Geometry in Learning

Article · October 1996

Source: CiteSeer

CITATIONS
75

READS
406

2 authors:

Kristin P. Bennett
Rensselaer Polytechnic Institute
196 PUBLICATIONS 9,891 CITATIONS
SEE PROFILE

Erin J. Bredensteiner
University of Evansville
6 PUBLICATIONS 876 CITATIONS
SEE PROFILE

Geometry in Learning

Kristin P. Bennett
Erin J. Bredensteiner
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180
bennek@rpi.edu
bredee@rpi.edu

Abstract

One of the fundamental problems in learning is identifying members of two different classes. For example, to diagnose cancer, one must learn to discriminate between benign and malignant tumors. Through examination of tumors with previously determined diagnosis, one learns some function for distinguishing the benign and malignant tumors. Then the acquired knowledge is used to diagnose new tumors. The perceptron is a simple biologically inspired model for this two-class learning problem. The perceptron is trained or constructed using examples from the two classes. Then the perceptron is used to classify new examples. We describe geometrically what a perceptron is capable of learning. Using duality, we develop a framework for investigating different methods of training a perceptron. Depending on how we define the "best" perceptron, different minimization problems are developed for training the perceptron. The effectiveness of these methods is evaluated empirically on four practical applications: breast cancer diagnosis, detection of heart disease, political voting habits, and sonar recognition. This paper does not assume prior knowledge of machine learning or pattern recognition.

1 Introduction

Imagine that your job is to determine whether breast tumors are benign or malignant. A surgeon inserts a needle into the breast tumor and aspirates a small amount of tissue. A microscope slide of the fine needle aspirate is prepared. Your job is to examine the cells on the slide, assess important attributes of the cells such as the uniformity of the cell shape and variability in the cell size, and then determine a diagnosis of benign or malignant. You would learn to do this by examining many tumors that were previously determined to be benign or malignant by an expert pathologist using surgical biopsies. Probably somebody would help you by pointing out which attributes were important. You would "generalize" the knowledge you learned by applying it to diagnosing new tumors.

At the University of Wisconsin-Madison, a computer system has been developed that has "learned" to diagnose breast cancer [35, 32, 34]. The prepared slide of the fine needle aspirate is inserted into a computer imaging system that measures and determines low-level features of the nuclei of the cells within the tumor. The tumor is then described as a vector of real numbers. Each number represents one attribute of the cells. The vector is input into a computer program that produces a suggested diagnosis. The computer program was "trained" by giving it hundreds of examples of tumors that are known to be benign or malignant. During training a mathematical function was developed to classify the given examples as benign or malignant. This function is subsequently used to diagnose new cases. The computer learned in the sense that it generated a classification function based on observing examples with known classification.

In this paper, we examine the underlying problem of constructing a function to discriminate between examples from two classes. Our goal is to create a geometrical investigation of the problem from initial conception to evaluation of the computational results. We do not assume prior knowledge of machine learning or pattern recognition. Using the biology of the brain as an inspiration, we examine a simple mathematical model of learning called the perceptron. Geometrically, we describe what concepts a perceptron can learn. Then geometrical arguments are used to motivate algorithms for training the perceptron. Duality is used to provide different mathematical models of perceptron training. Depending on how we characterize the "best" perceptron, different optimization methods are constructed to train the perceptron. The tradeoffs of the different methods are discussed. An empirical comparison of the methods is performed.

For each classification problem, we are given examples or points from two classes. Each example x is represented by an n-dimensional real vector. Each of the dimensions represents an attribute of the example. In the heart disease problem, each example represents a patient. The attributes of each example include the patient's age, the patient's sex, and the patient's cholesterol level. In the training phase, we are also given the class to which each example belongs. For example, the set $\mathcal{A}$ could correspond to patients with heart disease and set $\mathcal{B}$ could correspond to patients without heart disease. Our problem is to construct, using the two sets of examples, a function $f(x)$ that returns 1 if x belongs to $\mathcal{A}$ and 0 if x belongs to $\mathcal{B}$. In the training phase, the function is constructed using the two sets of sample points, one set from each class. In the testing phase, the function is used to classify future points whose classification is unknown. Currently there is no mathematical definition of the "best" function for any given problem. The goal is to make this function as accurate as possible on future points; i.e., the function should "generalize" well. But we can only guess what the future points will be while the function is being constructed. Many types of classification functions are possible, but in this paper we will restrict ourselves to a linear function called the perceptron [27, 21].

While the perceptron model is quite simple, it works very well on many practical problems including the Wisconsin Breast Cancer Diagnosis problem described above. We present computational results for several methods of training a perceptron for real-world classification problems. Specifically, we will examine the performance of the perceptron on the following problems: breast cancer diagnosis, detection of heart disease, determination of the party affiliation of United States Representatives based on their voting habits, and sonar recognition of mines.

The following notational conventions will be used. For a column vector x in the n-dimensional real space R^n, x_i denotes the i^{th} component of x. The notation $A \in R^{m \times n}$ will signify a real $m \times n$ matrix. For such a matrix, A_i will denote the i^{th} row. The transpose of x and A are denoted x' and A' respectively. The dot product of two vectors x and w will be denoted by $x'w$. A vector of ones in a space of arbitrary dimension will be denoted by e. The scalar 0 and a vector of zeros are both represented by 0. Thus, for $x \in R^m$, $x > 0$ implies that $x_i > 0$ for $i = 1, \ldots, m$. In general, for $x, y \in R^m$, $x > y$ implies that $x_i > y_i$ for $i = 1, \ldots, m$. Similarly, $x \geq y$ implies that $x_i \geq y_i$ for $i = 1, \ldots, m$. Several norms are used. The 1-norm of x, $\sum_{i=1}^{n} |x_i|$, is denoted by $\|x\|_1$. The 2-norm or Euclidean norm of x, $\sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{x'x}$, is denoted by $\|x\|$. Frequently, the 2-norm is squared to make it differentiable: $\|x\|^2 = x'x$. The infinity norm of x, $\max_{i=1 \ldots n} (|x_i|)$, is denoted by $\|x\|_\infty$.

2 A Simple Learning Model

The perceptron model can be motivated biologically. The brain consists of an interconnecting network of about 10^{11} neurons or nerve cells [13]. Each neuron receives stimuli from other cells. A stimulus may be excitatory or inhibitory. If the combined stimuli exceeds some threshold then the cell "fires". In the perceptron, the stimuli are modeled with an n-dimensional real input vector x. Each stimulus x_i has an associated real weight w_i. If the w_i is positive, the stimulus is excitatory. If w_i is negative, the stimulus is inhibitory. If the weighted sum of the stimuli $\sum_{i=1}^{n} w_i x_i = x'w$ is greater than some threshold γ then the perceptron fires and we say that x is in class $\mathcal{A}$. Otherwise we say that x is in class $\mathcal{B}$. Mathematically we define the perceptron as follows:

Definition 2.1 (Perceptron) *Let $x \in R^n$ be a point to be classified. A perceptron with weights $w \in R^n$ and threshold $\gamma \in R$ is defined as*

$$\begin{array}{rcl} x'w - \gamma > 0 & \Rightarrow & x \in \mathcal{A} \\ x'w - \gamma < 0 & \Rightarrow & x \in \mathcal{B} \end{array} \qquad (1)$$

In theory, if $x'w = \gamma$ then the class of the point x is undefined. In practice, we use the convention that if $x'w = \gamma$, then x is in $\mathcal{B}$.

Two questions immediately arise: What exactly can a perceptron learn and how does one train the perceptron, i.e., determine the weights w and threshold γ? The classic book by Minsky and Papert [21] may be consulted for an extensive discussion of both these questions. In our geometric approach, we will answer the first question using basic geometric arguments and then show how the geometry naturally motivates optimization methods for training a perceptron.

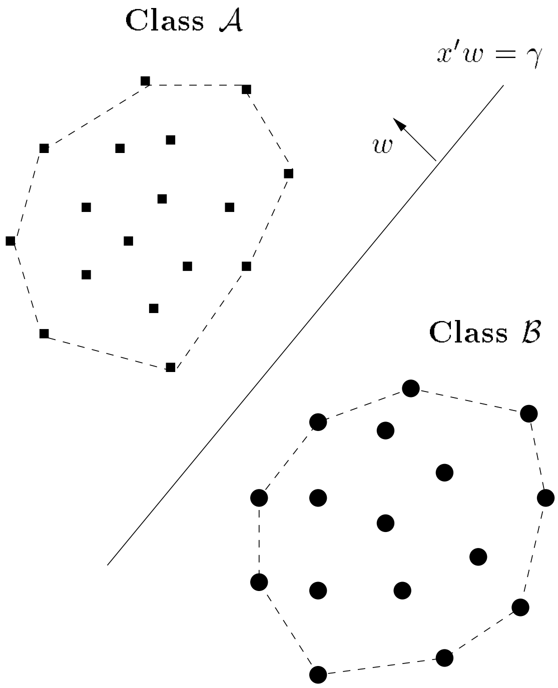

Figure 1: Separating plane for two linearly separable sets

3 Geometry of a Perceptron

Geometrically training a perceptron corresponds to finding a plane that separates the two given example sets $\mathcal{A}$ and $\mathcal{B}$. The perceptron determines the separating plane $x'w = \gamma$, where w is the normal of the plane and $\frac{|\gamma|}{\|w\|}$ is the Euclidean distance of the plane from the origin. Let the coordinates of the points in $\mathcal{A}$ be given by the m rows of the $m \times n$ matrix A. Let the coordinates of the points in $\mathcal{B}$ be given by the k rows of the $k \times n$ matrix B. By definition of the perceptron model (2.1) we know that a perceptron exists that correctly classifies the two finite sets if and only if there exist w and γ satisfying the inequalities

$$\begin{array}{c} Aw > e\gamma \\ Bw < e\gamma \end{array} \tag{2}$$

where e is a vector of ones of appropriate dimension. If such an w and γ exist, we say that the sets are **linearly separable**. Of course, w and γ are the weights and threshold, respectively, of the separating perceptron. An example of two linearly separable sets in R^2 is given in Figure 1. Notice all of the points in $\mathcal{A}$ are in the open half space $\{x \in R^2 | x'w > \gamma\}$ and all the points in $\mathcal{B}$ are in the open half space $\{x \in R^2 | x'w < \gamma\}$.

In Figure 1, the convex hulls of $\mathcal{A}$ and $\mathcal{B}$ are shown as the areas enclosed by the dashed lines. Note that when the sets are separable by a plane, the convex hulls of the two sets do not intersect. Recall that the convex hull of $\mathcal{A}$ is the set of all points that can be written as convex combinations of the points in $\mathcal{A}$. A convex combination c of $\mathcal{A}$ is defined by

$$c' \;=\; u_1 A_1 + u_2 A_2 + \ldots + u_m A_m \;=\; u'A \tag{3}$$

where $u \in R^m$, $u \geq 0$, and $\sum_{i=1}^{m} u_i = e'u = 1$. Similarly, the convex hull of $\mathcal{B}$ is the set of all points that can be written as convex combinations of the points in $\mathcal{B}$. A convex combination d of $\mathcal{B}$ is defined by

$$d' \;=\; v_1 B_1 + v_2 B_2 + \ldots + v_k B_k \;=\; v'B \tag{4}$$

where $v \in R^k$, $v \geq 0$, and $\sum_{i=1}^{k} v_i = e'v = 1$.

Thus, we know a perceptron that correctly classifies $\mathcal{A}$ and $\mathcal{B}$ exists if and only if no point can be written as a convex combination of both $\mathcal{A}$ and $\mathcal{B}$, i.e.,

$$\left.\begin{array}{c} A'u = B'v \\ e'u = e'v = 1 \\ u \geq 0 \quad v \geq 0 \end{array}\right\} \quad \textit{has no solution.} \tag{5}$$

At first glance, these two characterizations of what a perceptron can learn, equations (2) and (5), look unrelated. But in fact, using Gordan's Theorem of the Alternative [16], it can be shown that equations (2) have a solution if and only if equations (5) have no solution (see Theorem A.1 in Appendix). The two formulations are in different spaces but they solve the same underlying problem. Either problem can be used as a definition of linear separability of two sets. This is one of many examples of how duality is an important tool in geometry. In our case, the dual of the problem of finding the separating plane is the problem of determining whether the two convex hulls intersect. In the next section, we will use duality to develop methods for training the perceptron.

4 Training: Linearly Separable Case

In this section we will use an intuitive geometric argument to motivate several different optimization approaches to constructing a perceptron for linearly separable problems.

A geometric procedure to construct the "best" plane to separate two linearly separable sets is the following. Find the two closest points in the convex hulls of $\mathcal{A}$ and $\mathcal{B}$. Construct a line segment between the two points. The plane, orthogonal to the line segment, that bisects the line segment is the separating plane. An example of such a separating plane is given in Figure 2. Intuitively this plane is "best" because the two sets are as far away from the separating plane as possible, thus improving the likelihood of correctly classifying future points. Any definition of "best", based on the training sets, is only an approximation of the actual goal of finding the plane that generalizes best.

The problem of finding two closest points in the convex hulls can be written as an optimization problem:

$$\begin{array}{ll} \min\limits_{u,v} & \frac{1}{2}\left\|A'u - B'v\right\|^2 \\ s.t. & e'u = 1 \quad e'v = 1 \\ & u \geq 0 \quad v \geq 0 \end{array} \tag{6}$$

The perceptron, (w, γ), is constructed from the results of Problem (6). The weights w are the normal of the separating plane. The normal w is exactly the vector between the

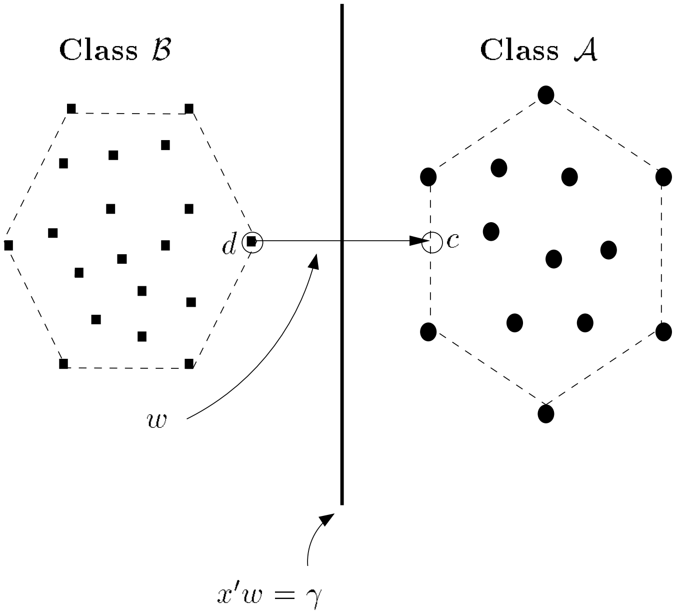

Figure 2: The two closest points of the convex hulls determine the separating plane.

two closest points in the convex hulls. Let $\bar{u}$ and $\bar{v}$ be an optimal solution of (6). The two closest points, c and d, are $c = A'\bar{u}$ and $d = B'\bar{v}$. The weights of the perceptron are the difference of these two points: $w = c - d = A'\bar{u} - B'\bar{v}$. The threshold, γ, is the distance from the origin to the point halfway between the two closest points along the normal w: $\gamma = (\frac{c+d}{2})'w = \frac{(\bar{u}'Aw+\bar{v}'Bw)}{2}$.

Using duality we can transform this problem to an optimization problem in the space of the perceptron. The dual of the following problem is Problem (6). See Theorem A.2 in the Appendix for the derivation.

$$\begin{array}{ll} \min\limits_{w,\alpha,\beta} & \frac{1}{2}\|w\|^2 - (\alpha - \beta) \\ s.t. & \quad Aw - \alpha e \geq 0 \\ & -Bw + \beta e \geq 0 \end{array} \tag{7}$$

There is a simple geometric interpretation of Problem (7). Examine Figure 3. All the points in $\mathcal{A}$ are constrained to be on the "greater than" side of the α-plane $x'w = \alpha$. All the points in $\mathcal{B}$ are constrained to be on the "less than" side of the β-plane $x'w = \beta$. The α-plane is a support plane of $\mathcal{A}$ and the β-plane is a support plane of $\mathcal{B}$. The distance between these two parallel support planes and thus between the two convex hulls is

$$\frac{\alpha - \beta}{\|w\|} \tag{8}$$

The objective function of (7) minimizes $\|w\|^2$ and maximizes $\alpha - \beta$. Thus Problem (7) maximizes the distance between the two parallel planes. Let $\hat{w}$, $\hat{\alpha}$, and $\hat{\beta}$ be a solution of (7). For separable problems, $\hat{\alpha} - \hat{\beta} > 0$. The two parallel supporting planes are shown in

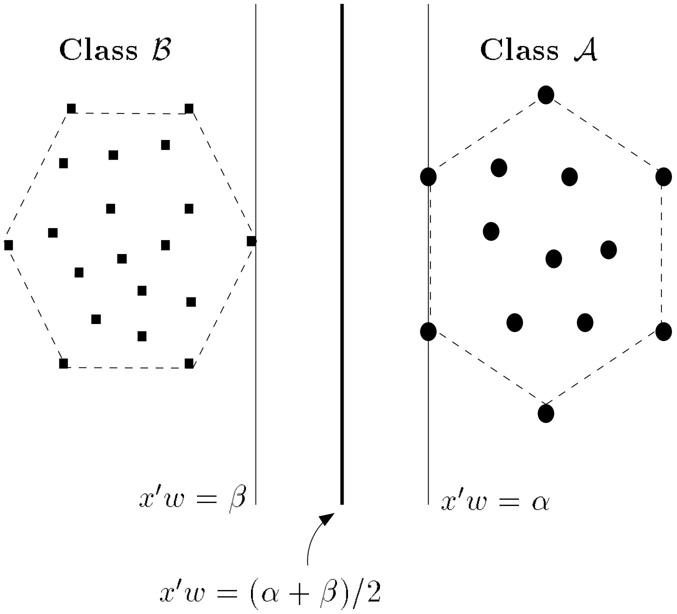

Figure 3: The dual problem maximizes the distance between two parallel supporting planes.

Figure 3. The final separating plane is the plane halfway between the two parallel planes: $x'\hat{w} = \frac{\hat{\alpha}+\hat{\beta}}{2}$.

Both Problems (6) and (7) are quadratic programming problems with linear constraints. They can be solved using standard mathematical programming packages [25, 22]. The choice of which problem to use in practice depends on the characteristics of the underlying problem. In Problem (6), the constraints are very simple, and the number of variables depends only on the total number of points. Thus, when the number of attributes is very large, Problem (6) would be preferable.

Problem (7) provides a unifying framework for explaining other prior optimization approaches to constructing the perceptron. By transforming (7) into mathematically equivalent optimization problems, different algorithms result. There are two simplifications that can be performed to transform the problem. Either $\|w\|^2 = 1$ is added as a constraint and $\alpha - \beta$ is maximized, or $\alpha - \beta$ is fixed at some positive value and $\|w\|^2$ is minimized.

The former approach results in the following problem:

$$\begin{array}{rc} \max\limits_{w,\alpha,\beta} & \alpha - \beta \\ s.t. & Aw - \alpha e \geq 0 \\ & -Bw + \beta e \geq 0 \\ & \|w\|^2 = 1 \end{array} \tag{9}$$

The difficulty with this approach is that the resulting Problem (9) is much harder to solve than Problem (7) since the constraints are now nonlinear and nonconvex. However, by substituting the constraint $\|w\|_\infty = 1$, the problem can be solved in polynomial time using $2n$ linear programs. Polynomial algorithms exist for linear programming problems. In practice, existing general purpose linear program solvers are very fast and numerically stable [22]. By changing the norm used in the constraints, the quality of the solution is not degraded and the solution of the problem becomes much easier.

Specifically, the first n linear programs are defined for $j = 1, \ldots, n$ as:

$$\begin{array}{ll} \max\limits_{w,\alpha,\beta} & \alpha - \beta \\ s.t. & Aw - \alpha e \geq 0 \\ & -Bw + \beta e \geq 0 \\ & -e \leq w \leq e \\ & w_j = \ 1 \end{array} \tag{10}$$

The second n linear programs are defined for $j = 1, \ldots, n$ as:

$$\begin{array}{ll} \max\limits_{w,\alpha,\beta} & \alpha - \beta \\ s.t. & Aw - \alpha e \geq 0 \\ & -Bw + \beta e \geq 0 \\ & -e \leq w \leq e \\ & w_j = -1 \end{array} \tag{11}$$

A solution of one the $2n$ problems with the greatest value of $\alpha - \beta$ is the optimal answer.[1] This approach, called the Multisurface Method of Pattern Recognition (MSM) [15], was used in the initial implementation of the automated breast cancer diagnosis system described in the introduction [19, 34].

The second general method is to fix $\alpha - \beta > 0$. If we set $\alpha - \beta = 2$ by defining $\alpha = \gamma + 1$ and $\beta = \gamma - 1$, then Problem (7) becomes

$$\begin{array}{ll} \min\limits_{w,\gamma} & \frac{1}{2} \|w\|^2 \\ s.t. & Aw - (\gamma + 1)e \geq 0 \\ & -Bw + (\gamma - 1)e \geq 0 \end{array} \tag{12}$$

Problem (12) is exactly the "Optimal Plane" proposed by Vapnik [33]. By using optimality conditions it can be shown that Problem (12) and Problem (6) are equivalent on separable problems. The proof is provided in the Appendix.

The above are only a few of the many existing methods for training a perceptron. We provide a few pointers to other approaches. This is not a comprehensive list. A notable class of algorithms is comprised of Rosenblatt's Perceptron algorithm [27, 10, 4] and the Motzkin-Schoenberg algorithm for finding the solution of linear inequalities [23]. A perceptron is also known as a linear discriminant. So any linear discriminant algorithms such as in the book [7] may be used. A single linear program can be used to construct a separating plane in polynomial time [14, 11]. Edelsbrunner proposed an algorithm with $O(\log\ m + \log\ k)$ complexity [8]. Statistical methods such as Fisher's Linear Discriminant may also be applied [9].

[1] When applied to inseparable problems, the solution that misclassified the least number of points is selected.

The problems formulated in this section are only for the linearly separable case. Care must be taken when applying any method for the linearly separable case to the linearly inseparable case. One pitfall is that the meaningless solution $w = 0$ is feasible and optimal for Problems (6) and (7) since the convex hulls of $\mathcal{A}$ and $\mathcal{B}$ intersect. Problem (12) is not even feasible in the inseparable case. Thus solving these problems yields no meaningful solution. In the next section we will discuss approaches for the linearly inseparable case.

5 Training: Linearly Inseparable Case

Frequently the sets $\mathcal{A}$ and $\mathcal{B}$ cannot be separated by any plane. For the linearly inseparable case, the separation problem becomes more difficult. There is no clear definition of what constitutes the "best" separating plane. The general approach is to develop some measure of the misclassification error and then minimize that error. Depending on the choice of error measurement, different optimization problems arise. We will begin by generalizing the above approach to the case when the two sets are not strictly linearly separable.

Figure 4 illustrates how Problem (9) is applied to a linearly inseparable problem. Recall Problem (9) was

$$\begin{array}{ll} \max\limits_{w,\alpha,\beta} & \alpha - \beta \\ s.t. & Aw - \alpha e \geq 0 \\ & -Bw + \beta e \geq 0 \\ & \|w\|^2 = 1 \end{array}$$

Geometrically, all the points in $\mathcal{A}$ are on the greater than side of the plane $x'w = \alpha$ and all the points in $\mathcal{B}$ are on the less than side of the plane $x'w = \beta$. Since the convex hulls of the sets $\mathcal{A}$ and $\mathcal{B}$ do intersect, we know that only α and β satisfying $\alpha - \beta \leq 0$ are feasible. Thus by maximizing $\alpha - \beta$ the two planes are being moved as close together as possible. At optimality, the point with the largest misclassification error is at a distance of exactly $|\alpha - \beta|/2$ from the separating plane. Thus Problem (9) minimizes the maximum distance of any misclassified point to the separating plane. However this property is also a limitation of the method, since adding or subtracting one "noisy" or hard-to-classify point can dramatically change the solution. Nor is it clear that minimizing the maximum error results in the best separating plane.

An alternate method is to minimize the sum of the misclassification errors. We will measure the misclassification error as the distance of a misclassified point from the appropriate supporting plane. We say a point A_i in $\mathcal{A}$ is misclassified if $-A_i w + \alpha \geq 0$. Define $y_i = -A_i w + \alpha$ if $-A_i w + \alpha > 0$, i.e. if A_i is misclassified. Otherwise let $y_i = 0$. Similarly, let $z_j = B_j w - \beta$ if $B_j w - \beta > 0$, i.e. if B_j is misclassified. Otherwise let $z_j = 0$. We can construct a linear program to minimize the sum of the y_i, $i = 1, \ldots, m$, and the sum of the z_j, $j = 1, \ldots, k$ where m and k are the number of points in $\mathcal{A}$ and $\mathcal{B}$ respectively.

Consider the following variant of the Robust Linear Programming (RLP) approach of

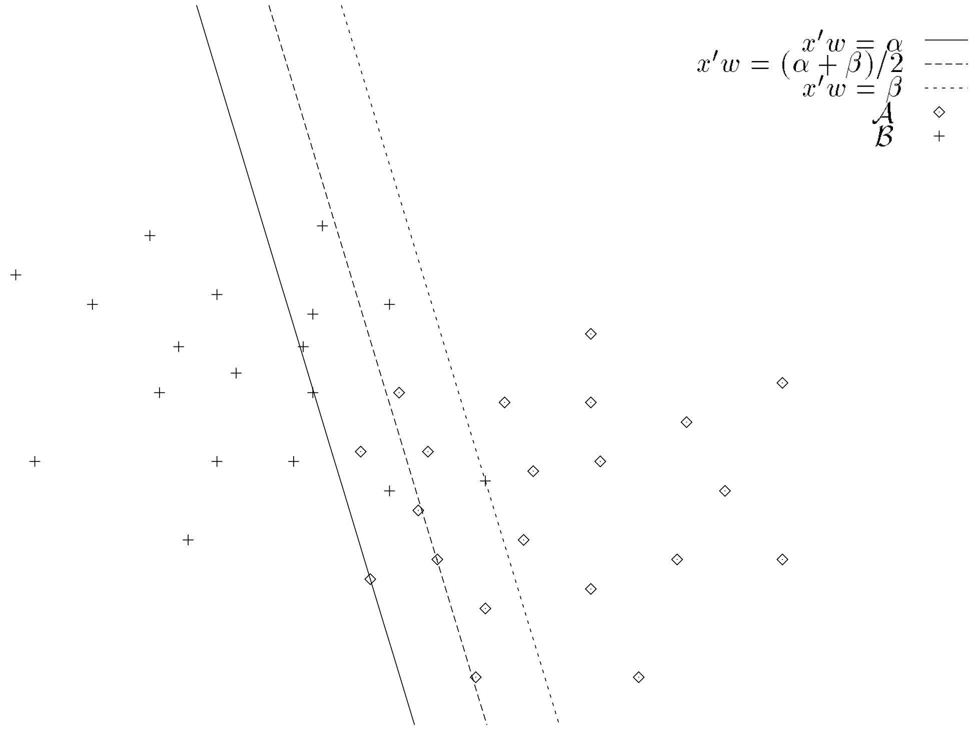

Figure 4: Minimizing the maximum error in the inseparable case.

Bennett and Mangasarian [2].

$$\begin{array}{ll} \min\limits_{w,\alpha,\beta,y,z} & \frac{1}{m}e'y + \frac{1}{k}e'z \\ s.t. & Aw - \alpha e + y \geq 0 \\ & -Bw + \beta e + z \geq 0 \\ & \alpha - \beta = 2 \\ & y \geq 0 \quad z \geq 0 \end{array} \tag{13}$$

The nonnegative slack variable y relaxes the constraints that all points in $\mathcal{A}$ be on the greater than side of the plane $x'w = \alpha$. If a point A_i is misclassified it will be on the less than side of the plane $x'w = \alpha$. If the point A_i is classified correctly, then $y_i = 0$. If the point A_i is misclassified then $y_i = -A_i w + \alpha > 0$ is proportional to the distance of that point from the plane $x'w = \alpha$. Similarly, variable z relaxes the constraints that all points in $\mathcal{B}$ be on the less than side of the plane $x'w = \beta$. The same arguments hold for z_j, the point B_j, and the plane $x'w = \beta$. The coefficients $\frac{1}{m}$ and $\frac{1}{k}$ were chosen to guarantee that there always exists a meaningful optimal solution $w \neq 0$. This is proved using optimality conditions in [2]. If $w = 0$ then the perceptron puts all the points in one class and the solution is useless. Problem (13) is used in the most recent version of the breast cancer diagnosis system described in the introduction [32, 35].

Both MSM (10 and 11) and RLP (13) have been successfully used in practice [34, 19, 35]. They do, however, have some limitations. MSM works well for the separable case. As noted above on problems with noisy data, the method performs poorly since it minimizes the maximum error [2]. RLP achieves very strong results for inseparable problems [3, 35]. For separable problems, however, any separating plane, if scaled appropriately, is optimal for RLP since $e'y = e'z = 0$. So RLP is not very well defined in the separable case. The next two approaches preserve the benefits of both RLP and MSM.

We present two ways of combining MSM with RLP (13). The first method minimizes the 2-norm of w as follows.

$$\begin{array}{ll} \min\limits_{w,y,z,\alpha,\beta} & (1-\lambda)(\frac{1}{m}e'y + \frac{1}{k}e'z) + \frac{\lambda}{2}w'w \\ s.t. & Aw - \alpha e + y \geq 0 \\ & -Bw + \beta e + z \geq 0 \\ & \alpha - \beta = 2 \\ & y \geq 0 \quad z \geq 0 \end{array} \tag{14}$$

where $0 < \lambda < 1$ is a fixed constant. This problem minimizes the misclassification error in two ways. The term $(1-\lambda)(\frac{1}{m}e'y + \frac{1}{k}e'z)$ reduces the average distance of the misclassified points from the relaxed supporting planes. The term $\frac{\lambda}{2}w'w$ decreases the maximum classification error which corresponds to the distance between the relaxed supporting planes. Since this problem is a very minor variation of the Generalized Optimal Plane of Cortes and Vapnik [5, 33], we refer to it as GOP. The problem is a nonlinear perturbation of the RLP. When λ is close to 0 the RLP objective is emphasized. Using thereoms on nonlinear perturbation of linear programming in [18], we know that there exists some positive number $\bar{\lambda}$ such that if $\lambda \in (0, \bar{\lambda}]$, there exists a solution of GOP (14) that also solves RLP (13). This means that for λ sufficiently small, GOP will choose one of the solutions of RLP that minimizes the distance between the supporting planes.

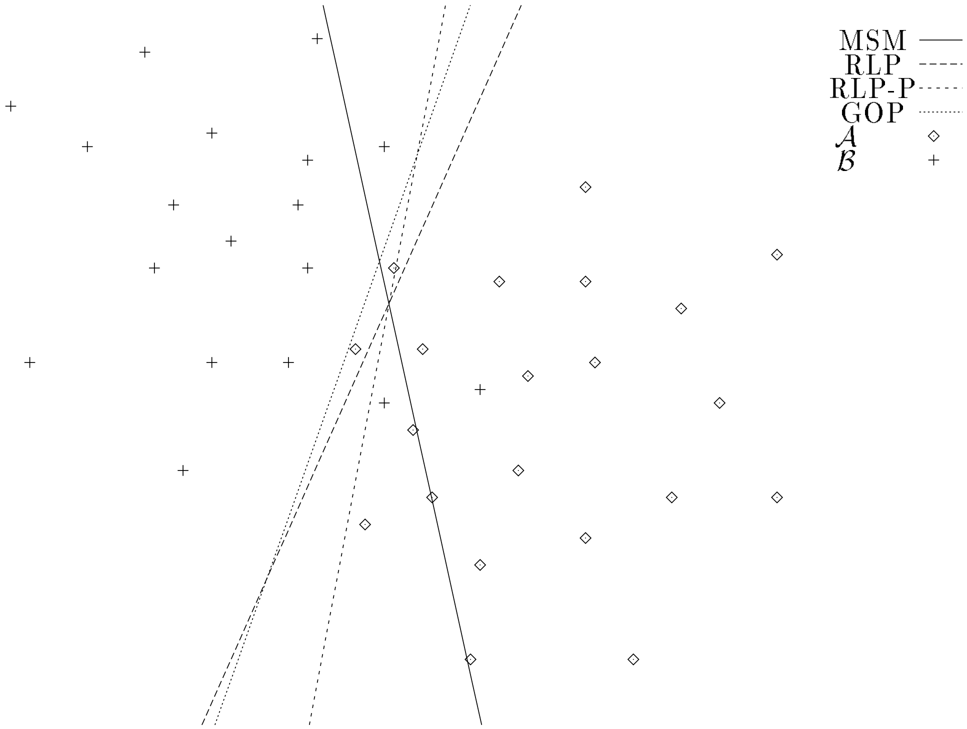

Figure 5: Results of MSM, RLP, RLP-P, and GOP on a linearly inseparable problem.

The dual of GOP (14) is closely related to the problem of finding the two closest points in the convex hulls. The dual of GOP (14) (see Appendix for derivation) is

$$\begin{array}{ll} \min\limits_{u,v} & \frac{1}{2\lambda}\left\|A'u - B'v\right\|^2 - 2\delta \\ s.t. & e'u = e'v = \delta \\ & \frac{1-\lambda}{m}e \geq u \geq 0 \quad \frac{1-\lambda}{k}e \geq v \geq 0. \end{array} \tag{15}$$

To better understand the problem, consider the case when $\delta = 1$. The points $A'u$ and $B'v$ are in the convex hulls of $\mathcal{A}$ and $\mathcal{B}$ respectively. The objective still minimizes the distance between these points. Each $u_i > 0$ corresponds to an example or vector that determines the convex combination c of $\mathcal{A}$. Vapnik refers to these examples with positive multipliers as support vectors [33]. Recall that the convex combination c of $\mathcal{A}$ is $c = A'u$ with $e \geq u \geq 0$ and $e'u = 1$. In this problem, the u and v are further constrained: $e > \frac{1-\lambda}{m}e \geq u \geq 0$ and $e > \frac{1-\lambda}{k}e \geq v \geq 0$, This forces more u_i and v_j to be positive, and therefore more points in $\mathcal{A}$ and $\mathcal{B}$ are used in the convex combinations. Thus the solution is dependent on more points from each of the classes. Roughly speaking, when the convex combination c is forced to contain many points, then c is forced away from the edge of the convex hull. Thus the two points selected must be more centrally located in the convex hulls of the two sets. As λ increases, more and more points in $\mathcal{A}$ and $\mathcal{B}$ must be used. Increasing δ has a similar effect.

A linear programming version of GOP can be constructed by replacing the norm used to minimize the weights w. Recall that the GOP objective minimizes the square of the 2-norm of w, $\|w\|^2 = w'w$. The 1-norm of w, $\|w\|_1 = e'|w|$, is used instead. The absolute value function can be removed by introducing the variable s and the constraints $-s \leq w \leq s$. The GOP objective is then modified by substituting $e's$ for $\frac{w'w}{2}$. At optimality, $s = |w|$. The resulting Perturbed Robust Linear Program (RLP-P) is:.

$$\begin{array}{ll} \min\limits_{w,\alpha,\beta,y,z} & (1-\lambda)(\frac{1}{m}e'y + \frac{1}{k}e'z) + \lambda e's \\ s.t. & Aw - \alpha e + y \geq 0 \\ & -Bw + \beta e + z \geq 0 \\ & \alpha - \beta = 2 \\ & -s \leq w \leq s \\ & y \geq 0 \quad z \geq 0 \quad s \geq 0 \end{array} \tag{16}$$

As in the GOP method, the RLP-P method minimizes both the average distance of the misclassified points from the relaxed supporting planes and the maximum classification error. The main advantage of the RLP-P method over the GOP problem is that RLP-P is a linear program solvable in polynomial time while GOP is a quadratic program.

We have discussed several methods for training the perceptron. Figure 5 illustrates the results of the four methods, MSM, RLP, RLP-P, and GOP, on a sample problem. On this simple dataset, GOP appears to choose a more intuitive plane. Recall the best plane is the plane that generalizes best. In the next sections, the generalization of these methods is examined through an empirical comparison on practical classification problems.

6 Practical Applications

The perceptron model is simple yet it can be very successful on many real-world applications. These applications are very diverse and some address challenging scientific questions. We will perform experiments on the following problems: diagnosis of heart disease, the diagnosis of breast cancer, using voting records of congressmen to determine party affiliation, and using sonar signals to differentiate mines from rocks. These are just a few of the many applications possible. In practice most applications involve linearly inseparable datasets. However, the dataset used in the sonar example is linearly separable. A brief description of each of these applications follows.

The first application is determining whether or not a patient has heart disease. Several relevant attributes, such as age, cholesterol level, and resting blood pressure, are collected for each patient. The perceptron model is applied to a dataset containing these attributes for a number of patients whose heart disease status is known. By evaluating a new patient's attributes with respect to the separating plane a diagnosis is made. The Cleveland Heart Disease Database (Heart) is a publicly available dataset that contains information on 297 patients using 13 attributes [6].

A second application, as discussed previously, is the diagnosis of breast cancer. To evaluate whether a tumor is benign or malignant, a fine needle aspiration is performed collecting a small amount of tissue from the tumor for examination. Several measurements such as clump thickness, uniformity of cell size, and uniformity of cell shape are collected. A mathematical programming approach incorporating the RLP has been employed in clinical practice [20, 34]. This data was collected before the computer imaging techniques were used for measuring attributes as discussed in the introduction. They report 100% correctness on 131 new cases that have been diagnosed. Additionally, many studies have been made on the Wisconsin Breast Cancer Database (Cancer) [34]. This publicly available dataset contains information on 682 patients using nine integer attributes.

The voting patterns of congressmen can be used to determine party affiliation. A specific example of this application is the 1984 United States Congressional Voting Records Database (House Votes) [30]. This is a publicly available dataset. Each instance of the dataset represents a U.S. House of Representatives Congressman. Information on 435 congressmen is given. Each congressman is classified as either democrat or republican. The attributes consist of 16 key votes. These attributes have values of y (yea), n (nay), and "?". A value of "?" indicates that a position was not made known. To solve the problem numerically, we let y, n, and "?" be 2, -2, and 0 respectively. An interesting use for this dataset is to determine whether a congressman is supporting their parties views. A congressman that is difficult to classify may represent someone who tends to vote outside their party.

The final application presented is the use of sonar signals to distinguish mines from rocks. Mine attributes are obtained by bouncing sonar signals off a metal cylinder. The sonar signal is transmitted at various angles with rises in frequency. A similar procedure is performed to obtain the rock attributes. The publicly available Sonar dataset represents 208 mines and rocks [12]. Sixty real-valued attributes between 0.0 and 1.0 are collected for each mine or rock. The value of the attribute represents the amount of energy within a particular frequency band, integrated over a certain period of time. As was stated previously, this dataset is an example of a practical application with linearly separable points. Most datasets for practical applications are linearly inseparable.

We have limited this investigation to datasets that are publicly available via the World Wide Web. All of the above datasets are available via anonymous file transfer protocol (ftp) from the UCI Repository of Machine Learning Databases and Domain Theories [24] at ftp://ftp.ics.uci.edu/pub/machine-learning-databases. The following section contains computational results for the MSM, RLP, RLP-P, and GOP methods on these datasets.

7 Computational Results

This section contains a computational comparison of the four methods, MSM, RLP, RLP-P and GOP. We report results on four datasets: Heart, Cancer, Sonar, and House Votes. The previous section contains a description of each dataset and its application.

All four methods were implemented using the MINOS [25] mathematical programming software package. Other optimization packages could easily be substituted. The results of GOP and RLP-P are dependent on the parameter λ. We report results for GOP using $\lambda = .1$ and $\lambda = .05$ and for RLP-P using $\lambda = .05$ and $\lambda = .02$.

The best algorithm is the one that generalizes best, i.e., the algorithm most accurate on future points. Since the future points are unknown, we use an experimental technique called cross validation [31] to estimate the accuracy on future points. In 10-fold cross validation a dataset is divided randomly into ten disjoint parts of equal size. The method is then trained on nine of these parts. These nine parts are called the training set. The tenth part, the testing set, is reserved to test the accuracy of the plane obtained during training. This procedure is repeated ten times allowing each part to be used for testing. The accuracies on the training and testing sets are averaged over the 10 trials. Both training and testing set accuracies are reported. The testing set accuracy is an estimate of generalization. While training set accuracy is important too, it is possible for the training set accuracy to be high while the testing set accuracy is low. This phenomena, called overfitting, results when the perceptron learns the wrong concept. Ideally, the training set accuracy should be just slightly greater than the testing set accuracy. The training and testing percent accuracies are given in Table 1. The best results are shown in bold.

Table 1 indicates that the GOP and RLP-P methods have larger testing set accuracies than both the RLP and MSM methods on three of the four datasets. However, many of the differences reported are not statistically significant. On the Heart dataset, GOP ($\lambda = .05$) performed as well as RLP and better than GOP ($\lambda = .1$), MSM and RLP-P. This illustrates one drawback to the GOP and RLP-P methods: The quality of the solution is extremely dependent on the choice of λ. With other choices of the parameter λ, the testing set accuracies for the GOP and RLP-P methods could continue to improve.

The Sonar dataset is linearly separable, i.e., a single hyperplane can successfully separate the two classes of points. For linearly separable datasets, RLP and MSM will always choose planes that completely separate the two classes. Recall that MSM attempts to push apart two parallel supporting planes of the two convex hulls of the classes. RLP can pick any linearly separable plane. Thus we anticipated MSM would test better than RLP on these datasets. The testing set accuracies reported for the Sonar dataset indicate that MSM did

Dataset		MSM	RLP	GOP (.1)	GOP (.05)	RLP-P (.05)	RLP-P (.02)
Heart	Training	75.01	85.07	85.11	**85.22**	84.92	84.89
	Testing	72.44	**83.50**	82.83	**83.50**	82.15	83.16
Cancer	Training	94.93	**97.72**	97.57	97.54	97.46	97.56
	Testing	95.01	97.21	97.51	97.36	**97.65**	97.07
House Votes	Training	93.54	**97.52**	96.14	97.34	95.35	95.91
	Testing	93.79	95.17	95.63	**95.86**	94.48	95.40
Sonar	Training	**100.0**	**100.0**	79.43	81.30	79.38	79.91
	Testing	74.05	69.71	75.00	76.44	75.48	**76.92**

Table 1: Average Training and Testing Accuracy (%)

perform better than RLP. Roughly speaking GOP and RLP-P try to push apart two relaxed supporting planes of the convex hulls. If λ is sufficiently large, GOP and RLP-P will not necessarily result in a separating plane even when the sets are linearly separable. This was the case on the Sonar Data. By selecting a plane with errors, the GOP and RLP-P methods found better solutions than both MSM and RLP. This is an illustration of overfitting: the best training set accuracy did not result in the best testing set accuracies.

As indicated in the previous section, these datasets along with many others are publicly available. Thus, with the aid of a mathematical programming software package, the reader can reconstruct results reported in this section. Further computational studies may also be of interest. For example, an investigation of the affects of varying choices of λ have on the GOP and RLP-P solutions could be performed.

8 Extension to Nonlinear Separation

As we discovered in Section 3, perceptrons are limited to problems that are totally or almost linearly separable. When the underlying problem is highly nonlinear, the perceptron will fail. We provide a few references to three major ways to extend the perceptron model to nonlinear separators. The most popular is the multilayer perceptron or neural network. A neural network is created from an interconnecting network of threshold/perceptron type units. We invite the reader to consult [13, 17] or one of the many books on this subject. Another approach is decision trees. In decision trees, a linear separation is constructed that divides the attribute space into two parts. If the parts contain points all or largely all of one class the algorithm stops. If not, the process is repeated in each of the half spaces until the desired accuracy is achieved. Perceptrons are one such way to construct the decision [26, 4, 1, 17]. The final method is to construct nonlinear mappings of the attributes and then construct a linear discriminant in the augmented attribute space. The resulting problem is linear in its parameters but a nonlinear decision surface is created. For example to create a quadratic separator with two attributes, x_1 and x_2, you would add three more attributes, x_1^2, x_2^2, and x_1x_2. The separating surface would now be $x_1w_1 + x_2w_2 + x_1^2w_3 + x_2^2w_4 + x_1x_2w_4 = \gamma$. Support Vector Networks [33] and Polynomial Networks [29, 28] are examples of this type of approach.

9 Conclusions

We have studied the problem of training a perceptron to classify points from two sets. We showed that a perceptron can only correctly classify two sets if the sets are either linearly separable or, equivalently, their convex hulls do not intersect. In the separable case, we argued geometrically that the perceptron can be constructed by finding the two closest points in the convex hulls of the two sets. This problem was formulated as a minimization problem. The dual problem for the separable case maximized the distance between the two parallel supporting planes of the two sets. From these two formulations we derived several existing optimization problems for training a perceptron. We explored different ways of extending these methods to the linearly inseparable case. Different approaches result depending on how we measure the misclassification error. Four different algorithms were explored both theoretically and computationally on practical problems. Our computational results indicated that the Generalized Optimal Plane and the Perturbed Robust Linear Programming method maintained the benefits of the other two approaches while avoiding some of their limitations. The datasets we investigated are publicly available, so interested readers can conduct their own experiments.

Acknowledgements

Thanks to Olvi Mangasarian, Harry McLaughlin and Anne McEntee for their valuable suggestions. This material is based on research supported by the National Science Foundation Grant 949427.

Appendix

A Supporting Proofs

The details of proofs and arguments of the main text are supplied in this appendix.

In this theorem we show that the existence of a separating plane and the non-intersection of the convex hulls of two sets are equivalent definitions of linear separability.

Theorem A.1 (Characterization of linear separability) *Let $\mathcal{A}$ and $\mathcal{B}$ be nonempty point sets in R^n with m and k points respectively. The following are equivalent:*

(a) $\mathcal{A}$ and $\mathcal{B}$ are linearly separable; that is, there exist $w \in R^n$ and $\gamma \in R$ such that

$$\begin{array}{r} Aw - e\gamma > 0 \\ -Bw + e\gamma > 0 \end{array} \qquad (17)$$

(b) *The convex hulls of $\mathcal{A}$ and $\mathcal{B}$ do not intersect; that is, there does not exist $u \in R^m$ and $v \in R^k$ such that*

$$\begin{array}{c} A'u = B'v \\ e'u = e'v = 1 \\ u \geq 0 \quad v \geq 0 \end{array} \tag{18}$$

Proof. Gordan's Theorem of the Alternative [16] states:

For each given matrix C, either
I $\quad Cx > 0$ has a solution x
or
II $\quad C'y = 0,\ y \geq 0$ has a solution y
but never both.

Theorem A.1 follows directly with $C = \begin{bmatrix} A & -e \\ -B & e \end{bmatrix}$, $x = \begin{bmatrix} w \\ \gamma \end{bmatrix}$, $y = \begin{bmatrix} u \\ v \end{bmatrix}$. □

The problem of finding the two closest points in the convex hulls of two sets is the dual of the problem of finding the best separating plane.

Theorem A.2 (Dual of separating problem) *Let $\mathcal{A}$ and $\mathcal{B}$ be nonempty point sets in R^n with m and k points respectively. The primal problem:*

$$\begin{array}{ll} \min\limits_{w,\alpha,\beta} & \frac{1}{2}\|w\|^2 - (\alpha - \beta) \\ s.t. & \ Aw - \alpha e \geq 0 \\ & -Bw + \beta e \geq 0 \end{array} \tag{19}$$

has the following dual problem:

$$\begin{array}{ll} \min\limits_{u,v} & \frac{1}{2}\|A'u - B'v\|^2 \\ s.t. & e'u = 1 \quad e'v = 1 \\ & u \geq 0 \quad v \geq 0 \end{array} \tag{20}$$

Proof. The dual problem maximizes the Lagrangian function of (19), $L(w, \alpha, \beta, u, v)$, subject to the constraints that the partial derivatives of the Lagrangian with respect to the primal variables are equal to zero [15]. Specifically, the dual of (20) is:

$$\begin{array}{ll} \max\limits_{w,\alpha,\beta,u,v} & L(w,\alpha,\beta,u,v) = \frac{1}{2}\|w\|^2 - (\alpha - \beta) - u'(Aw - e\alpha) - v'(-Bw + e\beta) \\ s.t. & \frac{\partial L}{\partial w} = w - A'u + B'v = 0 \\ & \frac{\partial L}{\partial \alpha} = -1 + e'u = 0 \\ & \frac{\partial L}{\partial \beta} = 1 - e'v = 0 \\ & u \geq 0, \quad v \geq 0 \end{array} \tag{21}$$

where $\alpha, \beta \in R$, $w \in R^n$, $u \in R^k$, and $v \in R^m$. To simplify the problem, substitute in $w = (A'u - B'v)$.

$$\begin{array}{ll} \max\limits_{\alpha,\beta,u,v} & \frac{1}{2}\|A'u - B'v\|^2 - (\alpha - \beta) - u'A(A'u - B'v) + e'u\alpha + v'B(A'u - B'v) - e'v\beta \\ s.t. & e'u = e'v = 1 \\ & u, v \geq 0 \end{array} \tag{22}$$

Using the fact that $\|A'u - B'v\|^2 = (u'A - v'B)(A'u - B'v)$, this simplifies to:

$$\begin{array}{ll} \min\limits_{u,v} & \frac{1}{2}\|A'u - B'v\|^2 \\ s.t. & e'u = e'v = 1 \\ & u, v \geq 0 \end{array} \tag{23}$$

□

The dual Problem (19) of finding the closest two points is equivalent to the Optimal Separating Plane problem (24) proposed by Vapnik [33]. Specifically, every solution of one problem can be used to construct a corresponding solution of the other.

Theorem A.3 (Equivalence of two separating problems) *For separable sets, Problem (19) and the following problem are equivalent*

$$\begin{array}{ll} \min\limits_{w,\gamma} & \frac{1}{2}\|w\|^2 \\ s.t. & \ \ Aw - (\gamma + 1)e \geq 0 \\ & -Bw + (\gamma - 1)e \geq 0 \end{array} \tag{24}$$

Proof. Since both problems are convex minimization problems with linear constraints, all we must show is that if the optimality conditions of one problem are satisfied then the optimality conditions of the other are satisfied.

First we will show that any solution of Problem (24) can be used to construct a solution of Problem (19). Let $\bar{u}, \bar{v}, \bar{w}, \bar{\gamma}$ be an optimal solution of Problem (24), then the following Karush-Kuhn-Tucker optimality conditions are satisfied [15]:

$$\begin{array}{l} \ \ A\bar{w} - (\bar{\gamma} + 1)e \geq 0 \\ -B\bar{w} + (\bar{\gamma} - 1)e \geq 0 \\ \bar{u}'(A\bar{w} - (\bar{\gamma} + 1)e) = 0 \\ \bar{v}'(-B\bar{w} + (\bar{\gamma} - 1)e) = 0 \\ \bar{w} = A'\bar{u} - B'\bar{v} \\ e'\bar{u} = e'\bar{v} \\ \bar{u} \geq 0, \bar{v} \geq 0. \end{array} \tag{25}$$

Let $\delta = e'\bar{u} = e'\bar{v}$. Define $\hat{u} = \frac{\bar{u}}{\delta}$, $\hat{v} = \frac{\bar{v}}{\delta}$, $\hat{w} = \frac{\bar{w}}{\delta}$, $\hat{\alpha} = \frac{\bar{\gamma}+1}{\delta}$, and $\hat{\beta} = \frac{\bar{\gamma}-1}{\delta}$ Then $(\hat{u}, \hat{v}, \hat{w}, \hat{\alpha}, \hat{\beta})$ satisfy

$$\begin{array}{l} \hat{w} = A'\hat{u} - B'\hat{v} \\ A\hat{w} - \hat{\alpha}e = \frac{1}{\delta}(A\bar{w} - (\bar{\gamma} + 1)e) \geq 0 \\ -B\hat{w} + \hat{\beta}e = \frac{1}{\delta}(-B\bar{w} + (\bar{\gamma} - 1)e) \geq 0 \\ \hat{u}'(A\hat{w} - \hat{\alpha}e) = \frac{\bar{u}'}{\delta^2}(A\bar{w} - (\bar{\gamma} + 1)e) = 0 \\ \hat{v}'(-B\hat{w} + \hat{\beta}e) = \frac{\bar{v}'}{\delta^2}(-B\bar{w} + (\bar{\gamma} - 1)e) = 0 \\ e'\hat{u} = e'\hat{v} = 1 \\ \hat{u} \geq 0, \hat{v} \geq 0. \end{array} \tag{26}$$

which are the optimality conditions of Problem (19).

Now we will show that any solution of Problem (19) can be used to construct a solution of Problem (24). Let $(\hat{u}, \hat{v}, \hat{w}, \hat{\alpha}, \hat{\beta})$ satisfy the optimality conditions (26) of Problem (19).

For the separable case, $\hat{\alpha} - \hat{\beta} > 0$. Define $\bar{u} = \frac{2\hat{u}}{\hat{\alpha}-\hat{\beta}}$, $\bar{v} = \frac{2\hat{v}}{\hat{\alpha}-\hat{\beta}}$, $\bar{w} = \frac{2\hat{w}}{\hat{\alpha}-\hat{\beta}}$, and $\bar{\gamma} = \frac{\hat{\alpha}+\hat{\beta}}{\hat{\alpha}-\hat{\beta}}$. Then we know

$$\begin{array}{l}
\bar{w} = A'\bar{u} - B'\bar{v} \\
A\bar{w} - (\bar{\gamma}+1)e = \frac{1}{\hat{\alpha}-\hat{\beta}}(2A\hat{w} - ((\hat{\alpha}+\hat{\beta}) + (\hat{\alpha}-\hat{\beta}))e) = \frac{2}{\hat{\alpha}-\hat{\beta}}(A\hat{w} - \hat{\alpha}e) \geq 0 \\
-B\bar{w} - (\bar{\gamma}-1)e = \frac{1}{\hat{\alpha}-\hat{\beta}}(-2B\hat{w} + ((\hat{\alpha}+\hat{\beta}) - (\hat{\alpha}-\hat{\beta}))e) = \frac{2}{\hat{\alpha}-\hat{\beta}}(-B\hat{w} + \hat{\beta}e) \geq 0 \\
\bar{u}'(\ A\bar{w} - (\bar{\gamma}+1)e) = \frac{4\hat{u}'}{(\hat{\alpha}-\hat{\beta})^2}(A\hat{w} - \hat{\alpha}e) = 0 \\
\bar{v}'(-B\bar{w} + (\bar{\gamma}-1)e) = \frac{4\hat{v}'}{(\hat{\alpha}-\hat{\beta})^2}(-B\hat{w} + \hat{\beta}e) = 0 \\
e'\bar{u} = e'\bar{v} \\
\bar{u} \geq 0, \bar{v} \geq 0.
\end{array} \tag{27}$$

Thus the optimality conditions of Problem (24) are also satisfied. □

The dual problem of the Generalized Optimal Plane is constructed as follows [33, 5]:

Theorem A.4 (Dual of generalized optimal plane) *The dual of problem*

$$\begin{array}{ll}
\min\limits_{w,y,z,\alpha,\beta} & (1-\lambda)(\frac{1}{m}e'y + \frac{1}{k}e'z) + \frac{\lambda}{2}w'w \\
s.t. & Aw - \alpha e + y \geq 0 \\
 & -Bw + \beta e + z \geq 0 \\
 & \alpha - \beta = 2 \\
 & y \geq 0 \quad z \geq 0
\end{array} \tag{28}$$

is

$$\begin{array}{ll}
\min\limits_{u,v,\delta} & \frac{1}{2\lambda}\left\|A'u - B'v\right\|^2 - 2\delta \\
s.t. & e'u = e'v = \delta \\
 & \frac{1-\lambda}{m}e \geq u \geq 0 \quad \frac{1-\lambda}{k}e \geq v \geq 0
\end{array} \tag{29}$$

Proof. Using multipliers u, v, s, t, and δ, the dual problem is

$$\begin{array}{ll}
\max\limits_{w,y,z,\alpha,\beta,u,v,s,t,\delta} & (1-\lambda)(\frac{1}{m}e'y + \frac{1}{k}e'z) + \frac{\lambda}{2}w'w \\
 & -u'(Aw - \alpha e + y) - v'(-Bw + \beta e + z) \\
 & -\delta(\alpha - \beta - 2) - s'y - t'z \\
 & \\
s.t. & \lambda w - A'u + B'v = 0 \\
 & e'u = e'v = \delta \\
 & \frac{1-\lambda}{m}e - u = s \geq 0 \\
 & \frac{1-\lambda}{k}e - v = t \geq 0 \\
 & u \geq 0 \quad v \geq 0
\end{array} \tag{30}$$

Substituting in $w = \frac{A'u - B'v}{\lambda}$ yields

$$\begin{array}{lll} \max\limits_{y,z,\alpha,\beta,u,v,\delta} & (1-\lambda)(\frac{1}{m}e'y + \frac{1}{k}e'z) + \frac{1}{2\lambda}(u'A - v'B)(A'u - B'v) & \\ & -\frac{1}{\lambda}u'A(A'u - B'v) + \alpha e'u & \\ & -u'y + \frac{1}{\lambda}v'B(A'u - B'v) + \beta e'v - z'v & \\ & -\delta(\alpha - \beta - 2) - \frac{1-\lambda}{m}e'y + u'y - \frac{1-\lambda}{k}e'z + v'z & (31) \\ & & \\ s.t. & e'u = e'v = \delta & \\ & \frac{1-\lambda}{m}e \geq u \geq 0 & \\ & \frac{1-\lambda}{k}e \geq v \geq 0 & \end{array}$$

Finally we use the fact that $(u'A - v'B)(A'u - B'v) = \|A'u - B'v\|^2$ to get

$$\begin{array}{ll} \max\limits_{u,v,\delta} & -\frac{1}{2\lambda}\|A'u - B'v\|^2 + 2\delta \\ s.t. & e'u = e'v = \delta \\ & \frac{1-\lambda}{m}e \geq u \geq 0 \\ & \frac{1-\lambda}{k}e \geq v \geq 0 \end{array} \qquad (32)$$

□

References

[1] K. P. Bennett. Decision tree construction via linear programming. In M. Evans, editor, *Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference*, pages 97–101, Utica, Illinois, 1992.

[2] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[3] K. P. Bennett and O. L. Mangasarian. Bilinear separation of two sets in n-space. *Computational Optimization and Applications*, 2:207–227, 1993.

[4] C. E. Brodley and P. E. Utgoff. Multivariate decision trees. *Machine Learning*, 19(1):45–77, 1995.

[5] C. Cortes and V. N. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[6] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J. Schmid, S. Sandhu, K. Guppy, S. Lee, and V. Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology*, 64:304–310, 1989.

[7] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, 1973.

[8] H. Edelsbrunner. Computing the extreme distances between two convex polytopes. *Journal of Algorithms*, 6:213–224, 1985.

[9] K. Fukunaga. *Statistical Pattern Recognition.* Academic Press, New York, 1990.

[10] S. Gallant. Optimal linear discriminants. In *Proceedings of the International Conference on Pattern Recognition*, pages 849–852. IEEE Computer Society Press, 1986.

[11] F. Glover. Improved linear programming models for discriminant analysis. *Decision Sciences*, 21:771–785, 1990.

[12] R.P. Gorman and T.J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89, 1988.

[13] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City, California, 1991.

[14] O. L. Mangasarian. Linear and nonlinear separation of patterns by linear programming. *Operations Research*, 13:444–452, 1965.

[15] O. L. Mangasarian. Multi-surface method of pattern separation. *IEEE Transactions on Information Theory*, IT-14:801–807, 1968.

[16] O. L. Mangasarian. *Nonlinear Programming.* McGraw–Hill, New York, 1969.

[17] O.L. Mangasarian. Mathematical programming in neural networks. *ORSA Journal on Computing*, 5(4):349–360, 1993.

[18] O.L. Mangasarian and R.R. Meyer. Nonlinear perturbation of linear programs. *SIAM Journal on Control and Optimization*, 17(6):745–752, November 1979.

[19] O.L. Mangasarian, R. Setiono, and W.H. Wolberg. Pattern recognition via linear programming: Theory and application to medical diagnosis. In *Proceedings of the Workshop on Large-Scale Numerical Optimization, 1989*, pages 22–31, Philadelphia, Pennsylvania, 1990. SIAM.

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

[21] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry.* MIT Press, Cambridge, Massachusetts, 1969.

[22] J. J. Moré and S. J. Wright. *Optimization Software Guide.* SIAM, Philadelphia, 1993.

[23] T. S. Motzkin and I. J. Schoenberg. The relaxation method for linear inequalities. *Canadian Journal of Mathematics*, 6:393–404, 1954.

[24] P.M. Murphy and D.W. Aha. UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, California, 1992.

[25] B.A. Murtagh and M.A. Saunders. MINOS 5.4 user's guide. Technical Report SOL 83.20, Stanford University, 1993.

[26] S. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

[27] F. Rosenblatt. The perceptron–a perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory, Itahca, New York, January 1957.

[28] A. Roy, S. Govil, and R. Miranda. An algorithm to generate radial basis function (rbf)-like nets for classification problems. *Neural Networks*, 8(2):179–202, 1995.

[29] A. Roy, L. S. Kim, and S. Mukhopadhyay. A polynomial time algorithm for the construction and training of a class of multilayer perceptrons. *Neural Networks*, 6:535–545, 1993.

[30] J.C. Schlimmer. *Concept acquisition through representational adjustment.* PhD thesis, Department of Information and Computer Science, University of California, Irvine, CA, 1987.

[31] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, 36:111–147, 1974.

[32] W.N. Street, W.H. Wolberg, and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology*, volume 1905, pages 861–870, San Jose, California, 1993.

[33] V. N. Vapnik. *The Nature of Statistical Learning Theory.* John Wiley & Sons, New York, 1996.

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

[35] W.H. Wolberg, W. N. Street, and O.L. Mangasarian. Image analysis and machine learning applied to breast cancer diagnosis and prognosis. *Quantitative Cytology and Histology*, 17(2):77–87, 1995.