Appl. Comput. Math. **8** (2009), no.1, pp.23-35

CLASSIFICATION VIA MATHEMATICAL PROGRAMMING
SURVEY

PANOS M. PARDALOS †, O. ERHUN KUNDAKCIOGLU †, §

Abstract. This survey concerns applications of mathematical programming in the context of classification. We mainly discuss two supervised learning methods: Support Vector Machines (SVMs) and consistent biclustering together with their extensions. We also refer to some recently proposed classification techniques that utilize optimization theory.

Keywords: Mathematical programming, support vector machines, consistent biclustering, convex optimization, integer programming.

AMS Subject Classification: 90Cxx, 68-02

1. Introduction

Machine learning can be defined as the process by which a computer system improves its performance based on previous results. There are very successful implementations of machine learning such as search engines, language processing, financial analysis, medical diagnosis, and DNA sequence classification. Most of these applications rely on *pattern recognition* which is generally concerned with classifying objects based on their characteristics. Characteristics of an object are generally referred to as *features*, which are the measures that distinguish that object from the other objects. Similarity between two objects can be evaluated as a function of features they possess. Objects can be grouped into classes based on their similarity. These classes are represented in different ways such as approximation functions or boundary functions between the classes. Arranging objects into such classes based on their position relative to these functions is called *classification*.

Machine learning within the classification framework can be categorized into two main classes. *Supervised learning* is the capability of a system to learn from a set of examples, which is a set of input/output pairs. The input is a *vector of features* of an object, and the output is the *label* for the object (i.e., class that the object belongs to). A set of objects with feature vectors and a class labels is called a *training set*. This set is used to derive classification functions. The trained system is capable of predicting the label of an unlabeled object. A set of objects with feature vectors whose label information is unknown is called a *test set*. 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, which can be considered as the *supervisor*. On the contrary, *unsupervised learning* is the case where the objects are not labeled with any class information, and learning is about forming classes of objects based on similarities between their features.

†Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, P.O. Box 116595, Gainesville, Florida 32611-6595; e-mail: pardalos@ufl.edu, erhun@ufl.edu

§*Manuscript received 17 March 2009.*

2. Support Vector Machines

Initially developed by Vapnik [48], *Support Vector Machines* (SVMs) are the state-of-the-art supervised machine learning methods. SVM classifiers classify pattern vectors which are assumed to belong to two different classes. The classification function is defined by a hyperplane that separates two classes. There are infinitely many hyperplanes that separate the two classes but the SVM classifier finds the hyperplane 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 for SVM classifiers are derived. When there are sufficiently many data points in the training set, SVMs are proven to minimize the generalization error for any distribution. SVMs can be extended to classification of nonlinear data by implicitly embedding the original data in a nonlinear space using *kernel functions* [45]. Techniques for generalizing the SVM classifiers for multiple classes are introduced theoretically in [48] but there are many drawbacks of hyperplane based multi-class learning techniques [6].

SVMs have a wide spectrum of application areas ranging from pattern recognition [31] and text categorization [23] to biomedicine [7, 13, 36, 39], brain-computer interface [28, 18], and financial applications [22, 47]. The training is performed by optimizing a quadratic convex function that is subject to linear constraints. There are many general purpose methods to solve QP problems such as quasi-newton, primal-dual and interior-point methods [4]. The general purpose methods are suitable for small size problems but are not fast enough for large problems. Faster methods usually involve chunking [37] and decomposition [40] techniques, which use subsets of points to find the optimal hyperplane. SVM Light [24] and LIBSVM [21] are among the most frequently used implementations that use chunking and decomposition methods efficiently. There are also alternative methods such as Generalized Proximal SVM (GEPSVM) [34] that approximate the two classes with two hyperplanes instead of a single hyperplane separating them.

We start the review for SVMs with *maximal margin classifier* and further extend it to *soft margin classifiers*. Alternative formulations for different error norms are also given for SVMs.

2.1. Formulation.

In a typical *binary classification* problem, sets $\mathbf{C}^+$ and $\mathbf{C}^-$ are composed of pattern vectors $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$. If $\mathbf{x}_i \in \mathbf{C}^+$ then it is given the label $y_i = 1$; otherwise $\mathbf{x}_i \in \mathbf{C}^-$ and is given the label $y_i = -1$. The classification problem deals with determining which class a new pattern vector $\mathbf{x}_i \notin \{\mathbf{C}^+ \cup \mathbf{C}^-\}$ belongs to. SVM classifiers solve this problem by finding a hyperplane $(\mathbf{w}, b)$ that separates these two classes from each other with the maximum interclass margin.

2.2. Maximal Margin Classifier.

Maximal margin classifier is the simplest form of SVM classifiers that solves the problem of finding a separating hyperplane with the maximum interclass 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 can be extended to produce feasible separation rules for data sets that are nor linearly separable. The solution to the optimization problem in the maximal margin classifier minimizes the bound on the generalization error [49]. 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 [4].

We start our review with the definition of a hyperplane, $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$, which is represented as the normal vector $\mathbf{w}$ and the offset parameter b. The *functional distance* between a data point $\mathbf{x}_i$ and the hyperplane is given by $\langle \mathbf{w}, \mathbf{x}_i \rangle + b$ and the *geometric distance* is $(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)/\|\mathbf{w}\|$.

There is inherent degree of freedom in specifying a hyperplane as $(\lambda\mathbf{w}, \lambda b)$. A *canonical hyperplane* is the one from which the closest pattern vector has a functional distance of 1, i.e., $\min_{i=1,\ldots,n} |\langle \mathbf{w}, \mathbf{x}_i \rangle + b| = 1$.

Next, consider two pattern vectors $\mathbf{x}^+$ and $\mathbf{x}^-$, belonging to classes $\mathbf{C}^+$ and $\mathbf{C}^-$, respectively and they are the closest pattern vectors to a canonical hyperplane, such that $\langle \mathbf{w}, \mathbf{x}^+ \rangle + b = 1$ and $\langle \mathbf{w}, \mathbf{x}^- \rangle + b = -1$. It is easy to show that the geometric margin between these pattern vectors and the hyperplane are both equal to $1/\|\mathbf{w}\|$.

Maximizing the margin $1/\|\mathbf{w}\|$ for the canonical hyperplane is equivalent to minimizing $\|\mathbf{w}\|$ or $\|\mathbf{w}\|^2$. In the following optimization problem, canonical hyperplane is ensured for each point $\mathbf{x}_i$ with a label y_i due to constraints (1b) while the margin is maximized by minimizing $\|\mathbf{w}\|$.

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 && && \text{(1a)} \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 && i = 1 \ldots, n. && \text{(1b)} \end{aligned}$$

Using the optimal solution for (1), a new pattern vector $\mathbf{x}'$ can be classified as positive if $\langle \mathbf{w}^*, \mathbf{x}' \rangle + b^* > 0$, and negative otherwise.

2.3. **Soft Margin Classifier.** Most real life problems are composed of non separable data which is generally due to noise. In this case *slack variables* ξ_i are introduced for each pattern vector $\mathbf{x}_i$ in the training set. Slack variables allow misclassifications for each pattern vector; however, they are subject to a penalty of $C/2$ to avoid trivial solutions. The maximum margin formulation can be augmented to soft margin formulation as

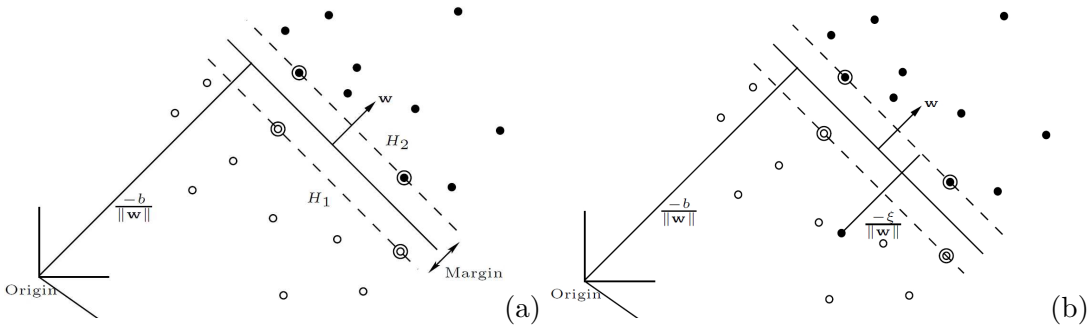

FIGURE 1. Maximal Margin Classifier (a) and Soft Margin Classifier (b)

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 && && \text{(2a)} \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) \geq 1 - \xi_i && i = 1, \ldots, n, && \text{(2b)} \end{aligned}$$

where nonnegativity of the slack variables are assured implicitly since the solution cannot be optimal when $\xi_i < 0$ for any pattern vector.

The 2-norm of the slack are penalized in the objective of (2). An alternative formulation involves penalization of the 1-norm slack variables in the objective. However, for the 1-norm case, we need to impose nonnegativity on the slack variables explicitly.

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i, && && \text{(3a)}\\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i && i = 1, \ldots, n, && \text{(3b)}\\ & \xi_i \geq 0 && i = 1, \ldots, n. && \text{(3c)} \end{aligned}$$

Both 1-norm and 2-norm SVM formulations are called the *primal* formulations and equivalent *dual* formulations can be obtained using mathematical programming theory. The significance of the dual formulations is that they do not involve inequality constraints, and they allow the *kernel trick* to be introduced for nonlinear classification. In order to obtain the dual formulation of the SVM problem, we first derive the *Lagrangian function* of the primal problem. This function provides a lower bound for the solution of the primal problem. Next, we differentiate the Lagrangian function with respect to the primal variables and impose stationarity. We substitute the equivalent expressions for each primal variable back in the Lagrangian function or add them as constraints. 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.

2.4. **Dual Formulation and Kernel Trick.** The Lagrangian function for the 2-norm SVM primal problem is given as follows.

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 - \sum_{i=1}^{n}\alpha_i\left[y_i(\langle \mathbf{w}, \mathbf{x_i}\rangle + b) - 1 + \xi_i\right]. \tag{4}$$

Differentiating L with respect to the primal variables $\mathbf{w}$ and b, and assuming stationarity, we obtain

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} y_i\alpha_i\mathbf{x}_i = 0; \quad \frac{\partial L}{\partial b} = \sum_{i=1}^{n} y_i\alpha_i = 0; \quad \frac{\partial L}{\partial \xi_i} = C\xi_i - \alpha_i = 0. \tag{5}$$

We can substitute the expressions in (5) back in the Lagrangian function to obtain the following dual formulation, which gives the hyperplane $\mathbf{w}^* = \sum_{i=1}^{n} y_i\alpha_i^*\mathbf{x}_i$.

$$\begin{aligned} \max \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_iy_j\alpha_i\alpha_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle - \frac{1}{2C}\sum_{i=1}^{n}\alpha_i^2, && && \text{(6a)}\\ \text{subject to} \quad & \sum_{i=1}^{n} y_i\alpha_i = 0 && && \text{(6b)}\\ & \alpha_i \geq 0 && i = 1, \ldots, n. && \text{(6c)} \end{aligned}$$

Note from Karush-Kuhn-Tucker complementarity conditions that, the constraints in the primal problem are binding for those with the corresponding dual variable $\alpha_i^* > 0$. Knowing $\mathbf{w}^*$, we can find b^* using

$$b^* = \sum_{i:\alpha_i^*>0} y_i - \langle \mathbf{w}^*, \mathbf{x}_i\rangle. \tag{7}$$

The derivation for the 1-norm dual formulation is very similar to that of 2-norm, which is given as

$$\max \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \tag{8a}$$

$$\text{subject to} \quad \sum_{i=1}^{n} y_i \alpha_i = 0 \tag{8b}$$

$$0 \leq \alpha_i \leq C \qquad i = 1, \ldots, n. \tag{8c}$$

Kernels are introduced in classification to provide enhanced similarity measures between pattern vectors. They basically transform the original *input space*, $\mathcal{X}$ to a usually higher dimensional dot-product space $\mathcal{H}$ called the *feature space*, with a map $\mathbf{\Phi} : \mathcal{X} \to \mathcal{H}$, such that $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) \rangle$. The main concept is focused on the dot product of two mapped pattern vectors. Mapping the pattern vectors may become computationally intractable, while implicitly finding their dot products in the feature space have the same complexity as in the input space, in general.

Kernel $\mathbf{K}$ is required to be positive semidefinite in order to define a dot product space and create a feature map. Here 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* [42]. In the machine learning literature, the kernel trick is introduced by Mercer's theorem and explains the geometry of feature spaces [14]. 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 $\mathcal{X}$. The convenience of kernels in SVMs is highlighted with the dual formulation. The linear dot product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ can be replaced with an appropriate non-linear kernel $\mathbf{K}$.

2.5. **Research Directions.** Recent advances in SVM classifiers are based on generalizations of traditional classification problem. Seref *et al.* [43] introduce novel *selective* linear and nonlinear classification methods, in which 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. Kernelized hard selection problems are also developed for classification. However, these combinatorial problems cannot be solved in polynomial time unless $\mathcal{P} = \mathcal{NP}$ [44]. Alternative approaches are proposed with relaxed formulations. The selective nature of the these formulations are satisfied 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 problematic pattern vectors, we can find those with better classification results.

Two variations of the free slack method, namely pooled free slack (PFS) and free slack per set (FSS), are introduced for selective linear classification together with kernelized dual formulations for selective nonlinear classification. These methods are further extended to direct separation by increasing the total free slack to diminish the effect of multiple pattern vectors per set and provide more flexibility for the hyperplane to reorient itself with respect to well separated pattern vectors. The performance of iterative elimination and direct selection algorithms are compared

with each other, as well as with a naïve elimination algorithm that uses standard SVM method and ideas from the proposed methods. Results are reported for linear and nonlinear simulated data.

Kundakcioglu *et al.* [27] consider the margin maximization 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. A combinatorial margin maximization problem is formulated for multiple instance classification which is proved to be $\mathcal{NP}$-hard. Kernel trick is applied on this formulation for classifying nonlinear MIL data. A branch and bound algorithm is proposed that outperforms a leading commercial solver in terms of the best integer solution and optimality gap in a majority of image annotation and molecular activity prediction test cases. The major difference between the MIL setting and the selective setting is the interpretation of negative bags. In selective learning, a selection is performed on negative bags as well as positive bags. In MIL, on the other hand, only actual positives are to be discovered where all negative instances must be kept.

Next, we introduce *consistent biclustering*, another classification technique that employs mathematical programming techniques.

3. Consistent Biclustering

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.

A data set is normally given as a rectangular $m \times n$ matrix A, where each column represents a data sample and each row represents a feature

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

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

Biclustering is applied by simultaneous classification of the samples and features 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). Formally biclustering can be 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

$$\begin{aligned}
S_1, S_2, \ldots, S_k &\subseteq \{a^j\}_{j=1,\ldots,n},\\
\bigcup_{r=1}^{k} S_r &= \{a^j\}_{j=1,\ldots,n},\\
S_\zeta \bigcap S_\xi &= \emptyset \Leftrightarrow \zeta \neq \xi,\\
F_1, F_2, \ldots, F_k &\subseteq \{a_i\}_{i=1,\ldots,m},\\
\bigcup_{r=1}^{k} F_r &= \{a_i\}_{i=1,\ldots,m},\\
F_\zeta \bigcap F_\xi &= \emptyset \Leftrightarrow \zeta \neq \xi,
\end{aligned}$$

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

The ultimate goal in a biclustering problem is to find a classification for which samples from the same class have *similar* values for that class' characteristic features. The visualization of a reasonable classification should reveal a block-diagonal or "checkerboard" pattern. A detailed survey on biclustering techniques can be found in [10] and [32].

One of the early algorithms to obtain an appropriate biclustering is proposed in [20], 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 [12], 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 [8].

Dhillon [16] proposes another biclustering method 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 [25].

In [17], 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 [46].

The concept of *consistent biclustering* is introduced by Busygin *et al.* [11]. 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.

An improved heuristic procedure is proposed in [35], where 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 [2, 3, 19, 50, 51]), confirm that the algorithm outperforms the previous results in the quality of solution as well as computation time. Busygin *et al.* [9] use consistent biclustering to analyze scalp EEG data obtained from epileptic patients undergoing treatment with a vagus nerve stimulator (VNS).

3.1. **Formulation.** 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

$$C_S = AS(S^TS)^{-1} = (c^S_{i\xi})_{m\times r}, \tag{9}$$

$$C_F = A^TF(F^TF)^{-1} = (c^F_{j\xi})_{n\times r}. \tag{10}$$

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

$$a_i \in \hat{F}_{\hat{r}} \Longrightarrow c^S_{i\hat{r}} > c^S_{i\xi}, \qquad \forall \xi, \xi \neq \hat{r}. \tag{11}$$

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.,

$$a^j \in \hat{S}_{\hat{r}} \Longrightarrow c^F_{j\hat{r}} > c^F_{j\xi}, \qquad \forall \xi, \xi \neq \hat{r}. \tag{12}$$

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 (11) and (12) hold for all elements of the corresponding classes, where matrices C_S and C_F are defined according to (9) and (10), respectively.

A data set is *biclustering-admitting* if some consistent biclustering for it 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.

Theorem 3.1. *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 [11] for the proof of Theorem 3.1. It also follows from the proven conic separability that convex hulls of classes do not intersect.

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.

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 α 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 (11) can be violated by adding a slightly

different sample to the data set. In other words, if α 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 [35] by relaxing (11) and (12) as

$$a_i \in F_{\hat{r}} \Longrightarrow c^S_{i\hat{r}} > \alpha^S_i + c^S_{i\xi}, \qquad \forall \xi, \xi \neq \hat{r}, \tag{13}$$

and

$$a^j \in S_{\hat{r}} \Longrightarrow c^F_{j\hat{r}} > \alpha^F_j + c^F_{j\xi}, \qquad \forall \xi, \xi \neq \hat{r}, \tag{14}$$

respectively, where $\alpha^F_j > 0$ and $\alpha^S_i > 0$.

Another relaxation in [35] is *multiplicative consistent biclustering* where (11) and (12) are replaced with

$$a_i \in F_{\hat{r}} \Longrightarrow c^S_{i\hat{r}} > \beta^S_i c^S_{i\xi}, \qquad \forall \xi, \xi \neq \hat{r}, \tag{15}$$

and

$$a^j \in S_{\hat{r}} \Longrightarrow c^F_{j\hat{r}} > \beta^F_j c^F_{j\xi}, \qquad \forall \xi, \xi \neq \hat{r}, \tag{16}$$

respectively, where $\beta^F_j > 1$ and $\beta^S_i > 1$.

Supervised biclustering uses accurate data sets that are called the *training set* to classify features to formulate consistent, α-consistent and β-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 α and β 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 $c^S_{i\xi}$ using (9). 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, α-consistent and β-consistent biclustering problems are formulated as follows.

CB:

$$\max_x \quad \sum_{i=1}^m x_i \tag{17a}$$

$$\text{subject to} \quad \frac{\sum_{i=1}^m a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^m f_{i\hat{r}} x_i} > \frac{\sum_{i=1}^m a_{ij} f_{i\xi} x_i}{\sum_{i=1}^m f_{i\xi} x_i}, \ \ \forall \hat{r}, \xi \in \{1, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}} \tag{17b}$$

$$x_i \in \{0, 1\}, \ \ \forall i \in \{1, \ldots, m\}, \tag{17c}$$

α-CB:

$$\max_x \quad \sum_{i=1}^m x_i \tag{18a}$$

$$\text{subject to} \quad \frac{\sum_{i=1}^m a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^m f_{i\hat{r}} x_i} > \alpha_j + \frac{\sum_{i=1}^m a_{ij} f_{i\xi} x_i}{\sum_{i=1}^m f_{i\xi} x_i}, \ \ \forall \hat{r}, \xi \in \{1, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}} \tag{18b}$$

$$x_i \in \{0, 1\}, \ \ \forall i \in \{1, \ldots, m\}, \tag{18c}$$

β-CB:

$$\max_{x} \quad \sum_{i=1}^{m} x_i \tag{19a}$$

$$\text{subject to} \quad \frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^{m} f_{i\hat{r}} x_i} > \beta_j \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i}{\sum_{i=1}^{m} f_{i\xi} x_i}, \quad \forall \hat{r}, \xi \in \{1, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}} \tag{19b}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, m\}. \tag{19c}$$

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 α-CB and β-CB problems are similar to the original CB problem but the aim is to select features that can be used to construct α-consistent and β-consistent biclusterings, respectively.

In (17), 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 (17b) ensures that the biclustering is consistent with respect to the selected features.

Theorem 3.2. *Feature selection for consistent biclustering (i.e. formulation (17)) is $\mathcal{NP}$-hard.*

See [26] for the proof of Theorem 3.2.

Corollary 3.1. *Formulations (18) and (19) are $\mathcal{NP}$-hard.*

Proof. Problem (17) is a special class of Problem (18) when $\alpha_j = 0$ for $j \in S_{\hat{r}}$. Similarly Problem (17) is a special class of Problem (19) when $\beta_j = 1$ for all $j \in S_{\hat{r}}$. Hence both (18) and (19) are $\mathcal{NP}$-hard. □

4. Other Classification Methods

Recently, Bertsimas and Shioda 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) [5]. 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 [1]. 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.

When there exist several classifiers, the problem of evaluation of classifiers' conclusions arises. In [15], the principal expert method (the PE-method) is described to resolve this conflict for the case of supervised classification. Another supervised learning method is by Mammadov *et al.* [33] where a multi-label classifier is considered. See [29, 30, 38, 41] for surveys on classification and disease prediction methods that use mathematical programming techniques.

[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.

5. Concluding Remarks

In this paper, we summarize some of the recent studies on classification that utilize mathematical programming techniques. This review is not exhaustive in that, we explore some techniques in depth and give references for other studies. Applications of optimization already improve quality of some applications but there are still many open problems in theoretical computer science. Mathematical programming techniques will certainly continue to provide ongoing revelations in the constantly growing field of pattern recognition and machine learning.

References

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

[2] Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M. and Yakhini, Z. Tissue classification with gene expression profiles. In *RECOMB '00: Proceedings of the fourth annual international conference on Computational molecular biology*, pages 54–64, New York, NY, USA, 2000. ACM Press.

[3] Ben-Dor, A., Friedman, N. and Yakhini, Z. Class discovery in gene expression data. In *RECOMB '01: Proceedings of the fifth annual international conference on Computational biology*, pages 31–38, New York, NY, USA, 2001. ACM Press.

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

[5] Bertsimas, D. and Shioda, R. Classification and regression via integer optimization. *Operations Research*, 55(2):252–271, 2007.

[6] Christopher,M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2006.

[7] Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugne, C., Furey,T., Ares, M. and Haussler, D. Knowledge-base analysis of microarray gene expressiondata by using support vector machines. *PNAS*, 97(1):262–267, 2000.

[8] Bryan, K. Biclustering of expression data using simulated annealing. In *CBMS '05: Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, pages 383–388, Washington, DC, USA, 2005. IEEE Computer Society.

[9] Busygin, S., Boyko, N., Pardalos, P.M., Bewernitz, M. and Ghacibeh, G. Biclustering EEG data from epileptic patients treated with vagus nerve stimulation. In Onur Seref, O. Erhun Kundakcioglu, and Panos M. Pardalos, editors, *Data mining, systems analysis and optimization in biomedicine*, volume 953, pages 220–231. American Institute of Physics, 2007.

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

[11] Busygin, S., Prokopyev, O. A. and Pardalos, P.M. Feature selection for consistent biclustering. *Journal of Combinatorial Optimization*, 10:7–21, 2005.

[12] Cheng Y. and Church G.M. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 93–103. AAAI Press, 2000.

[13] Cifarelli, C. and Patrizi, G. Solving large protein folding problem by a linear complementarity algorithm with 0-1 variables. *Optimization Methods and Software*, 22(1):25–49, 2007.

[14] Cristianini, N. and Shawe-Taylor, J. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

[15] Demyanova, V.V. The principal expert method in data mining. *Applied and Computational Mathematics*, 4(1):70–74, 2005.

[16] Dhillon, I.S. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274, New York, NY, USA, 2001. ACM Press.

[17] Dhillon, I.S., Mallela, S. and Modha, D.S. Information-theoretic co-clustering. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98, New York, NY, USA, 2003. ACM Press.

[18] Garcia, G.N., Ebrahimi, T. and Vesin, J.M. Joint time-frequency-space classification of eeg in a brain-computer interface application. *Journal on Applied Signal Processing*, pages 713–729, 2003.

[19] Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P. Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. and Lander, E.S. Molecular classification of cancer: Class discovery and class prodiction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[20] Hartigan, J.A. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[21] Hsu, C.W., Chang, C.C. and Lin, C.J. A practical guide to support vector classification. http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf, 2004.

[22] Huang, Z., Chen, H., Hsu, C.J., Chenb, W.H. and Wuc, S. Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision Support Systems*, 37:543–558, 2004.

[23] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of the European Conference on Machine Learning*, pages 137–142, Berlin, 1998. Springer.

[24] Joachims, T. Making large–scale SVM learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 169–184, Cambridge, MA, 1999. MIT Press.

[25] Kluger, Y., Basri, R., Chang, J.T. and Gerstein, M. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res*, 13(4):703–716, April 2003.

[26] Kundakcioglu, O.E. and Pardalos, P.M. *The complexity of feature selection for consistent biclustering*, pages 257–266. World Scientific, 2009.

[27] Kundakcioglu, O.E., Seref, O. and Pardalos, P.M. Multiple instance learning via margin maximization. *submitted*, 2009.

[28] Lal,T.N., Schroeder, M., Hinterberger, T., Weston, J., Bogdan, M., Birbaumer, N. and Sch, B.ölkopf. Support vector channel selection in bci. *IEEE Transactions on Biomedical Engineering*, 51(6):1003–1010, 2004.

[29] Lee, E.K. *Optimization in Medicine*, chapter Optimization-based predictive models in medicine and biology, pages 127–151. Springer, 2008.

[30] Lee, E.K. and Wu, T.L. *Data Mining, Systems Analysis And Optimization In Biomedicine*, chapter Classification and disease prediction via mathematical programming, pages 1–42. American Institute of Physics, 2007.

[31] Lee, S. and Verri, A. Pattern recognition with support vector machines. In *SVM 2002*, Niagara Falls, Canada, 2002. Springer.

[32] Madeira, S. and Oliveira, A. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[33] Mammadov, M., Rubinov, A. and Yearwood, J. *Data Mining in Biomedicine*, volume 7 of *Optimization and Its Applications*, chapter An optimization approach to identify the relationship between features and output of a multi-label classifier, pages 141–167. Springer, 2007.

[34] Mangasarian, O.L. and Wild, E.W. Multisurface proximal support vector classification via generalized eigenvalues. Technical Report 04-03, Data Mining Institute, September 2004.

[35] Nahapetyan, A., Busygin, S. and Pardalos, P.M. An improved heuristic for consistent biclustering problems. In *Mathematical Modelling of Biosystems*, Applied Optimization, pages 185–198. Springer, 2008.

[36] Noble, W.S. *Kernel Methods in Computational Biology*, chapter Support vector machine applications in computational biology, pages 71–92. MIT Press, 2004.

[37] Osuna, R.F.E. and Girosi, F. An improved training algorithm for support vector machines. In *IEEE Workshop on Neural Networks for Signal Processing*, pages 276–285, 1997.

[38] Pardalos, P.M. and Chinchuluun, A. Some recent developments in deterministic global optimization. *Applied and Computational Mathematics*, 5(1):16–34, 2006.

[39] Pardalos, P.M. and Romeijn, E. editors. *Handbook of Optimization in Medicine*. Springer, 2009.

[40] Platt, J. *Advances in Kernel Methods: Support Vector Learning*, chapter Fast training of SVMs using sequential minimal optimization, pages 185–208. MIT press, Cambridge, MA, 1999.

[41] Rubinov, A.M. Methods for global optimization of nonsmooth functions with applications (survey). *Applied and Computational Mathematics*, 5(1):3–15, 2006.

[42] Sch, B. ölkopf and Smola, A. J. *Learning with Kernels*. MIT Press, Cambridge, 2002.

[43] Seref, O., Kundakcioglu, O.E. and Pardalos, P.M. Selective linear and nonlinear classification. In P. M. Pardalos and P. Hansen, editors, *CRM Proceedings and Lecture Notes*, volume 45, pages 211–234, 2008.

[44] Seref, O., Kundakcioglu, O.E., Prokopyev, O.A. and Pardalos, P.M. Selective support vector machines. *Journal of Combinatorial Optimization*, 17(1):3–20, 2009.
[45] Shawe-Taylor, J. and Cristianini, N. *Kernel methods for pattern analysis*. Cambridge University Press, Cambridge, UK, 2004.
[46] Sheng, Q., Moreau, Y. and DeMoor, B. Biclustering microarray data by Gibbs sampling. *Bioinformatics*, 19:196–205, 2003.
[47] Trafalis, T.B. and Ince, H. Support vector machine for regression and applications to financial forecasting. In *International Joint Conference on Neural Networks (IJCNN'02)*, Como, Italy, 2002. IEEE-INNS-ENNS.
[48] Vapnik, V. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
[49] Vapnik, V. *Statistical Learning Theory*. Wiley, New York, 1998.
[50] Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. and Vapnik, V. Feature selection for SVMs. In *NIPS*, pages 668–674, 2000.
[51] Xing, E.P.and Karp, R.M. CLIFF: Clustering of high-dimensional microarray data via iterative feature filtering using normilized cuts. *Bioinformatics Discovery Note*, 17:306–315, 2001.

Panos M. Pardalos - obtained a Ph.D. degree from the University of Minnesota in Computer and Information Sciences. He has held visiting appointments at Princeton University, DIMACS Center, Institute of Mathematics and Applications, FIELDS Institute, AT & T Labs Research, Trier University, Linkoping Institute of Technology, and Universities in Greece.

He has received numerous awards including, University of Florida Research Foundation Professor, UF Doctoral Dissertation Advisor/Mentoring Award, Foreign Member of the Royal Academy of Doctors (Spain), Foreign Member Lithuanian Academy of Sciences, Foreign Member of the Ukrainian Academy of Sciences, Foreign Member of the Petrovskaya Academy of Sciences and Arts (Russia), and Honorary Member of the Mongolian Academy of Sciences.

Dr.Pardalos is the editor-in-chief of the Journal of Global Optimization, Journal of Optimization Letters, and Computational Management Science. He is the author of 8 books and the editor of several books. His research is supported by National Science Foundation and other government organizations. His recent research interests include network design problems, optimization in telecommunications, e-commerce, data mining, biomedical applications, and massive computing.

O. Erhun Kundakcioglu - received the B.S. degree from Bilkent University, Ankara, Turkey, in 2002 and the M.S. degree from Sabancı University, Istanbul, Turkey, in 2004, both in Industrial Engineering. He is a Ph.D. student in the Department of Industrial and Systems Engineering at the University of Florida. His primary research interests are in the areas of combinatorial optimization, nonlinear optimization, pattern recognition, and machine learning.