Efficient Optimal Linear Boosting of A Pair of Classifiers

Victor Boyarshinov Malik Magdon-Ismail

Dept. Computer Science, RPI, Troy, NY, USA.

{boyarv,magdon}@cs.rpi.edu

Abstract

Boosting is a *meta-learning algorithm* which takes as input a set of classifiers and combines these classifiers to obtain a better classifier. We consider the problem of efficiently and optimally boosting a pair of classifiers by reducing this problem to that of constructing the optimal linear separator for two sets of points in 2 dimensions. Specifically, let each point $\mathbf{z}$ be assigned a weight $W(\mathbf{z}) > 0$, where the weighting function can be an arbitrary positive function. We give efficient (low order polynomial-time) algorithms for constructing an *optimal* linear "separator" ℓ defined as follows. Let Q be the set of points mis-classified by ℓ. Then the weight of Q, defined as the sum of the weights of the points in Q, is minimized. If $W(\mathbf{z}) = 1$ for all points, then the resulting separator minimizes (exactly) the mis-classification error. Without an increase in computational complexity, our algorithm can be extended to output the leave-one-out error, an unbiased estimate of the expected performance of the resulting boosted classifier.

1 Introduction

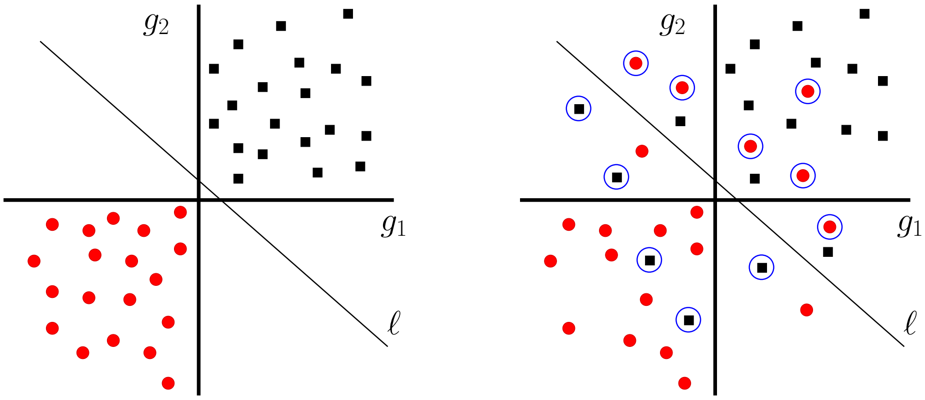
Model aggregation (for example boosting and bagging) are a well known techniques for enhancing the statistical performance of a set of *weak* classifiers to obtain a *stronger* classifier, i.e., one with better generalization performance [16, 18, 4, 1, 13]. Our main focus in this paper is to address some algorithmic issues of boosting. Specifically, we consider the problem of boosting a pair of classifiers: given two classifiers, what is the *optimal* linear combination of this pair of classifiers? To make these statements more precise, lets introduce some notation. The training set $\mathcal{D} = \{\mathbf{z}_i, y_i\}_{i=1}^n$ is a collection of data points where $\mathbf{z}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. A weighting function $W : \mathbb{R}^d \mapsto \mathbb{R}^+$ specifies a penalty for misclassifying a data point; let $w_i = W(\mathbf{z}_i)$. Let $g_1, g_2 : \mathbb{R}^d \mapsto \mathbb{R}$ be a pair of classification functions, where the corresponding classifiers would be given by $sign(g_1), sign(g_2) : \mathbb{R}^d \mapsto \{-1, +1\}$.

Definition 1.1 *A (linearly) boosted classification function* $g(\mathbf{z})$*has the form*

$$g(\mathbf{z}) = w_0 + w_1 g_1(\mathbf{z}) + w_2 g_2(\mathbf{z}).$$

The corresponding classifier is $sign(g(\mathbf{z})) = sign(w_0 + w_1 g_1(\mathbf{z}) + w_2 g_2(\mathbf{z}))$.

g_1 and g_2 can be viewed as defining a new two dimensional feature space. Each datapoint $\mathbf{z}_i \in \mathbb{R}^d$ can be mapped onto this two dimensional feature space by $\{\mathbf{z}_i, y_i\} \mapsto \{\left[\begin{smallmatrix} g_1(\mathbf{z}) \\ g_2(\mathbf{z}) \end{smallmatrix}\right], y_i\}$. A linearly boosted classifier corresponds exactly to a linear separator ℓ in this two dimensional space. We illustrate with two examples below (black squares are data points from class $\{+1\}$ and red circles are from class $\{-1\}$).



(a) Ideal individual classifiers. (b) Individual classifiers make errors.

In (a), we illustrate the "ideal" situation in which both classifiers individually make no errors. While there is no benefit from boostinig on the training set, there may still be benefit, from the statistical perspective, to optimal boosting. In (b), we illustrate the more general case in which the classifiers may make errors. A line ℓ (shown) represents one particular way in which to boost the classifiers. The training points misclassified by the boosted classifier are circled. The essential point is that there is an equivalence between the set of linearly boosted classifiers for g_1, g_2 and the set of "separating" lines ℓ in the two-dimensional feature space. We can thus rephrase our entire discussion regarding the optimal linear boosted classifier for g_1, g_2 (for whatever optimal means) in terms of its corresponding optimal linear classifier in the two dimensional space.

Optimal Linear Separation in $\mathbb{R}^2$. While our specific concern here is optimal boosting of a pair of classifiers, optimal linear separation plays a key role in other areas of pattern recognition and computational geometry. In statistical pattern recognition, a resurgence of linear separability has emerged through its role in Support Vector Machines and other kernel based methods [9, 19]. In computational geometry, determining the intersection of sets (mostly in two or three dimensions) is of immense practical importance, for example in CAD systems. Thus efficient optimal linear separation is a fundamental tool that may be of use outside our setting.

Our goal here is to develop efficient algorithms for *exact* linear separation, where we use "exact" to mean globally optimal with respect to the (arbitrarily specified) positive error criterion W. We emphasize that our algorithms are applicable to the case where the data points are *not* linearly separable.

We will use the notation $\mathbf{x}$ to refer to the two dimensional projection of $\mathbf{z}$ into the feature space given by $\begin{bmatrix} g_1(\mathbf{z}) \\ g_2(\mathbf{z}) \end{bmatrix}$. The data $\mathcal{D}$ is naturally partitioned into two sets $\mathcal{A}, \mathcal{B}$ corresponding to the positive and negative data points. Specifically, $\mathbf{z}_i \in \mathcal{A}$ ($\mathbf{z}_i \in \mathcal{B}$) *iff* $y_i = +1$ ($y_i = -1$).

Definition 1.2 *Two sets* $\mathcal{A}$, $\mathcal{B}$ *are* linearly separable *iff* $\exists \mathbf{v},\ v_0$ *such that* $\mathbf{v}^T\mathbf{x} + v_0 > 0,\quad \forall \mathbf{x} \in \mathcal{A}$ *and* $\mathbf{v}^T\mathbf{x} + v_0 < 0,\quad \forall \mathbf{x} \in \mathcal{B}$. *The pair* $(\mathbf{v}, v_0)$ *defines an* (oriented) separating hyperplane*.*

If two sets, $\mathcal{A}, \mathcal{B}$, are linearly separable, the *margin* of a separating hyperplane ℓ is the minimum distance of a data point to the hyperplane. The *maximum margin* separating hyperplane is a separating hyperplane with maximum possible margin. We define the optimal separator with respect to the weighting function W.

For hyperplane $\ell = (\mathbf{v}, v_0)$, let $\mathcal{Q}(\ell) = \mathcal{Q}_{\mathcal{A}}(\ell) \cup \mathcal{Q}_{\mathcal{B}}(\ell)$ denote the set of misclassified points, where $\mathcal{Q}_{\mathcal{A}}(\ell) = \{\mathbf{x} \in \mathcal{A} | \mathbf{v}^T\mathbf{x} + v_0 \leq 0\}$ and $\mathcal{Q}_{\mathcal{B}}(\ell) = \{\mathbf{x} \in \mathcal{B} | \mathbf{v}^T\mathbf{x} + v_0 \geq 0\}$ (note that we take the points on ℓ as being misclassified). The hyperplane ℓ is denoted a representative hyperplane for $\mathcal{Q}(\ell)$. The weight (or error) $\mathcal{E}(\ell)$ for ℓ is the total weight summed over the points misclassified by ℓ,

$$\mathcal{E}(\ell) = \sum_{i:\mathbf{x}_i \in \mathcal{Q}(\ell)} w_i.$$

Note that the sets $\mathcal{A}'(\ell) = \mathcal{A} \setminus \mathcal{Q}_{\mathcal{A}}(\ell)$ and $\mathcal{B}'(\ell) = \mathcal{B} \setminus \mathcal{Q}_{\mathcal{B}}(\ell)$ are linearly separable, and ℓ is a separator for them. Similarily, $\mathcal{Q}_{\mathcal{A}}(\ell)$ and $\mathcal{Q}_{\mathcal{B}}(\ell)$ are also linearly separable and ℓ is a separator for them as well. A separator ℓ^* is *optimal* if it has minimum weight, i.e., for any other separator ℓ, $\mathcal{E}(\ell^*) \leq \mathcal{E}(\ell)$.

Definition 1.3 (Optimal Weight Fat Separator) *A hyperplane* $\ell = (\mathbf{v}, v_0)$ *is an optimal weight fat separator for* $\mathcal{A}, \mathcal{B}$ *if it is optimal and is also a maximum margin separator for* $\mathcal{A}'(\ell)$ *and* $\mathcal{B}'(\ell)$.

Intuitively, the optimal weight fat separator ℓ is the hyperplane with minimum error such that if the misclassified points are viewed as noisy and are removed, then the entire set becomes separable, and ℓ is a maximum margin separator for the "noise-corrected" set. We could analogously define the optimal fat separator with respect to the new separable set that would be obtained if instead of removing the misclassified points, we flip the classes of these points – all our results apply here as well.

Our Contribution. We consider optimal boosting of a pair of classifiers, where we define the optimal boosted classifier as the optimal fat separator in the two dimensional feature space $\mathbf{x}$. An exponential algorithm to solve this problem results from removing every possible subset of the data and testing for separability (as the number of possible subsets is exponential). We first show how to significantly improve the efficiency by reducing the problem to considering only the separators ℓ passing through every pair of points ($O(n^2)$), which results in an $O(n^3)$ algorithm. By more carefully enumerating these separators, we finally establish the following theorem.

Theorem 1.4 *The optimal fat separator for the two sets $\mathcal{A}, \mathcal{B} \in \mathbb{R}^2$ with $|\mathcal{A}| \leq |\mathcal{B}|$, can be constructed in time $O(|\mathcal{A}| n \log n)$, where $n = |\mathcal{A} \cup \mathcal{B}|$.*

Note that the algorithm is exact, i.e. outputs a globally optimal solution, applies to non-separable sets $\mathcal{A}, \mathcal{B}$, and the weight function W can be an arbitrary positive function. In particular, if $W(\mathbf{x}) = 1$, then the resulting separator minimizes the classification error, i.e., obtains the linearly boosted classifier with minimum classification error. An important problem in computational geometry is set intersection, or intersection of solid objects. Our algorithm can be applied in this setting with $W(\mathbf{x}) = 1$ to obtain the minimum sized intersection for two objects represented as sets of points.

The leave-one-out (or cross-validation) error is an important unbiased measure of the out of sample (generalization) performance of any classifier. In particular, one would like to obtain the expected generalization performance for the resulting boosted classifier. The typical computation of this error would involve removing a data point $\mathbf{x}_L$, computing the optimal boosted classifier on $\mathcal{D} \setminus \mathbf{x}_L$, and evaluating its leave-one-out performance on $\mathbf{x}_L$. This leave-one-out performance can be averaged over every data point left out to give a lower variance estimate of the expected out of sample error. Since the entire optimal boosting has to be performed on n data sets of size $n-1$, the resulting computation adds a factor of n to the computational complexity. We show how to extend our algorithm to obtain the leave-one-out error with no increase in the asymptotic computational complexity. Specifically, we establish the following theorem.

Theorem 1.5 *For the two sets $\mathcal{A}, \mathcal{B} \in \mathbb{R}^2$, the leave-one-out error for optimal boosting can be computed in time $O(n^2 \log n)$, where $n = |\mathcal{A} \cup \mathcal{B}|$.*

All our proofs are constructive, and can hence be converted to algorithms without much difficulty.

Interpreting the Weight Function W. The most natural interpretation of the weight function W is as a user defined risk metric which specifies the penalty for misclassifying inputs (which may be a function of the actual input). The optimal boosted calssifier then corresponds to minimizing the empirical risk.

We briefly digress to give a probabilistic interpretation of weight function W. Let p_i be the probability that the class of the i^{th} data point is correct; p_i is a measure of the noise in the data point y_i. Assume, without loss of generality, that $p_i > \frac{1}{2}$ (if $p_i < \frac{1}{2}$, we could simply flip the value of y_i, and if $p_i = \frac{1}{2}$, the data point conveys no useful information and can be discarded). Assume that the classes of the data points are selected independently, and let $\mathcal{Q}$ be the set of points misclassified by a particular classifier g. We can compute the likelihood of g as

$$l(\mathcal{D}|g) = \prod_{i:\mathbf{x}_i \in \mathcal{D}\setminus\mathcal{Q}} p_i \prod_{i:\mathbf{x}_i \in \mathcal{Q}} (1 - p_i).$$

Taking the logarithm of both sides and collecting terms, we find that

$$\log l(\mathcal{D}|g) = W - \sum_{i:\mathbf{x}_i \in \mathcal{Q}} \hat{w}_i,$$

where $W = \sum_i \log p_i$ is a constant independent of g and $\hat{w}_i = \log \frac{p_i}{1-p_i}$. Thus, computing the maximum likelihood classifier corresponds exactly to computing the minimum weight classifier where the weights are given by $\hat{w}_i$. These weights are legitimate, i.e. $\hat{w}_i \geq 0$, because $p_i > \frac{1}{2}$.

Paper Outline. Next we discuss some related work, then we give our optimal boosting algorithms in Section 2. We discuss the leave-one-out error computation in Section 3, and we end with some concluding remarks in section 4.

Related Work.

The convex hull of a set of points is the smallest convex set that contains the points. The convex hull is a fundamental construction in mathematics and computational geometry, and convex hull operations play an important role in linear separability, because two sets are linearly separable *iff* their convex hulls are linearly separable. Chan [5] presented the state-of-the-art output sensitive algorithm for computing the convex hull in 2 or 3 dimensions:

Fact 1.6 ([5]) *The convex hull in 2 or 3 dimensions can be computed in $O(n \log h)$ operations, where h is the size of the output, and n is the number of points.*

When two sets of points are separable, an approach to constructing the maximum margin separator is to first construct the convex hulls, and then construct the maximum margin separator for the convex hulls. In 2 and 3 dimensions, this approach is very efficient. The maximum margin separator can be specified as the orthogonal bisector of the line joining two points on the convex hulls of the two sets. These two points are sometimes refered to as a *realization* of the maximum margin separator (also referred to as support vectors). Dobkin and Kirkpatrick [11] introduced hierarchical representations for convex hulls and established many useful properties of such representations. Specifically, given a standard representation of a convex hull (in 2 or 3 dimensions), a compact hierarchical representation of can be constructed in *linear* time. This representation has been exploited in a series of subsequent papers ([11], [12]). In particular, they construct a *sublinear* deterministic algorithm for obtaining the maximum margin separator for separable convex hulls in 2 and 3 dimensions (given the hierarchical representation for both convex hulls). Using the linear algorithm for constructing the hierarchical representations combined with Fact 1.6, one obtains an efficient deterministic algorithm for constructing the maximum margin separator for *separable* sets in 2 and 3 dimensions:

Fact 1.7 ([6],[11]) *The maximum margin separator (in 2 and 3 dimensions), and its realization, for two separable sets $\mathcal{A}$ and $\mathcal{B}$ can be found in $O(n \log n)$ operations.*

Generalizing to $d > 3$ dimensions is difficult, and a more popular approach is to re-formulate the linear separability problem as a linear program or the maximum margin separator problem as a quadratic program. Such problems can be handled using linear/convex programming techniques such as: the simplex method [7], with complexity $O(N^2)$ where the constant is exponential in d (in practice the simplex method has linear average-case complexity); or, interior point methods [10, 14, 15, 17].

Our work addresses the case when $\mathcal{A}$ and $\mathcal{B}$ are not linearly separable (i.e., their convex hulls intersect). Common approaches are to formulate some differentiable error as a function of the distance of a misclassified point from the hyperplane. One then seeks to minimize some heuristic function of this error, [19]. If the resulting error function is convex, then convex optimization techniques can be brought to bear, for example in [19] one obtains a convex quadratic program. Bennet and Mangasarian [2] propose minimizing the average distance from misclassified points using linear programming. Most often, however, such heuristic errors are minimized using iterative algorithms. Another approach, suggested in [3], is to solve the problem using linear programming on the reduced convex hulls (contracted convex hulls obtained by placing an upper bound on the multiplier in the convex combination for each point).

In contrast to these existing approaches, our work focuses on producing globally optimal solutions: for an arbitrary weight function W, the problem cannot be represented as the minimization of some differentiable error (convex or not). We output a minimum weight subset of the points $\mathcal{Q}$ such that after deleting these points, the remaining points are separable, and the algorithms given by Fact 1.7 can then be used. Alternatively, if the points in $\mathcal{Q}$ have their classes flipped, then once again, the algorithms in Fact 1.7 can be brought to bear.

2 Optimal Fat Separators in 2 Dimensions

The goal in this section is to prove Theorem 1.4. Let $\mathcal{A} = \{\mathbf{a}_i\}_{i=1}^m$ and $\mathcal{B} = \{\mathbf{b}_j\}_{j=1}^k$ be two sets of points, with $m \le k$, and let $n = m + k$. It is traditional to assign class $+1$ to one of the sets (say) $\mathcal{A}$ and -1 to the other. Every point $\mathbf{x}$ has a weight $W(\mathbf{x}) > 0$.

Definition 2.1 *A* separator set $\mathcal{Q} \subseteq \mathcal{A} \cup \mathcal{B}$ *is a set with the following property: if the points in* $\mathcal{Q}$ *are deleted, the remaining points are linearly separable.*

Every separator set $\mathcal{Q}$ has a weight, $W(\mathcal{Q}) = \sum_{\mathbf{x} \in \mathcal{Q}} W(\mathbf{x})$. An *optimal separator set* $\mathcal{Q}^*$ is one with minimum weight, i.e., for any other separator set $\mathcal{Q}$, $W(\mathcal{Q}) \ge W(\mathcal{Q}^*)$. The next lemma provides a correspondence between optimal separator sets and hyperplanes. As already discussed, to every oriented line ℓ, we can associate the separator set $Q(\ell)$ which is separable by ℓ. The converse is also true for an optimal separator set.

Lemma 2.2 *Let* $\mathcal{Q}^*$ *be any optimal separator set. Every hyperplane* ℓ *that separates* $(\mathcal{A} \cup \mathcal{B}) \setminus \mathcal{Q}^*$ *also separates* $\mathcal{Q}^*$*, so* $\mathcal{Q}^*$ *is also separable; further,* $\mathcal{Q}(\ell) = \mathcal{Q}^*$*.*

Proof: Let $\mathcal{Q}^*$ be an optimal separator set. Then $\mathcal{A}' = \mathcal{A} \setminus \mathcal{Q}^*$ and $\mathcal{B}' = \mathcal{B} \setminus \mathcal{Q}^*$ are linearly separable, so let ℓ be any separating hyperplane. First assume that no point of Q^* lies on ℓ. Then every point in $\mathcal{Q}^*$ must be misclassified, for if $\mathbf{x} \in Q^*$ is correctly classified, then $\mathcal{Q}^* \setminus \mathbf{x}$ is a separator set with smaller weight, contradicting the optimality of $\mathcal{Q}^*$. Now suppose that some points of $\mathcal{Q}^*$ lie on ℓ, then, since ℓ strictly separates $\mathcal{A}', \mathcal{B}'$, there exists a small enough shift of ℓ which will still separate $\mathcal{A}'$ and $\mathcal{B}'$. This shift can be chosen so as to classify at least one point $\mathbf{x} \in \mathcal{Q}^*$ correctly, which means that $\mathcal{Q}^* \setminus \mathbf{x}$ is a separator set with smaller weight, contradicting the optimality of Q^*.
■

For an optimal separator set $\mathcal{Q}^*$, by Lemma 2.2, *any* hyperplane ℓ that separates $(\mathcal{A} \cup \mathcal{B}) \setminus \mathcal{Q}^*$ *induces* $\mathcal{Q}^*$, i.e. $\mathcal{Q}(\ell) = \mathcal{Q}^*$. In particular, the optimal fat (maximum margin) separator for $(\mathcal{A} \cup \mathcal{B}) \setminus \mathcal{Q}^*$ will have minimum weight. Thus, once we have found an optimal separator set, we can easily construct an optimal fat separator using the result in Fact 1.7. To prove Theorem 1.4, it therefore suffices to give an efficient algorithm to compute an optimal separator set. A brute force search through all subsets of $\mathcal{A} \cup \mathcal{B}$ is clearly an exponential algorithm. We present here a polynomial time algorithm to find an optimal separator set, which will establish Theorem 1.4.

The general idea follows from Lemma 2.2 which implies that any optimal separator set is the separator set of some hyperplane. Thus, it suffices to consider all possible hyperplanes, and their separator sets. Though it appears that we have increased the difficulty of our enumeration problem, we will now show that not all possible hyperplanes need be considered. In fact, we can restrict ourself to hyperplanes passing through at least two points. This is a big step, because there are only $\Theta(n^2)$ such hyperplanes. The separator set for a given hyperplane can be computed in $O(n)$ operations, and so we immediately have an $O(n^3)$ algorithm. By being careful about reusing computations, we can reduce the complexity to $O(n^2 \log n)$, which is the content of the next theorem.

Theorem 2.3 (Optimal Separator Set) *An optimal separator set $\mathcal{Q}(\mathcal{A}, \mathcal{B})$ can be found in $O(mn \log n)$ time.*

Proof: We need to consider more carefully the definition of a separator set, especially when points lie on the hyperplane ℓ. According to the strict definition of separability, we would need to include all the points on ℓ into $\mathcal{Q}(\ell)$. We relax this condition in the definition of the *positive separator set* associated to the hyperplane ℓ.

Definition 2.4 *For hyperplane ℓ, the positive separator set $\mathcal{Q}^+(\ell)$ contains all misclassified points* **except** *the positive points (in $\mathcal{A}$) that lie on ℓ. ℓ is denoted the positive separator hyperplane of $\mathcal{Q}^+(\ell)$.*

For a hyperplane ℓ, the only difference between the usual separator set and the positive separator set is in how we treat the points that reside directly on the hyperplane ($\mathcal{Q}^+(\ell) \subseteq \mathcal{Q}(\ell)$). The next lemma which shows a correspondence between optimal separator sets and positive separator sets will be useful in the our proof.

Lemma 2.5 *Let $\mathcal{Q}^*$ be any optimal separator set. Then there exists a hyperplane ℓ such that $\mathcal{Q}^+(\ell) = Q^*$ and either:*

i. two or more positive points from $\mathcal{A}$ reside on ℓ;

ii. exactly one positive point from the $\mathcal{A}$ resides on ℓ, and no others.

Proof: Let $\mathcal{Q}^*$ be an optimal separator set, and let ℓ' be a hyperplane that separates $\mathcal{A}'$ and $\mathcal{B}'$ constructed from $\mathcal{A} \cup \mathcal{B} \setminus \mathcal{Q}^*$. By Lemma 2.2 $\mathcal{Q}(\ell') = \mathcal{Q}^*$ and ℓ' separates $\mathcal{Q}^*$. Let $\mathbf{a}^+$ be the closest positive point in $\mathcal{A}'$ to ℓ. Then all hyperplanes ℓ'' parallel to ℓ' that are closer to $\mathbf{a}^+$ and correctly classify $\mathbf{a}^+$ also separate $\mathcal{A}'$ and $\mathcal{B}'$. Hence, by Lemma 2.2 all such hyperplanes also separate $\mathcal{Q}^*$, i.e., for all such hyperplanes ℓ'', $\mathcal{Q}(\ell'') = \mathcal{Q}^*$. This means there are no points that are on any of these hyperplanes ℓ''. Now let ℓ be the hyperplane parallel to ℓ' and containing $\mathbf{a}^+$, and consider $\mathcal{Q}^+(\ell)$. Any negative points on ℓ belong to $\mathcal{Q}^+(\ell)$, but they already belonged to Q^*. Any positive points on ℓ do not belong to $\mathcal{Q}^+(\ell)$, and they also did not belong to Q^*. Thus, $\mathcal{Q}^+(\ell) = \mathcal{Q}^*$. If ℓ

contains exactly one positive point and no others, then we are done. Suppose that ℓ contains at least one negative point and no other positive points (other than $\mathbf{a}^+$). Then, there exists a small enough rotation of ℓ about $\mathbf{a}^+$ which still separates $\mathcal{A}'$ and $\mathcal{B}'$ but will classify at least one negative point $\mathbf{b}^- \in \mathcal{Q}^*$ correctly. This means that $\mathcal{Q}^* \setminus \mathbf{b}^-$ is a separator set with smaller weight that $\mathcal{Q}^*$, contradicting the optimality of $\mathcal{Q}^*$. Thus, if there is only one positive point on ℓ, there can be no other points. The only other case is that there are two or more positive points on ℓ, concluding the proof. $\square$

Lemma 2.5 shows that it suffices to consider only the positive separator sets of hyperplanes that pass through at least one positive point. This is the starting point of the algorithm. We try every positive point as a candidate "central" point and compute the best possible separator set for all hyperplanes that pass through this central point. We then keep the best separator set over all possible candidate central points. Since there are m possible candidate central points, the computational complexity will be the time it takes to compute the best separator set for all hyperplanes passing through a candidate central point multiplied by m. The next lemma gives a constructive algorithm to compute this best separator set.

Lemma 2.6 *The optimal positive separator set over all hyperplanes passing through a candidate central point can be computed in $O(n \log n)$.*

Proof: Let's consider how to efficiently find the best positive separator hyperplane that contains some fixed positive point $\mathbf{a}^+$. In order to do so efficiently, we introduce a *mirrored-radial* coordinate system in which all the points except $\mathbf{a}^+$ can be linearly ordered with respect to $\mathbf{a}^+$.

We start with an arbitrary (base) vector $\mathbf{u}$ that defines an axis as shown in Figure 1. The origin of $\mathbf{u}$ is at $\mathbf{a}^+$. With $\mathbf{a}^+$ as origin, we define the angle $\theta(\mathbf{x})$ of a point $\mathbf{x}$ with respect to the base vector $\mathbf{u}$ as the angle between the two vectors $\mathbf{x} - \mathbf{a}^+$ and $\mathbf{u}$. The upper hemisphere of the unit circle is the set of points on the unit circle with angle in the range $[0, \pi]$ (shaded in Figure 1). We define the mirrored-radial projection of a point $\mathbf{s}(\mathbf{x})$ as the projection of $\mathbf{x}$ onto the upper hemisphere of the unit circle, through the origin $\mathbf{a}^+$. The mirrored-radial projections of $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are illustrated by $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3$ in the Figure 1). The mirrored-radial coordinate $\theta(\mathbf{x})$ is then the angle of $\mathbf{s}(\mathbf{x})$, i.e., $\theta(\mathbf{s}(\mathbf{x}))$. Notice that many points may have the same mirrored-radial projection, in which case, they all have the same mirrored-radial coordinate.

Suppose that for some (arbitrary) choice of $\mathbf{u}$, the mirrored radial coordinates of all the points (except $\mathbf{a}^+$) have been sorted. Thus, $0 < \theta_1 \leq \theta_2 \cdots \leq \theta_{n-1} < \pi$. For convenience, define $\theta_0 = 0$ and $\theta_n = \pi$. An oriented hyperplane ℓ can also be uniquely specified by giving its angle θ_ℓ (see Figure 1), together with its orientation (± 1). For a given orientation, all hyperplanes with $\theta_\ell \in (\theta_i, \theta_{i+1})$,

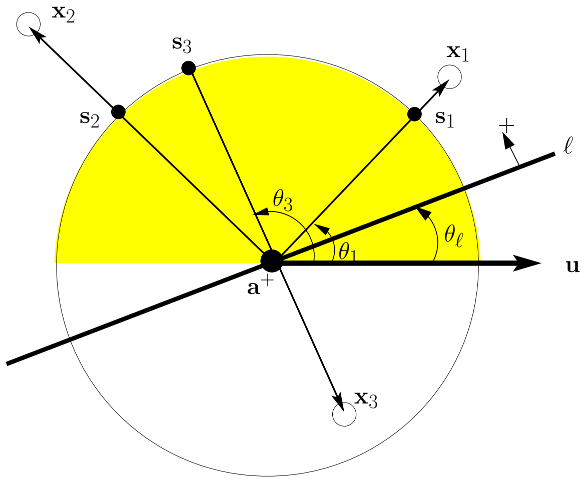


Figure 1: Mirrored-radial coordinates.

$0 \leq i < n$, partition the points into the same two sets, and hence have the same positive separator set. The other possible values for θ_ℓ are the actual mirrored-radial coordinates θ_i. Since there are only two possible orientation for a given θ_ℓ, we have the following lemma

Lemma 2.7 *There are at most $4n-2$ possible equivalence classes of positive separator hyperplanes, corresponding to the following ranges for for θ_ℓ,*

$$\{(\theta_0, \theta_1), \theta_1, (\theta_1, \theta_2), \theta_2, \ldots, \theta_{n-1}, (\theta_{n-1}, \theta_n)\}.$$

For any two values of θ_ℓ from the same range, and for a given orientation, the positive separator sets are identical.

The importance of Lemma 2.7 is that we now only have to check one representative from each equivalence class. Further, this can be done very efficiently with two linear time scans (one for each orientation) as follows. Lets consider hyperplanes with orientation $+1$ as shown in Figure 1 (the same argument applies to orientation -1). First consider the line ℓ with $\theta_\ell = \theta_0$, and we compute $\mathcal{Q}^+(\ell)$ and $W(\mathcal{Q}^+(\ell))$. This takes linear time $O(n)$. Now, we sequentially step through the hyperplane equivalence classes. Each time we move from one equivalence class to the next, some points will enter $\mathcal{Q}^+(\ell)$ and some points will leave $\mathcal{Q}^+(\ell)$, and correspondingly $W(\mathcal{Q}^+(\ell))$ must be updated.

(i) Move from equivalence class (θ_i, θ_{i+1}) to θ_i: All positive points with coordinate θ_{i+1} are removed from $\mathcal{Q}^+(\ell)$ (if they previously belonged to $\mathcal{Q}^+(\ell)$) and $W(\mathcal{Q}^+(\ell))$ is updated. All negative points with coordinate θ_{i+1} are added to $\mathcal{Q}^+(\ell)$ (if they are not already in $\mathcal{Q}^+(\ell)$)

and $W(\mathcal{Q}^+(\ell))$ is updated. This process only requires a single scan through all the points with coordinate θ_{i+1} to determine their sign.

(ii) Move from equivalence class θ_i to (θ_i, θ_{i+1}): All positive points with coordinate θ_i that now become misclassified are added to $\mathcal{Q}^+(\ell)$ and $W(\mathcal{Q}^+(\ell))$ is updated. All negative points with coordinate θ_i that now become correctly classified are removed from $\mathcal{Q}^+(\ell)$ and $W(\mathcal{Q}^+(\ell))$ is updated. This process also only requires a single scan through all the points with coordinate θ_i to determine their sign and if they are now misclassified or not.

In order to efficiently implement these updates, we store the current positive separator set $\mathcal{Q}^+(\ell)$ in an array q of size n. We set $q[k] = 1$ *iff* $\mathbf{x}_k \in \mathcal{Q}^+(\ell)$. When we process one of the changes in the hyperplane equivalence class described above, we process all the points with some value θ_i for their mirrored radial coordinate. Suppose that there are n_i such points. The status (misclassified or not) for these points for the new hyperplane θ_ℓ can be tested in constant time per point. Some (or all) of these points will then change their set membership in $\mathcal{Q}^+(\ell)$ which can be updated in constant time in q, and simultaneously, $W(\mathcal{Q}^+(\ell))$ is updated in $O(1)$ operations per point. Each time $W(\mathcal{Q}^+(\ell))$ is updated, in $O(1)$ operations, we can keep track of the minimum value attained. Thus the total cost of one such move is $O(n_i)$. On each move, only the points corresponding to some mirrored radial coordinate θ_i are processed ($O(n_i)$), each θ_i is processed at most twice, once in the move from (θ_{i-1}, θ_i) to θ_i and once in the move from θ_i to (θ_i, θ_{i+1}). Thus each θ_i contributes $O(2n_i)$ to the computational cost, resulting in a total computational cost of $O(2\sum n_i) = O(n)$ for the scan. This scan is repeated for orientation -1 of the hyperplane, for a total cost $O(n)$. Once the weight of the best separator is computed using the two scans above, an analogous single scan is all that is required to reconstruct the optimal positive separator set itself by performing the scan until the optimal weight positive separator set is reached.

Recap: For every positive point, $\mathbf{a}^+$, we first compute the mirrored-radial coordinates of all the other points, requiring $O(n)$ operations. We then sort these coordinates in $O(n \log n)$ operations. We now make two scans (in sorted order), one for each orientation of the hyperplane, updating the weight of the positive separator set, keeping track of the optimal weight. After the scan, one more scan suffices to construct the optimal weight positive separator set for this central point a^+. These scans are linear time, $O(n)$. Since the sorting operation has the dominant run time, this entire process is in $O(n \log n)$, and it constructs the optimal positive separator set. □

By Lemma 2.5, every optimal separator set is equivalent to a positive separator set passing through at least one positive point. Since the algorithm in Lemma 2.6 considers all such positive separator sets, it must have considered all optimal separator sets. All that remains is to argue that the *optimal* positive separator set is also an optimal separator set. The next lemma will be useful.

Lemma 2.8 *Any positive separator with weight W is also a separator set (with weight W).*

Proof: Consider any positive separator set with weight W, together with its positive separator line ℓ. Let $\mathbf{b}^-$ be the closest point to ℓ on the negative side of ℓ. Make a parallel shift of ℓ to ℓ' by a small enough distance so as not to cross $\mathbf{b}^-$. Now all the positive points on ℓ are correctly classified and all the negative points on ℓ are incorrectly classified. The classifications of no other points have changed, therefore $\mathcal{Q}(\ell') = \mathcal{Q}^+(\ell)$, i.e., $\mathcal{Q}^+(\ell)$ is also a separator set. □

Suppose that the optimal positive separator set with weight W constructed in Lemma 2.6 is not an optimal separator set. Suppose that the weight of an optimal separator set is $W^* < W$. Since every optimal separator set corresponds to some positive separator set, this positive separator set must have been considered in the the optimization over positive separator sets. Further, every positive separator set is also a separator set (Lemma 2.8) therefore $W \leq W^*$, which is a contradiction. Thus the optimal positive separator set constructed is also an optimal separator set. To conclude the proof of Theorem 2.3, we observe that the runtime of the entire algorithm is obtained by multiplying the runtime in Lemma 2.6 by m, since the algorithm in Lemma 2.6 must be run for every candidate central point. ■

2.1 Maximizing The Margin Subject to Minimum Weight

The algorithm in the previous section outputs a separator set $\mathcal{Q}$ which is guaranteed to have optimal weight. After removing this separator set, we may then construct the maximum margin separating hyperplane efficiently using the results in Fact 1.7. Since we know that $\mathcal{Q}$ and $\mathcal{A} \cup \mathcal{B} \setminus \mathcal{Q}$ are simultaneously separable (with opposite orientation) by the same set of hyperplanes (Lemma 2.2), one could instead flip the classes of the points in $\mathcal{Q}$ to obtain an optimal separable set, which can then be fed into the algorithms in Fact 1.7. However, the algorithm does not guarantee that the resulting fat separator has maximum margin among all separators having optimal weight. This issue is illustrated in Figure 2. In the figure, removing (or flipping the class of) the circled discs gives a larger margin boosted classifier than removing (or flipping the class of) the circled squares. A minor modification of the algorithm allows us to break ties optimally with respect to the margin of the classifier, without any increase in asymptotic computational complexity. We make use of the hierarchical convex hull representations developed in [6, 11].

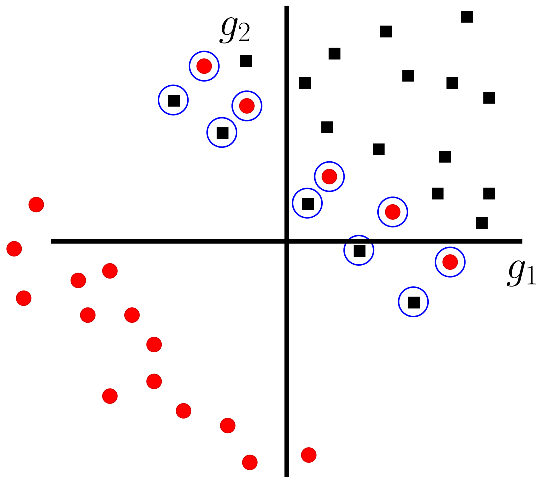


Figure 2: Maximizing the margin among optimal separator sets. The sum of the weights of the circled squares equals that for the circled discs.

Fact 2.9 ([6, 11]) *Given the hierarchical representation of two convex hulls on a total of n points in 2 or 3 dimensions, they can be merged into one hierarchical representation of a convex hull in $O(\log n)$. In particular, a single point may be added to a convex hull in $O(\log n)$.*

Fact 2.10 ([6, 11]) *The 2 or 3 dimensional hierarchical representation of a convex hull can be maintained in such a way that points can be removed from the convex hull in the reverse order in which they were added to the convex hull in time $O(\log n)$ per point.*

Fact 2.11 ([6, 11]) *Given the 2 or 3 dimensional hierarchical representation of a pair of convex hulls, the maximum margin separator (and its weight) can be found in $O(\log n)$.*

These tools are all the operations needed to extend the algorithm given in Lemma 2.6 to keep track of the margin of a separating set in addition to its weight. We will give the details for the case where we are interested in the margin of separation after the separator set is removed (analogous results hold when class of points in the separator set is flipped).

Definition 2.12 *For the set $\mathcal{A} \cup \mathcal{B}$, the margin of separator set $\mathcal{Q}$, $mar(\mathcal{Q})$, is the margin of the optimal fat separator for $\mathcal{A} \cup \mathcal{B} \setminus \mathcal{Q}$.*

Theorem 2.13 *An optimal separator set $\mathcal{Q}^*$ for $\mathcal{A} \cup \mathcal{B}$ can be found in time $O(mn \log n)$ such that for any other separator set $W(\mathcal{Q}^*) \leq W(\mathcal{Q})$, and if $W(\mathcal{Q}^*) = W(\mathcal{Q})$ then $mar(\mathcal{Q}^*) \geq mar(\mathcal{Q})$.*

Proof: The essence of the proof is exactly analogous to the proof of Theorem 2.3, the main difference is Lemma 2.6. In constructing the constructing the optimal weight separator set, we also need to keep track of its margin. The analogous lemma is

Lemma 2.14 *The optimal positive separator set with maximum margin over all hyperplanes passing through a candidate central point can be computed in $O(n \log n)$.*

Proof: We consider the set of hyperplanes in the equivalence classes defined in Lemma 2.7, and we assume that the data points have already been sorted according to their mirrored radial coordinates. The initial hyperplane θ_0 (with positive orientation) defines a separator set $\mathcal{Q}_0$. For this separator set, we construct the positive and negative hierarchical convex hulls on $\mathcal{A}_0, \mathcal{B}_0$, and obtain the maximum margin separator, together with its margin. This can be accomplished in $O(n \log n)$. The positive convex hull consists of all positive points that have un-mirrored radial coordinate in the upper hemisphere, and the negative hull consists of all negative points whose un-mirrored radial coordinate is in the lower hemisphere. We assume that the points in the positive and negative convex hull were added in order of decreasing un-mirrored radial coordinate. These convex hulls, including the maximum margin separator and the margin can be constructed in $O(n \log n)$ (Facts 2.9, 2.11). We now consider the transition from one equivalence class (say θ_i) to the next (say (θ_i, θ_{i+1})). Points get removed from the positive/negative convex hull in order of increasing un-mirrored radial coordinate, which is the reverse order in which they were added. The updated convex hull can be computed in $O(\log n)$. Unfortunately, points may also get added to the convex hulls, which may cause a problem for later removal. Thus we maintain two new positive and negative convex hulls for negative/positive points that need to be added to their respective convex hulls. The full positive (negative) convex hull will be the merged hulls from the two positive (negative) convex hulls, one for the original convex hull with points being removed, and the second for the convex hull with points being added.

By Facts 2.9 and 2.10, these two convex hulls can be maintained using $O(\log n)$ operations per point. Further, the two negative and positive convex hulls can be merged and their maximum margin separator found in $O(\log n)$ (Facts 2.9 and 2.11). Thus, using $O(\log n)$ operations per point, we can update both the weight and the margin of the positive separator set, keeping track of the minimum weight, and among minimum weight positive separator sets, the maximum margin. This constitutes a total of $n \log n$ operations for the entire scan. After this scan is performed, we have the optimal weight and maximum margin corresponding to this optimal weight. A single additional scan suffices to construct the separator set as the one which achieves this optimal weight and margin. $\square$

The remainder of the proof of the theorem follows the same line as the proof of Theorem 2.3, for a total runtime of $O(n \log n)$. ■

3 Leave-one-out error

An important issue in the design of efficient machine learning systems is the estimation of the accuracy of learning algorithms, in particular its sensitivity to noisy inputs. One classical estimator of the generalization performance is the *leave-one-out* error, which is commonly used in practice. Intuitively, the leave-one-out error is defined as the average error obtained by training a classifier on $n-1$ points and evaluating it on the point left out. For some learning algorithms, one can obtain estimates of the leave-one-out error, for example for Support Vector Machines: in the separable case, the leave one out error can be bounded in terms of the number of support vectors, [19]. To estimate the leave-one-out error algorithmically, we remove one point, train the classifier and test in on the point left out. This entire process is repeated n times, once for every possible data point that could be left out. The average error on the points left out during this process is the leave-one-out error. More formally,

Let $\mathcal{X}$ denote all the points, $\mathcal{X} = \mathcal{A} \cup \mathcal{B}$. Let $\mathcal{X}^{(i)}$ denote the points with point $\mathbf{x}_i$ left out, $\mathcal{X}^{(i)} = \mathcal{X} \setminus \mathbf{x}_i$. Let $C^{(i)}$ denote the classifier built from $\mathcal{X}^{(i)}$ – in our case this is an optimal weight fat separator. Let e_i denote the error of $C^{(i)}$ applied to the input point $\mathbf{x}_i$.

$$e_i = \begin{cases} 0 & \text{if } \mathbf{x}_i \text{ is classified correctly,} \\ w_i = W(\mathbf{x}_i) & \text{otherwise.} \end{cases}$$

The **leave-one-out** error, $\mathcal{E}_{loo}$ is given by $\mathcal{E}_{loo} = \frac{1}{n}\sum_{i=1}^{n} e_i$. A brute force application of Theorem 1.4 to the computation of $\mathcal{E}_{loo}$ results in a factor of n increase in the run time (since the optimal boosting must be computed n times), which would result in an $O(mn^2 \log n)$ algorithm. We show how to modify the algorithm so that it outputs $\mathcal{E}_{loo}$ with no increase in the computational complexity, establishing Theorem 1.5, which we restate here for convenience.

Theorem 3.1 *An optimal fat separator, together with its optimal separator set $\mathcal{Q}(\mathcal{A}, \mathcal{B})$* **and** *the leave-one-out error can be be found in time $O(mn \log n)$ time.*

Before we prove this theorem, we will need some preliminary results.

Let $\mathcal{Q}(\mathcal{X})$ be an optimal separator set for the set of points $\mathcal{X}$, and let $\mathcal{V}$ be any subset of $\mathcal{Q}(\mathcal{X})$. We consider the set $\mathcal{X}' = \mathcal{X} \setminus \mathcal{V}$, i.e., a set resulting from the removal of some part of an optimal separator set from the original set. Note that $\mathcal{Q}(\mathcal{X})$ is misclassified by the optimal fat separator trained on $\mathcal{X}$. Consider $\mathcal{Q}(\mathcal{X}')$, i.e. an optimal separator set for the reduced set of points, and its corresponding fat separator hyperplane ℓ'.

Lemma 3.2 *Let $\mathcal{Q}(\mathcal{X})$ be an optimal separator set for $\mathcal{X}$, let $\mathcal{V} \subseteq \mathcal{Q}(\mathcal{X})$ and let $\mathcal{X}' = \mathcal{X} \setminus \mathcal{V}$. Let $\mathcal{Q}(\mathcal{X}')$ be an optimal separator set for $\mathcal{X}'$. Then $W(\mathcal{Q}(\mathcal{X}')) = W(\mathcal{Q}(\mathcal{X})) - W(\mathcal{V})$, and any separator hyperplane ℓ' associated to $\mathcal{Q}(\mathcal{X}')$ misclassifies every point in $\mathcal{V}$.*

Proof: Certainly $\mathcal{Q}(\mathcal{X}) \setminus \mathcal{V}$ is a separator set for $\mathcal{X}'$, and so

$$W(\mathcal{Q}(\mathcal{X}')) \leq W(\mathcal{Q}(\mathcal{X}) \setminus \mathcal{V}) = W(\mathcal{Q}(\mathcal{X})) - W(\mathcal{V}). \qquad (*)$$

Now consider adding back the points in $\mathcal{V}$ to $\mathcal{X}'$. Let ℓ' be any separator associated to $\mathcal{Q}(\mathcal{X}')$, and let $\mathcal{V}' \subseteq \mathcal{V}$ be the points in $\mathcal{V}$ that ℓ' misclassifies. Note that $W(\mathcal{V}') \leq W(\mathcal{V})$. If we add back into $\mathcal{Q}(\mathcal{X}')$ all the points in $\mathcal{V}'$, then we get a separator set for $\mathcal{X}$ where ℓ' is a corresponding separator line. To see this, note that ℓ' separates the points in $\mathcal{X}' \setminus \mathcal{Q}(\mathcal{X}') = \mathcal{X} \setminus (\mathcal{V} \cup \mathcal{Q}(\mathcal{X}'))$. Since by definition of $\mathcal{V}'$, ℓ' separates the points in $\mathcal{V} \setminus \mathcal{V}'$, it follows that we can add back the set $\mathcal{V} \setminus \mathcal{V}'$ and ℓ' will continue to separate the resulting set. Thus, ℓ' separates $\mathcal{X} \setminus (\mathcal{V}' \cup \mathcal{Q}(\mathcal{X}'))$, i.e. $\mathcal{V}' \cup \mathcal{Q}(\mathcal{X}')$ is a separator set for $\mathcal{X}$. Therefore, by the optimality of $\mathcal{Q}(\mathcal{X})$,

$$W(\mathcal{Q}(\mathcal{X})) \leq W(\mathcal{V}' \cup \mathcal{Q}(\mathcal{X}')) = W(\mathcal{V}') + W(\mathcal{Q}(\mathcal{X}')) \leq W(\mathcal{V}) + W(\mathcal{Q}(\mathcal{X}')). \qquad (**)$$

We conclude that $W(\mathcal{Q}(\mathcal{X}')) = W(\mathcal{Q}(\mathcal{X})) - W(\mathcal{V})$. We now argue that $\mathcal{V}' = \mathcal{V}$, i.e., ℓ' misclassifies every point in $\mathcal{V}$. Suppose to the contrary, that $\mathcal{V}' \subset \mathcal{V}$, i.e. $W(\mathcal{V}') < W(\mathcal{V})$. Then $(**)$ becomes strict, i.e. $W(\mathcal{Q}(\mathcal{X})) < W(\mathcal{V}) + W(\mathcal{Q}(\mathcal{X}'))$, which contradicts $(*)$, concluding the proof. ■

By lemma 3.2, if $\mathbf{x}_i \in \mathcal{Q}(\mathcal{X})$ and $Q^{(i)}(\mathcal{X}^{(i)})$ is an optimal separator set for $\mathcal{X}^{(i)}$ with fat separator $\ell^{(i)}$, then $\ell^{(i)}$ misclassifies $\mathbf{x}_i$. So, $e_i = 1$ for any $\mathbf{x}_i \in \mathcal{Q}(\mathcal{X})$. We thus conclude that

Corollary 3.3 $\mathcal{E}_{loo} = \dfrac{W(\mathcal{Q}(\mathcal{X}))}{n} + \dfrac{1}{n} \displaystyle\sum_{x_i \in \mathcal{X} \setminus \mathcal{Q}(\mathcal{X})} e_i.$

Lemma 3.3 immediately gives a lower bound $\mathcal{E}_{loo} \geq \frac{W(\mathcal{Q}(\mathcal{X}))}{n}$ for any optimal separator set $\mathcal{Q}(\mathcal{X})$. Further, it suffices to compute e_i only for $\mathbf{x}_i \in \mathcal{X} \setminus \mathcal{Q}(\mathcal{X})$. With only a slight change in the algorithm given in the proof of the Theorem 2.3 we will be able to compute these terms, and hence obtain the exact value of the leave-one-out error. First observe that the following simple lemma holds.

Lemma 3.4 *If $\mathcal{Q}$ is a separator set for $\mathcal{X}^{(i)}$, then $\mathcal{Q} \cup \mathbf{x}_i$ is a separator set for $\mathcal{X}$.*

Thus, all separator sets of $\mathcal{X}^{(i)}$ are subsets of separator sets of $\mathcal{X}$. The complication that arises in the computation of the leave-one-out error is that neither the optimal separator set nor the optimal fat separator is unique. A point $\mathbf{x}_i \in \mathcal{X}$ can be one of three types:

Type I $\mathbf{x}_i$ is classified correctly by all distinct optimal fat separators constructed for $\mathcal{X}^{(i)}$. Such an $\mathbf{x}_i$ makes no contribution to the leave-one-out error.

Type II $\mathbf{x}_i$ is misclassified by all optimal fat separators constructed for $\mathcal{X}^{(i)}$. Such an $\mathbf{x}_i$ contributes w_i to the leave-one-out error.

Type III There are distinct optimal separator sets for $\mathcal{X}^{(i)}$. In the algorithm, an optimal separator set is the result of selecting a central point and selecting one of the separator sets for a line that passes through the central point. We denote this an occurrence of the optimal separator set. A reasonable probabilistic model is to say that one of these occurrences will be selected randomly with uniform probability. Let N_c of these occurrences result in fat separators that classify $\mathbf{x}_i$ correctly and N_e of them misclassify $\mathbf{x}_i$. In this case, there is ambiguity in how we compute the error e_i, and we choose the expected value of e_i under the assumption that each of these optimal separator set occurrences is selected with probability $1/(N_c + N_e)$, so

$$e_i = w_i \frac{N_e}{N_e + N_c}.$$

The leave-one-out error can be computed by summing the contributions of every point. Points of type I can be ignored. We now focus on points of type II and III. Let W_{opt} be the weight of any optimal separator set for $\mathcal{X}$. The next lemma gives a useful characterization of points that are misclassified after leaving them out and computing an optimal fat separator. The usefulness of the lemma lies in that it relates to the weight of separator sets for the entire data set (not the leave one out data set). We will use the notation $\mathcal{Q}_{\mathcal{X}}(\ell)$ $\mathcal{Q}^+_{\mathcal{X}}(\ell)$ to denote separator (positive separator) sets implied by hyperplane ℓ with respect to the points in set $\mathcal{X}$.

Lemma 3.5 *Let $\mathbf{x}_i$ be a positive point which is misclassified by some optimal fat separator ℓ for $\mathcal{X}^{(i)}$. Then there exists a positive representative hyperplane ℓ^* that passes through a positive point $\mathbf{a}^+ \neq \mathbf{x}_i$ such that $\mathbf{x}_i \in \mathcal{Q}^+(\ell^*)$ and $W_{opt} \leq W(\mathcal{Q}^+_{\mathcal{X}}(\ell^*)) \leq W_{opt} + w_i$.*

Proof: Since ℓ is an optimal fat separator for $\mathcal{X}^{(i)}$, no point of $\mathcal{X}^{(i)}$ can lie on ℓ as otherwise we could slightly shift ℓ to obtain a better separator set. Since ℓ misclassifies $\mathbf{x}_i$, $\mathbf{x}_i$ is either on ℓ or on its negative side. Let $\mathbf{u}$ be the closest point in $\mathcal{X}^{(i)}$ to ℓ that is on its positive side. Let ℓ^* be the hyperplane parallel to ℓ passing through $\mathbf{u}$. By definition of $\mathbf{u}$, there are no points in the region between ℓ and ℓ^*. If ℓ^* contains no positive points, then this would contradict the optimality of ℓ, as there would be a small enough shift of ℓ^* further to its positive side which would now correctly classify all the negative points on ℓ^* (at least one). This shifted hyperplane would have strictly smaller error than ℓ, contradicting the optimality of ℓ. Thus there is at least one positive point $\mathbf{a}^+ \neq \mathbf{x}_i$ on ℓ^*. For ℓ^* to satisfy the requirements of the lemma, it only remains to show the bounds on the weight of its positive separator set, where the positive separator set is defined with respect to the entire set $\mathcal{X}$.

Since any positive separator set is also a separator set (Lemma 2.8), $W_{opt} \le W_{\mathcal{X}}(\mathcal{Q}^+(\ell^*))$. Since there are no points between ℓ and ℓ^*, all the positive points on ℓ^* are not included in $\mathcal{Q}^+_{\mathcal{X}}(\ell^*)$ and $\mathbf{x}_i$ is a positive point not on ℓ^* that is misclassified by both ℓ and ℓ^*, we have that $\mathcal{Q}^+_{\mathcal{X}}(\ell^*) = \mathcal{Q}_{\mathcal{X}}(\ell)$. Consider any optimal separator set $\mathcal{Q}$ for $\mathcal{X}$. This is a separator set for $\mathcal{X}^{(i)}$, and so by the optimality of ℓ for $\mathcal{X}^{(i)}$, $W(\mathcal{Q}_{\mathcal{X}^{(i)}}(\ell)) \le W_{opt}$. Since $\mathbf{x}_i$ is misclassified by ℓ, $\mathcal{Q}_{\mathcal{X}}(\ell) = \mathcal{Q}_{\mathcal{X}^{(i)}}(\ell) \cup \{\mathbf{x}_i\}$, and so

$$W(\mathcal{Q}^+_{\mathcal{X}}(\ell^*) = W(\mathcal{Q}_{\mathcal{X}}(\ell)) = W(\mathcal{Q}_{\mathcal{X}^{(i)}}(\ell)) + w_i \le W_{opt} + w_i.$$

■

In words, Lemma 3.5 states that if a positive point $\mathbf{x}_i$ is misclassified by its leave-one-out fat separator, then there is some hyperplane passing through a different positive point that satisfies two conditions: it misclassifies $\mathbf{x}_i$; and, the weight of its positive separator set cannot be too large, at most $W_{opt} + w_i$. If there is no hyper-plane passing through a positive point satisfying these two conditions, then we conclude that the point is of type I. The algorithmic implication of this lemma is that to find points that are misclassified by their leave-one-out fat separator, it suffices to consider the positive separator sets in $\mathcal{X}$ which have sufficiently small weight. These positive separator sets are enumerated by the positive central points. The next lemma shows that if the upper bound $W_{opt} + w_i$ is strict, then *every* leave-one-out optimal fat separator for $\mathbf{x}_i$ misclassifies it.

Lemma 3.6 *Let $\mathbf{x}_i$ be positive point and ℓ^* any hyperplane that passes through a positive point $\mathbf{a}^+ \ne \mathbf{x}_i$ that strictly misclassifies $\mathbf{x}_i$, i.e., $\mathbf{x}_i \in \mathcal{Q}^+(\ell^*)$. Suppose that $W(\mathcal{Q}^+(\ell^*)) < W_{opt} + w_i$. Then,* every *optimal fat separator for $\mathcal{X}^{(i)}$ will misclassify $\mathbf{x}_i$.*

Proof: By construction, $W(\mathcal{Q}^+_{\mathcal{X}^{(i)}}(\ell^*)) = W(\mathcal{Q}^+_{\mathcal{X}}(\ell^*)) - w_i < W_{opt}$, where the inequality is implied by the assumption in the lemma. Let $\mathcal{Q}$ be an optimal separator set for $\mathcal{X}^{(i)}$ with fat separator ℓ. Suppose that ℓ correctly classifies $\mathbf{x}_i$. Since $\mathcal{Q}$ is optimal for $\mathcal{X}^{(i)}$, $W(\mathcal{Q}) \le W(\mathcal{Q}^+_{\mathcal{X}^{(i)}}(\ell^*)) < W_{opt}$ (Lemma 2.8). Further, since the fat separator for $\mathcal{Q}$ correctly classifies $\mathbf{x}_i$, $\mathcal{Q}$ is a separator set for $\mathcal{X}$, which means that $W(\mathcal{Q}) \ge W_{opt}$, a contradiction. Thus, ℓ must misclassify $\mathbf{x}_i$. ■

The implication of Lemma 3.6 is that if a hyperplane ℓ^* which passes through a positive point (not $\mathbf{x}_i$) is found which has sufficiently small weight $W(\mathcal{Q}^+(\ell^*))$ and it strictly misclassifies $\mathbf{x}_i$, then $\mathbf{x}_i$ is a type II point. Thus we can identify type I and type II points. Analogs of Lemmas 3.6 and 3.5 can also be shown for negative points, where we define negative separator sets and their corresponding negative separator lines as we did their positive siblings. Specifically,

Lemma 3.7 *Let* $\mathbf{x}_i$ *be a negative point which is misclassified by some optimal fat separator* ℓ *for* $\mathcal{X}^{(i)}$. *Then there exists a negative representative hyperplane* ℓ^* *that passes through a negative point* $\mathbf{a}^- \neq \mathbf{x}_i$ *such that* $\mathbf{x}_i \in \mathcal{Q}^-(\ell^*)$ *and* $W_{opt} \leq W(\mathcal{Q}^-_{\mathcal{X}}(\ell^*)) \leq W_{opt} + w_i$.

Lemma 3.8 *Let* $\mathbf{x}_i$ *be negative point and* ℓ^* *any hyperplane that passes through a negative point* $\mathbf{a}^- \neq \mathbf{x}_i$ *that strictly misclassifies* $\mathbf{x}_i$, *i.e.,* $\mathbf{x}_i \in \mathcal{Q}^-(\ell^*)$. *Suppose that* $W(\mathcal{Q}^-(\ell^*)) < W_{opt} + w_i$. *Then,* every *optimal fat separator for* $\mathcal{X}^{(i)}$ *will misclassify* $\mathbf{x}_i$.

The four lemmas above give us following algorithm for identifying the type I and II points.

1: : //**Algorithm to identify type I and II points**.
2: Run the algorithm to determine an optimal weight separator set with weight W_{opt}.
3: **for** all positive points $\mathbf{a}^+$ **do**
4: Consider all other points in their sorted angle coordinates (not mirrored angle coordinates) centered at $\mathbf{a}^+$.
5: Consider (in sorted order) all the different separator sets generated by positive representative lines that pass through $\mathbf{a}^+$.
6: For every positive representative line ℓ with weight W_ℓ

(a) Mark all positive points in $\mathcal{Q}^+(\ell)$ with weight greater than $W_\ell - W_{opt}$ as type II (the contribution of such a point to the LOO error is equal to its weight).

(b) Mark all positive points in $\mathcal{Q}^+(\ell)$ with weight equal to $W_\ell - W_{opt}$ as type III.

7: **end for**
8: Repeat steps 3-7 for negative points and negative representative lines.

Note that a type II point once marked, cannot be unmarked. A type III point could subsequently become marked as type II. All points which are left unmarked are of type I.

We show how to implement step 6 in total time $O(n \log n)$ for a given $\mathbf{a}^+$. This will mean that the runtime of the entire algorithm is $O(n^2 \log n)$, since the for loop must be run for every positive and negative point. We give the argument for the positive representative lines. For every positive representative line ℓ, we construct the opening angle $o(\ell) \in [0, \pi)$ and the closing angle $c(\ell) = o(\ell) + \pi$, and label each with the weight $\mathcal{W}(\ell)$, which is the weight of the positive separator set induced by ℓ. The angle interval $[o(\ell), c(\ell)]$ is the interval of angles in which points are classified -1 by ℓ, and hence will be in $\mathcal{Q}^+(\ell)$. Thus, we have a set of "negative" subintervals on the interval $[0, 2\pi)$, each corresponding to one positive representative line. Each subinterval is associated to a weight. For a positive point $\mathbf{x}_i$, we would like to determine the minimum weight negative subinterval which contains it. This can be done for all positive points with respect to a given central point as follows (note that each negative interval corresponds to two marks in $[0, 2\pi)$, and each point to one mark on $[0, 2\pi)$): first, sort all the interval opening and closing marks, together with all the marks corresponding to points ($O(n \log n)$); now process marks in sorted order; if a mark is an open interval, add it to a balanced binary search tree (BST) [8] where the search key is the weight of the interval; if a mark is a close interval, find and remove the corresponding weight from the BST; if the mark is a positive point, obtain the minimum weight interval currently in the BST, and mark the type of the point according to this minimum weight, W_{opt} and the weight of the point. Since each BST operation takes $O(\log n)$ for a total time of $O(n \log n)$ spent on BST operations, and the sorting takes $O(n \log n)$, the total run time is in $O(n \log n)$.

We now give a characterization of positive points which have been marked as type III. The same argument holds for the negative points that have been labeled type III. Remember that for such a point $\mathbf{x}_i$, there was a positive representative line ℓ such that $\mathbf{x}_i \in \mathcal{Q}^+(\ell)$ and $W(\mathcal{Q}^+(\ell)) = W_{opt} + w_i$.

Lemma 3.9 *Suppose that the positive point $\mathbf{x}_i$ is labeled type III by positive representative line ℓ. Then either $Q^+_{\mathcal{X}}(\ell) \setminus \mathbf{x}_i$ is an optimal separator set for $\mathcal{X}^{(i)}$ or $\mathbf{x}_i$ is of type II and will be labeled so at some time in the algorithm.*

Proof: Suppose that $Q^+_{\mathcal{X}}(\ell) \setminus \mathbf{x}_i$ is not an optimal separator set for $\mathcal{X}^{(i)}$. Then there is some positive representative line ℓ_i for $\mathcal{X}^{(i)}$ for which $W(\mathcal{Q}^+_{\mathcal{X}^{(i)}}(\ell_i)) < W(Q^+_{\mathcal{X}}(\ell) \setminus \mathbf{x}_i) = W_{opt}$. If ℓ_i correctly classifies $\mathbf{x}_i$, then $\mathcal{Q}^+_{\mathcal{X}^{(i)}}(\ell_i)$ is a separator set for $\mathcal{X}$ with smaller weight than W_{opt}, a contradiction. Thus, $\mathbf{x}_i$ is of type II. Further, for the positive representative line ℓ_i, $W(\mathcal{Q}^+_{\mathcal{X}}(\ell_i)) < W_{opt} + w_i$, which means that $\mathbf{x}_i$ will be marked as type II when this representative line is encountered. ∎

Lemma 3.9 essentially says that every time a type III point $\mathbf{x}_i$ is labeled as one, we have encountered an optimal separator set for $\mathbf{x}_i$. However, we do not know whether it will be classified correctly or not. Since every optimal separator set for $\mathcal{X}^{(i)}$ will manifest in this way, we may count the number of occurrences of optimal separator sets by simply counting the number of times that we try to label a point as type III. Thus, we can add a counter C_i for every point $\mathbf{x}_i \in \mathcal{X}$, to count this, and it will not increase the algorithm's time complexity. We also update the BST data structure for processing the negative intervals (positive intervals for the negative points) to keep track of the multiplicity of a weight in the BST, which keeps track of the number of positive intervals of a particular weight which are open. Thus, when C_i is updated, it is increased by the number of minimum weight intervals in the BST. At the end of the algorithm, $N_c(\mathbf{x}_i) + N_e(\mathbf{x}_i) = C_i$. We now show how to compute N_c. Let ℓ be a representative line and $W(\mathcal{Q}(\ell)) = W_{opt} + \alpha$. We partition

$\mathcal{Q}(\ell)$ into three sets: S_1, those points with weight less than α; S_2, those points with weight equal to α; S_3, those points with weight greater than α; Then the following lemma holds:

Lemma 3.10 *For representative line ℓ and corresponding separator set $\mathcal{Q}(\ell)$, define S_1, S_2, S_3 as above. Let ℓ_{fat} be the optimal fat separator for $\mathcal{X} \setminus \mathcal{Q}(\ell)$. Then*

(i) ℓ_{fat} misclassifies every point in S_3;

(ii) ℓ_{fat} correctly classifies at most one point in S_2;

(iii) if ℓ_{fat} correctly classifies any point in S_1, then it misclassifies all points in S_2.

Proof: Suppose ℓ_{fat} correctly classifies point $\mathbf{v} \in S_3$. Then $\mathcal{Q}(\ell) \setminus \mathbf{v}$ is a separator set for $\mathcal{X}$ with weight $W_{opt} + \alpha - W(\mathbf{v}) < W_{opt}$, a contradiction. Suppose that ℓ_{fat} correctly classifies a point $\mathbf{u} \in S_2$. To conclude the proof, we show that ℓ_{fat} cannot correctly classify any other point $\mathbf{w}$ in $S_1 \cup S_2$. Suppose, to the contrary, that ℓ_{fat} also correctly classifies $\mathbf{w}$. Then $\mathcal{Q}(\ell) \setminus \{\mathbf{u}, \mathbf{w}\}$ is a separator set for $\mathcal{X}$ with weight $W_{opt} + \alpha - W(\mathbf{u}) - W(\mathbf{w}) < W_{opt}$, a contradiction. ■

Suppose we are given a representative line ℓ, the convex hulls built from the positive and negative points in $\mathcal{Q}(\ell)$, and the convex hulls of the positive and negative points in $\mathcal{X} \setminus \mathcal{Q}(\ell)$. Then in time $O(\log n)$ we can construct the optimal fat separator ℓ_{fat} for $\mathcal{X} \setminus \mathcal{Q}(\ell)$ (Fact 2.11). In $O(\log n)$ time, one can determine if any point $\mathbf{x}_i$ in $\mathcal{Q}(\ell)$ is correctly classified [5, 11, 6]. By Lemma 3.10, if $w_i = W(\mathcal{Q}(\ell)) - W_{opt}$, then it is the only such point correctly classified, and we can increment the counter $N_c(\mathbf{x}_i)$ and move on. If, on the other hand, $w_i < W(\mathcal{Q}(\ell)) - W_{opt}$, then no type III points for this representative line can be classified correctly by ℓ_{fat}, and we can move on. This entire process takes $O(\log n)$ per representative line, given the convex hull representations.

We now show that $N_c(\mathbf{x}_i)$ for every type III point $\mathbf{x}_i$ can be computed in time $O(n^2 \log n)$ as follows. For every central point, sort all the other points according to their angle coordinates. Start enumerating all the representative lines (in sorted order). For every representative line ℓ, compute the optimal fat separator ℓ_{fat} and the necessary convex hulls on $\mathcal{Q}(\ell)$ and $\mathcal{X} \setminus \mathcal{Q}(\ell)$. Then, increase the N_c counter for the type III point from $\mathcal{Q}(\ell)$ with weight $W(\mathcal{Q}(\ell)) - W_{opt}$ (if any exists) that is classified correctly by ℓ_{fat}.

What remains is to show that we can update the necessary convex hulls for a new representative line ℓ from the convex hulls for the separator set of a previous representative line ℓ_{prev} in total time $O(n \log n)$ for all the updates of the representative lines for a given central point. This can be accomplished using Facts 2.9 2.10 and the trick we used in Section 2.1 to maintain these convex hulls - for positive and negative points, we have the add list and the remove list which we maintain separately. The points in the remove list are removed in the reverse order they were added. The details are discussed in Section 2.1. We are now ready to prove Theorem 3.1, which amounts to recapping and summarizing the results we have shown up to now in this section.

Proof: (Theorem 3.1.) We first run the algorithm to determine W_{opt} from the previous section, and then the algorithm to determine the type I, type II and type III points, where we use a counter C_i to keep track of $N_c(\mathbf{x}_i) + N_e(\mathbf{x}_i)$. For the type I points $\mathbf{x}_i$, $e_i = 0$; for the type II points $\mathbf{x}_j$, $e_j = w_j$. For all the type III points $\mathbf{x}_k$, we determine $N_c(\mathbf{x}_k)$ in $O(n^2 \log n)$ time using Lemma 3.9 and the algorithms to efficiently update the convex hulls. The leave one out error for these type III points is then given by $w_k \cdot (C_k - N_c(\mathbf{x}_k))/C_k$. ■

4 Discussion

We have given $O(n^2 \log n)$ algorithms for obtaining the (globally) optimal linear boosting of a pair of classification functions. We made heavy use of some powerful results on convex hull operations and representations. These algorithms are a significant improvement over the brute force exponential algorithm, and an improvment over the $O(n^3)$ which only considers lines through all pairs of points. Our main contribution is to give a clever way to enumerate these lines to result a speed up of a factor close to n. We showed that our algorithm can be extended to maximize the margin among all optimal separator sets. We also showed how to extend the algorithm to compute the leave-one-out error without any increase in the asymptotic computational complexity.

Similar ideas can be used to extend Lemma 2.5 to an analogous (though more complicated) Lemma in 3 dimensions. The generalized lemma can then be used to (efficiently) enumerate all possible separator hyperplanes and obtain an algorithm to compute the optimal separator set for the 3 dimensional case.

Theorem 4.1 (3-dimensions) *An optimal fat separator and its corresponding optimal separator set $\mathcal{Q}(\mathcal{A}, \mathcal{B})$ can be found in $O(mn^2 \log n)$ time.*

This theorem then gives an efficient algorithm to obtain the optimal linear boosting of 3 classification functions. However, in parallel to the convex hull algorithms which have the same computational complexity in both 2 and 3 dimensions, one expects that a similar result for our problem of boosting 3 classification functions should be possible, without any increase in computational complexity. This is certainaly an interesting direction for research.

Another interesting direction would be to compute the optimal linear boosting which maximizes some linear combination of the weight of the separator set and the margin, or more generally some error function $\mathcal{E}(W, mar)$, where W is the weight and mar is the margin. A crucial step in this direction would be to obtain analogous Lemmas to 2.2 and 2.5 in this setting.

A natural direction for progress is to extend the approach further to to obtain an optimal boosting of an arbitrary number (d) of classification functions. Unfortunately, a straightforward extension of the approaches suggested here suffer from a curse of dimensionality (the algorithms become exponential in d). We expect that a fruitful direction for our future research is to obtain some heuristics for the d-dimensional case which have provable approximation properties to the optimal solution. Some plausible directions are a greedy algorithm which succesively optimally boosts pairs of functions according to some greedy criterion until only one is left. Another alternative is a hierarchical (recursive) approach, which divides the classifiers into roughly two groups (say based upon some hierarchical clustering algorithm) and then recursively obtains the optimal boosting of each group. The optimal boostings of each group of classifiers results in two classifiers, which are then optimally boosted together to give the final boosted classifier.

We note that as an alternative to the arbitrarily weighted optimal boosting, one could define a problem in which the weights are somehow related to the linear boosting iteslf (such as the minimum or average distance to the linear boosting hyperplane). In this case linear and quadratic programming techniques can be brought to bear, which have weakly polynomial running time $O(n^3)$ where the constant has exponential dependence on d. For 2 dimensions, however, these algorithms are inferior to our algorithm, and in multi-dimensions, they solve a different problem.

References

[1] Peter Bartlett, Yoav Freund, Wee Sun Lee, and Robert E. Schapire. Boosting the margin: a new explanation for the effectiveness of voting methods. *Annals of Statistics*, 1998.

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

[3] Kristin P. Bennett and Erin J. Bredensteiner. Duality and geometry in SVM classifiers. In *Proc. 17th International Conf. on Machine Learning*, pages 57–64, 2000.

[4] Leo Breiman. Bagging predictors. *Machine Learning*, 1996.

[5] Timothy M. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. In *Proc. 11th Annual Symposium on Computational Geometry*, pages 10–19, 1995.

[6] B. Chazelle. An optimal algorithm for intersecting three-dimensional convex polyhedra. In *IEEE Sympos. on Found. of Comp. Sci. (FOCS)*, volume 30, pages 586–591, 1989.

[7] V. Chvtal. *Linear Programming.* W. H. Freeman and Company, New York, 1983.

[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* Mcgraw-Hill, Cambridge, MA, 2nd edition, 2001.

[9] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and other kernel-based learning methods).* Cambridge University Press, 2000.

[10] I. I. Dikin. Iterative solution of problems of linear and quadratic programming. *Sov. Math. Doklady*, 8(66):674–675, 1967.

[11] D. Dobkin and D. Kirkpatrick. A linear algorithm for determining the separation of convex polyhedra,. *J. Algorithms*, 6:381–392, 1985.

[12] David P. Dobkin and David G. Kirkpatrick. Determining the separation of preprocessed polyhedra - a unified approach. In *Proc. 17th International Colloquium on Automata, Languages and Programming*, pages 400 – 413, 1990.

[13] Yoav Freund and Robert E. Schapire. A decision theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997.

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

[15] L. G. Khachiyan. A polynomial algorithm in linear programming (in russian). *Doklady Akademii Nauk SSSR*, 244:1093–1096, 1979.

[16] Ron Meir and Gunnar R atsch. An introduction to boosting and leveraging. pages 118–183, 2003.

[17] Y. Nesterov and A. Nemirovsky. *Interior Point Polynomial Algorithms in Convex Programming.* SIAM Studies in Applied Mathematics. Society for Industrial and Applied Mathematics, 1994.

[18] V. Tresp. A bayesian committee machine. *Neural Computation*, 2000.

[19] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer–Verlag, 1995.