PATTERN SEPARATION AND PREDICTION VIA LINEAR AND SEMIDEFINITE PROGRAMMING

Xing Liu and Florian A. Potra
Department of Mathematics & Statistics
University of Maryland Baltimore County
Baltimore, Maryland 21250, USA

Revised Jan., 2004

Abstract: We present several optimization methods for separating two sets of points in the n-dimensional space that have nondisjoint convex closures. We give five methods based on linear programming techniques and two methods based on semidefinite programming techniques. For predictive purposes, we construct two parallel hyperplanes using linear programming or two similar concentric ellipsoids using semidefinite programming, so that the intersection of the convex hulls of the two sets is contained between the two hyperplanes or the two ellipsoids, and the rest of the two sets are separated completely. We then construct another hyperplane or ellipsoid between the old ones using linear search for the purpose of pattern separation. We illustrate our methods on two breast cancer databases.

Key words: pattern separation, data mining, linear programming, SemiDefinite programming

1 Introduction

Pattern separation, also known as pattern recognition, is one of the main machine learning problems, originally developed by Vapnik and co-workers[15, 16]. It is also one of the fundamental problems in data mining[2, 3, 7]. The support vector machine approach and sophisticated optimization techniques for solving this problem has been proved to be very efficient as shown by the pioneer work of O. L. Mangasarian among some others[2, 4, 7, 8, 9, 10]. Along the process of machine learning, we need to consider the set separation problem with each set contains only one pattern, which is assigned by the supervisor. It is known that the multi-set separation problem can be tackled by solving a sequence of two-set separation problems[13], so that in this paper, we limit our attention to the two-set separation problem, which can be stated as follows: Suppose we

have two sets of points $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(k)}\}$ and $Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(l)}\}$ in the n-dimensional real space R^n. We want to find a partition of R^n so that all the points in X are separated from the points in Y, if they can be separated strictly; or find an optimized partition in some sense if they cannot be separated completely. Our numerical experiments have been performed on two breast cancer data sets. In those examples, X represents patients with malignant tumors, and Y represents patients with benign tumors. The coordinates of the vectors in X and Y represent different measurements. For example in the data set [1, 10, 11, 17, 18] considered in section 5 of the present paper, we have 9 such attributes: Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, and Mitosis, which makes our problem a two-set separation problem in a 9-dimensional real space.

An elegant approach to separating X and Y is by trying to find a hyperplane in R^n such that all vectors in X lie on one side of this hyperplane and all the vectors in Y lie on the opposite side. A separating hyperplane is defined by a pair (a, γ) where a is an n-dimensional vector and γ is a scalar such that

$$a^T x < \gamma \quad for \quad all \quad x \in X \tag{1}$$

$$a^T y > \gamma \quad for \quad all \quad y \in Y \tag{2}$$

If the convex hulls of X and Y are disjoint, such a separating hyperplane always exists. Finding the vector a and the scalar γ may be important for predictive purposes. Relations (1) and (2) say that a certain linear combination of the coordinates of all vectors in X is below a certain threshold while the same linear combination of the coordinates of the vectors in Y is above this threshold. For the first breast cancer example this would imply that a certain linear combination of the 9 attributes can separate malignant from benign tumors.

The coefficients of the linear combination (i.e., the coordinates of the vector a) may reveal important facts for cancer research. Once a and γ are obtained from the existing data base, they can be used for predictive purposes: after determining the attributes z of a new patient we may deduce that the patient is likely to have a malignant tumor if $a^T z < \gamma$ and a benign tumor if $a^T z > \gamma$.

We have to mention that conditions (1) and (2) do not determine the separating hyperplane (a, γ) in a unique way. An optimal way of determining a and γ suggested in previous learning theories[4, 7, 15] is to determine a and γ as the solution of the following minimization problem

$$\min_{a,\gamma} \quad \|a\| \tag{3}$$

$$\text{s.t.} \quad a^T x \leq \gamma - 1 \quad for \quad all \quad x \in X \tag{4}$$

$$a^T y \geq \gamma + 1 \quad for \quad all \quad y \in Y \tag{5}$$

The solution of this optimization problem, which is known as the maximal margin classifier [15, 4], provides the strongest possible separation in a certain sense. It maximizes the distance in the dual norm $\|.\|'$ between the hyperplanes $(a, \gamma - 1)$ and $(a, \gamma + 1)$, which is the geometric margin[4]. If we choose $\|.\|$ to be the l_1-norm (and therefore $\|.\|'$ the l_∞-norm) the exact solution of (3) can be obtained by solving 2n linear programs.

Unfortunately, in most practical applications the convex hulls of X and Y intersect so that no separating hyperplane exists. Generalization of the above approach were made in order to deal with this situation[9, 15]. A hyperplane (a, γ) is determined in such a way that most points of X satisfy (2), most points of Y satisfy (1), and the distance of the misclassified points (i.e., the points of X that do not satisfy (1) and the points of Y that do not satisfy (2)) to the hyperplane is minimized. A similar approach is used to dealing with the formulation (3)-(5) by adding to the objective function in (3) a multiple of the distance of the misclassified points (i.e., the points of X that do not satisfy (4) and the points of Y that do not satisfy (5)) to the corresponding hyperplane).

In the present paper, rather than directly trying to minimize the number of the misclassified points to a single hyperplane, we propose to separate the sets X and Y by using two parallel hyperplanes H_1 and H_2 with the following properties:

(P1) All points of X lie on one side of H_1.

(P2) All points of Y lie on the opposite side of H_2.

(P3) The intersection of the convex hulls of X and Y is contained in region C between the hyperplanes H_1 and H_2.

The points in C are called unclassified points. We would like to determine the hyperplanes H_1 and H_2 such that the number of unclassified points is small. Mangasarian[10] and Falk[5] proposed similar approaches for the two-set separation problem with the hyperplanes have the same partition properties. However in the present paper, we consider this construction of the two-hyperplane-separation only used for predictive purposes: after taking measurement z of a new patient, we may draw conclusion such as the patient is likely to have a malignant tumor if z is on one side of H_1, or the patient is likely to have a benign tumor if z is on the other side of H_2, or if z is between H_1 and H_2, we would rather conclude that the patient needs further investigation than drawing a stronger conclusion. For the purpose of pattern separation, we go further to construct another hyperplane H_3 between and parallel to H_1 and H_2 such that most of the points of X lie on the same side of H_3 as H_2 does and most of the points of Y lie on the same side of H_3 as H_1 does. If a point fails to do so, it is called a misclassified point. We want to construct H_3 so that the number of misclassified points is minimized.

In section 3 we propose several approaches for obtaining hyperplanes satisfying **(P1)**-**(P3)** and the hyperplane lying between to minimize the number of misclassified points. All our formulations

are solvable by solving at most $2n$ linear programs[14] plus a linear search procedure, where n is the dimension of the underlying space.

In section 4 we investigate the possibility of using ellipsoids instead of hyperplanes for solving the separation problem. More precisely, in the first part of our construction which is used for predictive purposes, we want to determine two similar ellipsoids (in the sense that they have the same center and shape) $\mathcal{E}_1 \subset \mathcal{E}_2$ such that all points of X lie inside $\mathcal{E}_2$ and all points of Y lie outside $\mathcal{E}_1$, or all points of Y lie inside $\mathcal{E}_2$ and all points of X lie outside $\mathcal{E}_1$. In either case, the intersection of the convex hulls of X and Y is contained between $\mathcal{E}_1$ and $\mathcal{E}_2$. The points lying between $\mathcal{E}_1$ and $\mathcal{E}_2$ belong to either X or Y and are called unclassified points. The objective is to determine the ellipsoids $\mathcal{E}_1$ and $\mathcal{E}_2$ such that the number of unclassified points is small. The second part of our construction, which is used for pattern separation, is to find a third ellipsoid with the same shape center lying between the two ellipsoids we found in the first part so that the number of misclassified points is minimized. Francois Glineur in his master thesis[6] had a similar approach as the first part of our construction about the two ellipsoids separation . He constructed the two ellipsoids in the following way:

$$(x-c)^T E(x-c) \leq 1$$

$$(x-c)^T E(x-c) \geq \rho^2.$$

He claimed the problem is not convex and cannot be solved using SQL conic programming. Then he used some elegant transformations to change the it into a $n+1$-dimensional problem using homogeneous ellipsoids and solved the problem. In the present paper, we use a different formulation of the ellipsoids and reformulate the problem in the first part of our construction into a semidefinite problem, which can be solved in polynomial time using interior point methods - that is also the main reason why we chose ellipsoids over other second order surfaces for the separation. Moreover, the availability of reliable interior point methods for semidefinite programming[20] makes this approach feasible for moderate to high dimensional problems. For the second part of our construction, we use linear search technique to achieve the third ellipsoid with the number of misclassified points is minimized.

We note that an ellipsoid $\mathcal{E}$ can be determined by a triple (A, a, γ) where A is a positive semidefinite matrix, a is a vector, and γ is a scalar. A point x is inside $\mathcal{E}$ if

$$x^T A x + a^T x \leq \gamma\,.$$

If A has entries a_{ij} and a has coordinates a_i then the above relation can be written as

$$\sum_{i,j} a_{ij} x_i x_j + \sum_i a_i x_i \leq \gamma\,.$$

If the matrix A is the zero matrix, then the ellipsoid reduces to a hyperplane. Thus separation by ellipsoids is more general. The coefficients a_{ij} and a_i may contain important information about our data base. As mentioned before, we will choose $\mathcal{E}_1$ and $\mathcal{E}_2$ to be similar in the sense that $\mathcal{E}_1 = (A, a, \gamma_1)$ and $\mathcal{E}_2 = (A, a, \gamma_2)$ with $\gamma_1 \leq \gamma_2$. Thus separating by ellipsoids implies that a certain quadratic combination of most vectors in X is less than (or greater than) the same quadratic combination of most vectors in Y.

We should also notice that according to our constructions, the problem is always feasible - The two hyperplanes or ellipsoids for prediction always exist since the worst case scenario is all the points from both X and Y are between the hyperplanes or the ellipsoids. Once we have the two hyperplanes or the ellipsoids, it is obviously that a feasible solution exists between them by running linear search.

We note that using the classical support vector machine approach we could obtain different types of separation surfaces (decision boundaries). However, we feel that not only we need to separate the current data points, it is in fact more important to ask the question what is the meaning of the separation surface for predictive purposes. We choose to work on hyperplane and ellipsoid separations not only because they can be solved efficiently by sophisticated modern optimization techniques, but also we can help doctors and biologists to explain the data - what is the importance of a certain feature for a patient having breast cancer, or how important the interaction between two features is for a patient not likely to have breast cancer? We could get these valuable information by further studying the separation hyperplanes or ellipsoids.

In the last section of this paper we will apply both separation by hyperplanes and ellipsoids to two breast cancer data sets. Results of performing cross-validation for all the formulations will be shown.

2 Separating by hyperplanes via misclassification minimization

2.1 Formulation I

In [9] Mangasarian proposes to find a hyperplane (a, γ) with $\|a\|_\infty = 1$ such that most points of X satisfy (1) and most points of Y satisfy (2) as the solution of the following optimization problem:

$$\min_{a,\gamma,z_x,z_y,p,q} \quad e^T z_x + e^T z_y \tag{6}$$

$$\begin{aligned}
\text{s.t.} \quad & a^T x^{(i)} \leq \gamma + z_{xi} && i = 1, 2, \ldots, k \\
& a^T y^{(j)} \geq \gamma - z_{yj} && j = 1, 2, \ldots, l \\
& z_x \geq 0\,, \quad z_y \geq 0\,, \quad -e \leq a \leq e \\
& a_p = (-1)^q \\
& p \in \{1, 2, \ldots, n\}, \quad q \in \{1, 2\}
\end{aligned}$$

where e is the vector of all ones.

The solution of the above optimization problem can be obtained by solving $2n$ linear programs (one for each choice of $(p,q) \in \{1,2,\ldots,n\} \times \{1,2\}$. From the solution (a, γ, z_x, z_y) of (6) we can obtain two hyperplanes $H_1 = (a, \gamma + \sigma_1)$ and $H_2 = (a, \gamma - \sigma_2)$ satisfying **(P1)-(P3)**, where

$$\sigma_1 = max\{z_{xi} \mid i = 1,2,\ldots,k\}\ , \quad \sigma_2 = max\{z_{yj} \mid j = 1,2,\ldots,l\}\ . \tag{7}$$

We then solve the following minimization problem using linear search.

$$\min_{b} \quad \sum_{i=1}^{k}(a^T x^{(i)} - b)_* + \sum_{j=1}^{l}(b - a^T y^{(j)})_* \tag{8}$$

$$\text{s.t.} \quad \gamma - \sigma_2 \leq b \leq \gamma + \sigma_1,$$

where $()_*$ is the step function: $R \to R$ as:

$$(t)_* = \begin{cases} 1 & if \quad t > 0 \\ 0 & if \quad t \leq 0 \end{cases} \tag{9}$$

By using formulation I, we can get not only a set of solutions (a, b) for the purposes of pattern separation, but also two hyperplanes which can be used for predictive purposes. Moreover, it is guaranteed that the solution (a, b) is at least as good as the solution (a, γ), which is a well known linear classifier [9].

2.2 Formulation II

In [7, 15] one proposes to find a hyperplane (a, γ) such that most points x in X satisfy (4) and most of the points y in Y satisfy (5) by solving the following optimization problem, which attempts to maximize the l_∞-distance in between the hyperplanes $(a, \gamma - 1)$ and $(a, \gamma + 1)$, as well as to minimize the sum of violations of the misclassified points:

$$\min_{a,\gamma,s,z_x,z_y} \quad \nu e^T(z_x + z_y) + e^T s \tag{10}$$

$$\begin{aligned} \text{s.t.} \qquad & a^T x^{(i)} - z_{xi} \leq \gamma - 1, && i = 1,2,\ldots,k \\ & a^T y^{(j)} + z_{yj} \geq \gamma + 1, && j = 1,2,\ldots,l \\ & -s \leq a \leq s\,, \quad z_x \geq 0\,, \quad z_y \geq 0\ . \end{aligned}$$

Here $\nu > 0$ is a given (penalty) parameter.

This formulation is the l_1-norm version of one of the classic support vector machines, known as the maximal margin classifier using soft margin optimization in some literature[4]. Obviously, from the solution of the above optimization problems we can obtain two hyperplanes $H_1 = (a, \gamma - 1 + \sigma_1)$ and $H_2 = (a, \gamma + 1 - \sigma_2)$, with σ_1, σ_2 given by (7), satisfying **(P1)-(P3)**.

We then solve the following minimization problem using linear search.

$$\min_{b} \quad \sum_{i=1}^{k}(a^T x^{(i)} - b)_* + \sum_{j=1}^{l}(b - a^T y^{(j)})_* \tag{11}$$

$$\text{s.t.} \quad \gamma + 1 - \sigma_2 \leq b \leq \gamma - 1 + \sigma_1.$$

3 Direct methods for determining separating hyperplanes

3.1 Formulation III

In this formulation he want to find two hyperplanes $H_1 = (a, \beta)$ and $H_2 = (a, \beta - \tau)$, with $\|a\|_\infty = 1$ and $\tau \geq 0$ as small as possible, so that conditions **(P1)-(P3)** are satisfied. We note that the l_1-distance between the two hyperplanes is equal to τ. The hyperplanes are obtained via the following algorithm:

(i) For each $(p, q) \in \{1, 2, \ldots, n\} \times \{1, 2\}$ *solve the following program:*

$$\min_{a,\beta,\tau} \quad \tau \tag{12}$$

$$\begin{aligned} s.t. \qquad a^T x^{(i)} - \beta &\leq 0 \qquad i = 1, 2, \ldots, k \\ a^T y^{(j)} - \beta + \tau &\geq 0 \qquad j = 1, 2, \ldots, l \\ \tau \geq 0\,, \quad -e \leq a &\leq e \\ a_p &= (-1)^q \end{aligned}$$

(ii) Compare the number of unclassified points for each linear program above.
(iii) Return the solution (a, β, τ) *from the linear program that gives the minimum number of unclassified points.*

We then solve the following minimization problem using linear search.

$$\min_{b} \quad \sum_{i=1}^{k}(a^T x^{(i)} - b)_* + \sum_{j=1}^{l}(b - a^T y^{(j)})_* \tag{13}$$

$$\text{s.t.} \quad \beta - \tau \leq b \leq \beta.$$

3.2 Formulation IV

As in the above formulation we want to determine two hyperplanes $H_1 = (a, \beta)$ and $H_2 = (a, \beta - \tau)$, with $\tau \geq 0$ as small as possible, so that conditions **(P1)-(P3)** are satisfied. However, instead of normalizing $\|a\|_\infty = 1$, we just take $\beta = -1$ and obtain a and τ via the following algorithm:

(i) Solve the following two linear programming problems:

$$\min_{a,\tau} \quad \tau \tag{14}$$

$$\begin{aligned} s.t. \quad & a^T x^{(i)} + 1 \leq 0 \quad && i = 1, 2, \ldots, k \\ & a^T y^{(j)} + 1 + \tau \geq 0 && j = 1, 2, \ldots, l \\ & \tau \geq 0 \end{aligned}$$

$$\min_{a,\tau} \quad \tau \tag{15}$$

$$\begin{aligned} s.t. \quad & a^T y^{(j)} + 1 \leq 0 \quad && j = 1, 2, \ldots, l \\ & a^T x^{(i)} + 1 + \tau \geq 0 && i = 1, 2, \ldots, k \\ & \tau \geq 0 \end{aligned}$$

(ii) Compare the number of unclassified points for each linear program above.
(iii) Return the solution (a, τ) from the linear program that gives the minimum number of unclassified points.

Depending on whether the optimal solution (a, τ) comes from (14) or (15), we solve one of the following minimization problems correspondingly using linear search.

$$\min_{b} \quad \sum_{i=1}^{k} (a^T x^{(i)} - b)_* + \sum_{j=1}^{l} (b - a^T y^{(j)})_* \tag{16}$$

$$\text{s.t.} \quad -1 - \gamma \leq b \leq -1;$$

and

$$\min_{b} \quad \sum_{i=1}^{k} (b - a^T x^{(i)})_* + \sum_{j=1}^{l} (a^T y^{(j)} - b)_* \tag{17}$$

$$\text{s.t.} \quad -1 - \gamma \leq b \leq -1.$$

3.3 Formulation V

In this formulation we determine two hyperplanes of the form $H_1 = (a, \gamma - 1)$ and $H_2 = (a, \gamma + 1)$ satisfying **(P1)-(P3)** and such that the l_1-distance between them is minimized. This is accomplished by the following algorithm:
(i) For each $p = 1, 2, \ldots, n$, solve the following two linear programs:

$$\max_{a,\gamma} \quad a_p \tag{18}$$

$$\begin{aligned} s.t. \quad & a^T x^{(i)} \leq \gamma + 1 \quad && i = 1, 2, \ldots, k \\ & a^T y^{(j)} \geq \gamma - 1 && j = 1, 2, \ldots, l \\ & -a_p \leq a_q \leq a_p && q = 1, 2, \ldots, n \end{aligned}$$

$$\max_{a,\gamma} \quad -a_p \tag{19}$$

$$\begin{aligned} s.t. \quad & a^T x^{(i)} \leq \gamma + 1 \quad i = 1, 2, \ldots, k \\ & a^T y^{(j)} \geq \gamma - 1 \quad j = 1, 2, \ldots, l \\ & a_p \leq a_q \leq -a_p \quad q = 1, 2, \ldots, n \end{aligned}$$

(ii) Compare the number of unclassified points for each linear program above.
(iii) Return the solution (a, γ) from the linear program that gives the minimum number of unclassified points.

We then solve the following minimization problem using linear search.

$$\min_{b} \quad \sum_{i=1}^{k} (a^T x^{(i)} - b)_* + \sum_{j=1}^{l} (b - a^T y^{(j)})_* \tag{20}$$

$$\text{s.t.} \quad \gamma - 1 \leq b \leq \gamma + 1;$$

4 Separation by ellipsoids

4.1 Formulation VI

We look for separating ellipsoids of the form $\mathcal{E}_1 = (A, a, \beta - \tau)$ and $\mathcal{E}_2 = (A, a, \beta)$, where A is a positive semidefinite matrix (which we denote by $A \succeq 0$) of unit trace, and $\tau \geq 0$ is as small as possible. The ellipsoids are obtained via the following algorithm:

(i) Solve the following two semidefinite programming problems:

$$\min_{A,a,\beta,\tau} \quad \tau \tag{21}$$

$$\begin{aligned} s.t. \quad & (x^{(i)})^T A x^{(i)} + a^T x^{(i)} - \beta \leq 0 \quad i = 1, 2, \ldots, k \\ & (y^{(j)})^T A y^{(j)} + a^T y^{(j)} - \beta + \tau \geq 0 \quad j = 1, 2, \ldots, l \\ & trace(A) = 1 \\ & A \succeq 0 \\ & \tau \geq 0 \end{aligned}$$

$$\min_{A,a,\beta,\tau} \quad \tau \tag{22}$$

$$\begin{aligned} s.t. \quad & (y^{(j)})^T A y^{(j)} + a^T y^{(j)} - \beta \leq 0 \quad j = 1, 2, \ldots, l \\ & (x^{(i)})^T A x^{(i)} + a^T x^{(i)} - \beta + \tau \geq 0 \quad i = 1, 2, \ldots, k \\ & trace(A) = 1 \\ & A \succeq 0 \\ & \tau \geq 0 \end{aligned}$$

(ii) Compare the number of unclassified points for the above two problems.
(iii) Return the solution (A, a, β, τ) *from the problem that gives the minimum number of unclassified points.*
Here by $trace(A)$ we denote the sum of the diagonal entries of the matrix A.

Depending on whether the optimal solution (A, a, β, τ) comes from (21) or (22), we solve one of the following minimization problems correspondingly using linear search.

$$\min_{b} \quad \sum_{i=1}^{k}((x^{(i)})^T Ax^{(i)} + a^T x^{(i)} - b)_* + \sum_{j=1}^{l}(b - (y^{(j)})^T Ay^{(j)} - a^T y^{(j)})_* \tag{23}$$

$$\text{s.t.} \quad \beta - \tau \leq b \leq \beta;$$

and

$$\min_{b} \quad \sum_{i=1}^{k}(b - (x^{(i)})^T Ax^{(i)} - a^T x^{(i)})_* + \sum_{j=1}^{l}((y^{(j)})^T Ay^{(j)} + a^T y^{(j)} - b)_* \tag{24}$$

$$\text{s.t.} \quad \beta - \tau \leq b \leq \beta.$$

4.2 Formulation VII

In this formulation, we use the previous construction except that this time we normalize β. By taking $\beta = 1$ and $\beta = -1$, we have the following algorithm:

(i) Solve the following four semidefinite programming problems:

$$\min_{A,a,\tau} \quad \tau \tag{25}$$

$$\begin{aligned} s.t. \quad & (x^{(i)})^T Ax^{(i)} + a^T x^{(i)} + 1 \leq 0 \qquad i = 1, 2, \ldots, k \\ & (y^{(j)})^T Ay^{(j)} + a^T y^{(j)} + 1 + \tau \geq 0 \quad j = 1, 2, \ldots, l \\ & A \succeq 0 \\ & \tau \geq 0 \end{aligned}$$

$$\min_{A,a,\tau} \quad \tau \tag{26}$$

$$\begin{aligned} s.t. \quad & (x^{(i)})^T Ax^{(i)} + a^T x^{(i)} - 1 \leq 0 \qquad i = 1, 2, \ldots, k \\ & (y^{(j)})^T Ay^{(j)} + a^T y^{(j)} - 1 + \tau \geq 0 \quad j = 1, 2, \ldots, l \\ & A \succeq 0 \\ & \tau \geq 0 \end{aligned}$$

$$\min_{A,a,\tau} \quad \tau \tag{27}$$

$$
\begin{aligned}
s.t. \quad & (y^{(j)})^T A y^{(j)} + a^T y^{(j)} + 1 \le 0 \qquad j = 1, 2, \ldots, l \\
& (x^{(i)})^T A x^{(i)} + a^T x^{(i)} + 1 + \tau \ge 0 \quad i = 1, 2, \ldots, k \\
& A \succeq 0 \\
& \tau \ge 0
\end{aligned}
$$

$$
\min_{A,a,\tau} \quad \tau \tag{28}
$$

$$
\begin{aligned}
s.t. \quad & (y^{(j)})^T A y^{(j)} + a^T y^{(j)} - 1 \le 0 \qquad j = 1, 2, \ldots, l \\
& (x^{(i)})^T A x^{(i)} + a^T x^{(i)} - 1 + \tau \ge 0 \quad i = 1, 2, \ldots, k \\
& A \succeq 0 \\
& \tau \ge 0
\end{aligned}
$$

(ii) Compare the number of unclassified points for the above four problems.
(iii) Return the solution (A, a, τ) from the problem that gives the minimum number of unclassified points.

Depending on whether the optimal solution (A, a, τ) comes from (25), or (26), or (27), or (28), we solve one of the following minimization problems correspondingly using linear search.

$$
\min_b \quad \sum_{i=1}^{k} ((x^{(i)})^T A x^{(i)} + a^T x^{(i)} - b)_* + \sum_{j=1}^{l} (b - (y^{(j)})^T A y^{(j)} - a^T y^{(j)})_* \tag{29}
$$

$$
\text{s.t.} \quad -1 - \tau \le b \le -1;
$$

$$
\min_b \quad \sum_{i=1}^{k} ((x^{(i)})^T A x^{(i)} + a^T x^{(i)} - b)_* + \sum_{j=1}^{l} (b - (y^{(j)})^T A y^{(j)} - a^T y^{(j)})_* \tag{30}
$$

$$
\text{s.t.} \quad 1 - \tau \le b \le 1;
$$

$$
\min_b \quad \sum_{i=1}^{k} (b - (x^{(i)})^T A x^{(i)} - a^T x^{(i)})_* + \sum_{j=1}^{l} ((y^{(j)})^T A y^{(j)} + a^T y^{(j)} - b)_* \tag{31}
$$

$$
\text{s.t.} \quad -1 - \tau \le b \le -1;
$$

$$
\min_b \quad \sum_{i=1}^{k} (b - (x^{(i)})^T A x^{(i)} - a^T x^{(i)})_* + \sum_{j=1}^{l} ((y^{(j)})^T A y^{(j)} + a^T y^{(j)} - b)_* \tag{32}
$$

$$
\text{s.t.} \quad 1 - \tau \le b \le 1.
$$

	pct. error	Old solution (a, γ) in formulation I	Our solution (a, b) in formulation I
20% training data	error from training data	0.11%	0.11%
	error from test data	4.19%	4.19%
50% training data	error from training data	1.00%	1.00%
	error from test data	2.32%	2.32%
100% training data	error from training data	2.86%	2.72%
	error from test data	0	0

Table 1: Comparison of the solutions of formulation I using data set 1 - for the purposes of pattern separation

5 Numerical Results for the Breast Cancer Data Set

In this section we apply the seven formulations for the separation problem on the following breast cancer data sets:

Data set 1 is from the Wisconsin Breast Cancer Database, collected in 1991 by Dr. William H. Wolberg, University of Wisconsin, Madison[1, 10, 11, 17, 18]. Samples were collected periodically as Dr. Wolberg reported his clinical cases. The database therefore reflects this chronological grouping of the data. The samples consist of visually assessed nuclear features of fine needle aspirates (FNAs) taken from patients' breasts. There are 699 data points in total, including 458 benign points and 241 malignant points. Malignancy is determined by taking a sample tissue from the patient's breast and performing a biopsy on it. A benign diagnosis is confirmed either by biopsy or by periodic examination, depending on the patient's choice. Each single patient was assigned to a 9-dimensional vector, where the components of the vector are the corresponding 9 attributes of the patients.

Data set 2 is the Wisconsin Diagnostic Breast Cancer (WDBC) data set created in 1995 by Dr. William H. Wolberg, and Olvi L. Mangasarian[19]. There are 569 data points in this database, including 357 benign points and 212 malignant points. Each patient was assigned to a 30-dimensional vector. All the features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.

We used a PC Inter celeron 1.0 GHz processor with 512 MB syncDRAM for the numerical experiments. All the codes use the MATLAB environment and the SeDuMi solver[12] for both linear programming and semidefinite programming. In Formulation II, we took ν equally distributed in log space from 10^{-3} to 10^{+4} and selected the best result from 100 values.

	pct. error	Old solution (a, γ) in formulation II	Our solution (a, b) in formulation II
20% training data	error from training data	0.70%	0.39%
	error from test data	7.57%	3.76%
50% training data	error from training data	2.22%	0.97%
	error from test data	4.72%	2.15%
100% training data	error from training data	6.27%	2.72%
	error from test data	0	0

Table 2: Comparison of the solutions of formulation II using data set1 - for the purposes of pattern separation

Table 1 compares the old solutions (a, γ) and the new solutions (a, b) from formulation I. As we mentioned in early chapters, our solution in formulation I is guaranteed at least as good as the linear-classifier in [9], which is shown from Table 1, although there is only slight difference between the two.

Table 2 compares the old solutions (a, γ) and the new solutions (a, b) from formulation II. Although we don't have any proof so far to guarantee our solution is better, (i.e. b is always between $\gamma + 1 - \sigma_2$ and $\gamma - 1 + \sigma_1$), we observe a much better performance from our solution, compare to the old solution (a, γ), which is the l_1-norm version of the maximal margin classifier using soft margin optimization [4].

The comparisons in Table 1 and Table 2 are only for the purposes of pattern separation, in Table 3, we show for predictive purposes the percentage error of prediction and the percentage of unclassified points; and for the purposes of pattern separation the percentage of misclassified points during the learning process.

We compare the results of all the hyperplane separation, that is, formulation I to V. They are almost equally good while formulation III seems to be slightly better than the rest of the formulations. It is very stable even if only 20% of the data is used for training. Moreover if we use 100% data for training , which would not make sense for cross-validation test, but only for the sake of comparison, it gives the best separation result for predictive purposes. We notice the error of prediction will decrease if we increase the portion of the training data, however we cannot observe an obvious change regarding to the percentage of misclassified points. Sometimes the percentage is higher if we use more training data. This shows that for the purposes of pattern separation, 20% training data is good enough for our formulations, we may have overfitting problems if we use too much training data and cause a larger percentage of misclassified points.

Formulations I - VII	20% training data set 1	50% training data set 1	20% training data set 2	50% training data set 2
pct. of unclassified points	6.85	18.06	0	0
pct. error of prediction	3.76	1.00	5.22	1.89
pct. of misclassified points (sum of training and test data)	4.30	3.32	5.22	1.89
pct. of unclassified points	6.14	11.62	0	0
pct. error of prediction	3.33	0.75	5.55	3.16
pct. of misclassified points (sum of training and test data)	4.15	3.12	5.55	3.16
pct. of unclassified points	3.19	10.94	0	0
pct. error of prediction	4.43	0.82	4.52	2.55
pct. of misclassified points (sum of training and test data)	5.52	4.04	4.52	2.55
pct. of unclassified points	2.00	13.45	0	0
pct. error of prediction	4.23	1.68	5.71	2.64
pct. of misclassified points (sum of training and test data)	5.06	5.19	5.71	2.64
pct. of unclassified points	10.19	10.23	9.28	8.39
pct. error of prediction	4.09	1.65	4.89	2.46
pct. of misclassified points (sum of training and test data)	6.37	5.97	8.89	6.28
pct. of unclassified points	0	0	0	0
pct. error of prediction	5.97	3.54	6.92	2.07
pct. of misclassified points (sum of training and test data)	5.97	3.54	6.92	2.07
pct. of unclassified points	0	0	0	0
pct. error of prediction	4.96	3.76	8.15	2.90
pct. of misclassified points (sum of training and test data)	4.96	3.76	8.15	2.90

Table 3: Separation Results

We compare the results of the ellipsoids separation (formulation VI and VII) and we could not see much difference. Formulation VI is slightly better than formulation VII. For the the purposes of pattern separation, both do not have obvious advantages over formulations I to V. But for the predictive purposes, both are very good if we compare the percentage error of prediction and the percentage of unclassified points. As we discussed in the previous section, separation by ellipsoids is a generalized situation of separation by hyperplanes, so that we should expect better separation results for predictive purposes. We don't really mean to compare the results of separation by hyperplanes and ellipsoids, we just want to put into perspective all the results here and prepare them for the future study.

References

[1] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. In *Optimization Methods and Software 1*, pages 23–34. Gordon & Breach Science Publishers, 1992.

[2] P. S. Bradley, Usama M. Fayyad, and O. L. Mangasarian. Mathematical programming for data mining: formulations and challenges. *INFORMS J. Comput.*, 11(3):217–238, 1999.

[3] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Feature selection via mathematical programming. *INFORMS J. Comput.*, 10(2):209–217, 1998.

[4] Nello Cristianini and John Shawe-Taylor. *Support Vector Machines*. Cambridge University Press, Cambridge, United Kingtom, 2000.

[5] James E. Falk and Emma Lopez-Cardona. The surgical separation of sets. *INFORMS J. Comput.*, 11:433–462, 1997.

[6] Francois Glineur. Pattern separation via ellipsoids and conic programming. 1998.

[7] Y.-J. Lee, O. L. Mangasarian, and W. H. Wolberg. Breast cancer survival and chemotherapy: a support vector machine analysis. In *Discrete mathematical problems with medical applications (New Brunswick, NJ, 1999)*, pages 1–10. Amer. Math. Soc., Providence, RI, 2000.

[8] O. L. Mangasarian. Misclassification minimization. *J. Global Optim.*, 5(4):309–323, 1994.

[9] O. L. Mangasarian. Arbitrary-norm separating plane. *Oper. Res. Lett.*, 24(1-2):15–23, 1999.

[10] O. L. Mangasarian, R. Setiono, and W. H. Wolberg. Pattern recognition via linear programming: theory and application to medical diagnosis. In *Large-scale numerical optimization (Ithaca, NY, 1989)*, pages 22–31. SIAM, Philadelphia, PA, 1990.

[11] O. L. Mangasarian and W. H. Wolberg. Cancer diagnosis via linear programming. *SIAM News*, 23(5):1&18, September, 1990.

[12] Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, 11/12(1-4):625–653, 1999. Interior point methods.

[13] J. Ullman. *Pattern Recognition Techniques*. Crane, London, 1973.

[14] Robert J. Vanderbei. *Linear programming: foundations and extensions*, volume 4 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, Boston, MA, 1996.

[15] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, New York, 1995.

[16] Vladimir N. Vapnik. *Statistical learning theory*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons Inc., New York, 1998. A Wiley-Interscience Publication.

[17] W. H. Wolberg and O. L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In *Proceedings of the National Academy of Sciences, U.S.A.*, volume 87, pages 9193–9196. December, 1990.

[18] William H. Wolberg. Wisconsin breast cancer database. 1991.

[19] William H. Wolberg and Olvi L. Mangasarian. Wisconsin diagnostic breast cancer data set. 1995.

[20] Editor Henry Wolkowicz, Editor Romesh Saigal, and Editor Lieven Vandenberghe. *Handbook Of Semidefinite Programming: Theory, Algorithms, And Applications*. Kluwer Academic Publishers, 2000.