# COMBINATORIAL AND NONLINEAR OPTIMIZATION TECHNIQUES IN PATTERN RECOGNITION WITH APPLICATIONS IN HEALTHCARE

By

O. ERHUN KUNDAKCIOGLU

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

2009

To my wife – Aysan

ACKNOWLEDGMENTS

First, I would like to thank my supervisory committee chair Dr. Panos M. Pardalos for the encouragement, guidance, and opportunities he has provided. I am grateful to Dr. J. Cole Smith, Dr. Joseph Geunes, and Dr. My T. Thai for serving on my committee and their valuable feedback. I am also appreciative for the great contribution from Dr. Onur Seref, Dr. Wilco van den Heuvel, Dr. H. Edwin Romeijn, Dr. Georgios Pyrgiotakis, and Arda Yenipazarli.

I would like to thank my beloved wife Aysan for her caring love, understanding, support, and encouragement. I am also grateful to my mother Reyhan, my father Turgut, and my sister Gözde whose love and support have been invaluable.

I reserve my most special appreciation for my high school math teacher Mehmet Uz. I am forever indebted to him for his enthusiasm, inspiration, and great efforts to explain things clearly and simply.

TABLE OF CONTENTS

LIST OF TABLES

Table page

LIST OF FIGURES

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

COMBINATORIAL AND NONLINEAR OPTIMIZATION TECHNIQUES IN PATTERN RECOGNITION WITH APPLICATIONS IN HEALTHCARE

By

O. Erhun Kundakcioglu

August 2009

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

The first main contribution of this dissertation is the application of standard support vector machine (SVM) classifiers for death cell discrimination. SVMs are a set of machine learning algorithms widely used for classification and regression in data mining, machine vision, and bioinformatics. In this study, Raman spectroscopy is employed to assess the potential toxicity of chemical substances and SVM classifiers successfully assess the potential effect of the test toxins.

The second main contribution is the formulation, complexity result, and an efficient heuristic for Selective SVM classifiers that consider a selection process for both positive and negative classes. Selective SVMs are compared with other standard alignment methods on a neural data set that is used for analyzing the integration of visual and motor cortexes in the primate brain.

The third main contribution of this dissertation is the extension of SVM classifiers for multiple instance (MI) data where a selection process is required for only positive bags. Different formulations, complexity results, and an exact algorithm are presented with computational results on publicly available image annotation and molecular activity prediction data sets. MI pattern recognition methods are then further extended to support vector regression (SVR) and an exact algorithm is presented for the problem. Computational results are presented for a well established breast cancer prognosis data set that is added artificial noise to create synthetic MI regression data.

Finally, two open complexity results on feature selection for consistent biclustering and sparse representation for hyperplane clustering are presented.

# CHAPTER 1
# OPTIMIZATION IN PATTERN RECOGNITION AND HEALTHCARE

## 1.1 Introduction

Medical researchers and practitioners need to obtain information from data sets. This is a difficult task to perform without appropriate tools. The computational requirements for massive data sets have also created many challenging problems for the mathematical community as well as computer scientists. Optimization comes into the picture with most state-of-the-art pattern recognition and machine learning tools. In this chapter, we discuss these tools with applications in healthcare and employed optimization techniques. Furthermore, a survey on optimization applications in treatment planning and other biomedical problems is presented.

Part of this chapter is presented in (Kundakcioglu and Pardalos, 2009b) and (Seref et al., 2008a). Based on (Pyrgiotakis et al., 2009), Chapter 2 presents the application of standard support vector machine (SVM) classifiers for death cell discrimination. Chapter 3 presents Selective SVM classifiers and main findings of this chapter are published in (Seref et al., 2009). Chapters 4 and 5 on multiple instance generalization of support vector techniques are based on (Kundakcioglu et al., 2009b) and (Kundakcioglu et al., 2009a), respectively. One of the two complexity results in Chapter 6 is also presented in (Kundakcioglu and Pardalos, 2009a).

The remainder of this chapter is organized as follows: Section 1.2 presents the most commonly used unsupervised learning techniques. The classification techniques, particularly SVMs are explained in Section 1.3. Section 1.4 presents linear regression problems. In Section 1.5, treatment planning and other optimization applications in biomedical research are mentioned. Finally, Section 1.6 concludes this chapter.

We first explore unsupervised learning techniques and proceed with supervised learning techniques (i.e., classification and regression). *Unsupervised learning* is the case

where objects are not labeled with any class information. Density estimation is performed or clusters of objects are constructed based on similarities between their features.

On the other hand, *supervised learning* refers to the capability of a system to learn from a set of input/output pairs. The input is usually a *vector of features* for an object, and the output is the *label* for the class this object belongs to. The set of objects with a feature vector and a class label is called a *training set*. Based on this information, a function is derived and applied on a *test set*. The output of the *regression* function is a continuous number which is useful to forecast a label that can take any value. In *classification*, on the other hand, the output is a discrete class label that is used for categorical discrimination. The term *supervised* originates from the fact that the labels for the objects in the training set are provided as input, and therefore are determined by an outside source that can be considered as the *supervisor*.

## 1.2 Unsupervised Learning

Unsupervised learning is a machine learning technique where labels of inputs are not available. *Clustering* is partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure. Data clustering is a common technique for statistical data analysis, which is used in many fields including data mining, pattern recognition, image analysis and bioinformatics. The computational task of classifying data set into $k$ clusters is often referred to as $k$-clustering.

One of the most widely used clustering techniques is $k$-means clustering. Underlying optimization problem for $k$-means clustering for a data set of $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ can be formulated as

$$\min \quad J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \mu_k\|^2 \tag{1–1a}$$

$$\text{subject to} \quad r_{nk} \in \{0, 1\} \qquad n = 1, \ldots, N; k = 1, \ldots, K \tag{1–1b}$$

where $\mu_k$ is a prototype associated with the $k^{\text{th}}$ cluster representing the centers of clusters. $r_{nk}$ is the binary indicator variable that is 1 if and only if data point $\mathbf{x}_n$ is assigned to cluster $k$. This is also known as the 1-of-$K$ coding scheme. The objective function $J$ is sometimes called a *distortion measure*.

Formulation (1–1) is usually solved with *k-means algorithm*, which essentially is an expectation-maximization (EM) applied to mixtures of Gaussians. Although the algorithm does not guarantee optimality, its simplicity, efficiency, and reasonable solution quality makes it desirable. The most common form of the $k$-means algorithm uses an iterative refinement heuristic known as Lloyd's algorithm (Lloyd, 1982). EM algorithm consists of successive optimizations with respect to $r_{nk}$ and $\mu_k$. First some initial values of $\mu_k$ are chosen and $J$ is minimized with respect to $r_{nk}$ in the E (expectation) step. In the second phase, which is the M (maximization) step, $J$ is minimized with respect to $\mu_k$ keeping $r_{nk}$ fixed. Since each phase reduces the value of the objective function $J$, the convergence of the algorithm is assured.

$K$-means algorithm has been effectively applied in molecular biology for gene expression data analysis. Functionally related genes are identified by partitioning them into groups based on the similarity between their expression profiles. There are also several extensions of $k$-means that have been successfully applied to other biological problems (see e.g., (Lu et al., 2004)). Lee et al. (2008) present details on clustering applications in genomics. Next, we explore another clustering method that utilize optimization techniques.

*Biclustering* is a data mining technique allowing clustering of data samples and data attributes (features) at the same time. It is sometimes referred to as *co-clustering*, *two-mode* or *two way clustering*. The technique is performed in such a way that each class of features created within the biclustering is related to a class of data samples by a certain property distinguishing it from other classes.

A data set (e.g., from microarray experiments) is normally given as a rectangular $m \times n$ matrix $A$, where each column represents a data sample (e.g., patient) and each row represents a feature (e.g., gene)

$$A = (a_{ij})_{m \times n}$$

where $a_{ij}$ is the expression of $i^{th}$ feature in $j^{th}$ sample.

Biclustering is applied by simultaneous partitioning of the samples and features (i.e., columns and rows of matrix $A$, respectively) into $k$ classes. Let $S_1, S_2, \ldots, S_k$ denote the classes of the samples (columns) and $F_1, F_2, \ldots, F_k$ denote the classes of features (rows). Biclustering can be formally defined as a collection of pairs of sample and feature subsets $\mathcal{B} = \{(S_1, F_1), (S_2, F_2), \ldots, (S_k, F_k)\}$ such that

$$\begin{aligned}
S_1, S_2, \ldots, S_k &\subseteq \{a_{\cdot j}\}_{j=1,\ldots,n}, \\
\bigcup_{r=1}^{k} S_r &= \{a^j\}_{j=1,\ldots,n}, \\
S_v \bigcap S_u &= \emptyset \Leftrightarrow v \neq u, \\
F_1, F_2, \ldots, F_k &\subseteq \{a_{i\cdot}\}_{i=1,\ldots,m}, \\
\bigcup_{r=1}^{k} F_r &= \{a_i\}_{i=1,\ldots,m}, \\
F_v \bigcap F_u &= \emptyset \Leftrightarrow v \neq u,
\end{aligned}$$

where $\{a_{\cdot j}\}_{j=1,\ldots,n}$ and $\{a_{i\cdot}\}_{i=1,\ldots,m}$ denote the set of columns and rows of the matrix $A$, respectively.

A pair $(S_k, F_k)$ is called a *bicluster*. The ultimate goal in a biclustering problem is to find a partitioning for which samples from the same class have *similar* values for that class' characteristic features. The visualization of a reasonable biclustering should reveal a block-diagonal or "checkerboard" pattern as in Fig. 1-1. A detailed survey on biclustering techniques can be found in (Madeira and Oliveira, 2004) and (Busygin et al., 2008).

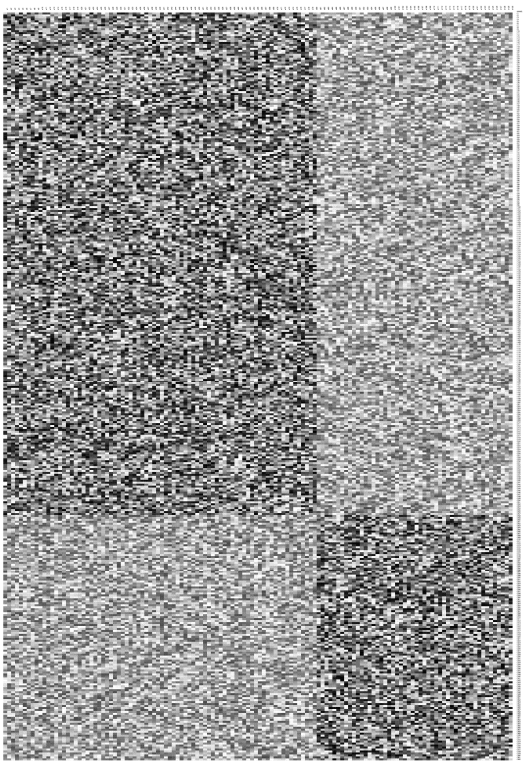


Figure 1-1. An example of checkerboard pattern after biclustering.

The criteria used to relate clusters of samples and features may have different properties. Most commonly, it is required that the submatrix corresponding to a bicluster is either overexpressed (i.e., mostly includes values above average), or has a lower variance than the whole data set. However, biclustering in general may rely on any kind of common patterns among elements of a bicluster.

Divina and Aguilar-Ruiz (2006) address the biclustering of gene expression data with evolutionary computation. Their approach is based on evolutionary algorithms and searches for biclusters following a sequential covering strategy. To avoid overlapping among biclusters, a weight is assigned to each element of the expression matrix. Weights are adjusted each time a bicluster is found. This is different from other methods that substitute covered elements with random values. Experimental results confirm the quality of the proposed method to avoid overlapping among biclusters.

One of the main goals in the analysis of large and heterogeneous gene expression data sets is to identify groups of genes that are co-expressed in subsets of experimental conditions. Identification of these local structures plays a key role in understanding the biological events associated with different physiological states as well as to identify gene expression signatures. Classical one-way clustering techniques, especially hierarchical clustering, have been commonly applied to cluster genes and samples separately in order to identify these types of local patterns. In the last few years, many authors have proposed the use of biclustering algorithms to identify gene-experiment relationships.

bioNMF estimates biclusters using a novel method based on a modified variant of the Non-negative Matrix Factorization algorithm (Pascual-Montano et al., 2006). This algorithm produces a suitable decomposition as a product of three matrices that are constrained to have non-negative elements.

Biclustering is used to analyze one or several of six expression matrices collected from yeast (see (Tanay et al., 2002; Segal et al., 2001)) and to analyze one or more of eleven different expression matrices with human gene expression levels (see (Tanay et al., 2002; Kluger et al., 2003; Busygin et al., 2002)). Almost all these data sets contain expression data related to the study of cancer. Some contain data from cancerous tissues at different stages of the disease; others from individuals suffering from different types of cancer; and the remaining data sets contain data collected from individuals with a particular cancer or healthy people. These data sets are used to test the applicability of biclustering approaches in three major tasks: Identification of coregulated genes, gene functional annotation, and sample classification.

Biclustering techniques are also applied to the problem of identification of coregulated genes (see e.g., (Segal et al., 2001; Ben-Dor et al., 2002)). More specifically, the objective is to identify sets of genes that, under specific conditions, exhibit coherent activations that indicate coregulation. These results are used to simply identify sets of coregulated genes or, more ambitiously, to identify specific regulation processes. A less obvious application

is to use the biclustering results directly to perform automatic gene functional annotation (see (Tanay et al., 2002; Segal et al., 2001)). The idea underlying this approach is to use biclusters where a large majority of genes belong to a specific class in the gene ontology to guess the class of nonannotated genes.

Another significant area of application is related with sample and/or tissue classification (Busygin et al., 2002; Tanay et al., 2002; Kluger et al., 2003). An example is the diagnosis of leukemia where the goal is to identify different responses to treatment and group of genes to be used as the most effective probe (Sheng et al., 2003).

The applications of biclustering mentioned above analyze data from gene expression matrices. However, biclustering can also be used in the analysis of other biological data. Liu and Wang (2003) apply biclustering to a drug activity data set. The goal in this study is to find groups of chemical compounds with similar behaviors when subsets of compound descriptors are taken into account. Lazzeroni and Owen (2002) analyze nutritional data to identify subsets of foods with similar properties on a subset of food attributes.

In (Genkin et al., 2002), it is shown how several problems in different areas of data mining and knowledge discovery can be viewed as finding the optimal covering of a finite set. Many such problems arise in biomedical and bioinformatics research. For example, protein functional annotation based on sequence information is an ubiquitous bioinformatics problem. It consists of finding a set of homolog (high similarity) sequences of known function to a given amino acid sequence of unknown function from the various annotated sequence databases. These can then be used as clues in suggesting further experimental analysis of the new protein. Genkin et al. (2002) show these optimization problems can be stated as maximization of submodular functions on the set of candidate subsets. This generalization may be especially useful when conclusions from data mining need to be interpreted by human experts as in diagnostic hypothesis generation, logical methods of data analysis, conceptual clustering, and proteins functional annotations.

Geva and Kerem (1998) present a novel electroencephalogram (EEG)-based, brain-state identification method, which could form the basis for forecasting a generalized epileptic seizure. 25 rats are exposed to hyperbaric oxygen until the appearance of a generalized EEG seizure. EEG segments from the preexposure, early exposure, and the period up to and including the seizure are processed by the fast wavelet transform. Features extracted from the wavelet coefficients are input to the unsupervised optimal fuzzy clustering (UOFC) algorithm. The UOFC is useful for classifying similar discontinuous temporal patterns in the semistationary EEG to a set of clusters which may represent brain-states. The unsupervised selection of the number of clusters overcomes the a priori unknown and variable number of states. The usually vague brain state transitions are naturally treated by assigning each temporal pattern to one or more fuzzy clusters.

Next, we start supervised learning techniques within classification framework.

## 1.3 Linear Classification

Given a training data set comprising the features and the corresponding labels, the goal in classification is to take an input vector $\mathbf{x}$ and assign to one of $K$ discrete classes $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_K$. The input space is divided into decision regions whose boundaries are called the *decision boundaries*. These regions are formed using the *discriminant functions*.

In a typical *binary classification* problem, each pattern vector $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \ldots, l$ belongs to one of two classes $\mathbf{S}^+$ and $\mathbf{S}^-$. A vector is given the label $y_i = 1$ if $\mathbf{x_i} \in \mathbf{S}^+$ or $y_i = -1$ if $\mathbf{x}_i \in \mathbf{S}^-$. The set of pattern vectors and their corresponding labels constitute the *training set*. The classification problem consists of determining which class new pattern vectors from the *test set* belong to.

### 1.3.1 Support Vector Machine Classifiers

In this section, we discuss Support Vector Machine (SVM) classifiers. Developed by Vapnik (1995), SVMs are the state-of-the-art supervised machine learning methods. SVM classifiers classify pattern vectors which are assumed to belong to two linearly separable sets from two different classes. Although there are infinitely many hyperplanes that

separate the two classes, an SVM classifier finds the hyperplane $(\boldsymbol{\psi}, b)$ that maximizes the distance from the convex hulls of both classes by solving a quadratic convex optimization problem. The success and robustness of SVM classifiers rely on strong fundamentals from the statistical learning theory, from which generalization bounds are derived.

SVMs solve binary classification problem by finding a hyperplane $(\boldsymbol{\psi}, b)$ that separates the two classes in the training set from each other with the maximum margin.

The underlying optimization problem for the maximal margin classifier is only feasible if the two classes of pattern vectors are linearly separable. However, most of the real life classification problems are not linearly separable. Nevertheless, the maximal margin classifier encompasses the fundamental methods used in standard SVM classifiers. The solution to the optimization problem in the maximal margin classifier minimizes the bound on the generalization error (Vapnik, 1998). The basic premise of this method lies in the minimization of a convex optimization problem with linear inequality constraints, which can be solved efficiently by many alternative methods (Bennet and Campbell, 2000).

A hyperplane can be represented by $\langle \boldsymbol{\psi}, \mathbf{x} \rangle + b = 0$, where $\boldsymbol{\psi}$ is the $n$-dimensional normal vector and $b$ is the offset parameter. There is an inherent degree of freedom in specifying a hyperplane as $(\lambda\boldsymbol{\psi}, \lambda b)$. A *canonical hyperplane* is the one from which the closest pattern vector has a distance $1/\|\boldsymbol{\psi}\|$, i.e., $\min_{i=1,\ldots,m} |\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b| = 1$.

Consider two pattern vectors $\mathbf{x}^+$ and $\mathbf{x}^-$ belonging to classes $\mathbf{S}^+$ and $\mathbf{S}^-$, respectively. Assuming these pattern vectors are the closest to a canonical hyperplane, such that $\langle \boldsymbol{\psi}, \mathbf{x}^+ \rangle + b = 1$ and $\langle \boldsymbol{\psi}, \mathbf{x}^- \rangle + b = -1$, it is easy to show that the geometric margin between these pattern vectors and the hyperplane are both equal to $1/\|\boldsymbol{\psi}\|$. Maximizing the geometric interclass margin while satisfying the canonical separating hyperplane condition for the pattern vectors results in the following optimization problem:

$$\min \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 \tag{1–2a}$$

$$\text{subject to} \quad y_i(\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) \geq 1 \qquad i = 1, \ldots, l \tag{1–2b}$$

From the solution to 1–2, a new pattern vector $\mathbf{x}^*$ can be classified as positive if $\langle \boldsymbol{\psi}, \mathbf{x}^* \rangle + b > 0$, and negative otherwise.

Most real life problems are composed of nonseparable data which is generally due to noise. In this case, *slack variables* $\xi_i$ are introduced for each pattern vector $\mathbf{x}_i$ in the training set. The slack variables allow misclassifications for each pattern vector with a penalty of $C/2$. In Fig. 1-3, *soft margin classifier* is demonstrated that incurs penalty for misclassified pattern vectors.

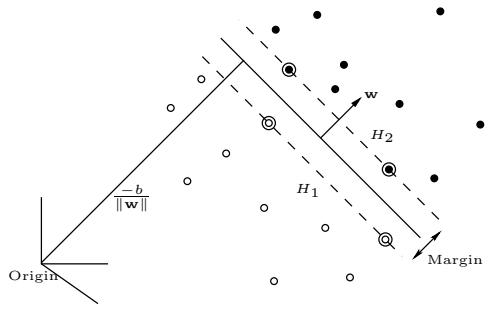



Figure 1-2. Maximal Margin Classifier

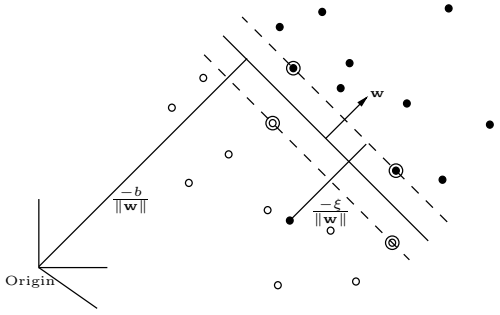



Figure 1-3. Soft Margin Classifier

The maximum margin formulation can be augmented to soft margin formulation as follows.

$$\min \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 \tag{1–3a}$$

$$\text{subject to} \quad y_i(\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i \qquad i = 1, \ldots, l \tag{1–3b}$$

In (1–3), it is unnecessary to enforce nonnegativity of the slack variables explicitly since a solution cannot be optimal when $\xi_i < 0$ for any pattern vector. It should be noted that the 2-norm of the slack variables are penalized in the objective of (1–3). An alternative formulation involves penalization of the 1-norm of slack variables. In this case, nonnegativity constraints on the slack variables are necessary as follows:

$$\min \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + C\sum_{i=1}^{l}\xi_i \tag{1–4a}$$

$$\text{subject to} \quad y_i(\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i \qquad i = 1, \ldots, l \tag{1–4b}$$

$$\xi_i \geq 0 \qquad i = 1, \ldots, l \tag{1–4c}$$

It should also be noted that (1–3) and (1–4) are essentially minimization of a convex functions with linear inequality constraints. These problems can be solved efficiently by numerous methods (see (Bennet and Campbell, 2000)). See (Cristianini and Shawe-Taylor, 2000) for further details on formulation and implementation details of SVMs.

Platt (1999) solves SVM problems by iteratively selecting subsets only of size 2 and optimizing the target function with respect to them. This technique is called the Sequential Minimal Optimization (SMO). It has good convergence properties and it is easily implemented. The key point is that for a working set of 2, the optimization subproblem can be solved analytically without explicitly invoking a quadratic optimizer.

*Dual* formulations for both 1-norm and 2-norm SVM classifiers can be obtained using the optimization theory. The significance of the dual formulations is that, they do not involve inequality constraints and allow the *kernel trick* to be introduced for nonlinear

classification. The standard method to obtain the dual formulation for the SVM problem consists of two parts. First the *Lagrangian function* of the primal problem if derived. This function provides a lower bound for the solution of the primal problem. Next, the Lagrangian function is differentiated with respect to the primal variables and stationarity is imposed. Equivalent expressions for each primal variable are substituted back in the Lagrangian function and added as constraints where appropriate. The dual problem is obtained by maximizing the resulting function with the new constraints. The dual problem is a concave maximization problem, which can also be solved efficiently. The Lagrangian function for the 2-norm SVM primal problem is given as follows.

$$L(\boldsymbol{\psi}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 - \sum_{i=1}^{n}\alpha_i\left[y_i(\langle\boldsymbol{\psi}, \mathbf{x_i}\rangle + b) - 1 + \xi_i\right] \tag{1–5}$$

Differentiating $L$ with respect to the primal variables $\boldsymbol{\psi}$ and $b$, and assuming stationarity,

$$\frac{\partial L}{\partial \boldsymbol{\psi}} = \boldsymbol{\psi} - \sum_{i=1}^{n} y_i\alpha_i\mathbf{x}_i = 0 \tag{1–6}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} y_i\alpha_i = 0 \tag{1–7}$$

$$\frac{\partial L}{\partial \xi_i} = C\xi_i - \alpha_i = 0 \tag{1–8}$$

Substituting the expressions back in the Lagrangian function, the following dual formulation is obtained which realizes the hyperplane $\boldsymbol{\psi}^* = \sum_{i=1}^{n} y_i\alpha_i^*\mathbf{x}_i$ with geometric margin $\gamma = 1/\|\boldsymbol{\psi}^*\|$.

$$\max \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_iy_j\alpha_i\alpha_j\langle\mathbf{x}_i, \mathbf{x}_j\rangle - \frac{1}{2C}\sum_{i=1}^{n}\alpha_i^2 \tag{1–9a}$$

$$\text{subject to} \quad \sum_{i=1}^{n} y_i\alpha_i = 0 \tag{1–9b}$$

$$\alpha_i \geq 0 \qquad\qquad i = 1, \ldots, n \tag{1–9c}$$

Note that from Karush-Kuhn-Tucker complementarity conditions, the constraints in the primal problem are binding for those with the corresponding dual variable $\alpha_i^* > 0$. The

decision rule $sgn(f(\mathbf{x}))$ is equivalent to the hyperplane $f(\mathbf{x}) = \sum_{i=1}^{n} y_i \alpha_i^* \langle \mathbf{x}, \mathbf{x}_i \rangle + b^*$ and $b^*$ is chosen such that $y_i f(\mathbf{x}_i) = 1 - \alpha_i^*/C$ for those pattern vectors with $\alpha_i^* > 0$. Using (1–9b), $b^*$ can be calculated as

$$b^* = \sum_{i:\alpha_i^*>0} y_i - \langle \boldsymbol{\psi}^*, \mathbf{x}_i \rangle. \tag{1–10}$$

Alternatively, $b^*$ can be calculated using

$$\begin{aligned} y_i[\langle \boldsymbol{\psi}^*, \mathbf{x}_i \rangle + b^*] &= 1 - \xi_i^* && i : \alpha_i^* > 0 \\ \langle \boldsymbol{\psi}^*, \mathbf{x}_i \rangle + b^* &= y_i(1 - \frac{\alpha_i^*}{C}) && i : \alpha_i^* > 0 \\ b^* &= y_i(1 - \frac{\alpha_i^*}{C}) - \sum_{j=1}^{l} y_j \alpha_j^* \langle \mathbf{x}_j, \mathbf{x}_i \rangle && i : \alpha_i^* > 0 \end{aligned} \tag{1–11}$$

The derivation for the 1-norm dual formulation is very similar to that of 2-norm. The Lagrangian function for the 1-norm SVM classification problem is given as follows.

$$L(\boldsymbol{\psi}, b, \xi, \alpha, \mathbf{r}) = \frac{1}{2}\|\boldsymbol{\psi}\|^2 + C\sum_{i=1}^{l} \xi_i - \sum_{i=1}^{l} \alpha_i \left[ y_i(\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) - 1 + \xi_i \right] - \sum_{i=1}^{l} r_i \xi_i$$

Differentiating $L$ with respect to the primal variables and using stationarity, we obtain

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{\psi}} &= \boldsymbol{\psi} - \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i = 0 \\ \frac{\partial L}{\partial b} &= \sum_{i=1}^{l} y_i \alpha_i = 0 \\ \frac{\partial L}{\partial \xi_i} &= C - \alpha_i - r_i = 0. \end{aligned}$$

The expressions are substituted back in the Lagrangian function and the following dual formulation is obtained.

$$\max \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{1–12a}$$

$$\text{subject to} \quad \sum_{i=1}^{l} y_i \alpha_i = 0 \tag{1–12b}$$

$$0 \leq \alpha_i \leq C \qquad i = 1, \ldots, l \tag{1–12c}$$

Kernel trick can be applied and the 1-norm formulation becomes,

$$\max \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{1–13a}$$

$$\text{subject to} \quad \sum_{i=1}^{l} y_i \alpha_i = 0 \tag{1–13b}$$

$$0 \leq \alpha_i \leq C \qquad i = 1, \ldots, l \tag{1–13c}$$

This problem is equivalent to the maximal margin hyperplane, with the additional constraint that all the $\alpha_i$ are upper bounded by $C$. This gives rise to the *box constraints* that is frequently used to refer to this formulation, since the vector $\alpha$ is constrained to lie inside the box with side length $C$ in the positive orthant. The trade-off parameter between accuracy and regularization directly controls the size of the $\alpha_i$. That is, the box constraints limit the influence of outliers, which would otherwise have large Lagrange multipliers. The constraint also ensures that the feasible region is bounded and hence the primal always has a non-empty feasible region.

Note that Karush-Kuhn-Tucker complementarity conditions can be used to obtain $b^*$ similar to the 2-norm case. However, in 1-norm case we look for both constraints to be binding, i.e., $\alpha_i^* > 0, r_i^* > 0$.

$$
\begin{aligned}
y_i(\langle \boldsymbol{\psi}^*, \mathbf{x}_i \rangle + b^*) &= 1 && i : C > \alpha_i^* > 0 \\
\langle \boldsymbol{\psi}^*, \mathbf{x}_i \rangle + b^* &= y_i && i : \alpha_i^* > 0 \\
b^* &= y_i - \sum_{j=1}^{l} y_j \alpha_j^* \langle \mathbf{x}_j, \mathbf{x}_i \rangle && i : \alpha_i^* > 0
\end{aligned} \tag{1–14}
$$

The decision rule $sgn(f(\mathbf{x}))$ is equivalent to the hyperplane $f(\mathbf{x}) = \sum_{i=1}^{n} y_i \alpha_i^* \langle \mathbf{x}, \mathbf{x}_i \rangle + b^*$ and $b^*$ can also be calculated using $y_i f(\mathbf{x}_i) = 1$ for those pattern vectors with $0 < \alpha_i^* < C$.

Kernels are introduced in classification to provide enhanced similarity measures between pattern vectors. They basically transform the so-called *input space*, $\mathcal{X}$, in which the original pattern vectors reside, to a usually higher dimensional dot-product space $\mathcal{H}$ called the *feature space*, via a map $\mathbf{\Phi} : \mathcal{X} \rightarrow \mathcal{H}$, such that $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) \rangle$. The kernel $\mathbf{K}$ is required to be positive semidefinite in order to define a dot product space and create a feature map. A positive semidefinite kernel is defined as a function on $\mathcal{X} \times \mathcal{X}$ for a nonempty set $\mathcal{X}$, which for all $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ gives rise to a positive semidefinite matrix $\mathbf{K}$ such that $\sum_{i,j} c_i c_j \mathbf{K}_{ij} \geq 0$ for all $c_i \in \mathbb{R}$. In the literature it was shown that any algorithm that works on dot products can be kernelized through the *kernel trick* (Schölkopf and Smola, 2002). SVMs provide nonlinear classification by replacing the linear dot product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ in the dual formulation with an appropriate nonlinear kernel $\mathbf{K}$.

In the machine learning literature, the kernel trick is introduced by Mercer's theorem and explains the geometry of feature spaces (see (Cristianini and Shawe-Taylor, 2000)). It can be considered as the characterization of a kernel $\mathbf{K}(\mathbf{x}, \mathbf{x}^*)$. The conditions for Mercer's theorem are equivalent to the requirement that the corresponding matrix is positive semidefinite for any finite subset of $\mathcal{X}$.

One of the most commonly used kernels both theoretically and practically is the *Gaussian kernel* which is represented as,

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}} \tag{1–15}$$

where, $\sigma$ is referred to as the bandwidth. Smaller bandwidths are better in classifying intricate patterns, but worse in generalization.

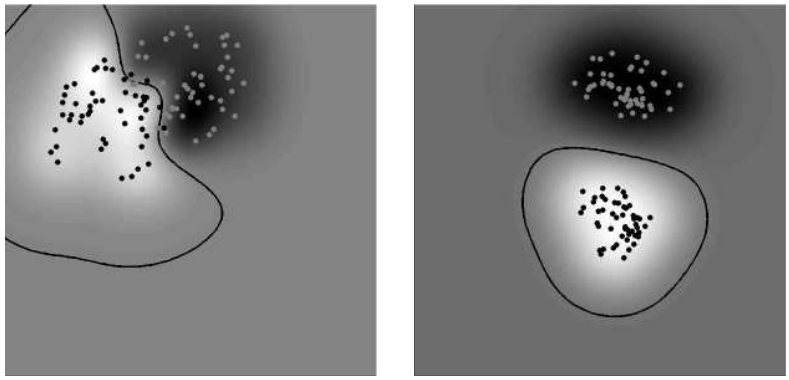


Figure 1-4. Examples of nonlinear classification using SVM with Gaussian Kernel.

### 1.3.2 Applications in Neuroscience

Although the introduction of SVMs is relatively recent, this classification technique is gaining rapid acceptance as a robust neuroscience and medical research tool. SVMs have been shown to be highly effective at detecting complex pattern changes in neuroimaging data. Often patterns that are difficult, time-consuming, or perhaps impossible to detect by the trained physician's eye can be detected rapidly and accurately using SVMs. In this manner, SVMs can assist doctors with the challenging task of performing rapid diagnosis of neurological diseases and disorders. Expediting diagnosis means expediting proper treatment, which may mitigate or even prevent permanent damage to the central nervous system.

Many of the medical SVM applications focus on image processing of magnetic resonance imaging (MRI) data to detect structural alterations in the brain over time. Magnetic resonance spectroscopy (MRS) data has also been analyzed with SVMs for a

similar purpose. Single photon emission computed tomography (SPECT) data can be used to detect blood flow pattern changes over time. Often these neurophysiologic measurement modalities can be utilized in an SVM classifier to detect the presence of a disease or disorder.

Numerous neuroscience studies have utilized SVMs to classify neural states. fMRI is a useful imaging modality for neural state classification due to its ability to track changes in blood oxygenation level dependent signal, which is correlated with blood flow. In addition, the electroencephalogram (EEG) is a highly useful measure for SVM neural state classification due to its ability to quantify brain electrical activity (e.g., voltage difference between a region of interest and a reference region on the scalp or in the brain) with exceptionally high temporal resolution.

The remainder of this chapter will provide an overview of the state-of-the-art SVM application to data from various neuroimaging modalities for the purposes of medical diagnosis, understanding the physiology of cognition, and classification of neural states.

#### 1.3.2.1 Magnetic resonance imaging

There are a number of studies which introduce different methods for image *segmentation*. Segmentation is the process of partitioning a digital image into different sections in order to change the representation of the image. This new representation may involve certain characteristics in the image such as curves, edges, color, intensity or texture. Segmented images are usually used to determine brain abnormalities using image classification, and it is shown that SVMs perform very well. Lee et al. (2005) used SVMs and a new SVM based method developed earlier called *support vector random fields* for segmenting brain tumors from MR images. Rinaldi et al. (2006) classified brain inflammation in *multiple sclerosis* patients based on the peripheral immune abnormalities from MR images using nonlinear SVMs. They determined that brain inflammation in patients with multiple sclerosis is associated with changes in subsets of peripheral lymphocytes. Thus, SVM classification helped detect a potential biomarker candidate for

the prognosis of patients in the early stages of multiple sclerosis. Quddus et al. (2005) combined SVMs and *boosting* (Schapire, 2001), another machine learning method, to perform segmentation (via nonlinear classification) on white matter lesions in the MRI scans. Their composite classification method was shown to be faster and just as reliable as manual selection. In another study by Martinez-Ramon et al. (2006), a similar approach was used to create segments of the brain with respect to their functions. Later, these segments were aggregated using boosting, which is used for multi-class SVM classification of an fMRI group study with interleaved motor, visual, auditory, and cognitive task design.

#### 1.3.2.2 Other imagery types

Other imaging modalities such as ultrasound and MRS are also used in medical classification problems. Kotropoulos and Pitas (2003) used SVMs for segmentation of ultrasonic images acquired near lesions in order to differentiate between lesions and background tissue. The radial basis function SVMs outperformed the process of thresholding of L2 mean filtered images for various lesions under numerous recording conditions. Darbellay et al. (2004) used SVM and other classification techniques to detect solid or gaseous *emboli* by transcranial Doppler (TCD) ultrasound. Since the leading cause of cerebral infarction is due to the extracranial atherosclerosis, rapid assessment of the physical characteristics of solid objects in the blood flow is important. Darbellay et. al. demonstrated that SVMs could distinguish between solid and gaseous embolisms from ultrasonic measures of the blood stream. A medical diagnostic device based on this technology may be able to prevent brain damage by allowing a means to expedite the diagnosis and treatment of embolisms. Devos et al. (2005) devised a system that can automatically discriminate brain tumors based on data from MRI and *MRSI*, which is a function of MR imaging that produces a spectroscopic profile of the scanned brain region. In this study, MR spectra from MRSI was used for comparison with linear and nonlinear *Least squares SVM* (LS-SVM) (Suykens and Vandewalle, 1999). A similar study was

carried out based on only MR spectra with *short echo time H MRS*, using the same set of classifiers (Devos et al., 2004). Lukas et al. (2004), on the contrary used *long echo H MRS* signals to classify brain tumors into four classes: *meningiomas*, *gliobastomas*, *astrocytomas grade II* and *metastases*, with a number of classifiers including standard SVM, and LS-SVM. The study demonstrated that kernel based SVMs were able to detect tumors without utilizing dimensionality reduction and still produce accuracy comparable to linear discriminant analysis. Automated tumor detection algorithms are a sought-after tool for assisting physicians to make more accurate and rapid detection of tumors. Support vector machine classifiers have contributed significantly in this area. Menze et. al. utilized SVM classification of MRI images to serve as an automated diagnostic tool for the detection of recurrent brain tumors (Menze et al., 2006). They report that SVM among other methods was able rule out lipid and lactate signals as being too unreliable, and that choline and N-acetylaspartate are the main sources of information (most important features). Kelm et al. (2007) performed an evaluation of numerous automated prostate tumor detection methods. Their study determined that the pattern recognition methods, such as SVM classification, were able to outperform quantization methods such as QUEST, AMARES, and VARPRO for prostate tumor detection.

Rapid diagnosis of stroke in patients is desirable as punctual treatment can reduce the chance of permanent brain damage. One potential method for rapidly diagnosing stroke is to examine the contents of a potential stroke patients blood for biomarkers indicative of a stroke. Prados et al. (2004) utilized support vector machines to help identify 14 potential biomarkers which could be used to distinguish the chemical profile of a control subject's blood from the chemical profile of an ischemic or hemorrhagic stroke patient. Surface enhanced laser desorption/ionization *mass spectometry* is used with SVMs for feature selection to find a small subset of potential biomarkers for early stroke diagnosis. Some images used do not directly come from MR scanning of the brain. Glotsos et al. (2005a) and Glotsos et al. (2005b) used digitized images of *biopsies* of *astrocytomas* to detect brain

tumors with an *unsupervised SVM clustering* method in conjunction with a decision-tree based multiple class classification in order to classify tumors. Using morphological brain data and high-dimensional shape transformations, SVMs are also shown by Lao et al. (2004) to classify between male and female brain and age differentiation for old adults. Although brain images are the primary sources for detecting brain abnormalities, electrical brain signals can also be used, such as in (Lehmann et al., 2007). They compared a number of classification methods including SVMs for the detection Alzheimer's disease from the EEG recordings and discovered that the SVMs performance was superior to other methods.

#### 1.3.2.3 Feature selection

Though it may seem counterintuitive, classification of large volumes of neurophysiologic data is not always very informative. A small portion of the data is usually sufficient for classification and the remaining data hinders the performance of the classifiers. *Feature selection* algorithms are developed to isolate such features which significantly contribute to the classification of the data. Considering very high dimensional brain images or multiple-channel brain recordings, feature selection plays a very important role in classification of neural data. Fan et al. (2007) introduced a method for classification of *schizophrenia* patients and healthy controls from brain structures whose volumetric features are extracted from processed MR images. The best set of such features are determined using an SVM-based feature selection algorithm, which in return significantly improved the classification performance. Yoon et al. (2007) extracted principal components derived from cortical thickness to differentiate between healthy controls and schizophrenic patients using SVMs for use as a diagnostic tool. Yushkevich et al. (2005) investigated the effect of abnormal development and brain structure in patients with schizophrenia with respect to the morphological characteristics and age related changes. They used deformed brain templates of a variety of subject images and used SVMs for classification and feature selection to classify between pathological cases from the healthy controls. A similar study

was carried out by Liu et al. (2004) for automated schizophrenia and Alzheimer's disease detection. Fung and Stoeckel (2007) also used an SVM feature selection algorithm applied to *SPECT* perfusion imaging to detect Alzheimer's disease. They used a *1-norm* linear SVM classifier, which is known to give sparse solutions, which in turn is used for feature selection.

Diabetes mellitus (DM) is a common disease in the industrialized countries and it is a prominent risk factor for ischemic cerebrovascular accidents. Diabetes alone is responsible for 7% of deaths in stroke patients. Diabetes mellitus often results in brain micro-blood flow disorders that may cause cerebral infarction. However, assessing the function of cerebral micro-vessels is difficult, since they are located within the bony skull. Kalatzis et al. (2003) performed a study where SVM was applied to distinguish between blood flow data in patients with diabetes versus control subjects using *SPECT* images from cerebral abnormalities. Li et al. (2006b) used SVMs with *floating search method* to find relevant features for assessing the degree of malignancy in brain glioma from MRI findings and clinical data prior to operations. Li et al. (2006a) further developed a novel algorithm that combines bagging of SVMs with embedded feature selection for the individual observations and compared the new algorithm using publicly available datasets.

#### 1.3.2.4 Brain computer interface

BCI is an extremely challenging research area which could revolutionize available treatment options for patients with paralysis. The main function of a brain computer interface is to detect brain activity patterns associated with the intention to execute some task which the paralyzed subject cannot physically perform. The BCI responds to the detected intentions by activating an attached prosthetic device which can enact the task for the subject.

BCI devices typically utilize neurophysiologic measures which can be acquired for extended durations and with high time time resolution. Though fMRI can provide highly useful information about the temporal hemodynamic response to changes in

brain activity, BCI research using electrographic signals has the significant advantage of relatively simple acquisition hardware and superior time resolution. The state of the art of electrographic acquisition systems give them potential for the creation of implantable BCI devices. One of numerous challenges that stands in the way of enacting such a device is the ability to classify the brain measurement patterns in real time while maintaining acceptable sensitivity and specificity. Of the many methods utilized for this task, SVMs have demonstrated a large degree of success.

The majority of the applications focus on prosthetics for patients suffering from conditions such as amyotrophic lateral sclerosis (ALS), brainstem stroke, and brain injury. Guigue et al. (2006) developed a new graph based method to classify non-stationary signals, each with a discriminant waveform with random time and location. The graph based representation was used to define an inner product between graphs to be used with SVMs, which increased the accuracy of the BCI system.

Many studies have utilized the P300 evoked potential as an SVM input for classifying text which is read by the user (see (Thulasidas et al., 2006)). The P300 evoked potential is an event-related electrical potential which appears approximately 300 ms after an infrequent event is perceived. A P300 spelling device could provide a means of communication for disabled individuals who would otherwise be unable to communicate with the world. This technique is frequently used to assess the performance of BCI related methods. Kaper and Ritter (2004) and Kaper et al. (2004) used SVMs on EEG recordings from the P300 speller BCI paradigm to reach high rates of data transfer and generalization. In these studies the subjects were given a 6 by 6 matrix with flashing symbols and were instructed to attend to only one symbol to count how many times it appears. The SVM classifier was used to detect this P300 component in the EEG, and was shown to perform with high accuracy. Guan et al. (2005) used a similar mental speller paradigm with a target and non-target symbols moving from right to left in a small window on a computer screen and detected significant differences in the EEG using SVMs.

Since BCI systems need to be computationally efficient, a robust feature selection algorithm is required for prediction of the EEG channels that may be involved in the cognitive task of interest. Lal et al. (2004) investigated the feature selection and EEG signal classification problems with the SVM-based *Recursive Feature Elimination* and *Zero-Norm Optimization* methods.

Although nonlinear methods can provide better results, linear methods may be preferred wherever possible. However, complex cases still require efficient methods in BCI which can handle nonlinear classification such as SVMs, as it was shown in (Müller et al., 2003). Garrett et al. (2003) also used SVMs to classify EEG signals from a well-known EEG data set (see (Keirn and Aunon, 1990)), which involve five different mental tasks, and showed that linear and nonlinear methods may perform similarly. Liang et al. (2006) used the Extreme Learning Machine (ELM) algorithm to classify EEG signals from the same data set and showed that ELM has similar performance to SVMs.

Some BCI systems are developed using non-human subjects. Rats are the most common subjects for this kind of research. A BCI system adapted for rats was developed by Hu et al. (2005), who showed that SVM classifiers and principal component analysis combined with a Bayesian classifier may perform equally well for classification. They also showed SVM classification of neuronal spike trains allow identification of individual neurons associated with the decision making process. Jakuczun et al. (2005) applied SVMs to classify habituated from aroused states using evoked potentials from a single barrel column of the rat's somatosensory cortex. Olson et al. (2005) used spike trains from rats to predict left and right hand commands in a binary paddle pressing task performed by rats.

Optical measurement methods have also demonstrated success in SVM BCI systems. Sitaram et al. (2007) used near infrared spectroscopy to detect oxygenation in the left hand versus right hand motor imagery of human subjects from a 20-channel NIRS system.

They used SVMs and hidden markov models to assess the feasibility of NIRS to be used as BCI and concluded that NIRS can be a significant alternative to EEG in BCI.

A study by Acir and Güzelis (2005) investigated the utility of SVMs for identifying EEG sleep spindles, an EEG pattern found in stage 2 of sleep. The study demonstrated that radial basis SVMs detected EEG sleep spindles with high accuracy. This application of SVMs may be useful in an automated sleep staging algorithm.

Epilepsy is the condition of recurrent seizures. Over the past few decades, the area of seizure detection and seizure prediction using quantitative EEG analysis has drawn great interest. Chaovalitwongse et al. (2006) developed a seizure prediction algorithm using SVMs which was able to successfully classify between EEG patterns associated with an interictal ("normal") brain state and EEG patterns associated with a pre-ictal ("seizure prone") state. Such an algorithm could be developed to become the basis for a bedside or implantable seizure control device.

#### 1.3.2.5 Cognitive prediction

Prediction of cognitive states plays an important role in enhancing our understanding of the physiologic mechanisms underlying these brain functions. Brouwer and van Ee (2007) used SVMs on functional fMRI data to predict the visual perceptual states from the retinotopic visual cortex and motion-sensitive areas in the brain. Cox and Savoy (2003) investigated visual presentation of various categories of objects. They used SVMs to classify the images based on similarity from predetermined regions of voxels (volume elements) over a short period of time. This method was shown to produce similar results using much less data than traditional fMRI data analysis, which requires numerous hours of data across many subjects. Pessoa and Padmala (2007) also used fMRI images to predict perceptual states. SVMs are used to detect near-threshold fear detection, and concluded that multiple regions of the brain are involved and that behavioral choice is distributed across these regions to help manage the emotional stimuli and prepare the appropriate response. Shoker et al. (2005) introduced a hybrid algorithm which combines

SVMs with blind source separation (BSS) and directed transfer functions (DTF), which are used to extract features from EEG recordings to distinguish between left and right finger movements. Seref et al. (2007) used intracranial local field potential recordings from macaque monkeys and developed a selective SVM-based classification method in conjunction with SVM-based feature selection methods to detect categorical cognitive differences in visual stimuli based on single-trials from a visuomotor task.

#### 1.3.2.6 Other modeling techniques

Faugeras et al. (2004) studied brain anatomy and modeled brain function from MR images. SVMs combined with methods from information theory are used in clustering of voxels in the statistical modeling of the fMRI signals. LaConte et al. (2005) used SVMs in block design fMRI and compared them to canonical variance analysis (CVA). Mourão-Miranda et al. (2006) investigated the performance of SVMs with time compression on single and multiple subjects, and showed that the time compression of the fMRI data improves the classification performance. In a similar study, Mourão-Miranda et al. (2007) introduced time series embedding into the classification framework. In this work spatial and temporal information was combined to classify different brain states in cognitive tasks in patients and healthy control subjects. In a study by Wang et al. (2003), a nonlinear framework for fMRI data analysis is introduced, which uses spatial and temporal information to perform *support vector regression* in order to find the spatio-temporal autocorrelations in the fMRI data. Finally, Parra et al. (2005) present an array of methods as "recipes" for linear analysis of EEG signals, among which performance of SVMs is compared with logistic regression.

See (Seref et al., 2008a) for a detailed survey on applications of SVM in neuroscience and (Lee et al., 2008) for classification applications in genomics. Next, we discuss some generalizations of the linear classification problem.

### 1.3.3 SVM Extensions and Generalizations

The multiple instance (MI) learning is introduced by Dietterich et al. (1997) in the context of drug activity prediction and developed in (Auer, 1997; Long and Tan, 1998). The problem consists of classifying positive and negative *bags* of points in the $n$-dimensional real space $\mathbb{R}^n$ where each bag contains a number of points. Patterns $\mathbf{x}_1, \ldots \mathbf{x}_l$ are grouped into bags $X_1, \ldots X_m$ with $X_j = \{\mathbf{x}_i : i \in I_j\}$, $I_j \subseteq \{1, \ldots, n\}$, and $\bigcup_j I_j = \{1, \ldots, n\}$. Each bag $X_j$ is associated with a label $y_j \in \{1, -1\}$. Classification is performed such that at least one point for each positive bag is classified as positive, and all the points for all negative bags are classified as negative. In (Dietterich et al., 1997), a hypothesis class of axis-parallel rectangles are assumed, and algorithms are developed to deal with the drug activity prediction problem. An efficient algorithm is described in (Long and Tan, 1998) for learning axis-aligned rectangles with respect to product distributions from MI examples in the PAC model. Auer (1997) gives a more efficient algorithm. Blum and Kalai (1998) show that learning from multiple-instance examples is reducible to PAC-learning with two sided noise and to the statistical query model. Integer programming, expectation maximization, and kernel formulations are also proposed for MI classification problem (see e.g., (Wang and Zucker, 2000; Zhang and Goldman, 2001; Gartner et al., 2002; Andrews et al., 2002; Mangasarian and Wild, 2008)). Ray and Craven (2005) provide a benchmark of several multiple instance classification algorithms and their non-multiple-instance counterparts. Kundakcioglu and Pardalos (2008) formulate MI classification problem as the following mixed 0–1 quadratic programming problem.

$$\min_{\boldsymbol{\psi},b,\xi,\eta} \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + C\sum_{i=1}^{l}\xi_i^2 \tag{1–16a}$$

$$\text{subject to} \quad \langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i - M(1-\eta_i) \qquad \forall i : i \in I_j \wedge y_j = 1 \tag{1–16b}$$

$$-\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle - b \geq 1 - \xi_i \qquad \forall i : i \in I_j \wedge y_j = -1 \tag{1–16c}$$

$$\sum_{i \in I_j} \eta_i \geq 1 \qquad \forall j : y_j = 1 \tag{1–16d}$$

$$\eta_i \in \{0,1\} \qquad \forall i : i \in I_j \wedge y_j = 1 \qquad \text{(1–16e)}$$

In this formulation $M$ is a sufficiently large number that ensures that the constraint is active if and only if $\eta_i = 1$. $\eta_i$ is a binary variable that is 1 if $i^{\text{th}}$ instance is one of the actual positive examples. In (Kundakcioglu and Pardalos, 2008), a branch and bound algorithm is proposed for this problem that outperforms a commercial solver for large scale problems. MI classification has very successful implementations in application areas such as drug design (see e.g., (Jain et al., 1994; Dietterich et al., 1997)) and protein family modeling (see e.g., (Tao et al., 2004)).

Seref et al. (2008c) introduce a generalized support vector classification framework, called the *Selective Support Vector Machine*: Let $S_i$, $i = 1, \ldots, l$ be mutually exclusive sets of pattern vectors such that all pattern vectors $\mathbf{x}_{i,k}$, $k = 1, \ldots, |S_i|$ have the same class label $y_i$. The goal is to select only one pattern vector $\mathbf{x}_{i,k^*}$ from each set $S_i$ such that the margin between the set of selected positive and negative pattern vectors are maximized. This problem is formulated as a quadratic mixed integer programming problem, which is a generalization of the standard support vector classifiers and multiple instance classifiers.

$$\min \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + C\sum_{i=1}^{l}\sum_{k=1}^{|S_i|}\xi_{i,k}^2 \qquad \text{(1–17a)}$$

$$\text{subject to} \quad y_i(\langle \boldsymbol{\psi}, \mathbf{x}_{i,k}\rangle + b) \geq 1 - \xi_{i,k} - M(1 - v_{i,k}) \quad i = 1\ldots, l; k = 1, \ldots, |S_i| \qquad \text{(1–17b)}$$

$$\sum_{k=1}^{|S_i|} v_{i,k} = 1 \qquad i = 1, \ldots, l \qquad \text{(1–17c)}$$

$$v_{i,k} \in \{0,1\} \qquad i = 1, \ldots, l; k = 1, \ldots, |S_i| \qquad \text{(1–17d)}$$

This quadratic mixed 0–1 programming problem is shown to be $\mathcal{NP}$-hard (see (Seref et al., 2009)). An alternative approach is proposed with the free slack concept as follows:

$$\min \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_{i,k}^2 \tag{1–18a}$$

$$\text{subject to} \quad y_i(\langle \boldsymbol{\psi}, \mathbf{x}_{i,k}\rangle + b) \geq 1 - \xi_{i,k} - v_{i,k} \qquad i = 1,\ldots,l; k = 1,\ldots,|S_i| \tag{1–18b}$$

$$\sum_{i=1}^{l}\sum_{k=1}^{|S_i|} v_{i,k} \leq V \tag{1–18c}$$

$$v_{i,k} \geq 0 \qquad i = 1,\ldots,l; k = 1,\ldots,|S_i| \tag{1–18d}$$

Dual formulation for (1–18) is derived for nonlinear classification. Formulations with free slack provide flexibility to the separating hyperplane to identify the pattern vectors with larger interclass margin. Iterative elimination and direct selection methods are developed to select such pattern vectors using the alternative formulations. These methods are compared with a naïve method on simulated data.

The iterative elimination method for selective SVM is also applied to neural data from a visuomotor categorical discrimination task to classify highly cognitive brain activities in (Seref et al., 2007). Standard and novel kernel based nonlinear classification methods are applied on a neural data recorded during a visuomotor task performed by a macaque monkey. The stages of the visuomotor task are the initial response of the visual cortex, the categorical discrimination of the visual stimuli and the appropriate response for the visual stimuli. A standard SVM classifier and an SVM based adaptive scaling method are used for feature selection in order to detect relevant time intervals and their spatial mapping on the brain. The first and the third stages of the visuomotor task are detectable with the standard SVM classifier. However, for the second stage, SVM classifier performs poorly. Dynamic Time Warping (DTW) is also applied in order to reduce the temporal variances. Motivated by the improvement in the first stage after DTW, selective SVM is applied. It is shown that the results obtained after selective SVM are exceptionally better compared to DTW for both classification accuracy and feature selection. The results

clearly point to the involvement of the superior temporal gyrus and the inferior parietal lobe in the categorical discrimination stage of the visuomotor task.

There are a number of biomedical studies which introduce different methods for image *segmentation*. Segmentation is the process of partitioning a digital image into different sections in order to change the representation of the image. This new representation may involve certain characteristics in the image such as curves, edges, color, intensity or texture. Segmented images are usually used to determine brain abnormalities using image classification, and it is shown that SVMs perform very well. Lee et al. (2005) use SVMs and a new SVM based method called *support vector random fields* for segmenting brain tumors from MR images. Rinaldi et al. (2006) classify brain inflammation in *multiple sclerosis* patients based on the peripheral immune abnormalities from MR images using nonlinear SVMs. They determine that brain inflammation in patients with multiple sclerosis is associated with changes in subsets of peripheral lymphocytes. Thus, SVM classification helps detect a potential biomarker candidate for the prognosis of patients in the early stages of multiple sclerosis. Quddus et al. (2005) combine SVMs and *boosting*, another machine learning method, to perform segmentation (via nonlinear classification) on white matter lesions in the MRI scans. Their composite classification method is shown to be faster and just as reliable as manual selection. In another study by Martinez-Ramon et al. (2006) a similar approach is used to create segments of the brain with respect to their functions. Later, these segments are aggregated using boosting, which is used for multi-class SVM classification of an fMRI group study with interleaved motor, visual, auditory, and cognitive task design.

Automated tumor detection algorithms are a sought-after tool for assisting physicians to make more accurate and rapid detection of tumors. Support vector machine classifiers have contributed significantly in this area. Menze et al. (2006) utilize SVM classification of MRI images to serve as an automated diagnostic tool for the detection of recurrent brain tumors. They report that SVM among other methods is able to rule out lipid and

lactate signals as being too unreliable, and that choline and N-acetylaspartate are the main sources of information (most important features).

Diabetes mellitus (DM) is a common disease in the industrialized countries and it is a prominent risk factor for ischemic cerebrovascular accidents. Diabetes alone is responsible for 7% of deaths in stroke patients. Diabetes mellitus often results in brain micro-blood flow disorders that may cause cerebral infarction. However, assessing the function of cerebral micro-vessels is difficult, since they are located within the bony skull. Kalatzis et al. (2003) apply SVMs to distinguish between blood flow data in patients with diabetes versus control subjects using *SPECT* images from cerebral abnormalities. Li et al. (2006b) use SVMs with *floating search method* to find relevant features for assessing the degree of malignancy in brain glioma from MRI findings and clinical data prior to operations. Li et al. (2006a) develop a novel algorithm that combines bagging of SVMs with embedded feature selection for the individual observations and compare the new algorithm using publicly available data sets. Next, we continue discussing Support Vector method within the regression framework.

### 1.3.4 Other Classification Techniques

In this section, we explore supervised learning methods that do not fall under the category of linear classification. We particularly present "consistent biclustering" and its biomedical applications.

Given a classification of the samples, $S_r$, let $S = (s_{jr})_{n\times k}$ denote a 0–1 matrix where $s_{jr} = 1$ if sample $j$ is classified as a member of the class $r$ (i.e., $a^j \in S_r$), and $s_{jr} = 0$ otherwise. Similarly, given a classification of the features, $F_r$, let $F = (f_{ir})_{m\times k}$ denote a 0–1 matrix where $f_{ir} = 1$ if feature $i$ belongs to class $r$ (i.e., $a_i \in F_r$), and $f_{ir} = 0$ otherwise. Construct corresponding *centroids* for the samples and features using these matrices as follows

$$C_S = AS(S^TS)^{-1} = (c^S_{i\xi})_{m\times r} \tag{1–19}$$

$$C_F = A^T F (F^T F)^{-1} = (c^F_{j\xi})_{n\times r} \tag{1–20}$$

The elements of the matrices, $c^S_{i\xi}$ and $c^F_{j\xi}$, represent average expression of the corresponding sample and feature in class $\xi$, respectively. In particular,

$$c^S_{i\xi} = \frac{\sum_{j=1}^n a_{ij} s_{j\xi}}{\sum_{j=1}^n s_{j\xi}} = \frac{\sum_{j|a^j\in S_\xi} a_{ij}}{|S_\xi|},$$

and

$$c^F_{j\xi} = \frac{\sum_{i=1}^m a_{ij} f_{i\xi}}{\sum_{i=1}^m f_{i\xi}} = \frac{\sum_{i|a_i\in F_\xi} a_{ij}}{|F_\xi|}.$$

Using the elements of matrix $C_s$, one can assign a feature to a class where it is over-expressed. Therefore feature $i$ is assigned to class $\hat{r}$ if $c^S_{i\hat{r}} = \max_\xi\{c^S_{i\xi}\}$, i.e.,

$$a_i \in \hat{F}_{\hat{r}} \Longrightarrow c^S_{i\hat{r}} > c^S_{i\xi}, \qquad \forall \xi, \xi \neq \hat{r}. \tag{1–21}$$

Note that the constructed classification of the features, $\hat{F}_r$, is not necessarily the same as classification $F_r$. Similarly, one can use the elements of matrix $C_F$ to classify the samples. Sample $j$ is assigned to class $\hat{r}$ if $c^F_{j\hat{r}} = \max_\xi\{c^F_{j\xi}\}$, i.e.,

$$a^j \in \hat{S}_{\hat{r}} \Longrightarrow c^F_{j\hat{r}} > c^F_{j\xi}, \qquad \forall \xi, \xi \neq \hat{r}. \tag{1–22}$$

As before, the obtained classification $\hat{S}_r$ does not necessarily coincide with classification $S_r$.

Biclustering $\mathcal{B}$ is referred to as a *consistent biclustering* if relations (1–21) and (1–22) hold for all elements of the corresponding classes, where matrices $C_S$ and $C_F$ are defined according to (1–19) and (1–20), respectively.

A data set is *biclustering-admitting* if some consistent biclustering for that data exists. Furthermore, the data set is called *conditionally biclustering-admitting* with respect to a given (partial) classification of some samples and/or features if there exists a consistent biclustering preserving the given (partial) classification.

Let $\mathcal{B}$ be a consistent biclustering. Then there exist convex cones $P_1, P_2, \ldots, P_k \subseteq \mathbb{R}^m$ such that only samples from $S_r$ belong to the corresponding cone $P_r$, $r = 1, \ldots, k$. Similarly, there exist convex cones $Q_1, Q_2, \ldots, Q_k \subseteq \mathbb{R}^n$ such that only features from class $F_r$ belong to the corresponding cone $Q_r$, $r = 1, \ldots, k$.

It also follows from the conic separability that convex hulls of classes do not intersect (see (Busygin et al., 2005)).

By definition, a biclustering is consistent if $F_r = \hat{F}_r$ and $S_r = \hat{S}_r$. However, a given data set might not have these properties. The features and/or samples in the data set might not clearly belong to any of the classes and hence a consistent biclustering might not be constructed. In such cases, one can remove a set of features and/or samples from the data set so that there is a consistent biclustering for the truncated data. Selection of a representative set of features that satisfies certain properties is a widely used technique in data mining applications. This feature selection process may incorporate various objective functions depending on the desirable properties of the selected features, but one general choice is to select the maximal possible number of features in order to lose minimal amount of information provided by the training set.

Given a set of training data, construct matrix $S$ and compute the values of $c^S_{i\xi}$ using (1–19). Classify the features according to the following rule: feature $i$ belongs to class $\hat{r}$ (i.e., $a_i \in F_{\hat{r}}$), if $c^S_{i\hat{r}} > c^S_{i\xi}$, $\forall \xi \neq \hat{r}$. Finally, construct matrix $F$ using the obtained classification. Let $x_i$ denote a binary variable, which is one if feature $i$ is included in the computations and zero otherwise. Consistent biclustering problem is formulated as follows.

$$\max_x \quad \sum_{i=1}^{m} x_i \tag{1–23a}$$

$$\text{subject to} \quad \frac{\sum_{i=1}^m a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^m f_{i\hat{r}} x_i} > \frac{\sum_{i=1}^m a_{ij} f_{i\xi} x_i}{\sum_{i=1}^m f_{i\xi} x_i}, \qquad \forall \hat{r}, \xi = 1, \ldots, k, \hat{r} \neq \xi, j \in S_{\hat{r}} \tag{1–23b}$$

$$x_i \in \{0, 1\}, \qquad i = 1, \ldots, m \tag{1–23c}$$

The optimization problem (1–23) is proved to be $\mathcal{NP}$-hard (see (Kundakcioglu and Pardalos, 2009a)) and is a specific type of *fractional 0–1 programming problem*, which can be solved using the approach described in (Busygin et al., 2005). When it has a feasible solution, the corresponding feature selection makes the data set conditionally biclustering-admitting with respect to the given classification of samples.

Two generalizations of (1–23) and an improved heuristic procedure is proposed in (Nahapetyan et al., 2008). In this method, a linear programming problem with continuous variables is solved at each iteration. Numerical experiments on the data, which consists of samples from patients diagnosed with *acute lymphoblastic leukemia (ALL)* or *acute myeloid leukemia (AML)* diseases (see (Golub et al., 1999; Ben-Dor et al., 2000, 2001; Weston et al., 2000; Xing and Karp, 2001)), confirm that the algorithm outperforms the previous results in the quality of solution as well as computation time.

Busygin et al. (2007a) apply biclustering to analyze the electroencephalogram (EEG) data. Some biomedical applications of biclustering are DNA microarray analysis and drug design (see e.g., (Busygin et al., 2008; Madeira and Oliveira, 2004; Tanay et al., 2004)). However, biclustering is shown to be also useful for feature selection which is the major concern of many biomedical studies (see (Busygin et al., 2007a)). Revealing subsets of channels whose Lyapunov exponents consistently change with switching the VNS stimulation ON and OFF is claimed to be very much in line with discovering upregulated and downregulated genes in a microarray data set. Therefore, each EEG channel is represented as a feature and data samples taken within the stimulation periods versus samples taken outside of these periods are analyzed. It is shown that the method of biclustering is able to perform successful feature selection. Another study where epilepsy treatment with vagus nerve stimulation is by Uthman et al. (2007). See (Chaovalitwongse et al., 2007; Sabesan et al., 2008) for other applications of optimization to epileptic brain disorders.

Recently, Bertsimas and Shioda (2007) introduce mixed-integer optimization methods to the classical statistical problems of classification and regression and construct a software package called CRIO (classification and regression via integer optimization). CRIO separates data points into different polyhedral regions. In classification, each region is assigned a class, while in regression each region has its own distinct regression coefficients. Computational experimentations with generated and real data sets show that CRIO is comparable to and often outperforms the current leading methods in classification and regression. These results illustrate the potential for significant impact of integer optimization methods on computational statistics and data mining.

Logical Analysis of Data (LAD) is a technique that is used for risk prediction in medical applications (see (Alexe et al., 2003)). This method is based on combinatorial optimization and boolean logic. The goal is essentially classifying groups of patients at low and high mortality risk and LAD is shown to outperform standard methods used by cardiologists.

Another supervised learning method is by Mammadov et al. (2007a) where a multi-label classifier is considered. See (Lee and Wu, 2007; Lee, 2008) for surveys on classification and disease prediction methods that use mathematical programming techniques.

### 1.4 Linear Regression

Given a training data set comprising the features and the corresponding target values, the goal in regression is to take an input vector $\mathbf{x}$ and predict the value of one or more continuous *target* variables. In a typical *regression* problem, each pattern vector in the training set $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \ldots, l$ is associated with a label (target value) $y_i \in \mathbb{R}$. A regression function is obtained using the training set and that function predicts the target values for the test set.

The Support Vector method can also be applied to the case of regression, maintaining all the main features that characterize the maximal margin algorithm. This method is

called the Support Vector Regression (SVR). As in the classification case the learning algorithm minimizes a convex functional.

As with the classification, we motivate the approach by seeking to optimize the generalization bounds given for regression. These rely on defining a loss function that ignores errors within a certain distance of the true value. This type of function is referred to as an $\epsilon$-insensitive loss function. With many reasonable choices of loss function, the solution is characterized as the minimum of a convex functional. Another motivation for considering the $\epsilon$-insensitive loss function is that it will ensure sparseness of the dual variables similar to the classification case. The idea of representing the solution by means of a small subset of training points has enormous computational advantages. $\epsilon$-insensitive loss function has that sparseness advantage, while still ensuring existence of a global minimum and the optimization of a reliable generalization bound.

In this section, we first describe the $\epsilon$-insensitive loss and then derive two approaches from the bounds relating to the 1-norm or 2-norm of the loss vector.

The linear $\epsilon$-insensitive loss function $L_1^\epsilon(\mathbf{x}, y, f)$ is defined as

$$L_1^\epsilon(\mathbf{x}, y, f) = |y - f(\mathbf{x})|_\epsilon = \max(0, |y - f(\mathbf{x})| - \epsilon)$$

and the quadratic $\epsilon$-insensitive loss function $L_2^\epsilon(\mathbf{x}, y, f)$ is defined as

$$L_2^\epsilon(\mathbf{x}, y, f) = |y - f(\mathbf{x})|_\epsilon^2 = \max(0, |y - f(\mathbf{x})|^2 - \epsilon).$$

Similar to the classification case, a parameter $C$ is introduced to measure the trade-off between complexity and losses. The SVR problem for the quadratic $\epsilon$-insensitive loss function is

$$\begin{aligned}
\min \quad & \frac{1}{2}\|\boldsymbol{\psi}\|^2 + C\sum_{i=1}^{l}(\xi_i^2 + \hat{\xi}_i^2) && && \text{(1–24a)}\\
\text{subject to} \quad & (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) - y_i \leq \epsilon + \xi_i && i = 1,\ldots,l && \text{(1–24b)}\\
& y_i - (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) \leq \epsilon + \hat{\xi}_i && i = 1,\ldots,l && \text{(1–24c)}
\end{aligned}$$

The SVR problem for the linear $\epsilon$-insensitive loss function is

$$\begin{aligned}
\min \quad & \frac{1}{2}\|\boldsymbol{\psi}\|^2 + C\sum_{i=1}^{l}(\xi_i + \hat{\xi}_i) && && \text{(1–25a)}\\
\text{subject to} \quad & (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) - y_i \leq \epsilon + \xi_i && i = 1,\ldots,l && \text{(1–25b)}\\
& y_i - (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) \leq \epsilon + \hat{\xi}_i && i = 1,\ldots,l && \text{(1–25c)}\\
& \xi_i, \hat{\xi}_i \geq 0 && i = 1,\ldots,l && \text{(1–25d)}
\end{aligned}$$

Linear regression is used in identification of a directly proportional relationship between two physicochemical properties and drug activity prediction (see (Jones, 2002)). Breast cancer prognosis is studied extensively using linear programming and a regression framework in (Street et al., 1995) and (Mangasarian et al., 1995).

For the Support Vector Regression, the derivation of the dual is similar to that of SVM classifiers. For the sake of completeness, we only present the dual for SVR.

2-norm dual for SVR is as follows:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{l} y_i\alpha_i - \epsilon\sum_{i=1}^{l}|\alpha_i| - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j\left(K(\mathbf{x}_i,\mathbf{x}_j) + \frac{1}{C}\delta_{ij}\right) && \text{(1–26a)}\\
\text{subject to} \quad & \sum_{i=1}^{l}\alpha_i = 0 && \text{(1–26b)}
\end{aligned}$$

$b^*$ is calculated using $\sum_{j=1}^{l}\alpha_j^* K(\mathbf{x}_j,\mathbf{x}) + b^* - y_i = -\epsilon - \frac{\alpha_j^*}{C}$ for any $\alpha_j^* > 0$. 1-norm dual for SVR is as follows:

$$\max \quad \sum_{i=1}^{l} y_i\alpha_i - \epsilon\sum_{i=1}^{l}|\alpha_i| - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j K(\mathbf{x}_i,\mathbf{x}_j) \tag{1–27a}$$

$$\text{subject to} \quad \sum_{i=1}^{l}\alpha_i = 0 \tag{1–27b}$$

$$-C \le \alpha_i \le C \qquad i = 1,\ldots,l \tag{1–27c}$$

$b^*$ is calculated using $\sum_{j=1}^{l}\alpha_j^* K(\mathbf{x}_j,\mathbf{x}) + b^* - y_i = -\epsilon$ for any $0 < \alpha_j^* < C$.

### 1.5 Biomedical Treatment and Other Applications

In this section, we first discuss multi-quadratic 0–1 programming with applications in biomedical research on epilepsy. Epilepsy consists of more than 40 clinical syndromes affecting 50 million people worldwide. At least 30% of patients with epilepsy continue to have seizures despite treatment with antiepileptic drugs. Epileptic seizure occurrences seem to be random and unpredictable. However, recent studies in epileptic patients suggest that seizures are deterministic rather than random. Subsequently, studies of the spatiotemporal dynamics in electroencephalograms (EEGs), from patients with temporal lobe epilepsy, demonstrate a preictal transition of approximately 30 minutes to 1 hour duration before the ictal onset. The enormous number of neurons and dynamic nature of connections between them makes the analysis of brain function especially challenging. In order to perform a quantitative analysis of brain, one can treat certain groups of neurons (functional units of the brain) as vertices of a graph and investigate the connections between these functional units. *Lyapunov exponent* is an important measure that characterizes chaotic behavior of nonlinear system. Global Lyapunov exponent highlights how fast nearby orbits of the system converge or diverge in infinitely large time interval. On contrary, local Lyapunov exponent characterize local predictability around a point in phase space. Lyapunov exponent has proven its efficiency in EEG analysis for predicting epileptic seizures.

Let a system be set by

$$\dot{X}(t) = F(X), \text{ where } X : \mathbb{R} \to \mathbb{R}^n, F : \mathbb{R}^n \to \mathbb{R}^n$$

The maximal Lyapunov exponent $\lambda$ can be defined as

$$\lambda = \lim_{t \to \infty} \lim_{\delta X(0) \to 0} \frac{1}{t} \log_2 \frac{\delta X(t)}{\delta X(0)} \tag{1–28}$$

For short term maximal Lyapunov exponent ($STL_{\max}$) we can take reasonable $t$ instead of external limit. In real life we often deal with one dimensional time series of noisy data (such as EEG signal) instead of explicit system of equations. Wolf et al. (1985) suggest an algorithm for Lyapunov Exponent calculation from time series. Pardalos et al. (2004) and Chaovalitwongse et al. (2006) use modification of Wolfs algorithm described in (Iasemidis, 1991) for $STL_{\max}$ calculation that handles noisy non-stationary data.

$$STL_{\max} = \frac{1}{t_M - t_0} \sum_{k=1}^{M} \log_2 \frac{L'(t_k)}{L(t_{k-1})} \tag{1–29}$$

Since the brain is a nonstationary system, algorithms used to estimate measures of the brain dynamics should be capable of automatically identifying and appropriately weighing existing transients in the data. In a chaotic system, orbits originating from similar initial conditions (nearby points in the state space) diverge exponentially (expansion process). Rate of divergence is an important aspect of the system dynamics and is reflected in the value of Lyapunov exponents. During the last decade, advances in studying brain are associated with extensive use of EEG which can be treated as the quantitative representation of the brain function. EEG data essentially represent time series recorded from the electrodes located in different functional units of brain. We utilize the concept of T-index to measure entrainment of two brain sites at a time moment. T-index at time $t$ between electrode sites $i$ and $j$ is defined as

$$T_{i,j}(t) = \frac{\sqrt{N}\,|E\{STL_{\max,i} - STL_{\max,j}\}|}{\sigma_{i,j}(t)}$$

where $E\{\}$ is the sample average difference for the $STL_{\max,i} - STL_{\max,j}$ estimated over a moving window $w_t(\lambda)$. At the moment of a seizure some brain sites exhibit the convergence of their EEG signals, which is characterized by drop of the corresponding T-index below $T_{\text{critical}}$. A natural graph representing the brain is where each vertex corresponds to a functional unit/electrode, and there is a edge between two of them if T-index is below $T_{\text{critical}}$. The number of edges in this graph dramatically increases at seizure points, and it decreases immediately after seizures.

One aspect of the analysis of the epileptic brain is finding a maximum clique in this graph. It provides us with the largest set of critical electrode sites most entrained during the seizure. If the number of critical sites is set equal to $k$, we can formulate the problem of selecting the optimal group of critical site as a multi-quadratic 0–1 programming as follows.

Let $x_i \in \{0, 1\}$ denote if site $i$ is selected. $a_{ij}$ is the T-index between sites $i$ and $j$ during the seizure point. $b_{ij}$ is the T-index between sites $i$ and $j$ 10 minutes after the onset of seizure.

$$\begin{aligned}
\min \quad & x^T A x && \text{(1–30a)}\\
\text{subject to} \quad & \sum_{i=1}^{n} x_i = k && \text{(1–30b)}\\
& x^T B x \geq T_{\text{critical}} k(k-1) && \text{(1–30c)}\\
& x \in \{0, 1\}^n && \text{(1–30d)}
\end{aligned}$$

Pardalos et al. (2004) develop a novel linearization technique to reformulate a quadratically constrained quadratic 0–1 programming problem as an equivalent mixed integer programming (MIP) problem. The practical importance of this reformulation is that number of 0–1 variables remains the same and number of additional continuous variables is $O(n)$, where $n$ is the number of 0–1 variables.

Next, we discuss *radiation therapy*, which is arguably the most popular domain in treatment planning, where optimization techniques are used. Radiation therapy is a method used to treat cancer with high-energy radiation that destroys the ability of cancerous cells to reproduce. There are two types of radiation therapy. The first one is *external beam radiotherapy*, or teletherapy, where high-energy rays are aimed to the cancerous tissues. Multileaf collimator shapes the beam by blocking out some parts of the beam. To precisely shape the beam, multileaf collimators consist of a small array of metal leaves for each beam. Each beam is specified by a set of evenly spaced strips and the treatment plan is defined by a collection of beams with the amount of radiation to be delivered along each strip within each beam. The second radiation therapy method is *brachytherapy*, also know as sealed source radiotherapy or endocurietherapy, where a radioactive source is placed inside or next to the area requiring treatment. Brachytherapy is commonly used to treat localized prostate cancer and cancers of the head and neck. Both types of therapy need to be planned to localize the radiation area so that the destroyed tissue is minimized.

Many optimization algorithms are developed for the treatment planning in radiation therapy which employ techniques such as multiobjective optimization (see (Lahanas et al., 2003a,b)), investigating tradeoffs between tumor coverage and critical organ sparing (see (Craft et al., 2006)), linear programming (see (Lodwick et al., 1999)), mixed-integer programming (see (Lee and Zaider, 2003; Lee et al., 2001)), non-linear programming (see (Billups and Kennedy, 2001; Ferris et al., 2001)), simulated annealing (see (Webb, 1991)), and inverse planning with a genetic algorithm-based framework (see (Bevilacqua et al., 2007)).

Recently, Men et al. (2007) consider the problem of intensity-modulated radiation therapy (IMRT) treatment planning using direct aperture optimization. In contrast to the heuristic approaches, an exact approach is used that explicitly formulates the fluence map optimization (FMO) problem as a convex optimization problem in terms of all

multileaf collimator (MLC) deliverable apertures and their associated intensities. However, the number of deliverable apertures, and therefore the number of decision variables and constraints in the new problem formulation, is typically enormous. To overcome this, an iterative approach is used that employs a subproblem whose optimal solution either provides a suitable aperture to add to a given pool of allowable apertures or concludes that the current solution is optimal. This approach has the additional advantage that it can explicitly account for transmission of dose through the part of an aperture that is blocked by the MLC system, yielding a more precise assessment of the treatment plan than what is possible using a traditional beamlet-based FMO problem. Tests on clinical head-and-neck cancer cases show the efficacy of this approach, yielding treatment plans comparable in quality to plans obtained by the traditional method with a reduction of more than 75% in the number of apertures and a reduction of more than 50% in beam-on time, with only a modest increase in computational effort. The results also show that delivery efficiency is very insensitive to the addition of traditional MLC constraints; however, jaws-only treatment requires about a doubling in beam-on time and number of apertures used.

For a review on optimization methods in radiation therapy, the reader is referred to (Shepard et al., 1999; Ehrgott et al., 2008). Acosta et al. (2008) study the influence of dose grid resolution on beam selection strategies in radiotherapy treatment design. Censor et al. (2006) study a unified model for handling dose constraints and radiation source constraints in a single mathematical framework based on the split feasibility problem. See (Brandeau et al., 2004) for description of other treatment problems. Future research directions for radiation therapy are discussed in (Lee et al., 2001).

Another problem that has been extensively studied is the non-unique probe selection. This problem consists of selecting oligonucleotide probes for use in hybridization experiments in which target viruses or bacteria are to be identified in biological samples. The presence or absence of these targets is determined by observing whether selected

probes bind to their corresponding sequences. The goal is to select a probe set that is able to uniquely identify targets while containing a minimal number of probes. Ragle et al. (2007) present the first exact method for finding optimal solutions to the non-unique probe selection problem within practical computational limits, without the a priori elimination of candidate probes. Previous methods have employed heuristics to find approximate solutions that are not provably optimal, and as a result, no knowledge has been obtained regarding the quality of those solutions relative to optimality. The computational results show that the method can find the optimal solution within 10 minutes, and is capable of reducing the number of probes required over state-of-the-art heuristic techniques by as much as 20%.

Using $d$-disjunct matrix, Thai et al. (2007b) present two $(1+(d+1)\log n)$-approximation algorithms to identify at most $d$ targets for the non-unique probe selection problem. Based on their selected non-unique probes, the decoding algorithms with linear time complexity are also presented. The proposed algorithms with fault tolerant solutions can identify at most $d$ targets in the presence of experimental errors.

Other optimization based studies in biomedicine are in DNA microarray experiments (see (Ugur and Weber, 2007; Kochenberger et al., 2005; Busygin et al., 2007b)), intensity modulated proton therapy (see (Pflugfelder et al., 2008)), ultrasound-mediated DNA transfection (see (Zarnitsyn and Prausnitz, 2004)), protein design and gene networks (see (Meneses et al., 2007; Balasundaram et al., 2005; Fung et al., 2005; Strickler et al., 2006; McAllister et al., 2007; Donahue et al., 2007; Thai et al., 2007a)), human motion analysis (see (Dariush, 2003)), imaging (see (Dube et al., 2007; Carew and Yuan, 2007; Louis, 2008)), ultrasound surgery (see (Huttunen et al., 2008)), corneal rotation (see (Karpouzas and Pouliquen, 1991)), drug design (see (Mammadov et al., 2007b; Pardalos et al., 2005)), vaccine formularies (see (Hall et al., 2008)) and query optimization in database integration (see (Sujansky, 2001)). Marchuk (1997) develops mathematical models of infectious diseases, antiviral immune response and antibacterial response. These models

are formulated as systems of delay-differential equations and applied to the study of hepatitis B, influenza A, infectious bacterial pneumonia, and mixed infections. For future research directions and open problems in other fields of biomedicine and computational biology, the reader is referred to (Greenberg et al., 2004).

## 1.6 Concluding Remarks

In this chapter, we summarize some of the recent studies on biomedicine that utilize optimization theory. This review is not exhaustive in that, some techniques are explored in depth and references are given for further studies. Applications of optimization already improve quality of healtcare research but there are still many open problems. As computer capabilities improve and novel implementations are developed with higher level of interdisciplinary effort, optimization will continue to provide ongoing revelations in the constantly growing field of healthcare.

# CHAPTER 2
# CELL DEATH DISCRIMINATION WITH RAMAN SPECTROSCOPY AND SUPPORT VECTOR MACHINES

In the present study, Raman spectroscopy is employed to assess the potential toxicity of chemical substances. Having several advantages compared to other traditional methods, Raman spectroscopy is an ideal solution for investigating cells in their natural environment. In the present work, we combine the power of spectral resolution of Raman with one of the most widely used machine learning techniques. Support vector machines (SVMs) are used in the context of classification on a well established database. The database is constructed on three different classes: healthy cells, Triton-X100 (necrotic death), and etoposide (apoptotic death). SVM classifiers successfully assess the potential effect of the test toxins (Triton X-100, etoposide staurosporine). The cells that are exposed to heat (45 $^{o}$C) are tested using the classification rules obtained. It is shown that the heat effect results in apoptotic death, which is in agreement with existing literature.

## 2.1 Introduction

Cell death and the post-mortem aftermath are integral to the study of many genetic and infectious diseases. It is strongly linked to the toxicological effect of various materials and is of great importance for safe toxicity assessment. In recent literature, cell death is classified in three alternative modes: apoptotic (or type I programmed), autophagic (or type II programmed), and necrotic (or oncotic) cell death. Apoptosis and autophagy are important considerations in the development and treatment of diseases such as cancer (Kanduc et al., 1999, 2003, 2005), and in certain pathogenic infections (Navarre and Zychlinsky, 2000). Usually apoptosis is marked by caspase activation, chromatin condensation, and the formation of apoptotic bodies. Autophagic is marked by autophagic engulfment of organelles and particles. Cells dying by necrosis display organelle swelling with the eventual loss of plasma, membrane integrity, and subsequent inflammation. Monitoring the cell death process, therefore, is an important step in understanding the

pathological processes induced by both disease and pharmaceutical treatments such as anti-cancer drugs.

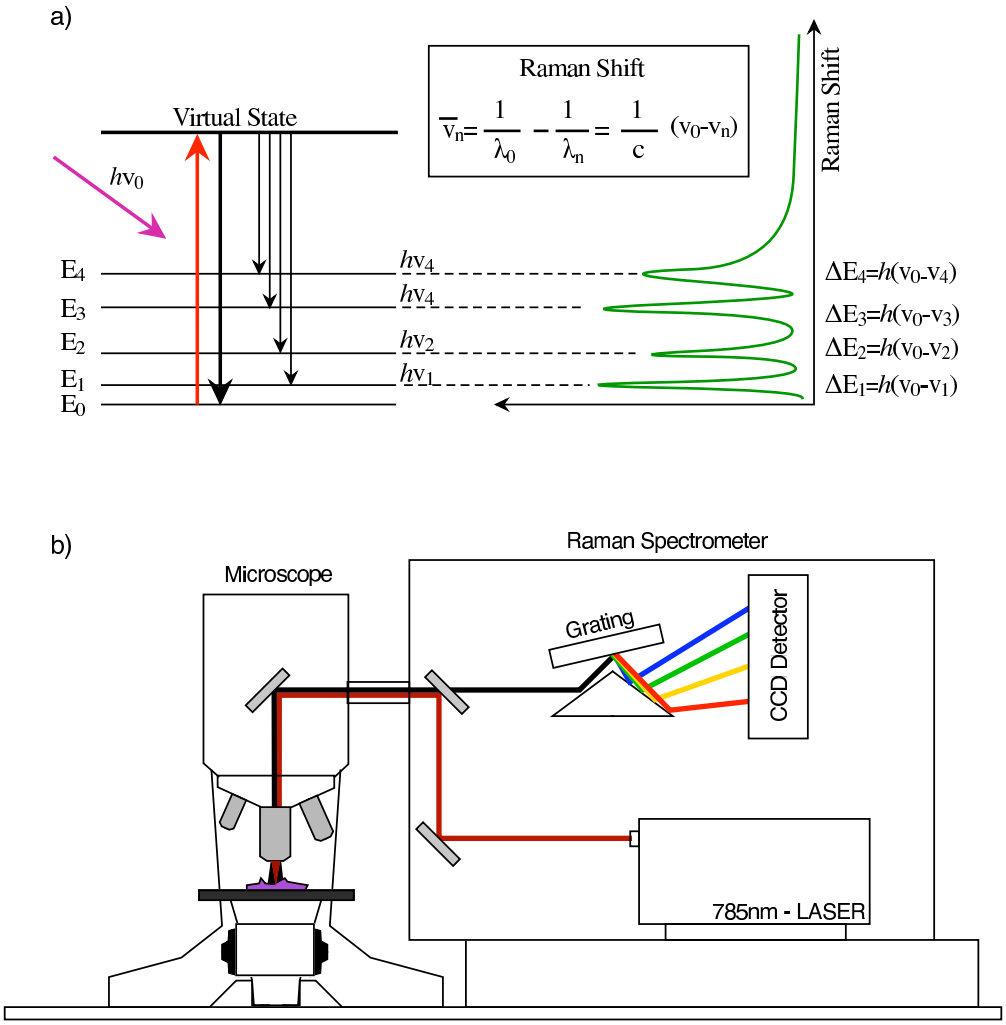


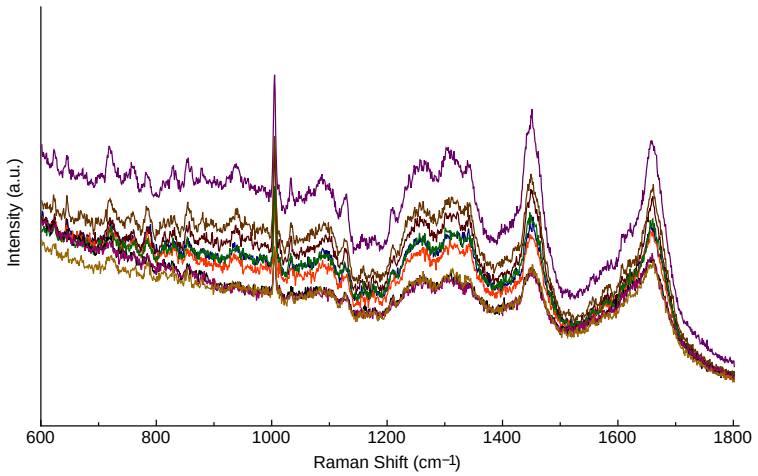

Figure 2-1. The basic principles of Raman spectroscopy. a) A photon of a certain energy and frequency induces vibrational transitions on the examined molecule, by giving a portion of its energy. The transition occurs through a virtual state, created due to the polarizability of the studied molecule. The scattered photon has lower energy than the incident and the energy difference in-between is measured by the detector. This is referred to as the *Raman Shift*. b) The micro Raman utilizes a microscope and focuses the laser through the objective lens on the sample. The scattered photons are collected by the same objective lens and travel the Raman spectrometer, where they are analyzed by a grating and a CCD detector.

Despite the importance of this topic, the pathology and biochemical factors and processes adding to cell death are not yet fully understood. Except morphology, *in-vivo* there is no single assay that can unambiguously identify cell death type (if such an absolute delineation exists) (Jaeschke et al., 2004), which makes characterizing cell death even more difficult. A wide range of cytotoxicity assays are presently in use for the determination of cell viability; however, these techniques have shortcomings. They are destructive, time consuming, and expensive. Current assays depend on large populations and cannot measure the health of individual cells. Furthermore, many factors must be considered when interpreting results. Because cytotoxicity assays rely on chemicals and biomarkers, problems may arise due to unwanted interactions during pharmaceutical testing. Furthermore, in the case where assays are dependent upon enzymatic reactions (e.g., MTT, LDH), results may be skewed by promiscuous enzymatic inhibitors. Specificity issues can also lead to complications in the interpretation of results. Kanduc et al. (2002) compared many of the conventional cytotoxicity assays and find that the reported viability of treated cells differed depending on the assay used. Moreover, a large number of cells is required to determine the exact cellular death and to conclude on the toxicological assessment.

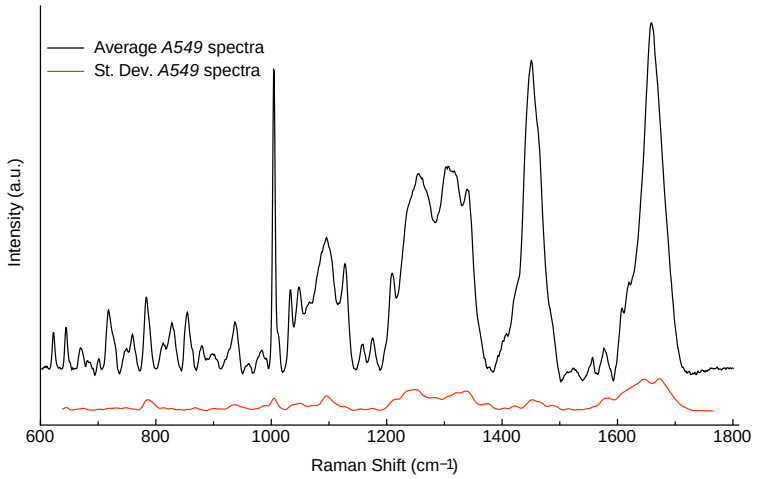
Raman spectroscopy, a well established analytical tool, is being employed as an alternative for studying cell health. It does not share many of the disadvantages inherent in traditional cytotoxicity assays described above (Notingher et al., 2002, 2003). Raman spectroscopy relies on the inelastic scattering of light on matter. It is a complementary technique to the Infra Red (IR) spectroscopy (FTIR, DRIFT etc.). The basic difference lies on the polarizability of the molecule that is required by Raman vs. the polarity that is required by traditional IR spectroscopy. In both cases, the material is radiated with a light of specific frequency that induces an electron transition to a different vibrational state, with an energy loss of the photon. In the case of Raman spectroscopy, due to the polarizability of the molecule, the transition occurs through an intermediate state,

usually called the *virtual state* (see Figure 1). The Raman measurements are based on vibrational modes specific to a molecule and its environment. The spectrum of a cell is a representation of its chemical composition and provides an insight for the bio-molecular changes accompanying cellular processes. There are four main advantages for using Raman spectroscopy: (i) the method is noninvasive and does not require chemicals or markers (ii) measurements can be taken rapidly and in real time, in vivo or in vitro without deleterious results to living cells (iii) it is possible to analyze the health of either a single cell or the entire population (iv) the asymmetric nature of water gives a weak Raman signal that, unlike in infrared spectroscopy, does not interfere with the spectrum of interest (Verrier et al., 2004). It has been successfully used to evaluate the toxicity of pharmaceuticals (Owen et al., 2006), toxins (Notingher et al., 2004), and more recently the toxic effect of particles (Pyrgiotakis et al., 2008).

While Raman spectroscopy has many advantages, there exists one large drawback; highly complex spectra. Because the spectrum of a cell contains information from all cellular components, detecting minute changes from one spectrum to the next can be a daunting task. Traditionally, peak fitting has been used to analyze Raman (and FTIR) spectra. Peak fitting relies on the recognition of peaks representing certain cellular components and correlating their relative peak intensities to their biochemical concentrations within the cell. The relative changes in peak intensity over time are in direct response to the changing biochemical and biophysical factors that are related to the health viability, and eventually to the cell death type and process. However, due to the large number of overlapping peaks, this task becomes very tedious and time consuming. The traditional methodology for analyzing the spectra includes an elaborate series of algorithms. A series of spectra is obtained (see Figure 2-2 (a)) and a series of mathematical procedures is followed to remove the baseline, the fluorescence, to normalize the spectra, to calculate the average and the standard deviation (see Figure 2-2 (b)). Furthermore, the analysis depends on the presumption that one already knows which




(a)




(b)

Figure 2-2. (a) Spectra acquired from 10 different cells after 24 hrs on $MgF_2$ crystal. (b) The average spectrum and standard deviation of 30 A549 cells spectra, after 24 hrs on the $MgF_2$.

peaks are discriminant, and those peaks must be prevalent spectral features with limited interference from background noise and overlapping peaks. Thus, it is critical to develop a method that is applicable for high throughput screening, is simpler than peak fitting to execute, and utilizes the entire spectrum instead of predetermined sections. Moreover, an automated method is desired that can derive results without any manual spectra processing.

For such scientific experiments that generate a large number of measurements and features in each measurement, *machine learning techniques* can be used to analyze the data and derive conclusions. *Supervised learning* refers to the capability of a system to learn from a set of examples, which is a set of input/output pairs. The input is usually a *vector of features* for an object, and the output is the *label* for the class this object belongs to. A set of objects with feature vectors and corresponding class labels is called a *training set*. This set can be used to derive classification or regression functions. The trained system is capable of predicting the label of an object. The term *supervised* originates from the fact that the labels for the objects, which are provided as input in the training set, have to be determined by an outside source. This outside source is usually referred to as the *supervisor*. In this study, cells are labeled as control or death depending on the type of the drug used. Next, classification functions are derived to validate the model and test the labels for cells that are subject to abnormal heat.

The remainder of the chapter is organized as follows: Section 2.2 presents the methods used and the details for the experiments. Computational results are presented in Section 2.3. Section 2.4 gives concluding remarks and directions for future research.

## 2.2 Methods

### 2.2.1 Cell Culture Protocols

For this set of experiments the A549 lung epithelia cells are used (from ATCC; cell line number CCL-185). This is a hypo-triploid human cell line with the modal chromosome number 12, occurring in 24% of cells. The line is initiated in 1972 by Giard et al. (1973) through explants culture of lung carcinomatous tissue from a 58-year-old Caucasian male.

The growth media is made by 89% RPMI-1640 with L-glutamine (from Cellgro; Cat #: 25-053-CI), 10% Fetal Bovine Serum (four times filtered through 0.1 $\mu$m filter, from Hyclone; Cat. #: SH30070.03) and 1% antibiotic-antimycotic solution (from Cellgro; Cat. #: 30-004-CL). The cells are grown with complete growth media in a 25 $cm^2$ cell

culture flask at 37 $^{o}$Cand 5% $CO_2$. In order to harvest the cells for various experimental applications, the culture medium is removed and the cells are rinsed with 1X Hank's Balanced Salt Solution (HBSS) without $Ca^{+2}$ or $Mg^{+2}$ (from ATCC; Cat: 30-22/3) to remove all traces of serum that contains trypsin inhibitor. 1 ml of 0.25% Trypsin/2.21 mM EDTA in HBSS without sodium bicarbonate, calcium and magnesium (from Cellgro; Cat. #: 25-053-CL) is added to the flask and the cells are observed under an inverted microscope until cell layer is detached (usually within 2 to 5 minutes). Occasionally gentle agitation is required to achieve the cell removal in reasonable time frame, since long exposure to the trypsin can damage the cells. 5 ml of complete growth medium is added to deactivate the trypsin, and the cell suspension is centrifuged for 3 minutes @ 1000 rpm. The supernatant is removed and 1 ml of fresh complete media is added, and the cells are gently aspirated resulting into a homogeneous solution of $\sim 3 \times 10^6$ cells/ml and viability ration above 95%. The cell count is done with the ViCell from Beckmann-Coulter (Fullerton, CA). This solution is used for seeding with proximately $\sim 5 \times 10^3$ cells on a 5 × 5 mm $MgF_2$ substrate (custom made by Red-Optronics) used in the Raman. The $MgF_2$ crystal is used to reduce the background radiation from the petri-dish. The seeded $MgF_2$ substrates are placed in a 6 well plate (9.6 $cm^2$ per well) in the incubator for 45 min, sufficient for the cells to attach on the $MgF_2$. 3 ml of growth media is added and the seeded cells are incubated at 37 $^{o}$Cand 5% $CO_2$ for a minimum of 24 hrs before the toxin dosing.

### 2.2.2 Toxic Agent Dosing

After reaching 80% confluency (on the $MgF_2$ plate) the growth media is removed and the cells are rinsed twice with HBSS to remove traces of proteins. Following this, the cells are rinsed with the full media twice to ensure that possible traces of HBSS will not be significant to alter the final concentration of growth media. Finally, 2 ml of media containing the toxic agent are added and the cells are moved in the incubator. Prior to each experiment, the media is removed and the cells are rinsed with HBSS twice and fresh

RPMI 1640 is used since the media proteins can interfere with the obtained spectrum. The absence of proteins does not have any effect on the cells for the time period the experiments last (approximately 1 hour) (Yogalingam and Pendergast, 2008). The RPMI 1640 provides all the necessary growth hormones and sugars essential for the cell viability.

### 2.2.3 Toxic Agents Standards

The triggers for the two different cellular deaths are, etoposide (apoptosis) and Triton X-100 (necrosis). Etoposide is a strong chemotherapeutic drug, used as a form of chemotherapy for malignancies such as Ewing's sarcoma, lung cancer, testicular cancer, lymphoma, non-lymphocytic leukemia, and glioblastoma multiforme. Etoposide is known to form a ternary complex with DNA and topoisomerase II causing double stranded DNA breaks which is one of the defined mechanisms for the apoptosis (Karpinich et al., 2002). It has also been shown to upregulate p53, an initiator of apoptosis (Huang and Plunkett, 1992; Solovyan et al., 1998). Triton X-100 is used as a benchmark in various assays, since it can rapture the cellular membrane and results in the necrotic death of the cells. Triton X-100 exposure is reported to increase the expression of apoptosis inhibitors and is known to solubilize and destabilize the cell membrane (Boesewetter et al., 2006).

The toxin concentrations are selected based on the literature that suggest that these values will impact the cells, but not catastrophically. For the experiments, the agents concentration is 100 $\mu$M for Triton-X (Notingher et al., 2003) and 80 $\mu$M (Yogalingam and Pendergast, 1997; Karpinich et al., 2002; Owen et al., 2006) for the etoposide. These concentrations are expected to induce damage in the cells without completely lysing the cells in the first 24 hrs of the experiment. The solution is prepared immediately prior to dosing. The etoposide is insoluble in water, so a stock solution is prepared with 100 mM of etoposide in di-Methyl-sulfo-oxide (DMSO).

### 2.2.4 Raman Spectroscopy Protocols and Procedures

The Raman microscope used is the InVia system by Renishaw, consisting of a Leica microscope connected to a Renishaw 2000 spectrometer. The high power diode laser (250

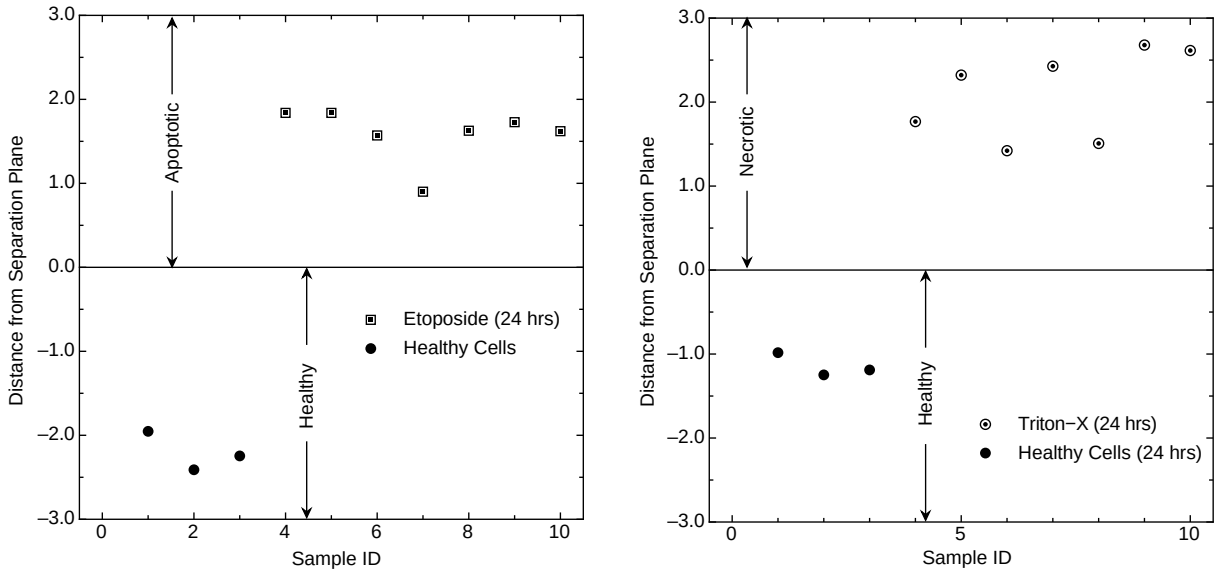



Figure 2-3. Demonstration of the pattern recognition based on SVM classification. (a) The classification of the etoposide induced apoptotic death after 24 hrs exposure. (b) The Triton X-100 induced apoptosis on the $MgF_2$.

mW) produces laser light of 785 nm and does not cause any damage to the cells even after 40 min exposure time. The $MgF_2$ plate after rinsing with the HBSS is moved on a Delta T Culture Dish (from Biotechs; Cat #: 04200415C), and 2 ml of RPMI 1640 is added. The dish is placed onto a heating stage (Delta T4 Culture Dish Controller, Biotechs, Butler, PA, USA) to maintain 37 $^o$Cthrough the entire measurement and induce the required heating. The laser is focused over the center of the cell, with the help of the crosshair, through the Leica microscope. The spot size is 20×40 $\mu$m when focused on dry Si wafer and 20×30 $\mu$m when in water based liquid. It can be assumed therefore that the laser spot can cover the whole cell (20×20 $\mu$m when 80% confluent, 40×40 $\mu$m when isolated). Although the laser spot can be larger than the cell, since the intensity of the laser follows a Gaussian distribution around the geometric center, the parts that are not from the measured cell, are not contributing significantly. However, for the isolated cells the relative position of the laser can potentially effect the spectrum and therefore they are not included in this study. The 785 nm laser beam passes through the 63x water

immersion Leica objective and the final output power is 48 mW. In this study, we collect spectra from cells adhered onto $MgF_2$ plate. The $MgF_2$ plate is used to eliminate the background signal from the petri dish. The collection time for each spectrum is set to 30 s. However, the instrument requires an additional 70 s to read the detector, during which the cell is exposed to the laser. Therefore, the total exposure time is 100 s and immediately after the measurement is preformed, the laser shutter is turned off, a new cell is selected, and the following measurement is immediately executed.

The RPMI media with or without the presence of the various toxins does not influence the spectra. In previous publications, we have developed an algorithm that takes the background, the fluorescence, and the normalization of the spectra into account (Maquelin et al., 1999; Bhowmick et al., 2008). In the present work, the background is obtained and subtracted from the spectra following non linear subtraction. The spectrum before and after are used for classification, but there is no significant difference in the final results. Therefore we omit this step since it is likely that these processes hinder or remove information, essential for the classification techniques.

### 2.2.5 Support Vector Machines

SVM classifiers classify two linearly separable sets of pattern vectors that belong to two different classes. SVMs can be extended to nonlinear classification by implicitly embedding the original data in a nonlinear space using *kernel functions* (Shawe-Taylor and Cristianini, 2004).

SVMs have a wide spectrum of application areas such as pattern recognition (Lee and Verri, 2002), text categorization (Joachims, 1998), biomedicine (Brown et al., 2000; Cifarelli and Patrizi, 2007; Noble, 2004; Seref et al., 2008b), brain-computer interface (Lal et al., 2004; Garcia et al., 2003), and finance (Huang et al., 2004; Trafalis and Ince, 2000). The training is performed by minimizing a quadratic convex function that is subject to linear constraints. Quadratic programming (QP) is an extensively studied field of optimization theory and there are many general purpose methods to solve QP

problems such as quasi-newton, primal-dual and interior-point methods (Bennet and Campbell, 2000). These general purpose methods are suitable for small size problems. In order to solve large problems, faster methods are required. For SVM classifiers, these faster methods involve chunking (Osuna and Girosi, 1997) and decomposition (Platt, 1999) techniques, which use subsets of points to find the optimal hyperplane. SVM Light (Joachims, 1999) and LIBSVM (Hsu et al., 2004) are among the most frequently used software applications that use chunking and decomposition methods efficiently.

The experimental procedure starts by constructing a basic 56×1301 matrix based on the two classes the data must be discriminated to. The discrimination is done always among two different classes. The 56 columns consist of 25 from class 1, 25 from class 2, 3 test subjects from class 1, and 3 test subjects from class 2. The rows represent the different frequencies (600 $cm^{-1}$-1800 $cm^{-1}$ with step 0.92 $cm^{-1}$), while the columns are spectra of different cells in different environmental conditions. There are three different matrices studied; *Necrotic* (NC): Triton X-100 and Control, *Apoptotic* (AC): Etoposide and Control, and *Necrotic vs. Apoptotic* (NA): Triton X-100 and Etoposide. For the validation of classification algorithm, in addition to the 50 data instances of the library, we use 3 control cells and 7 cells with toxins.

To represent the results, we plot the points with x-axis to be the sample ID and y-axis the distance from the hyperplane that separates the two classes. $SVM^{light}$ (Joachims, 1999) is used to train the data in this study. Linear classifiers are used and the trade-off parameter $C$ is set after leave-one-out cross validation technique is employed. When using the leave-one-out method, SVM is trained multiple times, using all but one of the instances in the training set that is selected randomly. The highest prediction accuracy is achieved for $C = 1000$ for training sets of all experiments. Therefore, we set parameter $C$ to 1000 in our computational studies.

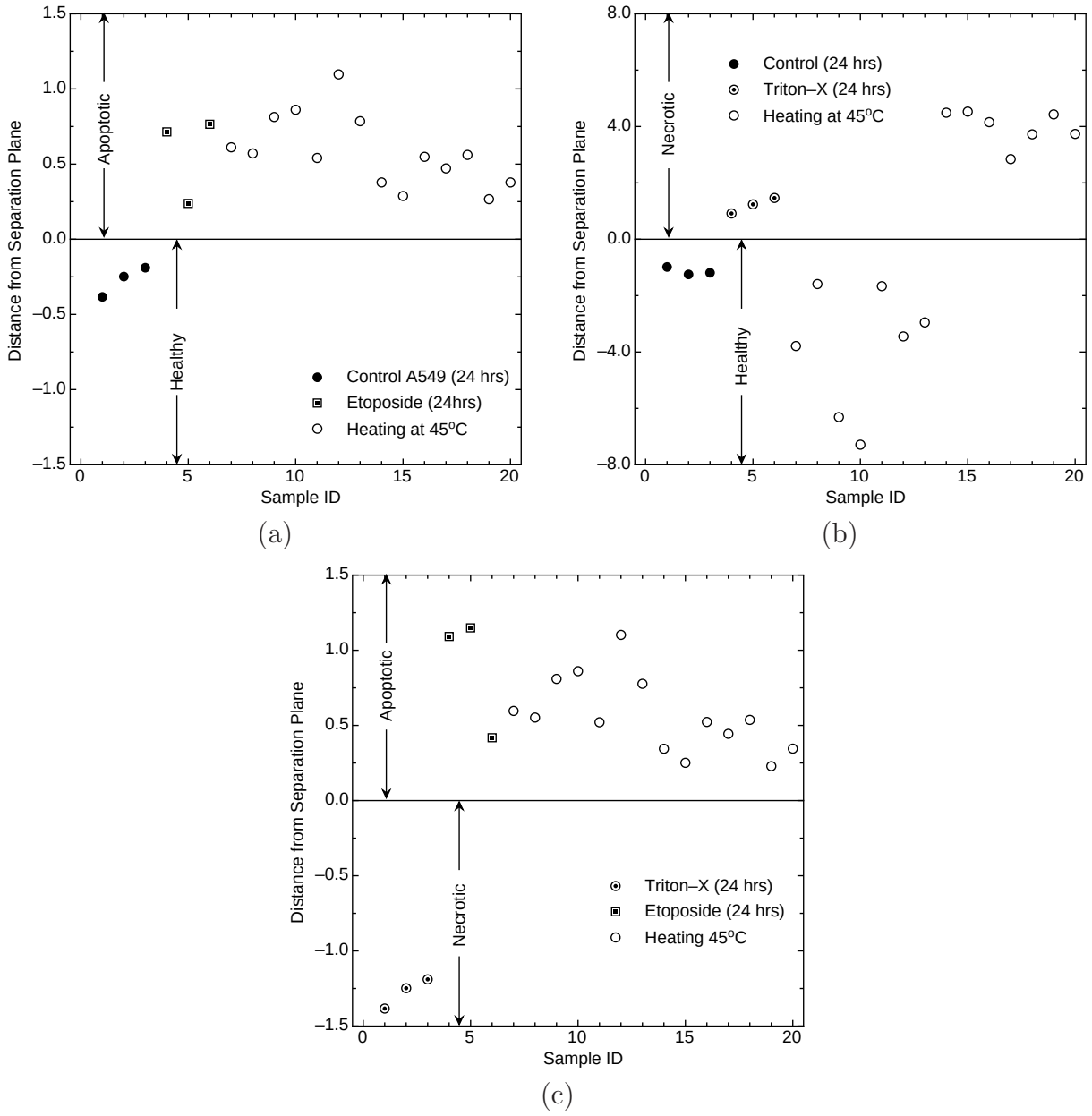



Figure 2-4. The classification of the heating effect. (a) The heating in comparison with the healthy and the apoptotic, (b) the heating in comparison with the healthy and the necrotic, (c) the heating in comparison to the necrotic and the apoptotic.

## 2.3 Results and Discussion

### 2.3.1 Triton-X100 and Etoposide Induced Cellular Death Discrimination

Although the data used for supervised classification are Triton X-100, etoposide, and healthy cells, the first test to validate the accuracy of the classification algorithms is to

attempt to classify same type of data (self validating). For these experiments a series of new spectra is obtained under the same conditions as described above. The concentration of the toxins are kept the same and the exposure time is 24 hrs 30 spectra are obtained from each case (Triton X-100 and etoposide) and randomly 7 are selected to evaluate the algorithm. In addition, parallel with every toxin measurement, a control cell experiment is conducted to further validate the model. From that data set, 3 spectra are randomly selected to even the number of unknowns to 10.

Ideally, the data is expected to have a functional margin of at least 1. However, since the cells are not from the same passage and there are other conditions (humidity, small alterations at the full growth media) that can induce variations, it is not always possible to keep the distance more than 1 (or less than -1). Furthermore, the interaction of each cell individually with the toxin is not the same, due to the complexity of its nature. As it can be seen in Figures 2-3(a) and 2-3(b), SVM classifiers successfully discriminate the control cells from etoposide and Triton X-100, respectively. The distance from the separating hyperplane and small variation showcase the classification and prove the ability of the algorithm to classify the obtained spectra in two classes.

### 2.3.2 Case Study: Heat Induced Cellular Death

For the past four decades hyperthermia has been used to potentiate the cytotoxic effects of ionizing radiation (Robinson et al., 1974; Gerner et al., 1975) and chemotherapy (Hildebrandt et al., 2002; Robinson et al., 1974). It has been established that elevated temperatures alone cause cell death in a predictable manner that is linearly dependent on exposure time and is non-linearly dependent on temperature (Sapareto and Dewey, 1984; Dewhirst et al., 1984). A variety of cell lines, including A549, have been reported to undergo apoptosis (Hayashi et al., 2005; Armour et al., 1993) during mild heat treatment and necrosis during prolonged or intensified exposure (Tadashi et al., 2004; Prasad et al., 2007; Hildebrandt et al., 2002). In this study, heat treatment at 45 $^{o}$Cover 30 minutes is used to test the predictive strength of the model by using a different cell death trigger

that would induce a form of programmed cell death. The heating stage Delta T4 Culture Dish Controller is used and the ramping rate is approximately $0.5^{o}$C/min. The spectra are obtained after 30 minutes of exposure to the heat. As in the previous case, in parallel to this experiment, control, apoptotic (etoposide), and necrotic (Triton X-100) cells are used after 24 hrs exposure.

Assuming that the effect of the heat is the unknown sample, we try to attempt classification, among all the three classes, *healthy*, *apoptotic*, and *necrotic*. Since there are many drawbacks of hyperplane-based multi-class learning techniques (Bishop, 2006), pairwise examination is performed across all the possible combinations. So in this particular case, we examine Healthy−Necrotic, Healthy−Apoptotic, and Apoptotic−Necrotic. In Figure 2-4 (a) are the results of the heating experiment as it is attempted for apoptotic death vs. healthy cells. The heating experiment is classified as apoptotic death. As it can be seen in the figure, most of the samples are lying between 0.3-1.0 in regards to the distance from the hyper-plane. The next step is to check the case of the necrotic cell death vs. healthy cells. In this case, the results of the classification appear to be scattered among both classes, while the test instances are classified correctly (see Figure 2-4 (b)). This is an inconclusive result since there is no particular trend. This can happen, either because the classification is wrong, or because some of the instances are indeed necrotic. If the second is true, then a classification among apoptotic vs. necrotic will classify them again as necrotic. Therefore the last classification is performed among the necrotic and apoptotic cells. Figure 2-4 (c) shows that all the heating spectra are classified again as apoptotic. So in the cases where the apoptotic death is used as one of the two classes, the heat exposed cells are classified as apoptotic.

## 2.4 Concluding Remarks

The coupling of Raman spectroscopy, a very powerful and non invasive technique, with SVM classification algorithms is used to identify cellular death induced by toxins and by low temperature heating. Although there are very few known reports (Widjaja et al.,

2008b,a) that combine these two fields, it is the first known attempt towards the issue of cell death identification. The classification models built with Raman spectral data can be used to discriminate between minute biochemical differences within cells rapidly, in real time, and in a nondestructive and noninvasive manner. A very important aspect, further highlighting the results, is the success to classify biological samples that can present alteration, and differences in their signal due to external (or internal) parameters. Those alterations are manifested to the current project by the variations in the distance from the separating hyperplane. Cases, however, in real biological systems always exhibit minute variations and alteration. The success of this technique (Raman-SVM) is showcased by the fact that although it is able to detect these minute changes, it does not prevent the algorithm from correctly classifying the results.

This study sets the foundation for developing diagnostic tools for cancer or other genetic diseases, the cellular response to chemotherapy and the toxicity assessment of drugs and particles. Future work will explore the sensitivity of this technique in terms of its ability to distinguish finer biochemical or biophysical processes related to cell death such as caspase activation or chromatin condensation. It is critical to expand this methodology to include more than two classes without pairwise comparison and therefore being able to distinguish immediately between various stages of the cell.

# CHAPTER 3
# SELECTIVE SUPPORT VECTOR MACHINES

In this study, we introduce a generalized support vector classification problem: Let $X_i$, $i = 1, \ldots, n$ be mutually exclusive sets of pattern vectors such that all pattern vectors $\mathbf{x}_{i,k}$, $k = 1, \ldots, |X_i|$ have the same class label $y_i$. Select only one pattern vector $\mathbf{x}_{i,k^*}$ from each set $X_i$ such that the margin between the set of selected positive and negative pattern vectors are maximized. This problem is formulated as a quadratic mixed 0-1 programming problem, which is a generalization of the standard support vector classifiers. The quadratic mixed 0-1 formulation is shown to be $\mathcal{NP}$-hard. An alternative approach is proposed with the free slack concept. Primal and dual formulations are introduced for linear and nonlinear classification. These formulations provide flexibility to the separating hyperplane to identify the pattern vectors with large margin. Iterative elimination and direct selection methods are developed to select such pattern vectors using the alternative formulations. These methods are compared with a naïve method on simulated data. The iterative elimination method is also applied to neural data from a visuomotor categorical discrimination task to classify highly cognitive brain activities.

## 3.1 Introduction

In this study, we introduce the concept of selective classification which is a generalization of SVM classification. In selective classification, we consider $n$ sets of positive and negative labeled pattern vectors with $t$ pattern vectors in each set. All of the pattern vectors in a set share the same label. Given this input, the objective is to select a single pattern vector from each of the $n$ sets such that the selected positive and negative pattern vectors produce the best possible solution for a binary classification problem $\mathbf{P}$. In the SVM context, this classification problem $\mathbf{P}$, given in Formulation 3–1, is the quadratic optimization problem that maximizes the margin between positive and negative pattern vectors. The standard SVM problem can be considered as a special case of selective SVM classification where $t = 1$.

Selective classification resembles the multiple instance learning (MIL) problem in its input (Dietterich et al., 1997). However, MIL involves classifying positive and negative *bags* of pattern vectors, where each bag contains a number of pattern vectors sharing the same label. Given a classification function for MIL problem, at least one pattern vector in a positive bag should be classified correctly for that bag to be counted as correctly classified. For a negative bag to be correctly classified, all of the pattern vectors in it should be classified correctly. The MIL problem is to find a classification function that obtains a high classification accuracy for the bags. The objective in selective classification is not classifying the bags. It is, rather, to select a single pattern vector from each set (bag) to maximize the margin between the selected positive and negative pattern vectors.

The selective classification problem poses a hard combinatorial optimization problem. In this chapter, we show that the selective SVM problem is $\mathcal{NP}$-hard. We provide alternative approaches to the hard selection. We introduce the *restricted free slack* concept, which provides flexibility to the hyperplane by decreasing the influence of the pattern vectors that are misclassified or very close to the hyperplane. The resulting optimization problem is also convex and quadratic with linear constraints, and therefore can be kernelized through its Lagrangian dual. We present theoretical results on how the restricted free slack is distributed among the pattern vectors. We introduce algorithms based on these results. These algorithms are tested on simulated data and compared with naive methods. This algorithm is also tested on a neural database to improve the classification accuracy and the performance of an SVM based feature selection method.

The remainder of the chapter is organized as follows. We introduce the concept of selective classification in Section 3.2, where the combinatorial selective classification problem is shown to be $\mathcal{NP}$-hard. The alternative formulations are discussed in Section 3.3. In Section 3.4, different algorithms based on the selective classification formulations are presented. In Section 3.5, computational results from the application of the proposed

methods on simulated data as well as real-life neural data from a visuomotor categorical discrimination task are presented. Finally, we conclude our results in Section 3.6.

### 3.2 A Combinatorial Selective SVM Problem

Here, we introduce a new combinatorial classification problem in which each pattern vector in a standard classification problem is replaced with a set of possible pattern vectors sharing the same class label. We give the definition of the selective margin maximization problem as follows.

**Definition 1.** Selective Margin Maximization Problem

*Let* $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ *be sets of pattern vectors with* $t$ *pattern vectors* $\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,t}$ *in each set* $\mathbf{X}_i$*. Let* $\mathbf{y} = \{y_1, \ldots, y_n\}$ *be the corresponding labels for each set* $\mathbf{X}_i$ *with each pattern vector* $\mathbf{x}_{i,k}$ *having the same label* $y_i$*. Choose exactly one pattern vector* $\mathbf{x}_{i,k^*}$ *from each set* $\mathbf{X}_i$ *such that the margin between the selected positive and negative pattern vectors is maximized.*

The selective margin maximization problem can be formulated as a quadratic mixed 0-1 programming problem as follows.

$$\min \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\sum_{k=1}^{t}\xi_{i,k}^2 \tag{3–1a}$$

$$\text{subject to} \quad y_i(\langle \boldsymbol{\psi}, \mathbf{x}_{i,k}\rangle + b) \geq 1 - \xi_{i,k} - M(1 - \upsilon_{i,k}) \qquad i = 1 \ldots, n; k = 1, \ldots, t, \tag{3–1b}$$

$$\sum_{k=1}^{t} \upsilon_{i,k} = 1 \qquad i = 1, \ldots, n, \tag{3–1c}$$

$$\upsilon_{i,k} \in \{0, 1\} \qquad i = 1, \ldots, n; k = 1, \ldots, t. \tag{3–1d}$$

Note that this formulation is similar to (1–3), except for the extra term $M(1 - \upsilon_{i,k})$ in (3–1b) and the new constraints (3–1c) and (3–1d). $M$ is a sufficiently large positive number. Binary variables $\upsilon_{i,k}$ indicate whether $k^{th}$ pattern vector from set $i$ is selected or not. Note that when $\upsilon_{i,k} = 0$, the right side of (3–1b) becomes sufficiently small such that the constraint is always satisfied, which is equivalent to removing the point from the

training set. Constraint (3–1c) ensures that only one pattern vector is included from each set.

It is clear that for sufficiently high penalty $C$, the selective SVM formulation can be considered as a hard selection problem without the slack variables $\xi_i$, whose solution would provide a hyperplane that can completely separate the selected positive and negative pattern vectors. Now, consider the following decision problem:

**Definition 2.** *Decision Selective SVM (**D-SSVM**) Problem:*

*Let $\mathbf{X}_i = \{\mathbf{x}_{i,j}\}$ denote a set of d-dimensional vectors, where $j = 1, \ldots, t$. Assume that there are $n$ such sets and all vectors $\mathbf{x}_{i,j}$ in each set $\mathbf{X}_i$ are labeled with the same label $y_i \in \{+1, -1\}$. Let $\boldsymbol{\upsilon}^*$ denote a selection where a single vector $\mathbf{x}_{i,j^*}$ is selected from each set $\mathbf{X}_i$. Is there a selection $\boldsymbol{\upsilon}^*$ such that all positive and negative pattern vectors can be separated by a hyperplane $(\boldsymbol{\psi}, b)$?*

**Theorem 1.** ***D-SSVM** is $\mathcal{NP}$-complete for $t \geq 2$.*

*Proof.* It is clear that one can find a separating hyperplane in polynomial type, for example using a standard SVM formulation, to determine whether the positive and negative pattern vectors are separable in a selection $\boldsymbol{\upsilon}^*$, therefore the $D - SSVM$ is in $\mathcal{NP}$. Next, we show that this decision problem is $\mathcal{NP}$-complete for $t \geq 2$ by a reduction from the classical **PARTITION** problem: Given a set of positive integers $S = \{s_1, s_2, \ldots, s_n\}$, does there exist a subset $S' \subseteq S$ such that

$$\sum_{i:s_i \in S'} s_i = \sum_{i:s_i \in S \setminus S'} s_i = \frac{1}{2} \sum_{i=1}^{n} s_i \ ? \tag{3–2}$$

This problem is known to be $\mathcal{NP}$-complete (Garey and Johnson, 1979). Now, let us consider the following equivalent formulation of the **PARTITION** problem: Given a set of $n$ positive integers $S = \{s_1, s_2, \ldots, s_n\}$, does there exist a vector $\mathbf{w} \in \{-1, +1\}^n$, such that $\sum_{i=1}^{n} s_i w_i = 0$?

Suppose we are given an instance of the **PARTITION** problem. Let $d = n + 1$. Let $\mathbf{e}_i$ be a $d$-dimensional vector whose components are zeros except for component $i$, which is

equal to 1. Let $s_+$ and $s_-$ be $d$-dimensional vectors such that $s_+ = (s_1, s_2, \ldots, s_n, 1)$ and $s_- = (s_1, s_2, \ldots, s_n, -1)$.

Next we construct an instance of the D-SSVM problem as follows.

(i) For $i = 1, \ldots, n$ add the sets of vectors, $\{e_i, -e_i\}$ with positive labels, $\{-e_i, e_i\}$ with negative labels.

(ii) Add the sets of vectors $\{e_{n+1}, e_{n+1}\}$ with positive labels, $\{-e_{n+1}, -e_{n+1}\}$ with negative labels.

(iii) Add the sets of vectors $\{s_+, s_+\}$ with positive labels, $\{s_-; s_-\}$ with negative labels.

Note that, regarding item i of the construction, following are the corresponding inequalities in the selective SVM formulation.

$$w_i + b \geq 1 - M(1 - v_{i,1}) \tag{3–3a}$$

$$-w_i + b \geq 1 - M(1 - v_{i,2}) \tag{3–3b}$$

$$v_{i,1} + v_{i,2} = 1 \tag{3–3c}$$

$$w_i - b \geq 1 - M(1 - v'_{i,1}) \tag{3–3d}$$

$$-w_i - b \geq 1 - M(1 - v'_{i,2}) \tag{3–3e}$$

$$v'_{i,1} + v'_{i,2} = 1 \tag{3–3f}$$

It can be verified that (3–3a)-(3–3b) and (3–3d)-(3–3e) have a feasible solution if and only if

$$v_{i,1} = v'_{i,1} = 1 \text{ and } v_{i,2} = v'_{i,2} = 0, \text{ or} \tag{3–4a}$$

$$v_{i,1} = v'_{i,1} = 0 \text{ and } v_{i,2} = v'_{i,2} = 1. \tag{3–4b}$$

From item ii of the construction we have

$$w_{n+1} + b \geq 1 \tag{3–5a}$$

$$w_{n+1} - b \geq 1 \tag{3–5b}$$

From the solution to the system of inequalities above, in order to minimize the objective $\sum_{i=1}^{d} w_i^2$ the values of $w_i$, $i = 1, \ldots, n$, can either be 1 or $-1$, the value of $w_{n+1}$ should be 1, and $b$=0. From item iii of the construction we have

$$\sum_{i=1}^{n} s_i w_i + w_{n+1} + b \geq 1 \tag{3–6a}$$

$$-\sum_{i=1}^{n} s_i w_i + w_{n+1} - b \geq 1 \tag{3–6b}$$

Taking into account our observations above, from (3–6a)-(3–6b) we can conclude that the objective $\sum_{i=1}^{d} w_i^2$ is equal to $d$ if and only if $\sum_{i=1}^{n} s_i w_i = 0$.

The presented reduction is polynomial, therefore, the decision version of the selective SVM problem is $\mathcal{NP}$-complete.

□

**Corollary 1.** *Selective SVM problem (Formulation 3–1) is $\mathcal{NP}$-hard.*

### 3.3 An Alternative Selective SVM Problem

We introduce an alternative approach to the combinatorial selective SVM problem to find good solutions efficiently. The main idea is to provide some flexibility for the hyperplane. This flexibility is provided by *restricted free slack* to decrease the influence of the pattern vectors with small or negative (misclassified) distance from the hyperplane. Given this flexibility, the hyperplane realigns itself with respect to the further pattern vectors with larger margin.

We provide restricted free slack amount of $V$ for all pattern vectors. Note that a very small amount of free slack would make a very small difference compared to the standard SVM formulation, whereas a very large free slack would yield trivial solutions. Depending on the selection scheme, the amount of total free slack may vary. The corresponding formulation is given as follows.

$$\min \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_{i,k}^2 \tag{3–7a}$$

$$\text{subject to} \quad y_i(\langle \boldsymbol{\psi}, \mathbf{x}_{i,k}\rangle + b) \geq 1 - \xi_{i,k} - \upsilon_{i,k} \qquad i = 1,\ldots,n; k = 1,\ldots,t \tag{3–7b}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{t}\upsilon_{i,k} \leq V \tag{3–7c}$$

$$\upsilon_{i,k} \geq 0 \qquad i = 1,\ldots,n; k = 1,\ldots,t. \tag{3–7d}$$

Note that this formulation is similar to the standard SVM formulation with a convex quadratic objective function and linear constraints. The Lagrangian dual of this formulation can also be derived for nonlinear classification. The dual formulation is given as follows.

$$\max \quad \left\{\sum_{i=1}^{n}\sum_{k=1}^{t}\alpha_{i,k} - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{t}\sum_{j=1}^{n}\sum_{l=1}^{t} y_i\ y_j\ \alpha_{i,k}\ \alpha_{j,l}\ \langle \mathbf{x}_{i,k}, \mathbf{x}_{\mathbf{j,l}}\rangle \right.$$
$$\left. -\frac{1}{2C}\sum_{i=1}^{n}\sum_{k=1}^{t}\alpha_{i,k}^2 - \beta V\right\} \tag{3–8a}$$

$$\text{subject to} \quad \sum_{i=1}^{n}\sum_{k=1}^{t} y_i\alpha_{i,k} = 0 \tag{3–8b}$$

$$0 \leq \alpha_{i,k} \leq \beta \qquad i = 1,\ldots,n; \quad k = 1,\ldots,t. \tag{3–8c}$$

From complementary slackness, we can directly find $b$ from a constraint that satisfies $0 < \alpha_{i,k} < \beta$. Kernel induced maps to nonlinear feature spaces can be used by replacing the linear dot product $\langle \mathbf{x}_{i,k}, \mathbf{x}_{\mathbf{j,l}}\rangle$ in (3–8) with a kernel $\mathbf{K}(\mathbf{x}_{i,k}, \mathbf{x}_{j,l})$. Then the classification function is given as $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{n}\sum_{k=1}^{t} y_i\alpha_{i,k}\langle \mathbf{x}, \mathbf{x}_{i,k}\rangle + b)$.

Next, we show that the pooled free slack acquired by each pattern vector is either 0 or linearly proportional to its distance from the optimal hyperplane depending on the total slack provided. For the following Lemma (1), Lemma (2), and Theorem (2), let

$D = (1, \ldots, n) \times (1, \ldots, k)$ be the set of indices for all pattern vectors, and $\boldsymbol{\psi} \neq \mathbf{0}, b, \boldsymbol{\xi}, \boldsymbol{\upsilon}$ be the solution to problem (3–7), with an objective function value $z^*$.

**Lemma 1.** *Constraint (3–7c) is binding, i.e.,* $\sum_{(i,k)\in D} \upsilon_{i,k} = V$ *in the optimal solution.*

*Proof.* Assume that $V - \sum_{(i,k)\in D} \upsilon_{i,k} > 0$ in the optimal solution, where $\boldsymbol{\psi} \neq \mathbf{0}$. From complementary slackness, the corresponding dual variable $\beta = 0$ in the dual formulation (3–8), which forces the dual objective, and thus the primal objective to be 0. This implies $\boldsymbol{\psi} = \mathbf{0}$, thus a contradiction. □

**Lemma 2.** *If* $\xi_{i,k} + \upsilon_{i,k} > 0$*, for some* $(i,k) \in D$ *then the corresponding constraint (3–7b) is binding, i.e.,* $y_i(\langle \boldsymbol{\psi}, \mathbf{x}_{i,k} \rangle + b) = 1 - \xi_{i,k} - \upsilon_{i,k}$.

*Proof.* Assume that $\xi_{i,k} + \upsilon_{i,k} > 0$ and the corresponding constraint is nonbinding, i.e. $y_i(\langle \boldsymbol{\psi}, \mathbf{x}_{i,k} \rangle + b) - 1 + \xi_{i,k} + \upsilon_{i,k} = \Delta > 0$. Then, $\xi'_{i,k} = \xi_{i,k} - \Delta$ contradicts optimality, and $\upsilon'_{i,k} = \upsilon_{i,k} - \Delta$ contradicts Lemma (1). □

**Theorem 2.** *Let,* $\xi_{max} = \max_{(i,k)\in D}\{\xi_{i,k}\}$ *in the optimal solution. Then,*

$$\textit{Let } (i,k) \in D, \quad \textit{then} \quad \xi_{i,k} < \xi_{max} \Rightarrow \upsilon_{i,k} = 0 \quad \textit{and} \quad \upsilon_{i,k} > 0 \Rightarrow \xi_{i,k} = \xi_{max}.$$

*Proof.* Assume that $\boldsymbol{\xi}$ and $\boldsymbol{\upsilon}$ in the optimal solution does not necessarily satisfy Theorem (2). Let,

- $d_{i,k} = \xi_{i,k} + \upsilon_{i,k}$,
- $\upsilon'_{i,k} = \max_{(i,k)\in D}\{d_{i,k} - \xi'_{max}, 0\}$, where $\xi'_{max}$ is such that $\sum_{(i,k)\in D} \upsilon'_{i,k} = V$,
- $\xi'_{i,k} = d_{i,k} - \upsilon'_{i,k}$,
- $\delta_{i,k} = \xi_{i,k} - \xi'_{i,k}$.

Note that $\xi'_{max}$, $\xi'_{i,k}$ and $\upsilon'_{i,k}$ values satisfy Lemmas (1) and (2), Theorem (2), and do not violate any of the constraints in 3–7. It is easy to verify that $\sum_{(i,k)\in D} \delta_{i,k} = 0$.

Let $S \subseteq D$ be the set of indices with $\upsilon'_{i,k} = 0$, and $z' = \|\boldsymbol{\psi}^*\|^2 + \sum_{(i,k)\in D} {\xi'_{i,k}}^2$. The objective function value, $z^* = \|\boldsymbol{\psi}^*\|^2 + \sum_{(i,k)\in D} \xi_{i,k}^2$, can be written as,

$$z^* = z' + \sum_{(i,k)\in S} 2\xi'_{i,k}\delta_{i,k} + \sum_{(i,k)\in D\setminus S} 2\xi'_{i,k}\delta_{i,k} + \sum_{(i,k)\in D} \delta^2_{i,k}. \tag{3–9}$$

Note that $\delta_{i,k} \leq 0\ \forall (i,k) \in D$, by definition, and $\xi'_{i,k} = \xi'_{max}\ \forall (i,k) \in D \setminus S$. Since,

$$\sum_{(i,k)\in S} \xi'_{i,k}\delta_{i,k} \geq \xi'_{max} \sum_{(i,k)\in S} \delta_{i,k},$$

and $\sum_{(i,k)\in D} \delta_{i,k} = 0$, the relationship between $z^*$ and $z'$ is,

$$z^* \geq z' + \sum_{(i,k)\in D} \delta_{i,k}. \tag{3–10}$$

From expression (3–10), $z^*$ can only be optimal if and only if $\delta_{i,k} = 0$, and thus $\xi_{i,k} = \xi'_{i,k}$ and $\upsilon_{i,k} = \upsilon'_{i,k}$ for all $(i,k) \in D$. □

Theorem (2) basically states that all pattern vectors with a functional margin $d_{i,k} = yi(\langle \boldsymbol{\psi}, \mathbf{x}_{i,k} \rangle + b) < 1$ incur penalty for $\xi_{i,k} = \min\{1 - d_{i,k}, \xi_{max}\}$. For pattern vectors $\xi_{i,k} = \xi_{max}$ the free slack is equal to $\upsilon_{i,k} = 1 - \xi_{max} - d_{i,k}$, the sum of which is always equal to $V$. Examples are demonstrated in Figure 3-1.

This result implies, without loss of generality, the free slack for a positive pattern vector is distributed linearly proportional to its distance from the hyperplane $\langle \boldsymbol{\psi}, \mathbf{x}_{i,k} \rangle + b = 1 - \xi_{max}$, as shown in Fig. 3-2. In this figure, free slack for each point is shown in the third dimension. The figure on the left is the top view showing the original data. The figures on the right are front views, only showing the amount of slack assigned.

This result leads to a few possible methods to maximize the margin between the selected points, which are discussed in the next section.

### 3.4 Selection Methods

The solution to the alternative problem allows pattern vectors that are close to the hyperplane to use free slack and provide more flexibility for the separating hyperplane. The selection is done regarding the orientation of the hyperplane. Since we do not use the combinatorial formulation, we refer to the alternative formulation as *soft selection* for the

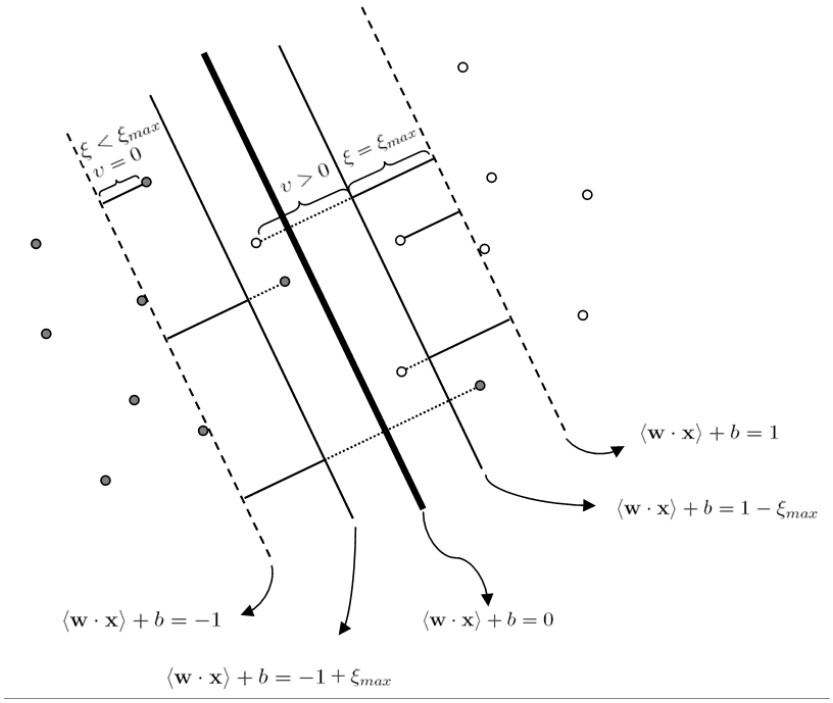



Figure 3-1. Example showing the relationship between penalized slack and free slack

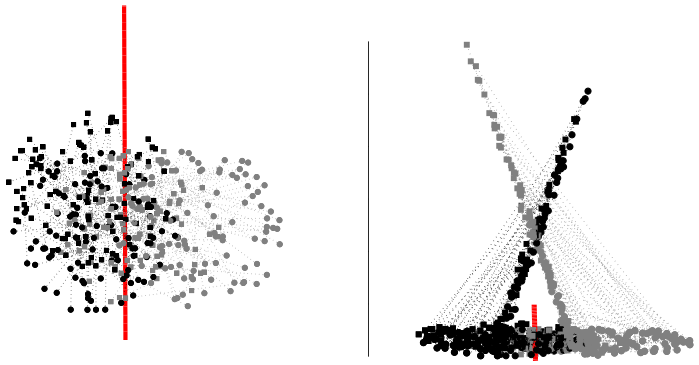



Figure 3-2. Distribution of restricted free slack shown in the third dimension on a two dimensional data: (a) Top view, (b) Front view

remainder of the chapter. The methods introduced in this section are based on the soft selection formulation and the result which states that the amount of free slack acquired by each pattern vector is linearly proportional to its distance from the hyperplane. Two methods are proposed: an iterative elimination method, and a direct selection method.

### 3.4.1 Iterative Elimination

The soft selection formulations are mainly developed to give the separating hyperplane more flexibility and, at the same time, to identify those pattern vectors which are misclassified or very close to the hyperplane. Such pattern vectors require more free slack among the points in their set. We can have a more separated subset of positive and negative pattern vectors if we remove such pattern vectors. Our intuitive basic approach is as follows: at each iteration, supply an incremental amount of free slack of $n$ (1 unit per set), solve the soft selection problem, identify the pattern vector with the minimum distance for each set and remove it, and repeat the iterations with the updated set of pattern vectors until only one pattern vector per set remains. This approach is summarized in Algorithm 1.

**Algorithm 1** Iterative Elimination

$\mathbf{X} \Leftarrow \mathbf{X}(0)$
$t \Leftarrow t(0)$
**while** $t > 1$ **do**
  $\{\boldsymbol{\psi}, b\} \Leftarrow$ SOFT SELECTION$(\mathbf{X}, \mathbf{y}, n)$
  $r \Leftarrow \emptyset$
  **for** $i = 1$ to $n$ **do**
    $k^* = \arg\min_{k=1,\cdots,t} \{y_i(\langle \boldsymbol{\psi}, \mathbf{x}_{i,k} \rangle + b)\}$
    $r \Leftarrow r \cup \mathbf{x}_{i,k^*}$
  **end for**
  $\mathbf{X} \Leftarrow \mathbf{X} \setminus r$
  $t \Leftarrow t - 1$
**end while**
**return** $\mathbf{X}$

In Algorithm 1, $\mathbf{X}(0)$ is the original input of pattern vectors, $\mathbf{y}$ is the vector of labels for each set $\mathbf{X}_i$, $n$ is total the free slack amount provided for the soft selection problem, $(\boldsymbol{\psi}, b)$ is the hyperplane, the amount $y_i(\langle \boldsymbol{\psi}, \mathbf{x}_{i,k} \rangle + b)$ is the distance of $\mathbf{x}_{i,k}$ from the hyperplane $(\boldsymbol{\psi}, b)$, $t(0)$ is the initial number of pattern vectors in each set, and $r$ is the set of pattern vectors to be removed at each iteration. Note that this distance can be negative if the pattern vector is misclassified.

Note that when total free slack is zero, the soft selection problem reduces to a standard SVM problem. Based on this observation, we also consider a naïve elimination method, which is basically the iterative elimination method with zero total free slack at each iteration. The naïve elimination is included to compare the performance of the iterative elimination and direct selection methods to a standard SVM based approach, which does not depend on the soft selection formulation.

### 3.4.2 Direct Selection

The alternative to the iterative elimination method is to provide enough free slack to eliminate $t-1$ points in a single iteration. This time, for each set, we directly select the pattern vector with maximum distance from the hyperplane. The direct selection algorithm can be summarized as solving the soft selection problem with $n(t-1)$ amount of total free slack, and from each set, returning the pattern vector furthest from the resulting hyperplane. The direct selection is summarized in Algorithm 2. The notation is similar to that of Algorithm 1.

**Algorithm 2** Direct Selection

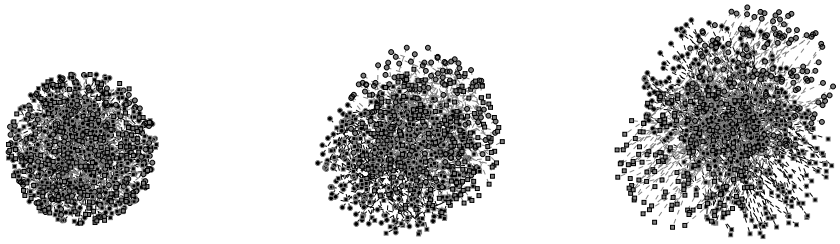
1: $\{\boldsymbol{\psi}, b\} \Leftarrow$ SOFT SELECTION$(\mathbf{X}(0), \mathbf{y}, n(t-1))$
2: $\mathbf{X} \Leftarrow \emptyset$
3: **for** $i = 1$ to $n$ **do**
4: $\quad k^* = \arg\max_{k=1,\cdots,t}\{y_i(\langle \boldsymbol{\psi}, \mathbf{x}_{i,k}\rangle + b)\}$
5: $\quad \mathbf{X} \Leftarrow \mathbf{X} \cup \mathbf{x}_{i,k^*}$
6: **end for**
7: **return** $\mathbf{X}$

## 3.5 Computational Results

In this section, we show the computational results of the proposed methods developed in Section 3.4. We start with the description of how the data is generated and how the performances of the methods are compared. Then, we present comparative results of the iterative elimination method, direct selection method and the naïve elimination method.

### 3.5.1 Simulated Data and Performance Measure

The simulated data is generated using two parameters that determine the *dimensionality* and the *separability* of the pattern vectors. Let $S_k$, $k = 1, \ldots, t$ denote the set of pattern vectors formed by including the $k^{th}$ pattern vector from each set $\mathbf{X}_i$. For each $S_k$, $n$ random pattern vectors are generated, uniformly distributed in a hypersphere with radius $r$. The center of each hypersphere is also distributed uniformly in a hypersphere with radius $c$. We keep $r$ is constant so that $c$ determines the separability of the data, which is the first parameter. The dimension of the data, denoted by $d$, is the second parameter. In Fig. 3-3 three instances with different separability values (a) $c = 0$ (b) $c = r/2$ and (c) $c = r$ are shown for $d = 2$.



(a) (b) (c)

Figure 3-3. 2-D data with separability (a) $c = 0$, (b) $c = r/2$, (c) $c = r$

For the simulated data, the performance measure is the objective function value obtained by the standard SVM formulation for the final set of selected pattern vectors. For this purpose we set the restricted total free slack to zero, in which case we obtain the standard SVM formulation. The results are later normalized for each combination of dimension $d$ and separability $c$ with all of the results obtained from the compared methods. The normalization is done by measuring the mean $\mu_{d,c}$ and the standard deviation $\sigma_{d,c}$ of all the objective function values obtained from all of the compared methods, and normalizing each objective function value using the mean and the standard deviation.

### 3.5.2 Iterative Elimination vs. Naïve Elimination

Simulated data is generated as explained in Section 3.5.1 for $d = 2, 4, ..., 20$ and $c = 0, r/2, r$. Note that $t = 6$ and free slack parameter $p = 1$ (per set). For each combination of the parameters 100 instances of simulated data sets are generated and tested using iterative elimination and naïve elimination. The results are normalized as explained in Section 3.5.1. Let $z_{PFS}$ and $z_N$ denote the average normalized objective function values obtained from iterative elimination and naïve elimination.

In Fig. 3-4, the values $z_N - z_{PFS}$, for $d = 2, 4, \cdots, 20$ are plotted for each $c$ value. It is clear from the figure that as the dimensionality increases the iterative elimination is significantly superior to the naïve elimination method. The difference becomes more apparent for higher levels of data separation. This result clearly shows the success of the iterative elimination due to the flexibility of the separating hyperplane incorporated by the restricted free slack.

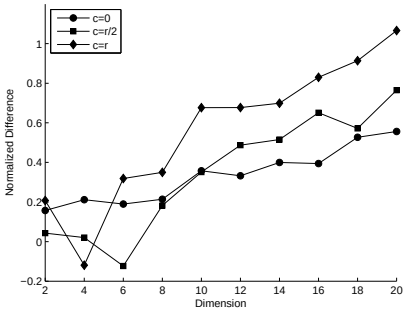
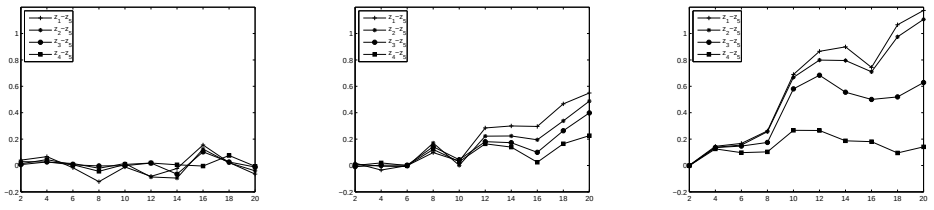



Figure 3-4. Normalized difference between Iterative Elimination and Naïve elimination methods

### 3.5.3 Direct Selection

We generated and evaluated data as explained in Section 3.5.1 for $d = 2, 4, \cdots, 20$ and $c = 0, r/2, r$ for total slack parameter $p = 1, \cdots, 5$ with 100 instances each. There are $t = 6$ pattern vectors in each set. In Fig. 3-5, the effect of the increase in total slack is shown. The three graphs in the figure are in the order of increasing separation in the data. In each graph, the objective function values for the highest total slack parameter

$p = 5$ is assumed to be the base value and the differences between the others and the base, $z_i - z_5$, $i = 1, \cdots, 4$ are graphed. The amount of free slack does not contribute significantly for completely overlapping data in graph (a). However, it is clear from graph (b) that when there is some separability in the data, increasing amount of slack improves the performance of the method for higher dimensions. This difference is even more amplified in graph (c) for higher separability values. In graphs (b) and (c), the increase in the difference between $p = 5$ and the others for higher dimensional data are also apparent. Based on these results, we can conclude that free slack parameter can be set as $t - 1$ (per set) for a dataset with $t$ pattern vectors in each set.



(a) (b) (c)

Figure 3-5. Effect of the amount of free slack on data with separability (a) $c = 0$, (b) $c = r/2$, (c) $c = r$

Next we compare the direct selection method with iterative elimination. Again, that are $t = 6$ pattern vectors in each set. Free slack parameter for direct selection is $p = t - 1$. In Fig. 3-6, the performances of iterative elimination and direct selection are shown with the values $z_{DS} - z_{IE}$, where $z_{IE}$ and $z_{DS}$ are the normalized objective function values obtained from iterative elimination and direct selection methods, respectively. The results fluctuate and there is no significant dominance of one method over the other. However, we observe from the figure that, on the average, the iterative elimination method performs slightly better than the direct elimination method.

### 3.5.4 An Application to a Visuomotor Pattern Discrimination Task

We applied the selective classification methods developed to a real life neuroscience problem. The neural data we study is the local field potentials (LFP) collected from

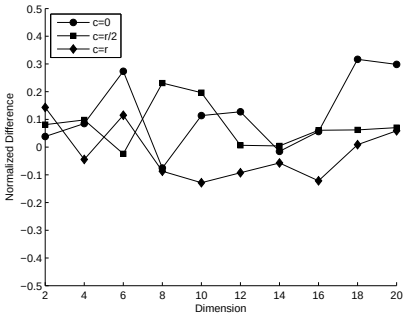



Figure 3-6. Comparison of iterative elimination and direct selection methods

multiple channels implanted in different cortical areas of a macaque monkey during a visual discrimination task. This task involves recognizing a visual *go* stimuli which is followed by a motor response. The visuomotor task is repeated multiple times for the same experiment with different stimuli-response combinations. These differences are grouped as different classes of data for classification. The main objective is to be able to detect these differences over the time course of the task, which requires extensive computational effort to achieve robust results from the multi-dimensional and highly nonlinear neural data.

The visual stimuli are designed to create *lines* and *diamonds*. The go stimuli is chosen to be either lines or diamonds from one session to another. We are interested in detecting different cognitive stages of the visual discrimination task over the time line. We distinguish different sets of labels for each cognitive stage. Three different stages are anticipated: *i)* the detection of the visual stimulus, *ii)* the categorical discrimination of the stimulus, and *iii)* the motor response. The first and the third stages are relatively easy to detect, however the second stage has not been detected in previous studies (Ledberg et al., 2007). This stage involves a complex cognitive process whose onset and length vary over time.

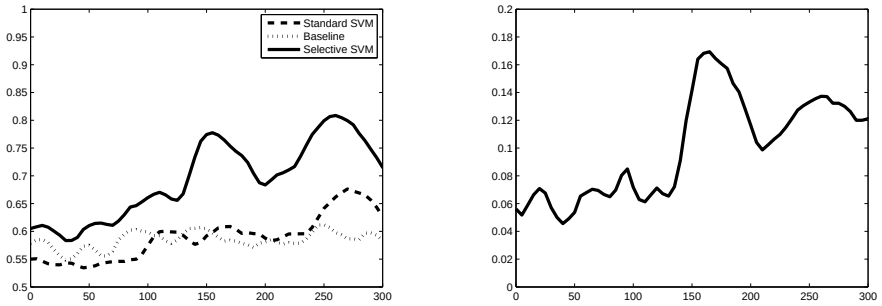
The classification is performed with the patter vectors collected at a specific time $T^*$ from each trial. The classification accuracy obtained from each time point shows the time intervals when the two observed states of the monkey brain are different. However, there are temporal variations in each trial regarding the timing of the observed stages.

The motivation behind the development of selective classification methods is to perform classification while accounting for these temporal variations in the underlying complex cognitive processes. The standard SVM classifier is hindered by the noisy recordings due to these temporal variations. We assume that one recording among the $t$ recordings in a time window from each trial comes from the underlying cognitive process we want to detect. Selecting the most distinguishable recording from each trial at a given time window centered around $T^*$ is a hard problem. Therefore we use the iterative elimination to detect and remove noisy recordings iteratively to achieve better recordings for the given time window.

The data consists of around 4000 trials. Because of the computational limitations of the optimization software (CPLEX 10.1), the entire data could not be processed simultaneously. Therefore we consider 200 trials at a time with equal numbers of positive and negative recordings. Nonlinear iterative elimination method is applied with a window of 3 recordings from each trial for each time point. This window correspond to 15 milliseconds. The recordings with the minimum distance is eliminated from each set at each iteration. This is repeated until there is only one pattern vector remains from each trial.

Each independent batch of 200 trials resulted in a consistently separated cumulative set of selected recordings. The classification accuracy of the selected recordings from each time window is evaluated with the standard SVM classifier using 10-fold classification. In Fig. 3-7 (a), the comparison of the classification accuracy results from iterative elimination and the results from the standard SVM classification. The iterative elimination shows a consistent increase around 10%. This increase can be adjusted by the *baseline* approach. In order to create a baseline, we randomly assign class labels to pattern vectors and apply the iterative elimination methods, so that we can detect the increase in the accuracy for random data and subtract it from the original accuracy results. The baseline is also given in Fig. 3-7 (a). The difference between the original accuracy results and the baseline

results are given in In Fig. 3-7 (b). The peak around 160 milliseconds in this graph is very clear. This result matches the anticipated interval of the categorical discrimination stage. The second peak around 275 milliseconds is too late for the categorical differentiation, however would probably be related to post processing of the categorical difference.




$(a)$ $(b)$

Figure 3-7. Comparative classification accuracy results. $(a)$: Standard SVM, baseline and after applying selective SVM. $(b)$: Difference between the baseline and selective SVM results.

In Fig. 3-8 the results for the feature (channel) selection are presented. We used an SVM based adaptive scaling method for feature selection. This method finds the channels that contribute to SVM classification. When adaptive scaling method is applied over the time line, it produces normalized weight vectors for each time point that can be transfered into a raster plot.

In Fig. 3-8 (a) the results obtained without iterative elimination are presented. In this plot, channels are significantly intermittent over time and the overall picture is not conclusive. The raster plot in Fig. 3-8 (b) shows the results obtained by iterative elimination. Due to the sparseness influence of the adaptive scaling method, we can clearly see the influence of three major channels on the data. We focus on the time intervals around the peaks observed in the classification accuracy graphs. The first peak corresponds to electrode 3, which is around the superior temporal gyrus. Physical damage in temporal lobe is known to impair visual discrimination (Horel and Misantone, 1976; Mendola and Corkin, 1999) and our results agree with the literature. The second peak

corresponds to electrode 10 which is close to the inferior parietal lobe, which also is known to be involved in visual discrimination and have a complementary role with the temporal lobe (Eacott and Gaffan, 1991).

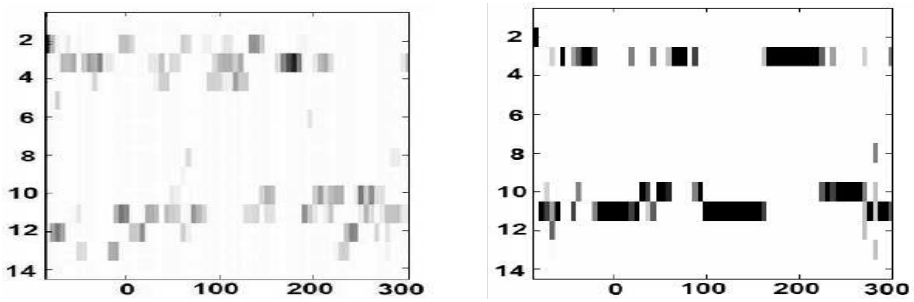



Figure 3-8. Raster plots for the adaptive scaling feature selection method ($a$): after DTW applied, ($b$): after selective SVM applied.

### 3.6 Conclusion

In this chapter, we introduce a novel selective classification method which is a generalization of the standard SVM classifier. Sets of pattern vectors sharing the same label are given as input. One pattern vector is selected from each set in order to maximize the classification margin with respect to the selected positive and negative pattern vectors. The problem of selecting the best pattern vectors is referred to as the *hard selection* problem. The hard selection problem is shown to be $\mathcal{NP}$-hard. We propose alternative linear and nonlinear approaches with tractable formulations, which we call *soft selection* problems. The selective nature of the these formulations is maintained by the restricted free slack concept. The intuition behind this concept is to reverse the combinatorial selection problem by detecting influential pattern vectors which require free slack to decrease their effect on the classification functions. Iteratively removing such pattern vectors, we can find those patter vectors with a larger margin. An *iterative elimination* method is proposed for this purpose. Another alternative approach is to provide enough free slack to identify all $t-1$ out of $t$ pattern vectors to be removed at once, which leads to the *direct selection* method. The iterative elimination and the direct selection methods

are found to produce similar results. Iterative elimination method is also compared with a naïve elimination method which uses standard SVM to eliminate pattern vectors. The results show that iterative elimination is superior to the naïve elimination method both in linear and nonlinear classification.

The motivation for the development of selective classification methods comes from the classification of cognitive states in a visuomotor pattern discrimination task. Due to the temporal noise in the data, the classification results obtained are poor with standard SVM methods. A sliding small time window of recordings are considered as sets of pattern vectors in selective classification. Well separated recordings are selected by the iterative elimination method. The selected recordings are evaluated with standard SVM methods, which result in a significant increase in the classification accuracy over the entire time line of the task. The increase is adjusted by a baseline method which isolates the actual improvement peaks. These peaks clearly mark the categorical discrimination stage of the visuomotor task, which involves a complex cognitive process that has not been detected by previous studies. This result suggest that the proposed selective classification methods are capable of providing promising solutions for other classification problems in neuroscience.

# CHAPTER 4
# MULTIPLE INSTANCE LEARNING VIA MARGIN MAXIMIZATION

In this chapter, we consider the classification problem within the multiple instance learning (MIL) context. Training data is composed of labeled bags of instances. Despite the large number of margin maximization based classification methods, there are only a few methods that consider the margin for MIL problems in the literature. We first formulate a combinatorial margin maximization problem for multiple instance classification and prove that it is $\mathcal{NP}$-hard. We present a way to apply the kernel trick in this formulation for classifying nonlinear multiple instance data. We also propose a branch and bound algorithm and present computational results on publicly available benchmark data sets. Our approach outperforms a leading commercial solver in terms of the best integer solution and optimality gap in the majority of image annotation and molecular activity prediction test cases.

## 4.1 Introduction

Multiple instance learning (MIL) is a supervised machine learning problem. In an MIL problem, instances are considered to be contained in bags and actual instance labels are not available. A bag is classified as a *positive bag* if one or more instances in that bag are positive, otherwise it is classified as a *negative bag*. MIL concept is first introduced in a drug activity prediction problem. In this problem, a molecule has the desired drug effect if at least one of its conformations binds to the target, and no effect is observed otherwise. The goal for the learning algorithm is to identify which conformations (instances) are binding (actual positive) using labeled molecule (bag) information. To generalize this concept, an MIL algorithm takes a training set of labeled bags as input and finds a hypothesis that correctly classifies the bags in the training set, and also predicts the labels of bags whose labels are unknown. MIL has numerous successful implementations in a number of application areas such as drug design (Jain et al., 1994; Dietterich et al., 1997), hard drive failure prediction (Murray et al., 2005), text categorization (Brow et al., 2005),

and content-based image retrieval (Carneiro et al., 2007; Qi and Han, 2007; Chen and Wang, 2004; Chuang et al., 2005).

There is an array of methods proposed for the MIL problem, most of which are hybrids of other well-known methods. A combination of lazy learning and Hausdorff distance is used for the MIL problem in (Wang and Zucker, 2000) with two extensions of k-nearest neighbor (k-NN) algorithm and applications on the drug discovery benchmark data. EM-DD technique, which combines expectation maximization (EM) with the diverse density (DD) algorithm, is proposed in (Zhang and Goldman, 2001). EM-DD is relatively insensitive to the number of features and scales up well to large bag sizes. In (Dooly et al., 2002), extensions of k-NN, citation-kNN, and DD algorithm are proposed with applications to boolean and real valued data.

Margin maximization is the fundamental concept in support vector machine (SVM) classifiers, which is shown to minimize the bound on the generalization error (Vapnik, 1998). An increasing number of methods that involve SVMs have been proposed to solve MIL problems. A generalization of SVM for MIL is introduced in (Andrews et al., 2003). This method is based on a heuristic that iteratively changes the labels of instances in positive bags and uses standard SVM formulation, until a local optimal solution is found. A novel automatic image annotation system that integrates an MIL-based SVM formulation together with a global-feature-based SVM is proposed in (Qi and Han, 2007). For region-based image categorization, a combination of DD and SVM is used in (Chen and Wang, 2004). In this method, a DD function is used to create instance prototypes that represent the instances which are more likely to belong to a bag with a specific label. Instance prototypes are classified using a standard SVM formulation. In (Chen et al., 2006), an instance similarity measure is used to map bags to a feature space. This method lifts the requirement for the existence of at least one positive instance to label a positive bag and uses a 1-norm SVM to eliminate redundant and irrelevant features. A formulation with linear objective and bilinear constraints is proposed to solve multiple

instance classification problems in (Mangasarian and Wild, 2008). Bilinear constraints are handled by an alternating method that uses successive fast linear programs that converge to a local solution in a few iterations. The linear classifiers found by this method are substantially sparse.

Recently, a fast training algorithm, MIL-boost, is proposed to detect objects in images (Viola et al., 2006). This method combines a cascade detector method optimized for MIL within a boost framework. A Bayesian MIL method is introduced in (Raykar et al., 2008), which automatically identifies relevant features and uses inductive transfer to learn multiple classifiers. In (Fung et al., 2007), a method that uses a convex hull representation of multiple instances is shown to perform significantly faster and better on unbalanced data with few positive bags and very large number of negative bags. The convex hull framework applies to most hyperplane based MIL methods.

This chapter mainly focuses on the maximal margin classifiers for MIL. Our goal is to find a hyperplane that maximizes the margin between a selection of instances from each positive bag and all of the instances from negative bags. The formulation proposed for the selection of actual positive instances renders this problem to be $\mathcal{NP}$-hard. A generalization of this formulation is proposed in (Seref et al., 2009), where the selection concept applies to both positive and negative instances. This selective learning method is used to classify neural time-series data. Another similar formulation is introduced within a new supervised learning problem that involves aggregate outputs for training (Musicant et al., 2007). Our main contribution in this study is to introduce the margin maximization formulation and its dual for multiple instance classification, discuss the complexity of the problem and propose a branch and bound algorithm to solve the problem.

The remainder of this chapter is organized as follows: Section 4.2 presents the mathematical formulation with some insights regarding the kernel trick and demonstrates $\mathcal{NP}$-hardness of margin maximization for multiple instance data. Section 4.3 gives the implementation details of our solution approach and Section 4.4 presents the

computational results. In Section 4.5, we provide concluding remarks and directions for future work on this class of problems.

## 4.2 Margin Maximization for Multiple Instance Data

In this section, we introduce a generalization of the above formulation for the multiple instance classification problem.

### 4.2.1 Problem Formulation for Classification of Multiple Instance Data

The formal definition of MIL setting in the context of classification is as follows: Given a set of patterns $\mathbf{x}_1, \ldots, \mathbf{x}_n$ that are grouped into bags $\mathbf{X}_1, \ldots, \mathbf{X}_m$ with $\mathbf{X}_j = \{\mathbf{x}_i : i \in I_j\}$, $I_j \subseteq \{1, \ldots, n\}$, and $\bigcup_j I_j = \{1, \ldots, n\}$; each bag $\mathbf{X}_j$ is associated with a label $y_j \in \{1, -1\}$. These labels are interpreted in the following way: "*If a bag has a negative label, then all patterns in that bag inherit the negative label. On the other hand, if a bag has a positive label, then at least one pattern in that bag is a positive example of the underlying concept.*"

Based on this definition, the maximum margin formulation can be generalized as the following Mixed 0–1 Quadratic Programming problem.

$$\min_{\boldsymbol{\psi}, b, \xi, \eta} \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 \tag{4–1a}$$

$$\text{s.t.} \quad \langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b \geq 1 - \xi_i - M(1 - \eta_i) \quad i \in I^+ \tag{4–1b}$$

$$-\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle - b \geq 1 - \xi_i \quad i \in I^- \tag{4–1c}$$

$$\sum_{i \in I_j} \eta_i \geq 1 \quad j \in J^+ \tag{4–1d}$$

$$\eta_i \in \{0, 1\} \quad i \in I^+ \tag{4–1e}$$

In this formulation, $I^+ = \{i : i \in I_j \wedge y_j = 1\}$, $I^- = \{i : i \in I_j \wedge y_j = -1\}$, and $J^+ = \{j : y_j = 1\}$. Note that, $M$ is a sufficiently large number that ensures that the corresponding constraint is active if and only if $\eta_i = 1$. $\eta_i$ is a binary variable that is 1 if $i$-th instance is one of the actual positive examples of its bag.

Next, we show the application of *kernel trick* for nonlinear multiple instance classification. In order to apply the kernel trick, the dot products of the input patterns are needed. We rewrite formulation (4–1) as

$$\begin{array}{c} \min \\ \eta \\ \sum_{i\in I_j}\eta_i \geq 1 \\ \eta_i \in \{0,1\} \end{array} \quad \min_{\boldsymbol{\psi},b,\xi} \ \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \tag{4–2a}$$

$$\text{s.t.} \quad \langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i - M(1-\eta_i) \quad i \in I^+ \tag{4–2b}$$

$$-\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle - b \geq 1 - \xi_i \qquad\qquad i \in I^-. \tag{4–2c}$$

In this formulation, the outer minimization sets the binary variables, and the inner minimization solves regular 2-norm soft margin problem based on these binary values. Therefore we can write the Lagrangian function for the inner minimization as

$$\begin{aligned} L(\boldsymbol{\psi}, b, \xi, \alpha) = \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 - \sum_{i\in I^-}\alpha_i\left[-\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle - b - 1 + \xi_i\right] \\ - \sum_{i\in I^+}\alpha_i\left[\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b - 1 + \xi_i + M(1-\eta_i)\right]. \end{aligned} \tag{4–3}$$

Differentiating $L$ with respect to the primal variables $\boldsymbol{\psi}$, $b$, and $\xi$, and using stationarity, we obtain

$$\frac{\partial L}{\partial \boldsymbol{\psi}} = \boldsymbol{\psi} - \sum_{i=1}^{n} y_i\alpha_i\mathbf{x}_i = 0; \tag{4–4a}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} y_i\alpha_i = 0; \tag{4–4b}$$

$$\frac{\partial L}{\partial \xi_i} = C\xi_i - \alpha_i = 0. \tag{4–4c}$$

We can substitute the expressions in (4–4) back in the Lagrangian function to obtain the dual formulation, which will give a maximization problem inside the minimization

problem (Mangasarian, 1994). Instead, we substitute the conditions (4–4) inside (4–2) directly:

$$\begin{array}{ll} \min\limits_{\eta} & \min\limits_{\alpha,b} \ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \frac{1}{2C}\sum_{i=1}^{n} \alpha_i^2 \\ \sum_{i \in I_j} \eta_i \geq 1 & \\ \eta_i \in \{0,1\} & \end{array}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{j=1}^{n} y_j \alpha_j \langle \mathbf{x}_j, \mathbf{x}_i \rangle + b \geq 1 - \alpha_i/C - M(1-\eta_i) && i \in I^+ && \text{(4–5a)} \\ & -\sum_{j=1}^{n} y_j \alpha_j \langle \mathbf{x}_j, \mathbf{x}_i \rangle - b \geq 1 - \alpha_i/C && i \in I^- && \text{(4–5b)} \\ & \sum_{i=1}^{n} y_i \alpha_i = 0 && && \text{(4–5c)} \\ & \alpha_i \geq 0 && i = 1, \ldots, n. && \text{(4–5d)} \end{aligned}$$

We finalize the discussion by applying the *kernel trick* on (4–5) and the resulting formulation is

$$\begin{aligned} \min_{\alpha,b,\eta} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{2C}\sum_{i=1}^{n} \alpha_i^2 && && \text{(4–6a)} \\ \text{s.t.} \quad & \sum_{j=1}^{n} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b \geq 1 - \alpha_i/C - M(1-\eta_i) && i \in I^+ && \text{(4–6b)} \\ & -\sum_{j=1}^{n} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) - b \geq 1 - \alpha_i/C && i \in I^- && \text{(4–6c)} \\ & \sum_{i=1}^{n} y_i \alpha_i = 0 && && \text{(4–6d)} \\ & \sum_{i \in I_j} \eta_i \geq 1 && j \in J^+ && \text{(4–6e)} \\ & \eta_i \in \{0,1\} && i \in I^+ && \text{(4–6f)} \\ & \alpha_i \geq 0 && i = 1, \ldots, n. && \text{(4–6g)} \end{aligned}$$

Note that, from Karush-Kuhn-Tucker complementarity conditions, the constraints in the primal problem are binding for those with corresponding dual variable $\alpha_i^* > 0$. It should also be noted that $\eta_i^* = 0$ implies that $\alpha_i^* = 0$ since the corresponding constraint cannot be binding. Knowing $\alpha^*$, we can derive $b^*$ using any $\alpha_i^* > 0$ and (4–6b) or (4–6c).

Next we present the complexity results on margin maximization for multiple instance data.

### 4.2.2 Complexity of the Problem

MIL setting is similar to the concept of Selective Learning introduced in (Seref et al., 2009). Selective learning is originally developed to efficiently solve a time series alignment problem in neural data. However, the problem definition in selective learning is slightly different; the patterns are chosen from each positive and negative set in such a way that the margin between the selected positive and negative pattern vectors is maximized. Selective learning, which is a generalization of MIL[1] , is proved to be $\mathcal{NP}$-hard (Seref et al., 2009). However, this is not enough to prove the complexity of MIL. To the best of our knowledge, there is no formal proof on the complexity of classifying multiple instance data and this section intends to fill this gap.

It is clear that for sufficiently high penalty $C$, formulation (4–1) will provide a separating hyperplane where $\xi_i = 0,\ i = 1, \ldots, n$, if data is linearly separable. Therefore, the decision version of the optimization problem in (4–1) is defined as follows:

**Multiple Instance Learning Decision (MILD) problem:** Given a set of $d$-dimensional patterns $\mathbf{x}_1, \ldots \mathbf{x}_n$ that are grouped into bags $\mathbf{X}_1, \ldots \mathbf{X}_m$ with $\mathbf{X}_j = \{\mathbf{x}_i : i \in I_j\}$, $I_j \subseteq \{1, \ldots, n\}$, and $\bigcup_j I_j = \{1, \ldots, n\}$; each bag $\mathbf{X}_j$ is associated with a label $y_j \in \{1, -1\}$. Is there a selection of at least one positive instance from each bag such that

[1] Multiple instance learning is a special case of selective learning where all negative bags are of size 1 (i.e., no selection is performed over negative bags).

all vectors with negative labels can be separated from selected positive instances with no misclassification by a hyperplane $(\boldsymbol{\psi}, b)$ that satisfies $\frac{1}{2}\|\boldsymbol{\psi}\|^2 \leq n$?

**Theorem 3.** *MILD is $\mathcal{NP}$-complete for bags of size at least 2.*

*Proof.* We show that MILD is $\mathcal{NP}$-complete for bags of size at least 2 by a reduction from the classical PARTITION problem.

The classical PARTITION problem is described as follows: Given a set of positive integers $S = \{s_1, s_2, \ldots, s_n\}$, does there exist a subset $S' \subseteq S$ such that

$$\sum_{i:s_i \in S'} s_i = \sum_{i:s_i \in S \setminus S'} s_i = \frac{1}{2}\sum_{i=1}^{n} s_i \ ? \tag{4–7}$$

This problem is known to be $\mathcal{NP}$-complete (Garey and Johnson, 1979). Next, we consider the following variant of the PARTITION problem.

Given a set of $n$ positive integers $S = \{s_1, s_2, \ldots, s_n\}$, does there exist a vector $\boldsymbol{\psi} \in \{-1, +1\}^d$, such that

$$\sum_{i=1}^{n} s_i \boldsymbol{\psi}_i = 0? \tag{4–8}$$

Suppose we are given an instance of the PARTITION problem. We will add $n$ dummy features and set the dimension of the space $d = 2n$ and construct an instance of the MILD problem as follows:

Let $\mathbf{e}_i$ be a $d$-dimensional vector whose components are zero except component $i$, which is equal to 1.

(i) Add the pattern $(s_1, s_2, \cdots, s_n, 1, 0, \cdots, 0)^T$ with positive label.

(ii) Add the pattern $(s_1, s_2, \cdots, s_n, -1, 0, \cdots, 0)^T$ with negative label.

(iii) Add patterns $\mathbf{e}_{n+1}, \mathbf{e}_{n+2}, \ldots, \mathbf{e}_{2n}$ with positive labels.

(iv) Add patterns $-\mathbf{e}_{n+1}, -\mathbf{e}_{n+2}, \ldots, -\mathbf{e}_{2n}$ with negative labels.

(v) Add $n$ bags with positive labels where bag $i$ consists of patterns $\mathbf{e}_i$ and $-\mathbf{e}_i$ for $i = 1, \ldots, n$.

After this reduction, the corresponding inequalities in (4–1) become

$$\sum_{i=1}^{n} s_i \boldsymbol{\psi}_i + \boldsymbol{\psi}_{n+1} + b \geq 1 \tag{4–9a}$$

$$-\sum_{i=1}^{n} s_i \boldsymbol{\psi}_i + \boldsymbol{\psi}_{n+1} - b \geq 1 \tag{4–9b}$$

$$\boldsymbol{\psi}_i + b \geq 1 \qquad i = n+1, \ldots, 2n \tag{4–9c}$$

$$\boldsymbol{\psi}_i - b \geq 1 \qquad i = n+1, \ldots, 2n \tag{4–9d}$$

$$(\boldsymbol{\psi}_i + b \geq 1) \quad \text{OR} \quad (-\boldsymbol{\psi}_i + b \geq 1) \qquad i = 1, \ldots, n. \tag{4–9e}$$

Note that, $C$ is a sufficiently large number and a hyperplane that has the maximum interclass margin with $\xi_i = 0,\ i = 1, \ldots, n$, is desired.

Let us assert that $b = 0$ and prove the constraints in (4–9) ensure a YES answer for MILD if and only if PARTITION has a YES answer.

It is apparent from (4–9c) and (4–9d) that $\boldsymbol{\psi}_i = 1,\ i = n+1, \ldots, 2n$, and from (4–9e) that $\boldsymbol{\psi}_i \in \{-1, +1\},\ i = 1, \ldots n$, since the goal is to minimize $\|\boldsymbol{\psi}\|^2$ and satisfy $\frac{1}{2}\|\boldsymbol{\psi}\|^2 \leq n$. Using this fact with (4–9a, 4–9b), the answer for MILD is YES if and only if $\sum_{i=1}^{n} s_i \boldsymbol{\psi}_i = 0$ (i.e., PARTITION has a YES answer).

Next, we prove by contradiction that $b = 0$ in the maximum margin solution. Note that, when $b = 0$, the solution described above is feasible with $\boldsymbol{\psi}_i \in \{-1, 1\},\ i = 1, \ldots, n$, and $\boldsymbol{\psi}_i = 1,\ i = n+1, \ldots, 2n$, provided that PARTITION has a YES answer. This separation gives an objective function of $n$. Assume that there is a better solution with $b = \delta \neq 0$. Then (4–9c,4–9d) force $\boldsymbol{\psi}_i \geq 1 + |\delta|,\ i = n+1, \ldots, 2n$, and (4–9e) forces $|\boldsymbol{\psi}_i| \geq 1 - |\delta|,\ i = 1, \ldots, n$. Even if (4–9a,4–9b) are ignored, the objective function value is at least $n + n|\delta|^2$ which is strictly more than $n$, thus a worse solution and a contradiction.

The presented reduction is polynomial. Hence MILD is $\mathcal{NP}$-complete for bags of size at least 2. □

**Corollary 2.** *Maximum margin formulation for MIL (i.e., formulation (4–1)) is $\mathcal{NP}$-hard for bags of size at least 2.*

Next, we prove a stronger complexity result for a special case of the problem.

**Theorem 4.** *MILD is strongly $\mathcal{NP}$-complete for bags of size at least 3.*

*Proof.* We show that MILD is strongly $\mathcal{NP}$-complete for bags of size at least 3 by a reduction from the classical 3SAT problem.

The classical 3SAT problem is described as follows: Given a collection $C = \{c_1, c_2, \ldots, c_m\}$ of clauses on a finite set $U$ of variables such that $|c_i| = 3$ for $1 \leq i \leq m$, is there a truth assignment for $U$ that satisfies all the clauses in $C$?

If $u$ is a variable in $U$, then $u$ and $\bar{u}$ are *literals* over $U$. This problem is known to be strongly $\mathcal{NP}$-complete (Garey and Johnson, 1979).

Suppose we are given an instance of the 3SAT problem. We will set the dimension of the space $d = 2n$ and construct an instance of the MILD problem as follows:

Note that, $\mathbf{e}_i$ is a $d$-dimensional vector whose components are zeros except for component $i$, which is equal to 1.

(i) Add $m$ bags with positive labels for each clause that consists of vectors $\mathbf{e}_i$ for literals $u_i$ and $-\mathbf{e}_i$ for literals $\bar{u}_i$ in the corresponding clause.

(ii) Add patterns $\mathbf{e}_{n+1}, \mathbf{e}_{n+2}, \ldots, \mathbf{e}_{2n}$ with positive labels.

(iii) Add patterns $-\mathbf{e}_{n+1}, -\mathbf{e}_{n+2}, \ldots, -\mathbf{e}_{2n}$ with negative labels.

(iv) Add $n$ bags with positive labels where bag $i$ consists of patterns $\mathbf{e}_i$ and $-\mathbf{e}_i$ for $i = 1, \ldots, n$.

After this reduction, the corresponding inequalities in (4–1) become

$$(\gamma_{il}\boldsymbol{\psi}_i + b \geq 1) \quad \text{OR} \quad (\gamma_{jl}\boldsymbol{\psi}_j + b \geq 1) \quad \text{OR} \quad (\gamma_{kl}\boldsymbol{\psi}_k + b \geq 1) \qquad l = 1, \ldots, m \tag{4–10a}$$

$$\boldsymbol{\psi}_i + b \geq 1 \qquad i = n+1, \ldots, 2n \tag{4–10b}$$

$$\boldsymbol{\psi}_i - b \geq 1 \qquad i = n+1, \ldots, 2n \tag{4–10c}$$

$$(\boldsymbol{\psi}_i + b \geq 1) \quad \text{OR} \quad (-\boldsymbol{\psi}_i + b \geq 1) \qquad i = 1, \ldots, n \tag{4–10d}$$

where $\gamma_{il}$ is 1 if $u_i$ appears in clause $c_l$, and $-1$ if $\bar{u}_i$ appears in clause $c_l$.

Note that, $C$ is a sufficiently large number and a hyperplane that has the maximum interclass margin with $\xi_i = 0,\ i = 1, \ldots, n$, is desired.

Let us assert that $b = 0$ and prove the constraints in (4–10) ensure a YES answer for MILD if and only if 3SAT has a YES answer.

It is obvious from (4–10a) that $\boldsymbol{\psi}_i$ are either greater than 1 or less than $-1$ and the objective of minimizing $\|\boldsymbol{\psi}\|^2$ ensures $\boldsymbol{\psi}_i$ are set to either 1 or $-1$, respectively. It is easy to see that the answer for 3SAT is YES if and only if, $\boldsymbol{\psi}_i = 1$ for variables that are set to TRUE and $\boldsymbol{\psi}_i = -1$ for those that are FALSE.

Next, we prove by contradiction that $b = 0$ in the maximum margin solution. Assume that there is a better solution with $b = \delta \neq 0$. Then (4–10b,4–10c) force $\boldsymbol{\psi}_i \geq 1 + |\delta|$, $i = n+1, \ldots, 2n$, and (4–10d) forces $|\boldsymbol{\psi}_i| \geq 1 - |\delta|$, $i = 1, \ldots, n$. The objective function value is at least $n + n|\delta|^2$ which is strictly more than $n$, thus a worse solution and a contradiction.

The presented reduction is polynomial. Hence MILD is strongly $\mathcal{NP}$-complete for bags of size at least 3. □

**Corollary 3.** *Maximum margin formulation for MIL (i.e., formulation (4–1)) is strongly $\mathcal{NP}$-hard for bags of size at least 3.*

Next, we describe our proposed branch and bound scheme that scales up better than a leading commercial solver.

### 4.3 A Branch and Bound Algorithm for MIL

A typical way to solve a combinatorial problem is via an enumeration tree where the leaves of the tree correspond to feasible solutions that should be examined in complete enumeration. *Branch and bound algorithm*, which uses an intelligent decomposition of the main problem and bound information on an enumeration tree, can help us solve problems that are impossible to solve using complete enumeration (Wolsey, 1998).

In a *minimization* problem with binary variables, the problem is decomposed into two problems at each node of the enumeration tree. These decompositions are obtained by branching on a binary variable whose value is not set. For each node, upper and lower bounds are obtained and a node is *pruned* (i.e., no further decomposition is necessary) if one of the following conditions hold.

- The node is infeasible[2] .
- The upper bound is equal to the lower bound.
- The lower bound is larger than the objective function value of the *incumbent* (i.e., current best) solution.

When the upper bound is equal to the lower bound, a node is *pruned by optimality*, since the optimal solution for this decomposition is known and further decomposition is redundant. A node can also be *pruned by bound*, which implies that it does not suggest a better solution than current best solution.

Upper bounds are obtained from the objective function value of feasible solutions. If a feasible solution is better than the incumbent solution, incumbent is set to that solution. Lower bounds on the other hand, are not necessarily feasible but they give a measure of how promising the decomposition is. Tight bounds lead to more pruning and faster convergence. Good branching strategies are also crucial in a successful branch and bound algorithm. Next, we explore our bounding and branching schemes for MIL problem.

### 4.3.1 Branching Scheme

We will denote binary variables $\eta_i$ that are set for a partial solution by $\eta_i^c$. At an intermediate step where some binary variables are set, we solve the following convex quadratic problem. This problem is a relaxation of the original problem, thus gives a

[2] Note that, in our branch and bound algorithm, a node cannot be pruned by infeasibility since the decompositions (i.e., soft margin classification problems) are always feasible.

lower bound. We consider relaxing the binary variable restrictions to avoid the intense computational burden accompanying other relaxations where binary variables are kept.

$$
\begin{aligned}
z_{LB} = \quad & \min_{\boldsymbol{\psi},b,\xi,\eta} && \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 && && \text{(4–11a)}\\
& \text{s.t.} && \langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i && i \in I^+ \wedge \eta_i^c = 1 && \text{(4–11b)}\\
& && \langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i - M(1-\eta_i) && i \in I_j \wedge j \in J^0 \wedge \eta_i^c \neq 0 && \text{(4–11c)}\\
& && \sum_{i \in I_j \wedge \eta_i^c \neq 0} \eta_i = 1 && j \in J^0 && \text{(4–11d)}\\
& && 0 \leq \eta_i \leq 1 && i \in I_j \wedge j \in J^0 \wedge \eta_i^c \neq 0 && \text{(4–11e)}\\
& && -\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle - b \geq 1 - \xi_i && i \in I^- && \text{(4–11f)}
\end{aligned}
$$

where $J^0$ is the set of positive bags whose actual positive instances are not discovered, i.e., $J^0 = \{j : y_j = 1 \wedge \eta_k^c \neq 1, \forall k \in I_j\}$. It is easy to see that when constraint (4–1d) is changed to equality, the optimal objective function value will not change for (4–1). On the other hand, selection of exactly one data instance per positive bag will significantly reduce the size of the feasible region. Therefore, we use the equality constraint for our lower bounding formulation (4–11). When an instance is selected for a decomposition, constraint (4–11d) will automatically ignore remaining instances that share the same bag, thus avoid redundant computational work.

If the obtained solution is integer feasible (i.e., $\eta_i^* \in \{0,1\}, \forall i : i \in I_j \wedge y_j = 1$) then we can prune the node since upper and lower bounds are equal (i.e., the optimal solution for that decomposition is known). However, we observe that without a careful selection of parameter $M$, the above formulation ignores (4–11c) by setting $0 < \eta_i^* < 1$ and associated $\xi^*$'s are set to 0. Therefore, we check the feasibility of the hyperplane for each undecided bag explicitly. Formally, a node of the branch and bound tree is pruned if the following boolean function is satisfied where $(\boldsymbol{\psi}^*, b^*)$ define the optimal hyperplane obtained from (4–11).

$$\bigwedge_{j \in J^0} \bigvee_{i : i \in I_j} \langle \boldsymbol{\psi}^*, \mathbf{x}_i\rangle + b^* \geq 1 \qquad \text{(4–12)}$$

If (4–12) is not satisfied, then *branching* is performed on $\eta_k$ where

$$k = \arg\max_{i:i\in I_{j^0}} \langle \boldsymbol{\psi}^*, \mathbf{x}_i \rangle + b^* \tag{4–13}$$

and

$$j^0 = \arg\min_{j\in J^0} \max_{i:i\in I_j} \langle \boldsymbol{\psi}^*, \mathbf{x}_i \rangle + b^*. \tag{4–14}$$

The problem is decomposed into two subproblems with additional constraints $\eta_k = 1$ and $\eta_k = 0$, respectively. The aim here is to branch on the critical bag $I_{j^0}$ that is currently misclassified or closest to being misclassified based on $(\boldsymbol{\psi}^*, b^*)$. (4–14) selects the critical bag whereas (4–13) selects the most promising instance from that bag.

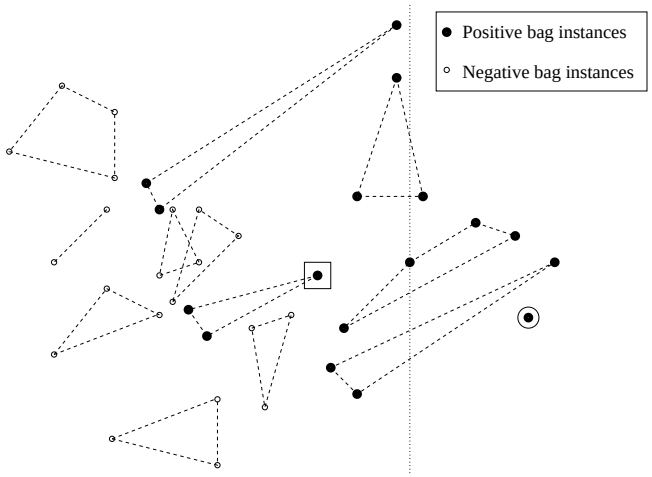



Figure 4-1. An example of critical bag.

Consider the example in Fig. 4-1. The algorithm starts by solving the relaxation in (4–11). There is one (circled) instance in one of the positive bags which should be selected and that solution defines the lower bound. The separating hyperplane for the relaxation is shown as a dotted line. The bag whose best instance is the most misclassified is considered next. Branching is performed on the most promising instance in square. For the first decomposition where the instance in square is selected, the corresponding node can be pruned by optimality since (4–12) is satisfied. When other instances in this bag are considered as actual positive, the lower bounds are larger, thus the optimal solution is

obtained. All instances in this bag should be checked in order to conclude optimality if $M$ is too large. In order to achieve optimal solutions quickly, we start with the critical bag and the instances of the critical bag that are promising (i.e., the least misclassified).

### 4.3.2 Bounding Scheme

To obtain an upper bound, we employ a two phase heuristic approach. In the first step, we find the optimal separating hyperplane considering the previous decisions (i.e., $\eta^c$ values) and all undecided bags. In the second step, we re-optimize based on a temporary selection of actual positive instances. Formally, the first phase solves the following problem.

$$\min_{\boldsymbol{\psi},b,\xi} \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \tag{4–15a}$$

$$\text{s.t.} \quad \langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i \qquad i \in I^+ \wedge \eta_i^c = 1 \tag{4–15b}$$

$$\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i \qquad i \in I_j \wedge j \in J^0 \wedge \eta_i^c \neq 0 \tag{4–15c}$$

$$-\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle - b \geq 1 - \xi_i \qquad i \in I^- \tag{4–15d}$$

For each undecided bag, we select the instance that is furthest away from the optimal hyperplane obtained from (4–15). Set $S$ of selected instances is defined as

$$S = \{s_j : s_j = \arg\max_{i \in I_j \wedge \eta_i^c \neq 0} \langle \boldsymbol{\psi}^*, \mathbf{x}_i\rangle + b^*, j \in J^0\} \tag{4–16}$$

where $(\boldsymbol{\psi}^*, b^*, \xi^*)$ define the optimal solution for (4–15).

The second phase computes the upper bound by solving the margin maximization problem based on this temporary selection.

$$z_{UB} = \min_{\boldsymbol{\psi},b,\xi} \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \tag{4–17a}$$

$$\text{s.t.} \quad \langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i \qquad i \in I^+ \wedge \eta_i^c = 1 \tag{4–17b}$$

$$\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i \qquad i \in S \tag{4–17c}$$

$$-\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle - b \geq 1 - \xi_i \qquad i \in I^- \tag{4–17d}$$

Next, we present the computational results and show the performance of our branch and bound algorithm on public data sets.

### 4.4 Computational Study

To demonstrate the capabilities of our algorithm, we report results on data sets from (Asuncion and Newman, 2007) and (Andrews et al., 2002). Two data sets from (Asuncion and Newman, 2007) represent the molecular activity prediction data sets. Molecules judged by human experts are labeled as musks or non-musks. The goal for MIL is to discriminate these two categories given the exact shape and conformation of each molecule. Three data sets from (Andrews et al., 2002) correspond to an image annotation task where the goal is to determine whether or not a given animal is present in an image. Color images from Corel data set are segmented with Blobworld system. Set of segments in each picture are characterized by color, shape, and texture descriptors. The sizes of these data sets are presented in Table 4-1.

| **Data set** | **Features (Nonzero)** | **+ Bags** | **+ Instances** | **- Bags** | **- Instances** |
|---|---|---|---|---|---|
| Musk1 | 166 | 47 | 207 | 45 | 269 |
| Musk2 | 166 | 39 | 1017 | 63 | 5581 |
| Elephant | 230(143) | 100 | 762 | 100 | 629 |
| Fox | 230(143) | 100 | 647 | 100 | 673 |
| Tiger | 230(143) | 100 | 544 | 100 | 676 |

Table 4-1. Size information for the Molecular Activity Prediction and the Image Annotation Data Sets

All computations are performed on a 3.4 GHz Pentium IV desktop computer with 2.0 Gb RAM. The algorithms are implemented in C++ and used in conjunction with MATLAB 7.3 environment in which the data resides. In our algorithm, we solved the convex minimization problems (i.e., formulations (4–11), (4–15), and (4–17)) using CPLEX 10.1 (ILOG, 2008). For benchmarking purposes, formulation (4–1) is solved using CPLEX 10.1 with default settings. In all experiments, trade-off parameter $C$ between training error and margin is set to $(\sum\langle x, x\rangle/n)^{-1}$, which is scaled based on the input vector.

In our attempt to find the global minimum for formulation (4–1), we report the best integer solution obtained (i.e., UB), optimality gap (i.e., UB-LB) and solution times instead of the prediction accuracy results for generalization. In cases where an algorithm terminates with optimality in the given timeframe, the lower bound is equal to the upper bound (i.e., incumbent solution), thus zero optimality gap.

In order to show the computational limitations of exact algorithms, all instances are obtained by a random feature and bag selection. Because the number of instances is restricted, the last bag selected might not have all instances from the original data set. The results show that when the number of instances increases, our algorithm outperforms CPLEX in terms of the best objective function value. However, when the number of features increases, there is additional computational task at each node of branch and bound tree that might deteriorate the performance of our implementation. Nevertheless, feature selection can be used to scale the problem whereas the instances are crucial.

| | $n$ | $\lvert J^+\rvert$ | $d$ | Our B&B Scheme | CPLEX 10.1 |
|---|---|---|---|---|---|
| ELEPHANT | 20 | 2 | 10 | 0.04 | 0.01 |
| | 20 | 2 | 5 | 0.01 | 0.01 |
| | 40 | 3 | 10 | 0.14 | 0.03 |
| | 40 | 3 | 5 | 0.20 | 0.03 |
| | 80 | 6 | 10 | 259.29 | 1.95 |
| | 80 | 6 | 5 | 91.56 | 3.00 |

| | $n$ | $\lvert J^+\rvert$ | $d$ | Our B&B Scheme | CPLEX 10.1 |
|---|---|---|---|---|---|
| FOX | 20 | 2 | 10 | 0.17 | 0.01 |
| | 20 | 2 | 5 | 0.14 | 0.01 |
| | 40 | 3 | 10 | 0.89 | 0.06 |
| | 40 | 3 | 5 | 0.45 | 0.01 |
| | 80 | 6 | 10 | 231.81 | 9.29 |
| | 80 | 6 | 5 | 618.01 | 86.87 |

| | $n$ | $\lvert J^+\rvert$ | $d$ | Our B&B Scheme | CPLEX 10.1 |
|---|---|---|---|---|---|
| TIGER | 20 | 2 | 10 | 0.20 | 0.01 |
| | 20 | 2 | 5 | 0.03 | 0.01 |
| | 40 | 4 | 10 | 0.26 | 0.01 |
| | 40 | 4 | 5 | 0.20 | 0.05 |
| | 80 | 8 | 10 | 265.71 | 12.18 |
| | 80 | 8 | 5 | 399.95 | 36.23 |

Table 4-2. Time (in seconds) to achieve the optimal solution for Our Branch and Bound Scheme vs. CPLEX Default Branch and Bound Algorithm for the Image Annotation Data

Table 4-2 shows the performance of exact algorithms for small test instances. The computation times to achieve optimal solutions are presented with different data sets and implementations. As seen on this table, CPLEX outperforms our branch and bound

scheme in small instances due to its preprocessing power and fast implementation at each node of the tree. Note that, neither our algorithm nor CPLEX is able to solve instances with more than 120 data instances to optimality in 3600 seconds.

Next, we consider larger problem sets. Tables 4-3 and 4-4 present benchmark results for our branch and bound implementation and CPLEX default implementation with time limits of 3 and 30 minutes, respectively. In these tests, all instances from the molecular activity prediction data set are used and a random feature selection is performed. Number of features selected is denoted by $d$.

Tables 4-3 and 4-4 show that our algorithm achieves better solutions than CPLEX in all tests. However, the lower bounds obtained by CPLEX are tighter. Musk2 is not used in our computational studies because only nonlinear classifiers are used on this data set in the literature.

| | Our B&B Scheme | | | CPLEX 10.1 | | |
|---|---|---|---|---|---|---|
| $d$ | UB | UB-LB | Time | UB | UB-LB | Time |
| 5 | **10304.05** | 10029.83 | 180 | 11263.03 | **9714.50** | 180 |
| 10 | **10802.55** | **10801.06** | 180 | 12259.66 | 11082.57 | 180 |

Table 4-3. Computational Results for Our Branch and Bound Scheme vs. CPLEX Default Branch and Bound Algorithm for the Molecular Activity Prediction Data (Musk1) with 3 minutes time limit.

| | Our B&B Scheme | | | CPLEX 10.1 | | |
|---|---|---|---|---|---|---|
| $d$ | UB | UB-LB | Time | UB | UB-LB | Time |
| 5 | **11876.10** | 11104.30 | 1800 | 13305.71 | **10612.31** | 1800 |
| 10 | **10178.45** | 10087.73 | 1800 | 11691.09 | **9367.82** | 1800 |

Table 4-4. Computational Results for Our Branch and Bound Scheme vs. CPLEX Default Branch and Bound Algorithm for the Molecular Activity Prediction Data (Musk1) with 30 minutes time limit.

Next, we study the image annotation data. In order to observe how the algorithms scale up, instance selection is performed as well as feature selection. Number of instances is denoted by $n$ and number of positive bags is denoted by $|J^+|$.

Table 4-5 shows that our algorithm scales up well and obtains generally better solutions than CPLEX for larger problems in 3 minutes. There are cases where CPLEX

| Data Set | $n$ | $\|J^+\|$ | $d$ | Our B&B Scheme | | | CPLEX 10.1 | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | UB | UB-LB | Time | UB | UB-LB | Time |
| ELEPHANT | 400 | 26 | 20 | **974.12** | **767.45** | 180 | 986.23 | 787.63 | 180 |
| | 400 | 26 | 10 | **3065.92** | **3064.59** | 180 | 3425.26 | 3230.09 | 180 |
| | 400 | 26 | 5 | **3072.25** | 3072.24 | 180 | 3305.80 | **2915.97** | 180 |
| | 800 | 50 | 20 | **3792.26** | **3792.22** | 180 | 4397.07 | 4295.40 | 180 |
| | 800 | 50 | 10 | **6272.77** | **6272.77** | 180 | 6757.60 | 6563.97 | 180 |
| | 800 | 50 | 5 | **6557.27** | **6557.27** | 180 | 7501.58 | 7308.48 | 180 |
| | 1200 | 78 | 20 | **6585.39** | **6585.39** | 180 | 9637.13 | 9637.13 | 180 |
| | 1200 | 78 | 10 | **10062.24** | **10062.24** | 180 | 11072.95 | 11072.95 | 180 |
| | 1200 | 78 | 5 | **9952.44** | **9952.44** | 180 | 11821.95 | 11631.28 | 180 |
| FOX | 400 | 33 | 20 | **3282.89** | 3088.27 | 180 | 3388.99 | **3017.85** | 180 |
| | 400 | 33 | 10 | 4751.69 | 4548.62 | 180 | **4578.98** | **3999.80** | 180 |
| | 400 | 33 | 5 | **4532.63** | 4239.47 | 180 | 4558.33 | **3977.54** | 180 |
| | 800 | 63 | 20 | 8792.20 | 8792.20 | 180 | **8618.59** | **8429.70** | 180 |
| | 800 | 63 | 10 | 10216.73 | 10050.18 | 180 | **9517.32** | **9321.82** | 180 |
| | 800 | 63 | 5 | 10045.32 | 9878.48 | 180 | **9681.97** | **9485.00** | 180 |
| | 1200 | 93 | 20 | **13034.06** | **13034.06** | 180 | 15440.33 | 15417.24 | 180 |
| | 1200 | 93 | 10 | 15395.31 | 15395.22 | 180 | **14486.01** | **14309.68** | 180 |
| | 1200 | 93 | 5 | 15547.77 | 15380.59 | 180 | **14653.29** | **14456.20** | 180 |
| TIGER | 400 | 33 | 20 | 1699.07 | 1699.01 | 180 | **1562.03** | **1484.61** | 180 |
| | 400 | 33 | 10 | **2886.13** | 2693.32 | 180 | 3058.04 | **2679.99** | 180 |
| | 400 | 33 | 5 | **3287.02** | 3093.72 | 180 | 3422.92 | **3033.86** | 180 |
| | 800 | 71 | 20 | **4761.77** | **4761.77** | 180 | 5472.10 | 5345.19 | 180 |
| | 800 | 71 | 10 | **6946.20** | **6946.20** | 180 | 7353.32 | 6953.16 | 180 |
| | 800 | 71 | 5 | **8519.69** | 8519.59 | 180 | 8898.57 | **8157.97** | 180 |
| | 1144 | 100 | 20 | **7480.61** | **7453.07** | 180 | 10433.51 | 10176.86 | 180 |
| | 1144 | 100 | 10 | **10522.63** | **10250.09** | 180 | 12190.93 | 11805.41 | 180 |
| | 1144 | 100 | 5 | **11994.36** | **11605.10** | 180 | 12774.59 | 11997.72 | 180 |

Table 4-5. Computational Results for Our Branch and Bound Scheme vs. CPLEX Default Branch and Bound Algorithm for the Image Annotation Data with 3 minutes time limit.

performs better but in these cases the differences are subtle. Table 4-6 shows that when the time limit is increased to 30 minutes, our algorithm still achieves better solutions in the majority of tests. There might be cases where the best solution found by an algorithm is optimal but there are active nodes that have lower bounds less than the incumbent solution, therefore optimality is not guaranteed. We do not report the number of remaining active nodes explicitly. However, it should be noted that CPLEX has significantly more number of active nodes than our algorithm on the average. It should

| Data Set | $n$ | $\lvert J^+\rvert$ | $d$ | Our B&B Scheme | | | CPLEX 10.1 | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | UB | UB-LB | Time | UB | UB-LB | Time |
| ELEPHANT | 400 | 26 | 20 | 711.05 | **132.45** | 1800 | 711.05 | 293.41 | 1800 |
| | 400 | 26 | 10 | 2956.12 | 2954.31 | 1800 | **2924.76** | **2482.72** | 1800 |
| | 400 | 26 | 5 | 3037.73 | 3037.57 | 1800 | **3022.99** | **2442.02** | 1800 |
| | 800 | 50 | 20 | **3482.16** | **3482.11** | 1800 | 4379.25 | 4193.66 | 1800 |
| | 800 | 50 | 10 | **6272.77** | **6272.58** | 1800 | 6594.63 | 6397.13 | 1800 |
| | 800 | 50 | 5 | **6540.22** | **6540.20** | 1800 | 7092.43 | 6707.46 | 1800 |
| | 1200 | 78 | 20 | **6585.39** | **6585.39** | 1800 | 7637.07 | 7470.57 | 1800 |
| | 1200 | 78 | 10 | **10062.24** | **10062.24** | 1800 | 10564.25 | 10370.84 | 1800 |
| | 1200 | 78 | 5 | **9874.41** | **9874.41** | 1800 | 11599.74 | 11402.11 | 1800 |
| FOX | 400 | 33 | 20 | 3130.62 | 2919.98 | 1800 | **3008.31** | **2553.51** | 1800 |
| | 400 | 33 | 10 | 4115.66 | 3886.59 | 1800 | **4074.98** | **3468.40** | 1800 |
| | 400 | 33 | 5 | **4504.43** | 4007.91 | 1800 | 4543.71 | **3773.47** | 1800 |
| | 800 | 63 | 20 | **8246.68** | 8246.56 | 1800 | 8406.36 | **8212.65** | 1800 |
| | 800 | 63 | 10 | **9121.37** | 9121.26 | 1800 | 9402.43 | **9020.16** | 1800 |
| | 800 | 63 | 5 | **9387.90** | 9175.31 | 1800 | 9539.56 | **9154.95** | 1800 |
| | 1200 | 93 | 20 | **13034.06** | **13034.06** | 1800 | 13588.41 | 13293.90 | 1800 |
| | 1200 | 93 | 10 | 14532.07 | 14531.79 | 1800 | **14419.72** | **14222.29** | 1800 |
| | 1200 | 93 | 5 | 14849.72 | 14650.02 | 1800 | **14639.85** | **14246.02** | 1800 |
| TIGER | 400 | 33 | 20 | 1429.96 | 1429.68 | 1800 | **1425.15** | **1208.82** | 1800 |
| | 400 | 33 | 10 | 2785.38 | 2589.39 | 1800 | **2765.82** | **2061.83** | 1800 |
| | 400 | 33 | 5 | **3287.02** | **2971.33** | 1800 | 3381.63 | 2973.43 | 1800 |
| | 800 | 71 | 20 | **4705.98** | 4705.98 | 1800 | 4813.83 | **4653.93** | 1800 |
| | 800 | 71 | 10 | **6943.98** | 6943.83 | 1800 | 7156.77 | **6530.96** | 1800 |
| | 800 | 71 | 5 | **8099.99** | 7903.04 | 1800 | 8307.19 | **7808.33** | 1800 |
| | 1144 | 100 | 20 | **7480.61** | 7447.46 | 1800 | 7973.10 | **7347.56** | 1800 |
| | 1144 | 100 | 10 | **10522.63** | 10250.09 | 1800 | 11225.02 | **10107.72** | 1800 |
| | 1144 | 100 | 5 | **11193.67** | **10803.19** | 1800 | 12202.86 | 11174.82 | 1800 |

Table 4-6. Computational Results for Our Branch and Bound Scheme vs. CPLEX Default Branch and Bound Algorithm for the Image Annotation Data with 30 minutes time limit.

also be noted that, lower bounds obtained by CPLEX are generally better than that of our implementation.

The bags are harder to separate when the number of features decreases. Therefore, the optimality gap with less number of features is usually larger. Tables 4-5 and 4-6 show that our algorithm usually finds better solutions than CPLEX despite larger optimality gap.

Table 4-7 summarizes the results for cases where an optimal solution is not achieved. # denotes the number of tests an algorithm outperforms the other. Average and largest

| | Best Solution | | Optimality Gap | |
|---|---|---|---|---|
| | Our B&B | CPLEX 10.1 | Our B&B | CPLEX 10.1 |
| # | **43** | 15 | 25 | **33** |
| AVG | **10.13**% | 3.19% | **15.32**% | 7.07% |
| BEST | **46.34**% | 8.77% | **121.53**% | 25.59% |

Table 4-7. Benchmark results for tests with time limits.

improvements achieved by an algorithm over the other are denoted by AVG and BEST, respectively. As seen on the table, our algorithm achieves significantly better solutions than CPLEX in general. Although optimality gap for CPLEX is smaller than our algorithm in 33 of 58 tests, the average improvement is relatively small. On the other hand, when our algorithm has a smaller optimality gap, the improvement over CPLEX is much more significant.

To sum up, when the number of problem instances is small and number of features is large, CPLEX default implementation can be more suitable because of its preprocessing power. Our algorithm, on the other hand, outperforms CPLEX for practical cases, where number of instances is large and feature selection is applied.

## 4.5 Concluding Remarks

This chapter presents the mathematical formulation, kernel trick application, complexity results, and a branch and bound algorithm for linear classification through margin maximization for multiple instance data. Experimental results show additional benefits of intelligent bounding and branching schemes. Our branch and bound algorithm outperforms a leading commercial solver for practical cases where the number of instances increases. We observe that the proposed heuristic gives tight upper bounds, but the lower bounding scheme needs to be improved. The lower bounding technique we propose helps mostly with pruning by optimality, but rarely with pruning by bound.

An interesting future study might be the selection of $M$ in formulation (4–1) based on input data. This number should satisfy the selection criteria, but it should be small enough to have tight lower bounds with the relaxations as well. Alternatively, $M$ selection

can be avoided by an alternative formulation and a performance benchmark for different formulations can be investigated.

# CHAPTER 5
# SUPPORT VECTOR REGRESSION WITH MULTIPLE INSTANCE DATA

This chapter presents a linear regression framework and a solution approach for multiple instance (MI) data. Introduced in the context of drug activity prediction, MI learning is a generalization of supervised learning methods. In this setting, learning methods are performed over the *bags* of pattern vectors instead of individual instances. This setting is particularly useful when there is ambiguity in the data set such as noise in clinical measurements or uncertainty on the binding conformation in a drug.

## 5.1 Introduction

Regression is a statistical learning technique which develops a mathematical formula that fits the data. Regression can be used for hypothesis testing, forecasting, inference, and modeling of relationships. Regression analysis is utilized in various circumstances and its significance in biomedical research is shown amid subsequent cases in point. For example, a Magnetic Resonance Image (MRI) Based Parcellation Method has been modified to present the reliable and reproducible regions of human brain. Regression analysis is used to demonstrate that the proposed method is more reliable than conventional MRI methods (see (Yoon et al., 2003)). A different study applies local regression to assess Esophageal Pressure in Gastroesophageal Reflux Disease (GERD). The results from both extensive simulations and real data demonstrate ability of local regression to characterize the pressure, which is consistent with the clinical observation (see (Liang and Chen, 2005)). In another biomedical study, regression analysis is used to evaluate smoke carcinogen deposition in a multi-generation human replica (see (Robinson et al., 2006)). Also, in a study of Fractional Brownian Motion (FBM), regression methods are compared for estimation accuracy on synthesized datasets (see (Russell and Akay, 1996)). Advanced techniques, such as multiple regression, permit use of more than one input variable and allow for the fitting of further complex models (e.g., quadratic equations).

Support Vector Regression (SVR) is an optimization based regression framework for solving machine learning problems, based on Support Vector Machines. Support Vector Machine (SVM) is a machine learning technique based on the statistical learning theory, which implements the structural risk minimization principle to obtain a good generalization from limited size data sets (see (Vapnik, 1995)).

SVR approach is based on estimation of a linear function in a kernel induced feature space. The objective is to optimize a certain boundary to the optimal regression line, therefore, errors within a certain distance ($\varepsilon$) of predicted value are disregarded. The learning algorithm minimizes a convex functional with sparse solution comparable to classification technique. For improved illustration, this can be considered a hyper-tube (insensitive band) about a linear function in the kernel induced nonlinear space, such that pattern vectors in this tube are assumed not to contribute any error. Fig. 5-1 shows the insensitive band for a one dimensional linear regression problem.

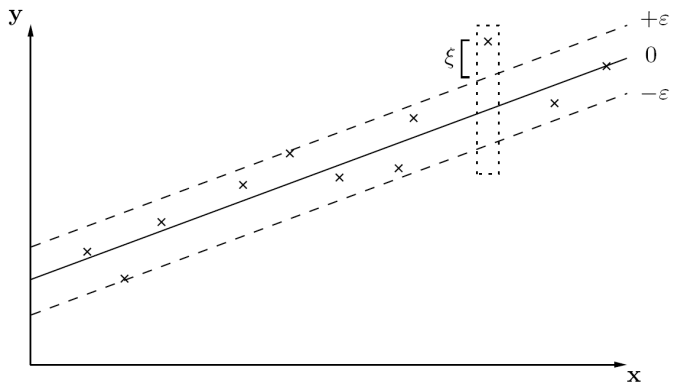



Figure 5-1. The $\varepsilon$-insensitive band for a linear regression problem.

This form of regression is called $\varepsilon$-insensitive because any point in the $\varepsilon$ of the anticipated regression function does not contribute an error. An important motivation for considering the $\varepsilon$-insensitive loss function is the sparseness of the dual variables similar to the case with SVM classifiers. The idea of representing the solution by means of a small subset of training points has enormous computational advantages. Furthermore, it ensures

the existence of a global minimum and the optimization of a reliable generalization bound (see (Cristianini and Shawe-Taylor, 2000)).

SVR has various applications in numerous technology (see e.g., (Sakhanenko and Luger, 2006), (Bergeron et al., 2005)), analytical (see e.g., (Lauer and Bloch, 2008), (Hyunsoo et al., 2005)), and scientific fields (see e.g., (Sun et al., 2004), (Yamamoto et al., 2006)). (Wu et al., 2007) performs location estimation using the Global System for Mobile communication (GSM) based on an SVR approach which demonstrates promising performances, especially in terrains with local variations in environmental factors. SVR method is also used in agricultural schemes in order to enhance output production and reduce losses (see e.g., (Xie et al., 2008), (Li et al., 2007), (Pai and Hong, 2007), (Choy and Chan, 2003)). Based on statistical learning theory, SVR has been used to deal with forecasting problems. Performing structural risk minimization rather than minimizing the training errors, SVR algorithms have better generalization ability than the conventional artificial neural networks (see (Hong and Pai, 2007)).

Occasionally all points within a dataset cannot determine the regression function distinctively. For example, one of the several feature vector encodings may be known to contribute a certain outcome, however, it may not be possible to identify which one. Therefore, it is beneficial to discover a regression function that consider *bags* of data points.

The main approach is to forecast value of a dependent variable, using regression facts, meant for datasets in which multiple instance features are at hand. For instance, in a drug that is known to be helpful for a certain disease, it is desired to discriminate the molecules that bind the target from useless ones. Numerous molecule configurations may share similar molecules in a dynamic balance. Experimental activity will be a function of one or more of these configurations; however, it is usually not viable to establish which one. Additionally, seldom is the condition that all configurations contribute to the experimental

activity. Therefore, it is enviable to take the multiple instance character of these cases into consideration and study a classifier with corresponding property (see (Ray, 2005)).

Multiple instance learning (MIL) problems are introduced by (Balasundaram et al., 2005) in the context of drug activity prediction. These problems are analyzed and studied using various proposed algorithms in the literature. Support Vector Machines are modified to express multiple instance problems by altering the kernels or the objective function (see e.g., (Andrews et al., 2003), (Gartner et al., 2002)). Gaussian notions are studied using a Diverse Density approach (see (Maron, 1998)). Further algorithms intended for extended MIL problems are introduced with a shifting time window approach for hard drive failure prediction (referred to as "Regular Supervised Learning Techniques") (see (Murray et al., 2005)). (Seref et al., 2007) employed a similar shifting time window approach and a selective learning technique to detect categorical discrimination in a visuomotor task performed by a macaque monkey. This selective learning technique is a generalization of MIL framework where the negative bag representations are different in that at least one instance from each negative bag is true negative (see (Seref et al., 2009)).

Multiple instance regression problems occur in an array of new areas. Numerous functions of multiple instance studies prefer real numbers as forecast values. To exemplify, in drug activity prediction, drug designers desire forecasted activity stages of the molecules to be articulated as real number values rather than anticipating active or inactive categorization of these molecules.

Studies are prepared to understand computational intricacy innate to multiple instance regression problems. Examples of such studies include protein family modeling (see (Tao et al., 2004)), stock prediction (see (Maron, 1998)), content-based image retrieval (see (Maron and Ratan, 1998)), and text classification (see (Andrews et al., 2003)).

The remainder of the chapter is organized as follows. Section 5.2 describes the formulation for the multiple instance support vector regression (MI-SVR) problem. Section 5.3 presents the exact solution approach to find the regression function in this setting.

Section 5.4 demonstrates computational results for comparison purposes. Section 5.5 reveals the conclusion and future research directions.

## 5.2 Problem Formulation

In this section, we extend the fundamental support vector regression problem for multiple instance data: Given a set of patterns $\mathbf{x}_1, \dots \mathbf{x}_n$ that are grouped into bags $X_1, \dots X_m$ with $X_j = \{\mathbf{x}_i : i \in I_j\}$, $I_j \subseteq \{1, \dots, n\}$, and $\bigcup_j I_j = \{1, \dots, n\}$; each bag $X_j$ is associated with a label $y_j \in \mathbb{R}$. These labels are interpreted in the following way: "*One pattern in each bag is an example of the associated label.*" These examples of the associated labels are called the *primary instances*.

MI-SVR problem reduces to selecting exactly one pattern vector from each bag such that the sum of the $\varepsilon$-insensitive errors between the selected pattern vectors and the regression function is minimized. The multiple instance support vector linear regression problem can be formulated as a quadratic mixed 0–1 programming problem as follows:

$$\min \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}(\xi_i^2 + \hat{\xi}_i^2) \tag{5–1a}$$

$$\text{subject to} \quad (\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) - y_j \leq \varepsilon + \xi_i + M(1 - \eta_i) \qquad \forall i : i \in I_j \tag{5–1b}$$

$$y_j - (\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) \leq \varepsilon + \hat{\xi}_i + M(1 - \eta_i) \qquad \forall i : i \in I_j \tag{5–1c}$$

$$\sum_{i \in I_j} \eta_i = 1 \qquad \forall j \tag{5–1d}$$

$$\eta_i \in \{0,\ 1\} \qquad \forall i \tag{5–1e}$$

In the above formulation, quadratic $\varepsilon$-insensitive loss is considered. $M$ is a sufficiently large number, such that for those points with $\eta_i = 0$, the related constraint is always satisfied, and thus, does not have any influence on the problem. This is equivalent to removing this pattern vector from the problem. Constraints (5–1b-5–1c) account for the case if a pattern vector is below or above the regression function. Finally, constraint (5–1d) ensures that only one of pattern vector from each set is selected.

The linear $\varepsilon$-insensitive loss version of MI-SVR is as follows:

$$\min \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}(\xi_i + \hat{\xi}_i) \tag{5–2a}$$

$$\text{subject to} \quad (\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) - y_j \leq \varepsilon + \xi_i + M(1 - \eta_i) \qquad \forall i : i \in I_j \tag{5–2b}$$

$$y_j - (\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) \leq \varepsilon + \hat{\xi}_i + M(1 - \eta_i) \qquad \forall i : i \in I_j \tag{5–2c}$$

$$\xi_i, \hat{\xi}_i \geq 0 \qquad i = 1, \ldots, n \tag{5–2d}$$

$$\sum_{i \in I_j} \eta_i = 1 \qquad \forall j \tag{5–2e}$$

$$\eta_i \in \{0,\ 1\} \qquad \forall i \tag{5–2f}$$

This problem is known to be strongly $\mathcal{NP}$-hard for bag sizes of at least 3 (see (Ray, 2005)). While ensuring the constraints drop when $\eta = 0$, setting $M$ as small as possible is crucial to obtain good lower bounds. However, given a set of pattern vectors with class labels, $M$ cannot even be conveniently set to the maximum distance between two pairs of pattern vectors. Consider the case where $\varepsilon = 0$ which implies $M \geq \max_{i:i \in I_j} |\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b - y_j|$. Assume that $C$ is large enough that the goal is to find a regression function (if possible) with no error. Next, consider one dimensional data given as $x_1 = 0$, $x_2 = 1$, $x_3 = 2$, $x_4 = \Delta$ ($\Delta > 0$) and associated labels $y_1 = 0$, $y_2 = 2$, $y_3 = 4$, $y_4 = 4$. In other words, there are two bags with single instances (labeled 0 and 2) and one bag with two instances (labeled 4). Clearly, 0-insensitive regression will select instance 3 as the primary instance with $\boldsymbol{\psi}^* = 2$ and $b = 0$. This solution directly implies that $M > 2\Delta$ whereas the largest distance is $\max(2, \Delta)$. In our computational experiments, we empirically set $M$ sacrificing the quality of the lower bound.

In order to apply the *kernel trick* for MI-SVR, the dot products of the input patterns are needed. We rewrite formulation (5–1) as follows:

$$\begin{aligned}
&\min_{\eta} \quad \min_{\boldsymbol{\psi},b,\xi,\hat{\xi}} && \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}(\xi_i^2 + \hat{\xi}_i^2) && \text{(5–3a)}\\
&\textstyle\sum_{i\in I_j}\eta_i = 1 \\
&\eta_i \in \{0,1\} \\
&\qquad\qquad \text{subject to} && (\langle\boldsymbol{\psi},\mathbf{x}_i\rangle + b) - y_j \le \varepsilon + \xi_i + M(1-\eta_i) \quad \forall i : i \in I_j && \text{(5–3b)}\\
&\qquad\qquad && y_j - (\langle\boldsymbol{\psi},\mathbf{x}_i\rangle + b) \le \varepsilon + \hat{\xi}_i + M(1-\eta_i) \quad \forall i : i \in I_j && \text{(5–3c)}
\end{aligned}$$

In this formulation, the outer minimization sets the binary variables, and the inner minimization solves quadratic $\varepsilon$-insensitive loss version of SVR problem based on these binary values. The Lagrangian function for the inner minimization is

$$\begin{aligned}
L(\boldsymbol{\psi},b,\xi,\hat{\xi}) &= \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i}(\xi_i^2 + \hat{\xi}_i^2)\\
&- \sum_{i:i\in I_j}\alpha_i(\varepsilon + \xi_i + y_j - \langle\boldsymbol{\psi},\mathbf{x_i}\rangle - b + M(1-\eta_i))\\
&- \sum_{i:i\in I_j}\hat{\alpha}_i(\varepsilon + \hat{\xi}_i - y_j + \langle\boldsymbol{\psi},\mathbf{x_i}\rangle + b + M(1-\eta_i)).
\end{aligned} \qquad \text{(5–4)}$$

Differentiating $L$ with respect to the primal variables $\boldsymbol{\psi}$, $b$, $\xi$, and $\hat{\xi}$, and using stationarity of the inner minimization problem, we obtain

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{\psi}} &= \boldsymbol{\psi} - \sum_{i=1}^{n}(\alpha_i + \hat{\alpha}_i)\mathbf{x}_i = 0;\\
\frac{\partial L}{\partial b} &= \sum_{i=1}^{n}(\alpha_i - \hat{\alpha}_i) = 0;\\
\frac{\partial L}{\partial \xi_i} &= C\xi_i - \alpha_i = 0;\\
\frac{\partial L}{\partial \hat{\xi}_i} &= C\hat{\xi}_i - \hat{\alpha}_i = 0.
\end{aligned} \qquad \text{(5–5)}$$

When the expressions in (5–5) are substituted back in the Lagrangian function, a maximization problem inside the minimization problem is obtained. Instead, we substitute the conditions (5–5) inside (5–1) directly.

$$\min_{\eta} \quad \min_{\alpha,b} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i+\hat{\alpha}_i)(\alpha_j+\hat{\alpha}_j)\langle \mathbf{x}_i,\mathbf{x}_j\rangle + \frac{1}{2C}\sum_{i=1}^{n}(\alpha_i^2+\hat{\alpha}_i^2) \tag{5–6a}$$

$\sum_{i\in I_j}\eta_i \geq 1$

$\eta_i \in \{0,1\}$

$$\text{s.t.} \quad \sum_{j=1}^{n}(\alpha_j+\hat{\alpha}_j)\langle \mathbf{x}_j,\mathbf{x}_i\rangle + b - y_j \leq \varepsilon + \alpha_i/C + M(1-\eta_i) \quad \forall i : i \in I_j \tag{5–6b}$$

$$y_j - (\sum_{j=1}^{n}(\alpha_j+\hat{\alpha}_j)\langle \mathbf{x}_j,\mathbf{x}_i\rangle + b) \leq \varepsilon + \hat{\alpha}_i/C + M(1-\eta_i) \forall i : i \in I_j \tag{5–6c}$$

$$\sum_{i=1}^{n}(\alpha_i-\hat{\alpha}_i) = 0 \tag{5–6d}$$

The *kernel trick* is applied by replacing the dot products with kernel functions in (5–6).

$$\min_{\alpha,b} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i+\hat{\alpha}_i)(\alpha_j+\hat{\alpha}_j)K(\mathbf{x}_i,\mathbf{x}_j) + \frac{1}{2C}\sum_{i=1}^{n}(\alpha_i^2+\hat{\alpha}_i^2) \tag{5–7a}$$

$$\text{subject to} \quad \sum_{j=1}^{n}(\alpha_j+\hat{\alpha}_j)K(\mathbf{x}_i,\mathbf{x}_j) + b - y_j \leq \varepsilon + \alpha_i/C + M(1-\eta_i) \quad \forall i : i \in I_j \tag{5–7b}$$

$$y_j - (\sum_{j=1}^{n}(\alpha_j+\hat{\alpha}_j)K(\mathbf{x}_i,\mathbf{x}_j) + b) \leq \varepsilon + \hat{\alpha}_i/C + M(1-\eta_i) \; \forall i : i \in I_j \tag{5–7c}$$

$$\sum_{i=1}^{n}(\alpha_i-\hat{\alpha}_i) = 0 \tag{5–7d}$$

$$\sum_{i\in I_j}\eta_i = 1 \qquad \forall j \tag{5–7e}$$

$$\eta_i \in \{0,\ 1\} \qquad \forall i \tag{5–7f}$$

Nonlinear regression function can be obtained for multiple instance data using (5–7) for quadratic $\varepsilon$-insensitive loss. The kernel trick is applied similarly for linear $\varepsilon$-insensitive

loss MI-SVR and is skipped here. Next, we turn back to the linear regression model with MI data and present a branch and bound method for quadratic $\varepsilon$-insensitive case that outperforms a commercial solver for problems of moderate sizes.

## 5.3 Solution Approach

### 5.3.1 Lower Bounding Scheme

Let $\eta^c$ be the current values of binaries. For a partial solution where some binary variables are set (i.e., $\eta_i^c = 0$ or 1), we solve the following convex quadratic problem. This problem is a relaxation of the original problem, hence gives a lower bound. We consider the LP-relaxation to avoid the intense computational burden accompanying other relaxations where binary variables are kept.

$$
\begin{aligned}
z_{LB} = \min_{\boldsymbol{\psi},b,\xi,\eta} \quad & \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}(\xi_i^2 + \hat{\xi}_i^2) && && \text{(5–8a)}\\
\text{s.t.} \quad & (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) - y_i \leq \epsilon + \xi_i && \forall i : i \in I_j \wedge \eta_i^c = 1 && \text{(5–8b)}\\
& y_i - (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) \leq \epsilon + \hat{\xi}_i && \forall i : i \in I_j \wedge \eta_i^c = 1 && \text{(5–8c)}\\
& (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) - y_i \leq \epsilon + \xi_i + M(1-\eta_i) && \forall i : i \in I_j \wedge 0 < \eta_i^c < 1 && \text{(5–8d)}\\
& y_i - (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) \leq \epsilon + \hat{\xi}_i + M(1-\eta_i) && \forall i : i \in I_j \wedge 0 < \eta_i^c < 1 && \text{(5–8e)}\\
& \sum_{i\in I_j} \eta_i = 1 && \forall i : i \in I_j \wedge y_j = 1 \wedge 0 < \eta_i^c < 1 && \text{(5–8f)}\\
& 0 \leq \eta_i \leq 1 && \forall i : i \in I_j \wedge y_j = 1 \wedge 0 < \eta_i^c < 1 && \text{(5–8g)}
\end{aligned}
$$

If the obtained solution is integer feasible (i.e., $\eta_i^* \in \{0,1\}\ \forall i : i \in I_j$) then we can prune the node since upper and lower bounds are equal (i.e., the optimal solution for that decomposition is known). However, we observe that without a careful selection of parameter $M$, the above formulation simply drops (5–8d-5–8e) by setting $0 < \eta^* < 1$ and associated $\xi$ and $\hat{\xi}$'s are set to 0. In other words, even if an integer feasible hyperplane is obtained, $\eta^*$ values are likely to be fractional. Therefore, we check the feasibility of the hyperplane for each undecided bag explicitly. Formally, a node of the branch and

bound tree is pruned if the following boolean function is satisfied where $(\boldsymbol{\psi}^*, b^*)$ define the optimal hyperplane obtained from (5–8).

$$\bigwedge_{I_j:\eta_k^c\neq 1\ \forall k\in I_j}\ \bigvee_{i:i\in I_j}\ (|\langle\boldsymbol{\psi}^*,\mathbf{x}_i\rangle + b^* - y_i| \leq \epsilon) \tag{5–9}$$

If (5–9) is not satisfied, the feasible space need to be decomposed further. The following is the *branching scheme* we employed.

### 5.3.2 Branching Scheme

The idea in branching is to discriminate between desired and undesired solutions using the bound information and overcome the difficulty associated with an exponentially growing tree. Branching on the bags, whose instances are close to each other will only increase the size of the tree and clearly cannot be pruned easily. Moreover, if one or more instances satisfy $|\langle\boldsymbol{\psi}^*,\mathbf{x}_i\rangle + b^* - y_i| \leq \epsilon$ for a bag, it is probable that these instances will be inside the insensitive band for further decompositions. Therefore, branching on such instances or other instances that are in the same bag might not even be necessary. This expectation leads us to the idea of branching on instances that correspond to the bags, whose instances are outside the insensitive band for the current decomposition.

In our scheme, *branching* is performed on $\eta_k$ where

$$k = \arg\max_{i:i\in I_0} |\langle\boldsymbol{\psi}^*,\mathbf{x}_i\rangle + b^* - y_i| \tag{5–10}$$

and

$$I_0 = \{I_j : \bigwedge_{i:i\in I_j\wedge\eta_k^c\neq 1\ \forall k\in I_j}\ (|\langle\boldsymbol{\psi}^*,\mathbf{x}_i\rangle + b^* - y_i| > \epsilon)\} \tag{5–11}$$

The problem is decomposed into two subproblems with additional constraints $\eta_k = 1$ and $\eta_k = 0$, respectively. The aim here is to branch on the critical bag that is currently out of the insensitive band. (5–11) selects the critical bag from $I_0$ whereas (5–10) constructs $I_0$, the set of bags out of the insensitive band.

### 5.3.3 Heuristic Algorithm

To obtain an upper bound, we employ a two phase heuristic approach. In the first step, we find a separating hyperplane considering the previous decisions (i.e., $\eta^c = 1$) and all undecided bags (i.e., $0 < \eta^c < 1$). In this step, we employ a free slack approach similar to the one in (Seref et al., 2009). In the second step, we re-optimize based on a temporary selection of actual positive instances which are closest to the hyperplane. Formally, the first phase solves the following problem.

$$\min_{\boldsymbol{\psi},b,\xi} \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}(\xi_i^2 + \hat{\xi}_i^2) \tag{5–12a}$$

$$\text{subject to} \quad (\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) - y_i \leq \epsilon + \xi_i \qquad \forall i : i \in I_j \wedge \eta_i^c = 1 \tag{5–12b}$$

$$y_i - (\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) \leq \epsilon + \hat{\xi}_i \qquad \forall i : i \in I_j \wedge \eta_i^c = 1 \tag{5–12c}$$

$$(\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) - y_i \leq \epsilon + \xi_i + v_i \qquad \forall i : i \in I_j \wedge 0 < \eta_i^c < 1 \tag{5–12d}$$

$$y_i - (\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) \leq \epsilon + \hat{\xi}_i + v_i \qquad \forall i : i \in I_j \wedge 0 < \eta_i^c < 1 \tag{5–12e}$$

$$\sum_{i \in I_j} v_i \leq \delta n \qquad \forall i : i \in I_j \wedge y_j = 1 \wedge 0 < \eta_i^c < 1 \tag{5–12f}$$

$$v_i \geq 0 \qquad \forall i : i \in I_j \wedge y_j = 1 \wedge 0 < \eta_i^c < 1 \tag{5–12g}$$

The idea here is to give pooled free slack to avoid the affect of non primary instances on the regression function to a certain extent. $\delta$ is a user defined parameter. In our experiments, we tune this parameter based on the objective function values.

Formulation (5–12) gives a temporary regression hyperplane. In order to achieve the corresponding binary variables, we select the instances that are closest to the regression hyperplane for each undecided bag. The set of selected instances $S$ is defined as

$$S = \{s_j : s_j = \arg\min_{i \in I_j \wedge 0 < \eta_i^c < 1} |\langle \boldsymbol{\psi}^*, \mathbf{x}_i \rangle + b^* - y_i|, \forall j : \eta_k^c \neq 1 \, \forall k \in I_j\} \tag{5–13}$$

where $(\boldsymbol{\psi}^*, b^*)$ define the optimal solution for (5–12).

The second phase computes the upper bound by solving the margin maximization problem based on this selection.

$$
\begin{aligned}
z_{UB} = \min_{\boldsymbol{\psi},b,\xi} \quad & \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i=1}^{n}(\xi_i^2 + \hat{\xi}_i^2) && && \text{(5–14a)}\\
\text{subject to} \quad & (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) - y_i \leq \epsilon + \xi_i && \forall i : i \in I_j \wedge \eta_i^c = 1 && \text{(5–14b)}\\
& y_i - (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) \leq \epsilon + \hat{\xi}_i && \forall i : i \in I_j \wedge \eta_i^c = 1 && \text{(5–14c)}\\
& (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) - y_i \leq \epsilon + \xi_i && \forall i : i \in S && \text{(5–14d)}\\
& y_i - (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) \leq \epsilon + \hat{\xi}_i && \forall i : i \in S && \text{(5–14e)}
\end{aligned}
$$

Next, we present computational results on publicly available breast cancer data sets. The algorithm described in this section is compared with a commercial solver.

### 5.4 Computational Results on Breast Cancer Data Set

In this section, we apply the proposed branch and bound scheme for combinatorial regression problems on publicly available breast cancer data set. The breast cancer database was obtained from the University of Wisconsin Hospitals and is available in (Asuncion and Newman, 2007). Breast cancer prognosis is studied extensively in (Street et al., 1995) and (Mangasarian and Wild, 2008).

Each record in this data set represents follow-up data for one breast cancer case. These are consecutive patients seen since 1984, and include only those cases exhibiting invasive breast cancer and no evidence of distant metastases at the time of diagnosis. There are 32 features for each record. These features are the size (diameter of the excised tumor in centimeters), lymph node status (number of positive axillary lymph nodes observed at time of surgery), and 30 features that are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. These 30 features describe characteristics of the cell nuclei present in the image and include the following information for each cell nucleus: radius, texture (standard deviation of gray-scale values), perimeter, area, smoothness (local variation in radius lengths), compactness, concavity, number of concave points, symmetry, and fractal dimension.

The problem considered is to predict the time to recur for recurrent cases. There are 47 recurrent cases. In order to create multiple instance data, we add artificial Gaussian noise to the original data. To simulate cases where the features of different instances of the same bag are correlated, the label for each artificial instance is randomly drawn among the original labels. The original features of an artificial instance are used as the mean of Gaussian noise. The standard deviation is $1/\beta$ times the average value for each feature over 47 cases.

All computations are performed on a 3.4 GHz Pentium IV desktop computer with 2.0 Gb RAM and the Windows XP operating system. The algorithms are implemented in C++ and used in conjunction with MATLAB 7.3 environment in which the data resides. In our algorithm, we solved the convex minimization problems (i.e., formulations (5–8), (5–12), and (5–14)) with ILOG CPLEX 10.1. For benchmarking purposes, formulation (5–1) is solved using CPLEX 10.1 with default settings. If an algorithm terminates with optimality in 180 seconds, the lower bound is equal to the upper bound (i.e., incumbent solution). In all experiments parameter $C$ is set to 100.

| | $\beta$ | | | |
|---|---|---|---|---|
| $\delta$ | 5 | 10 | 20 | 50 |
| 1 | 21354 | 27118 | 54403 | 89755 |
| 2 | 20667 | 27118 | **40051** | 88991 |
| 5 | 24335 | 32941 | **40051** | **48418** |
| 10 | **12423** | **23626** | **40051** | 99166 |
| 20 | 24447 | 27445 | 60545 | 119150 |

Table 5-1. Effect of free slack increase for 100 artificial instances with different deviations.

Table 5-1 shows how the change in $\delta$ effects the quality of solutions. The heuristic algorithm described in Section 5.3 is used with default branching and lower bounding scheme of CPLEX. When the deviation between the instances of a bag is larger (i.e., $\beta$ is smaller), formulation (5–12) needs more slack to ignore the constraints of non-primary instances. However, since the algorithm uses the heuristic for numerous decompositions, the difference in the solution quality might be subtle for different $\delta$ values. In our

experiments, the heuristic is run once for the root node with different values of $\delta$ and this parameter is set to the value that achieves the best heuristic solution. Next, we compare our branch and bound algorithm with CPLEX solver default options.

| # art. ins. | $\beta$ | **Our B&B Algorithm** | | | **CPLEX B&B Algorithm** | | |
|---|---|---|---|---|---|---|---|
| | | **LB** | **UB** | **Time** | **LB** | **UB** | **Time** |
| 0 | N/A | 182990 | 182990 | 4.91 | 182990 | 182990 | 1.45 |
| 50 | 5 | 123.38 | 20954 | 180 | 507.22 | 22479 | 180 |
| | 10 | 66.79 | 118540 | 180 | 197.98 | 145540 | 180 |
| | 20 | 855.49 | 71615 | 180 | 2697.4 | 117970 | 180 |
| | 50 | 20501 | 89191 | 180 | 28550 | 100030 | 180 |
| 100 | 5 | 4.79 | 25080 | 180 | 5.46 | 32929 | 180 |
| | 10 | 0.17 | 57269 | 180 | 0.34 | 90237 | 180 |
| | 20 | 9.89 | 31400 | 180 | 47.53 | 104750 | 180 |
| | 50 | 7.37 | 46707 | 180 | 22.03 | 80442 | 180 |
| 150 | 5 | 38.27 | 18511 | 180 | 28.73 | 37345 | 180 |
| | 10 | 0.00 | 16750 | 180 | 0.00 | 89518 | 180 |
| | 20 | 0.00 | 39806 | 180 | 0.00 | 84676 | 180 |
| | 50 | 0.00 | 60837 | 180 | 0.00 | 80025 | 180 |
| 200 | 5 | 0.00 | 11096 | 180 | 0.00 | 25012 | 180 |
| | 10 | 0.00 | 17407 | 180 | 0.00 | 34628 | 180 |
| | 20 | 0.00 | 70435 | 180 | 0.00 | 61281 | 180 |
| | 50 | 0.00 | 59730 | 180 | 0.00 | 135130 | 180 |

Table 5-2. Computational Results for Our Branch and Bound Scheme vs. CPLEX Default Branch and Bound Algorithm for 32 features

In Table 5-2, first column shows the number of artificial instances that are added to the original data. The second column is the $\beta$ value that adjusts the deviation of the Gaussian noise for the artificial instances. We compare our branch and bound scheme with CPLEX solver default options in terms of the lower bound achieved by the time of termination and the best solution obtained. The last column shows the time spent in seconds for the algorithms to terminate either by optimality or by time limit.

Table 5-2 shows that both CPLEX and our algorithm find the optimal solution in 180 seconds in small test case where no artificial data is added. Note that instances that share the same label are assumed to be in the same bag. Therefore, the original data set is also solved in a multiple instance framework. CPLEX performs better in terms of solution time

due to the its powerful preprocessing algorithms. However, our algorithm scales up well and finds better solutions than CPLEX for larger problems. We do not report the number of remaining active nodes explicitly but it should be noted that CPLEX has significantly more number of active nodes than our algorithm on the average.

The time spent in each node of the branch and bound tree is smaller for CPLEX, hence the large number of explored and active nodes. This leads to a case where CPLEX finds better solutions than our algorithm (200 artificial instances with $\beta = 20$). For the rest of the data sets, our algorithm outperforms CPLEX in terms of the best integer solution and the optimality gap.

Next, we randomly select 10 features and test two algorithms on this data set. The idea here is to see how our algorithm performs when relatively easier decompositions with less features are solved in each node of the tree.

| # art. | | **Our B&B Algorithm** | | | **CPLEX B&B Algorithm** | | |
|---|---|---|---|---|---|---|---|
| ins. | $\beta$ | **LB** | **UB** | **Time** | **LB** | **UB** | **Time** |
| 0 | N/A | 413268 | 413268 | 2.62 | 413268 | 413268 | 1.21 |
| 50 | 5 | 121090 | 186360 | 180 | 161810 | 186370 | 180 |
| | 10 | 90080 | 225600 | 180 | 70418 | 248130 | 180 |
| | 20 | 156340 | 294470 | 180 | 158830 | 350020 | 180 |
| | 50 | 135410 | 372270 | 180 | 134490 | 377770 | 180 |
| 100 | 5 | 19048 | 124760 | 180 | 20695 | 235420 | 180 |
| | 10 | 27636 | 223070 | 180 | 33314 | 236980 | 180 |
| | 20 | 14967 | 214360 | 180 | 24662 | 266300 | 180 |
| | 50 | 88788 | 371710 | 180 | 95028 | 383210 | 180 |
| 150 | 5 | 2271 | 74579 | 180 | 1669 | 152050 | 180 |
| | 10 | 29734 | 192280 | 180 | 31376 | 228060 | 180 |
| | 20 | 1247 | 279140 | 180 | 1695 | 397850 | 180 |
| | 50 | 2554 | 350130 | 180 | 4061 | 351910 | 180 |
| 200 | 5 | 0.33 | 96907 | 180 | 1.30 | 114800 | 180 |
| | 10 | 214.99 | 104060 | 180 | 943.67 | 129350 | 180 |
| | 20 | 11.29 | 217340 | 180 | 0.82 | 351560 | 180 |
| | 50 | 36.04 | 315930 | 180 | 44.58 | 361940 | 180 |

Table 5-3. Computational Results for Our Branch and Bound Scheme vs. CPLEX Default Branch and Bound Algorithm for 10 features

The artificial instances are created using the same distribution and parameters. In this case, the time spent for each node of our branch and bound tree is close to that of CPLEX. This is due to the easy solution of convex optimization problems with less features. Table 5-3 shows that our algorithm obtained better solutions than CPLEX in all test cases. It can be observed that when the deviation between the artificial instances are larger (i.e., $\beta$ is smaller), our intuition of branching works better. When $\beta = 50$, on the other hand, the difference between the solutions obtained by our algorithm and CPLEX is subtle.

## 5.5 Conclusions and Future Work

This chapter presents the mathematical formulation, kernel trick application, and a branch and bound algorithm for linear multiple instance regression problem. Experimental results show that our algorithm outperform a commercial solver in terms of the best integer solution in a majority of the test cases.

We observe that the employed heuristic gives tight upper bounds. The lower bounding scheme should be improved through a careful selection of $M$. This number should satisfy the selection criteria but it should be small enough to have tight lower bounds with the LP-relaxations as well. A different lower bounding approach might be an interesting future study. A similar framework can also be applied for the dual formulations to obtain nonlinear MI regression.

# CHAPTER 6
# OTHER PATTERN RECOGNITION TECHNIQUES

This chapter consists of two complexity results on different pattern recognition techniques. First, we consider the complexity of feature selection for consistent biclustering in Section 6.1. Next, we prove the complexity result on hyperplanes fitting problem in Section 6.2.

## 6.1 The complexity of feature selection for consistent biclustering

Biclustering is simultaneous classification of the samples and features in a way that samples from the same class have similar values for that class' characteristic features. A biclustering is *consistent* if in each sample (feature) from any set, the average expression of features (samples) that belong to the same class is greater than the average expression of features (samples) from other classes. Supervised biclustering uses a training set to classify features whose consistency is achieved by *feature selection*. The worst case complexity of this feature selection process is studied.

### 6.1.1 Introduction

Biclustering is a methodology allowing simultaneous partitioning of a set of samples and their features into classes. Samples and features classified together are supposed to have a high relevance with each other which can be observed by intensity of their expressions. The notion of consistency for biclustering is defined using interrelation between centroids of sample and feature classes. Previous works on biclustering concentrated on unsupervised learning and did not consider employing a training set, whose classification is given. However, with the introduction of consistent biclustering, significant progress has been made in supervised learning as well.

Biclustering is applied by simultaneous classification of the samples and features (i.e., columns and rows of matrix $A$, respectively) into $k$ classes. Let $S_1, S_2, \ldots, S_k$ denote the classes of the samples (columns) and $F_1, F_2, \ldots, F_k$ denote the classes of features (rows).

One of the early algorithms to obtain an appropriate biclustering is proposed in (Hartigan, 1972), which is known as *block clustering*. Given a biclustering $\mathcal{B}$, the variability of the data in the block $(S_r, F_r)$ is used to measure the quality of the classification. A lower variability in the resulting problem is preferable. The number of classes should be fixed in order to avoid a trivial, zero variability solution in which each class consists of only one sample. A more sophisticated approach for biclustering was introduced in (Cheng and Church, 2000), where the objective is to minimize the mean squared residual. In this setting, the problem is proven to be $\mathcal{NP}$-hard and a greedy algorithm is proposed to find an approximate solution. A simulated annealing technique for this problem is discussed in (Bryan, 2005).

Another biclustering method is discussed in (Dhillon, 2001) for text mining using a bipartite graph. In the graph, the nodes represent features and samples, and each feature $i$ is connected to a sample $j$ with a link $(i, j)$, which has a weight $a_{ij}$. The total weight of all links connecting features and samples from different classes is used to measure the quality of a biclustering. A lower value corresponds to a better biclustering. A similar method for microarray data is suggested in (Kluger et al., 2003).

In (Dhillon et al., 2003), the input data is treated as a joint probability distribution between two discrete sets of random variables. The goal of the method is to find disjoint classes for both variables. A Bayesian biclustering technique based on the Gibbs sampling can be found in (Sheng et al., 2003).

The concept of *consistent biclustering* is introducted in (Busygin et al., 2005). Formally, a biclustering $\mathcal{B}$ is consistent if in each sample (feature) from any set $S_r$ (set $F_r$), the average expression of features (samples) that belong to the same class $r$ is greater than the average expression of features (samples) from other classes. The model for supervised biclustering involves solution of a special case of fractional 0-1 programming problem whose consistency is achieved by feature selection. Computational results on microarray

data mining problems are obtained by reformulating the problem as a linear mixed 0-1 programming problem.

**Theorem 5.** *Let $\mathcal{B}$ be a consistent biclustering. Then there exist convex cones $P_1, P_2, \ldots, P_k \subseteq \mathbb{R}^m$ such that only samples from $S_r$ belong to the corresponding cone $P_r$, $r = 1, \ldots, k$. Similarly, there exist convex cones $Q_1, Q_2, \ldots, Q_k \subseteq \mathbb{R}^n$ such that only features from class $F_r$ belong to the corresponding cone $Q_r$, $r = 1, \ldots, k$.*

See (Busygin et al., 2005) for the proof of Theorem 5. It also follows from the proven conic separability that convex hulls of classes do not intersect.

A problem with selecting the most representative features is the following. Assume that there is a consistent biclustering for a given data set, and there is a feature, $i$, such that the difference between the two largest values of $c^S_{ir}$ is negligible, i.e.,

$$\min_{\xi \neq \hat{r}} \{c^S_{i\hat{r}} - c^S_{i\xi}\} \leq \alpha,$$

where $\alpha$ is a small positive number. Although this particular feature is classified as a member of class $\hat{r}$ (i.e., $a_i \in F_{\hat{r}}$), the corresponding relation (1–21) can be violated by adding a slightly different sample to the data set. In other words, if $\alpha$ is a relatively small number, then it is not statistically evident that $a_i \in F_{\hat{r}}$, and feature $i$ cannot be used to classify the samples. The significance in choosing the most representative features and samples comes with the difficulty of problems that require feature tests and large amounts of samples that are expensive and time consuming. Some stronger additive and multiplicative consistent biclusterings can replace the weaker consistent biclustering. *Additive consistent biclustering* is introduced in (Nahapetyan et al., 2008) by relaxing (1–21) and (1–22) as

$$a_i \in F_{\hat{r}} \Longrightarrow c^S_{i\hat{r}} > \alpha^S_i + c^S_{i\xi}, \qquad \forall \xi, \xi \neq \hat{r}, \tag{6–1}$$

and

$$a^j \in S_{\hat{r}} \Longrightarrow c^F_{j\hat{r}} > \alpha^F_j + c^F_{j\xi}, \qquad \forall \xi, \xi \neq \hat{r}, \tag{6–2}$$

respectively, where $\alpha_j^F > 0$ and $\alpha_i^S > 0$.

Another relaxation in (Nahapetyan et al., 2008) is *multiplicative consistent biclustering* where (1–21) and (1–22) are replaced with

$$a_i \in F_{\hat{r}} \Longrightarrow c_{i\hat{r}}^S > \beta_i^S c_{i\xi}^S, \qquad \forall \xi, \xi \neq \hat{r}, \tag{6–3}$$

and

$$a^j \in S_{\hat{r}} \Longrightarrow c_{j\hat{r}}^F > \beta_j^F c_{j\xi}^F, \qquad \forall \xi, \xi \neq \hat{r}, \tag{6–4}$$

respectively, where $\beta_j^F > 1$ and $\beta_i^S > 1$.

Supervised biclustering uses accurate data sets that are called the *training set* to classify features to formulate consistent, $\alpha$-consistent and $\beta$-consistent biclustering problems. Then, the information obtained from these solutions can be used to classify additional samples that are known as the *test set*. This information is also useful for adjusting the values of vectors $\alpha$ and $\beta$ to produce more characteristic features and decrease the number of misclassifications.

Given a set of training data, construct matrix $S$ and compute the values of $c_{i\xi}^S$ using (1–19). Classify the features according to the following rule: feature $i$ belongs to class $\hat{r}$ (i.e., $a_i \in F_{\hat{r}}$), if $c_{i\hat{r}}^S > c_{i\xi}^S$, $\forall \xi \neq \hat{r}$. Finally, construct matrix $F$ using the obtained classification. Let $x_i$ denote a binary variable, which is one if feature $i$ is included in the computations and zero otherwise. Consistent, $\alpha$-consistent and $\beta$-consistent biclustering problems are formulated as follows.

CB:

$$\max_x \quad \sum_{i=1}^m x_i \tag{6–5a}$$

$$\text{subject to} \quad \frac{\sum_{i=1}^m a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^m f_{i\hat{r}} x_i} > \frac{\sum_{i=1}^m a_{ij} f_{i\xi} x_i}{\sum_{i=1}^m f_{i\xi} x_i}, \quad \forall \hat{r}, \xi \in \{1, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}} \tag{6–5b}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, m\} \tag{6–5c}$$

$\alpha$-CB:

$$\max_{x} \quad \sum_{i=1}^{m} x_i \tag{6–6a}$$

$$\text{subject to} \quad \frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^{m} f_{i\hat{r}} x_i} > \alpha_j + \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i}{\sum_{i=1}^{m} f_{i\xi} x_i}, \quad \forall \hat{r}, \xi \in \{1, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}} \tag{6–6b}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, m\} \tag{6–6c}$$

$\beta$-CB:

$$\max_{x} \quad \sum_{i=1}^{m} x_i \tag{6–7a}$$

$$\text{subject to} \quad \frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^{m} f_{i\hat{r}} x_i} > \beta_j \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i}{\sum_{i=1}^{m} f_{i\xi} x_i}, \quad \forall \hat{r}, \xi \in \{1, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}} \tag{6–7b}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, m\} \tag{6–7c}$$

The goal in the CB problem is to find the largest set of features that can be used to construct a consistent biclustering[1] . The $\alpha$-CB and $\beta$-CB problems are similar to the original CB problem but the aim is to select features that can be used to construct $\alpha$-consistent and $\beta$-consistent biclusterings, respectively.

In (6–5), $x_i$, $i = 1, \ldots m$ are the decision variables. $x_i = 1$ if $i$-th feature is selected, and $x_i = 0$ otherwise. $f_{ik} = 1$ if feature $i$ belongs to class $k$, and $f_{ik} = 0$ otherwise. The objective is to maximize the number of features selected and (6–5b) ensures that the biclustering is consistent with respect to the selected features.

### 6.1.2 Complexity Results

The optimization problem (6–5) is a specific type of *fractional 0-1 programming problem* which is defined as

[1] Note that the number of selected features is the most commonly used objective function. Other objectives such as maximizing the weighted sum of selected features can also be considered.

$$\max \quad \sum_{i=1}^{m} w_i x_i \tag{6–8a}$$

$$\text{subject to} \quad \sum_{j=1}^{n_s} \frac{\alpha^s_{j0} + \sum_{i=1}^{m} \alpha^s_{ji} x_i}{\beta^s_{j0} + \sum_{i=1}^{m} \beta^s_{ji} x_i} \geq p_s, \qquad s = 1, \ldots, S \tag{6–8b}$$

This problem is $\mathcal{NP}$-hard since linear 0-1 programming is a special class of Problem (6–8) when $\beta^s_{ji} = 0$ and $\beta^s_{j0} = 1$ for $j = 1, \ldots, n_s$, $i = 1, \ldots m$ and $s = 1 \ldots, S$. A typical way to solve a fractional 0-1 programming problem is to reformulate it as a linear mixed 0-1 programming problem, and solve new problem using standard linear programming solvers (see (T.-H.Wu, 1997; Tawarmalani et al., 2002)).

In (Busygin et al., 2005), a linearization technique for a generalized $\mathcal{NP}$-hard formulation (6–8) is applied to solve (6–5). In (Nahapetyan et al., 2008) heuristics are proposed for (6–5) and generalizations. These attempts are appropriate if the problem is $\mathcal{NP}$-hard. However, whether (6–5) itself is $\mathcal{NP}$-hard or not was an open question. This chapter intents to fill this gap by proving the $\mathcal{NP}$-hardness of (6–5).

**Theorem 6.** *Feature selection for consistent biclustering (i.e. (6–5)) is $\mathcal{NP}$-hard.*

*Proof.* To prove that the problem is $\mathcal{NP}$-hard, a special case of the problem is proven to be $\mathcal{NP}$-hard. In the case considered, there are 2 samples and $m$ features. Suppose that there are two classes and all but one of the features belong to the same class. Without loss of generality, assume that $m$-th feature belongs to one class alone and hence it is selected in the optimal solution unless the problem is infeasible (i.e., $x_m = 1$). Then (6–5b) becomes

$$\frac{\sum_{i=1}^{m-1} a_{i1} x_i}{\sum_{i=1}^{m-1} x_i} > a_{m1} \tag{6–9}$$

$$\frac{\sum_{i=1}^{m-1} a_{i2} x_i}{\sum_{i=1}^{m-1} x_i} < a_{m2} \tag{6–10}$$

It has to be proven that the decision problem is $\mathcal{NP}$-complete in order to prove that the corresponding optimization problem is $\mathcal{NP}$-hard (see (Garey and Johnson, 1979)). The decision version of feature selection for consistent biclustering problem is

D-CB: Is there a set of features that ensures biclustering is consistent, i.e., satisfies (6–9)-(6–10)?

Clearly, D-CB is in NP since the answer can be checked in $O(m)$ time for a given set of features.

Next, the KNAPSACK problem will be reduced to D-CB in polynomial time to complete the proof.

In a knapsack instance, a finite set $U_1$, a size $s(u) \in Z^+$ and a value $v(u) \in Z^+$ for each $u \in U_1$, a size constraint $B \in Z^+$, and a value goal $K \in Z^+$ are given. The question is

KNAPSACK: Is there a subset $U' \subseteq U_1$ such that $\sum_{u \in U'} s(u) \leq B$ and $\sum_{u \in U'} v(u) \geq K$.

We can modify the knapsack problem as

$\Pi$: Is there a subset $U' \subseteq U$ such that

$$\sum_{u \in U'} s(u) \leq 0 \tag{6–11}$$

$$\sum_{u \in U'} v(u) \geq 0 \tag{6–12}$$

Obviously, $\Pi$ remains $\mathcal{NP}$-complete, since KNAPSACK can be reduced to its modified variant if we define $U = U_1 \cup t$, $s(t) = -B$, and $v(t) = -K$.

Defining $s'(u) = s(u) + \alpha$, $v'(u) = v(u) + \beta$ for each $u \in U$ and it can easily be seen that

$$\sum_{u \in U'} s(u) \leq 0 \quad \Leftrightarrow \frac{\sum_{u \in U'} s'(u)}{|U'|} \leq \alpha \tag{6–13}$$

$$\sum_{u \in U'} v(u) \geq 0 \quad \Leftrightarrow \frac{\sum_{u \in U'} v'(u)}{|U'|} \geq \beta \tag{6–14}$$

In microarray data sets, negative $a_{ij}$ values usually correspond to "bad" data points. Note that, by selecting sufficiently large $\alpha$ and $\beta$ values (i.e., $\alpha > B$ and $\beta > K$), the reduction is valid for the case where $a_{ij}$ are nonnegative.

The inequality signs in (6–13)-(6–14) can be changed to strong inequality as follows

$$\frac{\sum_{u\in U'} s'(u)}{|U'|} \leq \alpha \quad \Leftrightarrow \frac{\sum_{u\in U'} s'(u)}{|U'|} < \alpha + \epsilon_1 \tag{6–15}$$

$$\frac{\sum_{u\in U'} v'(u)}{|U'|} \geq \beta \quad \Leftrightarrow \frac{\sum_{u\in U'} v'(u)}{|U'|} > \beta - \epsilon_2 \tag{6–16}$$

where $0 < \epsilon_1 < \min_{u,w\in U, s'(u)\neq s'(w)}\{|s'(u)-s'(w)|\}/|U|$ and $0 < \epsilon_2 < \min_{u,w\in U, v'(u)\neq v'(w)}\{|v'(u)-v'(w)|\}/|U|$. Note that, another upper bound on $\epsilon_2$ is $\beta$ to ensure that the resulting problem has nonnegative $a_{ij}$ values.

As a result, the problem is reduced to selecting a subset $U' \subseteq U$ such that

$$\frac{\sum_{u\in U'} s'(u)}{|U'|} < \alpha + \epsilon_1 \tag{6–17}$$

$$\frac{\sum_{u\in U'} v'(u)}{|U'|} > \beta - \epsilon_2 \tag{6–18}$$

$$\tag{6–19}$$

which is in the form of (6–9)-(6–10). The reduction is polynomial and (6–17-6–18) holds true if and only if (6–11-6–12) holds true. Thus D-CB is $\mathcal{NP}$-complete and the proof is complete. □

**Corollary 4.** *Problems (6–6) and (6–7) are $\mathcal{NP}$-hard.*

*Proof.* Problem (6–5) is a special class of Problem (6–6) when $\alpha_j = 0$ for $j \in S_{\hat{r}}$. Similarly Problem (6–5) is a special class of Problem (6–7) when $\beta_j = 1$ for all $j \in S_{\hat{r}}$. Hence both (6–6) and (6–7) are $\mathcal{NP}$-hard. □

In this section, the concept of feature selection for consistent biclustering is discussed. The aim in this setting is to select a subset of features in the original data set such that the obtained subset of data becomes conditionally biclustering-admitting with respect to the given classification of training samples. The additive and multiplicative variations of the problem are considered to extend the possibilities of choosing the most representative set of features. It is shown that the feature selection for consistent biclustering is $\mathcal{NP}$-hard. Next, we discuss the complexity result on the hyperplanes fitting problem.

## 6.2 Sparse Representation by Hyperplanes Fitting

We consider the hyperplanes fitting problem: for given $m$ points in $n$-dimensional real space, find $k$ hyperplanes that minimize the sum of squares of the distances between each point and the nearest hyperplane. Its solution allows us to obtain a *sparse representation* of a given data matrix $\mathbf{X} \in \mathbb{R}^{m\times N}$ as $\mathbf{X} = \mathbf{AS} + \mathbf{E}$, where $\mathbf{A} \in \mathbb{R}^{m\times n}$, each column of $\mathbf{S} \in \mathbb{R}^{n\times N}$ has at most $m-1$ nonzero elements, and $\mathbf{E} \in \mathbb{R}^{m\times N}$ has minimal norm. We prove that the hyperplane fitting problem is $\mathcal{NP}$-hard, and consequently, the sparse representation problem is $\mathcal{NP}$-hard too.

### 6.2.1 Introduction

Clustering is the partitioning of a data set into subsets in a way that data instances in each subset ideally share some common attribute. In other words, for instances in the same subset (cluster), closeness according to some defined distance measure is expected. Data clustering is a common technique for statistical data analysis, which is used in many fields, including data mining, pattern recognition, image analysis and bioinformatics. The computational task of classifying data set into $k$ clusters is referred to as $k$-clustering.

Classical clustering techniques in the literature (e.g., $k$-means) generate cluster centers as points that minimize the sum of squares of distances of each given instance to its nearest cluster center. Bradley and Mangasarian (2000) introduced the notion of cluster center hyperplane. The justification for this approach is that data can be grouped around

flat surfaces such as planes. Georgiev (2008) later extended this notion to cluster center subspace and nonlinear analogs of them by reproducing Kernel Hilbert Spaces.

Consider the problem of linear representation of a data set $\mathbf{X}_{m\times N}$:

$$\mathbf{X} = \mathbf{AS},\ \mathbf{A} \in \mathbb{R}^{m\times n},\ \mathbf{S} \in \mathbb{R}^{n\times N}. \tag{6–20}$$

In this decomposition, the unknown matrices $\mathbf{A}$ (dictionary) and $\mathbf{S}$ (source signals) have certain properties under different problem settings. Some of the most widely studied problems and their corresponding properties are:

(i) Independent Component Analysis (ICA): the rows of $\mathbf{S}$ are considered as discrete random variables that are statistically independent as much as possible.

(ii) Sparse Component Analysis (SCA): $\mathbf{S}$ contains as many zeros as possible.

(iii) Nonnegative Matrix Factorization (NMF): the elements of $\mathbf{X}, \mathbf{A}$ and $\mathbf{S}$ are nonnegative.

These linear representations have several applications including decomposition of objects into "natural" components and learning the elements of each object (e.g., from a set of faces, learning a face consists of eyes, nose, mouth, etc.), redundancy and dimensionality reduction, micro-array data mining, enhancement of images in nuclear medicine (see e.g., (Lee and Seung, 1999), (Chen et al., 1998)).

There are numerous studies devoted to ICA problems in the literature but these studies often consider the complete case ($m = n$) (see e.g., (Cichocki and Amari, 2002), (Hyvärinen et al., 2001)). We refer to (Bofill and Zibulevsky, 2001), (Georgiev et al., 2005), (Georgiev et al., 2004), (Zibulevsky and Pearlmutter, 2001) for SCA and overcomplete ICA ($m < n$).

Another related problem is called the *Blind Source Separation* (BSS) problem. In this setting, existence of a representation such as in equation (6–20) is known *a priori* and the task is to recover sources and the mixing matrix as accurately as possible. A fundamental property of the complete BSS problem (i.e., $m = n$) is that a decomposition is unique

up to permutation and scaling of the sources when source signals are independent and non-Gaussian.

Under the term *sparse representation* of $\mathbf{X} \in \mathbb{R}^{m \times N}$ we understand the representation

$$\mathbf{X} = \mathbf{AS} + \mathbf{E},$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{S} \in \mathbb{R}^{n \times N}$ constitute a solution for the following *least square BSS* problem:

**Definition 3.** *Find a best approximation of* $\mathbf{X}$ *by* $\mathbf{AS}$*, such that each column of* $\mathbf{S}$ *contains at most* $m-1$ *nonzero elements, i.e.,*

$$\begin{aligned} &\min && \|\mathbf{X} - \mathbf{AS}\| && \text{(6–21a)} \\ &\textit{subject to} && \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{S} \in \mathbb{R}^{n \times N} && \text{(6–21b)} \\ & && \textit{each column of } \mathbf{S} \textit{ contains at most } m-1 \textit{ nonzero elements} && \text{(6–21c)} \end{aligned}$$

*and* $\mathbf{E}$ *is an error matrix (with minimal norm).*

In the sequel, we present the problem formulation and the analogy between BSS, sparse representation and huperplane fitting. In Section 6.2.3 we show that the problem is $\mathcal{NP}$-hard and conclude this Chapter.

### 6.2.2 Problem Formulation

We consider data skeletons defined in (Rubinov and Udon, 2003). Let $X$ be a finite set of points represented by the columns of $\mathbf{X}$. We can describe this set by a collection of hyperplanes.

The solution of the following minimization problem

$$\begin{aligned} &\min && \sum_{j=1}^{N} \min_{1 \le i \le k} |\mathbf{n}_i^T \mathbf{x}_j - b_i| && && \text{(6–22a)} \\ &\text{subject to} && \|\mathbf{n}_i\| = 1 && i = 1, \ldots, k && \text{(6–22b)} \\ & && b_i \in \mathbb{R} && i = 1, \ldots, k && \text{(6–22c)} \end{aligned}$$

defines $k^{(1)}$-*skeleton* of $\mathbf{X}$ where $\mathbf{x}_j$ is the $j^{th}$ column of $\mathbf{X}$. This skeleton consists of the union of $k$ affine hyperplanes such that the sum of minimum distances of every point $\mathbf{x}_j$ to these hyperplanes is minimized.

The solution of the following minimization problem

$$\begin{aligned} \text{min} \quad & \sum_{j=1}^{N} \min_{1\le i\le k} |\mathbf{n}_i^T \mathbf{x}_j - b_i|^2 && && \text{(6–23a)} \\ \text{subject to} \quad & \|\mathbf{n}_i\| = 1 && i = 1, \ldots, k && \text{(6–23b)} \\ & b_i \in \mathbb{R} && i = 1, \ldots, k && \text{(6–23c)} \end{aligned}$$

defines $k^{(2)}$-*skeleton* of $\mathbf{X}$ (first considered in (Bradley and Mangasarian, 2000)).

Our crucial observation is that the representation

$$\mathbf{X} = \mathbf{AS}$$

is sparse (in sense that each column of $\mathbf{S}$ contains at most $m-1$ non-zero elements) if and only if the above defined two skeletons coincide, and the columns of the data matrix $\mathbf{X}$ lie on them (and necessarily $b_i = 0, i = 1, ..., k$, so the skeletons are union of hyperplanes passing trough zero).

Now, let $\mathbf{X}$ be an arbitrary data matrix, and $U$ be the union of the $k$ hyperplanes, which best fit the columns of $\mathbf{X}$. Let $\mathbf{X}_1$ be the matrix, which columns are the projections of the columns of $\mathbf{X}$ over $U$ (i.e. the closest point in $U$ to the columns of $\mathbf{X}$). Then, obviously, the skeleton of the columns of $\mathbf{X}_1$ is exactly $U$, so we have the representation $\mathbf{X}_1 = \mathbf{AS}$, for some $\mathbf{A}_1$ and sparse $\mathbf{S}_1$ (each column of $\mathbf{S}_1$ contains at most $m-1$ nonzero elements), and we have the following sparse representation of the original $\mathbf{X}$, as

$$\mathbf{X} = \mathbf{A}_1\mathbf{S}_1 + \mathbf{E}, \tag{6–24}$$

where the matrix $\mathbf{E}$ has a minimal norm. This is exactly the sparse representation which we are looking for, using hyperplanes fitting algorithms. The uniqueness of such

representation (up to the permutation and scaling of the columns of $\mathbf{A}$) is assured under the identifiability conditions considered in (Georgiev et al., 2005), (Georgiev et al., 2007). Note that such identifiability conditions are mild, so they are satisfied almost surely in practical situations.

A very suitable algorithm for clustering data near affine hyperplanes (i.e., finding $k^{(2)}$ skeleton of data points) is the $k$-plane clustering algorithm (Bradley and Mangasarian, 2000). However, this algorithm has a serious disadvantage that it stops in local minimum, even $k$ is small. We have performed extensive experiments with this algorithm and noted that if $k \geq 7$, the algorithm in almost all runs stops in local minimum. So, a global optimization algorithm is needed.

Next, we prove (6–22) and (6–23) are $\mathcal{NP}$-hard and reformulate the problem as a bilinear programming problem. We also show directions to apply some global optimization techniques to solve the problem.

### 6.2.3 Complexity Results

Consider the following minimization problem:

$$\text{min} \qquad \sum_{j=1}^{N} \min_{1 \leq i \leq k} |\tilde{\mathbf{n}}_i^T \tilde{\mathbf{x}}_j|^l \tag{6–25a}$$

$$\text{subject to} \qquad \|\tilde{\mathbf{n}}_i\| = 1, \qquad i = 1, \ldots, k \tag{6–25b}$$

where $\tilde{\mathbf{n}}_i = \begin{pmatrix} \mathbf{n}_i \\ b_i \end{pmatrix}$ and $\tilde{\mathbf{x}}_j = \begin{pmatrix} \mathbf{x}_j \\ -1 \end{pmatrix}$. Its solution $\tilde{\mathbf{n}}_i$, $i = 1, \ldots, k$ defines hyperplane skeleton in $\mathbb{R}^{m+1}$, consisting of a union of $k$ hyperplanes. Affine hyperplane $k^{(l)}$-skeleton of $\mathbf{X} \in \mathbb{R}^{m \times N}$, introduced for $l = 1$ in (Rubinov and Udon, 2003) and for $l = 2$ in (Bradley and Mangasarian, 2000), is obtained from (6–25) as $(\mathbf{n}_i/(1 - |b_i|), b_i/(1 - |b_i|))$ for $i = 1, \ldots, k$.

### 6.2.4 Reduction from SET COVER

Consider the following decision version of the hyperplane clustering problem:

**Hyperplane Clustering Decision (HCD) problem:** Are there $k$ normal hyperplanes defined in $\mathbb{R}^{m+1}$ where the sum of ($l^{th}$ power of) the distances between each point and the nearest hyperplane is less than or equal to zero?

Using a reduction from the SET COVER problem to this decision problem, we next show the following result:

**Theorem 7.** *HCD is $\mathcal{NP}$-complete for $l > 0$.*

*Proof.* It is easy to see that HCD $\in \mathcal{NP}$ since a nondeterministic algorithm needs only check in polynomial time whether each data point coincides with at least one of the hyperplanes or not.

The classical SET COVER problem is described as follows: Given a collection $C = \{c_1, c_2, \ldots, c_m\}$ of subsets of a finite set $S = \{s_1, s_2, \ldots, s_n\}$, positive integer $K \leq |C|$, does $C$ contain a cover for $S$ of size $K$ or less? This problem is known to be $\mathcal{NP}$-complete (Garey and Johnson, 1979).

Suppose that we are give an instance of SET COVER problem. We will construct the data matrix $\mathbf{X}_{m\times n}$ as follows and set $k = K$:

$$\mathbf{x}_{ij} = \begin{cases} 0, & \text{if } s_j \in c_i \\ 1, & \text{otherwise} \end{cases} \tag{6–26}$$

Selection of a collection $i$ in SET COVER implies a hyperplane $e_i$ as an answer of HCD. $e_i$ is the standard basis column vector whose elements are 0 except $i^{th}$ element which is 1. This hyperplane ensures that $\sum_{j:s_j\in c_i} |\tilde{\mathbf{n}}_i^T \tilde{\mathbf{x}}_j|^l = 0$ due to the construction described above.

When there exist normal hyperplanes that satisfy $\sum_{j=1}^{N} \min_{1\leq i\leq k} |\tilde{\mathbf{n}}_i^T \tilde{\mathbf{x}}_j|^l = 0$ but are not standard basis vectors, we can make the following transformation:

$$\mathbf{n}_i' = e_t \text{ if } \tilde{\mathbf{n}}_i(t) > 0 \tag{6–27}$$

In other words we replace all hyperplanes with fractional elements with standard basis vectors $e_t$ where $t^{th}$ element of the original hyperplane is nonzero. It is easy to see that, we still have $k$ normal hyperplanes $\mathbf{n}'_i$, $i = 1, \dots k$ and $\sum_{j=1}^{N} \min_{1 \leq i \leq k} |\mathbf{n}_i'^T \tilde{\mathbf{x}}_j|^l = 0$.

Consequently, HCD has a YES answer for data matrix $\mathbf{X}$ if and only if original SET COVER problem has a YES answer. The presented reduction is polynomial and HCD is $\mathcal{NP}$-complete. □

**Corollary 5.** *Hyperplane clustering problem, or more generally finding affine hyperplane $k^{(l)}$ skeleton, i.e., formulation (6–25) is $\mathcal{NP}$-hard.*

**Corollary 6.** *Finding the* sparse representation *of $\mathbf{X} \in \mathbb{R}^{m \times N}$ as described in Definition 3 is $\mathcal{NP}$-hard.*

Next, we show that the problem is inapproximable for any $\epsilon > 0$.

### 6.2.5 Approximation results

The proof of Theorem 7 in fact implies that it is unlikely that the solution to the hyperplane clustering problem can be approximated efficiently. An algorithm is called a $(1+\epsilon)$-approximation algorithm if, for any minimization problem instance, the algorithm finds a solution with an objective function value $\Pi_A$ that satisfies

$$\Pi_A \leq (1+\epsilon)\Pi^*,$$

where $\Pi^* \geq 0$ is the optimal objective function value and $\epsilon > 0$.

**Theorem 8.** *There exists no polynomial-time $(1+\epsilon)$-approximation algorithm for the hyperplane clustering problem for any $\epsilon > 0$ unless $\mathcal{P} = \mathcal{NP}$.*

*Proof.* Consider a problem instance for the hyperplane clustering problem reduced from a SET COVER instance as described in the proof of Theorem 7. Recall that any solution for the hyperplane clustering problem instance with objective function value $\Pi = 0$ corresponds to a feasible covering with YES answer for the SET COVER instance and that a solution with $\Pi > 0$ corresponds to a NO answer. Assume there exists a polynomial-time $(1+\epsilon)$-approximation algorithm $A$ for some $\epsilon > 0$. If $\Pi^* > 0$, then the

SET COVER instance has NO answer and the solution found by algorithm $A$ satisfies $\Pi_A \leq (1+\epsilon)\Pi^*$. On the other hand, if $\Pi^* = 0$, then the SET COVER instance has a YES answer and the solution found by algorithm $A$ satisfies $\Pi_A \leq (1+\epsilon)\Pi^* = 0$. But this means that algorithm $A$ determines in polynomial time whether the SET COVER instance has a YES answer, which is a contradiction unless $\mathcal{P} = \mathcal{NP}$. □

In this study, we explore hyperplanes fitting problem and prove that this problem is $\mathcal{NP}$-hard. As a consequence, a corresponding sparse representation problem is $\mathcal{NP}$-hard too. Such sparse representation problem can be considered as a generalization of the Blind Signal Separation problem based on sparsity assumptions of the source matrix. We also proposed a new global optimization algorithm for finding the best hyperplane skeleton, based on a bilinear reformulation and cutting plane method. It is a base for a new algorithm for sparse representation and Blind Signal Separation problems for demixing unknown mixture of source signals under mild sparsity assumptions.

# CHAPTER 7
# CONCLUDING REMARKS AND FUTURE WORK

Our discussion on mathematical programming problems in pattern recognition starts with an introductory survey on general optimization based machine learning techniques with applications in healthcare. This chapter is based on (Kundakcioglu and Pardalos, 2009b) and (Seref et al., 2008a). Next, we consider linear classification problems in *death cell discrimination*. This aim of this study is to develop diagnostic tools for cancer and quantify the cellular response to chemotherapy and the toxicity assessment of different drugs. This study is based on (Pyrgiotakis et al., 2009).

Based on (Seref et al., 2009), we introduce a novel selective classification method which is a generalization of the standard SVM classifiers. Sets of pattern vectors sharing the same label are given as input. One pattern vector is selected from each set in order to maximize the classification margin with respect to the selected positive and negative pattern vectors. The problem of selecting the best pattern vectors is referred to as the *hard selection* problem. The hard selection problem is shown to be $\mathcal{NP}$-hard. We propose alternative linear and nonlinear approaches with tractable formulations, which we call *soft selection* problems. The selective nature of the these formulations is maintained by the restricted free slack concept. The intuition behind this concept is to reverse the combinatorial selection problem by detecting influential pattern vectors which require free slack to decrease their effect on the classification functions. Iteratively removing such pattern vectors, we can find those pattern vectors with a larger margin. An *iterative elimination* method is proposed for this purpose. Another alternative approach is to provide enough free slack to identify all $t-1$ out of $t$ pattern vectors to be removed at once, which leads to the *direct selection* method. The iterative elimination and the direct selection methods are found to produce similar results. Iterative elimination method is also compared with a naïve elimination method which uses standard SVM to eliminate

pattern vectors. The results show that iterative elimination is superior to the naïve elimination method both in linear and nonlinear classification.

Chapter 4 presents the mathematical formulation, kernel trick application, complexity results, and an exact algorithm for linear multiple instance classification through margin maximization. Experimental results show additional benefits of intelligent bounding and branching schemes. We observe that the employed heuristic gives tight upper bounds but the lower bounding scheme needs to be improved. The lower bounding technique we propose helps mostly with pruning by optimality but rarely with pruning by bound. This chapter is based on (Kundakcioglu et al., 2009b). Chapter 5 extends the exact algorithm for regression and is based on (Kundakcioglu et al., 2009a).

Next is a brief complexity result on feature selection for consistent biclustering. The aim in this setting is to select a subset of features in the original data set such that the obtained subset of data becomes conditionally biclustering-admitting with respect to the given classification of training samples. The additive and multiplicative variations of the problem are considered to extend the possibilities of choosing the most representative set of features. It is shown that the feature selection for consistent biclustering is $\mathcal{NP}$-hard. This study is published in (Kundakcioglu and Pardalos, 2009a). In the same chapter, we consider the hyperplanes fitting problem, where the goal is to find hyperplanes that minimize the sum of squares of the distances between each data point and the nearest hyperplane. We prove that this problem is $\mathcal{NP}$-hard.

Here, we present alternative formulations for support vector classifiers with multiple instance data. These nonconvex formulations do not utilize integer variables. A comparison of different formulations with different commercial solvers would also be an interesting future study.

$$\min_{\boldsymbol{\psi},b,\xi,\eta} \qquad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i\in I^-}\xi_i^2 + \frac{C}{2}\sum_{i\in I^+}\eta_i\xi_i^2 \tag{7–1a}$$

$$\text{subject to} \quad y_i\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i \qquad i = 1, \ldots, n \tag{7–1b}$$

$$\sum_{i \in I_j} \eta_i = 1 \qquad j \in J^+ \tag{7–1c}$$

$$\eta_i \geq 0 \qquad i \in I^+ \tag{7–1d}$$

$$\min_{\boldsymbol{\psi}, b, \xi, \eta} \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + \frac{C}{2}\sum_{i \in I^-} \xi_i^2 + \frac{C}{2}\sum_{j \in J^+} \hat{\xi}_j^2 \tag{7–2a}$$

$$\text{subject to} \quad \sum_{i \in I_j} (\eta_i \langle \boldsymbol{\psi}, \mathbf{x}_i\rangle) + b \geq 1 - \hat{\xi}_j \qquad j \in J^+ \tag{7–2b}$$

$$-\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle - b \geq 1 - \xi_i \qquad i \in I^- \tag{7–2c}$$

$$\sum_{i \in I_j} \eta_i = 1 \qquad j \in J^+ \tag{7–2d}$$

$$\eta_i \geq 0 \qquad i \in I^+ \tag{7–2e}$$

Formulation (7–1) and (7–2) consider the convex combination of all points in a bag and try to penalize the misclassification for this point. The objective of minimizing total misclassification ensures that this convex combination is the actual positive for the bag.

Our future work is on new exact methods for combinatorial classification and regression problems. Nonlinear extensions are only considered for Selective SVMs but exact methods can also be explored for nonlinear classification with multiple instance data. As far as the problems in Chapter 6 are concerned, exact and heuristic methods for hyperplanes fitting problem would be interesting future studies.

Acir, N., C. Güzelis. 2005. Automatic recognition of sleep spindles in EEG via radial basis support vector machine based on a modified feature selection algorithm. *Neural Computing and Applications* **14**(1) 56–65.

Acosta, R., M. Ehrgott, A. Holder, D. Nevin, J. Reese, B. Salter. 2008. *Optimization in Medicine*, chap. The influence of dose grid resolution on beam selection strategies in radiotherapy treatment design. Springer, 1–23.

Alexe, S., E. Blackstone, P. Hammer, H. Ishwaran, M. Lauer, C. Snader. 2003. Coronary risk prediction by logical analysis of data. *Annals of Operations Research* **119** 15–42.

Andrews, S., T. Hofmann, I. Tsochantaridis. 2002. Multiple instance learning with generalized support vector machines. *Eighteenth National Conference on Artificial Intelligence*. American Association for Artificial Intelligence, Menlo Park, CA, USA, 943–944.

Andrews, S., I. Tsochantaridis, T. Hofmann. 2003. *Advances in Neural Information Processing Systems*, vol. 15, chap. Support vector machines for multiple-instance learning. MIT Press, Vancouver, British Columbia, Canada, 561–568.

Armour, E.P., D. McEachern, Z. Wang, P.M. Corry, A. Martinez. 1993. Sensitivity of human cells to mild hyperthermia. *Cancer Research* **53**(12) 2740–2744.

Asuncion, A., D.J. Newman. 2007. UCI machine learning repository. URL http://mlearn.ics.uci.edu/. Last accessed June 24, 2009.

Auer, P. 1997. On learning from multi-instance examples: Empirical evaluation of a theoretical approach. *Proceedings 14th International Conference on Machine Learning*. 21–29.

Balasundaram, B., S. Butenko, S. Trukhanov. 2005. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization* **10**(1) 23–39.

Ben-Dor, A., L. Bruhn, N. Friedman, I. Nachman, M. Schummer, Z. Yakhini. 2000. Tissue classification with gene expression profiles. *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology* **7** 559–583.

Ben-Dor, A., B. Chor, R. Karp, Z. Yakhini. 2002. Discovering local structure in gene expression data: The order-preserving submatrix problem. *RECOMB '02: Proceedings of the Sixth Annual International Conference on Computational Biology*. 49–57.

Ben-Dor, A., N. Friedman, Z. Yakhini. 2001. Class discovery in gene expression data. *RECOMB '01: Proceedings of the Fifth Annual International Conference on Computational Biology*. ACM Press, New York, NY, USA, 31–38.

Bennet, K., C. Campbell. 2000. Support vector machines: Hype or hallelujah? *SIGKDD Explorations* **2**(2) 1–13.

Bergeron, C., F. Cheriet, J. Ronsky, R. Zernicke, H. Labelle. 2005. Prediction of anterior scoliotic spinal curve from trunk surface using support vector regression. *Engineering Applications of Artificial Intelligence* **18**(8) 973–983.

Bertsimas, D., R. Shioda. 2007. Classification and regression via integer optimization. *Operations Research* **55**(2) 252–271.

Bevilacqua, V., G. Mastronardi, G. Piscopo. 2007. Evolutionary approach to inverse planning in coplanar radiotherapy. *Image and Vision Computing* **25**(2) 196–203.

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

Billups, S., J. Kennedy. 2001. Minimum-support solutions for radiotherapy planning. *Annals of Operations Research* **119** 229–245.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Blum, A., A. Kalai. 1998. A note on learning from multiple-instance examples. *Machine Learning* **30**(1) 23–29.

Boesewetter, D., J. Collier, A. Kim, M. Riley. 2006. Alterations of A549 lung cell gene expression in response to biochemical toxins. *Cell Biology and Toxicology* **22**(2) 101–108.

Bofill, P., M. Zibulevsky. 2001. Underdetermined blind source separation using sparse representation. *Signal Processing* **81**(11) 2353–2362.

Bradley, P. S., O. L. Mangasarian. 2000. $k$-plane clustering. *Journal of Global Optimization* **16**(1) 23–32.

Brandeau, M. L., F. Sainfort, W. P. Pierskalla, eds. 2004. *Handbook of Operations Research and Health Care: Methods and Applications*. Kluwer Academic Publishers.

Brouwer, G. J., R. van Ee. 2007. Visual cortex allows prediction of perceptual states during ambiguous structure-from-motion. *The Journal of Neuroscience* **27**(5) 1015–1023.

Brow, T., B. Settles, M. Craven. 2005. Classifying biomedical articles by making localized decisions. *Proceedings of the Fourteenth Text Retrieval Conference (TREC)*.

Brown, M., W. Grundy, D. Lin, N. Cristianini, C. Sugne, T. Furey, M. Ares, D. Haussler. 2000. Knowledge-base analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences* **97**(1) 262–267.

Bryan, K. 2005. Biclustering of expression data using simulated annealing. *CBMS '05: Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*. IEEE Computer Society, Washington, DC, USA, 383–388.

Busygin, S., N. Boyko, P. M. Pardalos, M. Bewernitz, G. Ghacibeh. 2007a. Biclustering EEG data from epileptic patients treated with vagus nerve stimulation. O. Seref, O. E. Kundakcioglu, P. M. Pardalos, eds., *Data Mining, Systems Analysis, and Optimization in Biomedicine*. American Institute of Physics, 220–231.

Busygin, S., G. Jacobsen, E. Kramer. 2002. Double conjugated clustering applied to leukemia microarray data. *Proceedings of the Second SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data*.

Busygin, S., O. A. Prokopyev, P. M. Pardalos. 2005. Feature selection for consistent biclustering via fractional 0–1 programming. *Journal of Combinatorial Optimization* **10**(1) 7–21.

Busygin, S., O. A. Prokopyev, P. M. Pardalos. 2007b. An optimization-based approach for data classification. *Optimization Methods & Software* **22**(1) 3–9.

Busygin, S., O. A. Prokopyev, P. M. Pardalos. 2008. Biclustering in data mining. *Computers & Operations Research* **35**(8) 2964–2987.

Carew, J. D., M. Yuan. 2007. Nonparametric smoothing and its applications in biomedical imaging. O. Seref, O. E. Kundakcioglu, P. M. Pardalos, eds., *Data Mining, Systems Analysis, and Optimization in Biomedicine*. American Institute of Physics, 85–105.

Carneiro, G., A. B. Chan, P. J. Moreno, N. Vasconcelos. 2007. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(3) 394–410.

Censor, Y., T. Bortfeld, B. Martin, A. Trofimov. 2006. A unified approach for inversion problems in intensity-modulated radiation therapy. *Physics in Medicine and Biology* **51**(10) 2353–2365.

Chaovalitwongse, W., P. M. Pardalos, L. D. Iasemidis, W. Suharitdamrong, D.-S. Shiau, L. K. Dance, O. A. Prokopyev, V. L. Boginski, P. R. Carney, J. C. Sackellares. 2007. *Data Mining in Biomedicine*, *Optimization and Its Applications*, vol. 7, chap. Data mining in EEG: Application to epileptic brain disorders. Springer, 459–481.

Chaovalitwongse, W., O. Prokopyev, P. M. Pardalos. 2006. Electroencephalogram (EEG) time series classification: Applications in epilepsy. *Annals of Operations Research* **148**(1) 227–250.

Chen, S., D. Donoho, M. Saunders. 1998. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing* **20**(1) 33–61.

Chen, Y., J. Bi, J. Z. Wang. 2006. MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(12) 1931–1947.

Chen, Y., J. Z. Wang. 2004. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research* **5** 913–939.

Cheng, Y., G. M. Church. 2000. Biclustering of expression data. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 93–103.

Choy, K. Y., C. W. Chan. 2003. Modelling of river discharges and rainfall using radial basis function networks based on support vector regression. *International Journal of Systems Science* **34**(14–15) 763–773.

Chuang, S. C., Y. Y. Xu, H.-C. Fu. 2005. Neural network based image retrieval with multiple instance leaning techniques. *Lecture Notes in Computer Science* **3682** 1210–1216.

Cichocki, A., S. Amari. 2002. *Adaptive Blind Signal and Image Processing*. John Wiley.

Cifarelli, C., G. Patrizi. 2007. Solving large protein folding problem by a linear complementarity algorithm with 0–1 variables. *Optimization Methods and Software* **22**(1) 25–49.

Cox, D. D., R. L. Savoy. 2003. Functional magnetic resonance imaging (fMRI) "brain reading": Detecting and classifying distributed patterns of fMRI activity in human visual cortex. *NeuroImage* **19** 261–270.

Craft, D. L., T. F. Halabi, H. A. Shih, T. R. Bortfeld. 2006. Approximating convex pareto surfaces in multiobjective radiotherapy planning. *Medical Physics* **33**(9) 3399–3407.

Cristianini, N., J. Shawe-Taylor. 2000. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK.

Darbellay, G. A., R. Duff, J. M. Vesin, P. A. Despland, D. W. Droste, C. Molina, J. Serena, R. Sztajzel, P. Ruchat, T. Karapanayiotides, A. Kalangos, J. Bogousslavsky, E. B. Ringelstein, G. Devuyst. 2004. Solid or gaseous circulating brain emboli: Are they separable by transcranial ultrasound? *Journal of Cerebral Blood Flow & Metabolism* **24** 860–868.

Dariush, B. 2003. Human motion analysis for biomechanics and biomedicine. *Machine Vision and Applications* **14**(4) 202–205.

Devos, A., L. Lukas, J.A.K. Suykens, L. Vanhamme, A.R. Tate, F.A. Howe, C. Majos, A. Moreno-Torres, M. van der Graaf, C. Arus, S. Van Huffel. 2004. Classification of brain tumours using short echo time 1H MR spectra. *Journal of Magnetic Resonance* **170** 164–175.

Devos, A., A.W. Simonetti, M. van der Graaf, L. Lukas, J.A.K. Suykens, L. Vanhamme, L.M.C. Buydens, A. Heerschap, S. Van Huffel. 2005. The use of multivariate mr imaging intensities versus metabolic data from MR spectroscopic imaging for brain tumour classification. *Journal of Magnetic Resonance* **173** 218–228.

Dewhirst, M. W., D. A. Sim, S. Sapareto, W. G. Connor. 1984. Importance of minimum tumor temperature in determining early and long-term responses of spontaneous canine and feline tumors to heat and radiation. *Cancer Research* **44**(1) 43–50.

Dhillon, I. S. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 269–274.

Dhillon, I. S., S. Mallela, D. S. Modha. 2003. Information-theoretic co-clustering. *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 89–98.

Dietterich, T. G., R. H. Lathrop, T. Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence* **89** 31–71.

Divina, F., J. S. Aguilar-Ruiz. 2006. Biclustering of expression data with evolutionary computation. *IEEE Transactions on Knowledge and Data Engineering* **18**(5) 590–602.

Donahue, M. M., W. Zhang, M. L. Harrison, J. Hu, A. E. Rundell. 2007. Employing optimization and sensitivity analyses tools to generate and analyze mathematical models of T-cell signaling events. O. Seref, O. E. Kundakcioglu, P. M. Pardalos, eds., *Data Mining, Systems Analysis, and Optimization in Biomedicine*. American Institute of Physics, 43–63.

Dooly, D. R., Q. Zhang, S. A. Goldman, R. A. Amar. 2002. Multiple-instance learning of real-valued data. *Journal of Machine Learning Research* **3** 651–678.

Dube, S., J. J. Corso, T. F. Cloughesy, S. El-Saden, A. L. Yuille, U. Sinha. 2007. Automated MR image processing and analysis of malignant brain tumors: Enabling technology for data mining. O. Seref, O. E. Kundakcioglu, P. M. Pardalos, eds., *Data Mining, Systems Analysis, and Optimization in Biomedicine*. American Institute of Physics, 64–84.

Eacott, M.J., D. Gaffan. 1991. The role of monkey inferior parietal cortex in visual discrimination of identity and orientation of shapes. *Behavioural Brain Research* **46**(1) 95–98.

Ehrgott, M., H. W. Hamacher, M. Nußbaum. 2008. *Optimization in Medicine*, chap. Decomposition of matrices and static multileaf collimators: A survey. Springer, 25–46.

Fan, Y., D. Shen, R. C. Gur, R. E. Gur, C. Davatzikos. 2007. Compare: Classification of morphological patterns using adaptive regional elements. *IEEE Transactions on Medical Imaging* **26**(1) 93–105.

Faugeras, O., G. Adde, G. Charpiat, C. Chefd'Hotel, M. Clerc, T. Deneux, R. Deriche, G. Hermosillo, R. Keriven, P. Kornprobst, J. Kybic, C. Lenglet, L. Lopez-Perez, T. Papadopoulo, J.-P. Pons, F. Segonne, B. Thirion, D. Tschumperle, T. Vieville, N. Wotawa. 2004. Variational, geometric, and statistical methods for modeling brain anatomy and function. *NeuroImage* **23** 46–55.

Ferris, M., J. Lim, D. Shepard. 2001. Radiosurgery treatment planning via nonlinear programming. *Annals of Operations Research* **119** 247–260.

Fung, G., M. Dundar, B. Krishnapuram, R. B. Rao. 2007. Multiple instance learning for computer aided diagnosis. B. Schölkopf, J. Platt, T. Hoffman, eds., *Advances in Neural Information Processing Systems*, vol. 19. MIT Press, Vancouver, British Columbia, Canada, 425–432.

Fung, G., J. Stoeckel. 2007. SVM feature selection for classification of SPECT images of Alzheimer's disease using spatial information. *Knowledge and Information Systems* **11**(2) 243–258.

Fung, H. K., S. Rao, C. A. Floudas, O. Prokopyev, P. M. Pardalos, F. Rendl. 2005. Computational comparison studies of quadratic assignment like formulations for the in silico sequence selection problem in de novo protein design. *Journal of Combinatorial Optimization* **10**(1) 41–60.

Garcia, G.N., T. Ebrahimi, J.M. Vesin. 2003. Joint time-frequency-space classification of EEG in a brain-computer interface application. *Journal on Applied Signal Processing* 713–729.

Garey, M.R., D.S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co.

Garrett, D., D. A. Peterson, C. W. Anderson, M. H. Thaut. 2003. Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **11**(2) 141–144.

Gartner, T., P. A. Flach, A. Kowalczyk, A. J. Smola. 2002. Multi instance kernels. *Proceedings of the 19th International Conference on Machine Learning*. 179–186.

Genkin, A., C. A. Kulikowski, I. Muchnik. 2002. Set covering submodular maximization: An optimal algorithm for data mining in bioinformatics and medical informatics. *Journal of Intelligent & Fuzzy Systems* **12** 5–17.

Georgiev, P., F. Theis, A. Cichocki. 2004. Blind source separation and sparse component analysis of overcomplete mixtures. *Proceedings of ICASSP 2004*. Montreal, Canada.

Georgiev, P., F. Theis, A. Cichocki. 2005. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Transactions of Neural Networks* **16**(4) 992–996.

Georgiev, P., F. Theis, A. Ralescu. 2007. Identifiability conditions and subspace clustering in sparse BSS. *Lecture Notes Computer Science* **4666** 357–364.

Georgiev, P. G. 2008. Nonlinear skeletons of data sets and applications – methods based on subspace clustering. P. M. Pardalos, P. Hansen, eds., *Data Mining and Mathematical Programming*, *CRM Proceedings and Lecture Notes*, vol. 45. American Mathematical Society, 95–108.

Gerner, E. W., W. G. Connor, M. L. Boone, J. D. Doss, E. G. Mayer, R. C. Miller. 1975. The potential of localized heating as a adjunct to radiation therapy. *Radiology* **116**(02) 433–439.

Geva, A. B., D. H. Kerem. 1998. Forecasting generalized epileptic seizures from the EEG signal by wavelet analysis and dynamic unsupervised fuzzy clustering. *IEEE Transactions on Biomedical Engineering* **45**(10) 1205–1216.

Giard, D. J., S. A. Aaronson, G. J. Todaro, P. Arnstein, J. H. Kersey, H. Dosik, W. P. Parks. 1973. In vitro cultivation of human tumors: Establishment of cell lines derived from a series of solid tumors. *Journal of the National Cancer Institute* **51**(5) 1417.

Glotsos, D., P. Spyridonos, D. Cavouras, P. Ravazoula, P. Arapantoni Dadioti, G. Nikiforidis. 2005a. An image-analysis system based on support vector machines for automatic grade diagnosis of brain-tumour astrocytomas in clinical routine. *Medical Informatics and the Internet in Medicine* **30**(3) 179–193.

Glotsos, D., J. Tohka, P. Ravazoula, D. Cavouras, G. Nikiforidis. 2005b. Automated diagnosis of brain tumours astrocytomas using probabilistic neural network clustering and support vector machines. *International Journal of Neural Systems* **15** 1–11.

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

Greenberg, H., W. Hart, G. Lancia. 2004. Opportunities for combinatorial optimization in computational biology. *INFORMS Journal on Computing* **16** 211–231.

Guan, J., Y. Chen, J. Lin. 2005. Single-trial estimation of imitating-natural-reading evoked potentials in single-channel. *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. 2052–2055.

Guigue, V., A. Rakotomamonjy, S. Canu. 2006. Translation-invariant classification of non-stationary signals. *Neurocomputing* **69** 743–753.

Hall, S. N., S. H. Jacobson, E. C. Sewell. 2008. *Optimization in Medicine and Biology*, chap. Optimizing Pediatric Vaccine Formularies. Auerbach, 117–146.

Hartigan, J. A. 1972. Direct clustering of a data matrix. *Journal of the American Statistical Association* **67**(337) 123–129.

Hayashi, S., M. Hatashita, H. Matsumoto, Z.H. Jin, H. Shioura, E. Kano. 2005. Modification of thermosensitivity by amrubicin or amrubicinol in human lung adenocarcinoma A549 cells and the kinetics of apoptosis and necrosis induction. *International Journal of Molecular Medicine* **16**(3) 381–387.

Hildebrandt, B., P. Wust, O. Ahlers, A. Dieing, G. Sreenivasa, T. Kerner, R. Felix, H. Riess. 2002. The cellular and molecular basis of hyperthermia. *Critical Reviews in Oncology/Hematology* **43**(1) 33–56.

Hong, W. C., P. F. Pai. 2007. Potential assessment of the support vector regression technique in rainfall forecasting. *Water Resources Management* **21**(2) 495–513.

Horel, J.A., L.J. Misantone. 1976. Visual discrimination impaired by cutting temporal lobe connections. *Science* **193**(4250) 336–338.

Hsu, C.W., C.C. Chang, C.J. Lin. 2004. A practical guide to support vector classification. http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf.

Hu, J., J. Si, B. P. Olson, J. He. 2005. Feature detection in motor cortical spikes by principal component analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **13**(3) 256–262.

Huang, P., W. Plunkett. 1992. A quantitative assay for fragmented DNA in apoptotic cells. *Analytical Biochemistry* **207**(1) 163–167.

Huang, Z., H. Chen, C. J. Hsu, W. H. Chenb, S. Wuc. 2004. Credit rating analysis with support vector machines and neural networks: A market comparative study. *Decision Support Systems* **37** 543–558.

Huttunen, T., J. P. Kaipio, M. Malinen. 2008. *Optimization in Medicine*, chap. Optimal control in high intensity focused ultrasound surgery. Springer, 169–195.

Hyunsoo, K., Z. X. Jeff, M. C. Herbert, P. Haesun. 2005. A three-stage framework for gene expression data analysis by L1-norm support vector regression. *International Journal of Bioinformatics Research and Applications* **1**(1) 51–62.

Hyvärinen, A., J. Karhunen, E. Oja. 2001. *Independent Component Analysis*. John Wiley & Sons.

Iasemidis, L. 1991. On the dynamics of the human brain in temporal lobe epilepsy. Ph.D. thesis, University of Michigan, Ann Arbor.

ILOG. 2008. *CPLEX 10.1*. Incline Village, Nevada. URL http://www.ilog.com/products/cplex/.

Jaeschke, H., J. S. Gujral, M. L. Bajt. 2004. Apoptosis and necrosis in liver disease. *Liver International : Official journal of the International Association for the Study of the Liver* **24**(2) 85–89.

Jain, A. N., T. G. Dietterich, R. H. Lathrop, D. Chapman, R. E. Critchlow, B. E. Bauer, T. A. Webster, T. Lozano-Perez. 1994. A shape-based machine learning tool for drug design. *Journal of Computer-Aided Molecular Design* **8**(6) 635–652.

Jakuczun, W., E. Kublik, D. K. Wójcik, A. Wróbel. 2005. Local classifiers for evoked potentials recorded from behaving rats. *Acta neurobiologiae experimentalis* **65** 425–434.

Joachims, T. 1998. Text categorization with support vector machines: Learning with many relevant features. Claire Nédellec, Céline Rouveirol, eds., *Proceedings of the European Conference on Machine Learning*. Springer, Berlin, 137–142.

Joachims, T. 1999. Making large–scale SVM learning practical. B. Schölkopf, C. J. C. Burges, A. J. Smola, eds., *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, 169–184.

Jones, D. S. 2002. *Pharmaceutical Statistics*. Pharmaceutical Press.

Kalatzis, I., D. Pappas, N. Piliouras, D. Cavouras. 2003. Support vector machines based analysis of brain SPECT images for determining cerebral abnormalities in asymptomatic diabetic patients. *Medical Informatics and the Internet in Medicine* **28**(3) 221–230.

Kanduc, D., P. Bannasch, E. Farber. 1999. A critical perspective in cancer research (review). *International Journal of Oncology* **15**(6) 1213–1220.

Kanduc, D., F. Capuano, S. A. Capurso, J. Geliebter, D. Guercia, A. Lucchese, A. Mittelman, S. M. Simone, A. A. Sinha, R. Tiwari, E. Farber. 2003. Cancer prevention and therapy: strategies and problems. *Journal of Experimental Therapeutics & Oncology* **3**(3) 108–114.

Kanduc, D., J. Geliebter, A. Lucchese, R. Mazzanti, A. Mittelman, L. Polimeno, A. Ponzetto, R. Santacroce, S. Simone, E. Sinigaglia, A. A. Sinha, L. Tessitore, R. K. Tiwari, E. Farber. 2005. Gene therapy in cancer: The missing point. *Journal of Experimental Therapeutics & Oncology* **5**(2) 151–158.

Kanduc, D., A. Mittelman, R. Serpico, E. Sinigaglia, A. A. Sinha, C. Natale, R. Santacroce, M. G. Di Corcia, A. Lucchese, L. Dini, P. Pani, S. Santacroce, S. Simone, R. Bucci, E. Farber. 2002. Cell death: Apoptosis versus necrosis (review). *International Journal of Oncology* **21**(1) 165–170.

Kaper, M., P. Meinicke, U. Grossekathoefer, T. Lingner, H. Ritter. 2004. BCI competition 2003–data set iib: Support vector machines for the P300 speller paradigm. *IEEE Transactions on Biomedical Engineering* **51**(6) 1073–1076.

Kaper, M., H. Ritter. 2004. Generalizing to new subjects in brain-computer interfacing. *Proceedings of the 26th IEEE EMBS Annual International Conference (EMBC)*. 4363–4366.

Karpinich, N. O., M. Tafani, R. J. Rothman, M. A. Russo, J. L. Farber. 2002. The course of etoposide-induced apoptosis from damage to DNA and p53 activation to mitochondrial release of cytochrome *c*. *Journal of Biological Chemistry* **277**(19).

Karpouzas, I., Y. Pouliquen. 1991. Modelling and numerical optimization of corneal rotation. *Mathematical Medicine and Biology* **8**(1) 73–82.

Keirn, Z. A., J. I. Aunon. 1990. A new mode of communication between man and his surroundings. *IEEE Transactions on Biomedical Engineering* **37** 1209–1214.

Kelm, B. M., B. H. Menze, C. M. Zechmann, K. T. Baudendistel, F. A. Hamprecht. 2007. Automated estimation of tumor probability in prostate magnetic resonance spectroscopic imaging: Pattern recognition vs. quantification. *Magnetic Resonance in Medicine* **57** 150–159.

Kluger, Y., R. Basri, J. T. Chang, M. Gerstein. 2003. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Research* **13**(4) 703–716.

Kochenberger, G., F. Glover, B. Alidaee, H. Wang. 2005. Clustering of microarray data via clique partitioning. *Journal of Combinatorial Optimization* **10**(1) 7–21.

Kotropoulos, C., I. Pitas. 2003. Segmentation of ultrasonic images using support vector machines. *Pattern Recognition Letters* **24** 715–727.

Kundakcioglu, O. E., S. M. Nasseri, P. M. Pardalos. 2009a. Support vector regression with multiple instance data *submitted*.

Kundakcioglu, O. E., P. M. Pardalos. 2008. A branch and bound algorithm for multiple instance classification. H. R. Arabnia, Y. Mun, eds., *Proceedings of the 2008 International Conference on Machine Learning; Models, Technologies and Applications (MLMTA)*, vol. 2. 865–869.

Kundakcioglu, O. E., P. M. Pardalos. 2009a. *Clustering Challenges in Biological Networks*, chap. The complexity of feature selection for consistent biclustering. World Scientific, 257–266.

Kundakcioglu, O. E., P. M. Pardalos. 2009b. *Lectures on Global Optimization*, *Fields Institute Communications Series*, vol. 55, chap. Optimization in biomedical research. American Mathematical Society, 155–182.

Kundakcioglu, O. E., O. Seref, P. M. Pardalos. 2009b. Multiple instance learning via margin maximization. *Applied Numerical Mathematics* doi:10.1016/j.apnum.2009.05.013.

LaConte, S., S. Strother, V. Cherkassky, J. Anderson, X. Hu. 2005. Support vector machines for temporal classification of block design fMRI data. *NeuroImage* **26** 317–329.

Lahanas, M., D. Baltas, N. Zamboglou. 2003a. A hybrid evolutionary multiobjective algorithm for anatomy based dose optimization algorithm in high-dose-rate brachytherapy. *Physics in Medicine and Biology* **48**(3) 399–415.

Lahanas, M., E. Schreibmann, D. Baltas. 2003b. Multiobjective inverse planning for intensity modulated radiotherapy with constraint-free gradient-based optimization algorithms. *Physics in Medicine and Biology* **48**(17) 2843–2871.

Lal, T. N., M. Schroeder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, B. Schölkopf. 2004. Support vector channel selection in BCI. *IEEE Transactions on Biomedical Engineering* **51**(6) 1003–1010.

Lao, Z., D. Shen, Z. Xue, B. Karacali, S. M. Resnick, C. Davatzikos. 2004. Morphological classification of brains via high-dimensional shape transformations and machine learning methods. *NeuroImage* **21** 46–57.

Lauer, F., G. Bloch. 2008. Incorporating prior knowledge in support vector regression. *Machine Learning* **70** 89–118.

Lazzeroni, L., A. Owen. 2002. Plaid models for gene expression data. *Statistica Sinica* **12**(1) 61–86.

Ledberg, A., S.L. Bressler, M. Ding, R. Coppola, R. Nakamura. 2007. Large-scale visuomotor integration in the cerebral cortex. *Cerebral Cortex* **17** 44–62.

Lee, C.-H., M. Schmidt, A. Murtha, A. Bistritz, J. Sander, R. Greiner. 2005. Segmenting brain tumors with conditional random fields and support vector machines. *Computer Vision for Biomedical Image Applications*. 469–478.

Lee, D. D., H. S. Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* **40** 788–791.

Lee, E. K. 2008. *Optimization in Medicine*, chap. Optimization-based predictive models in medicine and biology. Springer, 127–151.

Lee, E. K., T.-L. Wu. 2007. Classification and disease prediction via mathematical programming. O. Seref, O. E. Kundakcioglu, P. M. Pardalos, eds., *Data Mining, Systems Analysis, and Optimization in Biomedicine*. American Institute of Physics, 1–42.

Lee, E.K., T. Fox, I. Crocker. 2001. Integer programming applied to intensity-modulated radiation treatment planning optimization. *Annals of Operations Research* **119** 165–181.

Lee, E.K., M. Zaider. 2003. Mixed integer programming approaches to treatment planning for brachytherapy – application to permanent prostate implants. *Annals of Operations Research* **119** 147–163.

Lee, J. K., P. D. Williams, S. Cheon. 2008. Data mining in genomics. *Clinics in Laboratory Medicine* **28**(1) 145–166.

Lee, S., A. Verri, eds. 2002. *Pattern Recognition with Support Vector Machines*. Springer, Niagara Falls, Canada. First International Workshop, SVM 2002.

Lehmann, C., T. Koenig, V. Jelic, L. Prichep, R. E. John, L.-O. Wahlund, Y. Dodge, T. Dierks. 2007. Application and comparison of classification algorithms for recognition of alzheimer's disease in electrical brain activity (EEG). *Journal of Neuroscience Methods* **161** 342–350.

Li, G.-Z., T.-Y. Liu, V. S. Cheng. 2006a. Classification of brain glioma by using SVMs bagging with feature selection. *BioDM*. 124–130.

Li, G.-Z., J. Yang, C.-Z. Ye, D.-Y. Geng. 2006b. Degree prediction of malignancy in brain glioma using support vector machines. *Computers in Biology and Medicine* **36** 313–325.

Li, Y. K., P. L. Yang, Y. J. Jian, S. M. Ren, H. X. Zhao. 2007. Application of support vector regression method in predicting soil erosion intensity of small watershed in the insensitive erosion areas. *Journal of Beijing Forestry University* **29**(3) 93–98.

Liang, H., J. D. Z. Chen. 2005. Assessment of the esophageal pressure in gastroesophageal reflux disease by the local regression. *Annals of Biomedical Engineering* **33**(6) 847–853.

Liang, Nan-Ying, Paramasivan Saratchandran, Guang-Bin Huang, Narasimhan Sundararajan. 2006. Classification of mental tasks from EEG signals using extreme learning machine. *International Journal of Neural Systems* **16**(1) 29–38.

Liu, J., W. Wang. 2003. OP-Cluster: Clustering by tendency in high dimensional space. *Third IEEE International Conference on Data Mining*. 187–194.

Liu, Y., L. Teverovskiy, O.T. Carmichael, R. Kikinis, M. Shenton, C.S. Carter, V.A. Stenger, S. Davis, H. Aizenstein, J. Becker, O. Lopez, C. Meltzer. 2004. Discriminative MR image feature analysis for automatic schizophrenia and alzheimer's disease classification. *Lecture Notes Computer Science* **3216** 393–401.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* **28**(2) 129–137.

Lodwick, W. A., S. McCourt, F. Newman, S. Humphries. 1999. *Computational Radiology and Imaging: Therapy and Diagnostics*, chap. Optimization Methods for Radiation Therapy Plans. Springer, 229–248.

Long, P. M., L. Tan. 1998. PAC learning axis aligned rectangles with respect to product distributions from multiple instance examples. *Machine Learning* **30** 7–22.

Louis, A. K. 2008. *Optimization in Medicine*, chap. Optimal reconstruction kernels in medical imaging. Springer, 153–168.

Lu, Y., S. Y. Lu, F. Fotouhi, Y. P. Deng, S. J. Brown. 2004. Incremental genetic k-means algorithm and its application in gene expression data analysis. *BMC Bioinformatics* **5**(172).

Lukas, L., A. Devos, J. A. K. Suykens, L. Vanhamme, F. A. Howe, C. Majós, A. Moreno-Torres, M. Van Der Graaf, A. R. Tate, C. Arús, S. Van Huffel. 2004. Brain tumor classification based on long echo proton MRS signals. *Artificial Intelligence in Medicine* **31**(1) 73–89.

Madeira, S. C., A. L. Oliveira. 2004. Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics* **1** 24–45.

Mammadov, M., A. Rubinov, J. Yearwood. 2007a. *Data Mining in Biomedicine*, *Optimization and Its Applications*, vol. 7, chap. An optimization approach to identify the relationship between features and output of a multi-label classifier. Springer, 141–167.

Mammadov, M. A., A. M. Rubinov, J. Yearwood. 2007b. The study of drug-reaction relationships using global optimization techniques. *Optimization Methods & Software* **22**(1) 99–126.

Mangasarian, O. L. 1994. *Nonlinear Programming*. SIAM, Philadelphia.

Mangasarian, O. L., W. N. Street, W. H. Wolberg. 1995. Breast cancer diagnosis and prognosis via linear programming. *Operations Research* **43** 570–577.

Mangasarian, O.L., E.W. Wild. 2008. Multiple instance classification via successive linear programming. *Journal of Optimization Theory and Applications* **137**(3) 555–568.

Maquelin, K., L.P. Choo-Smith, T. van Vreeswijk, H. P. Endtz, B. Smith, R. Bennett, H. A. Bruining, G. J. Puppels. 1999. Raman spectroscopic method for identification of clinically relevant microorganisms growing on solid culture medium. *Analytical Chemistry* **72**(1) 12–19.

Marchuk, Guri I. 1997. *Mathematical Modelling of Immune Response in Infectious Diseases*. Kluwer Academic Publishers.

Maron, O. 1998. Learning from ambiguity. Tech. rep., Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA. Ftp://publications.ai.mit.edu/ai-publications/pdf/AITR-1639.pdf.

Maron, O., A. L. Ratan. 1998. Multiple-instance learning for natural scene classification. *Proceedings of the Fifteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 341–349.

Martinez-Ramon, M., V. Koltchinskii, G. L. Heileman, S. Posse. 2006. fMRI pattern classification using neuroanatomically constrained boosting. *NeuroImage* **31** 1129–1141.

McAllister, S. R., R. Rajgaria, C. A. Floudas. 2007. Global pairwise sequence alignment through mixed-integer linear programming: a template-free approach. *Optimization Methods & Software* **22**(1) 127–144.

Men, C., H. E. Romeijn, Z. C. Taşkın, J. F. Dempsey. 2007. An exact approach to direct aperture optimization in IMRT treatment planning. *Physics in Medicine and Biology* **52** 7333–7352.

Mendola, J.D., S. Corkin. 1999. Visual discrimination and attention after bilateral temporal-lobe lesions: A case study. *Neuropsychologia* **37**(1) 91–102.

Meneses, C. N., C. A. S. Oliveira, P. M. Pardalos. 2007. *Data Mining in Biomedicine*, *Optimization and Its Applications*, vol. 7, chap. Mathematical programming formulations for problems in genomics and proteomics. Springer, 275–290.

Menze, B. H., M. P. Lichy, P. Bachert, B. M. Kelm, H.-P. Schlemmer, F. A. Hamprecht. 2006. Optimal classification of long echo time in vivo magnetic resonance spectra in the detection of recurrent brain tumors. *NMR in Biomedicine* **19**(5) 599–609.

Mourão-Miranda, J., K. J. Friston, M. Brammer. 2007. Dynamic discrimination analysis: A spatial-temporal SVM. *NeuroImage* **36** 88–99.

Mourão-Miranda, J., E. Reynaud, F. McGlone, G. Calvert, M. Brammer. 2006. The impact of temporal compression and space selection on SVM analysis of single-subject and multi-subject fMRI data. *NeuroImage* **33** 1055–1065.

Müller, Klaus Robert, Charles W. Anderson, Gary E. Birch. 2003. Linear and nonlinear methods for brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **11**(2) 165–169.

Murray, J. F., G. F. Hughes, K. Kreutz-Delgado. 2005. Machine learning methods for predicting failures in hard drives: A multiple-instance application. *The Journal of Machine Learning Research* **6** 783–816.

Musicant, D. R., J. M. Christensen, J. F. Olson. 2007. Supervised learning by training on aggregate outputs. *ICDM '07: Proceedings of the 2007 Seventh IEEEInternational Conference on Data Mining*. IEEE Computer Society, Washington, DC, USA, 252–261.

Nahapetyan, A., S. Busygin, P. M. Pardalos. 2008. *Mathematical Modelling of Biosystems*, chap. An improved heuristic for consistent biclustering problems. Springer, 185–198.

Navarre, W. W., A. Zychlinsky. 2000. Pathogen-induced apoptosis of macrophages: A common end for different pathogenic strategies. *Cellular Microbiology* **2**(4) 265–273.

Noble, W. S. 2004. *Kernel Methods in Computational Biology*, chap. Support vector machine applications in computational biology. MIT Press, 71–92.

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

Notingher, I., S. Verrier, S. Haque, J. M. Polak, L. L. Hench. 2003. Spectroscopic study of human lung epithelial cells (a549) in culture: living cells versus dead cells. *Biopolymers* **72**(4) 230–240.

Notingher, I., S. Verrier, H. Romanska, A. E. Bishop, J. M. Polak, L. L. Hench. 2002. In situ characterisation of living cells by Raman spectroscopy. *Spectroscopy – An International Journal* **16**(2) 43–51.

Olson, B. P., J. Si, J. Hu, J. He. 2005. Closed-loop cortical control of direction using support vector machines. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **13**(1) 72–80.

Osuna, R. F. E., F. Girosi. 1997. An improved training algorithm for support vector machines. *IEEE Workshop on Neural Networks for Signal Processing*. 276–285.

Owen, C. A., J. Selvakumaran, I. Notingher, G. Jell, L. L. Hench, M. M. Stevens. 2006. In vitro toxicology evaluation of pharmaceuticals using Raman micro-spectroscopy. *Journal of Cellular Biochemistry* **99**(1) 178–186.

Pai, P. F., W. C. Hong. 2007. A recurrent support vector regression model in rainfall forecasting. *Hydrological Processes* **21**(6) 819–827.

Pardalos, P. M., V. L. Boginski, O. A. Prokopyev, W. Suharitdamrong, P. R. Carney, W. Chaovalitwongse, A. Vazacopoulos. 2005. *Essays and Surveys in Global Optimization*, chap. Optimization techniques in medicine. Springer, 211–232.

Pardalos, P. M., W. Chaovalitwongse, L. D. Iasemidis, J. C. Sackellares, D.-S. Shiau, P. R. Carney, O. A. Prokopyev, V. A. Yatsenko. 2004. Seizure warning algorithm based on optimization and nonlinear dynamics. *Mathematical Programming* **101**(2) 365–385.

Parra, L. C., C. D. Spence, A. D. Gerson, P. Sajda. 2005. Recipes for the linear analysis of EEG. *NeuroImage* **28** 326–341.

Pascual-Montano, A., P. Carmona-Saez, M. Chagoyen, F. Tirado, J. M. Carazo, R. D. Pascual-Marqui. 2006. bioNMF: A versatile tool for non-negative matrix factorization in biology. *BMC Bioinformatics* **7**(366).

Pessoa, L., S. Padmala. 2007. Decoding near-threshold perception of fear from distributed single-trial brain activation. *Cerebral Cortex* **17** 691–701.

Pflugfelder, D., J. J. Wilkens, U. Oelfke. 2008. Worst case optimization: A method to account for uncertainties in the optimization of intensity modulated proton therapy. *Physics in Medicine and Biology* **53** 1689–1700.

Platt, J. 1999. Fast training of SVMs using sequential minimal optimization. B. Schölkopf, C. J. C. Burges, A. J. Smola, eds., *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, 185–208.

Prados, J., A. Kalousis, J.-C. Sanchez, L. Allard, O. Carrette, M. Hilario. 2004. Mining mass spectra for diagnosis and biomarker discovery of cerebral accidents. *Proteomics* **4** 2320–2332.

Prasad, K. V., A. Taiyab, D. Jyothi, U. K. Srinivas, A. S. Sreedhar. 2007. Heat shock transcription factors regulate heat induced cell death in a rat histiocytoma. *Journal of Biosciences* **32**(3) 585–593.

Pyrgiotakis, G., T. K. Bhowmick, K. Finton, A. K. Suresh, S. G. Kane, J. R. Bellare, B. M. Moudgil. 2008. Cell (A549)-particle (Jasada Bhasma) interactions using Raman spectroscopy. *Biopolymers* **89**(6) 555–64.

Pyrgiotakis, G., O. E. Kundakcioglu, K. Finton, P. M. Pardalos, K. Powers, B. M. Moudgil. 2009. Cell death discrimination with Raman spectroscopy and support vector machines. *Annals of Biomedical Engineering* **37**(7) 1464–1473.

Qi, X., Y. Han. 2007. Incorporating multiple SVMs for automatic image annotation. *Pattern Recognition* **40** 728–741.

Quddus, A., P. Fieguth, O. Basir. 2005. Adaboost and support vector machines for white matter lesion segmentation in MR images. *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. 463–466.

Ragle, M. A., J. C. Smith, P. M. Pardalos. 2007. An optimal cutting-plane algorithm for solving the non-unique probe selection problem. *Annals of Biomedical Engineering* **35**(11) 2023–2030.

Ray, S. 2005. Learning from data with complex interactions and ambiguous labels. Ph.D. thesis, University of Wisconsin - Madison.

Ray, S., M. Craven. 2005. Supervised versus multiple instance learning: An empirical comparison. *ICML '05: Proceedings of the 22nd international conference on Machine learning*. ACM Press, New York, NY, USA, 697–704.

Raykar, V. C., B. Krishnapuram, J. Bi, M. Dundar, R. B. Rao. 2008. Bayesian multiple instance learning: Automatic feature selection and inductive transfer. *ICML '08: Proceedings of the 25th international conference on Machine learning*. ACM, New York, NY, USA, 808–815.

Rinaldi, L., P. Gallo, F. Ranzato, D. Luise, D. Colavito, M. Motta, A. Guglielmo, E. Del Giudice, C. Romualdi, E. Ragazzi, A. Darrigo, M. Dalle Carbonare, B. Leontino, A. Leon. 2006. Longitudinal analysis of immune cell phenotypes in early stage multiple sclerosis: distinctive patterns characterize MRI-active patients. *Brain* **129** 1993–2007.

Robinson, J. E., M. J. Wizenberg, W. A. McCready. 1974. Combined hyperthermia and radiation suggest and alternative to heavy particle therapy for reduced oxygen enhancement ratios. *Nature* **251**(5475) 521–522.

Robinson, R. J., M. J. Oldham, R. E. Clinkenbeard, P. Rai. 2006. Experimental and numerical smoke carcinogen deposition in a multi-generation human replica tracheobronchial model. *Annals of Biomedical Engineering* **34**(3) 11–13.

Rubinov, A. M., J. Udon. 2003. Skeletons of finite sets of points. Tech. rep., Centre for Informatics and Applied Optimization of the University of Ballarat.

Russell, F., M. Akay. 1996. A comparison of analytical methods for the study of fractional brownian motion. *Annals of Biomedical Engineering* **24**(4) 1–3.

Sabesan, S., L. Good, N. Chakravarthy, K. Tsakalis, P. M. Pardalos, L. Iasemidis. 2008. *Optimization in Medicine*, chap. Global optimization and spatial synchronization changes prior to epileptic seizures. Springer, 103–125.

Sakhanenko, N. A., G. F. Luger. 2006. Shock physics data reconstruction using support vector regression. *International Journal of Modern Physics* **17**(9) 1313–1325.

Sapareto, S. A., W. C. Dewey. 1984. Thermal dose determination in cancer therapy. *International Journal of Radiation Oncology, Biology, Physics* **10**(6) 787–800.

Schapire, R. 2001. The boosting approach to machine learning: An overview.

Schölkopf, B., A. J. Smola. 2002. *Learning with Kernels*. MIT Press, Cambridge.

Segal, E., B. Taskar, A. Gasch, N. Friedman, D. Koller. 2001. Rich probabilistic models for gene expression. *Bioinformatics* **17** 243–252.

Seref, O., C. Cifarelli, O. E. Kundakcioglu, P. M. Pardalos, M. Ding. 2007. Detecting categorical discrimination in a visuomotor task using selective support vector machines. H. R. Arabnia, M. Q. Yang, J. Y. Yang, eds., *Proceedings of the 2007 International Conference on Bioinformatics & Computational Biology (BIOCOMP)*, vol. 2. 580–587.

Seref, O., O. E. Kundakcioglu, M. Bewernitz. 2008a. *Encyclopedia of Healthcare Information Systems*, chap. Support vector machines in neuroscience. IDEA Group Inc., 1283–1293.

Seref, O., O. E. Kundakcioglu, P. M. Pardalos, eds. 2008b. *Data Mining, Systems Analysis, and Optimization in Biomedicine*. 953, American Institute of Physics.

Seref, O., O. E. Kundakcioglu, P. M. Pardalos. 2008c. Selective linear and nonlinear classification. P. M. Pardalos, P. Hansen, eds., *CRM Proceedings and Lecture Notes*, vol. 45. 211–234.

Seref, O., O. E. Kundakcioglu, O. A. Prokopyev, P. M. Pardalos. 2009. Selective support vector machines. *Journal of Combinatorial Optimization* **17**(1) 3–20.

Shawe-Taylor, J., N. Cristianini. 2004. *Kernel methods for pattern analysis*. Cambridge University Press, Cambridge, UK.

Sheng, Q., Y. Moreau, B. De Moor. 2003. Biclustering microarray data by Gibbs sampling. *Bioinformatics* **19** 196–205.

Shepard, D. M., M. C. Ferris, G. H. Olivera, T. R. Mackie. 1999. Optimizing the delivery of radiation therapy to cancer patients. *SIAM Review* **41**(4) 721–744.

Shoker, L., S. Sanei, A. Sumich. 2005. Distinguishing between left and right finger movement from EEG using SVM. *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. 5420–5423.

Sitaram, R., H. Zhang, C. Guan, M. Thulasidas, Y. Hoshi, A. Ishikawa, K. Shimizu, N. Birbaumer. 2007. Temporal classification of multichannel near-infrared spectroscopy signals of motor imagery for developing a brain-computer interface. *NeuroImage* **34** 1416–1427.

Solovyan, V., Z. Bezvenyuk, V. Huotari, T. Tapiola, T. Suuronen, A. Salminen. 1998. Distinct mode of apoptosis induced by genotoxic agent etoposide and serum withdrawal in neuroblastoma cells. *Molecular Brain Research* **62**(1) 43–55.

Street, W. N., O. L. Mangasarian, W. H. Wolberg. 1995. An inductive learning approach to prognostic prediction. *in Machine Learning: Proceedings of the Twelfth International Conference*. Morgan Kaufmann, 522–530.

Strickler, S. S., A. V. Gribenko, A. V. Gribenko, T. R. Keiffer, J. Tomlinson, T. Reihle, V. V. Loladze, G. I. Makhatadze. 2006. Protein stability and surface electrostatics: A charged relationship. *Biochemistry* **45**(9) 2761–2766.

Sujansky, W. 2001. Heterogeneous database integration in biomedicine. *Journal of Biomedical Informatics* **34**(4) 285–298.

Sun, Y. F., Y. C. Liang, C. G. Wu, X. W. Yang, H. P. Lee, W. Z. Lin. 2004. Estimate of error bounds in the improved support vector regression. *Progress in Natural Science* **14**(4) 362–364.

Suykens, J.A.K., J. Vandewalle. 1999. Least squares support vector machine classifiers. *Neural Processing Letters* **9** 293–300.

T.-H.Wu. 1997. A note on a global approach for general 0-1 fractional programming. *European Journal Of Operational Research* **16** 220–223.

Tadashi, K., K. Takao, N. Takeo, A. Hiroshi, E. Shin, N. Masaaki, T. Hideaki, Y. Tadashi, K. Seiji, T. Ryuichi. 2004. Mild heat shock induces autophagic growth arrest, but not apoptosis in u251-mg and u87-mg human malignant glioma cells. *Journal of Neuro-Oncology* **68** 101–111.

Tanay, A., R. Sharan, R. Shamir. 2002. Discovering statistically significant biclusters in gene expression data. *Bioinformatics* **18** 136–144.

Tanay, A., R. Sharan, R. Shamir. 2004. Biclustering algorithms: A survey. Tech. rep., Tel Aviv University.

Tao, Q., S. Scott, N. V. Vinodchandran, T. T. Osugi. 2004. SVM-based generalized multiple-instance learning via approximate box counting. *ICML '04: Proceedings of the Twenty-first International Conference on Machine Learning*. ACM Press, New York, NY, USA, 799–806.

Tawarmalani, M., S. Ahmed, N. V. Sahinidis. 2002. Global optimization of 0–1 hyperbolic programs. *Journal of Global Optimization* **24**(4) 385–416.

Thai, M. T., Z. Cai, D. Z. Du. 2007a. Genetic networks: processing data, regulatory network modelling and their analysis. *Optimization Methods & Software* **22**(1) 169–185.

Thai, M. T., P. Deng, W. Wu, T. Znati. 2007b. Approximation algorithms of non-unique probes selection for biological target identification. O. Seref, O. E. Kundakcioglu, P. M. Pardalos, eds., *Data Mining, Systems Analysis, and Optimization in Biomedicine*. American Institute of Physics, 174–184.

Thulasidas, M., C. Guan, J. Wu. 2006. Robust classification of EEG signal for brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**(1) 24–29.

Trafalis, T. B., H. Ince. 2000. Support vector machine for regression and applications to financial forecasting. *International Joint Conference on Neural Networks*, vol. 6. Como, Italy, 348–353.

Ugur, O., G. W. Weber. 2007. Optimization and dynamics of gene-environment networks with intervals. *Journal of Industrial and Management Optimization* **3**(2) 357–379.

Uthman, B., M. Bewernitz, C.-C. Liu, G. Ghacibeh. 2007. Optimization of epilepsy treatment with vagus nerve stimulation. O. Seref, O. E. Kundakcioglu, P. M. Pardalos, eds., *Data Mining, Systems Analysis, and Optimization in Biomedicine*. American Institute of Physics, 308–315.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag.

Vapnik, V. 1998. *Statistical Learning Theory*. Wiley, New York.

Verrier, S., I. Notingher, J. M. Polak, L. L. Hench. 2004. In situ monitoring of cell death using Raman microspectroscopy. *Biopolymers* **74**(1-2) 157–162.

Viola, P., J. C. Platt, C. Zhang. 2006. Multiple instance boosting for object detection. *Neural Information Processing Systems*, vol. 18. MIT Press, Vancouver, British Columbia, Canada, 1419–1426.

Wang, J., J. Zucker. 2000. Solving the multiple-instance problem: A lazy learning approach. *Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1119–1125.

Wang, Y. M., R. T. Schultz, R. T. Constable, L. H. Staib. 2003. Nonlinear estimation and modeling of fMRI data using spatio-temporal support vector regression. *Lecture Notes in Computer Science* **2732** 647–659.

Webb, S. 1991. Optimization by simulated annealing of three-dimensional conformal treatment planning for radiation fields defined by a multileaf collimator. *Physics in Medicine and Biology* **36**(9) 1201–1226.

Weston, J., S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, V. Vapnik. 2000. Feature selection for SVMs. *NIPS*. 668–674.

Widjaja, E., G. H. Lim, A. An. 2008a. A novel method for human gender classification using Raman spectroscopy of fingernail clippings. *The Royal Society of Chemistry* **133** 493–498.

Widjaja, E., W. Zheng, Z. Huang. 2008b. Classification of colonic tissues using near-infrared Raman spectroscopy and support vector machines. *International Journal of Oncology* **32**(3) 653–662.

Wolf, A., J. B. Swift, H. L. Swinney, J. A. Vastano. 1985. Determining lyapunov exponents from a time series. *Physica D* **16** 285–317.

Wolsey, L. A. 1998. *Integer Programming*. Wiley-Interscience, New York.

Wu, Z.-L., C.-H. Li, J. K.-Y. Ng, K.R.P.H. Leung. 2007. Location estimation via support vector regression. *IEEE Transactions on Mobile Computing* **6**(3) 311–321.

Xie, X. S., W. T. Liu, B. Y. Tang. 2008. Space based estimation of moisture transport in marine atmosphere using support vector regression. *Remote Sensing of Environment* **112**(4) 1846–1855.

Xing, E.P., R.M. Karp. 2001. CLIFF: Clustering of high-dimensional microarray data via iterative feature filtering using normilized cuts. *Bioinformatics Discovery Note* **17** 306–315.

Yamamoto, K., F. Asano, T. Yamada, N. Kitawaki. 2006. Detection of overlapping speech in meetings using support vector machines and support vector regression. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences* **E89A**(8) 2158–2165.

Yogalingam, G., A. M. Pendergast. 1997. Serum withdrawal and etoposide induce apoptosis in human lung carcinoma cell line a549 via distinct pathways. *Apoptosis* **2**(2) 199–206.

Yogalingam, G., A. M. Pendergast. 2008. ABL kinases regulate autophagy by promoting the trafficking and function of lysosomal components. *Journal of Biological Chemistry* **283**(51) 35941–35953.

Yoon, U., J.-M. Lee, K. Im, Y.-W. Shin, B. H. Cho, I. Y. Kim, J. S. Kwon, S. I. Kim. 2007. Pattern classification using principal components of cortical thickness and its discriminative pattern in schizophrenia. *NeuroImage* **34** 1405–1415.

Yoon, U., J. M. Lee, J. J. Kim, S. M. Lee, I. Y. Kim, J. S. Kwon, S. I. Kim. 2003. Modified magnetic resonance image based parcellation method for cerebral cortex using successive fuzzy clustering and boundary detection. *Annals of Biomedical Engineering* **31**(4) 441–447.

Yushkevich, P., A. Dubb, Z. Xie, R. Gur, R. Gur, J. Gee. 2005. Regional structural characterization of the brain of schizophrenia patients. *Academic Radiology* **12**(10) 1250–1261.

Zarnitsyn, V. G., M. R. Prausnitz. 2004. Physical parameters influencing optimization of ultrasound-mediated DNA transfection. *Ultrasound in Medicine & Biology* **30**(4) 527–538.

Zhang, Q., S. Goldman. 2001. EM-DD: An improved multiple-instance learning technique. *Neural Information Processing Systems*, vol. 14. MIT Press, Vancouver, British Columbia, Canada, 1073–1080.

Zibulevsky, M., B. A. Pearlmutter. 2001. Blind source separation by sparse decomposition in a signal dictionary. *Neural Computation* **13**(4) 863–882.

# BIOGRAPHICAL SKETCH

O. Erhun Kundakcioglu received his Ph. D. degree in Industrial and Systems Engineering at the University of Florida. His research focuses on optimization techniques for pattern recognition and machine learning. Mr. Kundakcioglu is also interested in production and inventory planning problems. He is the 2008 recipient of the Florida Chapter Scholarship given by the Healthcare Information and Management Systems Society Foundation. His contribution to the department has also been recognized by rewarding him the Graduate Student Award for Excellence in Research and Teaching in 2008 and 2009, respectively.