ROBUST DATA MINING TECHNIQUES WITH APPLICATION IN BIOMEDICINE AND ENGINEERING

By

PETROS XANTHOPOULOS

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2011

© 2011 Petros Xanthopoulos

To my grandmother

ACKNOWLEDGMENTS

During my doctoral studies in Florida, I had the luck to meet and work with many interesting people from many different countries and technical backgrounds. Apart from the technical skills that I received in the PhD program of Industrial and Systems Department I mostly learned how to be a part of the team and how to communicate my ideas and thoughts more efficiently. So first of all, I would like to acknowledge my academic advisor Dr. Panos M. Pardalos who guided me through all the stages of doctoral studies. His mentoring helped my development in multiple levels.

Also I would like to acknowledge my first mentor and undergraduate thesis advisor, professor Michalis Zervakis from the Electronics and Computer Engineering department at the Technical University of Crete, the person who showed me how to conduct research and helped me significantly during the first steps of my career. I would like to thank all my colleagues and friends at the Center for Applied Optimization, University of Florida. I had the chance to develop fruitful collaborations and friendships all this time I want to thank my parents Lazaros and Kaiti for their support of my personal decision and for their freedom they gave me. I would also like to thank Sibel for her continuous help and support. Last but not least I would like to thanks Dr. Joseph P Geunes, Dr. Guanhui (George) Lan and Dr. My T. Thai for serving in my PhD committee and dedicating part of their valuable time to this task.

TABLE OF CONTENTS

page

ACKNOWLEDGMENTS 4

LIST OF TABLES 7

LIST OF FIGURES 8

ABSTRACT 9

CHAPTER

1 INTRODUCTION 10

1.1 A Brief Overview 10
1.2 A Methodological Perspective 14
1.3 A Brief History of Robustness 18
1.3.1 Robust optimization vs Stochastic Programming 20
1.3.2 Thesis Goal and Structure 21

2 DATA MINING METHODS 23

2.1 Linear Least Squares 23
2.1.1 Weighted Linear Least Squares 26
2.1.2 Computational Aspects of Linear Least Squares 27
2.1.3 Least Absolute Shrinkage and Selection Operator 28
2.2 Principal Component Analysis 29
2.2.1 Maximum Variance Approach 29
2.2.2 Minimum Error Approach 31
2.3 Linear Discriminant Analysis 32
2.3.1 Generalized Discriminant Analysis 35
2.4 Support Vector Machines 36
2.4.1 Alternative Objective Function 41
2.5 Regularized Generalized Eigenvalue Classification 42
2.6 Case Study - Cell death discrimination 45
2.6.1 Materials 47
2.6.1.1 Dataset 47
2.6.1.2 Raman spectroscope 48
2.6.1.3 Data preprocessing 49
2.6.2 Software 49
2.6.3 Nearest Neighbor Classification 49
2.6.4 Improved Iterative Scaling 50
2.6.5 Results and Discussion 51
2.6.5.1 Classification and model selection 51
2.6.5.2 Impact of training set dimension 52

3 ROBUST LEAST SQUARES 54
3.1 The Original Problem 54
3.2 Variations of the Original Problem 58
4 ROBUST PRINCIPAL COMPONENT ANALYSIS 60
5 ROBUST LINEAR DISCRIMINANT ANALYSIS 63
6 ROBUST SUPPORT VECTOR MACHINES 67
7 ROBUST GENERALIZED EIGENVALUE CLASSIFICATION 72
7.1 Under Ellipsoidal Uncertainty 72
7.1.1 Robust Counterpart Under Ellipsoidal Uncertainty Set 75
7.1.2 Balancing Between Robustness and Optimality 78
7.2 Computational Results 78
7.2.1 A Case Study 78
7.2.2 Experiments 80
7.3 Extentions 84
7.3.1 Matrix Norm Constrained Uncertainty 87
7.3.2 Non linear Robust Generalized Classification 91
8 CONCLUSIONS 94
APPENDIX
A OPTIMALITY CONDITIONS 95
B DUAL NORMS 99
REFERENCES 100
BIOGRAPHICAL SKETCH 106

LIST OF TABLES

Table page

2-1 Average classification accuracy for hold out cross validation (100 repetitions) . 51

7-1 Dataset description. 80

LIST OF FIGURES

Figure page

1-1 Moore's "law" . . . 12

1-2 Kryder's "law" . . . 12

1-3 The big picture . . . 14

2-1 The signle input single outcome case . . . 24

2-2 LLS and regularization . . . 26

2-3 Two paths for PCA . . . 30

2-4 Directed acyclic graph approach for a four class example . . . 41

2-5 Explanation of Raman . . . 46

2-6 The mean Raman spectra for each class . . . 48

2-7 Classification accuracy versus size of the training set (binary) . . . 52

2-8 Classification accuracy versus size of the training set (multiclass) . . . 53

7-1 Objective function value dependence on the ellipse parameters. . . . 79

7-2 Analysis for Pima indian dataset . . . 81

7-3 Analysis for iris dataset . . . 81

7-4 Analysis for wine dataset . . . 82

7-5 Analysis for dataset NDC . . . 83

7-6 Comparison graph for (a) Pima indian (b)iris datasets . . . 85

7-7 Comparison graph for (c) wine and (d) NDC datasets . . . 86

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

ROBUST DATA MINING TECHNIQUES WITH APPLICATION IN BIOMEDICINE AND ENGINEERING

By

Petros Xanthopoulos

August 2011

Chair: Panos M. Pardalos
Major: Industrial and Systems Engineering

Analysis and interpretation of large datasets is a very significant problem that arises in many areas of science. The task of data analysis becomes even harder when data are uncertain or imprecise. Such uncertainties introduce bias and make massive data analysis an even more challenging task. Over the years there have been developed many mathematical methodologies for data analysis based on mathematical programming. In this field, the deterministic approach for handling uncertainty and immunizing algorithms against undesired scenarios is robust optimization. In this work we examine the application of robust optimization is some well known data mining algorithms. We explore the optimization structure of such algorithms and then we state their robust counterpart formulation.

CHAPTER 1
INTRODUCTION

Data Mining (DM), conceptually, is very general term that encapsulates a large number of methods, algorithms and technologies. The common denominator among all these is their ability to extract useful patterns and associations from data usually stored in large databases. Thus DM techniques aim to extract knowledge and information out of data. This task is crucial, especially today, mainly because of the emerging needs and capabilities that technological progress creates. In the present work we investigate some of the most well known data mining algorithms from an optimization perspective and we study the application of Robust Optimization (RO) in them. This combination is essential in order to address the unavoidable problem of data uncertainty that arises in almost all realistic problems that involve data analysis.

1.1 A Brief Overview

Before we state the mathematical problems of this thesis, we provide, for the shake of completion, a historical and methodological overview of DM. Historically DM was evolved, in its current form, during last few decades from the interplay of classical statistics and Artificial Intelligence (AI). It is worth mentioning that through this evolution process DM developed strong bonds with computer science and optimization theory. In order to study modern concepts and trends of DM we first need to understand its foundations and its interconnections with the four aforementioned disciplines.

Artificial Intelligence: The perpetual need/ desire of human to create artificial machines/ algorithms able to learn, decide and act as humans gave birth to Artificial Intelligence. Officially AI was born in 1956 in a conference held at Dartmouth College. The term itself was coined by J. McCarthy at the same conference. The goals of AI stated in this first conference, today, can be characterized as superficial from a pesimist perspective or as open problems from an optimistic perspective. By reading again the goals of this conference we can have a rough idea of the expectations and the

goals of the early AI community: “To proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can be so precisely described that a machine can be made to simulate it” [46]. Despite the fact that even today understanding cognition remains an open problem for computational and clinical scientists, this founding conference of AI stimulated the scientific community and triggered the development of algorithms and methods that became the foundations of modern machine learning. For instance bayesian methods were further developed and studied as part of AI research. Computer programming languages like LISP and PROLOG were developed for serving AI purposes and algorithms and methods like perceptron, backpropagation and artificial neural networks (ANN) were invented.

Computer Science/Engineering: In literature DM is often classified as a branch of computer science (CS). Indeed most of DM research has been driven by CS society. In addition to this, there were several advances in CS area that boosted DM related research. Database modeling together with smart search algorithms made possible the indexing and processing of massive databases [1]. The advances in software level in the area of database modeling and search algorithms was accompanied by a parallel development of semiconductor technologies and computer hardware engineering.

In fact there is a feedback relation between DM and computer engineering that drives the research in both areas. Computer Engineering provides cheaper and larger storage and processing power. On the other hand these new capabilities pose new problems for DM society related to the processing of such amounts of data. These problems create new algorithms and new needs for processing power that is in turns adressed by computer engineering society. The progress in this area can be best described by the so called Moore’s “law” (named after Intel’s cofounder G. E. Moore) that predicted that the number of transistors on a chip will double every 24 months [47]. This simple rule was able to predict the count of transistor per chip at least until today (Fig. 1-1).

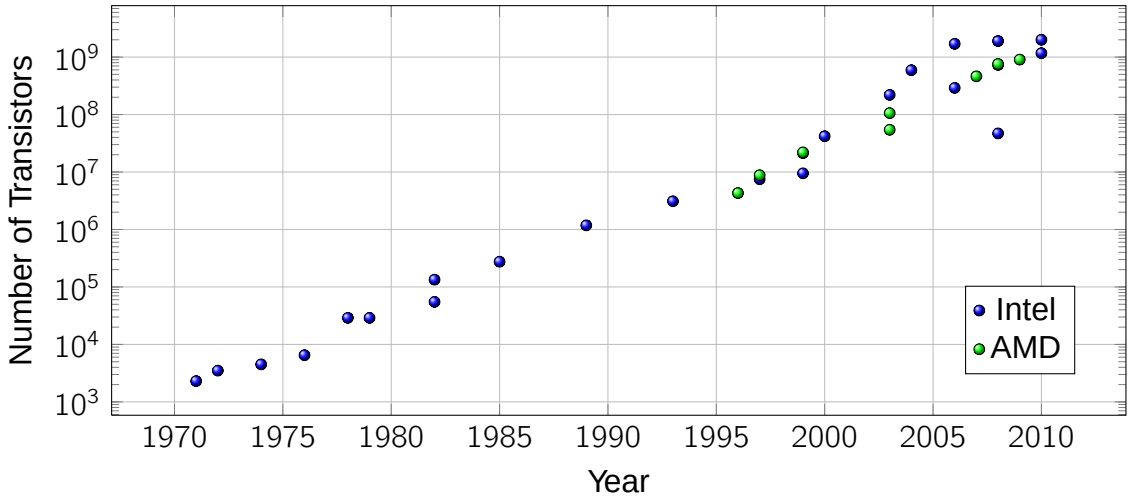


Figure 1-1. Moore's "law" drives the semiconductor market even today. This plot shows the transistor count of several processor's from 1970 until today for two major processor manufacturing companies (Intel and AMD).

Similar empirical "laws" have been stated for hard drive capacity and hard drive price. In act it has been observed that hard drive capacity increases ten times every five years and the cost becomes one tenth every five years. This empirical observation is known as Kryder's "law" (Fig. 1-2) [76]. Last a similar rule which is related to network bandwidth per user (Nielsen's "law") indicates that it increases by 50% annually [49].

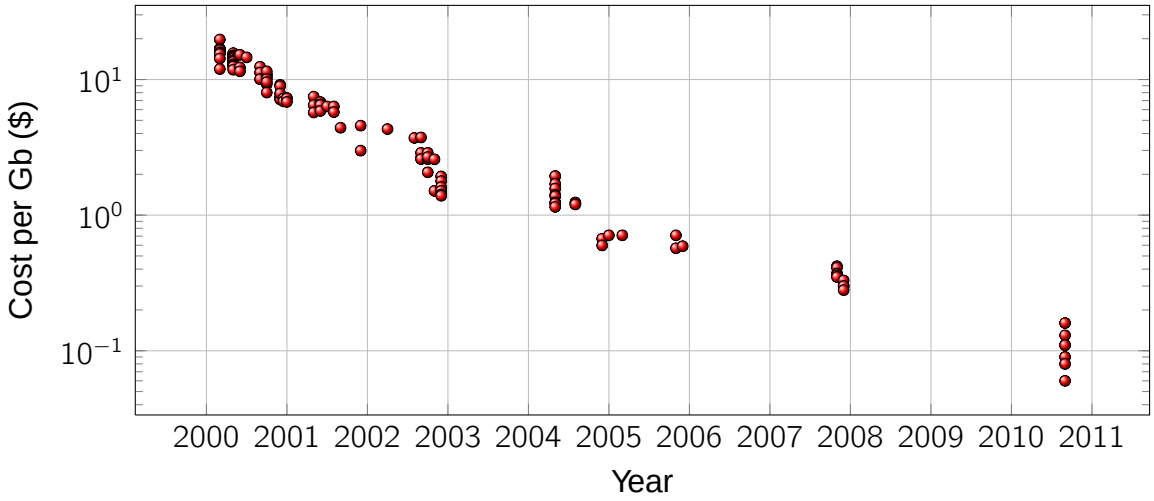


Figure 1-2. Kryder's "law" describes the exponential decrease of computer storage cost over time. this rule is able ti predict approximately the cost of storage space over the last decade.

The fast that computer progress is characterized by all these exponential empirical rules is in fact indicative of the continuous and rapid transformation of DM's needs and capabilities.

Optimization: Mathematical theory of optimization is a branch of mathematics that was originally developed for serving the needs of operations research (OR). It is worth noting that a large amount of data mining problems can be described as optimization problems, some times tractable, sometimes not. For example Principal Component Analysis (PCA) and Fisher's Linear Discriminant Analysis (LDA) are formulated as minimization/ maximization problems of certain statistical criteria [15]. Support Vector Machines (SVMs) can be described as a convex optimization problem [74] and many times linear programming can be used for development of supervised learning algorithms [44] and several metaheuristics have been proposed for adjusting the parameters of supervised learning models [16]. On the other side data mining methods are often used as preprocessing for before employing some optimization model (e.g., clustering). In addition a branch of DM involves network models and optimization problems on networks for understanding the complex relationships that underlie between the nodes and the edges. In this sense optimization is a tool that can be employed in order to solve DM problems. In a recent review paper the interplay of operations research data mining and applications was described by the scheme shown in Fig. 1-3 [51].

Statistics: Statistics set the foundation for many concepts broadly used in data mining. Historically one of the first attempts to understand interconnection between data was Bayes analysis in 1763 [8]. Other concepts include regression analysis, hypothesis testing, PCA and LDA. In modern DM it is very common to maximize or minimize certain statistical properties in order to achieve some grouping or for finding interconnections among groups of data.

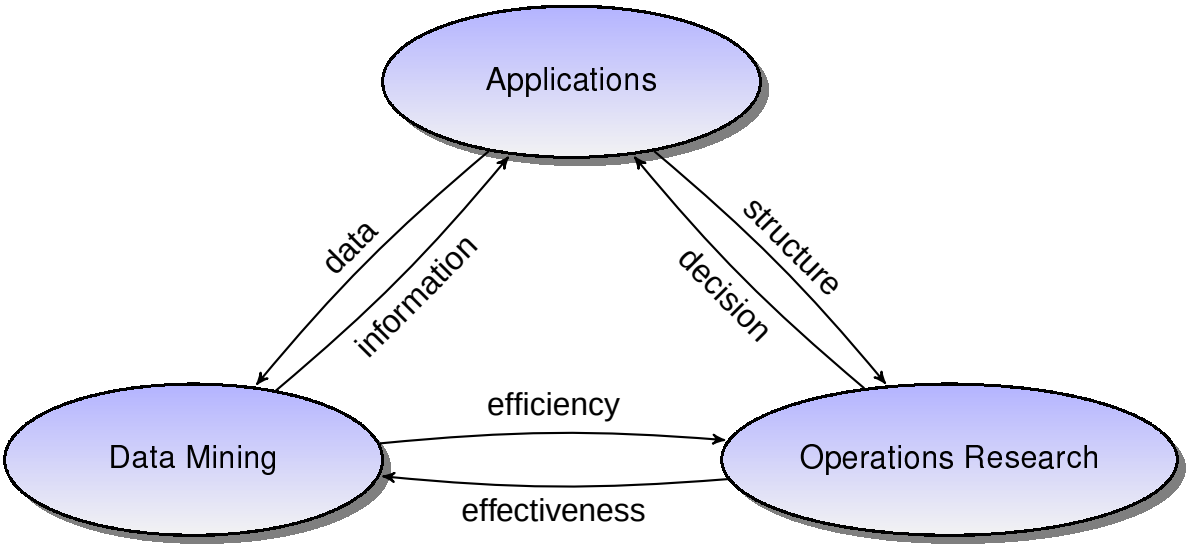


Figure 1-3. The big picture. Scheme capturing the inderdependence among DM, OR and the various application fields.

1.2 A Methodological Perspective

As it can be understood the study of DM passes essentially through the aforementioned disciplines. In addition within the field of DM algorithms and problems can be grouped in certain categories for a more systematic study and understanding. Here we briefly list these categories

- Data preprocessing methods
- Machine learning
  - Unsupervised Learning
  - Supervised Learning
  - Semi supervised learning
- Feature selection - feature extraction
- Data visualization/ representation

Now we will briefly discuss these major categories. Data preprocessing includes all algorithms responsible for data preparation. Time series filtering, outlier detection, data cleaning algorithms and data normalization algorithms fall in this category. Proper data preprocessing is essential for a more efficient performance of learning algorithms.

Machine Learning (ML), is maybe the most important part of data mining. Machine learning is a set of algorithms that have a dataset as input and also maybe some information about it. The output of ML is a set of rules that let us make inference about any new data point.

Unsupervised Learning (UL), sometimes also known as clustering, aims to find associations between data point (clusters). Clustering is performed usually when no information is given about the structure of the dataset. It can be used for exploratory purposes (e.g., identify specific data structure than can be used for more efficient supervised algorithm design). For a more extensive tour to data clustering we refer the reader to [36].

Supervised Learning (SL) is one of the most well known data mining algorithms. SL algorithms are given a set of data points (data samples) with known properties (features) and the classes they belong (labels). Then, the SL algorithm trains a model which at the end of the training phase is capable of deciding on the identity of new data points with unknown label (test dataset). In this category one can include the artificial neural networks, Bayesian classifiers, k-nearest neighbor classification, genetic algorithms and others [15]. If the samples contain qualitative feature values, then the rule based classification can be employed [58, 59]. Especially for the two class general case binary classification, one of the most commonly used approach is SVMs. Originally proposed by Vapnik [74], SVM aims to determine a separation hyperplane from the training dataset. SVM possesses a solid mathematical foundation in optimization theory. If the two classes of data cannot be discriminated with a linear hyperplane, then the problem can be addressed as a nonlinear classification problem. Such problems can be attacked using the so called kernel trick. In this case, original the data sets are embedded in higher dimension spaces where perfect linear separation can be achieved [66]. The combined use of supervised classification methods with kernels is the most common way to address data mining problems.

Semi Supervised Learning lies in between supervised and unsupervised learning. In this case, class labels are known only for a portion of available data and some partial information is given to the algorithm, usually in the form of pair wise constraints (e.g., points a and b belong do not belong to the same class). The goal in this case is to achieve optimal utilization of this information in order to obtain highest predictive accuracy. Biclustering tries to find associate groups of features with corresponding groups of samples. In this way, one can decide on the most important features that are responsible for a group of samples with specific characteristics. It has been extensively used in microarray data analysis for associating genes with specific phenotypes. There are numerous algorithmic approaches to biclustering, ranging from greedy algorithms, spectral biclustering, column reordering and 0-1 fractional programming. For a more mathematical rigorous review about biclustering and their applications we refer the reader to [18, 77]. It is worth mentioning that biclustering can have either supervised or unsupervised version.

Feature selection consists of determining the most important properties (called features) of each data point (called sample) that will be used for training. For example, given a set of people (i.e. sample) and some of their features like weight, height, eye color, hair color, we wish to distinguish the infants from the adults. A feature selection algorithm will try to select a small subset of features that have the largest combined discriminatory power (in this case, maybe weight and height). In case of problems described by hundreds or thousands of characteristics, feature selection permits to reduce the number of variables, with a great advantage in terms of computational time needed to obtain the solution. Examples of algorithms for feature selection are RFE, Relief, [32]. Here we need to point out the difference between feature selection and feature extraction.

Feature extraction consists of the construction of some features that do not necessarily belong to the set of the original features. The significant reduction of the

original number of features, due to a feature selection or feature extraction algorithm, in term of features is called dimensionality reduction which is essential in order to improve the processing time. A standard approach for feature extraction is through PCA.

Data visualization/representation: This last branch of data mining deals with methods where the extracted information can be represented to/visualized by the end user. Massive datasets need special type of algorithms for analysis and representation [1]. One common way to represent information and potential relationship between entities is through graphs (also referred to as networks). Network representation can be very useful in understanding the dynamics that govern a system. Software packages like Cytoscape software [65] deal with representation of complex biological data.

Two other topics related to data mining and more specifically to the supervised machine learning component of the process are the model selection and the cross validation of the learning scheme. Model selection is related to the tuning of the model itself in terms of the parameter selection. The most common method employed in this case is a uniform search over a grid spanned by the parameters. Lately there have been proposed some more advanced model selection methods based on uniform design theory [34]. On the other hand, cross validation of a model is very important, especially when one wants to statistically assure the independence of the output model accuracy from the specific properties of training and testing datasets. For this the given data are partitioned many times in testing and training dataset using some resampling approach. The reported accuracy of the method is the average accuracy of all of the cross validation repetitions. Cross validation methods usually vary by their strategy for partitioning the original data. Most of the well-studied cross validation techniques include k-fold, leave-one-out and hold out cross validation.

All the described methods are implemented in many open source problem solving environments like R (http://www.r-project.org/), Weka (http://www.cs.waikato.ac.nz/ml/weka) and Octave (http://www.gnu.org/software/octave/). They provide

simple graphical user interfaces that can be used also by non expert users. Some of these methods come in other popular technical computing programming languages such as Matlab and Mathematica.

1.3 A Brief History of Robustness

The term robust is used extensively in engineering and statistics literature. In engineering it is often used in order to denote error resilience in general (e.g., robust methods are these that are not affected that much for small error interferences). In statistics robust is used to describe all these methods that are used when the model assumptions are not exactly true. For example, it can be used when variables do not follow exactly the assumed distribution (existence of outliers). In optimization (minimization of maximization) robust is used in order to describe the problem of finding the best solution given that the problem data are not fixed but obtain their values within a well defined uncertainty set. Thus if we consider the minimization problem (without loss of generality)

$$\min_{x \in \mathcal{X}} \; f(A, x) \tag{1–1a}$$

where A accounts for all the parameters of the problem that are consider to be fixed numbers, and $f(\cdot)$ is the objective function, the robust counterpart (RC) problem is going to be a $\min - \max$ problem of the following form

$$\min_{x \in \mathcal{X}} \max_{A \in \mathcal{A}} \; f(A, x) \tag{1–2a}$$

where $\mathcal{A}$ is the set of all admissible perturbations. The maximization problem over the parameters A accounts, usually, for a worst case scenario. The objective of robust optimization is to determine the optimal solution when such a scenario occurs. In real data analysis problems it is very likely that data might be corrupted, perturbed or subject to error related to data acquisition. In fact most of modern data acquisition methods are prone to errors. The most usual source of such errors is noise which is usually

associated with instrumentation itself or due to human factors (when the data collection is done manually). Spectroscopy, microarray technology, electroencephalography (EEG) are some of the most commonly used data collection technologies that are subject to noise. Robust optimization is employed not only when we are dealing with data imprecisions but also when we want to provide stable solutions that can be used in case of input modification. In addition it can be used in order to avoid selection of "useless" optimal solution i.e. solutions that change drastically for small changes of data. Especially in case where optimal solution cannot be implemented precisely, due to technological constraints, we wish that the next best optimal solution will be feasible and very close to the one that is out of our implementation scope. For all these reasons Robust methods and solutions are highly desired.

In order to outline the main goal and idea of robust optimization we will use the well studied exmple of linear progrmming (LP). In this problem we need determine the global optimum of a linear function over the feasible region defined by a set of linear system.

$$\min \quad c^T x \tag{1–3a}$$

$$\text{s.t.} \quad Ax = b \tag{1–3b}$$

$$x \geq 0 \tag{1–3c}$$

where $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$. In this formulation x is the decision variable and A, b, c are the data and they have constant values. LP for fixed data values can be solved efficiently by many algorithms (e.g., SIMPLEX) and it has been shown that LP can be solved in polynomial time [37].

In the case of uncertainty we assume that data are not fixed but they can take any values within an uncertainty set. Then the robust counterpart (RC) problem is to find a vector x that minimizes the objective of Eq.1–3a for the "worst case" allowed perturbation. This worst case problem can be stated as a maximization problem with

respect to A, b, c. The whole process can be stated as the following $\min - \max$ problem

$$\min_{x} \max_{A,b,c} \; c^T x \tag{1–4a}$$

$$\text{s.t.} \; Ax = b \tag{1–4b}$$

$$x \geq 0 \tag{1–4c}$$

$$A \in \mathcal{A}, b \in \mathcal{B}, c \in \mathcal{C} \tag{1–4d}$$

where $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are the uncertainty sets of A, b, c correspondingly. Problem of Eq. 1–4 can might be tractable or untractable based on the uncertainty sets properties. For example it has been shown that if the columns of A follows ellipsoidal uncertainty constraints the problem is polynomially tractable [10]. Bertsimas and Sim showed that if the coefficients of A matrix are between a lower and an upper bound then this problem can be still solved with linear programming [12]. Also Bertsimas et al. have shown that an uncertain LP with general norm bounded constriants is a convex programming problem [11].

1.3.1 Robust optimization vs Stochastic Programming

Here it is worth noting that Robust Optimization is not the only approach for handling uncertainty in optimization problems. In the robust framework the information about uncertainty is given in a rather deterministic form of worst case bounding constraints. In a different framework one might not require the solution to be feasible for all data realization but to obtain the best solution given that problem data are random variables following a specific distribution. This is of particular interest when the problem possesses some periodic properties and historic data are available. In this case the parameters of such a distribution could efficiently be estimated through some model fitting approach. Then a probabilistic description of the constraints can be obtained and the corresponding optimization problem can be classified as a stochastic programming

problem. In this manner the stochastic equivalent of linear programming will be

$$\min_{x,t} \quad t \tag{1–5a}$$

$$\text{s.t.} \quad Pr\{c^T x \leq t, Ax \leq b\} \geq p \tag{1–5b}$$

$$x \geq 0 \tag{1–5c}$$

where c, A and b entries are random variable that follow some known distribution, p is a non negative number less than 1 and $Pr\{\cdot\}$ some probability function. This non deterministic description of the problem does not guarrantee that the provided solution would be feasible for all data set realizations but provide a less conservative optimal solution taking into consideration the distribution based uncertainties. Although the stochastic approach might be of more practical value in some cases there some assumptions associated with them that one should be aware of before using them [9]

1. The problem must be of stochastic nature and that indeed there is a distribution hidden behind each variable.
2. Our solution depends on our ability to determine the correct distribution from the historic data.
3. We have to be sure that our problem accepts probabilistic solutions i.e. a stochastic problem solution might not be immunized against a catastrophic scenario and a system might be vulnerable against rare event occurrence.

For this the choice of the approach strictly depends on the nature of the problem as well as the available data. For an introduction to stochastic programming we refer the reader to [14]. For a complete overview of robust optimization we refer the reader to [9].

1.3.2 Thesis Goal and Structure

In this work our goal is to explore in particular the optimization models employed in some of the well studied DM algorithms, formulate their RC problem and provide algorithms in order to solve them. The rest of this work is organized as follows. In Chapter 2 we give an overview of some of the most important data mining algorithms in the absence of uncertainty. We discuss Linear Least Squares (LLS), PCA, Fisher's LDA,

SVM and Regularized Generalized Eigenvalue Classification (ReGEC). At the end of their presentation we give a comparative study of some of them on a real classification problem from biomedicine [78]. In Chapter 3 though Chapter 6 we discuss the robust counterpart formulation of Linear Least Squares, PCA, LDA, SVM [80] whereas in Chapter 7 we give the robust formulation of Regularized Generalized Eigenvalue Classification [79]. In Chapter 8 we give discuss further research directions and provide some conclusions.

CHAPTER 2
DATA MINING METHODS

In this section we will give the mathematical description as several data mining methods. In Chapter 3 through Chapter 6 we will deal with their robust formulations. We will revised the basic formulations of the following data mining algorithms

1. Linear Least Squares (LLS)
2. Principal Component Analysis (PCA)
3. Linear Discriminant Analysis (LDA)
4. Support Vector Machines (SVM)
5. Regularized Generalized Eigenvalue Classification (ReGEC)

The first algorithm is broadly used for linear regression , but can also be used for supervised learning, whereas the last three methods are typical supervised learning algorithms. The goal is to give the necessary background before introducing their robust formulations. We conclude by providing a comparative case study on a supervised learning problem from Raman spectroscopy.

2.1 Linear Least Squares

In the original linear least squares (LS) problem one needs to determine a linear model that approximates “best” a group of samples (data points). Each sample might correspond to a group of experimental parameters or measurements and each individual parameter to a feature or, in statistical terminology, to a predictor. In addition each sample is characterized by an outcome which is defined by a real valued variable and might correspond to an experimental outcome. Ultimately we wish to determine a linear model able to issue outcome prediction for new samples. The quality of such a model can be determined by a minimum distance criterion between the samples and the linear model. Therefore if n data points, of dimension m each, are represented by a matrix $A \in \mathbb{R}^{n \times m}$ and the outcome variable by a vector $b \in \mathbb{R}^n$ (one entry corresponding to a row of matrix A) we need to determine a vector $x \in \mathbb{R}^m$ such that the residual error,

expressed by a norm, is minimized. This can be stated as

$$\min_x \|Ax - b\|_2^2 \tag{2–1}$$

Where $\|\cdot\|_2$ is the Euclidean norm of a vector. The objective function value is also called residual and denoted $r(A, b, x)$ or just r. The geometric interpretation of this problem is to find a vector x such that the sum of the distances between the points represented by the rows of matrix A and the hyperplane defined by $x^T w - b = 0$ is minimized. In this sense this problem is a first order polynomial fitting problem. Then by determining the optimal x vector we will be able to issue predictions for new samples by just computing their inner product with x. A 2D example can be seen in Fig. 2.1. In this case the data matrix will be $A = [a \;\; e] \in \mathbb{R}^{n\times 2}$ where a is the predictor variable and e a column vector full of one that accounts for the constant term.

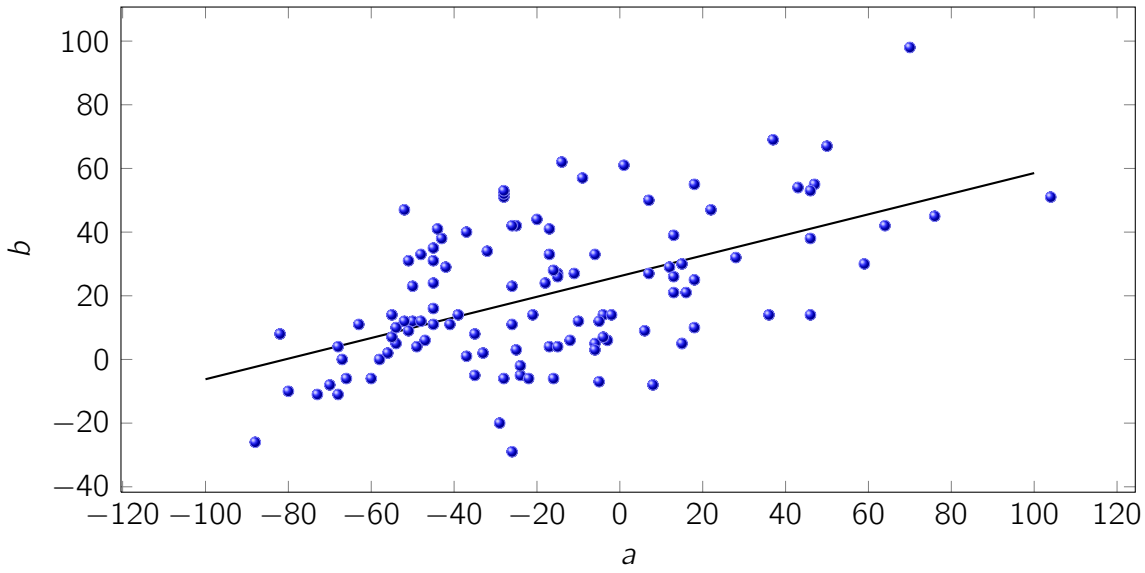


Figure 2-1. The Signle input single outcome case. This is a 2D example the predictor are represented by the a variable and the outcome by vertical axis b.

The problem can be solved, in its general form, analytically since we know that the global minimum will be at a KKT point (since the problem is convex and unconstrained). the lagrangian equation $\mathcal{L}_{LLS}(x)$ will be given by the objective function itself and the Karush Kuhn Tucker (KKT) points can be obtained by solving the following equation

$$\frac{d\mathcal{L}_{LLS}(x)}{dx} = 0 \Leftrightarrow 2A^TAx = A^Tb \tag{2–2}$$

In case that A is of full row, that is $\mathrm{rank}(A) = n$, matrix A^TA is invertible and we can write

$$x_{LLS} = (A^TA)^{-1}A^Tb \triangleq A^{\dagger}b \tag{2–3}$$

Matrix $A^{\dagger}$ is also called pseudoinverse or Moore Penrose matrix. It is very common that this full rank assumption is not always valid. In such case the most common way to address the problem is through regularization. One of the most famous regularization techniques is the one known as Tikhonov regularization. In this case instead of Eq. 2–1 we consider the following one

$$\min_x \left(\|Ax - b\|^2 \pm \delta\|x\|^2\right) \tag{2–4}$$

by using the same methodology we obtain

$$A^T(Ax - b) \pm \delta x = 0 \Leftrightarrow (A^TA \pm \delta I)x = A^Tb \tag{2–5}$$

where I is a unit matrix of appropriate dimension. Now even in case that A^TA is not invertable $A^TA + \delta I$ is, so we can compute x by

$$x_{RLLS} = (A^TA + \delta I)^{-1}A^Tb \tag{2–6}$$

This type of least square solution is also known as ridge regression. The parameter δ is some kind of penalty coefficient that controls the tradeoff between solution with low least square error and low euclidean norm solution. Originally regularization was proposed in order to overcome this practical difficulty that arises in real problem. The value of δ is determined usually by trial and error and its magnitude is usually smaller compared to the entries of data matrix. In Fig. 2.1 we can see how the least squares plane changes for different values of delta.

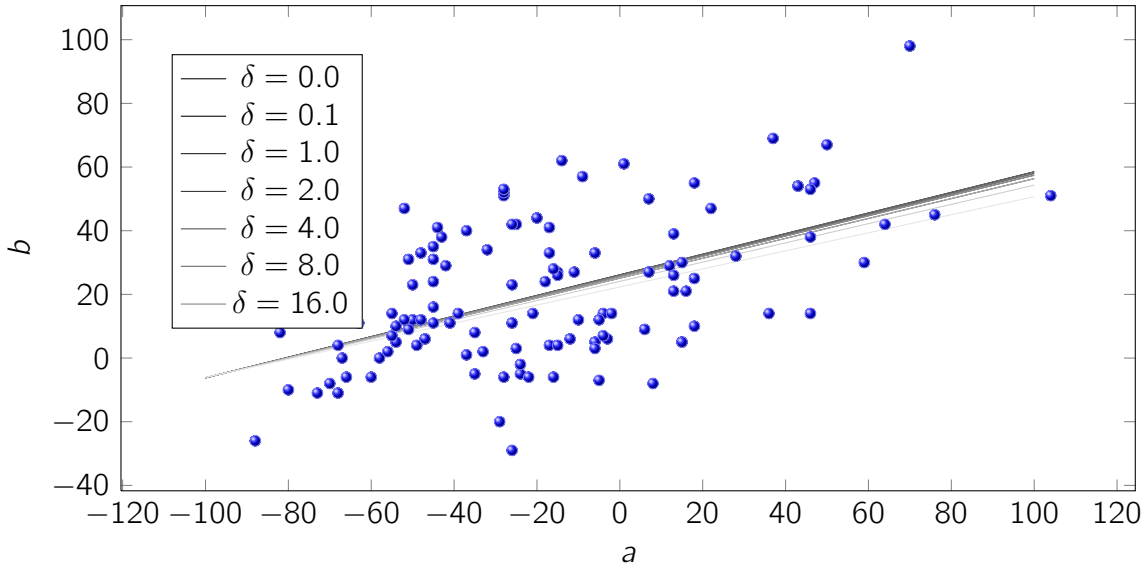


Figure 2-2. LLS and regularization. Change of linear least squares solution with respect to different δ values. As we can observed, in this particular example, the solution does not change drastically with regularization.

In Chapter 3 we will examine the relation between robust linear least squares and robust optimization.

2.1.1 Weighted Linear Least Squares

A slight, and more general, modification of the original least squares problem is the weighted least squares problem. In this case we have the following minimization problem

$$\min_x \quad r^T W r = \min_x \quad (Ax - b)^T W (Ax - b) = \min_x \|W^{1/2}(Ax - b)\| \tag{2–7}$$

Where W is the weight matrix. Note that this is a more general formulation since for $W = I$ the problem reduces to Eq. 2–1. The minimum can be again obtained by the solution of the corresponding KKT systems which is

$$2A^T W(Ax - b) = 0 \tag{2–8}$$

and gives the following solution

$$x_{RLLS} = (A^T WA)^{-1} A^T Wb \tag{2–9}$$

Assuming that $A^T WA$ is invertible. In different case regularization can be again employed yielding in the weighted regularized least squares problem

$$\min_x \left(\|W^{1/2}(Ax - b)\|^2 + \delta \|x\|^2 \right) \tag{2–10}$$

that attains its global minimum for

$$x_{RWLLS} = (A^T WA + \delta I)^{-1} AWb \tag{2–11}$$

Next we will discuss some practical approaches for computing least square solution for all the discussed variations of the problem.

2.1.2 Computational Aspects of Linear Least Squares

Computationally least squares solution can be obtained by computing an inverse matrix and applying a couple of matrix multiplications. In fact matrix inversion is avoided in practise, especially due to the high computational cost, other decomposition methods can be employed. For completion we will include three of the most popular here.

Cholesky factorization: In case tat matrix A is of full rank then AA^T is invertible and can be decomposed wit Cholesky decomposition is a product LL^T were L is lower triangular matrix. Then Eq. 2–2 can be written as

$$LL^T x = A^T b \tag{2–12}$$

can be solved by a forward substitution followed by a backward substitution. In case that A is not of full rank then this procedure can be applied to the regularized problem of Eq. 2–5.

QR factorization: An alternative method is the one of QR decomposition. In this case we decompose matrix AA^T into a product of two matrices were the first matrix Q is orthogonal and the second matrix R is upper triangular. This decomposition again requires data matrix A to be of full row rank. Orthogonal matrix Q as the property

$QQ^T = I$ thus the problem is equivalent to

$$Rx = Q^T A^T b \tag{2–13}$$

The last can be solved by backward substitution.

Singular Value Decomposition (SVD): This last method does not require full rank of matrix A. It uses the singular value decomposition of A

$$A = U\Sigma V^T \tag{2–14}$$

where U and V are orthogonal matrices and Σ is diagonal matrix that has the singular values. Every matrix wit real elements as a SVD and furthermore it can be proved that a matrix is of full row rank if and only if all of its singular values are non zero. Substituting A matrix with its SVD decomposition we get

$$AA^T x = (U\Sigma V^T)(V\Sigma U^T)x = U\Sigma^2 U^T x = A^T b \tag{2–15}$$

and finally

$$x = U(\Sigma^2)^\dagger U^T A^T b \tag{2–16}$$

The matrix $(\Sigma^2)^\dagger$ can be computed easily by inverting its non-negative entries. If A is full rank ten all singular values are non zero and then $(\Sigma^2)^\dagger = (\Sigma^2)^{-1}$. Although SVD can be applied to any kind of matrix it is computationally expensive and some times is not preferred especially when processing massive datasets.

2.1.3 Least Absolute Shrinkage and Selection Operator

An alternative regularization technique for the same problem is the one of least absolute shrinkage and selection operator (LASSO) [70]. In this case the regularization term contains a first norm term $\delta\|x\|_1$. Thus we ave the following minimization problem

$$\min_x \left(\|Ax - b\|^2 + \delta\|x\|_1\right) \tag{2–17}$$

Although this problem cannot be solved analytically as the one obtained after Tikhonov regularization, sometimes it is preferred as it provides sparse solutions. That is the solution x vector obtained by LASSO has more zero entries. For this this approach has a lot of applications in compressive sensing [5, 19, 43]. As it will be discussed later this regularization possesses further robust properties as it can be obtained trough robust optimization for a specific type of data perturbations.

2.2 Principal Component Analysis

The PCA transformation is a very common and well studied data analysis technique that aims to identify some linear trends and simple patterns in a group of samples. It has application in several areas of engineering. It is popular due to (it requires only an eigendecomposition, or singular value decomposition).

There are two alternative optimization approaches for obtaining principal component analysis solution, the one of variance maximization and the one of minimum error formulation. Both start with a "different" initial objective and end up providing the same solution. It is necessary to study and understand both of these alternative approaches. At this point we need to note that we assume that the mean of the data samples is equal to zero. In case this is not true we need to substract the maple mean as part of preprocessing.

2.2.1 Maximum Variance Approach

In this case we try to find a subspace of dimensionality $p < m$ for which the variability of the projection of the points is maximized. if we denote with $\bar{x}$ the sample arithmetic mean

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2–18}$$

then the variance of the projected data on a subspace defined by the direction vector u will be

$$\frac{1}{n}\sum_{i=1}^{N}\left(u^T x_i - u^T\bar{x}\right)^2 = \frac{1}{n}\sum_{i=1}^{N}\left(u^T(x_i-\bar{x})\right)^2 = u^T\left(\frac{\sum_{i=1}^{n}(x_i-\bar{x})^T(x_i-\bar{x})}{n}\right)u \tag{2–19}$$

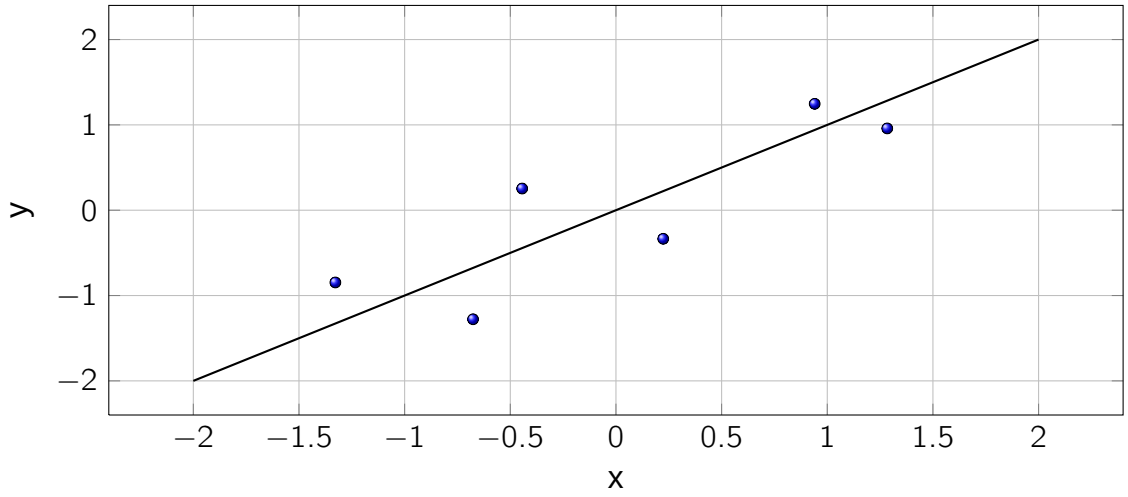


Figure 2-3. Two paths for PCA. PCA has two alternative optimzation formulations that result in the same outcome. One is to find a space where the projection of the original data will have maximum variance and the second is to find the subspace such that the projection error is minimized.

and given that the variance covariance matrix is defined by

$$S = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})}{n} \tag{2–20}$$

Then Eq. 2–19 can be written in matrix notation as

$$u^T S u \tag{2–21}$$

If we restrict, without loss of generality, our solution space just to the vectors u with unit Euclidean norm then PCA problem can be expressed as the following optimization problem

$$\max_{u} \; u^T S u \tag{2–22a}$$

$$\text{s.t.} \; \; u^T u = 1 \tag{2–22b}$$

The Lagrangian $\mathcal{L}_{PCA}(u, \lambda)$ for this problem will be

$$\mathcal{L}_{PCA}(u, \lambda) = u^T S u + \lambda(u^T u - 1) = 0 \tag{2–23}$$

where λ is the lagrange multiplier associated with the single constraint of the problem. The optimal points will be given by the roots of the Lagrangian (since S is positive semidefinite the problem is convex minimization). Thus

$$Su = \lambda u \tag{2–24}$$

This equation is satisfied by all the eigenpairs (λ_n, u_n), $i = 1, \dots, n$ where

$$\lambda_1 \leq \lambda_2 \leq \dots \lambda_n \tag{2–25}$$

are the ordered eigenvalues and u_i's are the corresponding eigenvectors. The objective function is maximized for $u = u_n, \lambda = \lambda_n$ and the optimal objective function value is $u_n^T S u_n = u_n^T \lambda_n u_n = \lambda_n \|u_n\|^2 = \lambda_n$.

2.2.2 Minimum Error Approach

An alternative derivation of PCA can be achieved through a different path. In this second approach the objective is to rotate the original axis system such that the projection error of the dataset to the rotated system will be minimized. Thus we define a set of basis vector $\{u_i\}_{i=1}^n$. As soon as we do this we are able to express every point, including our dataset points, as a linear combination of the basis vectors.

$$x_k = \sum_{i=1}^{m} a_{ki} x_i = \sum_{i=1}^{m} \left(x_k^T u_i\right) u_i \tag{2–26}$$

Our purpose is to approximate every point x_k with $\tilde{x}_k$ using just a subset $p < m$ of the basis. Thus the approximation will be

$$\tilde{x}_n = \sum_{i=1}^{p} \left(x_n^T u_i\right) u_i + \sum_{i=p+1}^{m} \left(\bar{x}^T u_i\right) u_i \tag{2–27}$$

Thus the approximation error can be computed through as a squared Euclidean norm summation over all data points

$$\sum_{k=1}^{n} \|x_k - \tilde{x}_k\|^2 \tag{2–28}$$

we can obtain a more compact expression for $x_k - \tilde{x}_k$

$$x_k - \tilde{x}_k = \sum_{i=1}^{m} \left(x_k^T u_i\right) u_i - \left(\sum_{i=1}^{p} \left(x_n^T u_i\right) u_i + \sum_{i=p+1}^{m} \left(\bar{x}^T u_i\right) u_i \right) \tag{2–29a}$$

$$= \sum_{i=p+1}^{m} \left((x_k^T u_i) u_i - (\bar{x}_k^T u_i) u_i \right) \tag{2–29b}$$

Then the solution can be estimated by minimizing Eq. 2–27 and by constraining the solution to the vectors of unit Euclidean norm

$$\min_u \sum_{i=p+1}^{m} u_i^T S u_i = \max_u \sum_{i=1}^{p} u_i^T S u_i \tag{2–30a}$$

$$\text{s.t.} \quad u^T u = 1 \tag{2–30b}$$

This optimization problem is similar to the one obtained through the maximum variance approach (the only difference is that we are looking for the first p components instead of just one) and the solution is given by the first p eigenvectors that correspond to the p highest eigenvalues (can be proved through analytical solution of KKT equation).

2.3 Linear Discriminant Analysis

The LDA approach is a fundamental data analysis method originally proposed by R. Fisher for discriminating between different types of flowers [27]. The intuition behind the method is to determine a subspace of lower dimension, compared to the original data sample dimension, in which the data points of the original problem are “separable”. Separability is defined in terms of statistical measures of mean value and variance. One of the avdvantages of LDA is that the solution can be obtained by solving a generalized eigenvalue system. This allows for fast and massive processing of data samples. In addition LDA can be extended to non linear discriminant analysis through the kernel trick [7]. The original algorithm was proposed for binary class problems but multi class generalizations have been proposed [60]. Here we will discuss both starting from the simple two class case.

Let $x_1, \dots, x_p \in \mathbb{R}^m$ be a set of p data samples belonging to two different class sets, A and B. For each class we can define the sample means

$$\bar{x}_A = \frac{1}{N_A} \sum_{x \in A} x, \; \bar{x}_B = \frac{1}{N_B} \sum_{x \in B} x \tag{2–31}$$

where N_A, N_B are the number of samples in A and B respectively. Then for each class we can define the positive semidefinite scatter matrices described by the equations

$$S_A = \sum_{x \in A} (x - \bar{x}_A)(x - \bar{x}_A)^T, S_B = \sum_{x \in B} (x - \bar{x}_B)(x - \bar{x}_B)^T \tag{2–32}$$

Each of these matrices expresses the sample variability in each class. Ideally we would like to find a hyperplane for which if we project the data samples their variance would be minimal. That can be expressed as

$$\min_{\phi} \left(\phi^T S_A \phi + \phi^T S_B \phi \right) = \min_{\phi} \phi^T (S_A + S_B) \phi \tag{2–33}$$

On the other side, the scatter matrix between the two classes is given by

$$S_{AB} = (\bar{x}_A - \bar{x}_B)(\bar{x}_A - \bar{x}_B)^T. \tag{2–34}$$

According to Fisher's intuition we wish to find a hyperplane in order to maximize the distance between the means between the two classes and at the same time to minimize the variance in each class. Mathematically this can be described by maximization of Fisher's criterion

$$\max_{\phi} \mathcal{J}(\phi) = \max_{\phi} \frac{\phi^T S_{AB} \phi}{\phi^T S \phi}. \tag{2–35}$$

For this optimization problem can have infinitely solution wit the same objective function value. That is for a solution ϕ^* all the vectors $c \cdot \phi^*$ give exactly the same value. For this, without loss of generality, we replace the denominator wit an equality constraint

in order to chose only one solution. Then the problem becomes

$$\max_{\phi} \ \phi^T S_{AB} \phi \tag{2–36a}$$

$$\text{s.t.} \ \ \phi^T S \phi = 1 \tag{2–36b}$$

The Lagrangian associated with this problem is

$$\mathcal{L}_{LDA}(x, \lambda) = \phi^T S_{AB} \phi - \lambda(\phi^T S \phi - 1) \tag{2–37}$$

where λ is the lagrange multiplier that is associated with the constraint of Eq. 2–36b. Since S_{AB} is positive semidefinite the problem is convex and the global minimum will be at the point for which

$$\frac{\partial \mathcal{L}_{LDA}(x, \lambda)}{\partial x} = 0 \Leftrightarrow S_{AB}\phi - \lambda S \phi = 0 \tag{2–38}$$

The optimal ϕ can be obtained as the eigenvector that corresponds to the smallest eigenvalue of the following generalized eigensystem

$$S_{AB}\phi = \lambda S \phi \tag{2–39}$$

Multi-class LDA is a natural extension of the previous case. Given n classes, we need to redefine the scatter matrices. The intra-class matrix becomes

$$S = S_1 + S_2 + \cdots + S_n \tag{2–40}$$

while the inter-class scatter matrix is given by

$$S_{1,\ldots,n} = \sum_{i=1}^{n} p_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T \tag{2–41}$$

where p_i is the number of samples in the i-th class, $\bar{x}_i$ is the mean for each class, and $\bar{\mathbf{x}}$ is the total mean vector calculated by

$$\bar{x} = \frac{1}{p} \sum_{i=1}^{n} p_i \bar{x}_i.$$

The linear transformation ϕ we wish to find can be obtained by solving the following generalized eigenvalue problem

$$S_{1,\dots,n}\phi = \lambda S\phi.$$

LDA can be used in order to identify which are the most significant features together with the level of significance as expressed by the corresponding coefficient of the projection hyperplane. Also LDA can be used for classifying unknown samples. Once the transformation ϕ is given, the classification can be performed in the transformed space based on some distance measure d. The class of a new point **z** is determined by

$$\text{class}(z) = \arg\min_n \{d(z\phi, \bar{x}_n\phi)\} \tag{2–42}$$

where $\bar{x}_n$ is the centroid of n-th class. This means that first we project the centroids of all classes and the unknown points on the subspace defined by ϕ and the we assign the points to the closest class with respect to d.

2.3.1 Generalized Discriminant Analysis

In the case that the linear projection model cannot interpret the data we need to obtain a non linear equivalent of LDA [7]. This can be achieved by the well studied kernel trick. In this case we embed the original data points (input space) to a higher dimension space (feature space) and then we solve the linear problem. The projection of this linear discriminant in the feature space is a non-linear discriminant in the input space. This kernel embedding is performed through a function $\kappa : \mathbb{R}^m \mapsto \mathbb{R}^q$ where q is the dimension of the feature space. Then the arithmetic mean on the feature space for each class will be

$$\bar{x}_1^\kappa = \frac{1}{N_A}\sum_{x\in A}\kappa(x), \dots, \bar{x}_n^\kappa = \frac{1}{N_B}\sum_{x\in B}\kappa(x), \tag{2–43}$$

Then the scatter matrices for each class in the feature space would be

$$V_1 = \sum_{x\in A}(\kappa(x) - \bar{x}_A^\kappa)(\kappa(x) - \bar{x}_A^\kappa)^T, \dots, V_n = \sum_{x\in B}(\kappa(x) - \bar{x}_B^\kappa)(\kappa(x) - \bar{x}_B^\kappa)^T \tag{2–44}$$

and the variance between classes in the feature space will be

$$B_{1,\ldots,n} = \sum_{i=1}^{n} p_i(\bar{x}_i^{\kappa} - \bar{x}^{\kappa})(\bar{x}_i^{\kappa} - \bar{x}^{\kappa})^T \tag{2–45}$$

Fisher's criterion in the feature space will be

$$\min_y \mathcal{J}^k(y) = \frac{y^T(B_{1,\ldots,n})y}{y^T(\sum_{i=1}^{n} V_i)y} \tag{2–46}$$

and the solution can be obtained from the eigenvector that corresponds to the smallest eigenvalue of the generalized eigensystem $B_{1,\ldots,n}y = \lambda\left(\sum_{i=1}^{n} V_i\right)y$. There are several functions that are used as kernel function generally in data mining literature. For a more extensive study of kernel theoretical properties we refer the reader to [66].

2.4 Support Vector Machines

Support Vector Machines (SVM) is one of the most well known supervised classification algorithm. It was originally proposed by V. Vapnik [74]. The intuition behind the algorithm is that we wish to obtain a hyperplane that "optimally" separates two classes of training data. The power of SVM lies in the fact that it has minimal generalization error (at least in the case of two classes) and the solution can be obtained computationally efficiently since it can be formulated as a convex programming problem. It's dual formulation can be used in order to boost the performance even more. As for other supervised classification methods SVM original formulation refers to binary classification problems.

Given a set of data points x_i, $i = 1, \ldots, n$ and an indicator vector $d \in \{-1, 1\}^n$ with the class information of the data points we aim to find a hyperplane defined by (w, b) such that the distance of between the hyperplane and the closest of the data points of each class (support vectors). This can be expressed as the following optimization

problem

$$\min_{w,b} \ \frac{1}{2}\|w\|^2 \tag{2–47a}$$

$$\text{s.t.} \ \ d_i\left(w^T x_i + b\right) \geq 1, \quad i = 1, \dots, n \tag{2–47b}$$

For this problem the Lagrangian equation will be

$$\mathcal{L}_{SVM}(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_i \left[d_i\left(w^T x_i + b\right) - 1\right] \tag{2–48}$$

where $\alpha = [\alpha_1 \ \alpha_2 \dots \alpha_n]$ are Lagrange multipliers. In order to determine them we need to take the partial derivatives with respect to each decision variable and set them equal to zero.

$$\frac{\partial \mathcal{L}_{SVM}(w, b, \alpha)}{\partial w} = 0 \Leftrightarrow w = \sum_{i=1}^{n} \alpha_i d_i x_i \tag{2–49a}$$

$$\frac{\partial \mathcal{L}_{SVM}(w, b, \alpha)}{\partial b} = 0 \Leftrightarrow \sum_{i=1}^{n} \alpha_i d_i = 0 \tag{2–49b}$$

And if we substitute in Eq. 2–48 we get

$$\mathcal{L}_{SVM}(w, b, \alpha) = \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j d_i d_j \langle x_i x_j \rangle - \sum_{i,j=1}^{n} \alpha_i \alpha_j d_j \langle x_j, x_i \rangle + b \sum_{i=1}^{n} \alpha d_i + \sum_{i=1}^{n} \alpha_i \tag{2–50a}$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j d_i d_j \langle x_i, x_j \rangle \tag{2–50b}$$

Then we can express the dual of the original SMV problem as follows

$$\max \ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j d_i d_j \langle x_i, x_j \rangle \tag{2–51a}$$

$$\text{s.t.} \ \sum_{i=1}^{n} d_i \alpha_i = 0 \tag{2–51b}$$

$$\alpha_i \geq 0 \qquad\qquad i = 1, \dots, n \tag{2–51c}$$

The last is a also a convex quadratic problem that can be solved efficiently. Once the optimal dual variables $\alpha_i^*, i = 1, \dots, n$ are found then the optimal separation

hyperplane w^* can be obtained from

$$w^* = \sum_{i=1}^{n} d_i \alpha_i^* x_i \tag{2–52}$$

Note that b does not appear in the dual formulation thus it should be estimated through the primal constraints

$$b^* = -\frac{\max_{d_i=-1}\langle w^* x_i\rangle + \min_{d_i=1}\langle w^* x_i\rangle}{2} \tag{2–53}$$

This model can give a separation hyperplane in case that the two classes are linearly separable. In case that this assumption does not hold the optimization problem will become infeasible. For this we need to slightly modify this original hard margin classification model so that it remains feasible even in case that some points are misclassified. The idea is to allow misclassified points but at the same time to penalize this misclassifications making it a less favorable solution.

$$\min_{w,b,\xi_i} \ \frac{1}{2}\left(\|w\|^2 + C\sum_{i=1}^{n}\xi_i^2\right) \tag{2–54a}$$

$$\text{s.t.} \ \ d_i\left(w^T x_i + b\right) \geq 1 - \xi_i, \quad i = 1, \ldots, n \tag{2–54b}$$

where C is the penalization parameter. Note that this model becomes the same as in Eq. 2–47a as $C \mapsto +\infty$. This modified SVM formulation is known as soft margin SVM. This formulation can be seen as a regularized version of formulation of Eq. 2–47a. The Lagrangian of Eq. 2–54a will be

$$\mathcal{L}_{SVM-S}(w, b, \xi, \alpha) = \frac{1}{2}w^T w + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 - \sum_{i=1}\alpha_i[d_i\left(w^T x_i + b - 1 + \xi\right)] \tag{2–55}$$

where, again, α_i are appropriate Lagrangian multipliers. The dual formulation can be easily obtained in a way similar to the hard margin classifier. The only difference is that now we will have an additional equation associated to the new ξ variables. Setting the derivation of the Lagrangian equal to zero for each of the decision variables gives the

following KKT system

$$\frac{\partial \mathcal{L}_{SVM}(w, b, \xi, \alpha)}{\partial w} = 0 \Leftrightarrow \; w = \sum_{i=1}^{n} d_i \alpha_i x_i \tag{2–56a}$$

$$\frac{\partial \mathcal{L}(w, b, \xi, \alpha)}{\partial \xi} = 0 \Leftrightarrow C\xi = \alpha \Leftrightarrow \xi = \frac{1}{C}\alpha \tag{2–56b}$$

$$\frac{\partial \mathcal{L}_{SVM}(w, b, \xi, \alpha)}{\partial b} = 0 \Leftrightarrow \sum_{i=1}^{n} d_i \alpha_i = 0 \tag{2–56c}$$

Substituting these equation to the primal Lagrangian we obtain

$$\mathcal{L}_{SVM-S}(w, b, \xi, \alpha) = \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{n} \xi_i^2 - \sum_{i=1} \alpha_i \left[d_i \left(w^T x_i + b - 1 + \xi \right) \right] \tag{2–57a}$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j d_i d_j \langle x_i, x_j \rangle + \frac{1}{2C} \langle \alpha, \alpha \rangle - \frac{1}{C} \langle \alpha, \alpha \rangle \tag{2–57b}$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j d_i d_j \langle x_i, x_j \rangle - \frac{1}{2C} \langle \alpha, \alpha \rangle \tag{2–57c}$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j d_i d_j \left(\langle x_i, x_j \rangle + \frac{1}{C} \delta_{ij} \right) \tag{2–57d}$$

where δ_{ij} is the kronecker δ where it is equal to 1 when $i = j$ and it is zero otherwise. The dual formulation of the problem is thus

$$\max \; \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j d_i d_j \left(\langle x_i, x_j \rangle + \frac{1}{C} \delta_{ij} \right) \tag{2–58a}$$

$$\text{s.t.} \; \sum_{i=1}^{n} d_i \alpha_i = 0 \tag{2–58b}$$

$$\alpha_i \geq 0 \qquad\qquad i = 1, \ldots, n \tag{2–58c}$$

Once the optimal dual variables have been obtained the optimal separation hyperplane can be recovered similar as in hard margin classifier. Once the hyperplane has been obtained a new point x_u can be classified in one of the two classes based on the following rule

$$d_{x_u} = sgn(w^T x_u + b) \tag{2–59}$$

where $sgn(\cdot)$ is the sign function. It is worth noting that the non linear version of SVM can be can be obtained if we just replace the dot product function with another kernel function $\kappa(x_i, x_j)$. For example the dual soft margin kernel SVM formulation will be

$$\max \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j d_i d_j \left(\kappa(x_i, x_j) + \frac{1}{C} \delta_{ij} \right) \tag{2–60a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} d_i \alpha_i = 0 \tag{2–60b}$$

$$\alpha_i \geq 0 \qquad\qquad i = 1, \dots, n \tag{2–60c}$$

In fact the linear case is a special kernel case as the dor product can be seen as an admissible kernel function i.e. $\kappa(\cdot, \cdot) = \langle \cdot, \cdot \rangle$. One of the fundamental limitations of the generic formulation of soft SVM is that it is proposed just for the two class case (binary classification). This might poses a problem as many of the real world problems involve data that belong in more than two classes.

Majority voting scheme [74]: According to this approach, given a total of N classes, we solve the SVM problem for all binary combinations (pairs) of classes. For example for a three class problem (class A, class B and class C) we find the separation hyperplanes that correspond to the problems A vs B, A vs C and B vs C. When a new point comes then each classifier “decides” on the class of this point. Finally the point is classified to the class with the most “votes”.

Directed acyclic graph approach [54]: For the majority voting process one needs to construct a large number of training binary classifiers in order to infer the class of an unknown sample. This can pose a computational problem to the performance. Thus in the directed acyclic graph we try to minimize the number of necessary classifiers required. This can be achieved by considering a tree that eliminates one class at each level. An example with four classes is illustrated in Fig. 2.4.

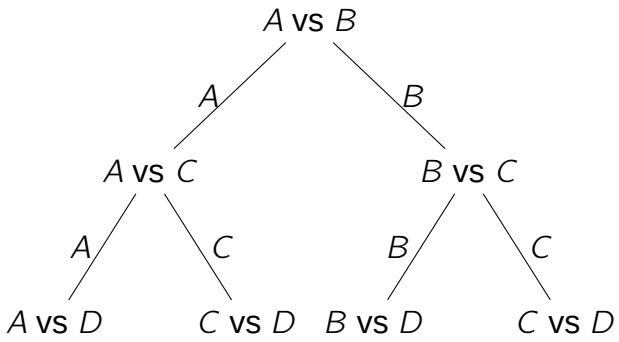


Figure 2-4. Directed acyclic graph approach for a four class example (A,B,C and D). At the first level of the tree the sample goes through the A vs B classifier. Then depending on the outcome the sample is tested through the A vs C or B vs C. The total number of binary classifiers needed equals the depth of the tree.

A straightforward observation regarding these two multiclass generalization strategies is that they can be used for any type of binary classifiers (not only SVM) with or without the use of kernel.

2.4.1 Alternative Objective Function

In a more general framework there have been proposed alternative objective function that can be used for SVM classification yielding in, computationally different problems. In the general form one can express the objective function as $f_p(w)$ where p corresponds to the type of norm. Along these lines we can express the penalty function as $g(C,\xi)$ where C can be either a number meaning that all points are penalized the came way or a diagonal matrix with every diagonal element corresponding to the penalization coefficient of each data sample. A practical application of different penalization coefficient might occur in the case where we have unbalanced classification problem (the training samples of one class is much higher that the other). For the case of SVM that we already presented we assumed quadratic objective and penalty functions

$$f_2(w) = \|w\|_2^2 = w^T w, \quad g(C,\xi) = \xi^T C \xi \tag{2–61}$$

Another popular choice is

$$f(w) = \|w\|_1, \quad g(C, xi) = \langle C, \xi \rangle \tag{2–62}$$

In this case the SVM formulation becomes

$$\min \ \|w\|_1 + \langle C, \xi \rangle \tag{2–63a}$$

$$\text{s.t.} \ \ d_i(w^T x_i + b) \geq 1 - \xi, \quad i = 1, \dots, n \tag{2–63b}$$

It is easy to shown that the last formulation can be solved as a linear program (LP). More specifically if we introduce the auxilary variable α we can obtain the following equivalent formulation of Eq. 2–63

$$\min \ \sum_{i=1}^{n} \alpha_i + \langle C, \xi \rangle \tag{2–64a}$$

$$\text{s.t.} \ \ d_i \left(\sum_{i=1}^{n} \alpha_i \langle x_i, x_j \rangle + b \right) \geq 1 - \xi, \quad i = 1, \dots, n \tag{2–64b}$$

$$\alpha_i \geq 0, \quad \xi \geq 0, \quad i = 1, \dots, n \tag{2–64c}$$

It is worth noting that the linear programming approach was developed independently from the quadratic one.

2.5 Regularized Generalized Eigenvalue Classification

Another supervised learning algorithm that makes use of the notion of hyperplane is the Regularized Generalized Eigenvalue Classifier (ReGEC). Given two classes of points, represented by the rows of matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times n}$, each row being a point in the features space $\mathbb{R}^n$, we wish to determine two hyperplanes, one for each class, with the following characteristics: *a)* The sum of the distances between each point in the class and the hyperplane to be minimum, and *b)* the the sum of the distances between the hyperplane and the points of the other class to be maximum. Originally this concept was developed by Mangassarian et al. [45] and then further developed by Guarracino et al. [29, 30]. First we will give a mathematical description of the

generalized eigenvalue classifiers and then we will illustrate the proposed regularization techniques proposed. If we denote the hyperplane related to class A with $w_A^T x - \gamma_A = 0$, the problem consists in determining the variables $w_A \in \mathbb{R}^n$ and $\gamma_A \in \mathbb{R}$ such that

$$\min_{w_A, \gamma_A \neq 0} \frac{\|Aw_A - e\gamma_A\|^2}{\|Bw_A - e\gamma_A\|^2}, \tag{2–65}$$

where e is a column vector of ones of proper dimension. If we let

$$G = [A \quad -e]^T [A \quad -e], \quad H = [B \quad -e]^T [B \quad -e], \tag{2–66}$$

then the problem of Eq. 2–65 becomes

$$\min_{z_A \neq 0} \frac{z_A^T G z_A}{z_A^T H z_A}, \tag{2–67}$$

with $z_A = \begin{bmatrix} w_A^T & \gamma_A \end{bmatrix}^T$. Note that this last problem has similar structure to Fisher Linear Discriminant Function minimization. It is a minimization of a generalized Rayleigh quotient. This problem can be transformed to the following one

$$\min \ z_A^T G z_A \tag{2–68a}$$

$$\text{s.t.} \ z_A^T H z_A = 1 \tag{2–68b}$$

Then the Lagrangian will be

$$\mathcal{L}_{ReGEC}(z_A, \lambda) = z_A^T G z_A - \lambda \left(z_A^T H z_A - 1 \right) \tag{2–69}$$

If we take the Lagrangian's derivative and set if equal to zero we get

$$\frac{\partial \mathcal{L}_{ReGEC}(z_A, \lambda)}{\partial z_A} = 0 \Leftrightarrow G z_A = \lambda H z_A. \tag{2–70}$$

Since H and G are real and symmetric matrices by construction, if H is positive definite, the optimal solution is attained at the eigenvector $z_A = z_{min}$ that correspond to the minimum eigenvalue λ_{min}. By following a similar process, we can determine the hyperplane for class B. Given the symmetry of the problem, if z_B is the eigenvector

that corresponds to the maximum eigenvalue of the problem of Eq. 2–70, then it is the eigenvector corresponding to the minimum eigenvalue of the following system

$$Hz = \lambda Gz. \tag{2–71}$$

This means we can compute the hyperplanes for A and B computing the eigenvectors related to minimum and maximumum eigenvalues of problem of Eq. 2–70. The model is also able to predict the class label of unknown samples by assigning them to the class with minimum distance from the hyperplane. That is, for an arbitrary unknown point x_u

$$\text{class}(x_u) = \arg \min_{i \in \{A,B\}} = \frac{|w_i^T x_u - \gamma_i|}{\|w_i\|}. \tag{2–72}$$

The generalization to multi-class problems is straightforward. For each class A_i, $i = 1, \ldots, k$, a model is built with respect to every other class $A_j, k \neq i$. The $k - 1$ models for each class are then merged using their normal vectors. The latter are averaged using singular value decomposition, which produces a vector with minimum angle with respect to each normal vector. Such vector identifies a single hyperplane, that is used as classification model for the class A_i. The assignment of a test point to a class is done in two steps. First, the points of each class are projected on their respective plane. Then, the test point is assigned to the class of the nearest neighbor projected point. A detailed description of the method and its performance can be found in [31].

In the original paper on generalized eigenvalue classification by Mangasarian et al [45], since H is not always positive definite, Tikhonov regularization was used for solving the generalized eigenvalue problem. The regularization avoids instabilities and difficulties related to possible singularities of the eigenvalues and multiple eigenvectors produced by a rank deficient H. In this framework, a regularization parameter δ is used, which is adjusted usually through some trial-and-error procedure; then, $G + \delta I$ and $H + \delta I$ are used one at a time, instead of G and H, in the solution of two distinct

eigenvalue problems that produce the two classification models for data points in A and B. The algorithm proves to have better performance compared to the one without regularization. In [30] Guarracino et. al. proposed a alternative regularization framework for this problem. The regularization framework proposed is the following

$$\min_{w_i,\gamma_i\neq 0} \frac{\|Aw_i - e\gamma_i\| + \delta_1^{(i)}\|\tilde{B}w_i - e\gamma_i\|}{\|Bw_i - e\gamma_i\| + \delta_2^{(i)}\|\tilde{A}w_i + e\gamma_i\|},\ i \in \{A, B\} \tag{2–73}$$

where $\delta_1^{(i)}, \delta_2^{(i)}$ are regularization parameters and $\tilde{A}, \tilde{B}$ are diagonal matrices whose diagonal elements are the diagonal elements of matrices A, B correspondingly. The last problem's solution can be obtained from the solution of the following generalized eigenvalue problem

$$(G + \delta_1\tilde{H})z = \lambda(H + \delta_2\tilde{G})z, \tag{2–74}$$

This alternative regularization approach was shown to yield higher classification accuracy compared to Tikhonov regularization [30].

2.6 Case Study - Cell death discrimination

In this section, before procceding to the description of the robust formulaltion of the already discussed data mining methods, we will present a comparative case study among some of the described supervised learning algorithms. This is done mainly in order to give a rough comparison among the methods in a real problem and to show that there is no de facto optimal method for all real data analysis problems. The example is taken from biomedicine and more specifically from cell death discrimination using Raman spectroscopy. This case study has appeared in [78].

The discrimination of cells has widespread use in biomedical and biological applications. Cells can undergo different death types (e.g., apoptotic, necrotic), due to the action of a toxic substance or shock. In the case of cancer cell death, the quantity of cells subject to necrotic death, compared with those going through apoptotic death, is an indicator of the treatment effect. Another application of cell discrimination is cell line characterization, that is to confirm the identity and the purity of a group of cells

that will be used for an experiment. The standard solution is either to use microarray technologies, or to relay on the knowledge of an expert. In the first case, analysis takes a long time, is subject to errors, and requires specialized equipments [56]. On the other hand, when the analysis is based only on observations, results can be highly subjective and difficult to reproduce.

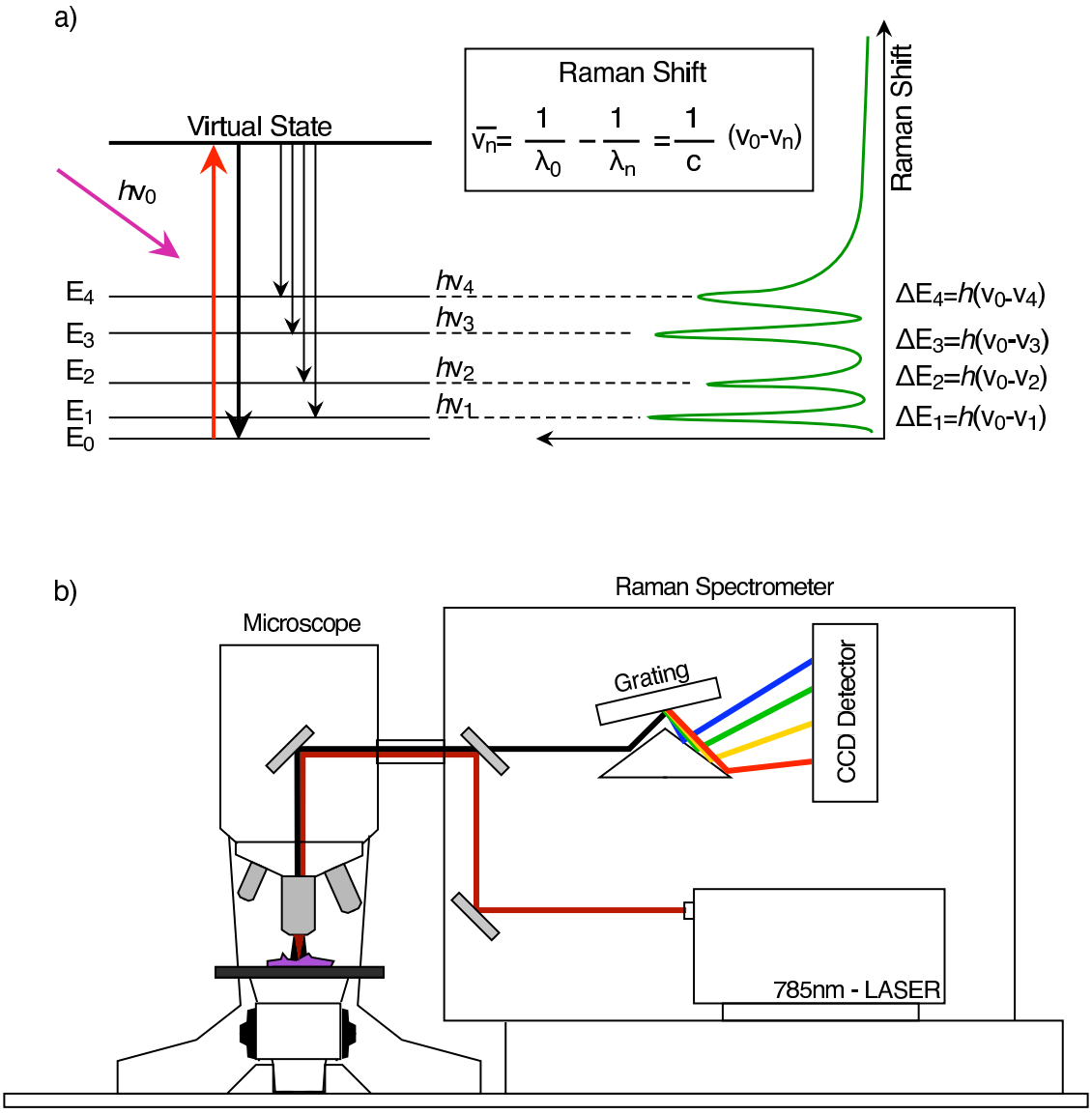


Figure 2-5. Explanation of Raman. Pictorial view of (a) Raman spectrometer's basic principle of operation and (b) instrumentation.

Recently, Raman spectroscopy has been applied to the analysis of cells. This method is based on a diffraction principle, called the Raman shift, that permits to

estimate the quantity and quality of enzymes, proteins and DNA present in a single cell. A microscope focuses the laser through the objective lens on the sample and the scattered photons are collected by the same objective lens and travel to the Raman spectrometer, where they are analyzed by a grating and a CCD detector, as depicted in Fig. 2-5.

Since low energy lasers do not deteriorate or kill cells, it is used in vitro and it can be used in vivo. Furthermore, Raman spectra are not affected by changes in water, which makes the results robust with respect to the natural changes in size and shape of cells. Finally, the spectrometer scan accomplish the experiment in less than a minute, and can even be brought outside a biological laboratory, which can make it potentially useful in many other applications, as in the case of biological threat detection in airports or battlefields.

Raman spectrum is usually analyzed to find peaks at specific wavelengths, which reveals the presence and abundance of a specific cell component. This in turn can be used as a biomarker for cell discrimination or cell death classification [13]. This strategy is highly dependent on the experiment and spectroscope tuning, thus giving rise to questions regarding normalization of spectra and peak detection.

We explore and compare alternative data mining algorithms that can analyze the whole spectrum. These techniques have been successfully applied to other biological and biomedical problems, and and are de facto standard methods for supervised data classification [53, 63]. Methods are tested through numerical experiments on real data and their efficiency with respect to their overall classification accuracy is reported.

2.6.1 Materials

2.6.1.1 Dataset

For evaluating the data mining algorithms, we used two different data sets. The first contains cells from two different cell lines. The 30 cells from the A549 cell line and 60 from MCF7 cell line. The first are breast cancer cells, whereas the later are

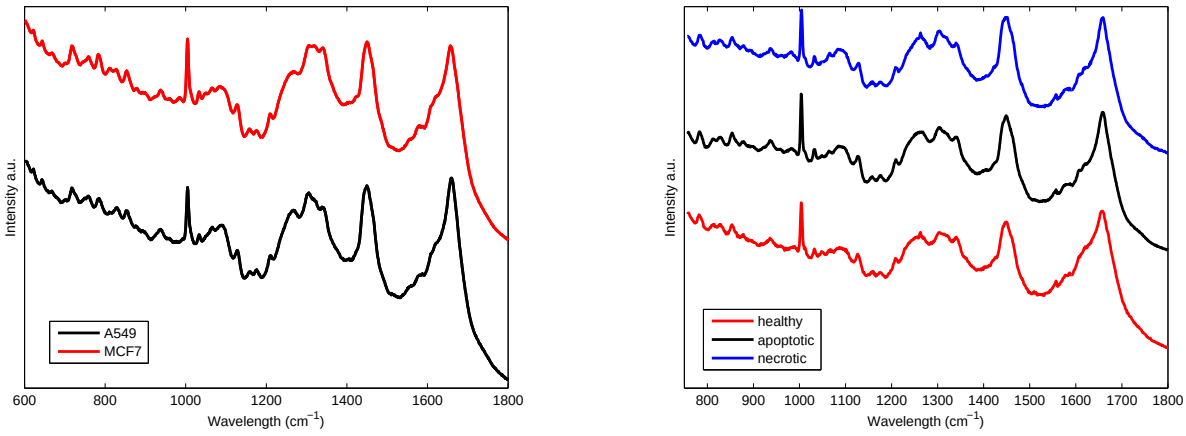


Figure 2-6. The mean Raman spectra for each class. In a) cells from A459 and MCF7 cell line are shown and in b) A549 cells treated with Etoposide (apoptotic), Triton-X (necrotic) and control cells. All spectra have been normalized so that they have zero mean and unitary standard deviation and then they were shifted for clarity.

cancer epithelia cells. All 90 cells of this class were not treated with any substance. The aim of this experiment is to evaluate the ability of various data mining techniques in discriminating between different cell lines.

The second dataset consists uniquely of A549 cancer epithelial cells. The first 28 cells are untreated cancer cells (control), the next 27 cells were treated with Etoposide and the last 28 cells were treated with Triton-X, so that they undergo apoptotic and necrotic death correspondingly. The detailed protocols followed for the biological experiments were standard and can be found at [57]. The mean spectrum of each class for the two datasets are shown in Fig. 2-6 (a & b).

2.6.1.2 Raman spectroscope

The Raman microscope is an InVia system by Renishaw. It consists of a Leica microscope connected to a Renishaw 2000 spectrometer. The high power diode laser (250 mW) produces laser light of 785 nm. Both data sets were acquired by Particle engineering Research Center (P.E.R.C.) at the University of Florida.

2.6.1.3 Data preprocessing

For peak analysis Raman spectra can be preprocessed in many ways. Once they have been acquired by the instrument, the first step consists in subtracting the background noise. This is usually done subtracting to each spectrum the value of a spectrum obtained without the biological sample. Then spectra are normalized subtracting a mean spectrum obtained with a polynomial approximation of fixed order. Other techniques are used to detect peaks and to delete spikes. In the present case study, we only normalized the data along the features of the training set, to obtain features with zero mean and unit variance. Those values of mean and variance are then used to normalize the test spectra.

2.6.2 Software

In this case study we are comparing LDA, SVM, ReGEC, that we have already explore their optimization structure, as well as k nearest neighbor algorithm and Improved Iterative scaling (further discussed in following subsections). For the computational experiments Matlab arsenal toolbox was used for LDA, IIS, k-NN [82], whereas for SVM, libsvm was employed [22]. For ReGEC classification, the author's implementation was used [29, 30].

2.6.3 Nearest Neighbor Classification

This is one of the simplest and most popular supervised learning algorithm. For every sample in the testing set we compute the distances (defined by some distance metric) between the unknown sample and all the known samples. For this study we used Euclidean distance. Then we keep the closest k known samples and the class of the unknown one is decided with a majority voting scheme. Usually, especially for binary classification problem k is chosen to be an odd number (most common 1,3,5,7). By definition k-NN is non linear and multi class algorithm. It is very popular as it is easy to implement and does not make use of advanced mathematical techniques. Also it is one

of the simplest techniques for non linear classification without use of kernel embedding.

On the other side it has been criticized major because

1. Testing process always require computation of all distances between the unknown sample and all the samples that belong to the training set. This dependence on the training set size might substantially slow down the testing process.

2. This algorithm is extremely sensitive to outliers. A small number of outlier (of order k) can cause can manipulate the classification accuracy as the algorithm “cares” only about the k closest points to the unknown samples.

In order to overcome the aforementioned drawbacks several preprocessing techniques can be employed.

2.6.4 Improved Iterative Scaling

Given a random process which produces, at each time step, some output value y which is a member of the set of possible outputs, IIS [23] computes the probability of the event y influenced by a conditioning information x. In this way we can consider, for example, in a text sequence, the probability $p(y|x)$ of the event that given a word x, the next word will be y. This leads to the following exponential model

$$p_\Lambda(y|x) = \frac{1}{Z_\Lambda(x)} \exp(\sum_{i=1}^{m} \lambda_i f_i(x,y)), \tag{2–75}$$

where $f_i(x,y)$ is a binary valued function called *feature function*, $\lambda_i \in \mathbf{R}$ is the Lagrange multiplier corresponding to f_i and $|\lambda_i|$ is a measure of the importance of the feature f_i, $Z_{\Lambda(x)}$ is a normalizing factor and finally we put $\Lambda = \{\lambda_1, \dots, \lambda_m\}$.

Given a joint empirical distribution $\bar{p}(x,y)$, the log-likelihood of $\bar{p}$ according to a conditional model $p_\Lambda(y|x)$, is defined as

$$L_{\bar{(p)}}(\Lambda) = \sum_{x,y} \bar{p}(x,y) \log p_\Lambda(y|x). \tag{2–76}$$

This can be regarded as a measure of the quality of the model p_Λ. Clearly we have that $L_{\bar{(p)}}(\Lambda) \leq 0$ and $L_{\bar{(p)}}(\Lambda) = 0$ if and only if p_Λ is *perfect* with respect to $\bar{p}$, i.e. $p_\Lambda(y|x) = 1 \Leftrightarrow \bar{p}(x,y) > 0$.

Given the set $\{f_1, \ldots, f_m\}$, the exponential form of Eq. 2–75 and the distribution $\bar{p}$, IIS solves the maximum likelihood problem computing

$$\Lambda^* = \arg\max_{\Lambda} L_{\bar{p}}(\Lambda) \in \mathbf{R}^m.$$

2.6.5 Results and Discussion

2.6.5.1 Classification and model selection

We applied the following supervised learning algorithms on both datasets: a) soft margin SVM b) Regularized generalized eigenvalue classification and c) k nearest neighbor classification (k-NN with $k = 3$), d) Linear Discriminant Analysis and e) Improved Iterative Scaling (IIS) classification. No kernel was applied in the classifiers. In particular, for soft margin SVM classifier the parameter C was chosen to be 10 for the first dataset and 100 for the second. For ReGEC the regularization parameter δ was chosen 0.01. The tuning was done through a grid search on the on the parameter space. At every repetition 90% of the samples were used for training and 10% for testing. The average cross validation accuracies are reported on Table 2-1.

Table 2-1. Average classification accuracy for hold out cross validation (100 repetitions). The LDA achieves the highest accuracy for the binary problem and ReGEC for the three class.

	Classification accuracy (%)	
	Cell line discrimination (two class)	Cell death discrimination (three class)
C-SVM	95.33	97.33
ReGEC	96.66	98.44
3-NNR	79.22	95.44
IIS	95.67	87.44
LDA	100	91.00

We can see that for both datasets only C-SVM and ReGEC achieve classification higher that 95%. Nearest neighbor classification although it performs very well for the three class problem it has poor results in the two class. This is related to the generic drawback of this method which makes it very sensitive to outliers. Linear Discriminant

analysis also achieves high classification results ($> 90\%$ in both cases) justifying its use in the literature [50, 52].

2.6.5.2 Impact of training set dimension

Next we examined the robustness of each classifier with respect to the size of the training dataset. For this we fixed the training size dataset and we repeated the cross validation process for different sizes of the training dataset. The results were evaluated through hold out cross validation (100 repetitions). Results are shown in Fig 2-7 & Fig. 2-8. We notice that ReGEC is considerably robust to the size of the training dataset maintaining classification accuracy higher that 95% for all the cases. Overall algorithms demonstrated a smooth performance meaning that the change os the classification accuracy was proportional to the change of the classification dataset size.

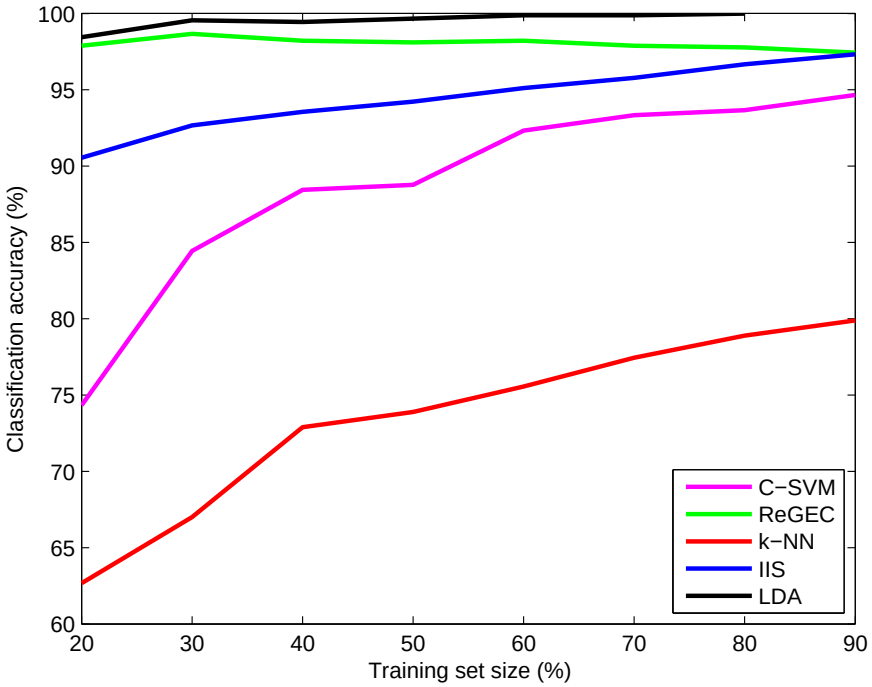


Figure 2-7. Classification accuracy versus size of the training set (binary. Accuracy is evaluated for 100 cross validation runs.

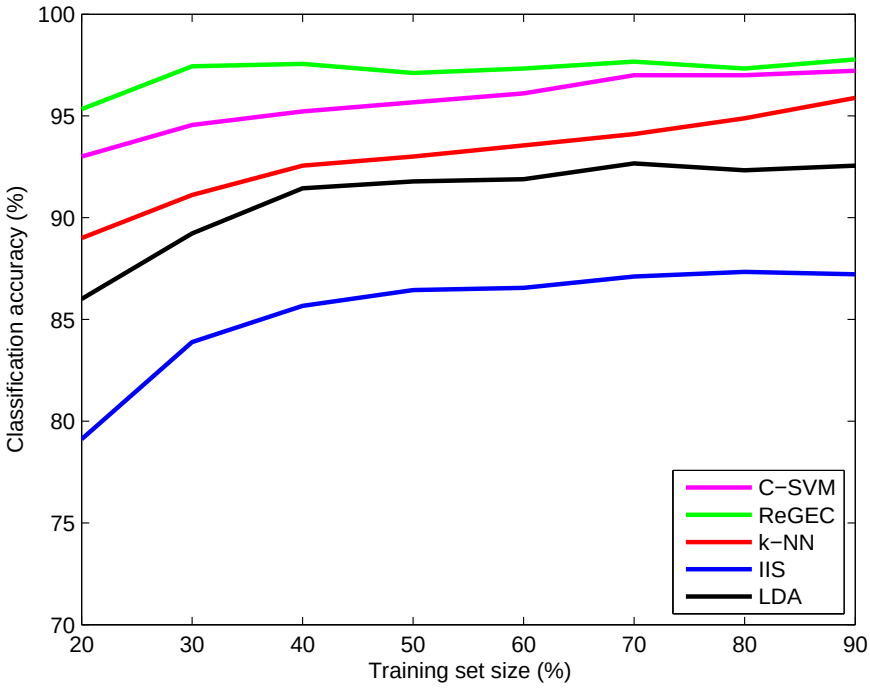


Figure 2-8. Classification accuracy versus size of the training set (multiclass). Accuracy is evaluated for 100 cross validation runs.

CHAPTER 3
ROBUST LEAST SQUARES

3.1 The Original Problem

In this chapter we will study the robust version of problem of Eq. 2–1. The results presented in this chapter were first described in [25] and similar results were independently obtained in [21]. At the end of the chapter we describe some extensions that were first described in [20]. As we discussed in earlier the RC formulation of a problem involves solution of a worst case scenario problem. This is expressed by a min-max (or max-min) type problem where the outer min (max) problem refers to the original one whereas the inner max (min) to the worst admissible scenario one. For the least squares case the generic RC formulation can be described from the following problem

$$\min_{x} \max_{\Delta A \in \mathcal{U}_A, \Delta b \in \mathcal{U}_b} \|(A + \Delta A)x + (b + \Delta b)\|_2 \tag{3–1}$$

where $\Delta A, \Delta B$ are perturbation matrices and $\mathcal{U}_A.\mathcal{U}_B$ are sets of admissible perturbations. As in many robust optimization problems the structural properties of $\mathcal{U}_A, \mathcal{U}_B$ are important for the computational tracktability of the problem. Here we will study the case where the two perturbation matrices are unknown but their norm is bounded by a known constant. Thus we have the following optimization problem

$$\min_{x} \max_{\|\Delta A\| \leq \rho_A, \|\Delta b\| \leq \rho_b} \|(A + \Delta A)x + (b + \Delta b)\|_2 \tag{3–2}$$

This type of uncertainty is often called coupled uncertainty because the uncertainty information is not given in terms of each sample individually but in terms of the whole data matrix. First we will reduce problem of Eq. 3–2 to a minimization problem through the following lemma

Lemma 1. *The problem of Eq. 3–2 is equivalent to the following problem*

$$\min_{x} (\|Ax + b\| + \rho_A \|x\| + \rho_b) \tag{3–3}$$

Proof. From triangular inequality we can obtain an upper bound on the objective function of Eq. 3–2

$$\begin{aligned} \|(A+\Delta A)x+(b+\Delta b)\| &\leq \|Ax+b\|+\|\Delta Ax+\Delta b\| && (3\text{–}4)\\ &\leq \|Ax+b\|+\|\Delta A\|\|x\|+\|\Delta b\| && (3\text{–}5)\\ &\leq \|Ax+b\|+\rho_A\|x\|+\rho_B && (3\text{–}6) \end{aligned}$$

Now if in the original problem of Eq. 3–2 we set

$$\Delta A = \frac{Ax-b}{\|Ax-b\|}\frac{x^T}{\|x\|}\rho_A, \quad \Delta b = -\frac{Ax-b}{\|Ax-b\|}\rho_B \tag{3–7}$$

we get

$$\begin{aligned} \|(A+\Delta A)x-(b+\Delta b)\| &= \|Ax+b-\Delta Ax+\Delta b\| && (3\text{–}8)\\ &= \|Ax-b\|\left(1+\frac{\|x\|}{\|Ax-b\|}\rho_A+\frac{1}{\|Ax-b\|}\rho_B\right) && (3\text{–}9)\\ &= \|Ax-b\|+\rho_A\|x\|+\rho_B && (3\text{–}10) \end{aligned}$$

This means that the upper bound obtained by the triangular inequality can be achieved by Eq. 3–7. Since the problem is convex this will be the global optimum for the problem. □

We can easily observe that the point of Eq. 3–7 satisfy the optimality conditions. Since problem of Eq. 3–3 is unconstraint its Lagrangian will be the same as the cost function. Since this function is convex we just need to examine the points for which the derivative is equal to zero and take separate case for the non differentiable points. At the points where the cost function is differentiable we have

$$\frac{\partial \mathcal{L}_{RLLS}(x)}{\partial x} = 0 \Leftrightarrow \frac{A^T(Ax-b)}{\|Ax-b\|}+\frac{x}{\|x\|}\rho_A = 0 \tag{3–11}$$

From this last expression we require $x \neq 0$ and $Ax \neq b$ (we will deal with this cases lately). If we solve with respect to x we obtain

$$\frac{1}{\|Ax-b\|}\left(A^T(Ax-b)+x\frac{\|Ax-b\|}{\|x\|}\rho_A\right)=0 \tag{3–12}$$

or

$$\left(A^TA+\rho_A\frac{\|Ax-b\|}{\|x\|}I\right)x=A^Tb \tag{3–13}$$

and finally

$$x=(A^TA+\mu I)^{-1}A^Tb, \quad \text{where} \quad \mu=\frac{\|Ax-b\|}{\|x\|}\rho_A \tag{3–14}$$

In case that $Ax = b$ then the solution is given by $x = A^\dagger b$ where $A^\dagger$ is the Moore Penrose or pseudoinverse matrix of A. Therefore we can summarize this result in the following lemma

Lemma 2. *The optimal solution to problem of Eq. 3–3 is given by*

$$x=\begin{cases} A^\dagger b & \text{if} \quad Ax=b \\ (A^TA+\mu I)^{-1}A^Tb, \quad \mu=\rho_A\frac{\|Ax-b\|}{\|x\|} & \text{otherwise} \end{cases} \tag{3–15}$$

Since in this last expression μ is a function of x we need to provide with a way in order to tune it. For this we need to use the singular value decomposition of data matrix A

$$A=U\begin{bmatrix}\Sigma\\0\end{bmatrix}V^T \tag{3–16}$$

where Σ is the diagonal matrix that contains the singular values of A in descending order. In addition we partition the vector U^Tb as follows

$$\begin{bmatrix}b_1\\b_2\end{bmatrix}=U^Tb \tag{3–17}$$

where b_1 contains the first n elements and b_2 the rest $m-n$. Now using this decompositions we will obtain two expressions for the enumerator and the denominator of μ. First for the

denominator

$$x = (A^T A - \mu I)^{-1} A^T b = \left(V \Sigma^2 V^T - \mu I\right)^{-1} V \Sigma b_1 = V \left(\Sigma^2 + \mu I\right)^{-1} \Sigma b_1 \tag{3–18}$$

thus the norm will be given from

$$\|x\| = \|\Sigma(\Sigma^2 + \mu I)^{-1}\| \tag{3–19}$$

and for the enumerator

$$Ax - b = \quad U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T V \left(\Sigma^2 + \mu I\right)^{-1} \Sigma b_1 - b \tag{3–20}$$

$$= \quad U \left(\begin{bmatrix} \Sigma \\ 0 \end{bmatrix} \left(\Sigma^2 + \mu I\right)^{-1} \Sigma b_1 - U^T b \right) \tag{3–21}$$

$$= \quad U \left(\begin{bmatrix} \Sigma(\Sigma^2 + \mu I)^{-1} \Sigma b_1 - b_1 \\ -b_2 \end{bmatrix} \right) \tag{3–22}$$

$$= \quad U \begin{bmatrix} -\mu(\Sigma^2 + \mu I)^{-1} b_1 \\ -b_2 \end{bmatrix} \tag{3–23}$$

and for the norm

$$\|Ax - b\| = \sqrt{\|b_2\|^2 + \alpha^2 \|(\Sigma^2 + \mu I)^{-1} b_1\|^2} \tag{3–24}$$

Thus μ will be given by

$$\mu = \frac{\|Ax - b\|}{\|x\|} = \rho_A \frac{\sqrt{\|b_2\|^2 + \alpha^2 \|(\Sigma^2 + \mu I)^{-1} b_1\|^2}}{\|\Sigma(\Sigma^2 + \mu I)^{-1} b_1\|} \tag{3–25}$$

Note that the present in the present analysis we assume that data matrix A is of full rank. If this is not the case similar analysis can be performed (for details see [20]). The

closed form solution can be obtained by the solution of Eq. 3–25. Next we will present some variations of the original least squares problem that are discussed in [20].

3.2 Variations of the Original Problem

In [20] authors introduced least square formulation for slightly different perturbation scenarios. For example in the case of the weighted least squares problem with weight uncertainty one is interested to find

$$\min_{x} \max_{\|\Delta W\| \leq \rho_W} \| \left((W + \Delta W)(Ax - b) \right) \| \tag{3–26}$$

using the triangular inequality we can obtain an upper bound

$$\begin{aligned} \|(W + \Delta W)(Ax - b)\| \leq & \quad \|W(Ax - b)\| + \|\Delta W(Ax - b)\| & (3–27) \\ \leq & \quad \|W(Ax - b)\| + \rho_W \|Ax - b\| & (3–28) \end{aligned}$$

Thus the inner maximization problem reduces to the following problem

$$\min_{x} \left(\|W(Ax - b)\| + \rho_W \|Ax - b\| \right) \tag{3–29}$$

by taking the corresponding KKT conditions, similar to previous analysis, we obtain

$$\begin{aligned} \frac{\partial \mathcal{L}_{WLLS}(x)}{\partial x} &= \frac{\partial \|W(Ax - b)\|}{\partial x} + \frac{\partial \|Ax - b\|}{\partial x} & (3–30) \\ &= \frac{A^T W^T (WAx - Wb)}{\|W(Ax - b)\|} + \rho_W \frac{A^T (Ax - b)}{\|Ax - b\|} & (3–31) \end{aligned}$$

By solving the equation

$$\frac{\partial \mathcal{L}_{WLLS}(x)}{\partial x} = 0 \tag{3–32}$$

we find that the solution should satisfy

$$A^T (W^T W + \mu I) Ax = A^T (W^T W + \mu I) b \ \text{ where } \ \mu_w = \frac{\|W(Ax - b)\|}{\|Ax - b\|} \tag{3–33}$$

Giving the expression for x

$$x = \begin{cases} A^{\dagger} b & \text{if } \; Ax = b \\ (WA)^{\dagger} Wb & \text{if } \; WAx = Wb \\ \left(A^T (W^T W + \mu_w I)\, A\right)^{-1} A^T \left(W^T W + \mu_w I\right) b & \text{otherwise} \end{cases} \tag{3–34}$$

where μ_w is defined in Eq. 3–33. The solution for the last one can be obtained through similar way as for the original least squares problem. In another variation of the problem the uncertainty can be given with respect to matrix A but in multiplicative form. Thus the robust optimization problem for this variation can be stated as follows

$$\min_x \max_{\|\Delta A\| \leq \rho_A} \|(I + \Delta A)Ax - b\| \tag{3–35}$$

which can be reduced to the following minimization problem

$$\min_x \left(\|Ax - b\| + \rho_A \|Ax\|\right) \tag{3–36}$$

then by similar analysis we obtain

$$\frac{\partial \mathcal{L}_{MLLS}(x)}{\partial x} \;=\; \frac{A^T(Ax - b)}{\|A^T(Ax - b)\|} + \rho_A \frac{A^T Ax}{\|Ax\|} = 0 \tag{3–37}$$

and finally

$$x = \begin{cases} (A^T A)^{\dagger} b & \text{if } \; A^T Ax = A^T b \\ \left(A^T A(1 + \mu_A)\right)^{-1} A^T b, \mu_A = \frac{\|A^T(Ax-b)\|}{\|Ax\|} & \text{otherwise} \end{cases} \tag{3–38}$$

CHAPTER 4
ROBUST PRINCIPAL COMPONENT ANALYSIS

In this chapter we will describe a robust optimization approach for principal component analysis (PCA) transformation. Again we need to clarify that the purpose of the present work is to investigate the application of robust optimization in the PCA transformation. There have been several Robust PCA papers in the literature that deal with the application of robust statistics in PCA [35] and they are of interest when outliers are present in the data. Unlike supervised learning approaches like SVM, that the purpose is to find the optimal solution for the worst case scenario, the purpose of robust formulation of PCA, as described in [6], is to provide components that explaining data variance while at the same time are as sparse as possible. This is in general called sparse component analysis (SPCA) transformation. By sparse solutions we mean the vectors with large number of zeros. In general sparsity can be enforced through different methods. Sparsity is the desired property, especially in telecommunications, because it allows more efficient compression and faster data transmission. Aa sparse component analysis formulation can be obtained if we add a cardinality constraint that strictly enforces sparsity. That is

$$\max \quad u^T S u \tag{4–1a}$$

$$\text{s.t} \quad u^T u = 1 \tag{4–1b}$$

$$\text{card(x)} \leq k \tag{4–1c}$$

where $\text{card}(\cdot)$ is the cardinality function and k is parameter defining the maximum allowed component cardinality. Matrix S is the covariance matrix defined in Chapter 4 and u is the decision variable vector. Alternatively this problem can be casted as a

semidefinite programming problem as follows

$$\max \ \mathrm{Tr}(US) \tag{4–2a}$$

$$\text{s.t} \ \ \mathrm{Tr}(U) = 1 \tag{4–2b}$$

$$\mathrm{card(U)} \leq k^2, \tag{4–2c}$$

$$U \succeq 0, \mathrm{Rank}(X) = 1 \tag{4–2d}$$

where U is the decision variable matrix and $\succeq$ denotes that the matrix is posistive semidefinite (i.e. $a^T X a \geq 0, \ \forall a \in \mathbb{R}^n$). Indeed the solution to the original problem can be obtained from the second one since conditions of Eq. 4–2d guarrantee that $U = u \cdot u^T$. Instead of strictly constrain the cardinality we will demand $e^T \cdot \mathrm{abs}(U) \cdot e \leq k$ (where e is the vector of 1's and $\mathrm{abs}(\cdot)$ returns the matrix whose elements are the absolute values of the original matrix). In addition we will drop the rank constraint as this is also a tough to handle constraint. Thus obtain the following relaxation of the original problem

$$\max \ \mathrm{Tr}(US) \tag{4–3a}$$

$$\text{s.t} \ \ \mathrm{Tr}(U) = 1 \tag{4–3b}$$

$$e^T \cdot abs(U) \cdot e \leq k \tag{4–3c}$$

$$U \succeq 0 \tag{4–3d}$$

the last relaxed problem is a semidefinite program with respect to matrix variable U. We can rewrite it as follows

$$\max \ \mathrm{Tr}(US) \tag{4–4a}$$

$$\text{s.t} \ \ \mathrm{Tr}(U) = 1 \tag{4–4b}$$

$$e^T \cdot abs(U) \cdot e \leq k \tag{4–4c}$$

$$U \succeq 0 \tag{4–4d}$$

If we now remove the constraint of Eq. 4–4c and instead get a penalization coefficient we obtain the following relaxation

$$\max \ \mathrm{Tr}(US) - \rho e^T \cdot abs(U) \cdot e \tag{4–5a}$$

$$\mathrm{s.t} \ \mathrm{Tr}(U) = 1 \tag{4–5b}$$

$$U \succeq 0 \tag{4–5c}$$

where ρ is problems parameter (Lagrange multiplier) determining the penalty's magnitude. By taking the dual of this problem we can have a better understanding for the nature of the problem.

$$\min \ \lambda^{\max}(S+V) \tag{4–6a}$$

$$\mathrm{s.t.} \ |V_{ij}| \leq \rho, \quad i,j = 1, \cdots, n \tag{4–6b}$$

Where $\lambda^{\max}(X)$ is the maximum eigenvalue of matrix X. The problem of Eq. 4–5 can be rewritten as the following $\min - \max$ problem

$$\max_{X \succeq 0, \mathrm{Tr}(U)=1} \min_{|V_{ij}| \leq \rho} \ \mathrm{Tr}(U(S+V)) \tag{4–7a}$$

More precisely the goal is to determine the component that correspond to the maximum possible variance (which is the original PCA objective) by chosing the most sparse solution (according to the sparsity constraints).

CHAPTER 5
ROBUST LINEAR DISCRIMINANT ANALYSIS

The RC formulation of robust linear discriminant analysis (LDA) was proposed by Kim et al [39, 40]. As in other approaches the motivation for a robust counterpart formulation of LDA comes from the fact that data might be imprecise thus the means and the standard deviations computed might not be trustworthy estimates of their real values. The approach that we will present here considers the uncertainty on the mean and standard deviation rather on the data points themselves. Given the original Fisher's optimization problem (described in Chapter 2)

$$\max_{\phi \neq 0} \mathcal{J}(\phi) = \max_{\phi \neq 0} \frac{\phi^T S_{AB} \phi}{\phi^T S \phi} = \max_{\phi \neq 0} \frac{\phi^T (\bar{x}_A - \bar{x}_B)(\bar{x}_A - \bar{x}_B)^T \phi}{\phi^T (S_A + S_B) \phi} \tag{5–1}$$

where $\bar{x}_A, \bar{x}_B$ are the means of each class and S_A, S_B are the class covariance matrices. The solution to this problem, that can be computed analytically is

$$\phi^{\text{opt}} = (S_A + S_B)^{-1} (\bar{x}_A - \bar{x}_B) \tag{5–2}$$

Which corresponds to the optimal Fisher criterion value

$$\mathcal{J}(\phi^{\text{opt}}) = (\bar{x}_A - \bar{x}_B)^T (S_A + S_B)^{-1} (\bar{x}_A - \bar{x}_B)^T \tag{5–3}$$

Computationally ϕ^{opt} and $\mathcal{J}(\phi^{\text{opt}}$ can be found from the eigenvectors and the eigenvalues of the generalized eigenvalue problem

$$S_{AB} \phi = \lambda S \phi \tag{5–4}$$

For the robust case we are interested to determine the optimal value of Fisher's criterion for some undesired, worst case scenario. In terms of optimization this can be described by the following $\min - \max$ problem.

$$\max_{\phi \neq 0} \min_{\bar{x}_A, \bar{x}_B, S_A, S_B} \frac{\phi^T (\bar{x}_A - \bar{x}_B)(\bar{x}_A - \bar{x}_B)^T \phi}{\phi^T (S_A + S_B) \phi} = \max_{\phi \neq 0} \min_{\bar{x}_A, \bar{x}_B, S_A, S_B} \frac{(\phi^T (\bar{x}_A - \bar{x}_B))^2}{\phi^T (S_A + S_B) \phi} \tag{5–5}$$

In other words we need to estimate the optimal vector ϕ, defining the Fisher's hyperplane, given that a worst case scenario, with respect to means and variances, occurs. This problems solution strongly depends on the nature of the worst case admissible perturbation set. In general we denote the set of all admissible perturbation $\mathcal{U} \subseteq \mathbb{R}^n \times \mathbb{R}^n \times S_{++}^n \times S_{++}^n$ where here with S_{++}^n we denote the set of all positive semidefinite matrices. Then the only constraint of the inner minimization problem would be $(\bar{x}_A, \bar{x}_B, S_A, S_B) \in \mathcal{U}$. In case that we are able to exchange the order or the minimization and the maximization problem without affecting the problem's structure. In such a case we could write

$$\max_{\phi \neq 0} \min_{(\bar{x}_A, \bar{x}_B, S_A, S_B) \in \mathcal{U}} \frac{(\phi^T(\bar{x_A} - \bar{x}_B))^2}{\phi^T(S_A + S_B)\phi} = \min_{(\bar{x}_A \bar{x}_B, S_A, S_B) \in \mathcal{U}} (\bar{x}_A - \bar{x}_B)(S_A + S_B)^{-1}(\bar{x}_A - \bar{x}_B)^T \quad (5\text{–}6)$$

For a general $\min - \max$ problem we can write

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y) = \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} f(x, y) \quad (5\text{–}7)$$

if $f(x, y)$ is convex function with respect to both x, concave with respect to y and also $\mathcal{X}, \mathcal{Y}$ are convex sets. This results is known as strong $\min - \max$ property [68]. When convexity does not hold we have the so called weak $\min - \max$ property

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y) \geq \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} f(x, y) \quad (5\text{–}8)$$

Thus in [39, 40] Kim et al. provide with such a minimax theorem for the problem under consideration that does not require the strict assumptions of Sion's result. This results is stated in the following theorem

Theorem 5.1. *For the following minimization problem*

$$\min \frac{(w^T a)^2}{w^T B w} \quad (5\text{–}9)$$

let $(a^{\text{opt}}, B^{\text{opt}})$ *be the optimal solution. Also let* $w^{opt} = (B^{\text{opt}})^{-1} \cdot a^{\text{opt}}$*. Then the point* $(w^{\text{opt}}, a^{\text{opt}}, B^{\text{opt}})$ *satisfies the following minimax property*

$$\frac{((w^{\text{opt}})^T a^{\text{opt}})^2}{(w^{\text{opt}})^T B^{\text{opt}} w^{\text{opt}}} = \min_{(a,B)} \max_{w} \frac{(w^T a)^2}{w^T B w} = \max_{w} \min_{(a,B)} \frac{(w^T a)^2}{w^T B w} \tag{5–10}$$

Proof. See [40] □

Here it is worth noting that this result has a varierty of application including signal processing and portfolio optimization (for details see [39]). Thus the solution for the robust problem can be obtained by solving the following problem

$$\min \quad (\bar{x}_A - \bar{x}_B)(S_A + S_B)^{-1}(\bar{x}_A - \bar{x}_B)^T \tag{5–11a}$$

$$\text{s.t} \quad (\bar{x}_A \bar{x}_B, S_A, S_B) \in \mathcal{U} \tag{5–11b}$$

Assuming that $\mathcal{U}$ is convex problem of Eq. 5–11 is a convex problem. This holds because the objective function is convex as a matrix fractional function (for detailed proof see [17]). Next we will examine the robust linear discriminant solution for a special case of uncertainty sets. More specifically let us assume that the frobenius norm of the differences between the real and the estimated value of the covariance matrices is bounded by a constant. That is

$$\mathcal{U}_S = \mathcal{U}_A \times \mathcal{U}_B \tag{5–12}$$

$$\mathcal{U}_A = \{S_A | \|S_A - \bar{S}_A\|_F \leq \delta_A\} \tag{5–13}$$

$$\mathcal{U}_B = \{S_B | \|S_B - \bar{S}_B\| \leq \delta_B\} \tag{5–14}$$

In general the worst case minimization problem can be expressed

$$\min_{(\bar{x}_A, \bar{x}_B, S_A, S_B)} = \min_{(\bar{x}_A, \bar{x}_B) \in \mathcal{U}_x} \frac{(\bar{x}_A - \bar{x_B})}{\max_{(S_A, S_B) \in \mathcal{U}_S} \phi^T (S_A + S_B)\phi} \tag{5–15}$$

The problem in the denominator can be further simplified

$$\max_{(S_A, S_B) \in \mathcal{U}_S} \phi^T (S_A + S_B)\phi = \phi^T (\bar{S}_A + \bar{S}_B + \delta_A I + \delta_B I)\phi \tag{5–16}$$

Thus the robust solution will be given by the solution to the convex optimization problem

$$\min_{(\bar{x}_A, \bar{x}_B)} (\bar{x}_A - \bar{x}_B)^T (\bar{S}_A + \bar{S}_B + \delta_A I + \delta_B I)^{-1} (\bar{x}_A - \bar{x}_B) \tag{5–17}$$

Which is simpler than Eq. 5–11.

CHAPTER 6
ROBUST SUPPORT VECTOR MACHINES

One of the most well studied application of robust optimization in data mining is the one of support vector machines (SVMs). The theoretical and practical issues have been extensivaly explored through the works of Trafalis et al. [71, 72], Nemirovski et al. [9] and Xu et al. [81]. It is of particular interest that robust SVM formulations are tracktable for a varierty of perturbation sets. At the same time there is clear theoretical connection between particular robustification and regularization [81]. On the other side several robust optimization formulation can be solved as conic problems. If we recall the primal soft margin SVM formulation presented in Chapter 2

$$\min_{w,b,\xi_i} \ \frac{1}{2}\left(\|w\|^2 + C\sum_{i=1}^{n}\xi_i^2\right) \tag{6–1a}$$

$$\text{s.t.} \ \ d_i\left(w^T x_i + b\right) \geq 1 - \xi_i, \quad i = 1, \dots, n \tag{6–1b}$$

$$\xi_i \geq 0, \quad i = 1, \dots, n \tag{6–1c}$$

For the robust case we replace each point x_i with $\tilde{x}_i = \bar{x}_i + \sigma_i$ where $\bar{x}_i$ are the nominal (known) values and σ_i is an additive unknown perturbation that belongs to a well defined uncertainty set. The objective is to reoptimize the problem and obtain the best solution that correspond to the worst case perturbation. Thus the general robust optimization problem formulation can be stated as follows

$$\min_{w,b,\xi_i} \ \frac{1}{2}\left(\|w\|^2 + C\sum_{i=1}^{n}\xi_i^2\right) \tag{6–2a}$$

$$\text{s.t.} \ \ \min_{\sigma_i}\left(d_i\left(w^T(\bar{x}_i + \sigma_i) + b\right)\right) \geq 1 - \xi_i, \quad i = 1, \dots, n \tag{6–2b}$$

$$\xi \geq 0, i = 1, \dots, n \tag{6–2c}$$

$$\sigma_i \in \mathcal{U}_{\sigma_i} \tag{6–2d}$$

Note that since the expression of constraint of Eq. 6–2b corresponds to the distance of the ith point to the separation hyperplane the worst case σ_i would be the one that

minimizes this distance. An equivalent form of constraint of Eq. 6–2b is

$$d_i\left(w^T\bar{x}_i + b\right) + \min_{\sigma_i} d_i\left(w^T\sigma_i\right) \geq 1 - \xi_i, \quad i = 1, \dots, n \tag{6–3}$$

Thus part of finding the solution to the robust SVM formulation passes through the solution of the following problem

$$\min_{\sigma_i \in \mathcal{U}_{\sigma_i}} d_i\left(w^T\sigma_i\right), \quad i = 1, \dots, n \tag{6–4}$$

for fixed w, where $\mathcal{U}_{\sigma_i}$ is the sets of admissible perturbations corresponding to i^{th} sample. Suppose that the l_p norm of the unknown perturbations are bounded by known constant.

$$\min \quad d_i\left(w^T\sigma_i\right), \quad i = 1, \dots, n \tag{6–5}$$

$$\text{s.t.} \quad \|\sigma_i\|_p \leq \rho_i \tag{6–6}$$

By using Hölders inequality (see appendix) we can obtain

$$|d_i(w^T\sigma_i)| \leq \|w\|_q\|\sigma_i\|_p \leq \rho_i\|w\|_q \tag{6–7}$$

where $\|\cdot\|_q$ is the dual norm of $\|\cdot\|_p$. Equivalently we can obtain

$$-\rho_i\|w\|_q \geq d_i(w^T\sigma_i) \tag{6–8}$$

Thus the minimum of this expression will be $-\rho_i\|w\|_q$. If we substitute this expression in the original problem we obtain

$$\min_{w,b,\xi_i} \frac{1}{2}\left(\|w\|^2 + C\sum_{i=1}^{n}\xi_i^2\right) \tag{6–9a}$$

$$\text{s.t.} \quad d_i\left(w^T(\bar{x}_i + \sigma_i) + b\right) - \rho_i\|w\|_q \geq 1 - \xi_i, \quad i = 1, \dots, n \tag{6–9b}$$

$$\xi_i \geq 0, \quad i = 1, \dots, n \tag{6–9c}$$

The structure of the obtained optimization problem depends on the norm p. Next we will present some "interesting" case. It is easy to determine the value of q from $1/p + 1/q = 1$ (for details see appendix). For $p = q = 2$ we obtain the following formulation

$$\min_{w,b,\xi_i} \ \frac{1}{2}\left(\|w\|^2 + C\sum_{i=1}^{n}\xi_i^2\right) \tag{6–10a}$$

$$\text{s.t.} \ \ d_i\left(w^T\bar{x}_i + b\right) - \rho_i\|w\|_2 \geq 1 - \xi_i, \ \ i = 1, \dots, n \tag{6–10b}$$

$$\xi_i \geq 0, \ \ i = 1, \dots, n \tag{6–10c}$$

The last formulation can we seen as a regularization of the original problem. Another interesting case is when the uncertainty is described with respect to the first norm (box constraints). In this case the robust formulation will be

$$\min_{w,b,\xi_i} \ \frac{1}{2}\left(\|w\|_\infty^2 + C\sum_{i=1}^{n}\xi_i^2\right) \tag{6–11a}$$

$$\text{s.t.} \ \ d_i\left(w^T(\bar{x}_i + \sigma_i) + b\right) - \rho_i\|w\|_\infty \geq 1 - \xi_i, \ \ i = 1, \dots, n \tag{6–11b}$$

$$\xi_i \geq 0, \ \ i = 1, \dots, n \tag{6–11c}$$

Since the dual of l_1 norm is the l_∞ norm. If we further more assume that the norm of the loss function is expressed with respect to the l_1 norm then the obtained optimization problem can be solved as a linear program (LP). More specificaly if we introduce the

auxilary variable α. We can obtain the following equivalent formulation of problem 6–11

$$\min_{\alpha,w,b,\xi_i} \quad \alpha + \langle C, \xi \rangle \tag{6–12a}$$

$$\text{s.t.} \quad d_i \left(w^T(\bar{x}_i + \sigma_i) + b \right) - \rho_i \alpha \geq 1 - \xi_i \qquad i = 1, \dots, n \tag{6–12b}$$

$$\xi_i \geq 0 \qquad i = 1, \dots, n \tag{6–12c}$$

$$\alpha \geq -w_k \qquad k = 1, \dots, n \tag{6–12d}$$

$$\alpha \geq w_k \qquad k = 1, \dots, n \tag{6–12e}$$

$$\alpha \geq 0 \tag{6–12f}$$

On the other side if the perturbations are expressed with respect to the l_∞ norm then the equivalent formulation of SVM is

$$\min \quad (\|w\|_1 + \langle C, \xi \rangle) \tag{6–13a}$$

$$\text{s.t.} \quad d_i \left(w^T x_i + b \right) - \rho_i \|w\|_1 \geq 1 - \xi_i \qquad i = 1, \dots, n \tag{6–13b}$$

$$\xi_i \geq 0 \qquad i = 1, \dots, n \tag{6–13c}$$

In the same way if we introduce a

$$\min_{\alpha_i,w,b,\xi_i} \sum_{i=1}^{n} \alpha_i + \langle C, \xi \rangle \tag{6–14a}$$

$$\text{s.t.} \quad d_i \left(w^T \bar{x} + b \right) - \rho_i \sum_{i=1}^{n} \alpha_i \geq 1 - \xi_i \qquad i = 1, \dots, n \tag{6–14b}$$

$$\xi_i \geq 0 \qquad i = 1, \dots, n \tag{6–14c}$$

$$\alpha_i \geq -w_i \qquad i = 1, \dots, n \tag{6–14d}$$

$$\alpha_i \geq w_i \qquad i = 1, \dots, n \tag{6–14e}$$

$$\alpha_i \geq 0 \qquad i = 1, \dots, n \tag{6–14f}$$

It is worth noting that for all robust formulations of SVM the classification rule remains the same as for the nominal case

$$\mathrm{class}(u) = \mathrm{sgn}\left(w^T u + b\right) \tag{6–15}$$

CHAPTER 7
ROBUST GENERALIZED EIGENVALUE CLASSIFICATION

In this chapter, we will present our results on the robust counterpart formulation of regularized generalized eigenvalue classifier (ReGEC). We will consider the case of ellipsoidal perturbations as well as the case of norm constrained correlated perturbations. At the end of the chapter we discuss the potential extension of the proposed algorithms for the non linear case. Part of this work has been published in [79].

7.1 Under Ellipsoidal Uncertainty

Suppose that we are given the problem of Eq. 2–67 with the additional information that the available value of data points for the first class is $\tilde{A} = A + \Delta A$ instead of A, and $\tilde{B} = B + \Delta B$ instead of B. We further impose $\Delta A \in \mathcal{U}_A, \Delta B \in \mathcal{U}_B$, where $\mathcal{U}_A, \mathcal{U}_B$ are the sets of admissible uncertainties. If we use the following transformation

$$\tilde{G} = [\tilde{A} \quad -e]^T[\tilde{A} \quad -e], \quad \tilde{H} = [\tilde{B} \quad -e]^T[\tilde{B} \quad -e], \tag{7–1}$$

the RC formulation is given by the following min-max problem

$$\min_{z \neq 0} \max_{\Delta A \in \mathcal{U}_A, \Delta B \in \mathcal{U}_B} \frac{z^T \tilde{G} z}{z^T \tilde{H} z}. \tag{7–2}$$

For general uncertainty sets $\mathcal{U}_A, \mathcal{U}_B$ the problem is intractable [75]. Similar $\min - \max$ optimization problems with a Rayleigh quotient objective function arise in other applications like Fisher Linear Discriminant Analysis (LDA), signal processing and finance [39, 40]. Unfortunately it is not possible to apply the robust results of LDA here because of the lack o the special structure of the enumerator matrix. More precisely the LDA's enumerator is a dyad (a matrix of rank 1 that is a result of the outer product of a vector with itself). In the ReGEC case this would correspond to a classification problem for which one of the classes contains just one sample. For this we will examine the problem from scratch.

In case that the uncertainty information is given in the form of norm bounded inequalities involving matrices ΔG and ΔH, then it is possible to associate Tikhonov regularization, as used in [45], with the solution to the following robust problem

$$\min_{z\neq 0} \max_{\|\Delta G\|\leq \eta_1} \frac{z^T(G+\Delta G)z}{z^T H z} \tag{7–3}$$

This can be stated through the following theorem

Theorem 7.1. *The solution for the following* $\min - \max$ *problem*

$$\min_{z\neq 0} \max_{\|\Delta G\|\leq \eta_1, \|\Delta H\|\leq \eta_2} \frac{z^T(G+\Delta G)z}{z^T(H+\Delta H)z} \tag{7–4}$$

is given by the eigenvector related to the smallest eigenvalue of the following generalized eigenvalue system

$$(G+\eta_1 I)z = \lambda(H-\eta_2 I)z \tag{7–5}$$

Proof. : The original problem of Eq. 7–2 can be written as

$$\min_{z\neq 0} \frac{\max_{\|\Delta G\|\leq \eta_1} z^T(G+\Delta G)z}{\min_{\|\Delta H\|\leq \eta_2} z^T(H+\Delta H)z} \tag{7–6}$$

We now consider the two individual problems for the enumerator and the denominator. More specifically one can write the corresponding KKT system for the enumerator problem

$$zz^T + \lambda 2\Delta G = 0 \tag{7–7}$$

$$\lambda\left(\|\Delta G\| - \eta_1\right) = 0. \tag{7–8}$$

Solving Eq. 7–7 with respect to ΔG yields

$$\Delta G = -\frac{1}{2\lambda} zz^T. \tag{7–9}$$

From Eq. 7–8 we can get

$$\lambda = \pm\frac{\|z\|^2}{2\eta_1}, \tag{7–10}$$

which gives the final expression for ΔG

$$\Delta G = \pm \frac{\eta_1}{\|z\|^2} z z^T. \tag{7–11}$$

These two matrices correspond to the maximum and the minimum of the problem, respectively. Substituting Eq. 7–11 in the enumerator of Eq. 7–4 we obtain

$$z^T (G + \eta_1 I) z \tag{7–12}$$

Repeating the same process for the denominator and substituting into the master problem shows that Eq. 7–4 is equivalent to the following minimization problem

$$\min_{z \neq 0} \frac{z^T (G + \eta_1 I) z}{z^T (H - \eta_2 I) z} \tag{7–13}$$

This problem attains its minimum when z is equal to the eigenvector that corresponds to the smallest eigenvalue of the generalized eigenvalue problem of the following system

$$(G + \eta_1 I) z = \lambda (H - \eta_2 I) z \tag{7–14}$$

□

The regularization used by Mangasarian et al. in [45] uses $\Delta G = \eta_1 I, \Delta H = 0$ for the solution of the first, and $\Delta G = 0, \Delta H = \eta_2 I$ for the second problem and therefore it is a form of robustification. Unfortunately, when bounds are given on the norms of ΔG and ΔH, we have no direct way to derive the perturbations introduced on points, namely ΔA and ΔB. This means that the regularization is providing a robustification, but we cannot know to what perturbation with respect to the training points.

It is worth noting that a similar theorem has already been stated in [64] for a completely different problem in the field of adaptive filter design. Here it is the first time that its relation with robust classification methods has been shown to provide a

straightforward connection between regularization and robustification for generalized eigenvalue classifiers.

In case that the uncertainty information is not given in the above form (e.g., we have information for each specific data point) the solution provided by Eq. 7–5 gives a very conservative estimate of the original solution. For this reason we have to propose alternative algorithmic solutions that can take into consideration all additional information available for the problem.

7.1.1 Robust Counterpart Under Ellipsoidal Uncertainty Set

Now we will focus on the following uncertainty sets where the perturbation information is explicitly given for each data point in the form of an ellipsis

$$\mathcal{U}_{\mathcal{A}} = \left\{ \Delta A \in \mathbb{R}^{m \times n}, \Delta A = [\delta_i^{(A)} \; \delta_2^{(A)} \; \dots \delta_m^{(A)}]^T : \delta_i^{(A)T} \Sigma_i \delta_i^{(A)} \leq 1, i = 1, \dots, m \right\} \tag{7–15}$$

and

$$\mathcal{U}_{\mathcal{B}} = \left\{ \Delta B \in \mathbb{R}^{p \times n}, \Delta B = [\delta_i^{(B)} \; \delta_2^{(B)} \; \dots \delta_p^{(B)}]^T : \delta_i^{(B)T} \Sigma_i \delta_i^{(B)} \leq 1, i = 1, \dots, p \right\} \tag{7–16}$$

where $\delta_i^{(A)}$, $i = 1, \dots, m$ and $\delta_i^{(B)}$, $i = 1, \dots, p$ are the individual perturbations that occur in each sample and $\Sigma_i \in \mathbb{R}^{n \times n}$ is a positive definite matrix that defines the ellipse's size and rotation. This covers the Euclidean norm case when Σ_i is equal to the unit matrix.

Since the objective functions in enumerator and denominator in Eq. 2–65 are nothing but the sum of distances of the points of each class from the class hyperplane, we can consider the problem of finding the maximum (or minimum) distance from an ellipse's point to the hyperplane defined by $w^T x - b = 0$. Since the distance of point to a hyperplane is given by $|w^T x - b|/\|w\|$ the problem can be written as

$$\max \; |w^T x - b| \tag{7–17a}$$

$$\text{s.t.} \; (x - x_c)^T \Sigma (x - x_c) - 1 \leq 0 \tag{7–17b}$$

where x_c is the ellipse's center. We can consider the two cases of the problem

$$\max \quad w^T x - b \tag{7–18a}$$

$$\text{s.t.} \quad (x - x_c)^T \Sigma (x - x_c) - 1 \leq 0 \tag{7–18b}$$

$$w^T x \geq 0 \tag{7–18c}$$

and

$$\max \quad -w^T x + b \tag{7–19a}$$

$$\text{s.t.} \quad (x - x_c)^T \Sigma (x - x_c) - 1 \leq 0 \tag{7–19b}$$

$$w^T x \leq 0 \tag{7–19c}$$

Now let us consider problem of Eq. 7–18. The corresponding KKT system will be

$$w^T - 2\mu_1 (x - x_c)\Sigma - \mu_2 w^T = 0 \tag{7–20a}$$

$$(x - x_c)^T \Sigma (x - x_c) - 1 \leq 0 \tag{7–20b}$$

also there exist μ_1, μ_2 such that

$$\mu_1 \left[(x - x_c)^T \Sigma (x - x_c) - 1 \right] = 0 \tag{7–21a}$$

$$-\mu_2 w^T x = 0 \tag{7–21b}$$

From Eq. 7–21b we derive $\mu_2 = 0$, because in different case the point should satisfy the equation of the line. If we solve Eq. 7–20a with respect to x and substitute in Eq. 7–20b we obtain the following expression for μ_1

$$\mu_1 = \pm \frac{\sqrt{w^T \Sigma^{-1} w}}{2} \tag{7–22}$$

which gives the expression for x

$$x = \pm \frac{\Sigma^{-1} w}{\sqrt{w^T \Sigma^{-1} w}} + x_c \tag{7–23}$$

The two different points correspond for the minimum and maximum distance point on to the ellipse. If we consider problem of Eq. 7–19 we will arrive to the same result. Since the solution is expressed as a function of w, and thus z, we will have for our original problem

$$\min_{z\neq 0} \max_{\Delta A\in\mathcal{U}_A, \Delta B\in\mathcal{U}_B} \frac{z^T G z}{z^T H z} = \min_{z\neq 0} \frac{z^T H(z) z}{z^T G(z) z} \tag{7–24}$$

The latter problem cannot be solved with the corresponding generalized eigenvalue problem, because the matrices depend on z. For this we use an iterative algorithm for finding the solution. First, we solve the nominal problem and we use this solution as the starting point of the iterative algorithm. Next, starting from the initial solution hyperplanes, we estimate the "worst case" point by Eq. 7–23. Then, we solve again the problem but this time for the updated points. The process is repeated until the solution hyperplanes do not change much, or a maximum number of iterations has been reached. An algorithmic description of the iterative procedure is shown in Algorithm 7.1.1.

Algorithm 7.1.1 Training Robust Iterative ReGEC

$z_1 = [w_1^T \ \gamma_1]^T = \arg\min \frac{\|Aw-e\gamma\|^2}{\|Bw-e\gamma\|^2}$
$A^{(1)} \leftarrow A, B^{(1)} \leftarrow B$
$z_0 \leftarrow$ any value such that $\|z_1 - z_0\| > \epsilon$
$i \leftarrow 1$
while $\|z_i - z_{i-1}\| \leq \epsilon$ or $i \leq i_{max}$ **do**
  $i \leftarrow i + 1$
  for each row $x_j^{(i)}$ of matrix $A^{(i)}$ **do**
    $x_j^{(i)} \leftarrow \max\left\{ x_j^{(1)} + \frac{\Sigma_j^{-1} w_i}{\sqrt{w_i^T \Sigma_j^{-1} w_i}}, x_j^{(1)} - \frac{\Sigma_j^{-1} w_i}{\sqrt{w_i^T \Sigma_j^{-1} w_i}} \right\}$
  end for
  for each row $x^{(i)}$ of matrix $B^{(i)}$ **do**
    $x_j^{(i)} \leftarrow \min\left\{ x_j^{(1)} + \frac{\Sigma_j^{-1} w_i}{\sqrt{w_i^T \Sigma_j^{-1} w_i}}, x_j^{(1)} - \frac{\Sigma_j^{-1} w_i}{\sqrt{w_i^T \Sigma_j^{-1} w_i}} \right\}$
  end for
  Form updated matrices $A^{(i)}, B^{(i)}$
  $z_i = [w_{i+1}^T \ \gamma_{i+1}]^T = \arg\min \frac{\|A^{(i)}w-e\gamma\|^2}{\|B^{(i)}w-e\gamma\|^2}$
end while
return $z_i = [w_i^T \ \gamma_i]^T$

A similar process is followed for class B. In case of k classes Algorithm 7.1.1 is applied for all the combinations of classes and then the final hyperplane is obtained through SVD on the matrix containing the $k-1$ normal vectors. This is the same process followed for the original ReGEC [31]. Also, it is worth noting that the testing part of the algorithm is exactly the same to the one for the nominal problem. Thus the labels of the unknown points are decided with the rule described by Eq. 2–72 where w_i are estimated from algorithm 7.1.1, in case of two classes, and with the training point projection method explained earlier, in case of three or more classes.

7.1.2 Balancing Between Robustness and Optimality

Robust approaches have been criticized for providing over conservative solutions in the sense that they are optimal only when the worst assumed scenario occurs [12]. In real life applications, it might be more interesting to robustify an algorithm against an average case scenario. For this reason, we aim at providing a model that can be adjustable. Instead of using Eq. 7–23, we can take the convex combination of the nominal data points and the "robust" ones

$$x^{balanced} = \xi \cdot x_c + (1-\xi)\left(\frac{\Sigma^{-1}w}{\sqrt{w^T\Sigma^{-1}w}} + x_c\right) = x_c + (1-\xi)\left(\frac{\Sigma^{-1}w}{\sqrt{w^T\Sigma^{-1}w}}\right), \quad 0 \leq \xi \leq 1. \tag{7–25}$$

Parameter ξ determines how close points will be to their nominal values or to their "worst scenario" ones. For computational purposes we can chose ξ by generating several average based scenarios and selecting the value that gives the lowest objective function value for the training data points. We are going to call this method ξ-Robust Regularized Generalized Eigenvalue Classifier (ξ-R-ReGEC).

7.2 Computational Results

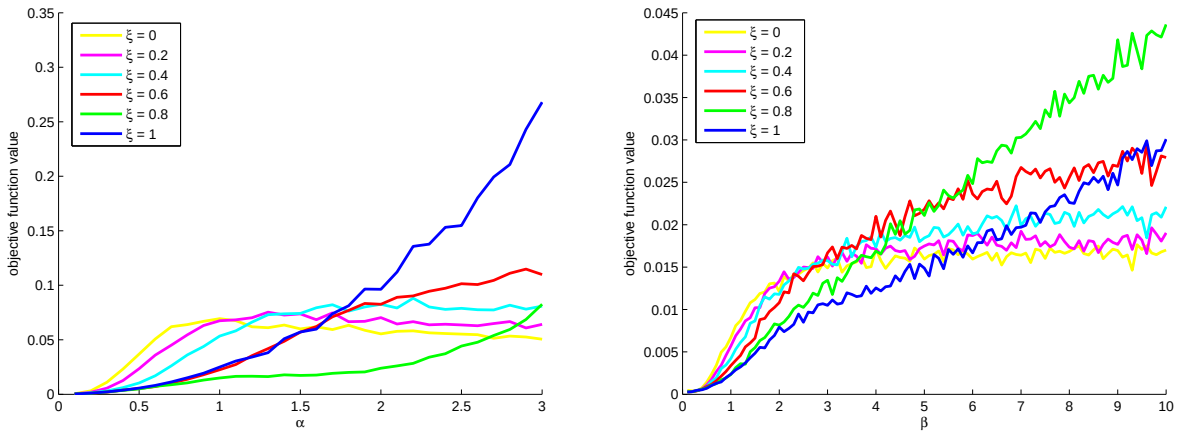
7.2.1 A Case Study

Now we will illustrate how ξ-R-ReGEC works under ellipsoidal uncertainty. In this example each class is composed of three points. Let us assume for all points the

ellipsoidal uncertainty is the same and it is described by the matrix $\Sigma = I \cdot [\alpha^{-2} \ \ \beta^{-2}]^T$ (where I is the unit matrix). We first consider the simple two class example where each class is represented by

$$A = \begin{bmatrix} 5.00 & 4.00 & 4.63 \\ 7.11 & 5.36 & 4.42 \end{bmatrix}^T, \ B = \begin{bmatrix} 2.82 & 2.00 & 1.00 \\ 1.44 & 7.11 & -0.68 \end{bmatrix}^T \tag{7–26}$$

In order to examine the behavior of the robust solution we perform the following test. We compute the nominal solution based on the data values without any perturbation. Then we assume an ellipsoidal perturbation in two dimensions and we compute the robust solution. We create 1000 different realizations (different sets of points perturbed with ellipsoid perturbation) and we compute the objective function value for the robust classifier and the nominal one. This experiment is repeated for increasing values of α and β. The results are shown in Fig. 7-1.



(a) **(b)**

Figure 7-1. Objective function value dependence on the ellipse parameters.

We note that for small values of ellipse parameters the robust solution is very conservative. This means that it might be optimal for the assumed worst case scenario, but it does not perform well in the average case. As the perturbation increases robust solution tends to have a constant behavior for any realization of the system.

7.2.2 Experiments

Now we will demonstrate the performance of the algorithm on data sets from UCI repository [28]. Dataset characteristics are shown in Table 7-1.

Table 7-1. Dataset description.

Dataset name	# of points	# of attributes	# of classes
Pima Indian [69]	768	8	2
Iris [27]	150	4	3
Wine [2, 3]	178	13	3
NDC [48]	300	7	2

For each run we used hold out cross validation with 50 repetitions. In every repetition 90% of the samples were used for training and 10% for testing. At each repetition we train the robust algorithm with the nominal data plus the uncertainty information and we test it on a random realization of the testing dataset, that is nominal values of testing dataset plus noise that satisfies the ellipsoidal uncertainty constraints. The uncertainty was kept constant and equal to 0.1 for all dimensions (features). For the non fixed dimension the perturbation is set equal to α that is a parameter for our experiments. All data features are initially normalized so that they have 0 mean and unitary standard deviation. All code was developed in MATLAB. The robust SVM solver used for comparison is written in Python and run under Matlab.

The results are reported in Figs. 7-2 , 7-3, 7-4 & 7-5 in form of heat maps. For each considered dataset, we plot the results of ReGEC algorithm in the left panel, and those attained by ξ-R-ReGEC in the right one. Results have been obtained with the above described cross validation technique, for each fixed value of ξ and α. Each tone of gray represents the average accuracy for all cross validation repetitions, as reported in the legend. We notice that, in all considered cases, right panels show higher classification values for larger values of α. This confirms it is possible to find values of ξ for which the classification model is resilient to perturbations. We notice that some datasets are less sensitive to the value of ξ. In the Pima Indian and Iris datasets, the classification accuracy increases as the value of α increases for $0.4 \leq \xi \leq 0.8$. For Wine and NDC

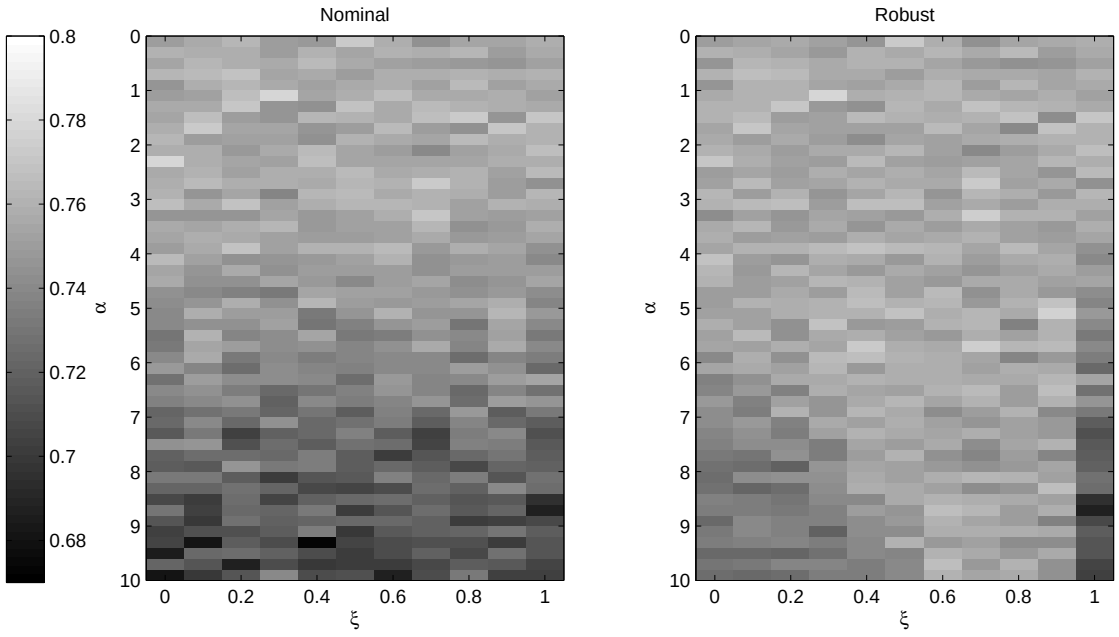


Figure 7-2. Analysis for Pima indian dataset. The horizontal axis determines the ξ parameter that balances between robust and nominal solution whereas the vertical axis is the perturbation parameter.

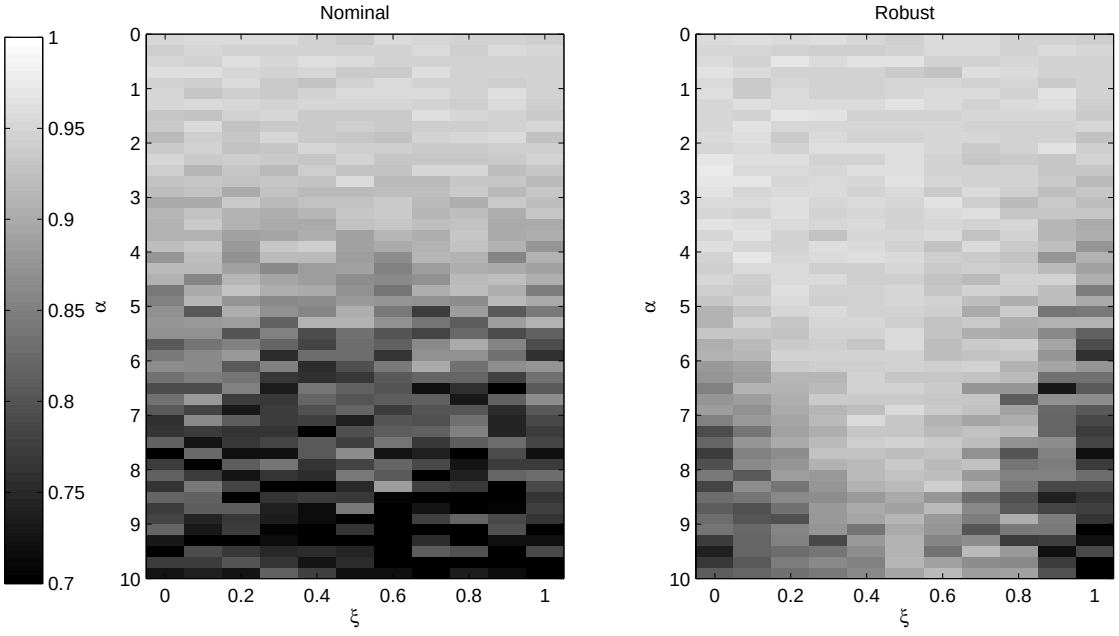


Figure 7-3. Analysis for iris dataset. The horizontal axis determines the ξ parameter that balances between robust and nominal solution whereas the vertical axis is the perturbation parameter.

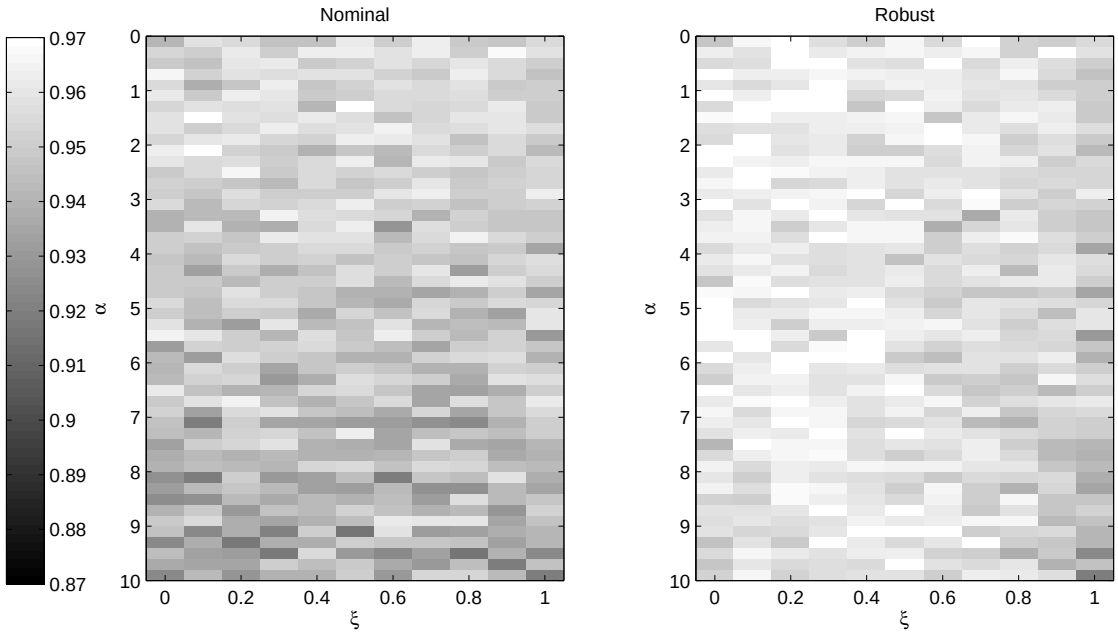


Figure 7-4. Analysis for wine dataset. The horizontal axis determines the ξ parameter that balances between robust and nominal solution whereas the vertical axis is the perturbation parameter.

datasets, the robust classifier obtains nearly constant classification accuracy for large values of α and $0 \leq \xi \leq .8$.

Next we compare our proposed algorithm against Robust SVM. For this we used the package CVXOPT [73] and in particular the customized solver for Robust SVM under ellipsoidal uncertainties also described in [4]. The original formulation of the problem is from [67]. More precisely, the R-SVM problem is solved as a second order cone quadratic program (QP) with second order cone constraints. For a two class classification problem, the separating hyperplane is given by the solution of the following

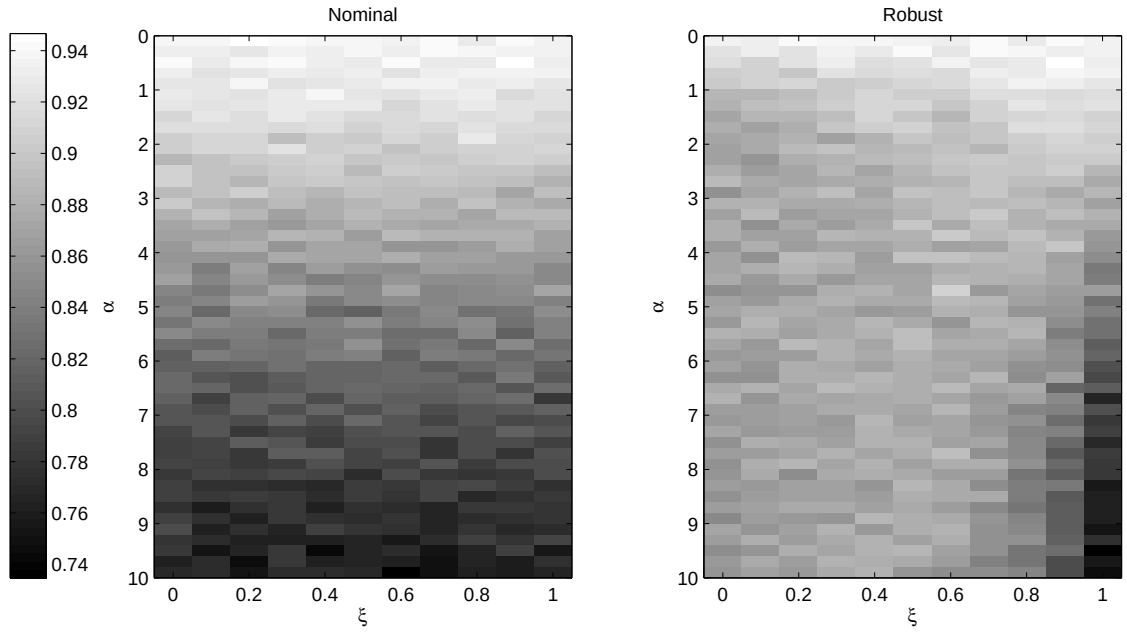


Figure 7-5. Analysis for dataset NDC. The horizontal axis determines the ξ parameter that balances between robust and nominal solution whereas the vertical axis is the perturbation parameter.

Second Order Cone Program (SOCP)

$$\min \ \frac{1}{2}\|w\|^2 + \xi e^T \upsilon \tag{7–27a}$$

$$\text{s.t.} \ \ \mathrm{diag}(d)(Xw - be) \succeq 1 - \upsilon - Eu \tag{7–27b}$$

$$u \succeq 0 \tag{7–27c}$$

$$\|\Sigma_j\| \leq u_j, \ \ j = 1, \ldots, t \tag{7–27d}$$

Where the $\mathrm{diag}(x)$ is the diagonal matrix that has vector x in the main diagonal, $d \in \{0, 1\}^m$ is the label vector, E is an indicator matrix that associates an ellipse with a corresponding data point, i.e. if $E_{ij} = 1$ means that the ith ellipsoid is associated with the jth data point, and Σ_j is a positive semidefinite matrix that defines the shape of the ellipse. The separation hyperplane is defined by $w, b.$ υ, u are additional decision variables and ξ is a parameter that penalizes the wrong classified samples. Note that for

$t = 0$ and $E = 0$ the problem reduces to the original soft margin SVM classifier

$$\min \ \frac{1}{2}\|w\|^2 + \xi e^T \upsilon \tag{7–28a}$$

$$\text{s.t.} \ \ \text{diag}(d)(Xw - be) \succeq 1 - \upsilon \tag{7–28b}$$

$$u \succeq 0 \tag{7–28c}$$

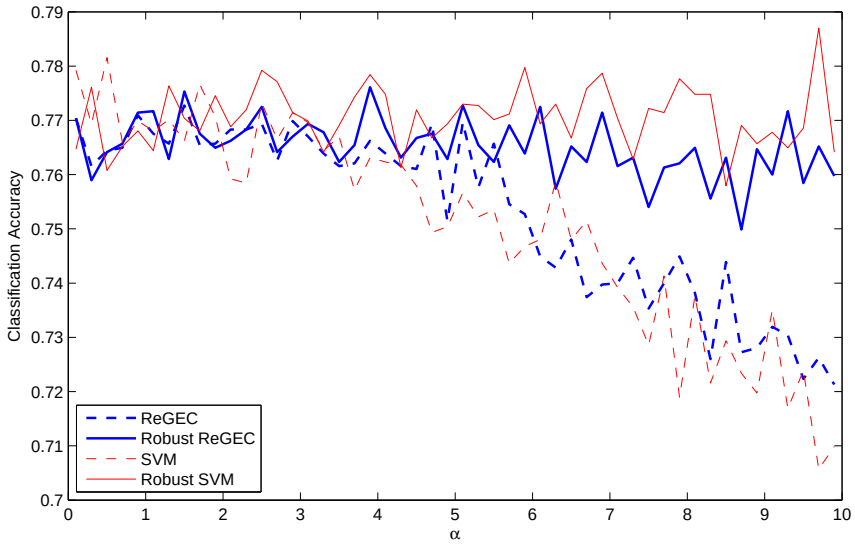
The comparison results between ReGEC and SVM for the four datasets of Table 7-1 are shown in Fig. 7-6 & Fig. 7-7.

In Fig. 7-6 & Fig. 7-7 horizontal axes correspond to mean classification accuracies. As before the hold out cross validation with 50 repetitions was used. For ReGEC the ξ parameter was adjusted in each run through the training phase and for SVM γ was chosen through a uniform grid search. It is worth mentioning that alterations of γ did not change dramatically the obtained classification accuracies.

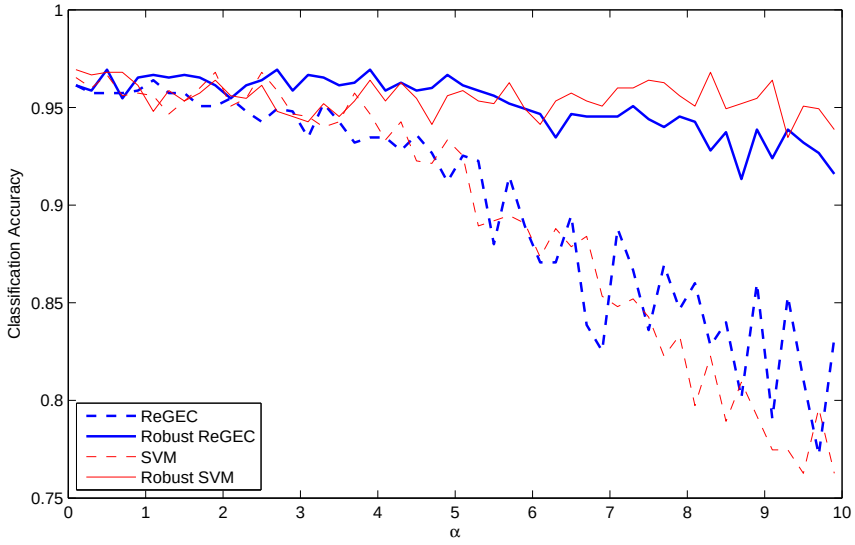
In all cases and for both algorithms, as α increases, the accuracy of the two nominal algorithms decreases more rapidly than for RCs. Furthermore, the RCs always obtain higher accuracy values. For three datasets (Pima Indian, Wine and Iris) both robust algorithms perform equally well even for large perturbation values. For NDC, ReGEC is better especially for perturbations less that $\alpha = 5$. In general, both non robust algorithms fail to achieve a high classification accuracy for high perturbation values. We can therefore conclude that the proposed robustification of the algorithm is well suited for problems in which uncertainty on training points can be modeled as belonging to ellipsis.

7.3 Extentions

In this work a robust formulation of ReGEC is introduced, that can handle perturbation in the form of ellipsoidal uncertainties in data. An iterative algorithm is proposed to solve the associated min-max problem. Results on UCI datasets show an increased classification accuracy, when synthetic ellipsoidal perturbations are introduced in training data.



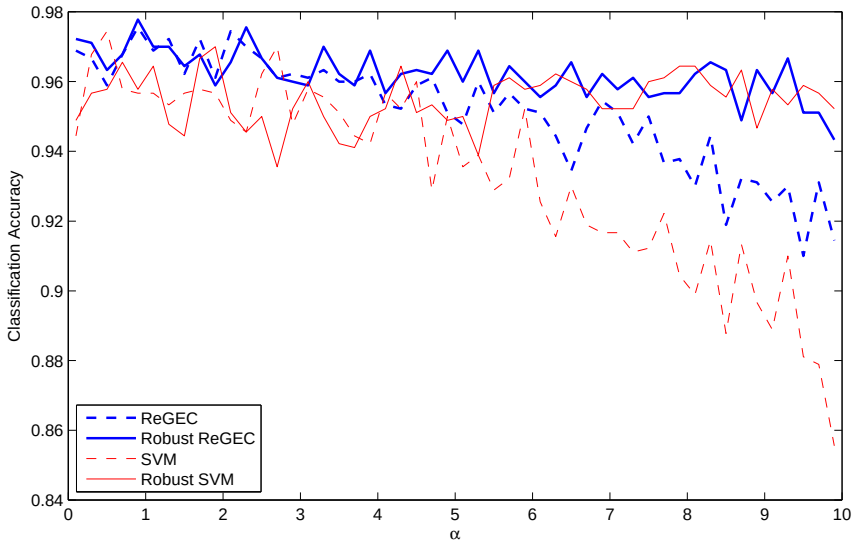
(a)



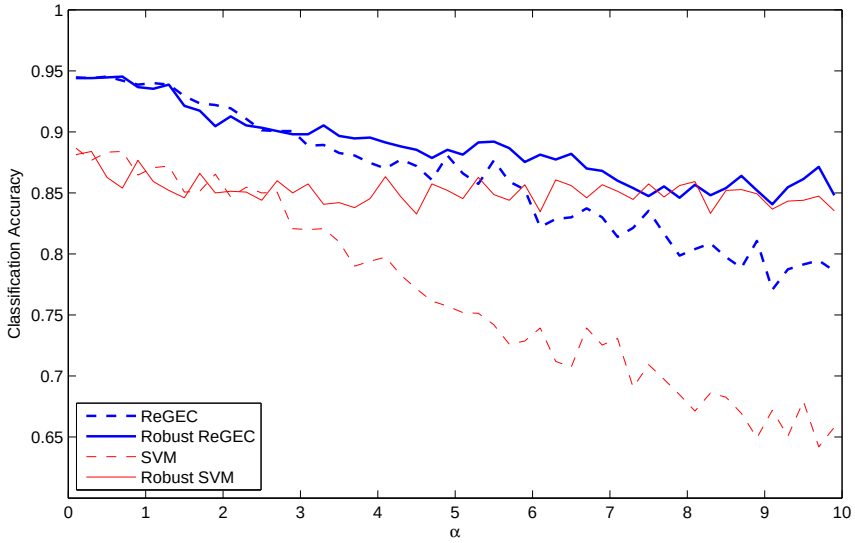
(b)

Figure 7-6. Comparison graph for (a) Pima indian (b)iris datasets. Red color corresponds to ReGEC whereas blue for SVM. Solid line corresponds to the robust and dashed to nominal formulation.

Some problems still remain open. First, in the kernel version of the algorithm. Since data are non linearly projected in the feature space, a representation of the ellipsoidal perturbation in the feature space is needed. Then, we note that this embedding is not trivial, as the convex shape of the ellipsoid in the original space is not preserved in



(c)



(d)

Figure 7-7. Comparison graph for (a) wine and (b)NDC datasets. Red color corresponds to ReGEC whereas blue for SVM. Solid line corresponds to the robust and dashed to nominal formulation.

the feature space, thus resulting in a much more difficult problem. For this reason, the direction proposed in [55] seems promising and worth investigating. In addition, further uncertainty scenarios need to be explored. Some interesting uncertainty sets include norm bounded uncertainties on the data matrix i.e. when $\|\Delta A\| \leq \rho$, box constraints i.e.

when $\|\delta_i\|_1 \leq \rho_i$ or polyhedral constraints i.e. when $A_i\delta_i \leq b_i$. Most of these scenarios have been explored for other classifiers (e.g., SVM) introducing significant performance improvement over the nominal solution schemes. Finally, since a natural application of this method is in the field of biomedical and bioinformatic applications, we will investigate methods to reduce the number of features involved in the solution process.

7.3.1 Matrix Norm Constrained Uncertainty

In this section we will provide some theoretical results for the case that the norm of the perturbation of each matrix is restricted by a known constant. This case is, sometimes, refer to as correlated perturbation because the perturbation of each sample is dependent on the perturbation of all the others. This is not the case in the ellipsoid perturbation scenario where the uncertainty set is independent for each data sample. As we discussed earlier it is possible to derive a robust formulation of ReGEC when the perturbation information is known for the matrices G and H. In practice this is not the case, thus we wish to derive a robust formulation for the case that the informatio is given directly for the data matrices A and B. In this case, data matrices A and B are replaced by $\tilde{A} = A + \Delta A$ and $\tilde{B} = B + \Delta B$. Then we can consider the following $\min - \max$ optimization problem

$$\min_{z\neq} \max_{\|\Delta A\|\leq\rho_1, \|\Delta B\|\leq\rho_2} \frac{\|\tilde{A}w - e\gamma\|^2}{\|\tilde{B}w - e\gamma\|^2} = \min_{z\neq 0} \frac{\max_{\|\Delta A\|\leq\rho_1} \|\tilde{A}w - e\gamma\|^2}{\min_{\|\Delta\tilde{B}\|\leq\rho_2} \|\tilde{B}w - e\gamma\|^2} \tag{7–29}$$

Again we consider the maximization problem on the enumerator

$$\max_{\|\Delta A\|\leq\rho_1} \|\tilde{A}w - e\gamma\|^2 \tag{7–30}$$

Since $\rho_1 > 0$ the constraint can be replaced by $\|\Delta A\|^2 \leq \rho_1^2$. Then the corresponding KKT system will be

$$(\Delta A w + A w - e\gamma)\, w^T + \lambda \Delta A = 0 \tag{7–31}$$

$$\lambda \left(\|\Delta A\|^2 - \rho_1^2\right) = 0 \tag{7–32}$$

From Eq. 7–31 we can get

$$\Delta A \left(w w^T + \lambda I \right) = - \left(A w - e \gamma \right) w^T \tag{7–33}$$

In general for a matrix $M = K + c d^T$ where K is nonsingular and c, d are column vectors of appropriate dimensions we can write [26]

$$M^{-1} = K^{-1} - \frac{1}{\beta} K^{-1} c d^T K^{-1} \tag{7–34}$$

where $\beta = 1 + d^T K^{-1} c$. Applying this result to matrix $w w^T + \lambda I$ yields

$$(w w^T + \lambda I)^{-1} = \left(\frac{1}{\lambda} I - \frac{1}{\lambda(\lambda + \|w\|^2)} w w^T \right) \tag{7–35}$$

Thus we obtain an expression of ΔA as a function of λ

$$\Delta A = -(A w - e \gamma) w^T \left(\frac{1}{\lambda} I - \frac{1}{\lambda(\lambda + \|w\|^2)} w w^T \right) = -(A w - e \gamma) w^T \left(\frac{1}{\lambda + \|w\|^2} \right) \tag{7–36}$$

In order to obtain an expression for λ we can take the norm of both sides in Eq. 7–36.

$$\|\Delta A\| = \|(A w - e \gamma) w^T\| \left| \frac{1}{\lambda + \|w\|^2} \right| \tag{7–37}$$

and from this

$$\lambda = \pm \frac{\|(A w - e \gamma) w^T\|}{\|\Delta A\|} - \|w\|^2 \tag{7–38}$$

Substituting back in Eq. 7–36 and using Eq. 7–32 we obtain the final expression for ΔA

$$\Delta A = \pm (A w - e \gamma) w^T \cdot \frac{\rho_1}{\|(A w - e \gamma) w^T\|} \tag{7–39}$$

At this point it is worth noting the symmetry between Eq. 7–39 and Eq. 7–11. Again the two points correspond to the minimum and the maximum correspondingly. Substituting in the enumerator of the original problem we get

$$\left\| \left(A + (A w - e \gamma) w^T \cdot \frac{\rho_1}{\|(A w - e \gamma) w^T\|} \right) w - e \gamma \right\|^2 = \left\| (A w - e \gamma) \left(1 + \frac{\|w\|^2 \rho_1}{\|(A w - e \gamma) w^T\|} \right) \right\|^2 \tag{7–40}$$

By using Cauchy-Schwartz inequality we have

$$\|(Aw - e\gamma)w^T\| \leq \|Aw - e\gamma\|\|w\| \tag{7–41}$$

On the other side by triangular inequality we can obtain

$$\|(A + \Delta A)w - e\gamma\| \leq \|Aw - e\gamma\| + \rho_1\|w\|, \text{for every } \Delta A \in \mathbb{R}^{n\times m} \tag{7–42}$$

and for

$$\Delta A = \frac{Aw - e\gamma}{\|Aw - e\gamma\|\|w\|}w^T\rho_1 \tag{7–43}$$

we obtain equality in Eq. 7–42. Convexity of the problem guarantees the unity of the global maximum. Then the robust solution to the original problem is obtained by solving the following problem

$$\min_{w\neq 0,\gamma} \frac{(\|Aw - e\gamma\| + \rho_1\|w\|)^2}{(\|Aw - e\gamma\| - \rho_2\|w\|)^2} \tag{7–44}$$

If we suppose, without loss of generality that $\|\omega\| = 1$, we can write

$$\min_{\|\omega\|=1,\gamma\neq 0} \left(\frac{\|A\omega - e\gamma\| + \rho_1}{\|B\omega - e\gamma\| - \rho_2}\right)^2 \tag{7–45}$$

If we set $G = [A \quad -e]^T[A \quad -e]$ and $H = [B \quad -e]^T[B \quad -e]$, we can write

$$\min_{\|\omega\|=1,\gamma\neq 0} \left(\frac{\sqrt{z^T Gz} + \rho_1}{\sqrt{z^T Hz} - \rho_2}\right)^2 \tag{7–46}$$

This is a fractional problem that cannot be solved directly. We can transform this problem into a parametric one by making use of Dinkelbach's following result

Theorem 7.2. *For two continuous and real valued functions f(x), g(x) that are defined on a compact and connected subset S of Euclidean space the problems*

$$\max_{x\in S} \frac{f(x)}{g(x)} = \alpha \tag{7–47}$$

and

$$\max_{x\in S} [f(x) - \alpha g(x)] = 0 \tag{7–48}$$

are equivalent.

Proof. [24] □

This is a strong and general result that associates fractional and parametric programming. This enables us to transform our original problem into

$$\min_z (\sqrt{z^T G z} + \rho_1)^2 - \lambda(\sqrt{z^T H z} - \rho_2)^2 \tag{7–49}$$

Based on Dinkelbach's the optimal solution (z, λ) is the one for which the objective function is equal to zero. The Lagrangian of the problem is

$$L(z, \lambda) = (\sqrt{z^T G z} + \rho_1)^2 - \lambda(\sqrt{z^T H z} - \rho_2)^2 \tag{7–50}$$

if we set $\bigtriangledown_z L(z, \lambda) = 0$ we get

$$(\sqrt{z^T G z} + \rho_1)\frac{Gz}{\sqrt{z^T G z}} = \lambda(\sqrt{z^T H z - \rho_2})\frac{Hz}{\sqrt{z^T H z}} \tag{7–51}$$

or

$$\frac{\sqrt{z^T G z} + \rho_1}{\sqrt{z^T G z}} Gz = \lambda \left(\frac{\sqrt{z^T H z} - \rho_1}{\sqrt{z^T H z}} \right) Hz \tag{7–52}$$

Note that when $\rho_1 = \rho_2 = 0$ the problem reduces to the original eigenvalue system

$$Gz = \lambda Hz \tag{7–53}$$

we can try solving the problem as follows

1. Solve the original problem of Eq. 7–53 to obtain z_0
2. Iterate and obtain z_i from

$$\frac{\sqrt{z_i^T G z_i} + \rho_1}{\sqrt{z_i^T G z_i}} Gz_{i+1} = \lambda \left(\frac{\sqrt{z_i^T H z_i} - \rho_2}{\sqrt{z_i^T H z_i}} \right) Hz_{i+1} \tag{7–54}$$

3. stop when $\|z_{i+1} - z_i\|/\|z_i\| \leq \epsilon$

Despite we do not have any theoretical guarrantee for the convergence of this algorithm we know that we are close to the optimal solution when $L(z, \lambda)$ is close to zero. Thus we can also use the following convergence criterion

$$L(z, \lambda) \leq \epsilon \tag{7–55}$$

In the next section we will discuss the potential non linear generalization of robust ReGEC through the use of kernel functions.

7.3.2 Non linear Robust Generalized Classification

In order to propose the non linear version of Robust ReGEC we need to introduce the concept of kernel functions. Kernel functions are used extensively in data mining clustering and classification in order to make use of powerful linear classifiers to non linearly separable problems. In this framework the points of original data space (of dimension n), which is often called *input space*, are embedded through a kernel function $\phi : \mathbb{R}^n \mapsto \mathbb{R}^m$ to a higher dimension space ($m > n$) known as the *feature space*. In such a space there is high chance that the problem is linearly separable thus application of a linear algorithm is possible. the projection of a linear classifier in the *feature space* is a non linear one in the *input space*. For ReGEC the kernelized version was proposed in [30]. For the robust case the algorithm refinement is related to the utilization of the uncertainty information. Thus embedding of ellipsoids, defined by a positive semidefinite matrix Σ_i, is different as compared to embedding of single points. In order to embed the ellipsoids in the feature space we need to express them in the following form

$$\Sigma_i = \sum_j \lambda_j x_j x_j^T \tag{7–56}$$

where λ_j are the eigenvalues and x_j the corresponding eigenvectors. The fact that the ellipsis matrices are positive semidefinite guarantees that such a decomposition exists. The mapping in feature space can be obtained if we replace each eigenvector with its

projection on the feature space. Thus we obtain the following mapping of Σ_i

$$C_i = \sum_j \lambda_j \phi(x_j)\phi(x_j)^T \tag{7–57}$$

For the computation of the worst case perturbation C_i^{-1} needs to be computed. In this case the computation is not as straightforward as in the input space. For computational purposes we replace the computation of the inverse with the least squares solution of the system

$$\min_r \| C_i r - z\|^2 \tag{7–58}$$

where Eq. 7–57 is used for C_i. For avoiding numerical instabilities with the regularized counterpart

$$\min_r \| C_i r - z\|^2 + \delta_{C_i} \|r\|^2 \tag{7–59}$$

This expression can be used in order to obtain the worst case scenario points on the feature space. Then the worst case scenario points in the feature space will be given by

$$x_{wc} = \max \left\{ x_c + \frac{r}{\sqrt{z^T r}}, x_c - \frac{r}{\sqrt{z^T r}} \right\} \tag{7–60}$$

in in order to balance between robustification and accuracy we consider the convex combination of the robust and the nominal points

$$x_{balanced} = \xi x_c + (1 - \xi) x_{wc} \tag{7–61}$$

where ξ is the parameter that determines the trade off and should be tuned during the model selection phase. These points are used for the robust algorithm. Along the same lines with the linear case the kernel version of robust ReGEC can be obtained through the following algorithmic scheme

Algorithm 7.3.1 Training Robust Iterative non linear ReGEC

$z_1 = [w_1^T \ \gamma_1]^T = \arg\min \frac{\|Aw - e\gamma\|^2}{\|Bw - e\gamma\|^2}$

$A^{(1)} \leftarrow A, B^{(1)} \leftarrow B$

$z_0 \leftarrow$ any value such that $\|z_1 - z_0\| > \epsilon$

$i \leftarrow 1$

while $\|z_i - z_{i-1}\| \leq \epsilon$ or $i \leq i_{max}$ **do**

$i \leftarrow i + 1$

for each row $x_j^{(i)}$ of matrix $A^{(i)}$ **do**

$$x_j^{(i)} \leftarrow \max \left\{ x_j^{(1)} + \frac{\Sigma_j^{-1} w_i}{\sqrt{w_i^T \Sigma_j^{-1} w_i}}, x_j^{(1)} - \frac{\Sigma_j^{-1} w_i}{\sqrt{w_i^T \Sigma_j^{-1} w_i}} \right\}$$

end for

for each row $x^{(i)}$ of matrix $B^{(i)}$ **do**

$$x_j^{(i)} \leftarrow \min \left\{ x_j^{(1)} + \frac{\Sigma_j^{-1} w_i}{\sqrt{w_i^T \Sigma_j^{-1} w_i}}, x_j^{(1)} - \frac{\Sigma_j^{-1} w_i}{\sqrt{w_i^T \Sigma_j^{-1} w_i}} \right\}$$

end for

Form updated matrices $A^{(i)}, B^{(i)}$

$z_i = [w_{i+1}^T \ \gamma_{i+1}]^T = \arg\min \frac{\|A^{(i)} w - e\gamma\|^2}{\|B^{(i)} w - e\gamma\|^2}$

end while

return $z_i = [w_i^T \ \gamma_i]^T$

CHAPTER 8
CONCLUSIONS

In this work, we presented some of the major recent advances of robust optimization in data mining. we also presented the robust counterpart formulation for ReGEC. We compared their performance with robust support vector machines (SVM) and suggested a non linear (kernalized) version of the algorithm.

In most the thesis, we examined most of data mining methods from the scope of uncertainty handling with only exception the principal component analysis (PCA) transformation. Nevertheless the uncertainty can be seen as a special case of prior knowledge. In prior knowledge classification for example we are given together with the training sets some additional information about the input space. Another type of prior knowledge other than uncertainty is the so called expert knowledge (e.g., binary rule of the type “if feature a is more than M_1 and feature b less than M_2 then the sample belongs to class x”). There has been significant amount of research in the area of prior knowledge classification [42, 62] but there has not been a significant study of robust optimization on this direction.

On the other side there have been several other methods able to handle uncertainty like stochastic programming as we already mentioned at the beginning of the manuscript. Some techniques, for example like conditional value at risk (CVAR), have been extensively used in portfolio optimization and in other risk related decision systems optimization problems [61] but their value for machine learning has not been fully investigated.

A last thought on application of robust optimization in machine learning would be as an alternative method for data reduction. In this case we could replace groups of points by convex shapes , such as balls,squares or ellipsoids, that enclose them. Then the supervised learning algorithm can be trained just by considering these shapes instead of the full sets of points.

APPENDIX A
OPTIMALITY CONDITIONS

Here we will briefly discuss the Karush Kuhn Tucker (KKT) Optimality Conditions and the method of Lagrange multipliers that is extensively used though this work. In this section, for the shake of completion we are going to describe the technical details related to optimality of convex programs and the relation with KKT systems and methods of Lagrange multipliers. First we will start by giving some essential definitions related to convexity. First we give the definition of a convex function and convex set.

Definition 1. *A function* $f : X \subseteq \mathbb{R}^m \mapsto \mathbb{R}$ *is called convex when* $\lambda f(x) + (1-\lambda)f(x) \geq f(\lambda x + (1-\lambda)x)$ *for* $0 \leq \lambda \leq 1$ *and* $\forall x \in X$.

Definition 2. *A set* X *is called convex when for any two points* $x_1, x_2 \in X$ *the point* $\lambda x_1 + (1-\lambda)x_2 \in X$ *for* $0 \leq \lambda \leq 1$.

Now we are ready to define a convex optimization problem

Definition 3. *An optimization problem* $\min_{x \in X} f(x)$ *is called convex when* $f(x)$ *is a convex function and* X *is a convex set.*

The class of convex problems is really important because in they are classified as problems that are computationally tractable. This allows the implementation of fast algorithms for data analysis methods that are realized as convex problems. Processing of massive datasets can be realized because of this property. Once we have defined the convex optimization problem in terms of the properties of its objective function and its feasible region we will state some basic results related to their optimality.

Corollary 1. *For a convex minimization problem a local minimum* x^* *is always a global minimum as well. That is if* $f(x^*) \leq (f(x))$ *for* $x \in S$ *where* $S \subseteq X$ *then* $f(x^*) \leq f(x)$ *for* $x \in X$.

Proof. Let x^* be a local minimum such that $f(x^*) < f(x), x \in S \subseteq X$ and another point $\bar{x}$ being the global minimum such that $f(\bar{x}) < f(x), x \in X$. Then by convexity of the

objective function it holds that

$$f(\lambda\bar{x} + (1-\lambda)x^*) = f(x^* + \lambda(\bar{x} - x^*)) \leq \lambda f(\bar{x}) + (1-\lambda)f(x^*) < f(x^*) \tag{A–1}$$

on the other side by local optimality of point $\bar{x}$ we have that there exist $\lambda^* > 0$ such that

$$f(x^*) \leq f(x^* + \lambda(\bar{x} - x^*)),\ 0 \leq \lambda \leq \lambda^* \tag{A–2}$$

which is a contradiction. □

This is a important consequence that explains in part the computational tracktability of convex problems. Next we define the critical points that are extremely important for the characterization of global optima of convex problems. But before that we need to introduce the notion of extreme directions.

Definition 4. *A vector $d\mathbb{R}^n$ is called feasible direction with respect to a set S at a point x if there exist $c \in \mathbb{R}$ such that $x + \lambda d \in S$ for every $0 < \lambda < c$.*

Definition 5. *For a convex optimization problem $\min_{x \in X} f(x)$ where f differentiable every point that satisfies $d^T \nabla f(x^*) \geq 0, d \in Z(x^*)$ (where $Z(x^*)$ is the set of all feasible directions of the point x^*) is called a critical (or stationary) point.*

Critical points are very important in optimization as they are used in order to characterize local optimality in general optimization problems. In a general differentiable setup stationary points characterize local minima. This is formalized through the following theorem.

Theorem A.1. *If x^* is a local minimum of a continuously diffentiable function f defined on a convex set S then it satisfies $d^T \nabla f(x^*) \geq 0, d \in Z(x^*)$.*

Proof. [33] p. 14. □

Due to the specific properties of convexity, in convex programming, critical points are used in order to characterize global optimal solutions as well. This is stated through the following theorem.

Theorem A.2. *if f is a continuously differentiable function on an open set containing S, and S is a convex set then $x^* \in S$ is a global minimum if and only if x^* is a stationary point.*

Proof. [33] pp. 14-15. □

The last theorem is a very strong result that connects stationary points with global optimality. Since stationary points are so important for solving convex optimization problems it is also important to establish a methodology that would allow us to discover such points. This is exactly the goal of Karush Kuhn Tucker conditions and method of Lagrangian multipliers (They are actually different sides of the same coin). this systematic methodology was first introduced by Lagrange in 1797 and it was generalized though the master thesis of W. Karush [38] and finally they became more popular know through the work of H. W. Kuhn and A. W. Tucker [41]. These conditions are formally stated through the next theorem.

Theorem A.3. *(KKT conditions) Given the following optimization problem*

$$\min \quad f(x) \tag{A–3a}$$

$$\text{s.t.} \quad g_i(x) \geq 0, \quad i = 1, \cdots, n \tag{A–3b}$$

$$h_i(x) = 0, \quad i = 1, \cdots, m \tag{A–3c}$$

$$x \geq 0 \tag{A–3d}$$

The following conditions (KTT) are necessary for optimality

$$\nabla f(x^*) + \sum_{i=1}^{n} \lambda_i \nabla g_i(x^*) + \sum_{i=1}^{m} \mu_i \nabla h_i(x^*) \tag{A–4a}$$

$$\lambda_i g_i(x^*) = 0 \qquad i = 1, \dots, n \tag{A–4b}$$

$$\lambda_i \geq 0 \qquad i = 1, \dots, n \tag{A–4c}$$

For the special case that $f(\cdot), g(\cdot), h(\cdot)$ are convex functions then the KKT conditions are also sufficient for optimality.

Proof. See [33] □

The Eq. A–4a is also known as Lagrangian equation and λ_i are also known as lagrange multipliers. Thus one can determine stationary for a problem by just finding the roots of the Lagrangian's first derivative. For the general case this method is formalized through the Karush-Kuhn-Tucker optimality conditions. The important of these conditions is that under convexity assumptions they are necessary and sufficient.

Due to the aforementioned results that connect stationary point with optimality we can clearly see that one can solve a convex optimization problem just by solving the corresponding KKT system. The corresponding points would be the solution to the original problem.

APPENDIX B
DUAL NORMS

Dual norms is a mathematical tool,necessary for the analysis of robust support vector machines formulation.

Definition 6. *For a norm* $\|\cdot\|$ *we define the dual norm* $\|\cdot\|_*$ *as follows*

$$\|x\|_* = sup\{x^T\alpha | \|x\| \leq \alpha\} \tag{B–1}$$

There are several properties associated with the dual norm that we will briefly discuss here.

Property 1. *A dual norm of a dual norm is the original norm itsself. In other words*

$$\|x\|_{**} = \|x\| \tag{B–2}$$

Property 2. *A dual of an* l_a *norm is* l_b *norm where* a *and* b *satisfy the following equation*

$$\frac{1}{a} + \frac{1}{b} = 1 \Leftrightarrow b = \frac{a}{a-1} \tag{B–3}$$

Immediate results of the previous property is that

- The dual norm od the Euclidean norm is the Euclidean norm (b=2/(2-1)=2).
- The dual norm of the l_1 norm is l_∞

Next we will state Hölders inequality and Cauchy Swartz inequality which are two fundamental inequalities that connect the primal and the dual norm.

Theorem B.1. *(Hölders inequality) For a pair of dual norms* a *and* b *the following inequality holds:*

$$\langle x \cdot y \rangle \leq \|x\|_a \|y\|_b \tag{B–4}$$

For the special case that $a = b = 2$ then Hölders inequality reduces to Cauchy Swartz inequality

$$\langle x \cdot y \rangle \leq \|x\|_2 \|y\|_2 \tag{B–5}$$

REFERENCES

[1] Abello, J., Pardalos, P., Resende, M.: Handbook of massive data sets. Kluwer Academic Publishers Norwell, MA, USA (2002)

[2] Aeberhard, S., Coomans, D., De Vel, O.: Comparison of classifiers in high dimensional settings. Dept. Math. Statist., James Cook Univ., North Queensland, Australia, Tech. Rep (1992)

[3] Aeberhard, S., Coomans, D., De Vel, O.: The classification performance of RDA. Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, Tech. Rep pp. 92–01 (1992)

[4] Andersen, M.S., Dahl, J., Liu, Z., Vandenberghe, L.: Optimization for Machine Learning, chap. Interior-point methods for large-scale cone programming. MIT Press (2011)

[5] Angelosante, D., Giannakis, G.: RLS-weighted Lasso for adaptive estimation of sparse signals. In: Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, pp. 3245–3248. IEEE (2009)

[6] d Aspremont, A., El Ghaoui, L., Jordan, M., Lanckriet, G.: A direct formulation for sparse pca using semidefinite programming. SIAM review **49**(3), 434 (2007)

[7] Baudat, G., Anouar, F.: Generalized discriminant analysis using a kernel approach. Neural computation **12**(10), 2385–2404 (2000)

[8] Bayes, T.: An essay towards solving a problem in the doctrine of chances. R. Soc. Lond. Philos. Trans **53**, 370–418 (1763)

[9] Ben-Tal, A., El Ghaoui, L., Nemirovski, A.S.: Robust optimization. Princeton Univ Pr (2009)

[10] Ben-Tal, A., Nemirovski, A.: Robust solutions of linear programming problems contaminated with uncertain data. Mathematical Programming **88**(3), 411–424 (2000)

[11] Bertsimas, D., Pachamanova, D., Sim, M.: Robust linear optimization under general norms. Operations Research Letters **32**(6), 510–516 (2004)

[12] Bertsimas, D., Sim, M.: The price of robustness. Operations Research **52**(1), 35–53 (2004)

[13] Bhowmick, T., Pyrgiotakis, G., Finton, K., Suresh, A., Kane, S., Moudgil, B., Bellare, J.: A study of the effect of JB particles on Saccharomyces cerevisiae (yeast) cells by Raman spectroscopy. Journal of Raman Spectroscopy **39**(12), 1859–1868 (2008)

[14] Birge, J., Louveaux, F.: Introduction to stochastic programming. Springer Verlag (1997)

[15] Bishop, C.: Pattern recognition and machine learning. Springer New York (2006)

[16] Blondin, J., Saad, A.: Metaheuristic techniques for support vector machine model selection. In: Hybrid Intelligent Systems (HIS), 2010 10th International Conference on, pp. 197 –200 (2010)

[17] Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge Univ Pr (2004)

[18] Busygin, S., Prokopyev, O., Pardalos, P.: Biclustering in data mining. Computers & Operations Research **35**(9), 2964–2987 (2008)

[19] Calderbank, R., Jafarpour, S.: Reed Muller Sensing Matrices and the LASSO. Sequences and Their Applications–SETA 2010 pp. 442–463 (2010)

[20] Chandrasekaran, S., Golub, G., Gu, M., Sayed, A.: Parameter estimation in the presence of bounded modeling errors. Signal Processing Letters, IEEE **4**(7), 195–197 (1997)

[21] Chandrasekaran, S., Golub, G., Gu, M., Sayed, A.: Parameter estimation in the presence of bounded data uncertainties. SIAM Journal on Matrix Analysis and Applications **19**(1), 235–252 (1998)

[22] Chang, C.C., Lin, C.C.: LIBSVM: a library for support vector machines (2001). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[23] Della Pietra, S., Della Pietra, V., Lafferty, J., Technol, R., Brook, S.: Inducing features of random fields. IEEE transactions on pattern analysis and machine intelligence **19**(4), 380–393 (1997)

[24] Dinkelbach, W.: On nonlinear fractional programming. Management Science **13**(7), 492–498 (1967)

[25] El Ghaoui, L., Lebret, H.: Robust solutions to least-squares problems with uncertain data. SIAM Journal on Matrix Analysis and Applications **18**, 1035–1064 (1997)

[26] Faddeev, D., Faddeeva, V.: Computational methods of linear algebra. Journal of Mathematical Sciences **15**(5), 531–650 (1981)

[27] Fisher, R.: The use of multiple measurements in taxonomic problems. Annals of Eugenics **7**(7), 179–188 (1936)

[28] Frank, A., Asuncion, A.: UCI machine learning repository (2010). URL http://archive.ics.uci.edu/ml

[29] Guarracino, M., Cifarelli, C., Seref, O., Pardalos, P.: A parallel classification method for genomic and proteomic problems. In: Advanced Information Networking and Applications, 2006., vol. 2, pp. 588–592 (2006)

[30] Guarracino, M., Cifarelli, C., Seref, O., Pardalos, P.: A classification method based on generalized eigenvalue problems. Optimization Methods and Software **22**(1), 73–81 (2007)

[31] Guarracino, M., Irpino, A., Verde, R.: Multiclass generalized eigenvalue proximal Support Vector Machines. In: Proceedings of International Conference on Complex, Intelligent and Software Intensive Systems - CISIS, pp. 25–32. IEEE CS (2010)

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

[33] Horst, R., Pardalos, P., Thoai, N.: Introduction to global optimization. Springer (1995)

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

[35] Hubert, M., Rousseeuw, P., Vanden Branden, K.: Robpca: a new approach to robust principal component analysis. Technometrics **47**(1), 64–79 (2005)

[36] Jain, A., Dubes, R.: Algorithms for clustering data. Prentice Hall (1988)

[37] Karmarkar, N.: A new polynomial-time algorithm for linear programming. Combinatorica **4**(4), 373–395 (1984)

[38] Karush, W.: Minima of functions of several variables with inequalities as side constraints. MSc Thesis, Department of Mathematics. University of Chicago (1939)

[39] Kim, S.J., Boyd, S.: A minimax theorem with applications to machine learning, signal processing, and finance. SIAM Journal on Optimization **19**(3), 1344–1367 (2008)

[40] Kim, S.J., Magnani, A., Boyd, S.: Robust fisher discriminant analysis. Advances in Neural Information Processing Systems **18**, 659 (2006)

[41] Kuhn, H., Tucker, A.: Nonlinear programming. In: Proceedings of the second Berkeley symposium on mathematical statistics and probability, vol. 481, p. 490. California (1951)

[42] Lauer, F., Bloch, G.: Incorporating prior knowledge in support vector machines for classification: A review. Neurocomputing **71**(7-9), 1578–1594 (2008)

[43] Lilis, G., Angelosante, D., Giannakis, G.: Sound Field Reproduction using the Lasso. Audio, Speech, and Language Processing, IEEE Transactions on **18**(8), 1902–1912 (2010)

[44] Mangasarian, O., Street, W., Wolberg, W.: Breast cancer diagnosis and prognosis via linear programming. Operations Research **43**(4), 570–577 (1995)

[45] Mangasarian, O., Wild, E.: Multisurface proximal Support Vector Machine classification via generalized eigenvalues. IEEE Trans. Pattern Anal. Mach. Intell. **28**(1), 69–74 (2006)

[46] McCarthy, J., Minsky, M., Rochester, N., Shannon, C.: A proposal for the Dartmouth summer research project on artificial intelligence. AI MAGAZINE **27**(4), 12 (2006)

[47] Moore, G.: Cramming more components onto integrated circuits. Electronics **38**(8), 114–117 (1965)

[48] Musicant, D.R.: NDC: normally distributed clustered datasets (1998). URL http://www.cs.wisc.edu/dmi/svm/ndc/

[49] Nielsen, J.: Nielsens law of Internet bandwidth. Online at http://www. useit. com/alertbox/980405. html (1998)

[50] Notingher, I., Green, C., Dyer, C., Perkins, E., Hopkins, N., Lindsay, C., Hench, L.: Discrimination between ricin and sulphur mustard toxicity in vitro using Raman spectroscopy. Journal of The Royal Society Interface **1**(1), 79–90 (2004)

[51] Olafsson, S., Li, X., Wu, S.: Operations research and data mining. European Journal of Operational Research **187**(3), 1429 – 1448 (2008)

[52] Owen, C.A., Selvakumaran, J., Notingher, I., Jell, G., Hench, L., Stevens, M.: In vitro toxicology evaluation of pharmaceuticals using Raman micro-spectroscopy. J. Cell. Biochem. **99**(1), 178–186 (2006)

[53] Pardalos, P., Boginski, V., Vazacopoulos, A.: Data Mining in Biomedicine. Springer Verlag (2007)

[54] Platt, J., Cristianini, N., Shawe-Taylor, J.: Large margin DAGs for multiclass classification. Advances in neural information processing systems **12**(3), 547–553 (2000)

[55] Pothin, J., Richard, C.: Incorporating prior information into support vector machines in the form of ellipsoidal knowledge sets. Citeseer (2006)

[56] Powers, K., Brown, S., Krishna, V., Wasdo, S., Moudgil, B., Roberts, S.: Research strategies for safety evaluation of nanomaterials. Part VI. Characterization of nanoscale particles for toxicological evaluation. Toxicological Sciences **90**(2), 296–303 (2006)

[57] Pyrgiotakis, G., Kundakcioglu, O., Finton, K., Pardalos, P., Powers, K., Moudgil, B.: Cell death discrimination with Raman spectroscopy and Support Vector Machines. Annals of Biomedical Engineering **37**(7), 1464–1473 (2009)

[58] Quinlan, J.: C4. 5: programs for machine learning. Morgan Kaufmann (1993)

[59] Quinlan, J.: Improved use of continuous attributes in c4. 5. Journal of Artificial Intelligence Research **4**, 77–90 (1996)

[60] Rao, C.: The utilization of multiple measurements in problems of biological classification. Journal of the Royal Statistical Society. Series B (Methodological) **10**(2), 159–203 (1948)

[61] Rockafellar, R., Uryasev, S.: Optimization of conditional value-at-risk. Journal of risk **2**, 21–42 (2000)

[62] Sch
”olkopf, B., Simard, P., Smola, A., Vapnik, V.: Prior knowledge in support vector kernels. Advances in neural information processing systems pp. 640–646 (1998)

[63] Seref, O., Kundakcioglu, O., Pardalos, P.: Data Mining, Systems Analysis, and Optimization in Biomedicine. In: AIP Conference Proceedings (2007)

[64] Shahbazpanahi, S., Gershman, A., Luo, Z., Wong, K.: Robust adaptive beamforming using worst-case SINR optimization: A new diagonal loading-type solution for general-rank signal models. In: Acoustics, Speech, and Signal Processing, vol. 5, pp. 333–336 (2003)

[65] Shannon, P., Markiel, A., Ozier, O., Baliga, N., Wang, J., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome research **13**(11), 2498 (2003)

[66] Shawe-Taylor, J., Cristianini, N.: Kernel methods for pattern analysis. Cambridge Univ Pr (2004)

[67] Shivaswamy, P., Bhattacharyya, C., Smola, A.: Second order cone programming approaches for handling missing and uncertain data. The Journal of Machine Learning Research **7**, 1283–1314 (2006)

[68] Sion, M.: On general minimax theorems. Pacific Journal of Mathematics **8**(1), 171–176 (1958)

[69] Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., Johannes, R.S.: Using the adap learning algorithm to forecast the onset of diabetes mellitus. Johns Hopkins APL Technical Digest **10**, 262–266 (1988)

[70] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288 (1996)

[71] Trafalis, T., Alwazzi, S.: Robust optimization in support vector machine training with bounded errors. In: Neural Networks, 2003. Proceedings of the International Joint Conference on, vol. 3, pp. 2039–2042

[72] Trafalis, T.B., Gilbert, R.C.: Robust classification and regression using support vector machines. European Journal of Operational Research **173**(3), 893–909 (2006)

[73] Vandenberghe, L.: The CVXOPT linear and quadratic cone program solvers (2010)

[74] Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1995)

[75] Verdú, S., Poor, H.: On minimax robustness: A general approach and applications. Information Theory, IEEE Transactions on **30**(2), 328–340 (2002)

[76] Walter, C.: Kryder's law. Scientific American **293**(2), 32 (2005)

[77] Xanthopoulos, P., Boyko, N., Fan, N., Pardalos, P.: Biclustering: Algorithms and Application in Data Mining. John Wiley & Sons, Inc. (2011)

[78] Xanthopoulos, P., De Asmundis, R., Guarracino, M., Pyrgiotakis, G., Pardalos, P.: Supervised classification methods for mining cell differences as depicted by raman spectroscopy submitted. In: Seventh international meeting on computational intelligence methods for bioinformatics and biostatistics, Palermo September 16-18 2010, pp. 112–122 (2011)

[79] Xanthopoulos, P., Guarracino, M., P.M., P.: Robust generalized eigenvalue classifier with ellipsoidal uncertainty. Annals of Operations Research (submitted)

[80] Xanthopoulos, P., Pardalos, P., Trafalis, T.: Robust Data Mining. Spinger (2012)

[81] Xu, H., Caramanis, C., Mannor, S.: Robustness and Regularization of Support Vector Machines. Journal of Machine Learning Research **10**, 1485–1510 (2009)

[82] Yan, R.: MATLAB Arsenal-A Matlab Package for Classification Algorithms. Informedia, School of Computer Science, Carnegie Mellon University (2006)

BIOGRAPHICAL SKETCH

Petros Xanthopoulos was born at Heraclion, Crete, Greece. He received the diploma of engineering degree from the Electronics and Computer Engineering Department, Technical University of Crete at Chania, Greece, in 2005. In fall 2006, Petros Xanthopoulos joined the graduate program of Industrial and Systems Engineering (I.S.E.) department at the University of Florida. He received a Master of Science in 2008 and a PhD in 2011. His research interest are operations research, data mining, robust optimization and biomedical signal processing. His work has been published in journals of the Instintute of Electrical and Electronics Engineers (IEEE), Elsevier and Wiley & Sons and he has edited a special issue for the Journal of Combinatorial Optimization entitled Data Mining in Biomedicine. He is the co-editor of two volumes published from Springer book series Springer Optimization and its Applications (SOIA) and one published from American Mathematical Society (Fields instintute book series). He has also co-organized many special sessions in international meeting and two conferences related to applications of data mining in biomedicine. While at the University of Florida, he taught several undergraduate classes, and in 2010, he received the I.S.E. department graduate award for excellence in research. He is student member of Instintute of Operations research and Management Science (INFORMS) and Society of Industrial and Applied Mathematics (SIAM). In August 2011 he joined the Industrial Engineering and Management Systems Department, at University of Central Florida as tenure track faculty at the level of assistant professor.