DATA MINING MODELS FOR TACKLING HIGH DIMENSIONAL DATASETS AND OUTLIERS

by

ORESTIS PANOS PANAGOPOULOS
B.S. Technical University of Crete, 2010
M.S. University of Florida, 2014

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2016

Major Professor: Petros Xanthopoulos

ABSTRACT

High dimensional data and the presence of outliers in data each pose a serious challenge in supervised learning.

Datasets with significantly larger number of features compared to samples arise in various areas, including business analytics and biomedical applications. Such datasets pose a serious challenge to standard statistical methods and render many existing classification techniques impractical. The generalization ability of many classification algorithms is compromised due to the so-called curse of dimensionality. A new binary classification method called constrained subspace classifier (CSC) is proposed for such high dimensional datasets. CSC improves on an earlier proposed classification method called local subspace classifier (LSC) by accounting for the relative angle between subspaces while approximating the classes with individual subspaces. CSC is formulated as an optimization problem and can be solved by an efficient alternating optimization technique. Classification performance is tested in publicly available datasets. The improvement in classification accuracy over LSC shows the importance of considering the relative angle between the subspaces while approximating the classes. Additionally, CSC appears to be a robust classifier, compared to traditional two step methods that perform feature selection and classification in two distinct steps.

Outliers can be present in real world datasets due to noise or measurement errors. The presence of outliers can affect the training phase of machine learning algorithms, leading to over-fitting which results in poor generalization ability. A new regression method called relaxed support vector regression (RSVR) is proposed for such datasets. RSVR is based on the concept of constraint relaxation which leads to increased robustness in datasets with outliers. RSVR is formulated using both linear and quadratic loss functions. Numerical experiments on benchmark datasets and computational comparisons with other popular regression methods depict the behavior of our proposed method. RSVR achieves better overall performance than support vector regression (SVR) in measures such as RMSE and R^2_{adj} while being on par with other state-of-the-art regression methods

such as robust regression (RR). Additionally, RSVR provides robustness for higher dimensional datasets which is a limitation of RR, the robust equivalent of ordinary least squares regression. Moreover, RSVR can be used on datasets that contain varying levels of noise.

Lastly, we present a new novelty detection model called relaxed one-class support vector machines (ROSVMs) that deals with the problem of one-class classification in the presence of outliers.

I dedicate this to the individuals who have inspired me.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, dissertation supervisor, and mentor Dr. Petros Xanthopoulos, for his guidance and inspiring instruction as well as for supporting and trusting me throughout this work. I would like to sincerely thank my Ph.D. committee members Dr. Luis Rabelo, Dr. Qipeng Phil Zheng, and Dr. Damian Dechev for their continued support during this work. Furthermore, I would like to thank Dr. Vijay Pappu, Dr. Andonis Mitidis, Dr. Athanasios Aris Panagopoulos, Dr. Talayeh Razzaghi, Dr. Onur Şeref, and Dr. Panos M. Pardalos for their valuable and constructive suggestions during the development of this study. Last but not least, I would like to express my love and gratitude to my family for supporting me unconditionally.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

High Dimensional Data and Supervised Learning

Classification tasks on high dimensional datasets pose significant challenges to the standard statistical methods and render many existing classification techniques impractical (Johnstone & Titterington, 2009). The generalization ability of many classification algorithms is compromised due to *curse of dimensionality* arising from high number of features of the input space (Köppen, 2000). Earlier studies have revealed the geometrical distortion that arises in high dimensional data spaces, where the ratio of distances between the farthest and nearest neighbors to a given target is almost equal to 1 for a wide variety of data distributions and distance functions (Beyer, Goldstein, Ramakrishnan, & Shaft, 1999). Moreover, several statistical methods require knowing class covariances *a-priori*. In the case that class covariances are unavailable, such estimates from sample data would be unreliable due to small sample sizes. One common approach to address the aforementioned challenges involves reducing the dimensionality of the dataset either by using feature extraction (Liu & Motoda, 1998) and/or feature selection prior to classification (Saeys, Inza, & Larrañaga, 2007; Carrizosa & Morales, 2013).

Feature selection is usually performed in different ways through filter, wrapper, and embedded methods. Filter methods access features during a separate process prior to classification. Variables are given a score according to a filtering function and are ordered accordingly. Features with the lowest scores are discarded while the rest are used from the classifier. Hypothesis testing and statistic tests such as t-test have also been used as filtering procedures (Guyon & Elisseeff, 2003). Wrapper methods on the other hand use the classifier structure itself to evaluate the importance of features based on the idea that the classifier can provide a better estimate of accuracy than a separate independent process (Blum & Langley, 1997). The main drawback of wrapper methods is that increased computational power is often required since the classification process has to be

repeated for each feature set considered. Metaheuristics used for feature selection can also be classified as wrapper methods (Unler & Murat, 2010; López, Torres, Batista, Pérez, & Moreno-Vega, 2006; J. Yang & Olafsson, 2006). Embedded methods perform feature selection in a way so that the classification algorithm is executed while variables are evaluated and selected. Examples include the weighting of features in support vector machines (Guyon, Weston, Barnhill, & Vapnik, 2002), where the authors developed the SVM method of recursive feature elimination for feature selection, and the use of random forests for feature evaluation (Jiang et al., 2004). In the later, feature elimination occurs for the attributes with the lowest raw importance score.

Feature extraction techniques transform the input data into a set of *meta*-features that extract the relevant information from the input data for classification. One popular technique called *principal component analysis (PCA)*, finds a set of linearly uncorrelated variables called *principal components* from a set of observations of possibly correlated variables (Jolliffe, 2005; Shanmugam & Johnson, 2007). PCA removes redundancy by transforming the data from a higher dimensional space into an orthogonal lower dimensional space [Appendix A]. This transformation is performed in a way that the first principal component captures as much variation in the data as possible, and each succeeding component accounts for a decreasing amount of variance (Vidal et al., 2006). The number of retained principal components is usually less than or equal to the number of original variables and are determined using several criteria like the eigenvalue-one criterion, scree test, proportion of variance accounted for, etc.

The aforementioned dimensionality reduction techniques decrease the complexity of the classification model and attempt to improve the classification performance (Saeys et al., 2007). The choice of the dimensionality reduction technique depends on the nature (e.g. level of correlation, presence of outliers) of the data that is used for classification.

Local Subspace Classifier (LSC) (Laaksonen, 1997) utilizes PCA to perform classification. During the training phase, a lower dimensional subspace is found for each class that approximates the data. In the testing phase, a new data point is classified by calculating the distance of the point from each

subspace and choosing the class with minimal distance. Although LSC is simple and relatively easy to implement, it has its own limitations (Fenn & Pappu, 2012). LSC finds the subspaces for each class *separately* without the *knowledge* of the presence of the other class. While each subspace approximates the data well, these projections may not be *ideal* from a classification perspective. In this work, we construct a novel classifier called *Constrained Subspace Classifier* (CSC) that accounts for the presence of another class while finding the individual subspaces. LSC formulation is modified to include the relative angle between the subspaces and is solved efficiently using alternate optimization techniques. The performance of CSC on publicly available datasets is evaluated and compared to that of LSC.

Outliers in Data and Regression Analysis

The regression problem is defined as a task where output variables are assigned real values, which are estimated from a set of input variables used for training. Ordinary least squares (OLS) regression has been a baseline for solving such problems (Johansen, 1988). While OLS regression is simple and effective it comes with some limitations. First, the generalization ability of OLS regression decreases when applied to data where the number of features is higher than the number of samples (G. Cao, Guo, Bouman, et al., 2010). The same applies to cases where the number of features is close to the number of samples. Increased variability when compared to the size of the training set may lead OLS regression to fit itself to the noise described by the redundant features usually found in high dimensional data (E. Yang, Lozano, & Ravikumar, 2014; Panagopoulos, Pappu, Xanthopoulos, & Pardalos, 2015; Pappu, Panagopoulos, Xanthopoulos, & Pardalos, 2015). Second, the performance of OLS degrades when applied to data with outliers (Rousseeuw & Leroy, 2005).

Outliers can be present in real world datasets due to noise or measurement errors (Hawkins, 1980). The presence of outliers can affect the training phase of machine learning algorithms (MLAs),

leading to overfitting which results in poor generalization ability. Enhancing MLAs' ability to deal with outliers results to better fitting and increased performance (Smith & Martinez, 2011; Peters & Lacic, 2012; F. Cao, Ye, & Wang, 2015; D'Urso, Massari, & Santoro, 2011). To that end, OLS regression has been improved through the development of robust regression (RR) to better handle outliers (Street, Carroll, & Ruppert, 1988). It has been shown that robust regression can outperform OLS regression on noisy datasets, yet increased dimensionality of the input space might influence its performance (Wolters & Kateman, 1989). The success of robust regression has led to incorporating it as a standard regression method into modern statistical packages such as Stata (Verardi & Croux, 2008) and JMP from SAS (Freund, Littell, & Creighton, 2003).

Another class of methods used for regression problems is support vector machines [Appendix B] These methods fit the model into a transformed feature space dictated by the properties of the chosen kernel function (V. N. Vapnik & Vapnik, 1998; Bishop, 2006). SVMs are able to handle higher dimensional data compared to OLS, they generalize better, but yet they are sensitive to outliers. Support vector regression (SVR) (Smola & Schölkopf, 2004) extends support vector machines, which were originally built for binary classification, to perform regression estimation. The performance of SVR has been well studied in the literature. Our goal is to enhance SVR to better handle outliers in data. We ultimately aim to increase the robustness of support vector regression.

One-Class Classification

One-class classification is different from the conventional classification problem in one essential way. That is, in one-class classification, we must assume that only one class, the target class, has available information, meaning that information about the other classes (outliers) is not known. In one-class support vector machines (OSVMs) the boundary has to be constructed by using the information available, that is, only the information of the target class (Schölkopf, Williamson, Smola,

Shawe-Taylor, & Platt, 1999). The aim is to construct a boundary that accepts the fewest number of outliers and the most target data possible. The presence of outliers in data may negatively influence the classification accuracy of one-class support vector machines . We propose a modified version of OSVMs, called relaxed one-class support vector machines (ROSMs) which aims to mitigate the negative influence that noise might have on classification performance.

The remainder of the dissertation is organized as follows. Chapter 2 presents constrained subspace classifier and chapter 3 introduces relaxed support vector regression. In chapter 4 we demonstrate relaxed one-class support vector machines for novelty detection. Lastly, in chapter 5 we discuss the main results and impact of this work.

CHAPTER 2: CONSTRAINED SUBSPACE CLASSIFIER

Datasets with significantly larger number of features, compared to samples, pose a serious challenge in supervised learning. Such datasets arise in various areas including business analytics. A new binary classification method called *Constrained subspace classifier (CSC)* is proposed for such high dimensional datasets. CSC improves on an earlier proposed classification method called *Local subspace classifier (LSC)* by accounting for the relative angle between subspaces while approximating the classes with individual subspaces. CSC is formulated as an optimization problem and can be solved by an efficient alternating optimization technique. Classification performance is tested in publicly available datasets. The improvement in classification accuracy over LSC shows the importance of considering the relative angle between the subspaces while approximating the classes. Additionally, CSC appears to be a robust classifier compared to traditional two step methods.

Local Subspace Classifier

Consider a binary classification problem. Let the matrices $\mathcal{X}_1 \in \mathbb{R}^{p\times m}$ and $\mathcal{X}_2 \in \mathbb{R}^{p\times l}$ be given, whose columns represent the training examples of two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. The number of samples in $\mathcal{C}_1$ and $\mathcal{C}_2$ are given by m and n respectively. The number of features is given by p. Local subspace classifier attempts to find two subspaces separately, one for each class that *best* approximates the data. Let $\boldsymbol{U}_1 = [\boldsymbol{u}_1^{(1)}, \boldsymbol{u}_2^{(1)}, \ldots, \boldsymbol{u}_k^{(1)}]_{p\times k}$ and $\boldsymbol{U}_2 = [\boldsymbol{u}_1^{(2)}, \boldsymbol{u}_2^{(2)}, \ldots, \boldsymbol{u}_k^{(2)}]_{p\times k}$ represent orthonormal bases of two k-dimensional linear subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ that approximate classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. We assume the dimensionality of subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ to be same and equal to k without loss of generality. $\mathcal{S}_1$ and $\mathcal{S}_2$ attempt to capture *maximal* variance in classes

$\mathcal{C}_1$ and $\mathcal{C}_2$ respectively by optimizing the following optimization problems:

$$\begin{aligned} \underset{\boldsymbol{U}_1 \in \mathbb{R}^{p \times k}}{\text{maximize}} \quad & \text{tr}(\boldsymbol{U}_1^T \mathcal{X}_1 \mathcal{X}_1^T \boldsymbol{U}_1) \\ \text{subject to} \quad & \boldsymbol{U}_1^T \boldsymbol{U}_1 = \boldsymbol{I}_k \end{aligned} \tag{2.1}$$

where $\boldsymbol{I}_k$ is the identity matrix of size k.

The solution to the optimization problem (2.1) is given by the eigenvectors corresponding to the k largest eigenvalues of matrix $\mathcal{X}_1 \mathcal{X}_1^T$ (Golub & Van Loan, 2012). Similarly, the following optimization problem is solved to obtain the orthonormal basis $\boldsymbol{U}_2$ representing $\mathcal{S}_2$:

$$\begin{aligned} \underset{\boldsymbol{U}_2 \in \mathbb{R}^{p \times k}}{\text{maximize}} \quad & \text{tr}(\boldsymbol{U}_2^T \mathcal{X}_2 \mathcal{X}_2^T \boldsymbol{U}_2) \\ \text{subject to} \quad & \boldsymbol{U}_2^T \boldsymbol{U}_2 = \boldsymbol{I}_k \end{aligned} \tag{2.2}$$

The orthonormal basis $\boldsymbol{U}_2$ is obtained by choosing eigenvectors corresponding to the k largest eigenvalues of matrix $\mathcal{X}_2 \mathcal{X}_2^T$. A new point $\boldsymbol{x}$ is classified by computing its distance from subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$:

$$\text{dist}(\boldsymbol{x}, \mathcal{S}_i) = \text{tr}(\boldsymbol{U}_i^T \boldsymbol{x} \boldsymbol{x}^T \boldsymbol{U}_i) \tag{2.3}$$

and the class of $\boldsymbol{x}$ is determined as:

$$\text{class}(\boldsymbol{x}) = \arg \min_{i \in \{1,2\}} \{\text{dist}(\boldsymbol{x}, \mathcal{S}_i)\} \tag{2.4}$$

Though the subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ approximate the classes well, these projections may not be *ideal* for classification tasks as each of them are obtained *without* the knowledge of another subspace. Hence, from a classification performance perspective, these subspaces may not be the *best* projections for the classes. In order to account for the presence of another subspace, we consider the relative orientation of the subspaces.

Constrained Subspace Classifier

Constrained subspace classifier finds two subspaces *simultaneously*, one for each class, such that each subspace accounts for maximal variance in the data in the *presence* of the other class/subspace. Thus, CSC allows for a *tradeoff* between approximating the classes well and the relative orientation among the subspaces. The relative orientation between subspaces is generally defined as principal angles (Hamm & Lee, 2008). We briefly review principal angles between subspaces below, which are further utilized to modify the formulation of LSC to include the relative orientation among the subspaces.

Definition 1. *Let $\boldsymbol{U}_1 \in \mathbb{R}^{p\times k}$ and $\boldsymbol{U}_2 \in \mathbb{R}^{p\times k}$ be two orthonormal matrices spanning subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$. The principal angles $0 \leq \theta_1 \leq \theta_2 \leq \theta_3 \leq \cdots \leq \theta_k \leq \pi/2$ between subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$, are defined recursively by:*

$$\begin{aligned} cos\theta_i = \max_{\boldsymbol{x}_m \in \mathcal{S}_1} \max_{\boldsymbol{y}_n \in \mathcal{S}_2} \quad & \boldsymbol{x}_m^\intercal \boldsymbol{y}_n \\ \textit{subject to} \quad & \boldsymbol{x}_m^\intercal \boldsymbol{x}_n = 1, \quad \boldsymbol{y}_m^\intercal \boldsymbol{y}_n = 1, \quad for\ m = n \\ & \boldsymbol{x}_m^\intercal \boldsymbol{x}_n = 0, \quad \boldsymbol{y}_m^\intercal \boldsymbol{y}_n = 0, \quad for\ m \neq n \\ & \forall m, n = 1, 2, \ldots, k. \end{aligned} \tag{2.5}$$

where $\boldsymbol{x}_m$ and $\boldsymbol{y}_n$ are the column vectors of $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ respectively. Intuitively, the first principal angle θ_1 is the smallest angle between all pairs of unit vectors in the first and second subspaces. The rest of the principal angles are similarly defined.

Theorem 1. *Let $U_1 \in \mathbb{R}^{p\times k}$ and $U_2 \in \mathbb{R}^{p\times k}$ be rectangular matrices whose column vectors span the subspaces $S_1 \in \mathbb{R}^k$ and $S_2 \in \mathbb{R}^k$ respectively. Let $M = U_1^\top U_2 \in \mathbb{R}^{k\times k}$,using singular value*

decomposition we can express M by:

$$M = YCZ^{\top} \tag{2.6}$$

where $Y^{\top}Y = I_k,\ Z^{\top}Z = I_k$ *and* $C = diag(\sigma_1, \sigma_2, \ldots, \sigma_k)$.
If we assume that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$ *then the principal angles are given by cos* $\theta_k\ = \sigma_k(M)\ \forall i = 1, 2, \ldots, k$.

Proof. See (Bjorck & Golub, 1973) □

The cosines of the principal angles are also sometimes known as *canonical correlations*.

We consider the metric that defines the relative orientation between S_1 and S_2 spanned by U_1 and U_2 respectively to be the projection F-norm (Edelman & Smith, 1998) defined by:

$$d_{pF}(\boldsymbol{U}_1, \boldsymbol{U}_2) = \frac{1}{\sqrt{2}} \|\boldsymbol{U}_1\boldsymbol{U}_1^{\top} - \boldsymbol{U}_2\boldsymbol{U}_2^{\top}\|_F \tag{2.7}$$

The projection F-norm is obtained by embedding the Grassmann manifold in the set of n-by-n projection matrices of rank p. The choice of the metric preserves convexity. It can be represented in terms of the sines of principal angles as follows.

The right hand side norm can be expressed as:

$$\|\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top\|_F^2 = \text{tr}((\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top)^\top(\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top)) \tag{2.8}$$

$$= \|\boldsymbol{U}_1\|_F^2 + \|\boldsymbol{U}_2\|_F^2 - 2\|\boldsymbol{U}_2^\top\boldsymbol{U}_1\|_F^2 \tag{2.9}$$

Using Theorem 1, (2.9) becomes:

$$= \sum_{i=1}^{k}\lambda_i + \sum_{i=1}^{k}\lambda_i - 2\sum_{i=1}^{k}\cos^2\theta_i \tag{2.10}$$

$$= k + k - 2\sum_{i=1}^{k}\cos^2\theta_i \tag{2.11}$$

$$= 2\sum_{i=1}^{k}\sin^2\theta_i \tag{2.12}$$

where λ_i are the eigenvalues of $U_j \quad \forall i = 1, 2, \ldots, k$ and $j = \{1, 2\}$.

Hence the projection F-norm becomes:

$$d_{pF}(\boldsymbol{U}_1, \boldsymbol{U}_2) = \frac{1}{\sqrt{2}}\|\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top\|_F = \sqrt{\sum_{i=1}^{k}\sin^2\theta_i} \tag{2.13}$$

The projection metric is utilized to incorporate the relative orientation between subspaces in LSC. The formulation of LSC is modified as shown below to obtain the *constrained subspace classifier*

(CSC):

$$\begin{aligned} \underset{\boldsymbol{U}_1,\boldsymbol{U}_2\in\mathbb{R}^{p\times k}}{\text{maximize}} \quad & \text{tr}(\boldsymbol{U}_1^T\mathcal{X}_1\mathcal{X}_1^T\boldsymbol{U}_1)+\text{tr}(\boldsymbol{U}_2^T\mathcal{X}_2\mathcal{X}_2^T\boldsymbol{U}_2)-C\|\boldsymbol{U}_1\boldsymbol{U}_1^\top-\boldsymbol{U}_2\boldsymbol{U}_2^\top\|_F^2 \\ \text{subject to} \quad & \boldsymbol{U}_1^T\boldsymbol{U}_1=\boldsymbol{I}_k \qquad (2.14a) \\ & \boldsymbol{U}_2^T\boldsymbol{U}_2=\boldsymbol{I}_k \qquad (2.14b) \end{aligned}$$

where the parameter C controls the tradeoff between the relative orientation of the subspaces and the approximation of the data.

From calculations in section 3.1:

$$\|\boldsymbol{U}_1\boldsymbol{U}_1^\top-\boldsymbol{U}_2\boldsymbol{U}_2^\top\|_F^2=2k-2\text{tr}(\boldsymbol{U}_1^\top\boldsymbol{U}_2\boldsymbol{U}_2^\top\boldsymbol{U}_1) \qquad (2.15)$$

Hence the optimization problem becomes:

$$\begin{aligned} \underset{\boldsymbol{U}_1,\boldsymbol{U}_2\in\mathbb{R}^{p\times k}}{\text{maximize}} \quad & \text{tr}(\boldsymbol{U}_1^T\mathcal{X}_1\mathcal{X}_1^T\boldsymbol{U}_1)+\text{tr}(\boldsymbol{U}_2^T\mathcal{X}_2\mathcal{X}_2^T\boldsymbol{U}_2)+C\text{tr}(\boldsymbol{U}_1^T\boldsymbol{U}_2\boldsymbol{U}_2^T\boldsymbol{U}_1) \\ \text{subject to} \quad & \boldsymbol{U}_1^T\boldsymbol{U}_1=\boldsymbol{I}_k \qquad (2.16a) \\ & \boldsymbol{U}_2^T\boldsymbol{U}_2=\boldsymbol{I}_k \qquad (2.16b) \end{aligned}$$

It is important to note here that when $C=0$, CSC reduces to LSC. Additionally, for larger positive values of C, the relative orientation between subspaces reduces, while for larger negative values of C, the relative orientation increases.

Here we introduce an alternating optimization algorithm to solve (2.16). For a fixed $\boldsymbol{U}_2$, (2.16)

reduces to:

$$\begin{aligned} \underset{\boldsymbol{U}_1 \in \mathbb{R}^{p\times k}}{\text{maximize}} \quad & \text{tr}(\boldsymbol{U}_1^T(\mathcal{X}_1\mathcal{X}_1^T + C\boldsymbol{U}_2\boldsymbol{U}_2^T)\boldsymbol{U}_1) \\ \text{subject to} \quad & \boldsymbol{U}_1^T\boldsymbol{U}_1 = \boldsymbol{I}_k \end{aligned} \tag{2.17}$$

The solution to (2.17) is obtained by choosing the eigenvectors corresponding to the k largest eigenvalues of symmetric matrix $\mathcal{X}_1\mathcal{X}_1^T + C\boldsymbol{U}_2\boldsymbol{U}_2^T$.

Similarly, for a fixed $\boldsymbol{U}_1$, (2.16) reduces to:

$$\begin{aligned} \underset{\boldsymbol{U}_2 \in \mathbb{R}^{p\times k}}{\text{maximize}} \quad & \text{tr}(\boldsymbol{U}_2^T(\mathcal{X}_2\mathcal{X}_2^T + C\boldsymbol{U}_1\boldsymbol{U}_1^T)\boldsymbol{U}_2) \\ \text{subject to} \quad & \boldsymbol{U}_2^T\boldsymbol{U}_2 = \boldsymbol{I}_k \end{aligned} \tag{2.18}$$

where the solution to (2.18) is again obtained by choosing the eigenvectors corresponding to the k largest eigenvalues of symmetric matrix $\mathcal{X}_2\mathcal{X}_2^T + C\boldsymbol{U}_1\boldsymbol{U}_1^T$. We define the following three termination rules:

- Maximum limit Z on the number of iterations,
- Relative change in $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ at iteration m and m+1,

$$\text{tol}_{\boldsymbol{U}_1}^m = \frac{\|\boldsymbol{U}_1^{(m+1)} - \boldsymbol{U}_1^{(m)}\|_F}{\sqrt{q}}, \quad \text{tol}_{\boldsymbol{U}_2}^m = \frac{\|\boldsymbol{U}_2^{(m+1)} - \boldsymbol{U}_2^{(m)}\|_F}{\sqrt{q}} \tag{2.19}$$

  where $q = pk$

- Relative change in objective function value of (2.16) at iteration m and m+1,

$$\text{tol}_f^m = \frac{F^{(m+1)} - F^{(m)}}{|F^{(m)}| + 1} \tag{2.20}$$

For proof of convergence see Theorem 2.

The algorithm for CSC can be summarized as follows:

Algorithm 1 CSC ($\mathcal{X}_1, \mathcal{X}_2, k, C$)

1. Initialize $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ such that $\boldsymbol{U}_1^T\boldsymbol{U}_1 = \boldsymbol{I}_k, \boldsymbol{U}_2^T\boldsymbol{U}_2 = \boldsymbol{I}_k$.
2. Find eigenvectors corresponding to the k largest eigenvalues of symmetric matrix $\mathcal{X}_1\mathcal{X}_1^T + C\boldsymbol{U}_2\boldsymbol{U}_2^T$.
3. Find eigenvectors corresponding to the k largest eigenvalues of symmetric matrix $\mathcal{X}_2\mathcal{X}_2^T + C\boldsymbol{U}_1\boldsymbol{U}_1^T$.
4. Alternate between 2 and 3 until one of the termination rules is satisfied.

Theorem 2. *Algorithm 1 converges.*

Proof. Let $\mathcal{S}_l$ be a subspace of the space $\mathcal{S}_L$, where L is the dimensionality of the original data points and l is the reduced dimensionality of those points when projected onto the subspace $\mathcal{S}_l$. There is a choice of γ many such subspaces where

$$\gamma = \frac{L!}{l!(L-l)!} \tag{2.21}$$

with each subspace choice having a basis whose elements are the vector columns of $\boldsymbol{U}^i = [\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \dots \quad \boldsymbol{u}_l] \in \mathbb{R}^{L\times l}$ where $u_k \in \mathbb{R}^L$ with $k = 1, 2, 3, ..., l$ and $i = 1, 2, 3, ..., \gamma$.

Each choice of $\boldsymbol{U}^i$ corresponds to a covariant matrix of the projected data points that has a trace given by $T^i \equiv \text{tr}(\boldsymbol{U}^{iT}XX^T\boldsymbol{U}^i)$. Since there is a finite number of subspaces, we also have a finite number of bases $\boldsymbol{U}^i$ and therefore a finite number of values for the T^i.

Define the set of all values of $\{T^1, T^2, T^3, ..., T^\gamma\}$ and also define the coresponding set of $\{\boldsymbol{U}^1, \boldsymbol{U}^2, \boldsymbol{U}^3, \dots, \boldsymbol{U}^\gamma\}$. Similarly for the second class of data points and the corresponding set of subspaces define the set of values of the traces $\{S^1, S^2, S^3, ..., S^\gamma\}$ and also define the corespond-

ing set of subspace basis $\{\boldsymbol{V}^1, \boldsymbol{V}^2, \boldsymbol{V}^3, \ldots, \boldsymbol{V}^\gamma\}$ where $S^j \equiv \text{tr}(\boldsymbol{V}^{jT} Y Y^T \boldsymbol{V}^j)$.

Let n and $n+1$ be two consecutive iterations. Then the objective function at each iteration r is given by

$$\begin{aligned} F_r &= T^{i_r} + S^{j_r} + M^{i_r j_r} \quad \text{where} \\ T^{i_r} &\equiv \text{tr}(\boldsymbol{U}^{i_r T} X X^T \boldsymbol{U}^{i_r}) \\ S^{j_r} &\equiv \text{tr}(\boldsymbol{V}^{j_r T} Y Y^T \boldsymbol{V}^{j_r}) \\ M^{i_r j_r} &\equiv \text{tr}(\boldsymbol{U}^{i_r T} \boldsymbol{V}^{j_r} \boldsymbol{V}^{j_r T} \boldsymbol{U}^{i_r}) \end{aligned} \tag{2.22}$$

For $r = n$ we fix $j_r = j_n$ (that is we fix S^{j_n}) and find i_n that maximizes

$$F_n = T^{i_n} + S^{j_n} + M^{i_n j_n} \tag{2.23}$$

Effectively, we solve

$$\arg \max_{i_n \in \{1,\ldots,\gamma\}} \{T^{i_n} + M^{i_n j_n} | j_n = constant\} \tag{2.24}$$

For $r = n+1$ we fix $i_r = i_{n+1} = i_n$ (that is we fix $T^{i_{n+1}} = T^{i_n}$) and find j_{n+1} that maximizes

$$\begin{aligned} F_{n+1} &= T^{i_{n+1}} + S^{j_{n+1}} + M^{i_{n+1} j_{n+1}} \\ &= T^{i_n} + S^{j_{n+1}} + M^{i_n j_{n+1}} \end{aligned} \tag{2.25}$$

Effectively, we solve

$$\arg \max_{j_{n+1} \in \{1,\ldots,\gamma\}} \{S^{j_{n+1}} + M^{i_n j_{n+1}} | i_n = constant\} \tag{2.26}$$

Therefore,

$$\begin{aligned} F_{n+1} - F_n &= (T^{i_n} + S^{j_{n+1}} + M^{i_n j_{n+1}}) - (T^{i_n} + S^{j_n} + M^{i_n j_n}) \\ &= (S^{j_{n+1}} - S^{j_n}) + (M^{i_n j_{n+1}} - M^{i_n j_n}) \end{aligned} \tag{2.27}$$

Since $j_n \in \{1, \ldots, \gamma\}$ then $S^{j_{n+1}} + M^{i_n j_{n+1}}$ and $S^{j_n} + M^{i_n j_n}$ are terms of the same sequence.

Therefore from (2.26) we have that

$$S^{j_{n+1}} + M^{i_n j_{n+1}} > S^{j_n} + M^{i_n j_n} \tag{2.28}$$

From (2.27) and (A.4) we get that $F_{n+1} - F_n > 0$. Hence $F_{n+1} > F_n$.

Therefore, the sequence $\{F_n\}$ is increasing.

Since $\{\boldsymbol{U}^i\}$ and $\{\boldsymbol{V}^i\}$ are finite sets then $\{T^i\}$, $\{S^j\}$ and $\{M^{ij}\}$ are also finite sets. Therefore, $\{T^i\}$, $\{S^j\}$ and $\{M^{ij}\}$ have maxima. Since each element of $\{F_n\}$ is a linear combination of elements from $\{T^i\}$, $\{S^j\}$ and $\{M^{ij}\}$ then $\{F_n\}$ also has a maximum. That means $\{F_n\}$ is bounded from above.

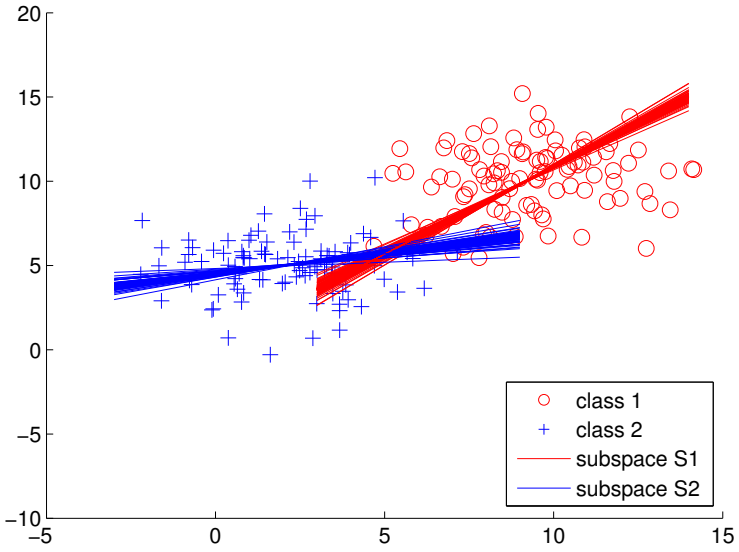
We have proven that $\{F_n\}$ is an increasing sequence of real numbers and also bounded from above. Therefore, it converges.

□

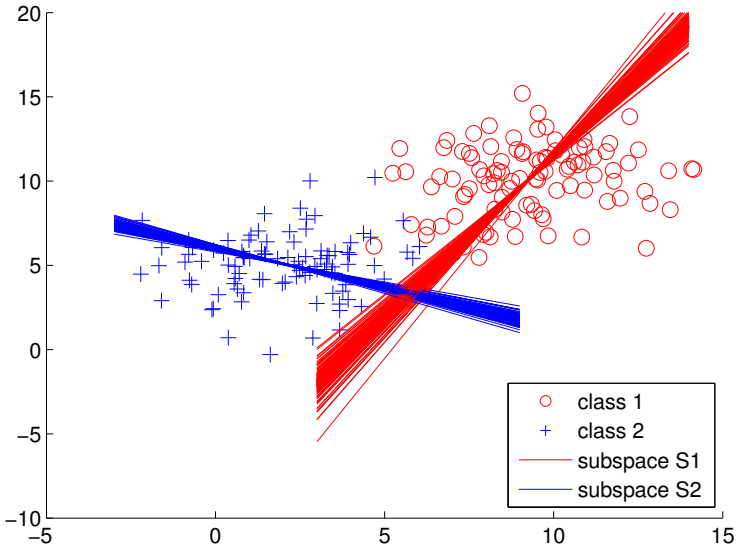
We consider two examples here showing the effect of changing the relative angle between subspaces generated by LSC. The datasets are generated from two bivariate normal distributions $\mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ representing classes $\mathcal{C}_1$ and $\mathcal{C}_2$. Each class consists of 100 randomly generated points from $\mathcal{N}_1$ and $\mathcal{N}_2$ respectively. The parameters of $\mathcal{N}_1$ and $\mathcal{N}_2$ for the two classes are shown in Table 3.1. The LSC and CSC are trained on the data with k = 1. The values of Z, tol_f^m, $tol_{\boldsymbol{U}_1}^m$ and $tol_{\boldsymbol{U}_2}^m$ are chosen to be 2000, 1e-6, 1e-6 and 1e-6 respectively. The value of C is set to -10^3 for example 1 and 10^3 for example 2. The classification accuracies are obtained via leave-one-out cross validation (LOOCV) (Kohavi et al., 1995). The subspaces obtained for each of the training folds in example 1 and example 2 are shown in Figures 2.1 and 2.2 respectively. The average classification accuracies and the average relative angle θ ($0 \leq \theta \leq \pi/2$) between the subspaces for LSC and CSC are reported in Table 2.1. In example 1, increasing the relative angle between the subspaces clearly improves the classification accuracy by $\approx 24\%$. However in example 2, decreasing the relative angle between the subspaces shows better classification performance and outperforms LSC by $\approx 11\%$. These examples show that the relative orientation of the subspaces should also be considered in addition to capturing the *maximal* variance in data.

Table 2.1: Average classification accuracies and relative angle between subspaces generated from LSC and CSC in two examples.

DATASETS	$\mathcal{N}_1$		$\mathcal{N}_2$		LSC		CSC	
	$\boldsymbol{\mu}_1$	$\boldsymbol{\Sigma}_1$	$\boldsymbol{\mu}_2$	$\boldsymbol{\Sigma}_2$	ACC(%)	ANGLE(θ)	ACC(%)	ANGLE(θ)
EXAMPLE 1	$\begin{bmatrix} 9 \\ 10 \end{bmatrix}$	$\begin{bmatrix} 4 & 1.1 \\ 1.1 & 4 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 5 \end{bmatrix}$	$\begin{bmatrix} 4 & 0 \\ 0 & 3 \end{bmatrix}$	74	0.54	**92**	0.99
EXAMPLE 2	$\begin{bmatrix} 3 \\ 5 \end{bmatrix}$	$\begin{bmatrix} 4 & -2 \\ -2 & 6 \end{bmatrix}$	$\begin{bmatrix} 10 \\ 10 \end{bmatrix}$	$\begin{bmatrix} 5 & 2 \\ 2 & 5 \end{bmatrix}$	87	0.92	**97**	0.16

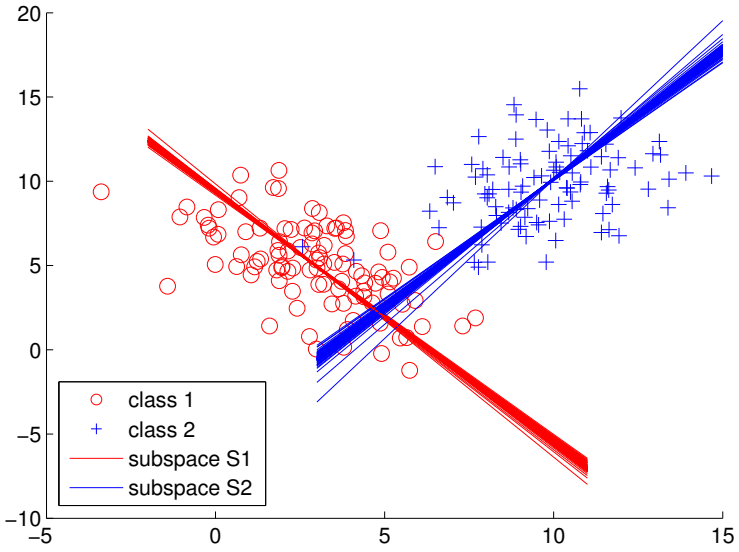


(a) LCS

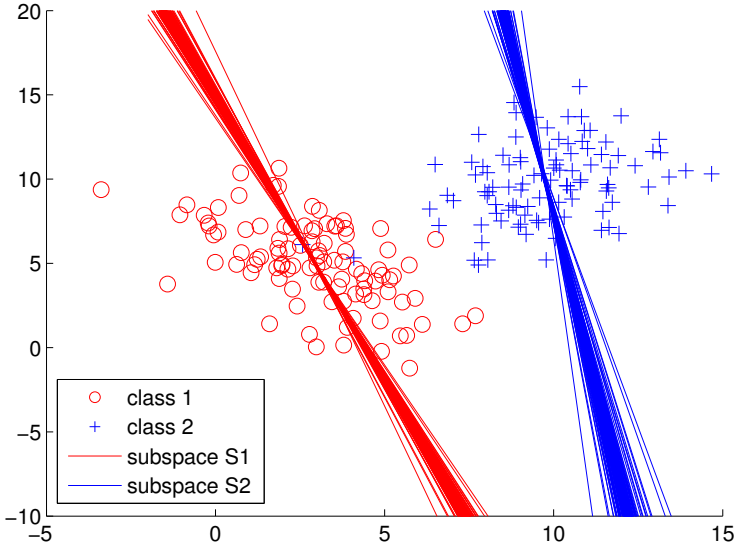


(b) CSC

Figure 2.1: Data points generated by $\mathcal{N}_1$ and $\mathcal{N}_2$ in example 1 and the subspaces generated by LSC and CSC in each of the training folds.



(a) LCS



(b) CSC

Figure 2.2: Data points generated by $\mathcal{N}_1$ and $\mathcal{N}_2$ in example 2 and the subspaces generated by LSC and CSC in each of the training folds.

The performance of CSC is evaluated on six publicly available datasets and they are summarized in Table 2.2. Four of them (DLBCL, Breast, Colon, DBWorld) are high dimensional (#features>> #samples) and two of them (Mushroom, Spambase) have significantly more samples than features.

Table 2.2: Summary table of datasets used for experiments. The first four datasets, namely DLBCL, Breast, Colon and DBWorld are high dimensional since the number of features greatly outnumbers the number of samples. The last two (Mushroom and Spambase) are not high dimensional since the number of samples is greater than the number of features.

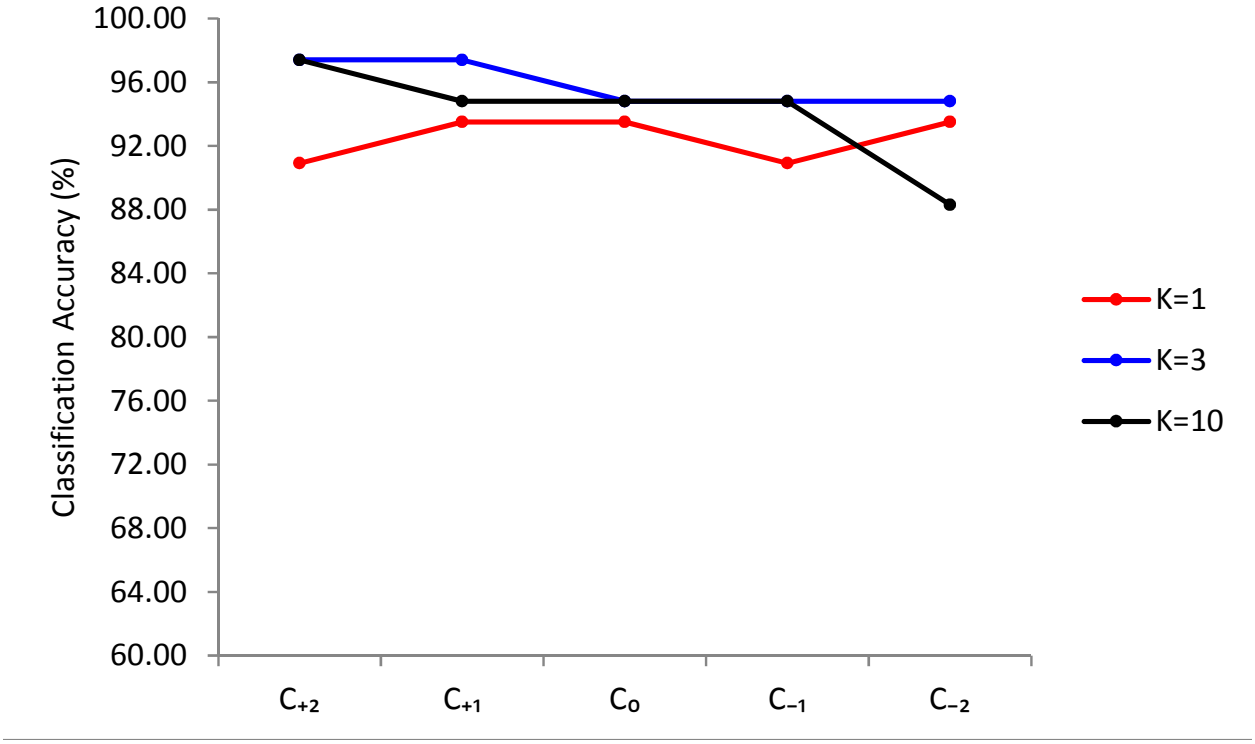
Dataset	**Reference**	**#Samples**	**#Features**
DLBCL	(Shipp et al., 2002)	77	5469
Breast	(Van't Veer et al., 2002)	77	4869
Colon	(Alon et al., 1999)	62	2000
DBWorld	(Hassell & Arpinar, 2006)	64	4702
Mushroom	(Satsangi & Zaiane, 2007)	8124	126
Spambase	(Lee et al., 2010)	4601	57

The performance of CSC is evaluated for different values of C, and compared to that of LSC. The values of C are chosen in such a way that the relative angle between the subspaces varies uniformly. The relative angle between the subspaces is evaluated in terms of the projection metric d_{pF}. The value of d_{pF} varies between 0 and k, where k is the dimensionality of the subspaces. The value of k is chosen as $\{1, 3, 10\}$. Experiments are performed with a 2.60GHz Intel Core i5 CPU running OS X with 8.0 GB of main memory. The classification performance is evaluated using LOOCV technique.

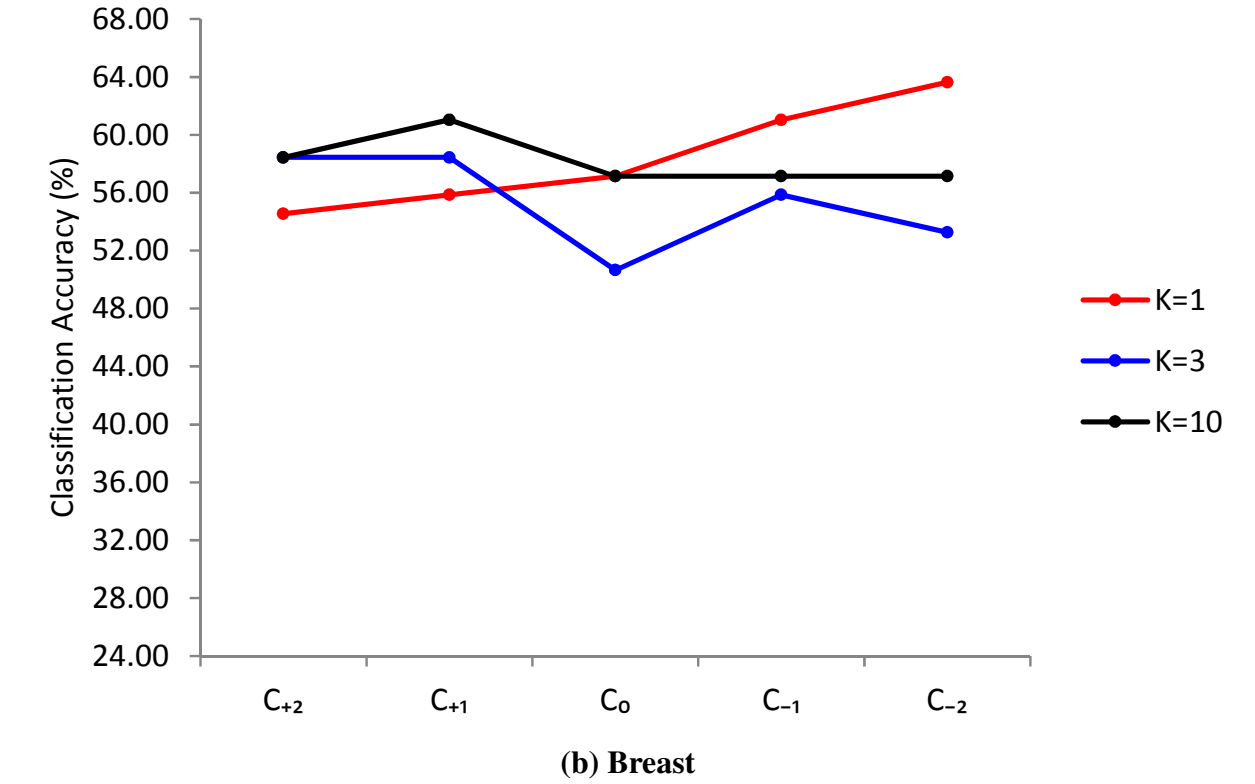
The classification accuracies as a function of C for different values of k are shown in Figures 2.3 - 2.5. C_0 represents the results of LSC since for $C = 0$ the CSC reduces to LSC. C_{-1}, C_{-2} correspond to $C < 0$ and C_{+1}, C_{+2} correspond to $C > 0$. As mentioned earlier, positive values of C decrease the relative angle between the subspaces while negative values of C increase the relative angle. The values of Z, tol_f^m, $tol_{\boldsymbol{U}_1}^m$ and $tol_{\boldsymbol{U}_2}^m$ are chosen to be 2000, 1e-6, 1e-6 and 1e-6 respectively.

For DLBCL and Colon datasets, classification accuracy is improved by reducing the relative angle between subspaces for $k = 3$, $k = 10$ and $k = 1$, $k = 3$ respectively. In the case of Breast dataset, increasing the relative angle for $k = 1$ considerably improves the classification accuracy. For the DBWorld dataset the classification accuracy of CSC was almost identical to that of LSC.

With respect to the lower dimensional datasets, CSC performed at least as good as LSC. In the case of Spambase dataset, CSC was able to slightly increase the accuracy of classification for positive values of C. The penalty parameter C gives the flexibility to adjust.

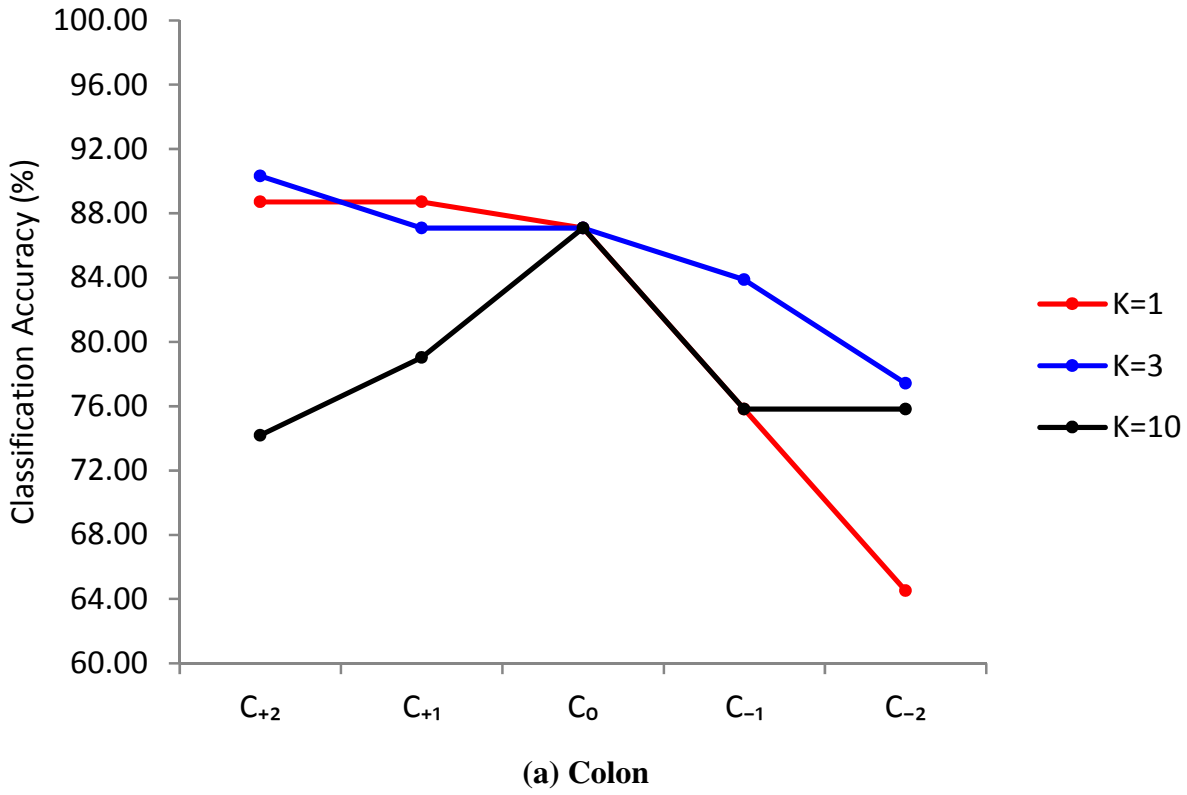


(a) DLBCL

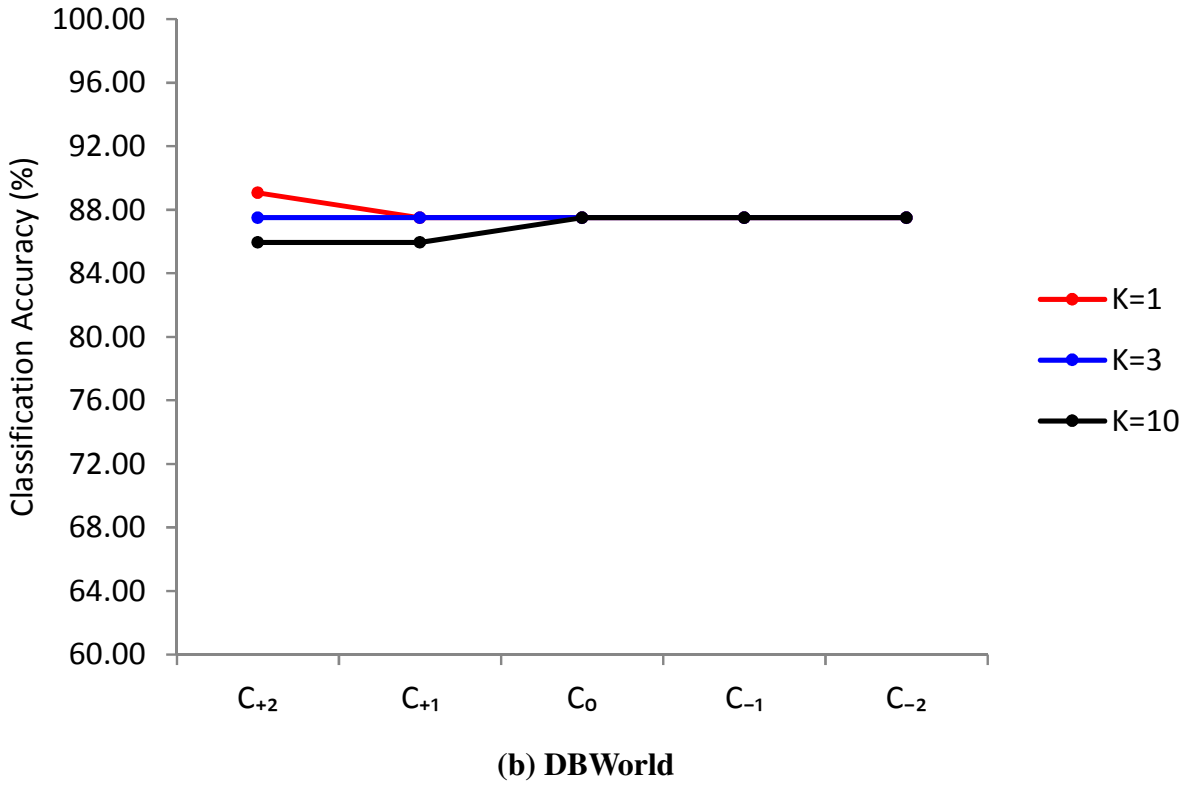


(b) Breast

Figure 2.3: Classification accuracy for (a) DLBCL and (b) Breast datasets



(a) Colon



(b) DBWorld

Figure 2.4: Classification accuracy for (a) Colon and (b) DBWorld

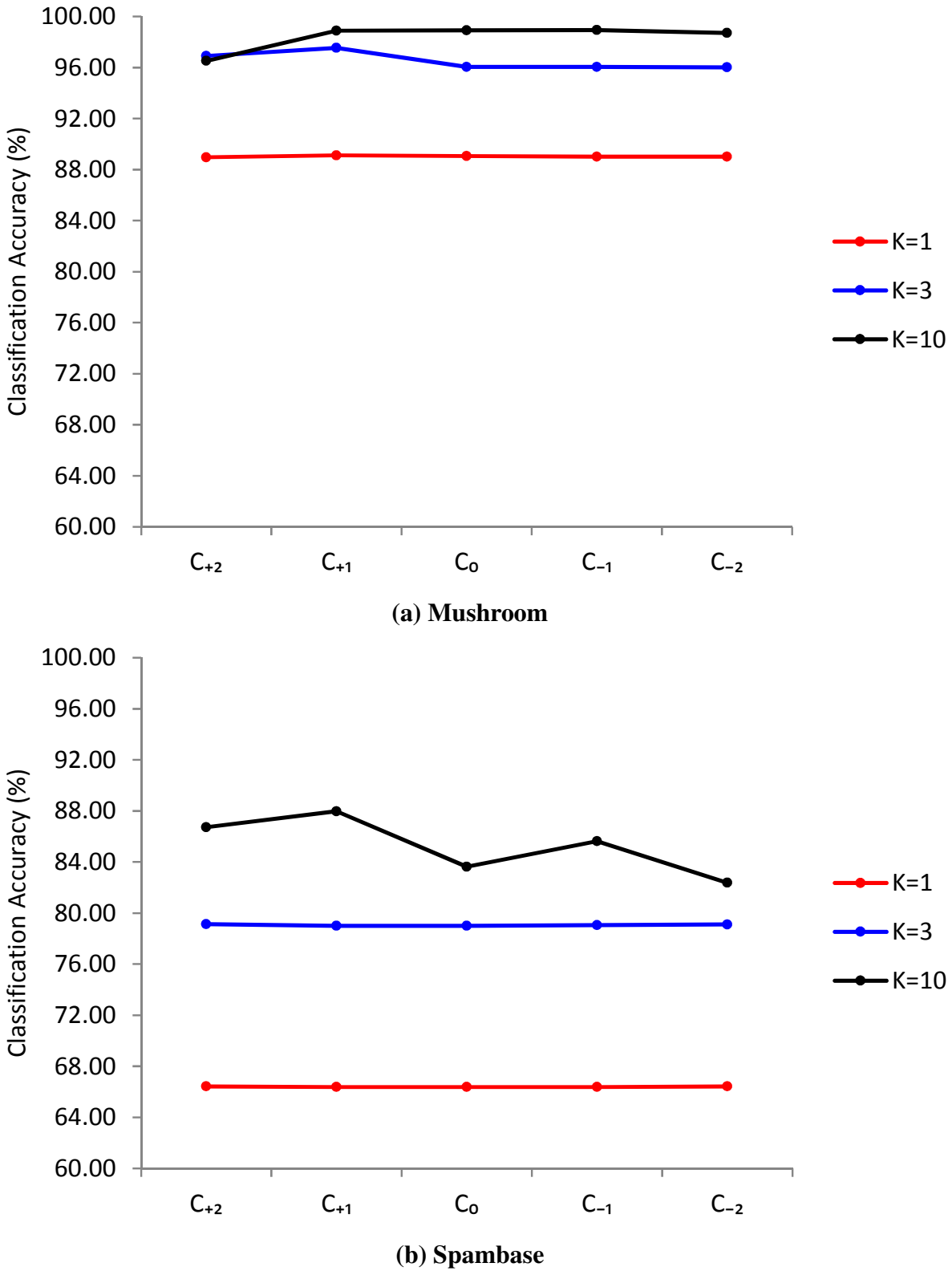


Figure 2.5: Classification accuracy for (a) Mushroom and (b) Spambase

We provide the comparative computational results for CSC against SVM , PCA/SVM and Naive Bayes classifier summarized in Table 2.3. PCA was used to reduce the dimensionality of the datasets prior to SVM classification. Through PCA, components that correspond to 80% of the total variance were used for classification. The contributed variance of the factors maintained exceed the 70% threshold (Stevens, 2012) due to the relative small amount of samples compared to the number of features of the data. SVM was trained using a Radius Basis Function (RBF) kernel. The pair of parameter settings (k, C) used in CSC in each dataset was: DLBCL (3, 2E+10), Breast (1, -5E+03), Colon (3, 5E+09), DBWorld (1, 2E+03), Mushroom(10, -1E+03) and Spambase (10, 5E+03). Naive Bayes classifier shows the lowest overall accuracy. CSC demonstrates competitive behavior with respect to dataset dimensionality. The performance of SVM degrades in high dimensional datasets and the combined use of PCA/SVM does not perform well as the number of features decreases. However, CSC remains robust although it does not necessarily achieve the highest accuracy in every experiment.

Table 2.3: Computational comparisons with corresponding classification accuracies Acc(%). Naive Bayes demonstrates the lowest overall accuracy. Performance of SVM degrades in high dimensional datasets. PCA/SVM does not perform well as the number of features decreases. CSC remains robust although it does not necessarily achieve the highest accuracy in every experiment. Parameter settings k, C of CSC also appear on the table.

Dataset	**SVM**	**PCA/SVM**	**Naive Bayes**	**CSC**	k	C
DLBCL	94.8	97.5	75	97.4	3	2E+10
Breast	68	68	62.5	63.6	1	-5E+03
Colon	75.9	92.1	71.4	90.3	3	5E+09
DBWorld	88	88	57.1	89	1	2E+03
Mushroom	100	100	88.1	98.9	10	-1E+03
Spambase	91	66	56.3	87.9	10	5E+03

Remarks

A new classification algorithm, called constrained subspace classifier, was proposed and designed for high dimensional datasets. We have shown that the proposed algorithm outperforms LSC. In addition to approximating the classes well by individual subspaces, CSC also accounts for the relative angle between the subspaces by utilizing the projection metric. An efficient alternating optimization technique is also proposed. CSC has been evaluated on publicly available datasets and is compared to LSC. The improvement in classification accuracy shows the importance of considering the relative angle between subspaces while approximating the classes. Additionally, CSC seems to be effective when introduced for lower dimensional subspaces. The robust nature of CSC reveals that it can serve as a one step method for for preprocessing-free classification. To this end, CSC presents an advantage over other popular models for high dimensional binary classification.

Potential future research directions include a *cost sensitive* version for *imbalanced classification* problems where the sample numbers of one class greatly outperform the samples of the other. Imbalanced classification problems are common in many business analytics areas (Razzaghi, Otero, & Xanthopoulos, 2014) and in quality control (Xanthopoulos & Razzaghi, 2014). In this setup one of the most popular cost sensitive algorithmic schemes is SVM; however, it is well known that it does not perform well for such large number of features. Therefore, alternative algorithms able to simultaneusly handle high dimensional datasets and the problem of imbalanced classes are particularly useful for a number of applications.

Another extension is the development of the stream mining version that will incrementally retrain as new training data samples arrive in the form of a data stream. Incremental learning is useful in cases where the full retraining of a model is not desired. Such extensions have been proposed for generic SVM (Cauwenberghs & Poggio, 2001; Diehl & Cauwenberghs, 2003) and other classifiers (Pang, Ozawa, & Kasabov, 2005; Guarracino, Cuciniello, & Feminiano, 2009; Cifarelli,

Guarracino, Seref, Cuciniello, & Pardalos, 2007; Dulá & López, 2013).

Lastly, a robust optimization version of this algorithm needs to be proposed for handling datasets that are inexact or uncertain. In these cases the *robust counterpart* of the optimization problem needs to be defined and the solution corresponds to the worst case realization of the uncertain data (Xanthopoulos, Pardalos, & Trafalis, 2012; Xanthopoulos, Guarracino, & Pardalos, 2014).

CHAPTER 3: RELAXED SUPPORT VECTOR REGRESSION

Datasets with outliers pose a serious challenge in regression analysis. In this section, a new regression method called *relaxed support vector regression (RSVR)* is proposed for such datasets. RSVR is based on the concept of constraint relaxation which leads to increased robustness in datasets with outliers. RSVR is formulated using both linear and quadratic loss functions. Numerical experiments on benchmark datasets and computational comparisons with other popular regression methods depict the behavior of our proposed method. RSVR achieves better overall performance than *support vector regression (SVR)* in measures such as RMSE (Levinson, 1947) and R^2_{adj} (Theil, 1959) while being on par with other state-of-the-art regression methods such as *robust regression* (RR). Additionally, RSVR provides robustness for higher dimensional datasets which is a limitation of RR, the robust equivalent of *ordinary least squares* regression. Moreover, RSVR can be used on datasets that contain varying levels of noise.

Support Vector Regression

We develop RSVR by first introducing SVR and then incorporating the concept of free slack.

Let $S = \{(\mathbf{x}_i, y_i)\},\ \mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R} \quad \forall i = 1, \ldots, n$ be a training set.

Define the ϵ-insensitive loss function as

$$l(y_i, f(\mathbf{x}_i)) = \begin{cases} 0 & \text{if } |f(\mathbf{x}_i) - y_i| \leq \epsilon \\ |f(\mathbf{x}_i) - y_i| - \epsilon & \text{otherwise} \end{cases} \tag{3.1a}$$

Support vector regression minimizes the ϵ-insensitive loss function regularized by the L_2-norm

(Smola & Schölkopf, 2004). In particular it solves the following optimization problem,

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} l(y_i, f(\mathbf{x}_i)) \tag{3.2}$$

were C is a positive regularization parameter and $f(\mathbf{x}_i) = \langle \mathbf{w}, \mathbf{x}_i \rangle + b$ is the desired linear classifier (V. Vapnik, 2000).

Since the absolute value function has a kink at the origin we further modify (3.2) by introducing slack variables ξ_i and $\overline{\xi}_i,\ \forall i = 1, \ldots, n$. It becomes,

$$\min_{\mathbf{w},b,\xi} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \left(\xi_i + \overline{\xi}_i\right) \tag{3.3a}$$

$$\text{s.t. } -\langle \mathbf{w}, \mathbf{x}_i \rangle - b + y_i \leq \epsilon + \xi_i, \qquad \forall i \in 1, \ldots, n \tag{3.3b}$$

$$\langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \leq \epsilon + \overline{\xi}_i, \qquad \forall i \in 1, \ldots, n \tag{3.3c}$$

$$\xi_i \geq 0, \qquad \forall i \in 1, \ldots, n \tag{3.3d}$$

$$\overline{\xi}_i \geq 0, \qquad \forall i \in 1, \ldots, n \tag{3.3e}$$

The minimization of the ϵ-insensitive loss function results to the formation of an ϵ-tube around the regression line. SVR can become non-linear through a transformation $\Phi : \mathbb{R}^d \to \mathcal{H}$, such that $\Phi(\mathbf{x}_i) \in \mathcal{H}$, where $\mathcal{H}$ is a reproducing kernel Hilbert space with $\dim(\mathcal{H}) > \dim(\mathbb{R}^d)$ and Φ is the kernel transformation which is a standard methodology for employing linear machine learning tools in non-linear data.

The response of a new datapoint $\boldsymbol{x}$ is determined as,

$$\text{response}(\boldsymbol{x}) = \langle \mathbf{w}^*, \Phi(\boldsymbol{x}) \rangle + b^* \tag{3.4}$$

where $\mathbf{w}^*$ is the optimal weight vector of the regression hyperplane and b^* is the corresponding bias term.

Quadratic Loss Function Formulation

We use the concept of relaxed support vector machines (RSVM) (Şeref, Chaovalitwongse, & Brooks, 2014) to relax support vector regression. SVR is modified accordingly to produce our proposed model, relaxed support vector regression. We start off by formulating RSVR with a quadratic loss function as follows,

$$\min_{\mathbf{w},b,\xi,\overline{\xi},v,\overline{v}} \ \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle + \frac{C}{2}\sum_{i=1}^{n}\left({\xi_i}^2 + {\overline{\xi}_i}^2\right) \tag{3.5a}$$

$$\text{s.t.} \ \ -\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle - b + y_i \leq \epsilon + \xi_i + v_i, \qquad \forall i \in 1,\ldots,n \tag{3.5b}$$

$$\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b - y_i \leq \epsilon + \overline{\xi}_i + \overline{v}_i, \qquad \forall i \in 1,\ldots,n \tag{3.5c}$$

$$\sum_{i=1}^{n}(v_i + \overline{v}_i) \leq n\ \Upsilon \tag{3.5d}$$

$$v_i \geq 0, \qquad \forall i \in 1,\ldots,n \tag{3.5e}$$

$$\overline{v}_i \geq 0, \qquad \forall i \in 1,\ldots,n \tag{3.5f}$$

where Υ is controlled by the user and determines the average amount of free slack distributed to each sample.

The Lagrangian function for Formulation (3.5) can be written as,

$$\mathcal{L}(\mathbf{w}, \xi, \overline{\xi}, b, \epsilon, \alpha^+, \alpha^-, \beta, \lambda, \overline{\lambda}, v, \overline{v}) =$$
$$\frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + \frac{C}{2}\sum_{i=1}^{n}\left({\xi_i}^2 + {\overline{\xi}_i}^2\right) - \sum_{i=1}^{n}{\alpha_i}^+\left(\epsilon + \xi_i + v_i + \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b - y_i\right)$$
$$- \sum_{i=1}^{n}{\alpha_i}^-\left(\epsilon + \overline{\xi}_i + \overline{v_i} - \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle - b + y_i\right) - \beta\left(n\Upsilon - \sum_{i=1}^{n}(v_i + \overline{v}_i)\right)$$
$$- \sum_{i=1}^{n}\lambda_i v_i - \sum_{i=1}^{n}\overline{\lambda}_i\overline{v}_i \tag{3.6a}$$

where α^+, α^-, β, λ and $\overline{\lambda}$ are the Lagrange multipliers. Since (3.5) is a convex problem, its Wolfe dual can be obtained from the following stationary first order conditions of the primal variables $\mathbf{w}$, b, ξ, $\overline{\xi}$, v and $\overline{v}$.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n}\left({\alpha_i}^+ - {\alpha_i}^-\right)\Phi(\mathbf{x}_i) = 0 \tag{3.7a}$$
$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^{n}\left({\alpha_i}^- - {\alpha_i}^+\right) = 0 \tag{3.7b}$$
$$\frac{\partial \mathcal{L}}{\partial \xi_k} = C\xi_k - {\alpha_k}^+ = 0, \qquad \forall k = 1, 2, \ldots n \tag{3.7c}$$
$$\frac{\partial \mathcal{L}}{\partial \overline{\xi}_k} = C\overline{\xi}_k - {\alpha_k}^- = 0, \qquad \forall k = 1, 2, \ldots n \tag{3.7d}$$
$$\frac{\partial \mathcal{L}}{\partial v_k} = -{\alpha_k}^+ + \beta - \lambda_k = 0, \qquad \forall k = 1, 2, \ldots n \tag{3.7e}$$
$$\frac{\partial \mathcal{L}}{\partial \overline{v}_k} = -{\alpha_k}^- + \beta - \overline{\lambda}_k = 0, \qquad \forall k = 1, 2, \ldots n \tag{3.7f}$$

By substituting the equivalent expressions for $\mathbf{w}$, b, ξ, $\overline{\xi}$, v and $\overline{v}$ from equations (3.7a) - (3.7f) back in expression (3.6), the Wolfe dual of (3.5) is derived as,

$$\max_{\alpha^+,\alpha^-,\beta} \; -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\alpha_i{}^+ - \alpha_i{}^-\right)\left(\alpha_j{}^+ - \alpha_j{}^-\right)\left(K(\mathbf{x}_i,\mathbf{x}_j) + \frac{\delta_{ij}}{C}\right)$$

$$-\epsilon\sum_{i=1}^{n}\left(\alpha_i{}^+ + \alpha_i{}^-\right) + \sum_{i=1}^{n} y_i\left(\alpha_i{}^+ - \alpha_i{}^-\right) - \beta n\Upsilon \tag{3.8a}$$

$$\text{s.t.} \;\; \sum_{i=1}^{n}\left(\alpha_i{}^+ - \alpha_i{}^-\right) = 0 \tag{3.8b}$$

$$0 \le \alpha_i{}^+ \le \beta, \qquad \forall i \in 1,\ldots,n \tag{3.8c}$$

$$0 \le \alpha_i{}^- \le \beta, \qquad \forall i \in 1,\ldots,n \tag{3.8d}$$

where α^+, α^- and β are the corresponding Lagrange multipliers.

We present some theoretical results about the free slack variable values at optimality.

Lemma 1. *At optimality, if* $||w|| > 0$*, then the "total free slack" constraint,* $\sum_{i=1}^{n}(v_i + \overline{v}_i) \le n\Upsilon$*, is always binding, i.e., the total free slack is always consumed* (3)*.*

The proof follows from the Karush-Kuhn-Tucker complementary slackness condition $\beta\left(n\Upsilon - \sum_{i=1}^{n}(v_i + \overline{v}_i)\right) = 0$. Using the fact that $||w|| > 0$ and constraints (3.8c) and (3.8d), equation (3.7a) implies that $\beta > 0$. Thus, constraint (3.5d) is binding, i.e., the total free slack amount $n\Upsilon$ is always consumed completely.

Theorem 3. *Let* $\xi_{\max} = \max_{i\in 1,\ldots,n}\{\xi_i\}$ *and* $\overline{\xi}_{\max} = \max_{i\in 1,\ldots,n}\{\overline{\xi}_i\}$ *be the maximum penalties for all samples that lie 'above' the optimal hyperplane of* ϵ*-tube and all the samples that lie 'below' it, respectively, in the optimum solution to formulation* (3.5)*. Then,*

1. $v_i = 0$ *for any sample such that* $\xi_i < \xi_{\max}$*, and* $\overline{v}_i = 0$ *for any sample such that* $\overline{\xi}_i < \overline{\xi}_{\max}$*,* $\forall i \in 1,\ldots,n$*, and*

2. $\xi_i = \xi_{\max}$ *for* $v_i > 0$*, and* $\overline{\xi}_i = \overline{\xi}_{max}$ *for* $\overline{v}_i > 0$*,* $\forall i \in 1,\ldots,n$*.*

Proof. Let $V = \Big\{ i |\, i \in 1, \ldots, n \Big\}$ or equivalently $V = \Big\{ i |\, \xi_i \leq \xi_{\max}, \forall i \in 1, \ldots, n \Big\}$. Without loss of generality, assume that $\exists i \in V$ with x_i in the optimal solution to formulation (3.5) such that $\xi_i < \xi_{\max}$ and $v_i > 0$. Let $\overline{V} \subset V$ be defined by $\overline{V} = \Big\{ i |\, \xi_i < \xi_{\max}, \text{ for some } i \in 1, \ldots, n \Big\}$. Let the penalty difference between the x_i samples with i $\in \overline{V}$ and the next highest penalty be $\delta = \xi_{\max} - \xi_{i^*}$, where $i^* = \arg\max \Big\{ \xi_i |\, i \in V \setminus \overline{V} \Big\}$. From Lemma 1, one can shift the amount of free slack from sample x_i over to samples $x_{i'}$ with $i' \in \overline{V}$ given by $\delta_{min} = \min\{v_i + \overline{v}_i, \delta\}$. For samples above the hyperplane $\overline{v}_i = 0$, hence $\delta_{min} = \min\{v_i, \delta\}$. Then the new penalty values for samples x_i with $i \in \overline{V}$ are $\xi_i' = \xi_i - \delta_i$, where $\delta_{min} = \sum_{i \in \overline{V}} \delta_i$. Let ΔZ be the reduction in the total penalty. Then,

$$\Delta Z = |\overline{V}|\, {\xi_{\max}}^2 + {\xi_i}^2 - \left(\textstyle\sum_{i \in \overline{V}} (\xi_{\max} - \delta_i)^2 + (\xi_i + \delta_{\min})^2 \right) \tag{3.9a}$$

$$= 2\delta_{min}(\xi_{max} - \xi_i) - \delta_{min}^2 - \sum_{i \in \overline{V}} \delta_i^2 \tag{3.9b}$$

$$\geq 2\delta_{min}(\xi_{max} - (\xi_i + \delta_{min})) > 0, \tag{3.9c}$$

which contradicts with the optimality of the solution. The reduction in (3.9b) is maximized when $\delta_i = \frac{\delta_{\min}}{|\overline{V}|} \, \forall i \in \overline{V}$ with a new maximum penalty value ${\xi_{\max}}' = \xi_{\max} - \frac{\delta_{\min}}{|\overline{V}|}$ From Lemma 1, the corresponding new free slack values are ${v_i}' = v_i + \frac{\delta_{min}}{|\overline{V}|} \, \forall i \in \overline{V}$. This shows that all free slack is consumed by the samples with maximum penalty, thus proving item (1) for all samples that lie 'above' the optimal hyperplane. The proof for the samples that lie 'below' the optimal hyperplane is similar. □

Optimal Hyperplane Parameters

The solution to (3.8) is used to evaluate,

$$\mathbf{w}^* = \sum_{j=1}^{n} \left({\alpha_j}^+ - {\alpha_j}^- \right) \Phi(\mathbf{x}_j) \tag{3.10a}$$

Let $S^+ = \left\{ {\alpha_i}^+ | \, (0 < {\alpha_i}^+ < \beta) \right\}, I^+ = \left\{ i | \, {\alpha_i}^+ \in S^+ \right\}$.
Let $S^- = \left\{ {\alpha_i}^- | \, (0 < {\alpha_i}^- < \beta) \right\}, I^- = \left\{ i | \, {\alpha_i}^- \in S^- \right\}$.

The bias can be computed as,

$$b^* = \frac{1}{|S^+|} \sum_{i \in I^+} \left(-\epsilon - \frac{{\alpha_i}^+}{C} - \sum_{j=1}^{n} \left({\alpha_j}^+ - {\alpha_j}^- \right) K(\mathbf{x}_i, \mathbf{x}_j) + y_i \right) + \frac{1}{|S^-|} \sum_{i \in I^-} \left(\epsilon + \frac{{\alpha_i}^-}{C} - \sum_{j=1}^{n} \left({\alpha_j}^+ - {\alpha_j}^- \right) K(\mathbf{x}_i, \mathbf{x}_j) + y_i \right) \tag{3.11a}$$

The response of a new datapoint $\boldsymbol{x}$ is determined from (4.2).

Linear Loss Function Formulation

For the linear loss function formulation, since the penalty term is not squared, Theorem (3) does not necessarily hold. That implies that it is possible to shift free slack from one sample to another with the total penalty staying the same. To deal with that, we introduce variables s and $\overline{s}$ such that $s \geq \xi_i$ and $\overline{s} \geq \overline{\xi}_i, \ \forall i \in 1, \ldots, n$. By doing that, the amount of free slack that has been distributed to samples with low penalties will be transfered to those with higher penalties. That is, the s and $\overline{s}$ are going to be driven to lower values, which will effectively lead to a solution where the maximum penalty is as low as possible. The linear loss function formulation of RSVR is,

$$\min_{\mathbf{w},b,\xi,\overline{\xi},v,\overline{v},s,\overline{s}} \quad \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle + C\left(\sum_{i=1}^{n}\left(\xi_i + \overline{\xi}_i\right) + s + \overline{s}\right) \tag{3.12a}$$

$$\text{s.t.} \quad -\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle - b + y_i \leq \epsilon + \xi_i + v_i, \quad \forall i \in 1, \ldots, n \tag{3.12b}$$

$$\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b - y_i \leq \epsilon + \overline{\xi}_i + \overline{v}_i, \quad \forall i \in 1, \ldots, n \tag{3.12c}$$

$$\sum_{i=1}^{n}\left(v_i + \overline{v}_i\right) \leq n\ \Upsilon \tag{3.12d}$$

$$v_i \geq 0, \quad \forall i \in 1, \ldots, n \tag{3.12e}$$

$$\overline{v}_i \geq 0, \quad \forall i \in 1, \ldots, n \tag{3.12f}$$

$$s \geq \xi_i \geq 0, \quad \forall i \in 1, \ldots, n \tag{3.12g}$$

$$\overline{s} \geq \overline{\xi}_i \geq 0, \quad \forall i \in 1, \ldots, n \tag{3.12h}$$

where Υ is controlled by the user.

The Lagrangian function for Formulation (3.12) can be written as,

$$\begin{aligned}
&\mathcal{L}(\mathbf{w}, \xi, \overline{\xi}, b, \epsilon, \alpha^+, \alpha^-, \beta, \gamma, \overline{\gamma}, \delta, \overline{\delta}, \lambda, \overline{\lambda}, v, \overline{v}, s, \overline{s}) = \\
&\frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle + C\left(\sum_{i=1}^{n}\left(\xi_i + \overline{\xi}_i\right) + s + \overline{s}\right) - \sum_{i=1}^{n}{\alpha_i}^+\left(\epsilon + \xi_i + v_i + \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b - y_i\right) \\
&- \sum_{i=1}^{n}{\alpha_i}^-\left(\epsilon + \overline{\xi}_i + \overline{v}_i - \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle - b + y_i\right) - \beta\left(n\Upsilon - \sum_{i=1}^{n}\left(v_i + \overline{v}_i\right)\right) - \sum_{i=1}^{n}\lambda_i v_i \\
&- \sum_{i=1}^{n}\overline{\lambda}_i\overline{v}_i - \sum_{i=1}^{n}\gamma_i\xi_i - \sum_{i=1}^{n}\overline{\gamma}_i\overline{\xi}_i - \sum_{i=1}^{n}\delta_i\left(s - \xi_i\right) - \sum_{i=1}^{n}\overline{\delta}_i\left(\overline{s} - \overline{\xi}_i\right)
\end{aligned} \tag{3.13a}$$

where α^+, α^-, β, γ, $\overline{\gamma}$, δ, $\overline{\delta}$, λ and $\overline{\lambda}$ are the Lagrange multipliers. Since (3.12) is a convex problem, its Wolfe dual can be obtained from the following stationary first order conditions of the primal

variables $\mathbf{w}$, b, ξ, $\overline{\xi}$, s, $\overline{s}$, v and $\overline{v}$.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \left({\alpha_i}^{+} - {\alpha_i}^{-} \right) \Phi(\mathbf{x}_i) = 0 \tag{3.14a}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^{n} \left({\alpha_i}^{-} - {\alpha_i}^{+} \right) = 0 \tag{3.14b}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_k} = C - {\alpha_k}^{+} - \gamma_k + \delta_k = 0, \qquad \forall k = 1, 2, \ldots n \tag{3.14c}$$

$$\frac{\partial \mathcal{L}}{\partial \overline{\xi}_k} = C - {\alpha_k}^{-} - \overline{\gamma}_k + \overline{\delta}_k = 0, \qquad \forall k = 1, 2, \ldots n \tag{3.14d}$$

$$\frac{\partial \mathcal{L}}{\partial v_k} = -{\alpha_k}^{+} + \beta - \lambda_k = 0, \qquad \forall k = 1, 2, \ldots n \tag{3.14e}$$

$$\frac{\partial \mathcal{L}}{\partial \overline{v}_k} = -{\alpha_k}^{-} + \beta - \overline{\lambda}_k = 0, \qquad \forall k = 1, 2, \ldots n \tag{3.14f}$$

$$\frac{\partial \mathcal{L}}{\partial s} = C - \sum_{i=1}^{n} \delta_i = 0 \tag{3.14g}$$

$$\frac{\partial \mathcal{L}}{\partial \overline{s}} = C - \sum_{i=1}^{n} \overline{\delta}_i = 0 \tag{3.14h}$$

By substituting the equivalent expressions for $\mathbf{w}$, b, ξ, $\overline{\xi}$, v, $\overline{v}$, s and $\overline{s}$ from equations (3.14a) - (3.14h) back in expression (3.13), the Wolfe dual of (3.12) is derived as,

$$\max_{\alpha^+,\alpha^-,\beta,\overline{\beta}} \; -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left({\alpha_i}^+ - {\alpha_i}^-\right)\left({\alpha_j}^+ - {\alpha_j}^-\right)K(\mathbf{x}_i,\mathbf{x}_j)$$

$$-\epsilon\sum_{i=1}^{n}\left({\alpha_i}^+ + {\alpha_i}^-\right) + \sum_{i=1}^{n} y_i\left({\alpha_i}^+ - {\alpha_i}^-\right) - \beta n\Upsilon \tag{3.15a}$$

$$\text{s.t.} \;\; \sum_{i=1}^{n}\left({\alpha_i}^+ - {\alpha_i}^-\right) = 0 \tag{3.15b}$$

$$0 \le {\alpha_i}^+ \le \beta, \qquad \forall i \in 1,\ldots,n \tag{3.15c}$$

$$0 \le {\alpha_i}^- \le \beta, \qquad \forall i \in 1,\ldots,n \tag{3.15d}$$

where α^+, α^- and β are the corresponding Lagrange multipliers.

Optimal Hyperplane Parameters

The solution to (3.15) is used to evaluate,

$$\mathbf{w}^* = \sum_{j=1}^{n}\left({\alpha_j}^+ - {\alpha_j}^-\right)\Phi(\mathbf{x}_j) \tag{3.16a}$$

Let $S^+ = \left\{{\alpha_i}^+ | \, (0 < {\alpha_i}^+ < C)\right\}$, $I^+ = \left\{i | \, {\alpha_i}^+ \in S^+\right\}$.
Let $S^- = \left\{{\alpha_i}^- | \, (0 < {\alpha_i}^- < C)\right\}$, $I^- = \left\{i | \, {\alpha_i}^- \in S^-\right\}$.

The bias can be computed as,

$$b^* = \frac{1}{|S^+|}\sum_{i\in I^+}\left(-\epsilon - \sum_{j=1}^{n}\left({\alpha_j}^+ - {\alpha_j}^-\right)K(\mathbf{x}_i,\mathbf{x}_j) + y_i\right) + \frac{1}{|S^-|}\sum_{i\in I^-}\left(\epsilon - \sum_{j=1}^{n}\left({\alpha_j}^+ - {\alpha_j}^-\right)K(\mathbf{x}_i,\mathbf{x}_j) + y_i\right) \tag{3.17a}$$

The response of a new datapoint $\boldsymbol{x}$ is determined from (4.2).

Illustrating Example

We consider an example here showing the effect of introducing a limited amount of penalty-free slack to SVR. The dataset used is Motorcycle (Eubank, 1999). Data is split 75/25, where 75% of the data is randomly selected for training and the remaining 25% is used for testing. Let $\rho \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$. Outliers $\mathbf{x}$ are generated such that $\mathbf{x} \in \big\{(x_1, x_2)|x_1 = (.8{+}.1{\times}\rho), x_2 = (-120{+}10{\times}\rho)\big\}$. In total 10 outliers are added to the training set. SVR and RSVR are equipped with a RBF kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2), \gamma \geq 0$ (Lin, Hsu, & Chang, 2003) and are trained on the data. The values of C, γ, ϵ and Υ are chosen, through 10-fold cross validation, to be 210, 128, 0.1 and 21.5 respectively. The root mean square error (RMSE) values as well as the adjusted coefficient of determination R^2_{adj} values, obtained from the testing set, are shown in Table 3.1. The fit of the two methods is shown in Figure 3.1. RMSE value for RSVR is significantly lower than for SVR, by approximately 11%. Adjusted coefficient of determination is approximately 2% higher in RSVR than in SVR. This example shows that introducing a limited amount of free slack reduces the influence of the outliers, effectively improving the performance of SVR.

Table 3.1: Root mean square error and adjusted coefficient of determination values for SVR and RSVR for Motorcycle dataset.

Method	**RMSE**	$\mathbf{R^2_{adj}}$
SVR	27.0233	0.7781
RSVR	24.0744	0.7913

Numerical Experiments

The performance of RSVR is evaluated on seven publicly available datasets that are summarized in Table 3.2.

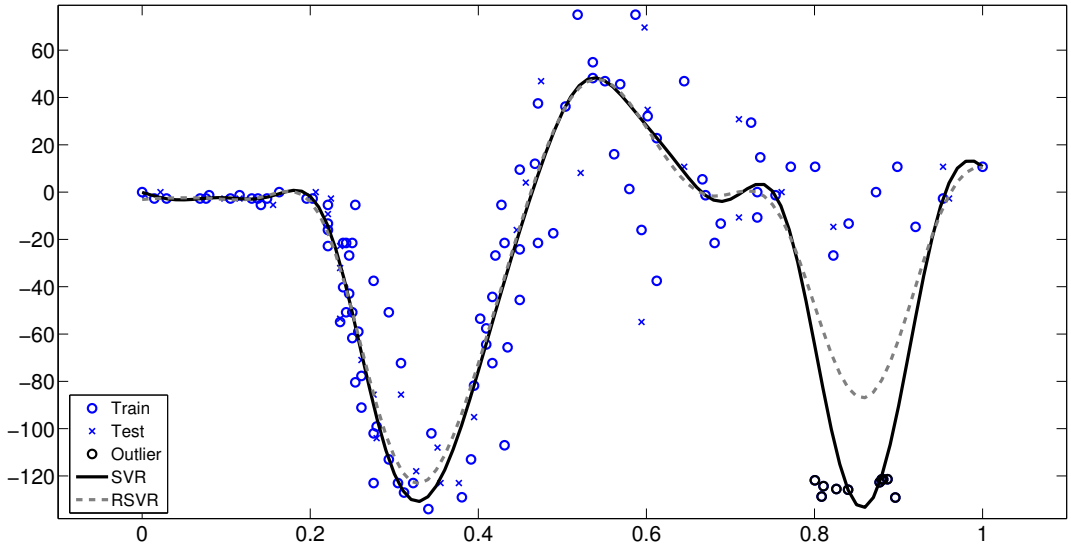


Figure 3.1: SVR and RSVR are fit to the training set contaminated with outliers. Testing data also appears on the plot.

Table 3.2: Summary table of datasets used for experiments.

Dataset	**Reference**	**#Samples**	**#Features**
Bodyfat	(Behnke & Wilmore, 1974)	252	14
Housing	(Harrison & Rubinfeld, 1978)	506	13
Weather Ankara	(Guvenir & Uysal, 2000)	321	9
Computer Hardware	(Kibler, Aha, & Albert, 1989)	209	9
Concrete Slump Test	(Yeh, 2007)	103	7
Triazines	(Hirst, King, & Sternberg, 1994)	186	6
Yacht Hydrodynamics	(Lichman, 2013)	308	6

Outliers are generated from a normal distribution such that the corresponding response vector consists of entries whose values differ from the training labels by a factor of 6 to 10 while the predictors are of the same magnitude. Outliers are induced in 5%, 10% and 20% outliers-to-data ratios. The performance of RSVR equipped with both linear and quadratic loss functions is compared to that of SVR. The two kernel functions used for the methods are RBF kernel and

linear kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. A uniform design (UD) approach (Huang, Lee, Lin, & Huang, 2007) is utilized for the model selection for each one of the methods that are evaluated and require parameter tuning. We adopt a 13- and 9-point run design for the first and second stages of the nested UD for SVR with linear kernel. We use 5 points for the third stage of the nested UD for SVR with RBF kernel and for RSVR with linear kernel. We adopt a 13-9-9-5 point run design for the four stages of the nested UD for RSVR with RBF kernel. The dual of SVR is solved using LIBSVM (Chang & Lin, 2011), while the dual of RSVR is solved using CPLEX (IBM, 2013) packages respectively. Experiments are performed with a 2.60GHz Intel Core i5 CPU running OS X with 8.0 GB of main memory. The classification performance is evaluated using hold out cross validation technique (hold out 10%, repeat 100 times)(Kohavi et al., 1995). The acquired root mean square errors and adjusted coefficient of determination values for the different outlier ratios are summarized in Tables 3.3-3.5.

For Housing, Bodyfat, Concrete Slump Test, Weather Ankara, Computer Hardware and Yacht Hydrodynamics datasets, RSVR significantly improves the prediction accuracy which drives the reported root mean squared errors lower in comparison to those of SVR. This is true for all different outliers-to-data ratios. Moreover, the explanatory power of RSVR is greater than that of SVR since it achieves significantly higher adjusted R-squared scores compared to the later. This result applies to all the different outliers-to-data ratios. In the case of Triazines dataset, RSVR is able to increase the prediction accuracy of SVR when the outliers introduced corresponded to 5% and 10% of the data. Additionaly, the square of the correlation between the response values and the predicted response values is higher when the dataset contains 10% and 20% induced outliers for RSVR. Overall, we observe that for all tested outliers-to-data ratios RSVR outperforms SVR in every aspect for most of the datasets achieving lower RMSE and higher R^2_{adj} scores respectively. These results suggest that RSVR can be very effective for data where different levels of outliers are present. We provide the comparative computational results for RSVR, SVR, OLS, RR, Lasso, Ridge and Elastic net. The acquired root mean square errors and adjusted coefficient of determination values for the different outlier ratios are summarized in Tables 3.6-3.8.

Table 3.3: Computational results of RSVR(Quadratic Loss/Linear Kernel), RSVR(Linear Loss/Linear Kernel), RSVR(Quadratic Loss/RBF Kernel), RSVR(Linear Loss/RBF Kernel), SVR(RBF Kernel), SVR(Linear Kernel) with corresponding RMSE and R^2_{adj} values for 5% outliers-to-data ratio.

Dataset	**Performance**	**RSVR(QL/L)**	**RSVR(LL/L)**	**RSVR(QL/RBF)**	**RSVR(LL/RBF)**	**SVR(RBF)**	**SVR(L)**
Bodyfat	RMSE	0.5174	0.5206	**0.1299**	0.2152	0.1951	0.4933
	R^2_{adj}	0.7399	0.7426	**0.9753**	0.9432	0.9508	0.7488
Housing	RMSE	0.7161	0.7213	**0.3396**	0.3987	0.3697	0.7194
	R^2_{adj}	0.4899	0.4681	**0.8731**	0.8542	0.8518	0.4641
Weather Ankara	RMSE	0.3039	0.3305	**0.1720**	0.1851	0.2903	0.3008
	R^2_{adj}	0.8991	0.8864	**0.9563**	0.9504	0.9078	0.9083
Computer Hardware	RMSE	1.7544	**0.5549**	2.505	2.5906	0.9852	1.6428
	R^2_{adj}	0.4001	**0.6637**	0.4269	0.4439	0.0272	0.4855
Concrete Slump Test	RMSE	1.0354	0.9828	1.0370	**0.8461**	0.9717	0.9595
	R^2_{adj}	0.1466	0.1292	0.0881	**0.4590**	0.0101	0.1528
Triazines	RMSE	1.0589	1.019	**0.9089**	1.1037	1.1080	1.1561
	R^2_{adj}	0.0969	0.0936	-0.2962	0.1409	**0.1420**	0.1201
Yacht Hydrodynamics	RMSE	1.8217	1.4296	0.4617	**0.4588**	0.6620	1.3416
	R^2_{adj}	0.3990	0.3169	**0.7363**	0.6772	0.3884	0.3021

We observe that RSVR is effective for both lower and higher outlier ratios. More specifically, for the case of 5% outliers-to-data ratio RSVR performs better than other regression methods in Housing, Concrete Slump Test, Weather Ankara and Yacht Hydrodynamics while RR, OLS and Elastic net achieve the best performance in the rest of the datasets. For 10% outliers-to-data ratio, RSVR is superior in most of the datasets, with RR regression following in terms of performance. When the number of outliers doubles, RSVR behaves better than other regression methods in Housing, Bodyfat and Yacht Hydrodynamics while RR, Ridge, Lasso and OLS achieve the best performance in the rest of the datasets.

Table 3.4: Computational results of RSVR(Quadratic Loss/Linear Kernel), RSVR(Linear Loss/Linear Kernel), RSVR(Quadratic Loss/RBF Kernel), RSVR(Linear Loss/RBF Kernel), SVR(RBF Kernel), SVR(Linear Kernel) with corresponding RMSE and R^2_{adj} values for 10% outliers-to-data ratio.

Dataset	**Performance**	**RSVR(QL/L)**	**RSVR(LL/L)**	**RSVR(QL/RBF)**	**RSVR(LL/RBF)**	**SVR(RBF)**	**SVR(L)**
Bodyfat	RMSE	0.8211	0.6824	**0.1472**	0.3194	0.4155	0.6347
	R^2_{adj}	0.6387	0.6847	**0.9635**	0.8870	0.8141	0.7009
Housing	RMSE	0.9278	0.7974	**0.3905**	0.5510	0.5057	0.8181
	R^2_{adj}	0.4012	0.4679	**0.8333**	0.6745	0.7264	0.4583
Weather Ankara	RMSE	1.2794	0.7129	0.2893	**0.0969**	0.5935	0.6183
	R^2_{adj}	0.4864	0.7104	0.9106	**0.9868**	0.8192	0.7031
Computer Hardware	RMSE	3.5039	**0.4953**	4.0143	4.7267	3.0868	7.5662
	R^2_{adj}	0.6155	**0.7410**	0.4705	0.47785	0.4640	0.5090
Concrete Slump Test	RMSE	1.0744	1.0468	0.9770	**0.8225**	0.9641	1.0296
	R^2_{adj}	0.1415	0.1270	0.0783	**0.3546**	0.1845	0.1318
Triazines	RMSE	**0.0475**	0.1179	0.1483	0.1517	0.1661	0.1148
	R^2_{adj}	1.3817	1.1184	1.2386	**1.3897**	1.0633	1.0657
Yacht Hydrodynamics	RMSE	1.7723	1.6437	**0.3280**	0.6074	0.6161	1.5319
	R^2_{adj}	0.3990	0.3169	**0.7363**	0.6772	0.3884	0.3021

Table 3.9 demonstrates the cumulative computational results in terms of the number of times that each one of the methods achieved the best overall performance: lowest RMSE and highest R^2_{adj} values respectively.

Overall, RSVR shows robustness with respect to the different levels of noise on the datasets. To that end, RSVR can be used for regression tasks in datasets where outliers are present in varying numbers.

Table 3.5: Computational results of RSVR(Quadratic Loss/Linear Kernel), RSVR(Linear Loss/Linear Kernel), RSVR(Quadratic Loss/RBF Kernel), RSVR(Linear Loss/RBF Kernel), SVR(RBF Kernel), SVR(Linear Kernel) with corresponding RMSE and R^2_{adj} values for 20% outliers-to-data ratio.

Dataset	**Performance**	**RSVR(QL/L)**	**RSVR(LL/L)**	**RSVR(QL/RBF)**	**RSVR(LL/RBF)**	**SVR(RBF)**	**SVR(L)**
Bodyfat	RMSE	1.0147	0.7369	**0.1521**	0.1700	0.6244	0.6319
	R^2_{adj}	0.5044	0.5573	**0.9626**	0.9551	0.7117	0.6125
Housing	RMSE	0.9442	0.8582	**0.3831**	0.5414	0.4587	0.7211
	R^2_{adj}	0.3173	0.3714	**0.8407**	0.6702	0.8057	0.4837
Weather Ankara	RMSE	1.4291	0.7872	**0.3758**	0.4071	0.4587	0.4502
	R^2_{adj}	0.4028	0.6214	**0.8670**	0.8401	0.8527	0.8074
Computer Hardware	RMSE	2.430	**0.5825**	5.9810	9.4901	4.002	9.9167
	R^2_{adj}	0.6010	**0.6968**	0.4808	0.4842	0.4747	0.4982
Concrete Slump Test	RMSE	1.0091	1.014	**0.9455**	1.0290	0.9850	1.0451
	R^2_{adj}	**0.1711**	0.1599	0.1426	0.1691	0.1668	0.1599
Triazines	RMSE	1.7459	1.6425	1.8832	1.7237	1.6361	**1.6246**
	R^2_{adj}	0.1736	0.1835	0.2394	**0.2671**	0.1721	0.1881
Yacht Hydrodynamics	RMSE	2.0906	1.9389	0.5735	**0.2700**	0.4689	1.6321
	R^2_{adj}	0.4067	0.3867	0.6558	**0.9013**	0.8268	0.3432

Remarks

A modified SVM method for regression, called relaxed support vector regression, was proposed and designed for datasets with noise/outliers. We have shown that the proposed algorithm performs competitively when compared to SVR. Relaxed support vector regression accounts for the presence of outliers by utilizing a silo of free slack. The improvement in regression performance shows the importance of relaxing the most influential support vectors to mitigate the influence of noise. Additionally, RSVR performs competitively when compared to other popular methods that are used for regression.

Table 3.6: Computational comparisons of RSVR, SVR, OLS, RR, Lasso, Ridge and Elastic net with corresponding RMSE and R^2_{adj} values for 5% outliers-to-data ratio.

Dataset	**Performance**	**RSVR**	**SVR**	**OLS**	**RR**	**Lasso**	**Ridge**	**Elastic net**
Bodyfat	RMSE	0.1299	0.1951	0.5134	**0.2025**	0.5888	0.4802	1.2511
	R^2_{adj}	**0.9753**	0.9508	0.7806	0.9417	0.6125	0.7875	0.5245
Housing	RMSE	**0.3396**	0.3697	0.7706	0.6010	0.7779	0.7723	0.7812
	R^2_{adj}	**0.8731**	0.8518	0.4144	0.6075	0.3761	0.4120	0.3803
Weather Ankara	RMSE	**0.1720**	0.2903	0.4056	0.3168	0.4251	0.4087	0.4417
	R^2_{adj}	**0.9563**	0.9083	0.8585	0.8956	0.8435	0.8560	0.8313
Computer Hardware	RMSE	0.5549	0.9852	0.9535	**0.5250**	0.8221	0.9283	0.7891
	R^2_{adj}	0.6637	0.4855	0.6363	0.6165	0.6654	0.6445	**0.6892**
Concrete Slump Test	RMSE	**0.8461**	0.9595	0.9041	0.8951	0.9637	0.9041	1.2170
	R^2_{adj}	**0.4590**	0.1292	0.1869	0.2494	0.1306	0.1869	0.3001
Triazines	RMSE	0.9089	1.1080	**0.0974**	0.2648	2.6429	0.7074	2.6531
	R^2_{adj}	0.1409	0.1420	0.4696	**0.7162**	0.1291	0.2407	0.1290
Yacht Hydrodynamics	RMSE	**0.4588**	0.6620	1.7685	0.7300	1.6771	1.7660	1.9611
	R^2_{adj}	**0.7363**	0.3884	0.3909	0.3398	0.3826	0.3906	0.40961

Overall, RSVR demonstrates robust behavior and can be used on datasets with varying levels of noise.

A potential future research direction is the development of a version of RSVR for data that belong in uncertainty sets. We also aim to apply the concept of constraint relaxation to the multi-class classification problem.

Table 3.7: Computational comparisons of RSVR, SVR, OLS, RR, Lasso, Ridge and Elastic net with corresponding RMSE and R^2_{adj} values for 10% outliers-to-data ratio.

Dataset	Performance	RSVR	SVR	OLS	RR	Lasso	Ridge	Elastic net
Bodyfat	RMSE	**0.1472**	0.4155	0.7009	0.2522	0.7824	0.7005	0.8286
	R^2_{adj}	**0.9635**	0.8141	0.9266	0.7033	0.6690	0.7221	0.6740
Housing	RMSE	**0.3905**	0.5057	0.8966	0.6237	0.9056	0.8963	0.8963
	R^2_{adj}	**0.8333**	0.7264	0.4094	0.5813	0.3357	0.4073	0.3538
Weather Ankara	RMSE	**0.0969**	0.5935	0.6513	0.3309	0.7692	0.6545	0.8283
	R^2_{adj}	**0.9868**	0.8192	0.6697	0.8875	0.5417	0.6673	0.5602
Computer Hardware	RMSE	0.4953	3.0868	0.7150	**0.5095**	0.8955	0.7150	1.1680
	R^2_{adj}	**0.7410**	0.5090	0.5151	0.6165	0.4099	0.5151	0.5351
Concrete Slump Test	RMSE	**0.8225**	0.9641	1.0389	0.8990	1.0318	1.0291	1.0318
	R^2_{adj}	**0.3546**	0.1845	0.2082	0.1226	0.2066	0.1973	0.2065
Triazines	RMSE	**0.0475**	0.1148	1.8556	1.9916	1.0652	0.0626	1.1682
	R^2_{adj}	**1.3897**	1.0657	0.1674	0.1562	-0.2472	0.1041	-0.1196
Yacht Hydrodynamics	RMSE	**0.3280**	0.6161	1.7634	0.6077	1.6870	1.7630	2.1028
	R^2_{adj}	**0.7363**	0.3884	0.3909	0.3398	0.3826	0.3906	0.4096

Table 3.8: Computational comparisons of RSVR, SVR, OLS, RR, Lasso, Ridge and Elastic net with corresponding RMSE and R^2_{adj} values for 20% outliers-to-data ratio.

Dataset	Performance	RSVR	SVR	OLS	RR	Lasso	Ridge	Elastic net
Bodyfat	RMSE	**0.1521**	0.6244	0.8295	0.5295	0.8301	0.8292	0.7877
	R^2_{adj}	**0.9626**	0.7117	0.7800	0.7030	0.6851	0.6814	0.6841
Housing	RMSE	**0.3831**	0.4587	0.9562	0.7211	0.9560	0.9576	0.9559
	R^2_{adj}	**0.8407**	0.8057	0.2883	0.5075	0.2873	0.2862	0.2874
Weather Ankara	RMSE	0.3758	0.4502	1.2737	**0.3001**	1.2718	1.2786	1.2722
	R^2_{adj}	0.6599	0.5563	0.3865	**0.9021**	0.3861	0.3856	0.3862
Computer Hardware	RMSE	0.5825	4.002	0.2326	0.2648	0.2597	**0.2325**	0.2750
	R^2_{adj}	0.7410	0.4982	0.9256	**0.9878**	0.8976	0.9253	0.8857
Concrete Slump Test	RMSE	**0.9455**	0.9850	1.1612	1.1093	1.1446	1.1536	1.1279
	R^2_{adj}	0.1711	0.1668	**0.1968**	0.1671	0.1761	0.1919	0.1716
Triazines	RMSE	1.6425	1.6246	1.5778	1.2835	**1.2770**	1.5220	1.4068
	R^2_{adj}	**0.2671**	0.1881	0.2090	0.1615	-0.0090	0.1959	0.0528
Yacht Hydrodynamics	RMSE	**0.2700**	0.4689	1.9727	1.6137	2.0045	1.9766	2.7772
	R^2_{adj}	**0.9013**	0.8268	0.3955	0.3227	0.4019	0.3960	0.4472

Table 3.9: Cumulative computational results of RSVR, SVR, OLS, RR, Lasso, Ridge and Elastic net in terms of the number of times they achieved the best overall performance: lowest RMSE and highest R^2_{adj} values respectively. Best possible score is 7 which is the total number of datasets.

	RMSE			$\mathbf{R^2_{adj}}$		
	5%	**10%**	**20%**	**5%**	**10%**	**20%**
RSVR	4	6	4	5	7	4
SVR	-	-	-	-	-	-
OLS	1	-	-	-	-	1
RR	2	1	1	1	-	2
Lasso	-	-	1	-	-	-
Ridge	-	-	1	-	-	-
Elastic net	-	-	-	1	-	-

CHAPTER 4: RELAXED ONE-CLASS SUPPORT VECTOR MACHINES FOR NOVELTY DETECTION

In this section, we propose a modified version of one-class support vector machines (OSVMs), called relaxed one-class support vector machines (ROSVMs) which aims to mitigate the negative influence that noise might have on classification performance.

Preliminaries

Let $S = \{(\mathbf{x}_i, y_i)\}$, $\mathbf{x}_i \in \mathbb{R}^d, y_i = +1 \quad \forall i \in I^+$ be a training set, where I^+ is the set of sample indices for the available positive class.

One-class Support Vector Machines (OSVMs) (Schölkopf et al., 1999) solve the following optimization problem,

$$\min_{\mathbf{w},\rho,\xi} \ \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i \in I^+} \xi_i - \rho \tag{4.1a}$$

$$\text{s.t.} \ \ \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle \geq \rho - \xi_i, \qquad \forall i \in I^+ \tag{4.1b}$$

$$\xi_i \geq 0, \qquad \forall i \in I^+ \tag{4.1c}$$

where ρ is the offset term.

OSVMs can handle non-linearly separable data through a transformation $\Phi : \mathbb{R}^d \to \mathcal{H}$, such that $\Phi(\mathbf{x}_i) \in \mathcal{H}$, where $\mathcal{H}$ is a reproducing kernel Hilbert space with $\dim(\mathcal{H}) > \dim(\mathbb{R}^d)$ and Φ is the kernel transformation.

The class of a new datapoint $\boldsymbol{x}$ is determined as,

$$\text{class}(\boldsymbol{x}) = sgn\left(\langle \mathbf{w}^*, \Phi(\boldsymbol{x})\rangle - \rho^*\right) \tag{4.2}$$

where $\mathbf{w}^*$ is the weight vector of the optimal hyperplane and ρ^* is the corresponding offset term.

Formulation

We use the concept of RSVM to relax one-class support vector machines. OSVMs are modified accordingly to produce our proposed model, relaxed one-class support vector machines (ROSVMs). ROSVMs are formulated with a quadratic loss function as follows,

$$\min_{\mathbf{w},\rho,\xi,v} \quad \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle + \frac{C}{2}\sum_{i\in I^+} {\xi_i}^2 - \rho \tag{4.3a}$$

$$\text{s.t.} \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle \geq \rho - \xi_i - v_i, \qquad \forall i \in I^+ \tag{4.3b}$$

$$\sum_{i\in I^+} v_i \leq n\ \Upsilon \tag{4.3c}$$

$$v_i \geq 0, \qquad \forall i \in I^+ \tag{4.3d}$$

where Υ is controlled by the user and is used to determine the average amount of free slack distributed to each sample.

The Lagrangian function for Formulation (4.3) can be written as,

$$\mathcal{L}(\mathbf{w}, \xi, \rho, \alpha, \beta, \lambda, v) =$$
$$\frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + \frac{C}{2} \sum_{i \in I^+} {\xi_i}^2 - \rho - \sum_{i \in I^+} \alpha_i \left(\xi_i - \rho + v_i + \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle \right)$$
$$- \beta \left(n\Upsilon - \sum_{i \in I^+} v_i \right) - \sum_{i \in I^+} \lambda_i v_i \tag{4.4a}$$

where α, β and λ are the Lagrangian multipliers. Since (4.3) is a convex problem, its Wolfe dual can be obtained from the following stationary first order conditions of the primal variables $\mathbf{w}$, ξ, v and ρ .

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i \in I^+} \alpha_i \Phi(\mathbf{x}_i) = 0 \tag{4.5a}$$
$$\frac{\partial \mathcal{L}}{\partial \xi_k} = C\xi_k - \alpha_k = 0, \qquad \forall k \in I^+ \tag{4.5b}$$
$$\frac{\partial \mathcal{L}}{\partial \rho} = -1 + \sum_{i \in I^+} \alpha_i = 0 \tag{4.5c}$$
$$\frac{\partial \mathcal{L}}{\partial v_k} = -\alpha_k + \beta - \lambda_k = 0, \qquad \forall k \in I^+ \tag{4.5d}$$

Substituting the equivalent expressions for $\mathbf{w}$, ξ, v and ρ from equations (4.5a) - (4.5d) back in expression (4.4), the Wolfe dual can be written as shown in (4.6).

The Wolfe dual of (4.3) is,

$$\max_{\alpha,\beta} \; -\frac{1}{2}\sum_{i\in I^+}\sum_{j\in I^+}\alpha_i\alpha_j\left(K(\mathbf{x}_i,\mathbf{x}_j)+\frac{\delta_{ij}}{C}\right)-\beta n\Upsilon \tag{4.6a}$$

$$\text{s.t.} \; \sum_{i\in I^+}\alpha_i = 1 \tag{4.6b}$$

$$0 \le \alpha_i \le \beta, \qquad \forall i \in I^+ \tag{4.6c}$$

where α and β are the corresponding Lagrange multipliers

The quadratic problem as defined in (4.6) can be solved with sequential minimal optimization (SMO) (Platt et al., 1998). The recommended kernel function to be used for training is the Radial Basis Function (RBF).

CHAPTER 5: CONCLUSION

Data analytics are allowing businesses and scholars to become efficient, more productive, and better in decision-making. During the analysis step we frequently encounter high dimensional datasets and data with outliers. High dimensional datasets arise in various areas, including business analytics and biomedical applications while outliers can be present due to noise or measurement errors. High dimensional data and the presence of outliers in data each render many existing data analytics techniques impractical. In this work we presented novel models that deal with issues that arise from the high dimensionality of data as well as the presence of outliers.

In chapter 2 we developed the novel binary classification method called constrained subspace classifier. CSC optimized local subspace classifier by accounting for the relative angle between the subspaces and utilizing the projection metric. An efficient alternate optimization technique was also proposed. Simulations demonstrated that the model improves the accuracy of LSC, showing the significance of considering the relative angle between subspaces while approximating the classes. Moreover, CSC appears to be a robust classifier, compared to traditional two step methods that perform feature selection and classification in two distinct steps.

In chapter 3 we proposed relaxed support vector regression and in chapter 4 one-class relaxed support vector machines. These methods constitute extensions of relaxed support vector machines. Relaxed support vector regression was formulated using both linear and quadratic loss functions. Numerical experiments on public datasets and computational comparisons with other popular regression methods depicted the improved behavior of our method. RSVR achieved better overall performance than support vector regression in several evaluation measures. Additionally, RSVR provided robustness for higher dimensional datasets and can be used on datasets that contain varying levels of outliers.

Our models demonstrate very competitive performance compared to other state-of-the-art methods

that are used for data analysis and can be a real asset to both academicians and industry-oriented professionals. This work has the potential to benefit both the academic community and the business industry and we are thrilled to share our research findings.

APPENDIX A: PRINCIPAL COMPONENT ANALYSIS

Let S be a data space of dimension equal to the number of features, d, of the selected dataset. We can always find an orthonormal basis for S (using the Gram-Schmidt process) given by

$$U_d = \{u_1, u_2, \ldots, u_d\} \text{ with } u_i \in \mathbb{R}^d \ \forall i = 1, 2, ..., d \tag{A.1}$$

i.e. $U_d \in \mathbb{R}^{d \times d}$. We seek to find a subspace of S of dimension $d_1 < d$. Since reducing the dimensionality brings the data points closer to each other, thus reducing the variance, we try to reduce the number of features from d to d_1 while trying to maintain the variance of the data distribution as high as possible.

To achieve the dimensionality reduction we seek to find a projection operator that projects the data points from $\mathbb{R}^d$ to a (dimensionally reduced) subspace $\mathbb{R}^{d_1}$ of orthonormal basis given by

$$U_{d_1} = \{u_1, u_2, \ldots, u_{d_1}\} \text{ with } u_i \in \mathbb{R}^d \ \forall i = 1, 2, ..., d_1 \tag{A.2}$$

i.e. $U_{d_1} \in \mathbb{R}^{d \times d_1}$. By definition the projection operator is given by

$$P = Q(Q^\intercal Q)^{-1} Q^\intercal \tag{A.3}$$

and projects a vector onto the space spanned by the columns of Q. Therefore, we may take the columns of Q to be the orthonormal vectors given in (A.2), that is $Q = U_{d_1}$. In that case, equation (A.3) becomes

$$P = U_{d_1}(U_{d_1}^\intercal U_{d_1})^{-1} U_{d_1}^\intercal \tag{A.4}$$

which is the projection operator onto the space spanned by the column vectors of U_{d_1}.

Since equation (A.1) is an orthonormal basis for $\mathbb{R}^d$ then $U_{d_1}^\intercal U_{d_1} = I_{d_1}$. Therefore, the expression of the projection operator that can project the (data) vectors in $\mathbb{R}^d$ onto its subspace $\mathbb{R}^{d_1}$ is given

by

$$P = U_{d_1} U_{d_1}^{\intercal}. \tag{A.5}$$

In case $d_1 = d$ then $P = U_d U_d^{\intercal}$. Since U_d is a square matrix whose columns are orthonormal, this implies that its rows are also orthonormal. Orthonormality of the columns of U_d implies $U_d^{\intercal} U_d = I_d$ (i.e. $U_d^{\intercal}$ is the left inverse of U_d) and orthonormality of the rows of U_d implies $U_d U_d^{\intercal} = I_d$ (i.e. $U_d^{\intercal}$ is the right inverse of U_d). Therefore, for the special case that $d_1 = d$ we have that $U_d^{\intercal}$ is the inverse of U_d or

$$U_d^{\intercal} = U_d^{-1}. \tag{A.6}$$

To introduce PCA we let data matrix X_1' be a $N \times d$ matrix. Using the projection operator as given by expression (A.5) we want to project the contents of X_1' in a subspace $\mathbb{R}^{d_1}$ of $\mathbb{R}^d$ ($d_1 < d$). Let x_i be the original $1 \times d$ row vector in $\mathbb{R}^d$. We project the column vector $x_i^{\intercal}$ onto $\mathbb{R}^{d_1}$ thus defining $\tilde{x_i}^{\intercal} = U_{d_1} U_{d_1}^{\intercal} x_i^{\intercal}$. Then the norm of the difference between the original and the projected (column) vectors can be expressed as

$$\|x_i^{\intercal} - \tilde{x_i}^{\intercal}\| = \|x_i^{\intercal} - U_{d_1} U_{d_1}^{\intercal} x_i^{\intercal}\| \tag{A.7}$$

where $U_{d_1} \in \mathbb{R}^{d \times d_1}$. In PCA we want to find the subspace $\mathbb{R}^{d_1}$ such that

$$\begin{aligned} &\sum_{i=1}^{n} \|x_i^{\intercal} - U_{d_1} U_{d_1}^{\intercal} x_i^{\intercal}\|^2 \quad \text{is minimized} \\ &\text{subject to} \quad U_{d_1}^{\intercal} U_{d_1} = \mathcal{I}_{d_1}. \end{aligned} \tag{A.8}$$

This subspace $\mathbb{R}^{d_1}$ is defined as the d_1-dimensional hypersurface that is spanned by the (reduced) orthonormal basis $\{u_1, u_2, u_3, \ldots, u_{d_1}\}$. i.e. finding such a basis is equivalent to defining the subspace $\mathbb{R}^{d_1}$.

Using the definition of the Frobenius norm for a $m \times n$ matrix A,

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\text{trace}(A^*A)} \tag{A.9}$$

where A^* is the conjugate transpose of A, we get

$$\sum_{i=1}^{n}\|x_i^\intercal - U_{d_1}U_{d_1}^\intercal x_i^\intercal\|_F^2 = \text{tr}\left\{X_1'^\intercal X_1'(\mathcal{I} - U_{d_1}U_{d_1}^\intercal)\right\} \tag{A.10}$$

where $X_1' \in \mathbb{R}^{n\times d}$ (where $n = 2N$). Thus the optimization problem in equation (A.8) reduces to

$$\begin{aligned} &\min_{U_{d_1}} \text{tr}\left\{X_1'^\intercal X_1'(\mathcal{I} - U_{d_1}U_{d_1}^\intercal)\right\} \\ &\text{subject to } \; U_{d_1}^\intercal U_{d_1} = \mathcal{I}_{d_1}. \end{aligned} \tag{A.11}$$

Since $\text{tr}\left\{X_1'^\intercal X_1'\right\}$ is a constant, the optimization problem can be re-written as

$$\begin{aligned} &\max_{U_{d_1}} \text{tr}\{U_{d_1}^\intercal X_1'^\intercal X_1' U_{d_1}\} \\ &\text{subject to } \; U_{d_1}^\intercal U_{d_1} = \mathcal{I}_{d_1}. \end{aligned} \tag{A.12}$$

To solve equation (A.12) we define the Lagrangian dual problem by

$$\begin{aligned} \mathcal{L}(U_{d_1}, \lambda_{ij}) = &\text{tr}(U_{d_1}^\intercal X_1'^\intercal X_1' U_{d_1}) - \\ &- \sum_{i=1}^{d_1}\sum_{j=1}^{d_1}\lambda_{ij}(\sum_{k=1}^{d}U_{jk}^\intercal U_{ki} - \delta_{ji}) \\ &\text{where } \; \delta_{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j. \end{cases} \end{aligned} \tag{A.13}$$

Since $U_{d_1}^{\intercal} U_{d_1}$ is a symmetric $d_1 \times d_1$ matrix then the orthonormality condition in equation (A.12) represents a total of $d_1 \times (d_1 + 1)/2$ conditions. Therefore, for the Lagrangian dual problem (as shown in equation (A.13)) we need to introduce $d_1 \times (d_1 + 1)/2$ Lagrange multipliers λ_{ij}. Hence we require that λ_{ij} is a symmetric matrix. Also since each term in (A.13) involves symmetric matrices then the following first order optimality conditions

$$\frac{\partial \mathcal{L}}{\partial \lambda_{pq}} = 0 \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial U_{lm}} = 0. \tag{A.14}$$

can be solved for λ_{ij} only if the latter is symmetric. Using equations (A.13) and (A.14) we get

$$\begin{aligned} \frac{\partial}{\partial \lambda_{pq}} \Bigg[& \sum_{i=1}^{d_1} \sum_{j=1}^{d} \sum_{k=1}^{d} U_{ij}^{\intercal} (X_1'^{\intercal} X_1')_{jk} U_{ki} - \\ & - \sum_{i=1}^{d_1} \sum_{j=1}^{d_1} \lambda_{ij} \Big(\sum_{k=1}^{d} U_{jk}^{\intercal} U_{ki} - \delta_{ji} \Big) \Bigg] = 0 \end{aligned} \tag{A.15}$$

and

$$\begin{aligned} \frac{\partial}{\partial U_{lm}} \Bigg[& \sum_{i=1}^{d_1} \sum_{j=1}^{d} \sum_{k=1}^{d} U_{ij}^{\intercal} (X_1'^{\intercal} X_1')_{jk} U_{ki} - \\ & - \sum_{i=1}^{d_1} \sum_{j=1}^{d_1} \lambda_{ij} \Big(\sum_{k=1}^{d} U_{jk}^{\intercal} U_{ki} - \delta_{ji} \Big) \Bigg] = 0. \end{aligned} \tag{A.16}$$

Equation (A.15) implies the $d_1 \times (d_1 + 1)/2$ equations

$$\sum_{k=1}^{d} U_{qk}^{\intercal} U_{kp} = \delta_{qp} \tag{A.17}$$

while equation (A.16) implies the $d \times d_1$ equations

$$\sum_{j=1}^{d} U_{mj}^{\intercal}(X_1'^{\intercal}X_1')_{jl} + \sum_{k=1}^{d}(X_1'^{\intercal}X_1')_{lk}U_{km} - \\ - \sum_{j=1}^{d_1}\lambda_{mj}U_{jl}^{\intercal} - \sum_{i=1}^{d_1}\lambda_{im}U_{li} = 0. \tag{A.18}$$

Using the fact that $X_1'^{\intercal}X_1'$ is symmetric, the first two terms of equation (A.18) can be combined to a single term and similarly (using the symmetry of λ_{ij}) the last two terms of equation (A.18) can be combined to a single term to get

$$\sum_{j=1}^{d} U_{mj}^{\intercal}(X_1'^{\intercal}X_1')_{jl} - \sum_{i=1}^{d_1}\lambda_{mi}U_{il}^{\intercal} = 0. \tag{A.19}$$

Equations (A.19) and (A.17) are sufficient to solve for λ_{ij} and U_{kl}. Right-multiplying equation (A.19) by U_{ln} and summing over $1 \leq l \leq d$ we get

$$\sum_{l=1}^{d}\sum_{j=1}^{d} U_{mj}{}^{\intercal}(X_1'^{\intercal}X_1')_{jl}U_{ln} - \sum_{i=1}^{d_1}\lambda_{mi}\sum_{l=1}^{d} U_{il}^{\intercal}U_{ln} = 0. \tag{A.20}$$

Using equation (A.17) then equation (A.20) becomes

$$\sum_{l=1}^{d}\sum_{j=1}^{d} U_{mj}{}^{\intercal}(X_1'^{\intercal}X_1')_{jl}U_{ln} = \lambda_{mn}. \tag{A.21}$$

Equations (A.21) and (A.19) represent a set of $d_1 \times (d_1+1)/2$ and $d_1 \times d$ equations respectively. These can be solved to obtain the $d_1 \times (d_1+1)/2$ degrees of freedom of λ_{ij} and the $d_1 \times d$ degrees of freedom of U_{d_1}.

The left hand side (LHS) of equation (A.21) represents the a_{mn} elements of a $d_1 \times d_1$ matrix and similarly the right hand side (RHS) of (A.21) represents the λ_{mn} elements of another $d_1 \times$

d_1 matrix. Equation (A.21) implies an entry-by-entry equation ($a_{mn} = \lambda_{mn}$) between the two matrices. Choosing $m = n$ and summing equation (A.21) over $1 \leq m \leq d_1$ implies that the sum along the diagonal of the matrix on the LHS is equal to the sum along the diagonal of the matrix on the RHS or equivalently

$$\sum_{m=1}^{d_1} \sum_{l=1}^{d} \sum_{j=1}^{d} U_{mj}{}^{\intercal} (X_1'^{\intercal} X_1')_{jl} U_{lm} = \sum_{m=1}^{d_1} \lambda_{mm}. \tag{A.22}$$

Noting that the LHS of (A.22) is the trace of the LHS of (A.21) we can re-write (A.22) as

$$\mathrm{tr}(U_{d_1}^{\intercal} X_1'^{\intercal} X_1' U_{d_1}) = \sum_{m=1}^{d_1} \lambda_{mm}. \tag{A.23}$$

We can identify λ_{mm} for $1 \leq m \leq d_1$ as the eigenvalues of the symmetric matrix $(X_1' U_{d_1})^{\intercal} (X_1' U_{d_1})$. However, these d_1 eigenvalues are d_1 out of the total d eigenvalues of $X_1'^{\intercal} X_1'$. This can be shown by using the invariance of trace under similarity transformations (in this case under conjugacy). Using equation (A.6) we can re-write equation (A.23) for $d_1 = d$ as

$$\mathrm{tr}(U_d^{-1} X_1'^{\intercal} X_1' U_d) = \mathrm{tr}(X_1'^{\intercal} X_1') = \sum_{m=1}^{d} \lambda_{mm}. \tag{A.24}$$

Therefore, the maximum of the objective function $F = \mathrm{tr}(U_{d_1}^{\intercal} X_1'^{\intercal} X_1' U_{d_1})$ in expression (A.12) is equal to the summation of the d_1 largest eigenvalues of $X_1'^{\intercal} X_1'$. Therefore the orthonormal basis for the lower dimensional subspace is given by the set of the eigenvectors corresponding to the d_1 largest eigenvalues of the symmetric matrix $X_1'^{\intercal} X_1'$.

APPENDIX B: SUPPORT VECTOR MACHINES

Let $S = \{(\mathbf{x}_i, y_i)\},\ \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, +1\} \quad \forall i = 1, \ldots, n$ be a training set.

Define the hinge loss function as

$$l(y_i, f(\mathbf{x}_i)) = |1 - y_i f(\mathbf{x}_i)|_+ \tag{B.1}$$

Support vector machines minimize the hinge loss function regularized by the L_2-norm (V. Vapnik, 2000). In particular they solve the following optimization problem,

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} l(y_i, f(\mathbf{x}_i)) \tag{B.2}$$

were C is a positive regularization parameter and $f(\mathbf{x}_i) = \langle \mathbf{w}, \mathbf{x}_i \rangle + b$ is the desired linear classifier.

The optimization problem in (B.2) can be rewritten as,

$$\min_{\mathbf{w},b,\xi} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \tag{B.3a}$$

$$\text{s.t. } y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \qquad \forall i \in 1, \ldots, n \tag{B.3b}$$

$$\xi_i \geq 0, \qquad \forall i \in 1, \ldots, n \tag{B.3c}$$

SVMs can become non-linear through a transformation $\Phi : \mathbb{R}^d \to \mathcal{H}$, such that $\Phi(\mathbf{x}_i) \in \mathcal{H}$, where $\mathcal{H}$ is a reproducing kernel Hilbert space with $\dim(\mathcal{H}) > \dim(\mathbb{R}^d)$ and Φ is the kernel transformation.

The Lagrangian function for Formulation (B.3) can be written as,

$$\mathcal{L}(\mathbf{w}, \xi, b, \alpha, \beta) =$$
$$\frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left(y_i (\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) - 1 + \xi_i \right) - \sum_{i=1}^{n} \beta_i \xi_i \tag{B.4a}$$

where α and β are the Lagrangian multipliers. The Wolfe dual of (B.3) can be obtained from the following stationary first order conditions of the primal variables $\mathbf{w}$, ξ and b .

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \Phi(\mathbf{x}_i) = 0 \tag{B.5a}$$
$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{B.5b}$$
$$\frac{\partial \mathcal{L}}{\partial \xi_k} = C - \alpha_k - \beta_k = 0, \qquad \forall k \in 1, \ldots, n \tag{B.5c}$$

Substituting the equivalent expressions for $\mathbf{w}$, ξ, and b from equations (B.5a) - (B.5c) back in expression (B.4), the Wolfe dual can be written as,

$$\max_{\alpha} \; -\frac{1}{2} \sum_{i=1}^{n} \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{n} \alpha_i \tag{B.6a}$$
$$\text{s.t.} \; \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{B.6b}$$
$$0 \leq \alpha_i \leq C \qquad \forall k \in 1, \ldots, n \tag{B.6c}$$

where α and β are the corresponding Lagrange multipliers

The solution to (B.6) is used to evaluate,

$$\mathbf{w}^* = \sum_{j=1}^{n} \alpha_j y_j \Phi(\mathbf{x}_j) \tag{B.7a}$$

Let $S = \left\{\alpha_i | (0 < \alpha_i < C)\right\}$, $I = \left\{i | \alpha_i \in S\right\}$.

The bias term can be computed as,

$$b^* = \frac{1}{|S|} \sum_{i \in I} \left(y_i - \sum_{j \in I} \alpha_j y_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) \right) \tag{B.8a}$$

The class of a new datapoint $\boldsymbol{x}$ is determined as,

$$\text{class}(\boldsymbol{x}) = \text{sign}\left(\langle \mathbf{w}^*, \Phi(\boldsymbol{x}) \rangle + b^*\right) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K\left(\mathbf{x}_i, \mathbf{x}\right) + b^* \right) \tag{B.9}$$

where $\mathbf{w}^*$ is the weight vector of the optimal hyperplane and b^* is the corresponding bias term.

Two popular kernel functions used to train SVMs are the linear kernel,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \tag{B.10}$$

and the radial basis function (RBF) kernel,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2 \right) \tag{B.11}$$

where $\gamma \geq 0$. The use of RBF kernel implies that neither the feature transformation Φ nor the dimensionality of $\mathcal{H}$ is required to be explicitly known.

Parameters C and γ are usually obtained through K-fold cross-validation. Several possible values of each of the parameters are tested and the parameter value that gives the lowest cross-validation

average error $E(z)$ is chosen; where

$$E(z) = \frac{1}{K} \sum_{k=1}^{K} \sum_{i \in k} \left(class(\mathbf{x}_i) - f\left(\mathbf{x}_i, z\right) \right)^2 \tag{B.12}$$

and z is the parameter tuned and k represents each fold. The lowest error corresponds to highest accuracy.

REFERENCES

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, *96*(12), 6745–6750.

Behnke, A. R., & Wilmore, J. H. (1974). *Evaluation and regulation of body build and composition*. Prentice Hall.

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is nearest neighbor meaningful? *Database Theory ICDT 99*, 217–235.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bjorck, Å., & Golub, G. H. (1973). Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, *27*(123), 579–594.

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence*, *97*(1), 245–271.

Cao, F., Ye, H., & Wang, D. (2015). A probabilistic learning algorithm for robust modeling using neural networks with random weights. *Information Sciences*, *313*, 62–78.

Cao, G., Guo, Y., Bouman, C., et al. (2010). High dimensional regression using the sparse matrix transform (smt). In *Acoustics speech and signal processing (icassp), 2010 ieee international conference on* (pp. 1870–1873).

Carrizosa, E., & Morales, D. R. (2013). Supervised classification and mathematical optimization. *Computers & Operations Research*, *40*(1), 150–165.

Cauwenberghs, G., & Poggio, T. (2001). Incremental and decremental support vector machine learning. *Advances in Neural Information Processing Systems*, 409–415.

Chang, C.-C., & Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 27.

Cifarelli, C., Guarracino, M. R., Seref, O., Cuciniello, S., & Pardalos, P. M. (2007). Incremental

classification with generalized eigenvalues. *Journal of classification*, *24*(2), 205–219.

Diehl, C. P., & Cauwenberghs, G. (2003). Svm incremental learning, adaptation and optimization. In *Neural networks, 2003. proceedings of the international joint conference on* (Vol. 4, pp. 2685–2690).

Dulá, J., & López, F. (2013). Dea with streaming data. *Omega*, *41*(1), 41–47.

D'Urso, P., Massari, R., & Santoro, A. (2011). Robust fuzzy regression analysis. *Information Sciences*, *181*(19), 4154–4174.

Edelman, T. A. A., Alan, & Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, *20*(2), 303–353.

Eubank, R. L. (1999). *Nonparametric regression and spline smoothing*. CRC press.

Fenn, M. B., & Pappu, V. (2012). Data mining for cancer biomarkers with raman spectroscopy. *Data Mining for Biomarker Discovery*, 143–168.

Freund, R. J., Littell, R. C., & Creighton, L. (2003). *Regression using jmp*. J. Wiley.

Golub, G. H., & Van Loan, C. F. (2012). *Matrix computations* (Vol. 3). JHU Press.

Guarracino, M. R., Cuciniello, S., & Feminiano, D. (2009). Incremental generalized eigenvalue classification on data streams. In *International workshop on data stream management and mining* (pp. 1–12).

Guvenir, H. A., & Uysal, I. (2000). *Bilkent university function approximation repository.* (Accessed: 2015-08-10)

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, *3*, 1157–1182.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, *46*(1), 389–422.

Hamm, J., & Lee, D. D. (2008). Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th international conference on machine learning* (pp. 376–383).

Harrison, D., & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air.

Journal of Environmental Economics and Management, *5*(1), 81–102.

Hassell, B. A.-M., Joseph, & Arpinar, I. B. (2006). *Ontology-driven automatic entity disambiguation in unstructured text* (Vol. 4273). Springer.

Hawkins, D. M. (1980). *Identification of outliers* (Vol. 11). Springer.

Hirst, J. D., King, R. D., & Sternberg, M. J. (1994). Quantitative structure-activity relationships by neural networks and inductive logic programming. ii. the inhibition of dihydrofolate reductase by triazines. *Journal of Computer-Aided Molecular Design*, *8*(4), 421–432.

Huang, C.-M., Lee, Y.-J., Lin, D. K., & Huang, S.-Y. (2007). Model selection for support vector machines via uniform design. *Computational Statistics & Data Analysis*, *52*(1), 335–346.

IBM. (2013). *IBM ILOG CPLEX: High-performance mathematical programming engine.*

Jiang, H., Deng, Y., Chen, H., Tao, L., Sha, Q., Chen, J., ... Zhang, S. (2004). Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC bioinformatics*, *5*(1), 81.

Johansen, S. (1988). Statistical analysis of cointegration vectors. *Journal of Economic Dynamics and Control*, *12*(2), 231–254.

Johnstone, I. M., & Titterington, D. M. (2009). Statistical challenges of high-dimensional data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *367*(1906), 4237–4253.

Jolliffe, I. (2005). *Principal component analysis*. Wiley Online Library.

Kibler, D., Aha, D. W., & Albert, M. K. (1989). Instance-based prediction of real-valued attributes. *Computational Intelligence*, *5*(2), 51–57.

Kohavi, R., et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI* (Vol. 14, pp. 1137–1145).

Köppen, M. (2000). The curse of dimensionality. In *5th online world conference on soft computing in industrial applications (wsc5)* (pp. 4–8).

Laaksonen, J. (1997). Local subspace classifier. In *Artificial neural networksicann'97* (pp. 637–642). Springer.

Lee, S. M., et al. (2010). Spam detection using feature selection and parameters optimization. *Intelligent and Software Intensive Systems (CISIS)*, 883–888.

Levinson, N. (1947). The wiener rms (root mean square) error criterion in filter design and prediction. *Institute of Electrical and Electronics Engineers*, *1*(3), 129–148.

Lichman, M. (2013). *UCI machine learning repository.* Retrieved from `http://archive.ics.uci.edu/ml`

Lin, C. J., Hsu, C.-W., & Chang, C.-C. (2003). A practical guide to support vector classification. *National Taiwan U., www.csie.ntu.edu.tw/cjlin/papers/guide/guide.pdf*.

Liu, H., & Motoda, H. (1998). *Feature extraction, construction and selection: A data mining perspective*. Springer.

López, F. G., Torres, M. G., Batista, B. M., Pérez, J. A. M., & Moreno-Vega, J. M. (2006). Solving feature subset selection problem by a parallel scatter search. *European Journal of Operational Research*, *169*(2), 477–489.

Panagopoulos, O. P., Pappu, V., Xanthopoulos, P., & Pardalos, P. M. (2015). Constrained subspace classifier for high dimensional datasets. *Omega*(10.1016/j.omega.2015.05.009).

Pang, S., Ozawa, S., & Kasabov, N. (2005). Incremental linear discriminant analysis for classification of data streams. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *35*(5), 905–914.

Pappu, V., Panagopoulos, O. P., Xanthopoulos, P., & Pardalos, P. M. (2015). Sparse proximal support vector machines for feature selection in high dimensional datasets. *Expert Systems With Applications*(10.1016/j.eswa.2015.08.022).

Peters, G., & Lacic, Z. (2012). Tackling outliers in granular box regression. *Information Sciences*, *212*, 44–56.

Platt, J., et al. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines.

Razzaghi, T., Otero, A., & Xanthopoulos, P. (2014). Encyclopedia of business analytics and optimization. In J. Wang (Ed.), (p. 1145-1154). IGI Global.

Rousseeuw, P. J., & Leroy, A. M. (2005). *Robust regression and outlier detection* (Vol. 589). John Wiley & Sons.

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, *23*(19), 2507–2517.

Satsangi, A., & Zaiane, O. R. (2007). Contrasting the contrast sets: An alternative approach. *11th International Database Engineering and Applications Symposium*, 114 – 119.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., & Platt, J. C. (1999). Support vector method for novelty detection. In *Nips* (Vol. 12, pp. 582–588).

Şeref, O., Chaovalitwongse, W. A., & Brooks, J. P. (2014). Relaxing support vectors for classification. *Annals of Operations Research*, *216*(1), 229–255.

Shanmugam, R., & Johnson, C. (2007). At a crossroad of data envelopment and principal component analyses. *Omega*, *35*(4), 351–364.

Shipp, M. A., Ross, K. N., Tamayo, P., Weng, A. P., Kutok, J. L., Aguiar, R. C., ... others (2002). Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature medicine*, *8*(1), 68–74.

Smith, M. R., & Martinez, T. (2011). Improving classification accuracy by identifying and removing instances that should be misclassified. In *The 2011 international joint conference on neural networks (ijcnn)* (pp. 2690–2697).

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222.

Stevens, J. P. (2012). *Applied multivariate statistics for the social sciences*. Routledge.

Street, J. O., Carroll, R. J., & Ruppert, D. (1988). A note on computing robust regression estimates via iteratively reweighed least squares. *The American Statistician*, *42*(2), 152–154.

Theil, H. (1959). Economic forecasts and policy. *The American Economic Review*, *49*(4), 711–716.

Unler, A., & Murat, A. (2010). A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research*, *206*(3), 528–

539.

Van't Veer, L. J., Dai, H., Van De Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., ... others (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, *415*(6871), 530–536.

Vapnik, V. (2000). *The nature of statistical learning theory*. Springer.

Vapnik, V. N., & Vapnik, V. (1998). *Statistical learning theory* (Vol. 1). Wiley New York.

Verardi, V., & Croux, C. (2008). Robust regression in stata. *Available at SSRN 1369144*.

Vidal, R., et al. (2006). *Generalized principal component analysis* (Vol. 1). Electronics Research Laboratory, College of Engineering, University of California.

Wolters, R., & Kateman, G. (1989). The performance of least squares and robust regression in the calibration of analytical methods under non-normal noise distributions. *Journal of Chemometrics*, *3*(2), 329–342.

Xanthopoulos, P., Guarracino, M. R., & Pardalos, P. M. (2014). Robust generalized eigenvalue classifier with ellipsoidal uncertainty. *Annals of Operations Research*, *216*(1), 327–342.

Xanthopoulos, P., Pardalos, P., & Trafalis, T. B. (2012). *Robust data mining*. Springer.

Xanthopoulos, P., & Razzaghi, T. (2014). A weighted support vector machine method for control chart pattern recognition. *Computers & Industrial Engineering*, *70*, 134–149.

Yang, E., Lozano, A., & Ravikumar, P. (2014). Elementary estimators for high-dimensional linear regression. In *Proceedings of the 31st international conference on machine learning (icml-14)* (pp. 388–396).

Yang, J., & Olafsson, S. (2006). Optimization-based feature selection with adaptive instance sampling. *Computers & Operations Research*, *33*(11), 3088–3106.

Yeh, I.-C. (2007). Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cement and Concrete Composites*, *29*(6), 474–480.