated Henze-Penrose divergence for this study set indicates that there is significant divergence between the inferred class distributions, but that the distributions also have significant overlap.

For experiments A1, A2, and A3, the trainings and tests were based on detection of membership in Group A vs. Group B. For experiments A4, A5, and A6, the trainings and tests were based on detection of failed vs. non-failed units.

Experiment A1

Eight vectors (from units that had actually failed characterization testing) were randomly selected from Group A and eight other vectors (from units with no record of characterization test failure) were randomly chosen from among several of the lot sets in Group B. These vectors were then used to train the support vector machine (SVM), yielding a weight vector w_{adj} and a bias term b_{adj}, which have both been scaled or adjusted so that the minimum functional margin of the hyperplane $h_{w_{adj},b_{adj}}$ is 1. To simulate $\eta_i = 0 \,\forall\, i$, we found that setting $p \geq 100$ was sufficient. The graph in Figure 5.1 shows the data values (referenced to the left y-axis labels) for each of the sixteen vectors on an element-by-element basis. The vectors from Group A are labeled A1 thru A8 and those from Group B are labeled B1 thru B8. The vector element (or parameter) labels are indicated along the x-axis. In each parameter column, the associated weight vector element value (referenced to the right y-axis labels) is also shown. The calculated value of b_{adj} is approximately -0.60560.

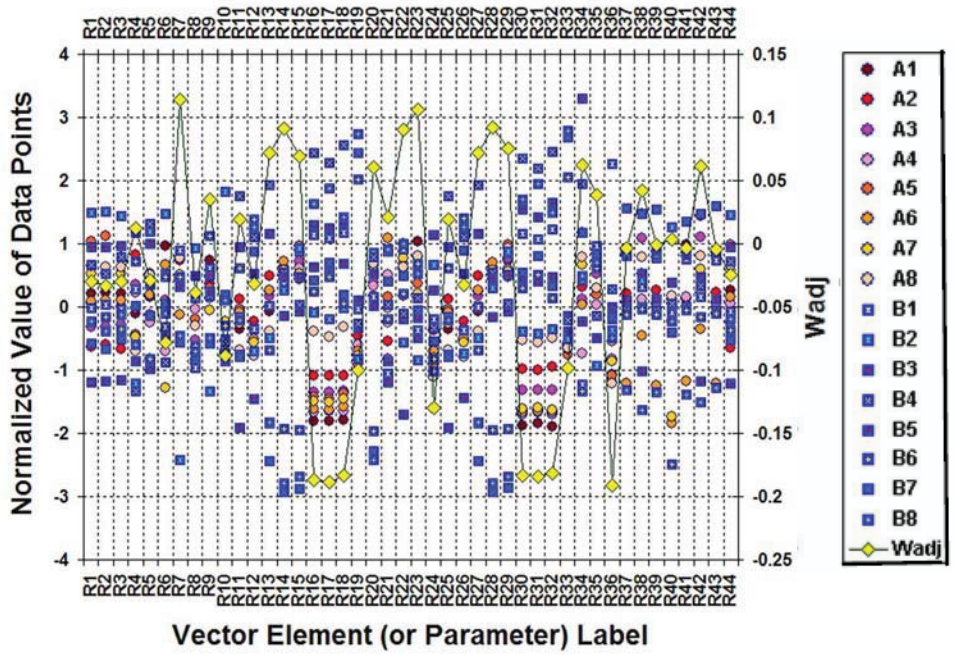

Figure 5.1: Training and Weight Vectors for Experiment A1

The modified SVM (with $\eta_i = 0$ in this case) identified a separating hyperplane (in 44-dimensional space) that did correctly classify the training vectors. Notice on the graph that the weight vector elements with the largest absolute magnitude are associated with the parameters whose training elements are the most separable. Using this weight vector and the calculated bias term (or a positive scalar multiple of this set of variables) to classify the remaining 845 vectors results in a Type 1 error or false positive rate (α) of 31.75% and a Type II error or false negative rate (β) of 23.68%, where a positive indication for a particular vector means that the associated sensor has been classified as belonging to Group A, a group of sensors suspected to have a higher likelihood to fail the characterization error test than sensors from Group B. For the twenty-six (26) Group A vectors associated with actual characterization test failures (excluding the training vectors), the false negative rate was 19.23%. This means that

the detection rate or percentage of actual failure units classified correctly by this SVM was 80.76%, which implies that 21 of the 26 failed units could possibly have been screened out at the sensor level, prior to end-item manufacture, using this classifier. This detection rate and the overall detection rate of 76.32% appear to be useful levels of detection for screening out potential downstream failures at the sensor level, prior to end-item manufacture. However, the trade-off is an unacceptably high false positive rate of 31.75%. One approach to this dilemna is to adjust the parameters of the SVM to achieve a very low false positive rate, sacrificing much of the detection power of the classifier while still providing some limited benefit of accurately screening out a portion of the units that would fail characterization tests at the end-item level.

Experiment A2

To increase the power of the test (i.e. $1-\beta$), the SVM was retrained using the eight vectors from the 34 Group A test failures that were closest to the hyperplane determined above along with the same set of previously selected vectors from Group B non-failed units. The training vectors and resultant weight vector are depicted in figure 5.2.

It can be discerned from the graph that for each of the 44 parameters, the associated training data points are not linearly separable in any single dimension. However, this training set was classified with no errors and $b_{adj} = -0.19972$ (for $\eta_i = 0$).[1] For the 845 test vectors, classification under with this new hyperplane resulted in $\alpha = 41.56\%$ and $\beta = 8.55\%$. For the thirty-four (34) Group A test failures, the false negative rate (β) was 0%. This time the overall detection rate had increased to 90.45%, but the false positive rate also increased.

[1]The training set was also correctly classified using equation 3.21 to determine η_i with $p = 1$.

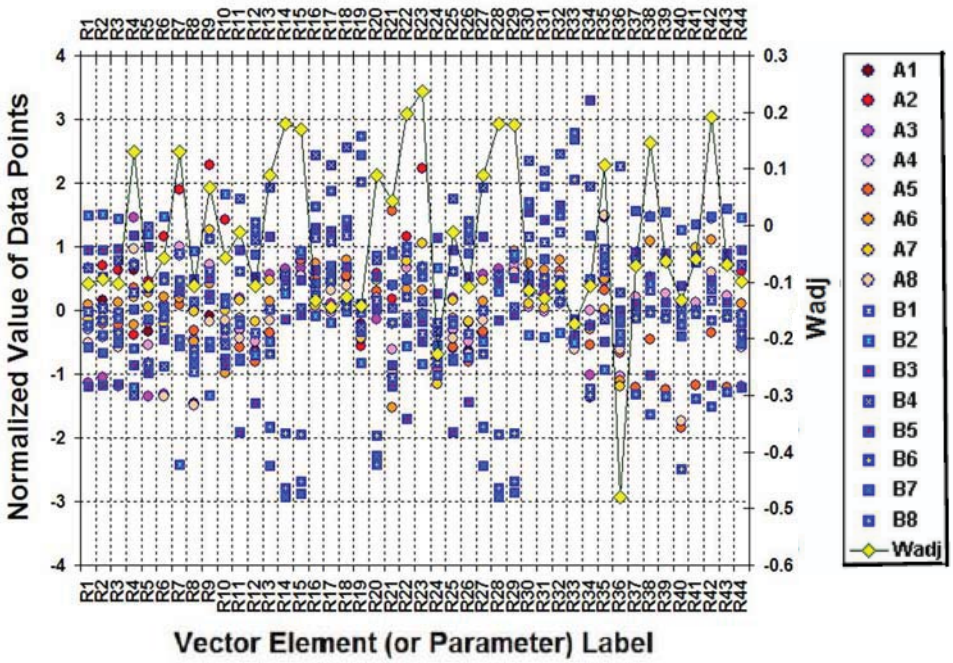

Figure 5.2: Training and Weight Vectors for Experiment A2

Experiment A3

In another experiment, a modified SVM was trained with the 34 vectors from the Group A test failures and 166 vectors from Group B nonfailed units (including the 8 vectors used in the previous two experiments). With $\eta_i = 0$, the training error was 2% with one (1) Type I error and three (3) Type II errors. When η_i was determined by equation 3.21 with $p = 0.5$, the training error was 1% with no Type I errors. Using this SVM to classify the remaining 661 vectors results in a false positive rate (α) of 10.28% and a false negative rate (β) of 80.16%. The level of significance of the test (i.e. $1-\alpha$) appeared to improve (increase) over that of the SVM trained with only 16 vectors, but the power of the test (i.e. $1-\beta$) or ability of the SVM to pre-identify sensors from Group A had greatly decreased. However, this interpretation implicitly assumes that Group A is homogenous. A closer review of the 34 failures from this lot

reveals that the failures from this lot were manufactured during the same 3-day period and designated with the same sublot letter. A total of 36 units from this sublot were included in the Group A lot. Thus, the SVM based on 200 training vectors turned out to be more selective with respect to detecting potential characterization test failures from Group A than the two earlier SVM's, which were more effective in detecting Group A membership.

In this experiment, along with some other 16-training-vector SVM experiments, it was noted that the percentage of Type I errors increased dramatically for the last 100 test parameter vectors from Group B. The set of test vectors were arranged such that the vectors for sensors from the same manufacturing lot tended to be listed together. For the 533 Group B test vectors, twenty-two (22) vectors within the last 100 listed accounted for 40% of the overall false positive errors under this experiment. One particular lot (a set of 8 sensors) from this subgroup had a false positive rate of 100%. This large deviation from the nominal false positive rate indicates the possibility that these sensors may have parameter characteristics similar to those from lots associated with Group A that tend to fail the characterization test.

Experiment A4

Again, using 200 training vectors, an SVM was trained; this time using vectors from the 36 characterization test failures (17 from Group A and 19 from Group B) for one class and 164 vectors from non-failed units (41 from Group A and 123 from Group B) for the second class. Hence, for this experiment, a positive indication is associated directly with membership in the group of sensors likely to fail characterization test rather than membership in Group A. The remaining vectors were used as a test set, which now contained 36 vectors belonging to the class of known characterization failures and 625 non-failed sensors. For this SVM, η_i was determined by equation 3.21 with $p = 1$. An errorless classification hyperplane was not found, so b_{adj} (the

SVM offset parameter) was adjusted to achieve minimal training error. At $b_{adj} = -32.7$, the training error rate was 6.5% (3.5% of which was due to Type I error). Using this SVM to classify the remaining 661 vectors results in a false positive rate of 18.88% and a false negative rate of 41.67%.

For the four SVM implementations above, the false positive rates were too high to enable practical direct use of these SVM's as predictive discriminators for potential characterization test failure. A predictive discriminator with a high false negative rate (β) or low positive-detection rate ($1-\beta$) might still provide an economic or time-savings value if this value is not outweighed by the loss of units removed from production due to a false positive indication. For example, if the false positive rate is zero or suitably small, then even a 5% positive-detection rate ($\beta = 0.95$) might enable significant time and cost savings through avoidance of processing some units likely to fail characterization test. An attempt had been made to reduce the false positive rate for the last SVM by modifying the value of b_{adj}. However, as the false positive rate approached zero in this case, the positive-detection rate also approached zero.

Experiment A5 and Experiment A6

In reviewing the weight vector (W_{adj}) for the last SVM described above, it was noted (see the figure 5.3) that 30 of the 44 weight vector elements were less than 25% of the magnitude of the largest weight vector element. The other 30 elements of each training vector were zeroed out. This resulted in the corresponding elements of the resultant weight vector also being zeroed out, effectively reducing the 861 case study vectors to include only the remaining 14 elements. Under these conditions, the SVM was retrained using the same training set as used for the previous SVM. This time, η_i was determined by equation 3.21 with $p = 0.415$. A satisfactory compromise between the false positive rate and the positive-detection rate was achieved with $b_{adj} = -116.24$. The training error rate was 14% with no Type I error. Using this SVM to classify the

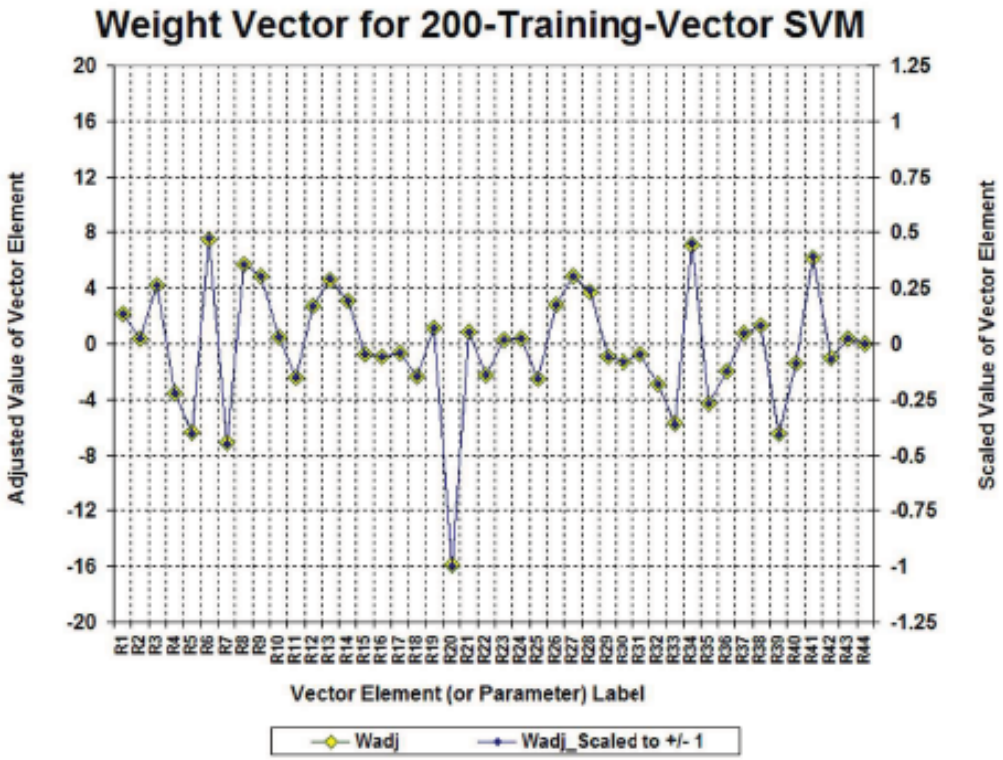

Figure 5.3: Weight Vector for Experiment A4

remaining 661 test vectors resulted in a false positive rate of 1.92% and a positive-detection rate of 11.11% ($\beta = 0.8889$). Alternately, setting $b_{adj} = -18.017$, resulted in a false positive rate of 0.96% with a positive-detection rate of 8.33%. The training error rate for this setting was 17%, again with no Type I error.

Observations and Conclusions

Since the 72 characterization failures noted in this 861-unit sample set tended to largely occur in sublot groupings, this latter SVM might be used as follows:

1. Use the SVM as a detector on a sublot basis.

2. Indicate that a sublot is suspect if its SVM detection rate exceeds, say, 6% (where the detection rate equals number of SVM-resultant positive indications divided by the tested sublot population).

Using the SVM as a pass/fail detector on a sublot basis requires only that there exists a sufficient spread between the false positive rate (α) and the positive-detection rate ($1-\beta$), with the positive-detection the rate being the greater of the two. Then, one can set a threshold between these two rates or, for a more rejection-cautious approach, above the initially determined SVM positive-detection rate and use that threshold to decide whether a particular test sublot is likely to have a high characterization test failure rate based on the SVM positive indication rate (positive test indications/number of test samples). For instance, when the fourth SVM described above was used to classify the 661 test vectors, the result was a false positive rate of 18.9% and a positive-detection rate of 58.3%. If a positive-indication threshold of 65% is selected, then application of this SVM detector to each of the 41 sublots of the 861-unit study group results in rejection of 3 sublots:

1. The original 36-unit sublot from Group A that contained 34 characterization test failures with an SVM positive-indication rate of 81%.

2. A 23-unit sublot from Group B that contained 20 characterization test failures and had an SVM positive-indication rate of 82%.

3. An 8-unit sublot from Group B that contained only 1 characterization test failure, but had a SVM positive-indication rate of 75%.

Suppose this SVM discriminator had been applied to identify these same sublots as suspect. Then not using these sublots would have resulted in avoiding the manufacturing efforts and test time spent on 55 end-item characterization test failures with the trade-off of falsely rejecting 12 sensors that have no record of characterization test failure. If a positive-indication threshold of 80% were chosen, the result would be rejection of 54 end-item characterization test failures with false rejection of 5 sensors with no record of test failure.

Experiment	Training Vectors	Training Error	Test Vectors	Type I Error (α)	Type II Error (β)	Detection Rate ($1-\beta$)
A1	16	0	845	0.318	0.237	0.763
A2	16	0	845	0.416	0.086	0.915
A3	200	0.01	661	0.103	0.802	0.198
A4	200	0.065	661	0.189	0.417	0.583
A5	200	0.14	661	0.0192	0.889	0.111
A6	200	0.17	661	0.0096	0.917	0.083

Table 5.1: Summary of Results for Case Study A

Of course, actual implementation in a manufacturing environment of an SVM sublot-based discriminator to prescreen sensors would likely require obtaining additional sensor test data, particularly more samples of units that failed characterization testing, and further validating of the performance and cost-effectiveness of the SVM discriminator(s) given various detection rate thresholds. In discussing some of these results with one of the company's design engineers, it was also suggested that pre-sensor level data be reviewed for detection cues as well. This is possible due to the fact that some electrical testing of sensor components occurs before the construction of the sensor itself. If an SVM discriminator could detect failure cues in that data, then such results could be used to (1) avoid utilization of suspect components (or component lots) in the further manufacture of sensors and/or (2) possibly determine the root causes of the failure mechanism(s), enabling process or product improvements that could remove or reduce these root sources of failure.

5.2 Case B: Latent Sensor Failure

A group of products that had passed production testing at the sensor level later failed in a particular mode sometimes prior to and sometimes after shipment to the customer. Failure analysis of several of the sensors indicated that the failure mode was not related to mechanical shock or mishandling, but rather was related to manufacturing process limitations or variations. Also, the failure rate was very small compared to the

field population of product. One question is whether there are cues in the sensor production test data that correlate with a propensity of a sensor to later fail in this mode. In an attempt to find these cues (if any), an SVM was developed and subsequently trained using statistically normalized input data and equation 3.21 with $p = 1$. The parameters vectors in this case contained 28 elements. Both the training set (16 vectors) and the test set (152 vectors) had vectors whose elements were all within 4.2 sigma of the element mean[2]. Of the 168 sensors selected for this study, five (5) had failed after shipment to the customer (i.e. in the "field") and 27 had failed during end-item factory testing. The remaining 136 sensors were not known to have failed (or had indeed passed end-item testing). Since the dormancy of the latent failure mode would have some dependencies on the various environments to which the sensors are exposed, some of these remaining 136 samples, while treated as negative, might have cues (if they exist) that correlate to this failure mode.

The Henze-Penrose divergence of this study set was estimated through use of minimum spanning trees on groups of 64 vectors (32 from each class) to determine Friedman-Rafsky (FR) statistics. The range of five (5) such estimates, encompassing the entire study set, was from 0.531 to 0.549, averaging 0.541 with a standard deviation of 0.008. The estimated Henze-Penrose divergence for this study set indicates that these two classes are likely drawn from the same approximate distribution. Another Henze-Penrose estimate was calculated using an unbalanced test set comprised of 32 vectors in one class and 136 vectors in the second class. In this case, the divergence estimate was 0.720. For this unbalanced test set, the expected Henze-Penrose estimate, given the hypothesis that the samples come from the same

[2]Initially, element sigmas and means were derived based on element-by-element ensemble statistics for 188 vectors. Twenty (20) of these were between 3 to 6 sigma away from the mean and were subsequently removed from the test set, leaving the subject 168 vectors referenced above. Recalculation of the element sigmas and means based on the 168 vectors results in each element value remaining within 4.2 sigma of the element mean.

distribution is 0.692. Again, the divergence estimate indicates that there is little divergence between the two sample sets and favors the hypothesis that the samples are drawn from the same underlying distribution.

For experiment B1, the training and tests were based on detection of field failures vs. non-failed units. For experiment B2, the training and tests were based on detection of field and factory failures vs. non-failed units.

Experiment B1

In one experiment, the five field failures were used as positive training examples.. The eleven "negative" training samples taken from the non-failed population were selected based on iterative SVM training and subsequent testing in an attempt to maximize the SVM detection rate for the 5 field failure failures while minimizing Type I error among the non-failed samples. The SVM was trained with no errors. The training vectors and associated weight vector are shown in figure 5.4. The Type I error rate for the test set was 51.2%. The Type II error rate was 70.4%. The SVM positive-detection rate for the non-failed test group exceeded that of the known group of factory failures.

Experiment B2

In another experiment, three of the factory failed units were also included in the training group, for a total of eight positive examples. The eight negative examples were iteratively selected from the non-failed population to minimize Type II test error while ensuring that Type I error was not 100%. Errorless training was achieved and the Type II test error attained was 20.8%. However, the Type I test error was 82.8%, implying similar SVM positive-detection rates for both the group of factory failures and the "nonfailed" group. The training results are shown in figure 5.5.

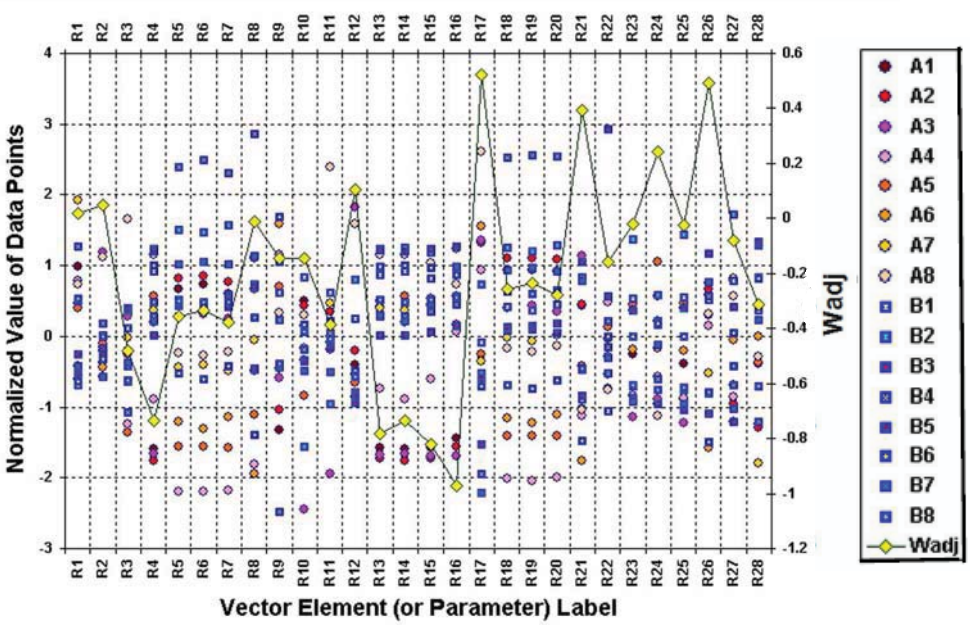

Figure 5.4: Training and Weight Vectors for Experiment B1

Observations and Conclusions

In these and several other experiments, it was observed that while the field failures as a group appeared to be detectable and largely separable (based on functional margins) from the nonfailed group[3], the factory failures appeared to have similar SVM positive-detection rates as those of the nonfailed group. In other words, the factory failures appear likely to belong to the same class as the remaining population with respect to the implemented SVM discriminators. This latter result was to be expected on the basis of the Henze-Penrose divergence estimates on the data which favored the hypothesis that the two classes were samples drawn from the same distribution.

[3]Of course, since the training set includes the data from only five field failures, the statistical significance of this 100 % detectability is in doubt. Based on the Type I error (α) obtained using 136 samples for the first SVM described for this case study, the empirical level of significance ($1-\alpha$) is only about 30%.

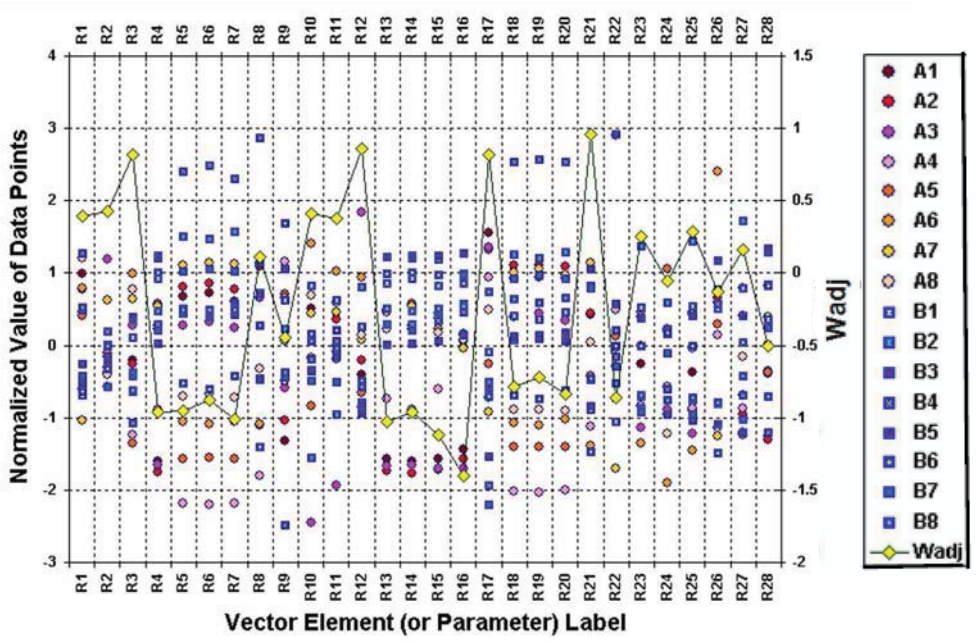

Figure 5.5: Training and Weight Vectors for Experiment B2

So far, no cues had surfaced with respect to predetection of the factory failures. It may be that the sensor level test data does not reveal the manufacturing anomalies (if any) that through subsequent environmental or end-item test exposure result in sensor failure. Or, perhaps these factory failures occur as a result of the end-item manufacturing processes apart from any manufacturing flaws in the sensors.

Concerning the field failure units, more samples of field failures would need to be obtained in order to test whether either of the two SVM's described above could indeed predetect cues for field failures (assuming that these cues could be different than those of the factory failures). Since the time that data was obtained for the 168 units of this study, at least five other units with the failure of interest have been returned from the field. Root cause investigation eventually resulted in identifying process improvements and product testing scenarios that have since resulted in the

Experiment	Training Vectors	Training Error	Test Vectors	Type I Error (α)	Type II Error (β)	Detection Rate ($1-\beta$)
B1	16	0	152	0.50	0.70	0.30
B2	16	0	152	0.828	0.208	0.792

Table 5.2: Summary of Results for Case Study B

field occurence of this latent failure mode becoming a very rare event. The identified root cause is supportive of the hypothesis that the sensor performance effects of this latent fault mode are not apparent until actual failure occurs. Note that the high false positive rates obtained from the SVM under both experiments are also consistent with this hypothesis. While the training was errorless for both experiments, the resultant predictor did not generalize well with respect to its classification performance given test samples from data not included in the training.

5.3 Case C: Range Capability

A group of sensors designed for a specified performance range were utilized in end items with extended measurement range. However, it was later discovered that only a portion of these sensors could function adequately within the entire extended range. Some sensors had upper or lower performance limits that were less than extended range requirements. It was determined through correlation analysis of past end-item test data that preselection of sensors that could meet extended range requirement could be based on two parameter elements of the 44-element sensor-level test data vector, namely parameter vector elements 17 and 31. While other vector elements could also be used, these two elements provided a means of adequate prediction of which sensor could pass extended range testing. Without this preselection, the historical test data analyzed indicated that about half (30% to 70%) of the sensors manufactured for the nominal range were probably not capable of passing extended range testing. The thresholds for these two parameter elements were chosen to minimize the probability

of that a non-capable sensor would be selected for use in the extended range end item. Some of the sensors rejected as extended range candidates may well have been able to pass extended range testing.

Let the variables $R17$ and $R31$ represent values of the parameter elements 17 and 31, respectively. In general terms, $R17$ represents the maximum output of the sensor under certain input conditions. $R31$ represents the change in sensor output (or output span) given a particular minimum and maximum input range. Using the statistically normalized ranges of these parameter elements (rather than the native ranges), the preselection criteria is to accept as extended range capable those sensors for which $R17 < 0.49183$ and $R31 < -0.425353$. The pass/fail results for end items whose sensors were selected based on this criteria were used to validate these criteria limits. Review of several end-item production runs revealed no failures due to range capability when this criteria was used. These results were also consistent with the underlying sensor characteristics represented by the selection variables. To be specific, the upper and lower measurement ranges of the sensor could be extended to the higher range if the nominal maximum response was held below some fixed maximum value and the nominal span was also held below an associated fixed maximum value.

This case study is based on the sensor-level data for 753 sensors. The samples for each vector element of the data for this study population were within 4.5 standard deviations of the associated ensemble mean. The assumed "true" classification of each sensor is based on the preselection criteria previously described. Three SVM's were developed and trained using portions of this sensor-level data. The data not used for training was used as test data as a means of evaluating the performance of the SVM's. Statistically normalized input data was utilized for all three SVM's with η_i selected in accordance with equation 3.21. A positive classification is taken to indicate that a sensor is predicted to pass extended range testing. Therefore, the classification of a sensor by an SVM as capable when the "true" classification (based on our

conservative selection criteria) should be non-capable is a Type I error (false positive). Classification of a sensor by an SVM as non-capable when the true classification is capable is a Type II error (false negative). The 753-unit dataset contains 277 positive samples and 476 negative samples. Since our base selection criteria is believed to be overly restrictive (risk-adverse) with respect to selecting extended-range-capable sensors, then a more accurate discriminator (with respect to the initial dataset) would be expected to yield reduced Type II errors (rejections of capable sensors) when evaluated in terms of future end item results. However, the generalization trade-off is an increased risk in the future of admitting some sensors that are actually non-capable.

The Henze-Penrose divergence of this study set was estimated through use of minimum spanning trees on groups of 200 vectors (100 from each class) to determine Friedman-Rafsky (FR) statistics. The range of fifteen (15) such estimates, encompassing the entire study set, was from 0.775 to 0.975, averaging 0.905 with a standard deviation of 0.053. The estimated Henze-Penrose divergence for this study set indicates that there is significant divergence between the inferred class distributions, but that the distributions also have some overlap.

All three experiments were based on detection of extended range capable vs. non-capable units.

Experiment C1

The first SVM was trained with eight positive (extended range capable) samples and eight negative (rejected) samples. The training data and resultant weight vector are shown in figure 5.6. The training vectors were classified with no errors using $p = 1$ in equation 3.21 and $b = 0.1058275$ for the bias term of equation 3.7. For this implementation, using the remaining 737 samples as test samples, the Type I test error was 29.7%. The Type II test error was 5.2%.

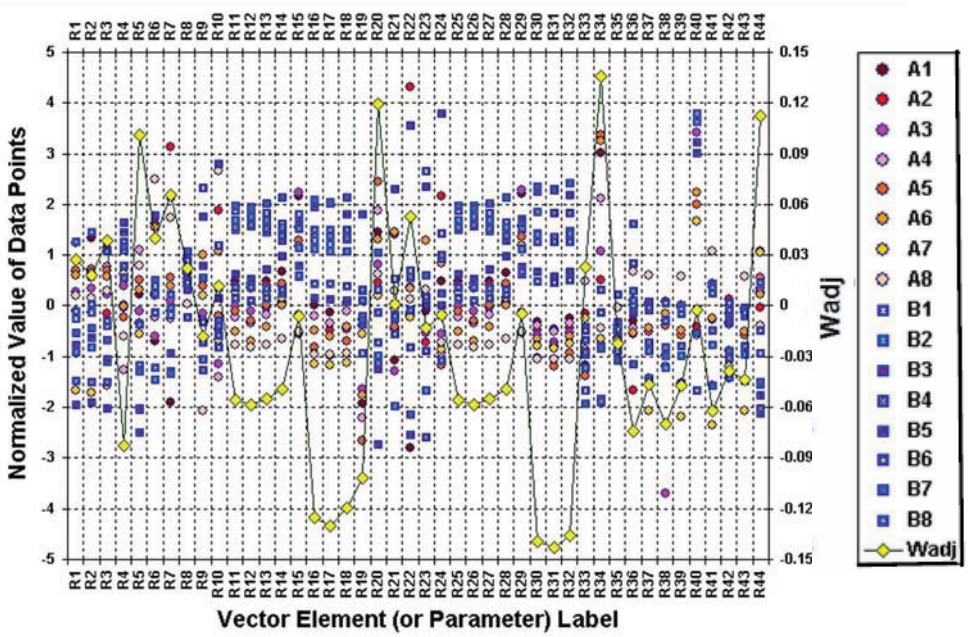

Figure 5.6: Training and Weight Vectors for Experiment C1

Experiment C2

Another SVM was trained with 100 positive samples and 100 negative samples using $p = 5$ and $b = -0.1$. The average training error was 3% with a Type I error of 5% and Type II error of 1%. The resultant weight vector for this SVM is depicted in figure 5.7. Using the remaining 553 samples to test this SVM resulted in a Type I test error (α) of 21.3% and a Type II test error (β) of 2.8%. Both the power of the test ($1-\beta$) and the specificity of the test or significance level ($1-\alpha$) had apparently improved in comparison with the first SVM (even with a first-order proportional accounting for the test set size difference).

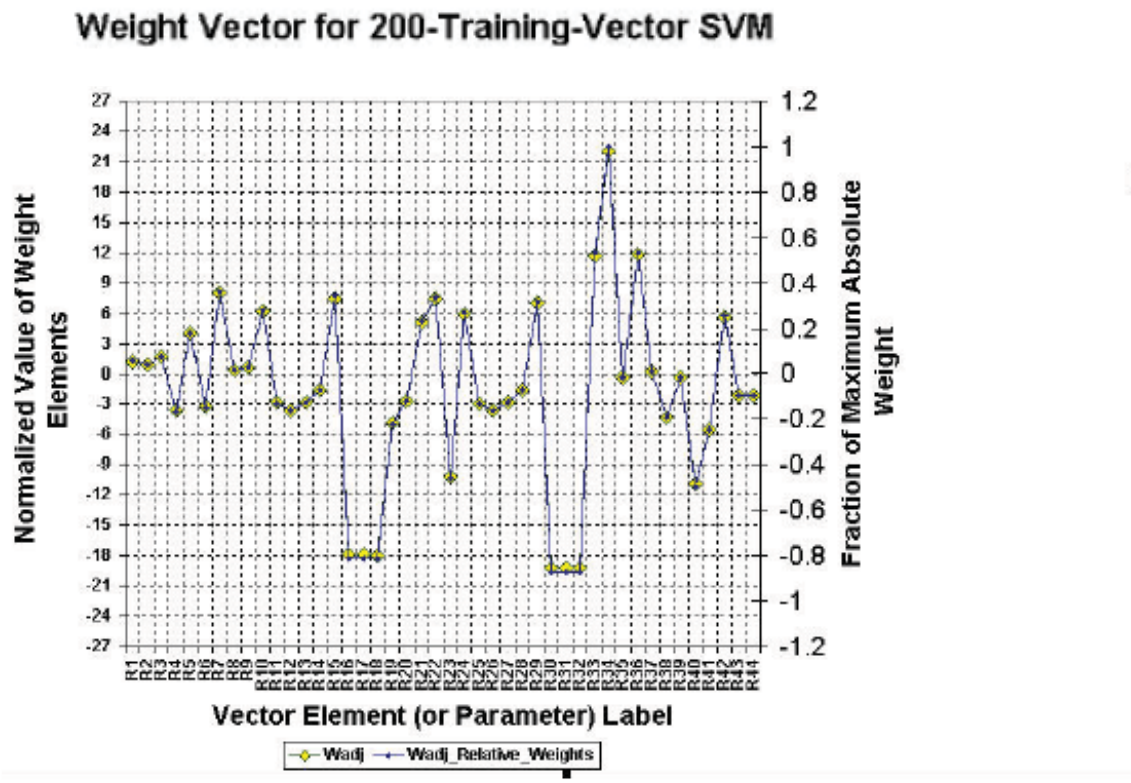

Figure 5.7: Weight Vector for Experiment C2

Experiment C3

For the first SVM, the largest absolute value of an element of the weight vector (W_{adj}) was attained by element $R31$. In figure 6, note that five(5) other elements of the W_{adj} vector have an absolute value that exceeds 85% of this maximum value. In order of decreasing magnitude these are $R30, R32, R34, R17,$ and $R16$. For the second SVM (see figure 7), the largest absolute element value of W_{adj} was attained by element $R34$, which is associated with a measure of the non-linearity of the sensor response to input stimuli. The next five elements in order of decreasing magnitude were $R31, R32, R30, R18,$ and $R17$. The appearance of $R17$ and $R31$ among the top six

elements of the weight vector with respect to absolute magnitude is not unexpected, of course, since two elements are the basis of the criteria use to determine the "true" classification of each sensor. $R16, R17,$ and $R18$ are identical measurements of the same underlying parameter (maximum nominal output) taken under different environmental conditions. Similarly, $R30, R31,$ and $R32$ share the same underlying parameter (nominal span). Both SVM implementations identified $R34$ as being among the four most significant elements of the weight vector.

Based on the weight vector results described above for the first SVM, and including only the highest magnitude element from the two measurement triplets, the three most significant weight vector elements are $R31, R34,$ and $R17$. A third SVM was trained using only these three elements (the other vector elements were set to zero during the training phase only). The training vectors used were those used for the first SVM. For this experiment, $p = 1$ and $b = -0.119389$. The resultant SVM had no training errors. The training data and resultant weight vector are shown in figure 5.8. Again using the remaining 737 samples as test samples, the Type I test error was 16.7%. The Type II test error was 1.5%.

Observations and Conclusions

The third SVM, which uses only three active (non-zero) elements for its weight vector, provided classification results that had the least test errors with respect to the assumed base line classification of the data. Application of this SVM to the preselection of sensors for use in extended range applications would result in rejecting less sensors overall and thus increasing the number of sensors from a given manufactured quantity that are available for extended range end items. Note that while elements $R17$ and $R31$ were separable on an element-by-element basis for the selected training vectors, $R34$ was not. The addition of the non-linearity factor ($R34$) might serve to avoid use of sensors that would not be capable at either the higher or lower end of the extended

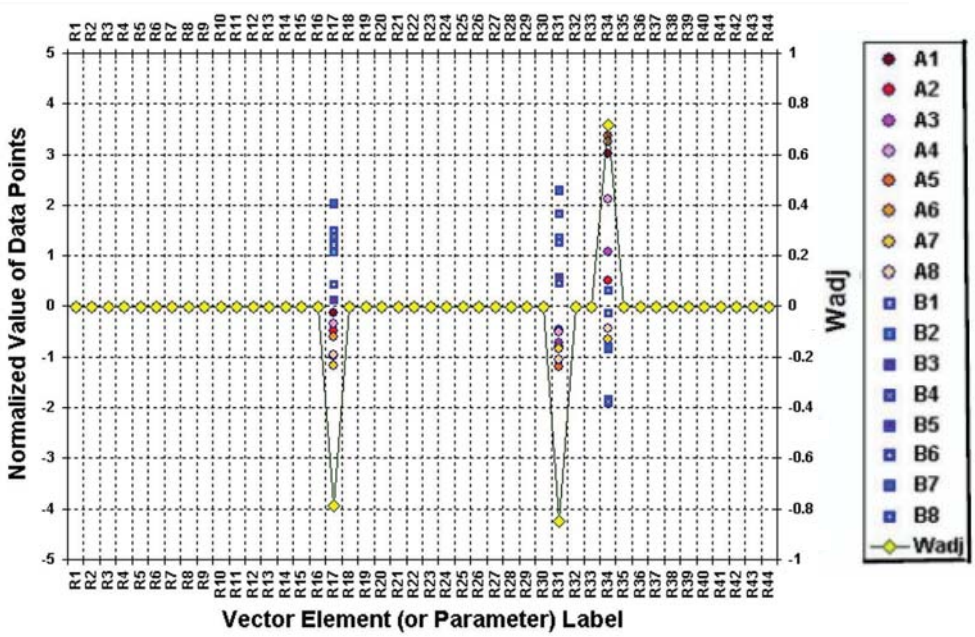

Figure 5.8: Training and Weight Vectors for Experiment C3

measurement range even though the maximum nominal (tested) output and nominal span results might appear to indicate a capable sensor based on linear extrapolation. The implementation of this SVM is feasible in a production environment since classification of each sensor would now be dependent on the sign of the result of summing the weighted values of only three vector elements. The false positive rate of 16.7% in Experiment C3 may be of some concern, even though the detection rate of 98.5% is good. Based on current selection criteria, this false positive rate implies that some “non-capable” sensors would be falsely accepted as extended range capable. However, the current criteria are known to be overly restrictive in acceptance of extended range sensors. This is not deemed a problem, because sensors that are falsely rejected as extended range capable may still be used in normal range applications. It is uncertain whether use of this SVM would actually result in an increase in the rate of end-item extended range capability failures or, instead, would result in an increase in

Experiment	Training Vectors	Training Error	Test Vectors	Type I Error (α)	Type II Error (β)	Detection Rate ($1-\beta$)
C1	16	0	737	0.297	0.052	0.948
C2	200	0.03	553	0.213	0.028	0.972
C3	16	0	737	0.167	0.015	0.985

Table 5.3: Summary of Results for Case Study C

the number of detected extended range sensors available due a less restrictive selection criteria. Whether this is indeed a better discriminator would need to be validated by end-item test results for sensors that have been classified by this SVM as capable. In any case, use of this SVM might still be justified depending on desired tradeoffs between the need to increase extended range sensor supplies, the avoidance of costs associated with end-item test failure, and the need to optimize sensor production flow (by reducing the total amount of sensors that may need to be built in order to obtain a given amount of extended range sensors).

Chapter 6

KERNELS AND THE HYPERPLANE CLASSIFIER

6.1 Basic Definitions

In chapter 3, the use of the hyperplane classifer was discussed with a brief allusion to use of a similarity measure k, defined on the input product space $X \times X$, which measure was in turn the inner product of the transforms of the inputs or patterns x into a separable Hilbert space H:

$$k(x,x') = \left\langle \Phi(x), \Phi(x') \right\rangle .$$

Prior to further discussion of the use of this similarity measure in extending the usefulness of the hyperplane classifiers, we will overview some fundamental definitions of Hilbert space, kernels, and reproducing kernel Hilbert spaces.

Hilbert Space

Let $\mathscr{H}$ be a vector space equipped with an inner product

$$\langle\,\rangle : \mathscr{H} \times \mathscr{H} \to \mathbb{C} \text{ or } \mathbb{R}$$

$$(x,y) \mapsto z := \langle x,y \rangle$$

that induces a norm on $\mathscr{H}$ by

$$\|x\| = \sqrt{\langle x,x \rangle}.$$

This norm in turn induces a metric d on $\mathscr{H}$ by $d(x,y) = \|x-y\|$.

$\mathscr{H}$ is a *Hilbert Space* if it is complete[1] with respect to the metric d [33, p.11][28, p.164][34, p.307]. In addition, the Hilbert space has a countable

[1] i.e. every Cauchy sequence in $\mathscr{H}$ converges to a member of $\mathscr{H}$ [32, p.49][28, p.14]

orthonormal basis if and only if it is separable [28, p.168]. The Riesz Representation Theorem shows that each element of a Hilbert space defines a linear functional on that space by the inner product [28, pp.164-165, 205][32, p.50]. That is, given $x, y \in \mathscr{H}$, we can define a corresponding linear functional f_x as

$$f_x(y) = \langle x, y \rangle .$$

And every continuous linear functional on $\mathscr{H}$ can be represented in this form. [35, pp.40-47,130-132]

Kernel Function

A function k on $X \times X$ is a *kernel* if it has the form $k(x,y) = \langle \Phi(x), \Phi(y) \rangle$ for some $\Phi : X \to \mathscr{H}$.

If X itself is already embedded in Hilbert space , then one possible kernel is $k(x,x') = \langle x, x' \rangle$, using the identity function $\Phi(x) = x$. However, even with X being embedded in a separable Hilbert space, there may be instances in which separable classes in X may not be linearly separable in the X domain but may be linearly separable if transformed non-linearly into another isomorphic Hilbert space. Under certain conditions, we may be able to exploit this resultant linear separability through direct use of the kernel k as a similarity measure without resort to the intermediate step of explicit transformation of the input variables.

Reproducing Kernel Hilbert Space (RKHS)

Suppose $\mathscr{H}$ is a Hilbert space of vector-valued functions defined on a nonempty domain set X [36, p.23]. That is, $\mathscr{H}$ is associated with an inner product (or dot product) $\langle f, g \rangle = z$ and a norm $\|f\| := \sqrt{\langle f, f \rangle}$, where $f, g \in \mathscr{H}$ and $z \in \mathbb{C}$ or $\mathbb{R}$ [27, p.36]. Then $\mathscr{H}$ is a *reproducing kernel Hilbert space* if there exists a *reproducing kernel (r.k.) K* defined on $X \times X$ such that [33, p.12][36, p.23]:

1. For any fixed $y \in X$, the function $K_y(x)$, where $K_y(x) \equiv K(x,y)$, is an element of $\mathscr{H}$.

2. For every $y \in X$ and every $f \in \mathscr{H}$,

$$f(y) = \left\langle f(x), K_y(x) \right\rangle .$$

Since $K_y \in \mathscr{H}\ \forall y \in X$, then from (2) we have for any fixed $w, y \in X$ [27, p.36],

$$\left\langle K_y(\cdot), K_w(\cdot) \right\rangle = K_y(w) = K(w,y) \tag{6.1}$$

6.2 Properties of Reproducing Kernel Hilbert Spaces

1. Uniqueness: For the inner product space $\mathscr{H}$, $\langle f, g \rangle = \overline{\langle g, f \rangle}$ if $\mathscr{H}$ is complex (where $\overline{\langle \cdot \rangle}$ is complex conjugation) [28, p.163]. If K is an r.k. for a particular Hilbert space $\mathscr{H}$ and L is also an r.k. for $\mathscr{H}$, then equation 6.1 implies the identity of L and K [33, p.12][36, p.23].

2. Existence: A necessary and sufficient condition for the existence of an r.k. for a Hilbert space $\mathscr{H}$ is the existence of a functional $f \in \mathscr{H}$ that is continuous at every $y \in X$ [33, p.12].

3. Reproducing Kernels on Finite-Dimensional Hilbert Spaces: If $\mathscr{H}$ has finite dimension n and $f_1, \cdots, f_n$ are linearly independent functions in $\mathscr{H}$, then a reproducing kernel K exists if and only if the conjugate of the inverse of the Gram matrix of the system of functions is positive definite. If K exists, it can be determined as a function the inner products of the vectors $f_1, \cdots, f_n$ [33, pp.15-16]. (See also Mercer's theorem [13, p.35], which details the conditions under which $K(x,z)$ can be expanded as a uniformly convergent series in terms of $f_1, \cdots, f_n$)

6.3 The "Kernel Trick" and the Hyperplane Classifier

Non-Linear Transformation and Linear Separability

Consider the following training set S consisting of elements x_i in $\mathbb{R}^2$ and their associated class labels $y_i \in \{-1, 1\}$:

$$
\begin{aligned}
S &= \{(x_1, y_1), ..., (x_8, y_8)\} \\
&= \left\{ \left(\begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 0 \\ 0.5 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} -0.5 \\ 0 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 0 \\ -0.5 \end{bmatrix}, 1 \right), \right. \\
&\quad \left. \left(\begin{bmatrix} 1.5 \\ 0 \end{bmatrix}, -1 \right), \left(\begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, -1 \right), \left(\begin{bmatrix} -1.5 \\ 0 \end{bmatrix}, -1 \right), \left(\begin{bmatrix} 0 \\ -1.5 \end{bmatrix}, -1 \right) \right\}
\end{aligned}
$$

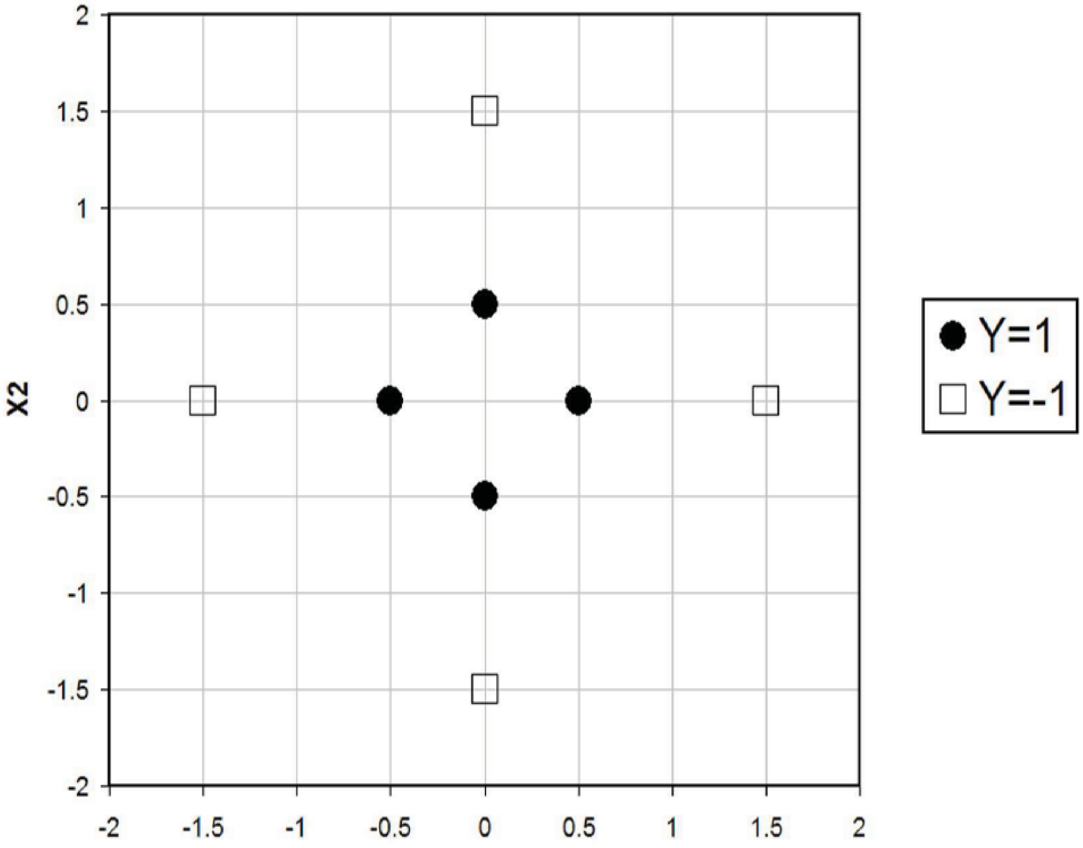

Figure 6.1: X (Input) Domain

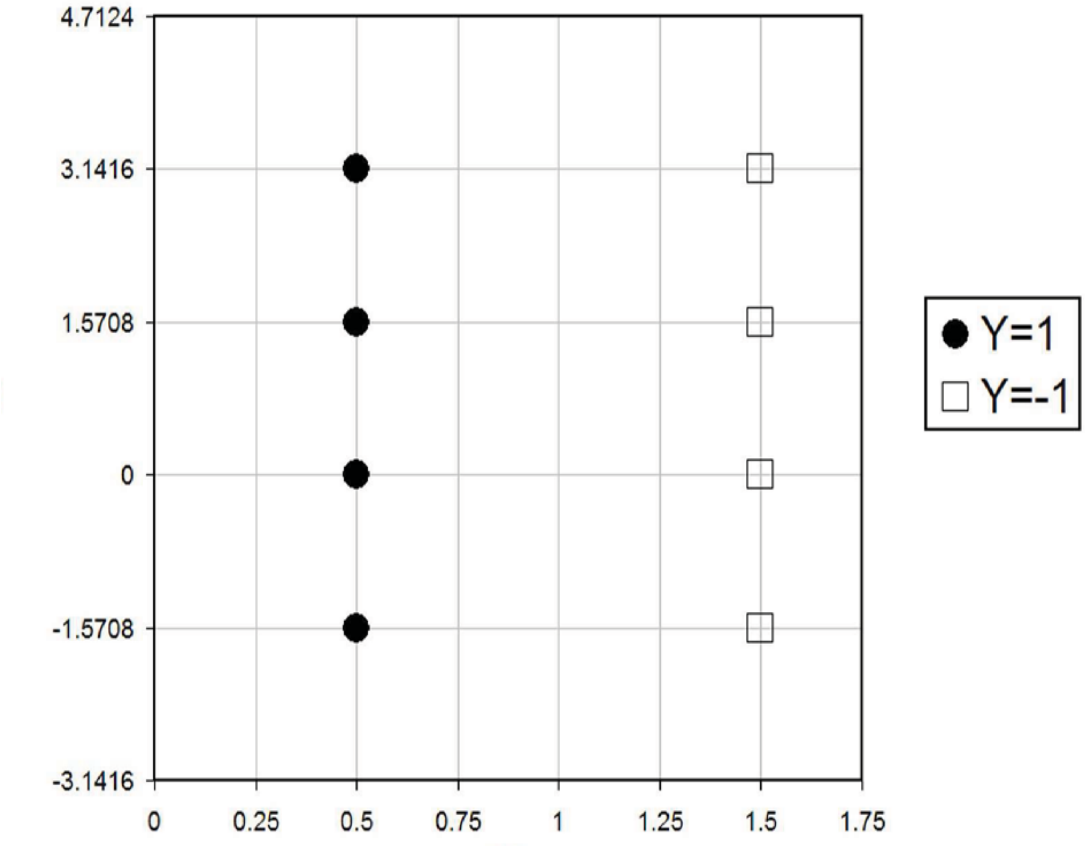

Figure 6.2: V (Transform) Domain

Figure 1 depicts this training set. The estimated Henze-Penrose divergence between the two classes (based on FR statistics) is 0.500, indicating little or no distribution divergence between the two classes. Clearly, in the X input domain, the two classes within the training set cannot be separated by a single line (i.e. a hyperplane in $\mathbb{R}^2$). However, treating the elements of X as rectangular coordinate pairs, we can transform each element into an $\mathbb{R}^2$ domain *V,* wherein each element in X is represented by the principal polar coordinate equivalent element in V:

$$\Phi : X \rightarrow V$$

$$x \mapsto v := \Phi(x) = \Phi(\begin{bmatrix} a \\ b \end{bmatrix}) = \begin{bmatrix} \sqrt{a^2+b^2} \\ \arctan\left(\frac{b}{a}\right) \end{bmatrix} = \begin{bmatrix} \|x\| \\ \arg(x) \end{bmatrix},$$

where $\arg(x) \in (-\pi, \pi]$. Mapping the training set S to the V domain results in the following training set T, which is depicted in Figure 2:

$$
\begin{aligned}
T &= \{(v_1, y_1), \ldots, (v_8, y_8)\} \\
&= \left\{ \left(\begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 0.5 \\ \frac{\pi}{2} \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 0.5 \\ \pi \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 0.5 \\ -\frac{\pi}{2} \end{bmatrix}, 1 \right), \right. \\
&\quad \left. \left(\begin{bmatrix} 1.5 \\ 0 \end{bmatrix}, -1 \right), \left(\begin{bmatrix} 1.5 \\ \frac{\pi}{2} \end{bmatrix}, -1 \right), \left(\begin{bmatrix} 1.5 \\ \pi \end{bmatrix}, -1 \right), \left(\begin{bmatrix} 1.5 \\ -\frac{\pi}{2} \end{bmatrix}, -1 \right) \right\}
\end{aligned}
$$

In the V domain, the training set T (which is a non-linear transformation of S) is now linearly separable. The estimated Henze-Penrose divergence between the two classes (based on FR statistics) is 0.500, signaling little distribution divergence between the two classes in the V domain. However, if the aforementioned vectors in the V domain were to be statistically normalized on an element-by-element basis across the training sets, then the Henze-Penrose divergence estimate would become 0.875, signaling significant divergence between the two classes. In the X domain, statistical normalization would not change the divergence estimate from its initial value of 0.500. Applying the criteria of equation 3.4, the optimal separating hyperplane in the V domain can be described by the set $\{v : \langle w_v, v \rangle + b = 0\}$ with weight vector

$$
w_v = \begin{bmatrix} -1 \\ 0 \end{bmatrix}
$$

and offset $b = 1$. Formally, w_v can be determined using Equations 3.19 and 3.20 (with the elements v_i instead of x_i and setting $\eta_i = 0$) to determine the coefficients a_i and then using the relation from equation 3.10, so that $w_v = \sum_{i=1}^{8} a_i y_i v_i$. Based upon equation 3.2, the classification of a new test input vector x_n is determined (with appropriate choice of b_v) by the decision function:

$$y_t = f(x_n) = \text{sgn}(\langle w_v, \Phi(x_n)\rangle + b_v) = \text{sgn}\left(\left(\sum_{i=1}^{8} a_i y_i \langle \Phi(x_i), \Phi(x_n)\rangle\right) + b_v\right)$$

Note that the classification of the new test vector x_n does not require explicit knowledge of w_v if the inner products of $\Phi(x_n)$ with the elements of the training set T are known. Further note that the inner product terms $\langle \Phi(x_i), \Phi(x_n)\rangle$ can be replaced by a kernel function $K(x_i, x_n)$, so that explicit mapping of input test vectors into the V domain (or feature space) in not required in order to classify new input test vectors:

$$K : X \times X \quad \rightarrow \quad \mathbb{R}$$

$$(x, y) \mapsto r := K(x, y) = \langle \Phi(x), \Phi(y)\rangle = \|x\| \, \|y\| + \arg(x) \arg(y) \tag{6.2}$$

Hence,

$$y_t = f(x_n) = \text{sgn}\left(\left(\sum_{i=1}^{8} a_i y_i K(x_i, x_n)\right) + b_v\right) \tag{6.3}$$

$K(x_i, x_j)$ can also be used to replace the inner product terms in Equation 3.19, which is used to determine the coefficients a_i associated with the separating hyperplane in the feature space. This substitution of the inner products of elements from a feature space with a kernel function on elements of the related input space is known as the "kernel trick" [27, p. 15]. The use of kernels enables the application of hyperplane classifiers to training sets that are not linearly separable in their native or input domain. In some instances, multiple kernels can be combined to effect non-linear separation manifolds for specific sets of input data.

Deriving Kernels from Data

Given a training set for a binary classifier, inferred differences in the characteristics of the two classes of data within the training set can be used as an aid in determining a

suitable capacity-limited set of decision functions available to the learning algorithm. For the support vector machine (hyperplane classifier), the choice of kernel determines the ability of the associated learning machine to classify the training set samples and to generalize in classifying new test samples. In the previous section, the visually-evident structure of the training set S was used to choose apriori a transform function and associated kernel function.[2] The two classes of input samples could be discriminated based on the distance of the sample from the origin without regard to angle. Therefore, another workable choice of kernel would have been the Gaussian kernel or radial basis function (RBF) kernel [32, p.77]:

$$K : X \times X \quad \rightarrow \quad \mathbb{R}$$

$$(x,y) \mapsto r := K(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \tag{6.4}$$

If we define a transform Φ to another feature space as $\Phi(x) = \|x\|$, then yet a third choice of kernel that would result in errorless training is

$$K(x,y) = \langle \Phi(x), \Phi(y) \rangle = \|x\| \, \|y\| \, . \tag{6.5}$$

While the training data serves to constrain the choice of kernels that will result in successful classification, it does not necessarily uniquely determine such a kernel. In other words, the selection of a kernel or set of candidate kernels is a *choice* of an expected solution space for the binary classification problem under consideration and should take advantage of our prior knowledge of the associated datasets.[37, p.73][27, p.407][32, p.72] Even after exposition of kernels based on generative probability

[2]The assumption, of course, is that the training set, though having a very limited number of samples, contains fairly representative distributions from the two classes.

models, including the Fisher kernel, Schölkopf and Smola indicate that the choice of kernel cannot generally be determined solely on the basis of a finite dataset:

> ... the choice of kernel function is crucial in all kernel algorithms. The kernel constitutes prior knowledge that is available about a task, and its proper choice is thus crucial for success. Although the question of how to choose the best kernel for a given dataset is often posed, it has no good answer. Indeed, it is impossible to come up with the best kernel *on the basis of the dataset* — the kernel reflects prior knowledge, and the latter is, by definition, knowledge that is available *in addition* to the empirical observations. [27, p.423]

For some training sets, which may even be known to be somewhat separable based on class divergence measures (such as the Heine-Penrose divergence), the choice of suitable kernel is not always initially apparent, so it may be necessary to test a set of candidate kernels in order to determine the most viable individual or composite kernel for the classifier. [32, p.72] In this regard, it is useful to note that kernels can constructed from the linear combinations, products, polynomial functions and exponentials of other kernels. [27, pp.408-412][32, pp.75-77][13, pp.42-44]

If the input data is already embedded in a Hilbert space, another method of effecting an implicit selection of a kernel function is to transform each input vector to a composite feature space by augmenting the vector with transforms of the baseline input vector. For example, in the toy problem of section 6.3, the two-element vectors of training set S could be transformed into four-element vectors to produce a new training set R in feature domain U, where the added elements correspond to the polar coordinates of the the first two elements:

$$
\begin{aligned}
R &= \left\{ \begin{pmatrix} x_1 \\ v_1 \end{pmatrix}, y_1 \right), \ldots, \begin{pmatrix} x_8 \\ v_8 \end{pmatrix}, y_8 \right) \Bigg\} \\
&= \left\{ \left(\begin{bmatrix} 0.5 \\ 0 \\ 0.5 \\ 0 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 0 \\ 0.5 \\ 0.5 \\ \frac{\pi}{2} \end{bmatrix}, 1 \right), \left(\begin{bmatrix} -0.5 \\ 0 \\ 0.5 \\ \pi \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 0 \\ -0.5 \\ 0.5 \\ -\frac{\pi}{2} \end{bmatrix}, 1 \right), \right. \\
&\quad \left. \left(\begin{bmatrix} 1.5 \\ 0 \\ 1.5 \\ 0 \end{bmatrix}, -1 \right), \left(\begin{bmatrix} 0 \\ 1.5 \\ 1.5 \\ \frac{\pi}{2} \end{bmatrix}, -1 \right), \left(\begin{bmatrix} -1.5 \\ 0 \\ 1.5 \\ \pi \end{bmatrix}, -1 \right), \left(\begin{bmatrix} 0 \\ -1.5 \\ 1.5 \\ -\frac{\pi}{2} \end{bmatrix}, -1 \right) \right\}
\end{aligned}
$$

In this case, the class samples are linearly separable in the feature domain U. Again applying the criteria of equation 3.4, the optimal separating hyperplane in domain U can now be described by the set $\{u : \langle w_u, u \rangle + b = 0\}$ with weight vector $w_u = [0,0,-1,0]^T$ and offset $b = 1$. Noting that the weight vector elements are zero except for the third element, the choice of kernel could be reduced to that of equation 6.5 corresponding to the one-dimensional feature space determined by the transform $\Phi(x) = \|x\|$.

In summary, two of the ways in which the selection of a suitable kernel for a particular binary hyperplane classification problem can be accomplished are (1) by direct trial of one or more candidate kernel functions, or (2) by element extension (or reduction) of input training vectors, where the extension elements are determined using transforms into kernel domains having tractable transform functions. Under both approaches, the initial selection of kernels or transform functions is assisted by available apriori knowledge of the classification problem and its associated dataset. Under the element extension (vector augmentation) approach, the weight vector obtained via a support vector machine can subsequently be used to determine which of several candidate kernel functions (or transforms) can best be used to classify the training set(s).

Application of Non-linear Kernel Techniques to a Statistically Controlled Dataset

In this section, we provide an example application of non-linear kernels to one of the datasets used in chapter five. For convenience, we include here a copy of the summary table from Case Study A.[3] This application example will exhibit the use of both vector element expansion and reduction (i.e. transformation to a lower-dimensional feature space) in improving the test performance of the hyperplane classifier.

To apply the non-linear kernel techniques discussed above to the dataset of Case Study A, each normalized 44-element input vector was extended by adding 3 elements comprised of the standard deviation, vector length, and skew of the forty-four (44) elements of the subject vector. These added elements were then normalized by subtracting the element's ensemble mean taken over the 861 input vectors and scaling the result by the element ensemble standard deviation. The vectors selected for the training set were those used in experiment A1 now extended by the three added elements. The weighting vector obtained from the training set was used to determine which of the vector elements were more significant with respect to separability of the class samples. By zeroing out some vector elements corresponding to the smaller weighting factors and retraining the classifier, it was found that the classification error on the test set could be improved. By iterative experimentation, the 47-element vector was able to be reduced to a 15-element vector (including the 3 added elements)[4] that resulted in a Type I (false positive) test error of 0.310 and a

[3]Recall the for experiments A1, A2, and A3, the trainings and tests were based on detection of membership in Group A vs. Group B. For experiments A4, A5, and A6, the trainings and tests were based on detection of failed vs. non-failed units.

[4]Note that this reduction in vector dimension is effectively the same (in the context of the hyperplane classifier) as multiplying each 47-element input vector by a transform matrix consisting of a 47-by-47 identity matrix with all except selected diagonal elements set to zero. This type of "element selection" matrix is a subset of the broader class of non-linear vector transform matrices.

Experiment	Training Vectors	Training Error	Test Vectors	Type I Error (α)	Type II Error (β)	Detection Rate ($1-\beta$)
A1	16	0	845	0.313	0.237	0.763
A2	16	0	845	0.437	0.165	0.835
A3	200	0.01	661	0.105	0.881	0.119
A4	200	0.065	661	0.188	0.417	0.583
A5	200	0.14	661	0.0192	0.861	0.139
A6	200	0.17	661	0.0096	0.917	0.083

Table 6.1: Recap of Results for Case Study A

detection rate of 0.875 (Type II error = 0.125). Subsequent removal of the 3 added elements results in a Type I test error of 0.348 and a detection rate of 0.908. Thus the reduction of the initial 44-element input vector to a particular subset consisting of 12 elements results in a 14.5% increase in detection rate but at the expense of a 3.5% increase in the Type I test error rate. Inclusion of the 3 added elements results in an 11.2% increase in detection rate (from that obtained in experiment A1) while maintaining approximately the same Type I test error rate (0.310 vs. 0.313).

Chapter 7

L-MOMENT KERNELS

7.1 Definitions and Derivations

Order statistics and L-statistics (including L-moments) were introduced earlier, in chapter 2. To recap, we suppose a set $\{X_i\}_{i=1}^n$ is a real-valued set of n iid samples of a random variable X, which has a cumulative distribution function (CDF) F

$$\begin{aligned} F &: \mathbb{R} \to [0,1] \\ x &\mapsto p := F(x) \equiv \Pr(X \leq x) \end{aligned} \tag{7.1}$$

Let the samples be sorted in ascending order and relabeled as $X_{k:n}, (k \in [1,\ldots,n])$, such that $X_{j:n} \leq X_{k:n}$ whenever $j < k$. Then the sample $X_{k,n}$ is called the k^{th} order statistic of this set of n sample elements. L-moments, so-called because they are derived from linear combinations of order statistics, are "difference moments", a family of which can be used to characterize any distribution with a finite mean [21, pp.194-195]. The r^{th} L-moment, λ_r, is the weighted expected value of the $(r-1)^{th}$difference of sets composed of r order statistics (i.e. an ordered set of r iid samples) of the subject random variable X. The weighting factor is equal to $\frac{1}{r}$. For example, λ_3, which is a measure of distribution skewness, is one-third ($\frac{1}{3}$) the expected value of the 2^{nd} difference of random sets composed of iid samples of X taken three (3) at a time:

$$\begin{aligned} \lambda_3 &= \tfrac{1}{3}E([X_{3:3} - X_{2:3}] - [X_{2:3} - X_{1:3}]) \\ &= \tfrac{1}{3}E(X_{3:3} - 2X_{2:3} + X_{1:3}). \end{aligned}$$

Similarly, λ_4, a measure of distribution peakedness (kurtosis), is one-fourth ($\frac{1}{4}$) the expected value of the 3^{rd} difference of random sets composed of iid samples of X taken four (4) at a time:

$$\begin{aligned} \lambda_4 &= \tfrac{1}{4}E([(X_{4:4} - X_{3:4}) - (X_{3:4} - X_{2:4})] - [(X_{3:4} - X_{2:4}) - (X_{2:4} - X_{1:4})]) \\ &= \tfrac{1}{4}E(X_{4:4} - 3X_{3:4} + 3X_{2:4} - X_{1:4}) \end{aligned}$$

In general, as stated in chapter 2, the r^{th} L-moment, λ_r, can be expressed in terms of expected values of order statistics as [20, p.106]:

$$\lambda_r \equiv \frac{1}{r}\sum_{k=0}^{r-1}(-1)^k \begin{pmatrix} r-1 \\ k \end{pmatrix} EX_{r-k:r}, \quad r = 1,2,...$$

Note that the distribution of the Gaussian random variable is uniquely determined by its first two L-moments just as it is by the first two classical moments. Given a normal random variable $Z \sim (\mu,\sigma^2)$:

$$\lambda_1 = E(Z_{1:1}) = E(Z) = \mu$$

$$\begin{aligned}
\lambda_2 &= \tfrac{1}{2}E\left(Z_{2:2} - Z_{1:2}\right) \\
&= \tfrac{1}{2}E\left([Z_{2:2} - \mu] - [Z_{1:2} - \mu]\right) \\
&= \tfrac{1}{2}\left(\tfrac{1}{2\pi\sigma^2}\right)\int\int |x-y| \exp\left(\tfrac{-x^2}{2\sigma^2}\right)\exp\left(\tfrac{-y^2}{2\sigma^2}\right) dx\,dy \\
&= \tfrac{1}{2}\left(\tfrac{1}{2\pi\sigma^2}\right)\left[\int_{-\infty}^{\infty}\int_{y}^{\infty}(x-y)\exp\left(\tfrac{-x^2}{2\sigma^2}\right)\exp\left(\tfrac{-y^2}{2\sigma^2}\right) dx\,dy\right. \\
&\quad \left. + \int_{-\infty}^{\infty}\int_{x}^{\infty}(y-x)\exp\left(\tfrac{-x^2}{2\sigma^2}\right)\exp\left(\tfrac{-y^2}{2\sigma^2}\right) dy\,dx\right] \\
&= \tfrac{1}{2\pi\sigma^2}\int_{-\infty}^{\infty}\int_{y}^{\infty}(x-y)\exp\left(\tfrac{-x^2}{2\sigma^2}\right)\exp\left(\tfrac{-y^2}{2\sigma^2}\right) dx\,dy \\
&= \tfrac{1}{2\pi\sigma^2}\int_{-\frac{3\pi}{4}}^{\frac{\pi}{4}}\int_{0}^{\infty}(r\cos\theta - r\sin\theta)\exp\left(\tfrac{-r^2\cos^2\theta}{2\sigma^2}\right)\exp\left(\tfrac{-r^2\sin^2\theta}{2\sigma^2}\right) r\,dr\,d\theta \\
&= \tfrac{1}{\sqrt{2\pi\sigma^2}}\int_{-\frac{3\pi}{4}}^{\frac{\pi}{4}}(\cos\theta - \sin\theta)\left[\tfrac{1}{\sqrt{2\pi\sigma^2}}\int_{0}^{\infty} r^2\exp\left(\tfrac{-r^2}{2\sigma^2}\right) dr\right] d\theta \\
&= \tfrac{1}{\sqrt{2\pi\sigma^2}}\left[\tfrac{\sigma^2}{2}\right]\int_{-\frac{3\pi}{4}}^{\frac{\pi}{4}}(\cos\theta - \sin\theta)\,d\theta \\
&= \tfrac{\sigma}{\sqrt{\pi}}
\end{aligned}$$

For other distributions, it may take several orders of L-moments to adequately characterize the distribution. However, for the purposes of detecting differences of statistical distribution between two classes (or the lack thereof), use of the first few L-moments may suffice. Given input vectors already embedded in $\mathbb{R}^n$ (i.e. vectors with real-valued elements) and composed of elements which have been

ensemble-normalized, we may treat the elements of each resultant vector as a set of random samples and extend the vector with estimates of the first few L-moments based on this set of vector elements. This will implicitly induce a modified kernel function within the hyperplane classifier as discussed in the previous chapter. If each input vector can be classified primarily or solely on the basis of its inter-element statistics, then we would expect that the statistical summary information reflected in the derived L-moment vector-extension sets could also be successfully used as a basis of classification decision. Note that inter-element correlation differences between two classes of input vectors may be detectable on the basis of vector element set statistics even when differences between the classes of vectors may elude detection on an element-by-element basis. In the next sections, we plan to discuss the estimation of L-moments and their use in detection of classes within statistically controlled datasets.

7.2 Estimation of L-moments from Sample Data

Since L-moments are by definition expected values of linear combinations of order statistics of iid random variables, a convenient estimate ($\widehat{\lambda}_{r,N}$) for a given L-moment (λ_r) is the average value of N instantiations of the combination of order statistics associated with the L-moment:[1]

$$\widehat{\lambda}_{r,N} \equiv \frac{1}{N}\sum_{j=1}^{N}\frac{1}{r}\sum_{k=0}^{r-1}(-1)^k \begin{pmatrix} r-1 \\ k \end{pmatrix} X_{r-k:r,j}, \quad N,r \in \{1,2,...\},$$

where the set $\left\{X_{r-k:r,j}\right\}_{k=0}^{r-1}$ is the j^{th} independent set of r iid random samples. If the random samples are drawn independently from a distribution with finite mean and variance, then $\widehat{\lambda}_{r,N}$ is a consistent estimate of λ_r:

[1]Note that the estimators described here are not the same as U-statistics alluded to Chapter 2 (Background and Literature Review) since a U-statistic estimate of λ_r requires averaging "over all subsamples of size r which can be constructed from the observed sample of size n" [20, p.113-114], where $n \geq r$.

$$
\begin{aligned}
E\left(\widehat{\lambda}_{r,N}\right) &= E\left(\frac{1}{N}\sum_{j=1}^{N}\frac{1}{r}\sum_{k=0}^{r-1}(-1)^k\binom{r-1}{k}X_{r-k:r,j}\right) \\
&= \frac{1}{N}\sum_{j=1}^{N}\frac{1}{r}\sum_{k=0}^{r-1}(-1)^k\binom{r-1}{k}EX_{r-k:r,j} \\
&= \lambda_r
\end{aligned}
$$

$$
\begin{aligned}
Var\left(\widehat{\lambda}_{r,N}\right) &= Var\left(\frac{1}{N}\sum_{j=1}^{N}\frac{1}{r}\sum_{k=0}^{r-1}(-1)^k\binom{r-1}{k}X_{r-k:r,j}\right) \\
&= \frac{1}{N^2}\sum_{j=1}^{N}\frac{1}{r^2}\sum_{k=0}^{r-1}(-1)^{2k}\binom{r-1}{k}^2 Var\left(X_{r-k:r,j}\right) \\
&= \frac{1}{N^2}\sum_{j=1}^{N}\frac{1}{r^2}\sum_{k=0}^{r-1}(-1)^{2k}\binom{r-1}{k}^2 Var\left(X_{r-k:r}\right) \\
&= \frac{1}{N^2}\sum_{j=1}^{N}Var\left(\frac{1}{r}\sum_{k=0}^{r-1}(-1)^k\binom{r-1}{k}X_{r-k:r}\right) \\
&= \frac{1}{N}Var\left(\lambda_r\right) \to 0 \text{ as } N \to \infty
\end{aligned}
$$

In using L-moment estimates to extend input vectors drawn from statistically controlled processes, our objective is determine whether the group of L-moment sets for one class of input vectors is separable from those of the opposite class. Hence, the concern for statistical independence of the vector elements from each other can be relaxed, especially given the working assumptions that the entire data set under consideration is both statistically controlled and has been statistically normalized prior to application of the hyperplane classifier.

7.3 Applying L-moment Kernels to Data

Given an input vector x with V elements ($V \geq 2$), let the vector-valued L-statistics function ϕ_m be defined by

$$\phi_m(x) \triangleq \left[\widehat{\lambda}_{1,V}(x), \widehat{\lambda}_{2,\lfloor V/2 \rfloor}(x), \ldots, \widehat{\lambda}_{m,\lfloor V/m \rfloor}(x)\right]^T,$$

where $1 \leq m \leq V$ and $\widehat{\lambda}_{r,N}(x)$ is the estimator $\widehat{\lambda}_{r,N}$ described in the previous section with the elements of x being treated as the set of data samples used for the L-moment estimation. Assume the data input vectors are defined on a $V-$dimensional real Hilbert space X. Then for $x, y \varepsilon X$, a kernel function K can be defined on X as an inner product in terms of ϕ_m:

$$K(x,y) = \langle \phi_m(x), \phi_m(y) \rangle = \sum_{k=1}^{m} \left(\widehat{\lambda}_{k,\lfloor V/k \rfloor}(x)\right)\left(\widehat{\lambda}_{k,\lfloor V/k \rfloor}(y)\right).$$

If we choose to use the m-dimensional feature space $\oplus$ induced by ϕ_m as the domain of a hyperplane classifier, then this kernel function would be used in equation 3.19 in place of the inner product terms shown therein in order to solve for the coefficients a_i (see equation 3.20). These coefficients are then used in equation 3.10 to solve for the normal vector (or weight vector) w. Alternately, as was illustrated earlier, we may choose to use $\phi_m(x)$ to extend the input vector x or to extend another vector-valued function of x (such as a statistical normalization function based on a specific subset of available population data). Of course, in this case, K would need to be defined in terms of functions of the new extended vector.

For arbitrary vector structures, the usefulness of the L-statistic function as the basis of a discriminator may be affected (positively or negatively) by the colinear resolution and vector element scaling anomalies discussed in Chapter 4. To avoid these effects for the data under consideration in this research, we choose to statistically normalize the input vectors on an element-by-element basis by subtracting from each vector element an ensemble mean and scaling the result by the ensemble standard

deviation, where the ensemble statistics have been determined from a suitably large, or at least representative, sample set of data from a statistically controlled process. After determination of the L-statistic estimates for a set of input vectors, these statistics themselves can be ensemble normalized on an element-by-element basis prior to use in hyperplane classification training, especially if colinear resolution or vector element scaling error effects would result from use of the native estimate sets.

The L-statistics function serves to summarize the statistical characteristics of the input vector. In some cases, this summary may elucidate a characteristic difference that can be used to discriminate between the data from two classes. In other cases, this summary may result in the loss of important order-dependent information that could be easily used to determine the input vector class.

For example, suppose the vectors in one class each consists of ten iid Gaussian random elements with mean 0 and variance π, while the vectors in the other class each consists of ten copies of one instantiation of a Gaussian random variable with mean 0 and variance π. On an element-by-element basis, the expected values of mean and variance are the same for both classes. For the first class, the expected values of the first two L-statistics of a member vector are 0 and 1, respectively. For the other class, these expected values are both 0. Hence, the L-statistic function would, in this case, serve as a good feature detector on which to base a binary classifier, even for low sample sizes.

Now, suppose both classes consist of iid Gaussian random elements with variance π and means alternating from -1 to 1 between elements with one class having the first element's mean equal to 1 while the first element of the other class has mean equal to -1. In this case, the expected values of the first two L-statistics of a member vector for either class are the same by construction, but the classes could be distinquished with high probability using the dot product of each input vector with the weighting vector $[1,-1,1,-1,1,-1,1,-1,1,-1]^T$.

Note that for both example cases just given, using the L-moment estimates to extend (rather than replace) the feature vector would still allow high probability class detection if less weight is assigned in the decision function to the set of vector elements that provide little or no discrimination information for the dataset of interest. This weighting assignment is an automatic feature of the SVM approach to training the hyperplane classifier and is to be further explored with respect to L-statistics functions in the next section.

7.4 SVM's and L-moments

First, a brief review. In training the SVM or hyperplane classifer on vector-valued samples within a real Hilbert space X, we define a hyperplane $h_{w,b}$ that separates X into two half-spaces, where $h_{w,b} = \{x : \langle w,x\rangle + b = 0\}$, $w \in X$, and $b \in \mathbb{R}$. By construction, w is a normal vector of the hyperplane and $\frac{||b||}{||w||}$ is the distance of the hyperplane from the origin (zero vector) in X. A classification function $f_{w,b}(x)$ assigns a label $y_t \in \{-1,1\}$ to each test vector x_t based on which of the two subspaces contain x_t:

$$y_t = f_{w,b}(x_t) = \text{sgn}(\langle w,x_t\rangle + b)\,,$$

where

$$\text{sgn}(r) \equiv \begin{cases} 1, & \text{if } r \geq 0 \\ -1, & \text{if } r < 0 \end{cases}, \quad r \in \mathbb{R}\,.$$

Given a training set of m samples (with at least one sample from each class), the hyperplane parameters (w,b) are chosen to minimize the empirical risk (or average training error rate [27, p.8][14, p.123]):

$$R_{\text{emp}}\ \left[f_{w,b}\right] = \frac{1}{m}\sum_{i=1}^{m} 0.5\left|y_i - f_{w,b}(x_i)\right|.$$

As we have seen earlier, in cases where the training set is not linearly separable, a suitable kernel function, $K(w,x_t)$ used in place of the inner product $\langle w,x_t\rangle$ may result

in a decision function with a reduced minimum empirical risk. In this case, the separation hyperplane is in some transformed domain of X.

In the decision function $f_{w,b}(x)$, w acts a weighting vector and b as an offset. Intuitively, a component (or vector-element) of the training vectors that are not separable between classes based on that component should have less influence on the classification decision than a component for which the training set is separable, assuming the vector components are independent. From an empirical risk perspective, reducing the weighting factor on the non-separable vector-element and increasing the weighting factor on the separable vector element tends to decrease the risk of misclassification. In cases where there is virtually no class divergence of values for one component within the training set, one would expect that a relatively low or even zero weight for that component if significant divergence exists for one or more other components of the training vectors.

As implemented, the modified SVM used in this research tends to place greater weight on those vector elements have greater influence on the hyperplane classifier's ability to separate the training set vectors. More specifically, assuming the input data has been statistical normalized on an element-by-element basis, the elements of the weight vector generated on the basis of a specific set of training vectors will have greater numerical values at those element positions associated with the more separable training set vector elements. In some cases, this allowed us to reduce the number of vector elements (components) used in the classifier. However, as was illustrated in the previous section, some component correlations can be missed by application of the SVM or hyperplane classifier to the input vector space alone. L-moment statistics on the statistically normalized dataset can be used to test whether vector element statistics differ between the two classes of the training vectors. By using L-moment statistics to extend the input vectors, classification information that may be contained within the initial vector element set is retained while possibly providing additional information

now discernable to the classifier. As the resultant set of vectors is used to train the hyperplane classifier, the weight vector w can be used to indicate the relative contribution of the extension elements to the power of the classifier. If the larger weighting factors correspond to the one or more of the L-moment statistical elements, then that element might be beneficially included in a reduced set of vector components to be used by the classifer. In data that is separable based purely on vector element L-moment summary statistics, the L-moment statistical elements might even be used in place of the initial input vectors in training the SVM. In this case, the input vectors would have effectively each been transformed into an "L-moment" Hilbert space prior to the hyperplane classification process or, equivalently, an L-moment kernel would have been utilized in the decision function. In cases where the classes are not generally separable, the SVM misclassification rate on test sets could be used to compare the relative effectiveness of utilizing the SVM with various combinations of the identity kernel (initial input space), the L-moment kernel, or some other kernels.

Use of L-moment statistics with the statistically controlled datasets we have studied may also serve to extend the power of the hyperplane classifier while still minimizing the risk of overfitting the decision function to a specific training set (i.e. training the classifier to correctly classify a specific set in a manner that might not generalize well to a subsequent test set of samples. To find a classifier that is not overfit to a specific training set, we must make trade-off decisions between the discrimination power of the classification function family (the learning machine) and its generalization ability. For the supervised subgroup classification (or detection) problem under study, an additional decision criterion might prove useful, namely, the statistical likelihood that the selected subgroup represents simply a random sampling of the total training set or, in information theoretic terms, the conditional entropy of the selected subset (or subgroup) given the entire training set. If a selected or pre-classified subgroup can be shown, with some given confidence level, to

significantly differ in distribution from the total training set, then it is likely that the trained classifier machine will provide significant detection or classification information when applied to future test sets. Otherwise, even though the classifier may appear to classify or segment the input data, it may not be actually "detecting" a process variant subset, but be simply providing an artificial segmentation of the data. Some conditional entropy or statistical significance test may be used in this case to determine whether a labeled training set is admissible with respect to training the learning machine to provide informative (statistically significant) classification or detection. The use of L-moments to preprocess input distributions of test error results enables the use of the SVM misclassification rate to simultaneously provide a measure of the separability of the training data and its statistical information content .

In the next section, L-moment kernels are applied to the manufacturing test datasets studied in Chapter 5.

7.5 Application of L-Moment Kernels to Case Studies

As an initial approach in exploring the effect of applying L-moment kernels to the case studies of Chapter 5, the data vectors for these datasets were extended using estimates of up to five L-moments, the L-coeficient of variation (τ) and L-moment ratios τ_3 (and/or τ_4). For each of the case studies (Cases A, B, and C), the effect of extending the statiscally normalized data vectors, including the training vectors, was observed by revisiting the first experiment listed for the case (i.e. experiments A1, B1, and C1). Recall that the Heinze-Penrose divergence estimates for Cases A, B, and C averaged 0.884, 0.541, and 0.905, respectively. Thus, the datasets of Cases A and C had exhibited significant class divergence, with some overlap; while the dataset of Case B exhibits little class divergence based on a geometric or spatial measure.

For Case B (Experiment B1), extension of the dataset vectors with estimates of λ_1,λ_2, and τ resulted in increasing the SVM detection rate of the twenty-seven (27)

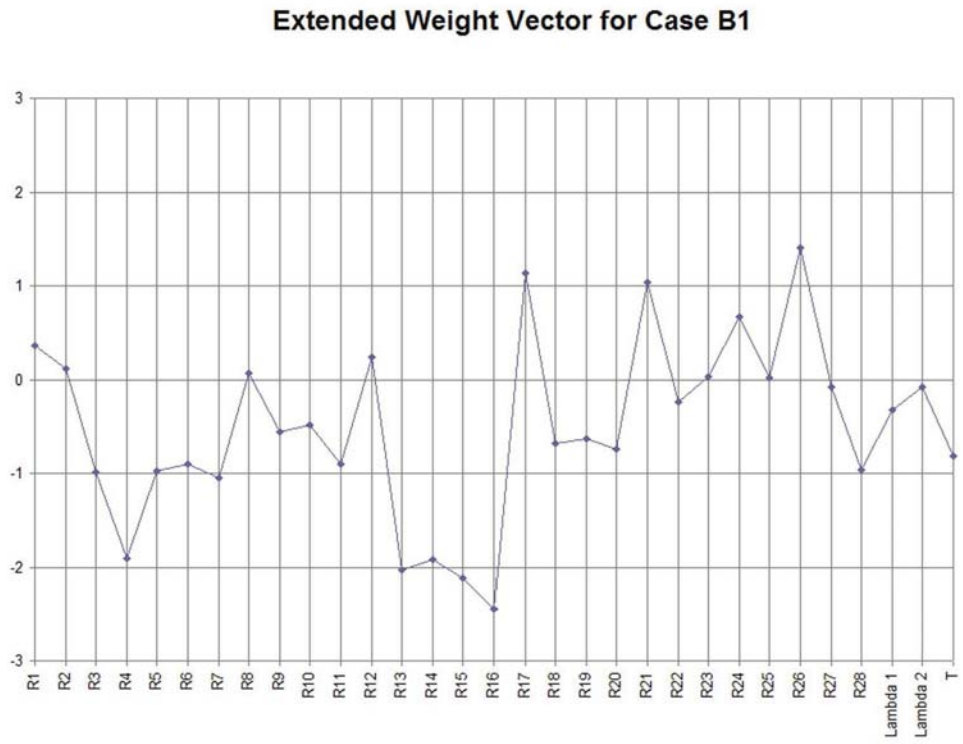

Figure 7.1: Extended Weight Vector for Experiment B1

factory failed units from 29.6% to 40.7%. However, the false positive rate for the non-failed test group (125 units) also increased, from 51.2% to 60.8%. Figure 7.1 depicts the extended weight vector. Based on the weights assigned to the vector elements (λ_1,λ_2, and τ), it is evident that these additional elements provide some but not an overriding influence on the SVM classification of the dataset, with the L-coefficient of variation (τ) carrying the greatest weight of the three.

For both Case A and Case C, the vectors were extended with estimates of the set $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \tau\}$. Case A is additionally extended with estimates of τ_3 and τ_4.

For Case A, the resultant false positive rate was 27.8%, a decrease of 5.5%, with the trade-off that the detection rate fell from 75.7% to 69.7%. The extended

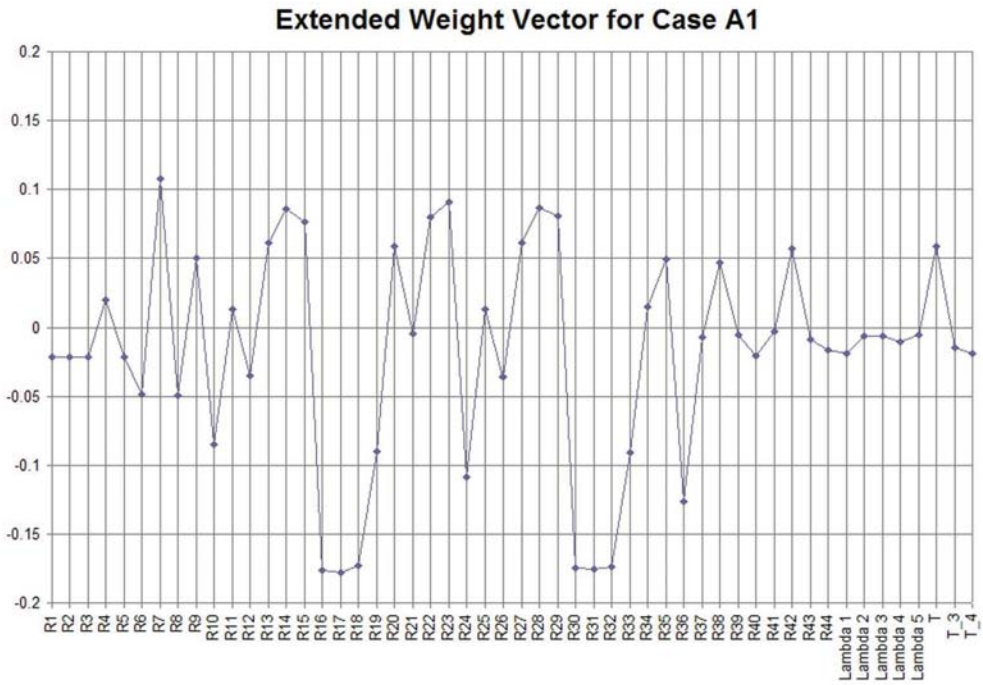

Figure 7.2: Extended Weight Vector for Experiment A1

weight vector is shown in Figure 7.2. τ appeared to be the most influential of the extension elements, but was significantly less influential than other vector elements with respect to affecting the decision outcomes of the SVM classifier.

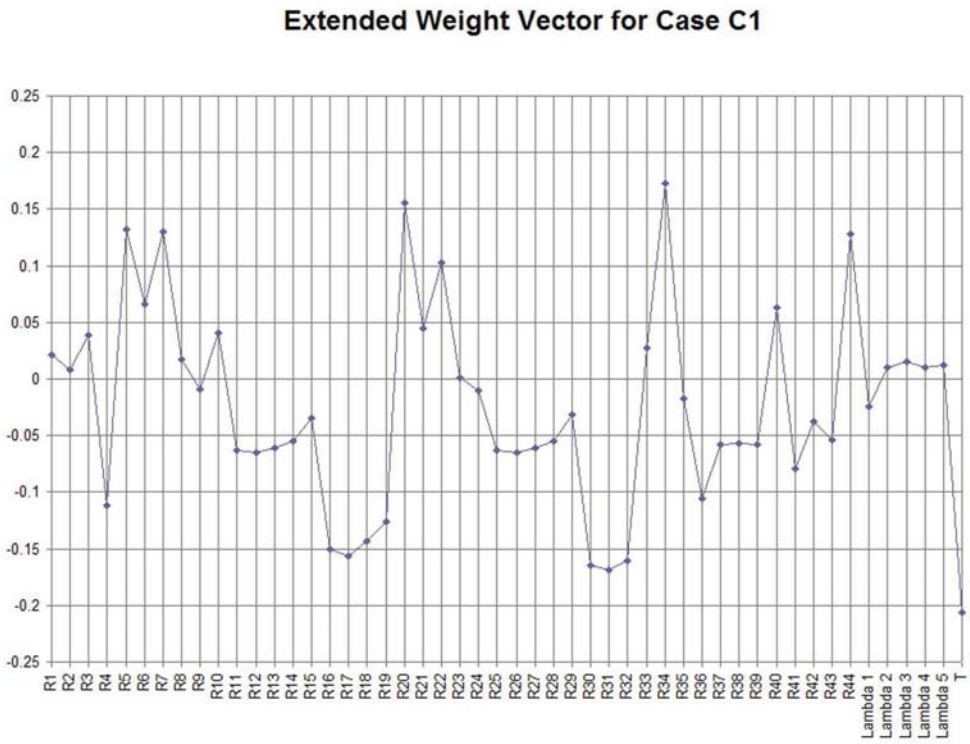

Figure 7.3: Extended Weight Vector for Experiment C1

For Case C, the false positive rate decreased from 29.7% to 25.4%, but the detection rate changed from 94.8% to 88.1%, an decrease of 6.7%. Figure 7.3 displays the extended weight vector for Case C. In this case, τ was assigned the greatest weight among both the original and extension elements.

Note that in all three of these particular cases, the effect of extending the dataset vectors using estimates of L-moments resulted in a tradeoff between the false positive rate and the detection rate. Therefore, the choice of retaining this extension in practice would be dependent on whether the desired end effect depended more heavily on avoidance of false rejections or avoidance of escapes (false acceptances).

Chapter 8

SVM IMPLEMENTATION OF WESTERN ELECTRIC COMPANY RULES

8.1 Definitions

Given a series of real-valued outputs from a statistical process, the Western Electric Company (WECO) rules are a set of decision rules that serve to identify low probability events and data sequences whose occurence may indicate that the process is exhibiting a non-random effect or special cause. Under the classical WECO rules an "out of control" alarm condition is set when at least one of the following conditions occurs within the data stream [1, Section 6.3.2]:

1. Any point is outside the $\pm 3\sigma$ control limits.
2. At least 2 of 3 consecutive points lie on the same side of the mean in a region that is greater than 2σ away from the estimated process mean.
3. At least 4 of 5 consecutive points lie on the same side of the mean in a region that is greater than 1σ away from the estimated process mean.
4. 8 or more consecutive points lie on the same side of the mean.
5. 6 or more consecutive points trend up or down.
6. 14 or more consecutive points alternate up and down.

8.2 Using the Modified SVM Construction to Utilize WECO Rules

Each of these WECO alarm conditions require tests on an indexed set (or series) of real-valued data points $x[i],\ i \in \mathbb{N}_0$, to determine whether the condition has been met. Suppose a subset of such data points is organized into a set of indexed vectors S_k, each

of length m, with $m \geq 14$:[1]

$$S_k = \{x[k \cdot m + j]\}_{j=0}^{m-1},\ k \in \mathbb{N}_0$$

Let V_k be a vector of length six (6), consisting of the outcomes of six indicator functions, one for each WECO rule, operating on the associated input vector S_k:

$$V_k = \{v_n(S_k)\}_{n=1}^{6}$$

$$v_n : S_k \mapsto [0,1]$$

The outcome of v_n is "1" when the associated WECO alarm condition is true (i.e. the WECO rule is violated).

A direct approach to implementing the WECO alarm conditions with an SVM construction is to extend or replace each input vector S_k with the vector V_k. The SVM weight vector and offset could then be assigned to yield a positive (or negative) result when any WECO rule was violated and a negative (or positive) result otherwise. For example, if the weight vector W is chosen to be a vector of six ones and the offset $b = -0.5$, then the quantity $d_k = \langle W, V_k \rangle + b$ will be greater than 0 if any alarm condition is true (set) and equal to b, otherwise. One obvious drawback to this approach is that since each indicator function v_n already operates on an input vector to indicate whether the associated rule is violated, the further complication of extending or replacing a vector in an SVM is superfluous if the end goal is simply to determine whether an alarm condition has occurred. However, in a manufacturing situation, the parameter data being analyzed may be associated with a larger set of parameters and indicators. It may be useful not only to know that some WECO condition was violated but also to know whether particular combinations of regular or intermittent WECO

[1]Since m is finite, there is always the chance of missing an alarm condition that occurs across the artificially imposed vector endpoints. However, as m increases, the likelihood of missing an alarm condition due to this anomaly decreases.

alarm conditions are correlated to classifiable conditions of other associated parameters or indicators. In this case, the SVM permits a convenient construction if the problem can be resolved by a two-class discriminator or a series of two-class discriminators. A extension of the SVM input vector with WECO condition indicators (on a portion of the vector that represents serial process data) can thus be used in much the same manner earlier proposed for L-moment estimators—as a means of providing additional information about the input vector that might enhance the classification power of the SVM.

If the objective of utilizing WECO conditions is to extend the information content of a vector or subvector consisting of process data, then another though less direct approach is to use extension vectors consisting of functions of selected order statistics on the process data stream. The particular choice of order statistics is based on their relation (or pseudo-relation) with the WECO conditions.

Let a set of SVM training vectors each consist of m iid samples of time-ordered process data (where $m \geq 14$), possibly with additional vector elements representing other real-valued parameters. Further assume that the training vectors have been statistically normalized. In this manner, process mean and sigma information are then incorporated (on an element-by-element basis) into the SVM training vectors. As will be outlined below, information similar to that provided by the WECO conditions (as enumerated in the previous section) can be included with the SVM by extending each input vector X with a particular set of order statistics. For ease of explanation, we assume below that X consists only of the m elements of iid samples. If the vector contains additional elements, the discussion applies to the subvector consisting of said iid samples.

WECO Condition 1 can be determined using the information provided by $X_{m,m}$ and $X_{1,m}$, which are the maximum and minimum values of the vector elements of X. Since the data have been normalized to have a standard deviation of the estimated

process sigma and an offset of zero, WECO Condition 1 will be positive (true) when either $X_{m,m} > 3$ or $X_{1,m} < -3$.

WECO Condition 2 can be determined using the maximum and minimum values of the order statistic $X_{2,3}$ taken over each of the $m-2$ three-element sets of consecutive data in X. If $X_{2,3} > 2$, then $X_{3,3} > 2$ and the condition is true. The condition is also true if $X_{2,3} < -2$, since this implies that $X_{1,3} < -2$.

Similarly, WECO Condition 3 is informationally related to the maximum value of $X_{2,5}$ and the minimum value of $X_{4,5}$, both taken over each of the $m-4$ five-element sets of consecutive data in X. Condition 3 is true when $X_{2,5} > 1$ or when $X_{4,5} < -1$.

WECO Condition 4 can be detected using the maximum value of $X_{1,8}$ and the minimum value of $X_{8,8}$ each taken over the $m-7$ eight-element sets of consecutive data in X. Condition 4 is true if $X_{1,8} > 0$ or $X_{8,8} < 0$.

For WECO Condition 5 we begin by generating a vector Y of length $m-1$ that consists of the first difference of adjacent elements of the input vector X. Then Condition 5 can be detected using the maximum value of $Y_{1,5}$ and the minimum value of $Y_{5,5}$ taken over each of the $m-5$ five-element sets of consecutive data in Y. To add the WECO Condition 5 information to the input vector, the maximum and minimum values of $Y_{1,5}$ and $Y_{5,5}$, respectively, are used to extend the input vector X. If $Y_{1,5} > 0$ or $Y_{5,5} < 0$, then Condition 5 is true.

WECO Condition 6 can be detected as follows. First, generate a first difference vector Y as described for Condition 5. Next, for each thirteen-element sub-vector Z of consecutive data in Y, determine the maximum value of $Z_{1,2}$ and the minimum value of $Z_{2,2}$ taken over each of the twelve (12) two-element sets of consecutive data in the sub-vector. Condition 6 is true if, over any thirteen-element sub-vector Z of Y, both of the following conditions are true: (a) the maximum value of $Z_{1,2} < 0$ AND (b) the minimum value of $Z_{2,2} > 0$. To add WECO Condition 6 information to an input vector

X, the following derived quantities are used to augment the elements of X:

$$\min_{Z \in Y} \max\left(Z_{1,2}\right)$$

$$\max_{Z \in Y} \min\left(Z_{2,2}\right)$$

Note that each of the six WECO Conditions can be determined based on functions of two order statistics on the input vector X or its first difference vector Y. By using these twelve derived values as extension elements to each vector X, the informational content of the WECO Conditions can be included in the SVM classifier for use, for example, in detecting classification differences of similar process data generated from several manufacturing test stands. In this case, while the vector extensions could be readily used to determine whether process alarms should be set, the extension of the SVM input vector with WECO Condition-related information is accomplished with variable data rather than attribute or discrete data. This may serve to increase the classification power of the SVM in instances where the process vectors generated from a particular source are exhibiting non-random behavior that can be detected as belonging to a different class than the central group of sources even though the deviation has not yet reached the point of setting a WECO condition alarm.

8.3 Effects of Extending the SVM Input Vectors with WECO Conditions

Since the WECO condition extensions to an input are based on order statistics of that vector, we would expect that the extensions would add little or only incidental (or overfitted) classification information to the SVM if the vectors from two classes are randomly drawn from the same stationary source. In this case, the expected values of the extension elements are the same for both classes. On the other hand, if the classes are statistically divergent, the group of order statistic functions represented by the extensions would be expected to display divergence as well.

As an experiment, WECO condition extensions were used to augment the vectors of Case Study C in chapter 5. Recall that of the three case studies, Case Study C had the largest estimate of Henze-Penrose divergence (0.905 compared to 0.815 and 0.541 for Case Studies A and B). Without the extensions, the detection rate was 94.8% with a false positive rate of 29.7%. With WECO condition extensions added to each vector, the resultant SVM detection rate increased to 97.8 percent, the tradeoff being an increase in the false positive rate to 38.0%. Next, the WECO conditions extensions were statistically normalized on an element-by-element ensemble basis to have a zero mean and unit variance. This time the resultant SVM detection rate remained at 94.8% and the false positive rate fell to 28.4%, a decrease of 1.3%.

The vectors in the Case Studies A, B, and C of Chapter 5 are non-homogenous in the sense that they are each comprised of elements that represent different parameters. Some of the parameters are correlated to each other while others are independent. In the classical paradigm, the WECO conditions are typically applied to homogeneous data streams or sets of homogenous data vectors in which the elements of each vector are instantiations of the same parameter. As in our case studies, the elements of each vector may or may not be correlated with other elements of the vector. In fact, due to statistical normalization, the modified input vectors of the case studies somewhat resemble data vectors of random elements with correlation between some of the elements. For homogenous data streams generated by processes within statistical control, the colinearity issue, discussed in Chapter 4, is intrinsically addressed. In the case of the non-homogenous data vector, including that produced by augmentation with WECO condition elements, application of statistical normalization seems to enhance the performance of the SVM classifier by not prematurely over or under-emphasizing the classification contribution of particular input vector elements. In the next section, we explore the application of the SVM and WECO condition extensions to a fourth case study involving homogenous data streams of the same

measurement parameter from differing manufacturing builds.

8.4 Case Study D: Homogenous Data Streams

A new supplier (Supplier B) was chosen to provide a special functional part of a particular sensor type. The new supplier utilized some different materials and deposition methods than the previous supplier (Supplier A), but the project goal was that the change should result in no detrimental impact to the performance of the manufactured sensor when it was installed into the end-item measurement device (transducer). The input data for this case study are a statistically normalized (i.e. adjusted) set of 800 vectors (one per sensor/transducer) each having 44 elements. Each vector element is a real number corresponding to the adjusted measurement error of the transducer at a particular applied source stimulus level and temperature. Both the training set (16 vectors) and the test set (784 vectors) had vectors whose elements were each within 4.7 sigma of the its element mean. Of these 800 vectors, 773 vectors consisted of elements all within 3.0 sigma of their element means. Let the measurement vectors of sensors manufactured with a part from Supplier A be designated Class A and considered the negative samples. Let the remaining vectors be designated Class B and considered to be the positive samples. One-hundred (100) of the input vectors were from Class B; seven-hundred (700) vectors were from Class A.

The Henze-Penrose divergence of this study set was estimated through use of minimum spanning trees on groups of 200 vectors (100 from each class) to determine Friedman-Rafsky (FR) statistics. The range of seven (7) such estimates, encompassing the entire study set, was from 0.530 to 0.920 averaging 0.694 with a standard deviation of 0.133. The estimated Henze-Penrose divergence average ($\widehat{HP}_{avg}$) for this study set is close to 0.7, suggesting that there is some distribution divergence between the two classes, though the standard deviation and relatively wide range for the 7 estimates hints that that the overall distribution for the Class A vectors may be a

composite mixture of several distributions.

Each of the 800 vectors were extended with a 21-element set consisting of estimates of L-statistics $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \tau, \tau_3, \tau_4, \tau_5)$ and the twelve WECO condition elements, as defined previously. The 21 elements were then statistically normalized to each have zero mean and unit variance across the 800-sample set. Using the same 100-member groupings of vectors described above, $\widehat{HP}_{avg}$ was estimated for various subvectors of the extended vector set. With all 65 elements (Data + Full Extension) included for each vector, $\widehat{HP}_{avg} = 0.702$ with a standard deviation $\sigma_{HP} = 0.122$. Replacing each original 44-element input vector with only its 21-element extension (Full Extension) results in $\widehat{HP}_{avg} = 0.661$ with a standard deviation $\sigma_{HP} = 0.141$. Other vector subsets for which the Henze-Penrose divergence was calculated include

- L-moments $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)$
- L-moment Ratios $(\tau, \tau_3, \tau_4, \tau_5)$, including the L-coefficient of variation τ.
- L-statistics $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \tau, \tau_3, \tau_4, \tau_5)$
- WECO Information (12-element extension vector), as defined in Section 8.2
- Data + WECO Information
- Data + L-moments $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)$

The Henze-Penrose divergence estimates for all nine aforementioned vector subsets is shown in Table 8.1.

Based on the tabulated $\widehat{HP}_{avg}$ estimates of divergence between the Class B and each of the seven Class A subgroups, the third subgroup (A3) of samples from Class A appears to be the most divergent from the Class B. Subgroups A5, A6, or A7 appear to be the least divergent, depending on which vector subset is used for the estimates. Two

Henze-Penrose (HP) Divergence Estimates For Group B vs. Group A Subgroups for Various Vector Subsets									
Subgroup	Input Data Only	Data + Full Extension	L-moments	L-moment Ratios	L-statistics	WECO	Full Extension	Data + WECO	Data + L-moments
A1	0.735	0.745	0.750	0.645	0.760	0.645	0.715	0.730	0.765
A2	0.720	0.675	0.680	0.600	0.670	0.645	0.655	0.710	0.705
A3	0.920	0.900	0.880	0.660	0.880	0.775	0.855	0.895	0.900
A4	0.765	0.810	0.765	0.635	0.730	0.715	0.805	0.775	0.800
A5	0.625	0.620	0.510	0.525	0.500	0.615	0.610	0.615	0.615
A6	0.565	0.595	0.525	0.500	0.510	0.525	0.485	0.565	0.565
A7	0.530	0.570	0.525	0.570	0.540	0.520	0.505	0.530	0.550
Mean (Average)	0.694	0.702	0.662	0.591	0.656	0.634	0.661	0.689	0.700
Standard Deviation	0.134	0.122	0.145	0.062	0.145	0.093	0.141	0.128	0.131

Table 8.1: HP Estimates for Case Study D

subgroups of Class A were chosen, along with the Class B set, as the basis of the two SVM experiments that follow. For Class A training sample candidates, experiment D1 utilizes subgroup A3, the most divergent subgroup of Class A with respect to the Class B set. In experiment D2, Class A training samples were was chosen from subgroup A7 (one of the less divergent subgroups). For both SVM's, equations 3.20 and 3.21, with $p = 1$, were invoked to solve for the SVM weight vectors and offsets.

Experiment D1

For this experiment, Class B is considered the positive set and Class A is considered the negative set. An SVM was trained using 8 samples from class B and 8 samples from subgroup A3 of Class A. The training samples for each class were randomly chosen from among the 100 training candidate members.[2] The remaining 784 samples (92 from Class B and 692 from Class A) were used as test samples. Figure 8.1 shows the resultant weight vector for this SVM obtained using the fully extended vector set. Table 8.2 shows the trainings errors and classification performance obtained for various subvector combinations.

Experiment D2

As in the prior experiment, Class B is considered the positive set. The SVM for this experiment used 16 training samples (8 from each class), leaving 784 test samples. The positive training samples were the same as those used in Experiment D1. The negative training samples were randomly chosen from among the members of subgroup A7. Figure 8.2 shows the weight vector for this SVM after training with the

[2] A random number generator (a component within a commercially available spreadsheet software program) was used to generate 3 columns of 8 unique integers each within the inclusive range of 1 to 100. It was not required that the numbers be unique between columns, only within columns, resulting in the generation of 22 unique numbers and 2 replicates. The generated numbers in the three columns were then used as the indices by which to select the training vectors from the Class B, Class A3, and Class A7 sets, respectively.

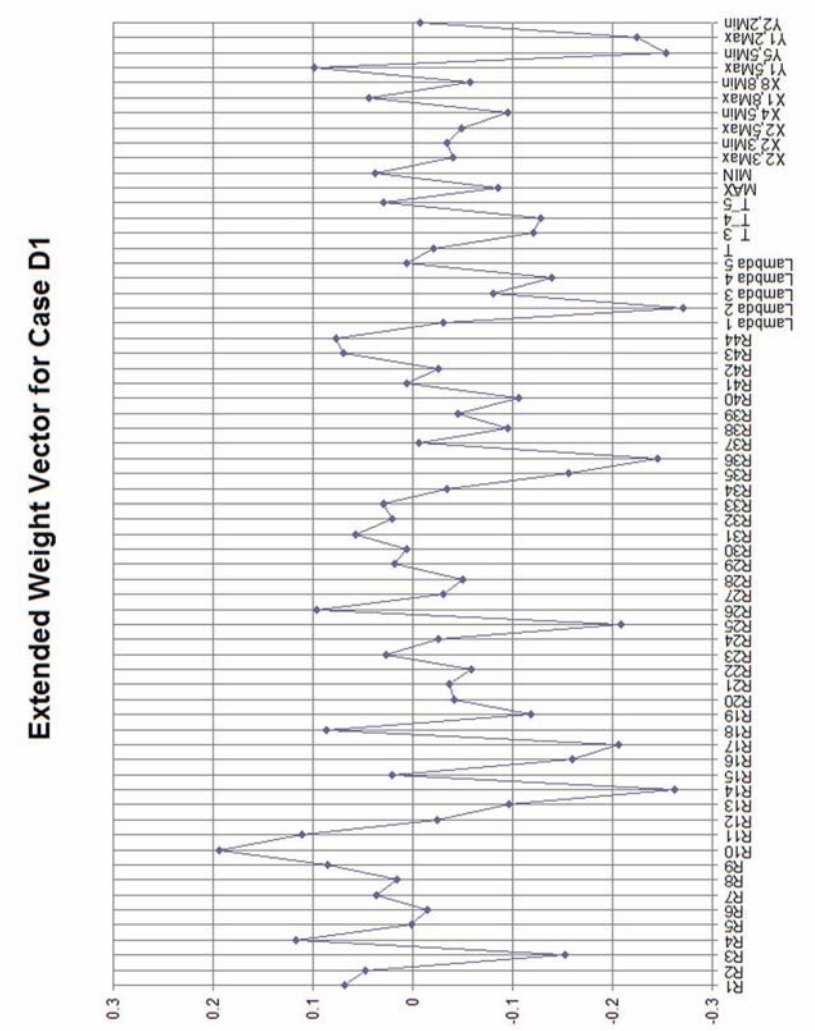

Figure 8.1: Extended Weight Vector for Experiment D1

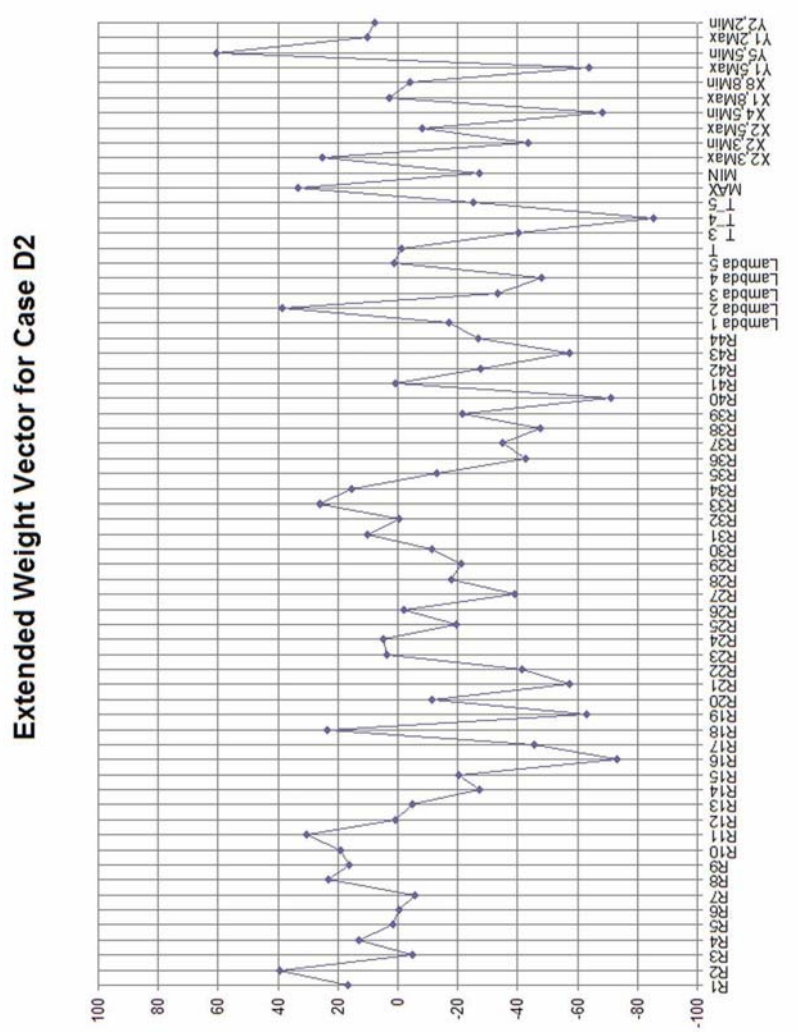

Figure 8.2: Extended Weight Vector for Experiment D2

fully extended vector set. Table 8.2 shows the trainings errors and classification performance.

Observations and Conclusions

For experiment D1, the detection rates under any of the subvector cases were, with the exception of the L-moment ratio case, 84% or better, indicating that given all 700 positive samples from class B, the associated SVM (SVM_D1) correctly labeled at least 588 of the samples. And, for the L-moment extension, the false positive rate was reduced as well. However, the high false positive rates, which were all greater than 51%, support the hypothesis that, overall, class B is not generally separable from class A, at least by the methods employed for this experiment.

Note that for experiment D1, extending the data vector with the full extension set, the first five L-moments, or the WECO information did improve the detection rate.Also of interest is the fact that under experiment D1, using the first five L-moments or the L-statistics in place of the data vector provide classification results on par with (actually, slightly better than) the results attained using the input data alone. This is supportive of the notion that, for datastream classification scenarios in which separability may be largely depend on datastream statistics (rather than element by element inter-vector statistics), the first five L-moments can be used in a classifier to represent the original input datastream or vector.

For experiment D2, which used a training set from class A that had a distribution with little divergence from members of class B, the resultant SVM classifier had poor detection rates ($\leq 50\%$) and training errors for most of the sub-vector cases, including the input-data-only case. This result indicates that the training data did not provide sufficient information on which to base a "good" separating hyperplane by which the larger set (training + test samples) could be classified. Even in the sub-vector cases where there were no training errors, the

		SubVector Used for SVM (Hyperplane Classifier)								
		Input Data Only	Data + Full Extension	L- moments	L- moment Ratios	L-statistics	WECO	Full Extension	Data + WECO	Data + L- moments
Experiment D1	Training Errs.	0	0	0	3	0	0	0	0	0
	False Pos. Rate	0.591	0.579	0.516	0.546	0.523	0.699	0.547	0.616	0.571
	Detection Rate	0.850	0.870	0.880	0.670	0.890	0.900	0.840	0.880	0.880
Experiment D2	Training Errs.	4	2	3	8	3	6	0	0	0
	False Pos. Rate	0.310	0.367	0.451	0.356	0.486	0.556	0.544	0.399	0.329
	Detection Rate	0.300	0.380	0.400	0.440	0.490	0.420	0.500	0.350	0.310

Table 8.2: Classification Results for Case Study D

resultant classifier did not generalize well to the larger sample set. In all three cases of errorless training, the detection rate was $\leq 50\%$ and the false positive rate exceeded the detection rate.

An additional indicator of the relative ease or difficultly of training the SVM could also be discerned by observing the SVM coefficient values (i.e. weight vector element values). The large magnitudes of the weight vector element values for experiment D2 (see graph 8.2) compared to the relatively low magnitudes seen in experiment D1 (see graph 8.1) have a loosely inverse correlation to the estimated HP divergence of the training sets in each case. The training set with the lower divergence between its two classes resulted in coefficients with larger magnitudes than those resulting from the training set with higher divergence.

Taken together, the results of both experiments indicate that there is likely little performance difference between members of class A and class B that can be discerned on the basis of the transducer end item accuracy testing represented by the input data. In other words, the sampled process data used as input data for this SVM case study did not signal general performance degradation of the sensor based on used of component part provided by Supplier B relative to those historically provided by Supplier A.

Chapter 9

SUMMARY AND DIRECTIONS FOR FURTHER RESEARCH

9.1 Summary

Statistical process control (SPC) techniques have been widely applied to the product manufacturing arena, one objective being to improve product quality and field reliability. Over the years, SPC approaches has demonstrated added advantage toward the objective of product reliability over simply using inspection criteria and specification limits as the sole means of screening product on a PASS/FAIL basis. However, there are cases in which, even though the available manufacturing test data for the components of a product are within statistical process control limits, the product later fails a downstream manufacturing test at the end-item level or fails during use in the field. In some cases, there are cues in the manufacturing test data, which were generated upstream from the point of failure, that are highly correlated to the occurrence of the particular failure scenario. Such cues have been used as failure predictors, enabling implicated components or end-items to be screened out prior to further production processing or field use. In practice, the cues may not be a perfect predictor of the failure mode of interest, so there is often a trade-off to be negotiated, in adjusting the predictor-based screening criteria, between the rate of detection and the risks of either rejecting "good" units or allowing "bad" units to proceed further downstream.

As part of this research, a binary classification methodology was developed that can be used to design (analyze) and implement (synthesize) predictors of end-item field failure/survival or downstream product test pass/fail performance based on upstream test data. Additionally, the methodology can be used as a forensic tool for failure analysis and root-cause investigations. The implementation form of the prediction classifier is given by equation 3.2.

Once trained for a specific dataset, the prediction machine is most effective in cases where the methodology has identified high correlation between one or more parameter elements of the upstream data and the downstream failure mode. The methodology also provides a weight vector (w) whose element values serve to indicate the relative importance of the elements of a parameter input (or time-series) vector. This enables the use of this methodology as a forensic tool in determining likely contributing factors in support of investigations of manufacturing yield issues or field failures. Such identification can also prove helpful in validating the results of product improvements aimed at correcting failure modes known to be associated with detectable cues in manufacturing test data. In real-world scenarios, the correlation between the downstream failure and cues in the upstream manufacturing data is not generally ideal, so some cost/benefit analysis may need to be incorporated in the decision process of what trade-offs need to be made between the detection rate and the false positive rate when introducing a predictor into the manufacturing process stream as a screening tool. Or, indeed, whether the predictor should be implemented at all or only used as a data analysis or forensics tool. There are cases where the upstream test data would not be expected to provide true cues, especially if a latent defect or field induced defect is unrelated to any currently tested performance effects. In such cases, the prediction machine developed under this methodology might fail to perform well on members of the target dataset that were not included in training the predictor. Ironically, in such a case, this might serve as one piece of evidence, not necessarily conclusive, that the manufacturing data under review offers little or no cues related to the failure mode of interest.

In this dissertation, we have explored the use of product field reliability data (or end-item test data) to infer data mining or pattern recognition criteria onto manufacturing process data by means of a hyperplane classifier in conjuction with transform mappings in order to provide reliability prediction models. The

Henze-Penrose divergence estimate on the classifier test input data is used as a means of gaging the relative performance of the prediction models across several case studies. The data we have chosen to analyze for reliability cues is prior manufacturing test data in which any data that lie beyond some chosen statistical control limits have already been eliminated. However, downstream tests or field use may later reveal units that fail or function uncharacteristically. The question then becomes whether there are some characteristics in the statistically controlled prior data that can be exploited via a hyperplane classifier to detect, based on this prior data, which of the units under test is more likely to belong to a particular class of units whose downstream or field performance differs from that of the general population.

We have made the following contributions:

- Algorithmic details of a modified SVM classifier that can be trained on labeled subsets of data from a statistically controlled process along with performance analysis of the classifier on several sets of actual manufacturing test data. The classifier so trained could then be used as a predictor function on the elements of the dataset for inclusion in the class represented by the training data.

- The use of L-moment vectors and/or L-moment extensions to the input data vectors as means of increasing the discrimination power of the SVM upon the data streams from a statistically controlled process or upon multi-parameter vectors that may have correlation between elements.

- Algorithmic details and performance analysis of a modified SVM classifier that uses specific functions of order statistics of input vectors in order to embed discriminant information into the classifier equivalent to that required in the implementation of the classical 3-sigma process limits and Western Electric Rules.

Use of kernel functions allowed substitution or extension of classifier input vectors, at times improving the performance of the classifier on the test set. L-moment and WECO information extensions can serve to provide additional discriminating power to the SVM, especially in cases were the input data from both classes comes from an overall dataset that is already in statistical control.

9.2 Directions for Further Research

In determining the SVM weight vector, we first determined the solution to the Wolfe dual optimization equation 3.12, arriving at a solution in the form of equation 3.20. Rather than solve this equation iteratively, with η_i set to zero, we chose to fix the solution for the modified SVM by setting η_i with a relation based on the relative lengths of the input vectors as expressed in equation 3.21. We observed that the value of p could be adjusted to obtain a maximum margin hyperplane that, depending on the particular training set, approximates or actually matches the optimal hyperplane as defined in equation 3.4. In most cases, we left the value of p set to 1. One area of research might be to explore automatic optimization of the choice of p with constraints to avoid solution instability due to overfitting. A related area would be to explore other relations for η_i.

In the one case study of statistically controlled datastreams, it appeared that the first five L-moments were able to serve as a type of compressed data representation of the 44-element input data vector, providing classification results on par with those achieved using the input vector itself. However, while adding the WECO information to the SVM resulted in some improvement to the detection rate, it also resulted in increasing the false positive rate. This would seem to indicate, as might be expected in this case, that the WECO condition states are significantly similiar between the two classes. This suggests at least two potential areas of research. One would be to explore the robustness of L-moment representations of statistically controlled datastreams

(possibly with some outliers included) in training predictive classifiers both with respect to their ability to "compact" the input data and with respect to whether using only the first five L-moments is sufficient for various types of manufacturing or process test data. The other area involves exploring the effectiveness of SVM's that are trained utilizing WECO condition information derived from the order statistics of input data representing the classes of in-control (non-alarm) and out-of-control (alarm) processes.

Another open area is the extension of this binary classification methodology to multi-class problems. The simplicity of classifier implementation might justify exploration of the following scenario. For some multi-class problems, one could envision the use of several binary classifiers each used to classify different groupings of the subject classes. Then the joint outcomes of the classifiers could be used as the basis of deciding the class label that should be assigned to a test input.

SVM's are but one approach to detecting correlations between input data and classification assignments. Some of the methodology employed in this study (such as element-by-element ensemble statistical normalization and input vector extension or substitution via the use of kernels) are generalizable for use with other types of classification engines or paradigms including, for instance, information theoretic or entropy-based approaches. Also open for further research is exploration of the performance optimality bounds of this methodology using SVM's or other types of classifiers.

BIBLIOGRAPHY

[1] J. Prins, *NIST/Sematech e-Handbook of Statistical Methods*, ch. 6. National Institute of Standards and Technology, 2006.

[2] F. Kear, *Statistical Process Control in Manufacturing Practice*. New York: Marcel Dekker, 1998.

[3] S. J. Wierda, *Multivariate Statistical Process Control*. The Netherlands: WoltersgroepGroningen, 1994.

[4] C. P. Quesenberry, *SPC Methods for Quality Improvement*. New York: John Wiley & Sons, 1997.

[5] J. R. Thompson and J. Koronaki, *Statistical Process Control: The Deming Paradigm and Beyond*. Boca Raton: Chapman & Hall/CRC, second ed., 2002.

[6] W. A. Shewhart, *Economic Control of Quality of Manufactured Product*. New York: D. van Nostrand Co., 1931.

[7] G. Shmueli and A. Cohen, "Run length distribution for control charts with runs rules," *Communications in Statistics - Theory & Methods*, vol. 32, no. 2, pp. 475–495, 2003.

[8] B. J. Nelson, *Multivariate State Space Mapping Models for Process and Quality Improvement*. PhD thesis, Arizona State University, Department of Electrical Engineering, 2000.

[9] N. Niang, "Multidimensional methods for statistical process control: Some contributions of robust statistics," in *Multivariate Total Quality Control: Foundations and Recent Advances* (V. E. V. G. S. Carlo Lauro, Jaromir Antoch, ed.).

[10] V. N. Vapnik, *Statistical Learning Theory*. New York: John Wiley & Sons, 1998.

[11] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, second ed., 2000.

[12] S. V. Kartalopoulos, *Understanding Neural Networks and Fuzzy Logic*. New York: IEEE Press, 1996.

[13] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge: Cambridge University Press, 2000.

[14] C. J. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[15] M. Vidysagar, *Learning and Generalisation: With Applications to Neural Networks*. London: Springer-Verlag, second ed., 2003.

[16] A. J. Smola and B. Scholkopf, “A tutorial on support vector regression,” tech. rep., NeuroCOLT, Berlin, October 1998. NC2-TR-1998-030.

[17] J. Zhu and T. Hastie, “Kernel logistic regression and the import vector machine,” *Journal of Computational and Graphical Statistics*, vol. 14, no. 1, pp. 185–205, 2005.

[18] V. N. Vapnik, *Estimation of Dependencies Based on Empirical Data*. New York: Springer, second ed., 2006. Contains Reprint of 1982 edition plus a second part entitled "Empirical Inference Science: Afterword of 2006".

[19] H. Neemuchwala and A. Hero, “Image registration in high dimensional feature space,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 52, no. 1, pp. 105–124, 1990.

[20] J. R. M. Hosking, “L-moments: Analysis and estimation of distributions using linear combinations of order statistics,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 52, no. 1, pp. 105–124, 1990.

[21] J. R. M. Hosking, “On the characterization of distributions by their l-moments,” *Journal of Statistical Planning and Inference*, vol. 136, pp. 193–198, 2006. Available online 25 July 2004.

[22] G. M. Bonnin, D. Todd, B. Lin, T. Parzybok, M. Yekta, and D. Riley, *Semiarid Southwest(Arizona, Southeast California, Nevada, New Mexico, Utah)*, vol. 1 of *NOAA Atlas 14: Precipitation-Frequency Atlas of the United States*. Silver Spring, Maryland: U.S. Department of Commerce, National Oceanic and Atmospheric Administration, National Weather Service, 2004.

[23] J. R. M. Hosking and J. R. Wallis, *Regional Frequency Analysis: An Approach Based on L-Moments*. New York: Cambridge University Press, 1997.

[24] H. A. David and H. N. Nagaraja, *Order Statistics*. Hoboken, New Jersey: John Wiley & Sons, Inc., third ed., 2003.

[25] G. Holton, "Quantile," April 2006. Available Online: www.riskglossary.com/link/quantile.htm.

[26] J. A. Greenwood, J. M. Landwehr, N. C. Matalas, and J. R. Wallis, "Probability weighted moments: Definition and relation to parameters of several distributions expressable in inverse form," *Water Resources Research*, vol. 15, pp. 1049–1054, 1979.

[27] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, Massachusetts: The MIT Press, 2002.

[28] G. B. Folland, *Real Analysis: Modern Techniques and Their Applications*. New York: John Wiley & Sons, Inc., 1984.

[29] B. Scholkopf, "Statistical learning and kernel methods," tech. rep., Microsoft Research Limited, Microsoft Coporation, Cambridge, February 2000. MSR-TR-2000-23.

[30] C. R. Bector, S. Chandra, and J. Dutta, *Principles of Optimization Theory*. Harrow, U.K.: Alpha Science International Ltd., 2005.

[31] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," tech. rep., Microsoft Research Limited, Microsoft Coporation, April 1998. MSR-TR-98-14.

[32] J. Shawe-Taylor and N. Christianini, *Kernel Methods for Pattern Analysis*. Cambridge: Cambridge University Press, 2004.

[33] N. Aronszajn, "Theory of reproducing kernels, reprinted by permission from am. math. soc. trans. 68:337-404 (1950)," in *Reproducing Kernel Hilbert Spaces: Applications in Statistical Signal Processing* (H. L. Weinert, ed.).

[34] W. Rudin, *Functional Analysis*. New York: McGraw-Hill, Inc., 1991.

[35] W. Rudin, *Real and Complex Analysis*. New York: McGraw-Hill, Inc., 1987.

[36] H. Dym, *J Contractive Matrix Functions, Reproducing Kernel Hilbert Spaces and Interpolation*. Providence, Rhode Island: American Mathematical Society, 1989.

[37] S. Abe, *Support Vector Machines for Pattern Classification*. London, U.K.: Springer-Verlag, 2005.