

On the Equivalence Between the Support Vector Machine for Classification and Sparsified Fisher's Linear Discriminant

A. Amnon Shashua
Institute of Computer Science
Hebrew University of Jerusalem
91904 Jerusalem, Israel
<http://www.cs.huji.ac.il/~shashua/>

(Received ; Accepted in final form)

Abstract. We show that the orientation and location of the separating hyperplane for 2-class supervised pattern classification obtained by the Support Vector Machine (SVM) proposed by Vapnik and his colleagues, is equivalent to the solution obtained by Fisher's Linear Discriminant on the set of Support Vectors. In other words, SVM can be seen as a way to "sparsify" Fisher's Linear Discriminant in order to obtain the most generalizing classification from the training set.

Key words: Unsupervised

1. Introduction

The goal of 2-class supervised learning is stated as follows: Let X be a set of random variables over the real numbers, and let Y be a set of random binary variables over the field $\{1, -1\}$. We are given a "training" set $\{(\mathbf{x}_i, y_i) \in X \times Y\}_{i=1}^l$ obtained by sampling the set $X \times Y$. The classification problem is then to find a function $f(\mathbf{x})$, such that given \mathbf{x} which does not belong to the training set (a "test" example), $f(\mathbf{x})$ will give the "correct" classification (-1 if \mathbf{x} belongs to the first class, or $f(\mathbf{x}) = 1$ if \mathbf{x} belongs to the second class).

Since the test example \mathbf{x} is not part of the training set, the word "correct" often means that $f(\mathbf{x})$ models the probabilistic relationship between X and Y , and thus in turn we tacitly assume that the joint probability distribution $P(\mathbf{x}, y)$ is captured (not necessarily estimated or modeled) by the training set. An equivalent statement of "correctness" is that the desired estimation is the one that provides the best generalization from the given training set.

In a linear classifier approach, the classifier is represented as a linear combination of the input training examples $\{\mathbf{x}_i\}_{i=1}^l$, i.e., if we let $\{\mathbf{x}_i\}_{i=1}^l$ be the columns of the $n \times l$ matrix A , then $w = A\mathbf{c} = \sum_i c_i \mathbf{x}_i$ for some set of coefficients $\mathbf{c} = (c_1, \dots, c_l)$, is a classifier of the form:

$$f(\mathbf{x}) = \text{sign}(\mathbf{x}^\top \mathbf{w} + b)$$

for some scalar b . In most practical situations, a separating hyperplane (\mathbf{w}, b) does not exist, but a non-linear hypersurface does. This can be achieved by projecting the input space into a higher dimensional space (say the space of d -degree monomials) and looking for a separating hyperplane there (see Appendix A for more details).

The question is therefore what should be the criteria for choosing the right coefficient vector \mathbf{c} ? Vapnik and his colleagues (15; 2; 4) have proposed an induction principle, called Structural Risk Minimization, which among all possible models that classify correctly the training data finds the one with the smallest complexity — where the complexity is measured by the VC dimension of the model. In the case of linear classifiers (in some chosen space, input space or higher dimensional feature space), the SRM principle is equivalent to selecting among all possible separating hyperplanes the one that maximizes the margin between the two classes of training data, where the margin is defined as the sum of the distances of the hyperplane from the closest point of the two classes. Vapnik shows that the implementation of this idea can be described as a Quadratic Linear Programming problem. Furthermore, as such, it follows from the Kuhn-Tucker necessary conditions for optimality that the vector \mathbf{c} is *sparse* and the corresponding examples \mathbf{x}_i associated with the non-vanishing coefficients c_i are the vectors on the margin, which Vapnik refers to as Support Vectors.

Girosi (7) has recently shown that Vapnik's Support Vector Machine (SVM) can be rederived directly from a sparsity constraint principle, rather than through the principle of minimizing the complexity of the model measured by its VC dimension. Girosi's rederivation applies to functional approximation (the regression problem), not classification, but given the similarity between the two problems it seems possible that such an approach would extend to the classification problem as well.

The concept of sparsity aims at finding the most parsimonious representation for the problem and is widely spreading in the recent years. One can find the concept of sparsity in the context of Linear Coding or Functional Approximation with the use of overcomplete representations in which a signal is approximated by a linear combination of basis functions taken from a redundant set of signals (12; 8; 3) . In this case, among all approximating functions with the same reconstruction error, the sparsity criteria favors the one with the least number of non-vanishing coefficients. Sparsity is relevant for implementing non-metric similarity measurements (violating the triangle inequality), which appears more naturally suited to similarity judgments performed by humans (14) (see also (9)), which in turn is also

relevant to the recent wave of using Robust estimation methods in Computer Vision (11).

In this paper we make the connection between applying Fisher's Linear Discriminant on a sparse set and the optimal hyperplane found by the Support Vector Machine. We show that the optimal hyperplane is equivalent to the vector maximizing Rayleigh's quotient on the set of Support Vectors. In other words, SVM can be seen as a way to "sparsify" Fisher's Linear Discriminant in order to obtain the most generalizing classification from the training set.

For the sake of completeness, Vapnik's Support Vector Machine is described in Appendix A and Fisher's Linear Discriminant is described in Appendix B. The main result of this paper is described in Section 2, and issues of implementation are described in Appendix C.

2. SVM and Fisher's Linear Discriminant

Let $\mathbf{x}_i, y_i, i = 1, \dots, l$, where $\mathbf{x}_i \in \mathcal{R}^n, y_i \in \{1, -1\}$, be the training set. The linear classifier $f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$ that maximizes the margin between the two classes is a solution to the following Quadratic Programming (QP) problem:

$$\begin{aligned} & \text{Minimize} && \frac{1}{2} \mathbf{w}^\top \mathbf{w} \\ & && \mathbf{w} \\ & \text{Subject to} && y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \geq 0 \end{aligned} \tag{1}$$

and the dual QP problem has the form,

$$\begin{aligned} & \text{Minimize} && \left\{ \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j + b \sum_i \alpha_i y_i - \sum_i \alpha_i \right\} \\ & \text{Subject to} && \alpha_i \geq 0 \end{aligned} \tag{2}$$

where the minimization is over α_i and maximization over b (i.e., b is a saddle point), and $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$. From the Kuhn-Tucker conditions of optimality, $\alpha_i > 0$ for all points \mathbf{x}_i that lie on the boundary (Support Vectors), i.e., $y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 = 0$ whereas $\alpha_i = 0$ for all remaining points. Therefore, the orientation of the separating hyperplane \mathbf{w} is solely determined from the (sparse) set of support vectors. More details can be found in Appendix A. We prove the following result:

Theorem 1. Let \mathcal{S} be the set of support vectors, i.e., $\alpha_i \neq 0$ for all $\mathbf{x}_i \in \mathcal{S}$. The separating hyperplane $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$ where $\{\alpha_i\}$ are the solution to the QP problem (2), is the null space of S_W , i.e., $S_W \mathbf{w} = 0$, where S_W is the sum of the scatter matrices associated with classes one and two (out of \mathcal{S}).

The remainder of this section is devoted to the proof of this statement. Let $c_i = \alpha_i y_i$ (note that $\alpha_i = c_i y_i$ because $y_i \in \{-1, 1\}$), then the QP problem (2) applied only to the subset of training examples \mathcal{S} (the support vectors) has the form:

$$\text{Minimize}_{\mathbf{c}, \mathbf{b}} \left\{ \frac{1}{2} \mathbf{c}^\top A^\top A \mathbf{c} + \mathbf{b} \mathbf{c}^\top \mathbf{1} - \mathbf{c}^\top \mathbf{y} \right\} \quad (3)$$

where A is a $n \times s$ matrix whose columns are the members of \mathcal{S} , $\mathbf{1}$ is the vector of 1s, s is the total number of support vectors (cardinality of \mathcal{S}) and $\mathbf{w} = A\mathbf{c}$. The global minima satisfies the necessary condition:

$$A^\top A \mathbf{c} = \mathbf{y} - \mathbf{b} \mathbf{1} \quad (4)$$

from which we represent \mathbf{b} as a function of \mathbf{w} ,

$$\mathbf{b} = \frac{s_1 - s_2}{s} - \mathbf{w}^\top \boldsymbol{\mu} \quad (5)$$

where s_1, s_2 are the number of support vectors associated with class one and two, respectively, and $\boldsymbol{\mu}$ is the mean of all support vectors. Let $A = UDV^\top$ be the Singular Value Decomposition (SVD) of A , where the columns of U are orthonormal, the rows of V^\top are orthonormal and D is a diagonal matrix of singular values $\{\lambda_i\}$ with the number of non-vanishing entries λ_i being equal to the rank of A . Since $\mathbf{w} = A\mathbf{c}$ we have,

$$\mathbf{w} = UDV^\top (VD^2V^\top)^{-1} (\mathbf{y} - \mathbf{b} \mathbf{1})$$

Note that in case the rank of A is smaller than s , then the solution with the smallest norm (which is what we desire) is obtained by defining D^{-1} to have vanishing entries for every $\lambda_i = 0$ and $1/\lambda_i$ for $\lambda_i \neq 0$ — thus we have:

$$\mathbf{w} = UD(D^2)^{-1}V^\top (\mathbf{y} - \mathbf{b} \mathbf{1}) \quad (6)$$

$$= U(D^2)^{-1}U^\top UDV^\top (\mathbf{y} - \mathbf{b} \mathbf{1}) \quad (7)$$

$$UD^2U^\top \mathbf{w} = UDV^\top (\mathbf{y} - \mathbf{b} \mathbf{1}) \quad (8)$$

which gives us the following relation:

$$AA^\top \mathbf{w} = A(\mathbf{y} - \mathbf{b} \mathbf{1}) \quad (9)$$

After substituting the value of \mathbf{b} from (5), we obtain:

$$AA^\top \mathbf{w} + \mathbf{b} A \mathbf{1} = A \mathbf{y} \quad (10)$$

$$AA^\top \mathbf{w} + \mathbf{b} s \boldsymbol{\mu} = s_1 \boldsymbol{\mu}_1 - s_2 \boldsymbol{\mu}_2 \quad (11)$$

$$\left[AA^\top - s \boldsymbol{\mu} \boldsymbol{\mu}^\top \right] \mathbf{w} = s_1 (\boldsymbol{\mu} - \boldsymbol{\mu}_1) - s_2 (\boldsymbol{\mu} - \boldsymbol{\mu}_2) \quad (12)$$

$$\left[AA^\top - s \boldsymbol{\mu} \boldsymbol{\mu}^\top \right] \mathbf{w} = \frac{2s_1 s_2}{s} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \quad (13)$$

Let S_T be the total scatter matrix of the set \mathcal{S} , i.e.,

$$S_T = \sum_{\mathbf{x}_i \in \mathcal{S}} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top.$$

It can be easily verified that,

$$S_T = AA^\top - s\boldsymbol{\mu}\boldsymbol{\mu}^\top,$$

thus we have,

$$S_T \mathbf{w} = \frac{2s_1s_2}{s}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2). \quad (14)$$

Note that in case S_T is singular, the minimum norm solution \mathbf{w} is found using the pseudo-inverse of S_T (i.e., by use of SVD). Let S_W be the sum of scatter matrices one for each class, i.e.,

$$S_W = \sum_{j=1}^2 \sum_{\mathbf{x}_i \in \mathcal{S}_j} (\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^\top,$$

where \mathcal{S}_j is the subset of elements of \mathcal{S} that belong to class $j = 1, 2$. It can be easily verified that,

$$S_T = S_W + s_1(\boldsymbol{\mu}_1 - \boldsymbol{\mu})(\boldsymbol{\mu}_1 - \boldsymbol{\mu})^\top + s_2(\boldsymbol{\mu}_2 - \boldsymbol{\mu})(\boldsymbol{\mu}_2 - \boldsymbol{\mu})^\top. \quad (15)$$

From eqn. 4 we obtain,

$$\boldsymbol{\mu}^\top \mathbf{w} = \frac{s_1 - s_2}{s} - b \quad (16)$$

$$\boldsymbol{\mu}_1^\top \mathbf{w} = 1 - b \quad (17)$$

$$\boldsymbol{\mu}_2^\top \mathbf{w} = -1 - b \quad (18)$$

Substituting the above in eqn. 15 we obtain,

$$S_T \mathbf{w} = S_W \mathbf{w} + \frac{2s_1s_2}{s}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}) - \frac{2s_1s_2}{s}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}) \quad (19)$$

and after substitution of eqn. 14 we obtain the desired result that

$$S_W \mathbf{w} = 0.$$

□

3. Discussion

The Support Vector Machine is appealing from the standpoint of providing a rigorous and coherent framework for “the correct” classification from training examples. The drawback is the complexity introduced by working with Quadratic Programming especially for large

problems (see Appendix C). The result presented here shows that SVM can be seen as a way to “sparsify” Fisher’s Linear Discriminant in order to obtain the most generalizing classification from the training set. Conversely, this result may motivate an approach for seeking an “approximately correct” classification that may require a much simpler and tractable machinery. For example, since the norm of \boldsymbol{w} is related to the distance between the centers $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ of the two classes taken from the sparse set \mathcal{S} , one may consider a sparsification approach that would gradually reduce the distance between the centers, rather than employing the full strength of the QP machinery.

Recasting the problem of classification as a “sparsified” FLD problem, may also be useful for obtaining a handle for approaching the multi-class classification problem. Currently there is no rigorous approach for extending the SVM method for dealing with $m > 2$ classes. Yet, FLD naturally extends to any number of classes. Therefore, a “sparsified” FLD would enforce constraints among the separating hyperplanes (for example, by adding orthogonality as an additional optimizing criteria).

Finally, since FLD is optimal for Normally distributed classes, there may be further connections between the use of Gaussian processes, sparsification and SVM.

Since FLD has and is being used for classification problems in Vision (see for example, face recognition approaches in (1)), the drive for finding synergies between those classical approaches and modern approaches, like SVM, may provide fruits for better applications as well.

Acknowledgments

I thank Tomaso Poggio, Federico Girosi and Alessandro Verri for helpful discussions and for creating a stimulating research environment throughout my Summer visit at CBCL. Special thanks to Bernhard Schoelkopf for comments on an earlier draft of this report.

Appendix

A. Vapnik’s Support Vector Machine

The SVM approach of (15; 2; 4) seeks to find a separating hyperplane that divides the two classes while maximizing the distance between them (the margin). This criteria follows from the observation that by doing so one would find a solution that possesses the best generalization

properties, in the sense that the VC dimension of the model is the smallest possible.

The set of training examples $\mathbf{x}_i \in \mathcal{R}^n$, $i = 1, \dots, l$ is separable if there exists \mathbf{w}, b such that,

$$\mathbf{w}^\top \mathbf{x}_i + b \geq 1 \quad \text{if } y_i = 1 \quad (20)$$

$$\mathbf{w}^\top \mathbf{x}_i + b \leq -1 \quad \text{if } y_i = -1 \quad (21)$$

or equivalently if $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$. Since \mathbf{w}, b are determined up to a mutual scale, let the scale be defined such that,

$$\min_{\mathbf{x}_i} \{y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1\}$$

In other words, that the distance between the closest point to the hyperplane becomes $1/\sqrt{\mathbf{w}^\top \mathbf{w}}$. Since maximizing the margin is maximizing the distance between the closest point to the hyperplane, we obtain the following optimization criteria:

$$\begin{aligned} & \text{Minimize} \quad \frac{1}{2} \mathbf{w}^\top \mathbf{w} \\ & \mathbf{w} \\ & \text{Subject to } y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \geq 0 \end{aligned} \quad (22)$$

Recall that a non-linear optimization problem of minimizing $f(\mathbf{x})$ under a set of inequality constraints $\mathbf{g}(\mathbf{x}) \geq 0$, has its local minima satisfy the necessary Kuhn-Tucker conditions:

$$\nabla f(\mathbf{x}) - \sum_i \alpha_i \nabla g_i(\mathbf{x}) = 0$$

where $\alpha_i \geq 0$ and $\alpha_i g_i(\mathbf{x}) = 0$, $i = 1, \dots, l$, i.e., $\alpha_i = 0$ when $g_i(\mathbf{x}) > 0$. In this case, since $f(\mathbf{x})$ is convex, the local minimum is also the global one as well. The Lagrange functional in our case is therefore,

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^\top \mathbf{w} - \sum_{i=1}^l \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1) \quad (23)$$

which is to be minimized with respect to \mathbf{w}, b and maximized with respect to the Lagrange multipliers $\alpha_i \geq 0$. Since at the saddle point $\frac{\partial L}{\partial \mathbf{w}} = 0$, we obtain the necessary condition for \mathbf{w} to satisfy,

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i.$$

From the Kuhn-Tucker conditions we have that α_i is non-zero only for the \mathbf{x}_i that satisfy $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$ which are the vectors at the

margin, referred to as Support Vectors, thus the sum is only over the set of support vectors. Putting the expression for \mathbf{w} back into the Lagrange functional we obtain the expression (the dual optimization problem),

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j - b \sum_i \alpha_i y_i \quad (24)$$

which is to be maximized with respect to α_i and minimized with respect to b ¹, under the inequality constraints $\alpha_i \geq 0$. One may then employ one of many software packages for solving for the QP (dual) problem — see also Appendix C.

Vapnik (2; 15) also considers the issue of mapping the input space $\mathbf{x} \in \mathcal{R}^n$ into a higher dimensional feature space Z (possibly of infinite dimensional space) through some, a priori chosen, nonlinear mapping. A separating hyperplane in feature space Z would correspond to a nonlinear decision surface (hypersurface) in the original input space. For example, one may choose the feature space to span the set of polynomials in d variables (the dimension of Z is then exponential in d). If $\mathbf{z}_i = z(\mathbf{x}_i)$ is the mapping from input to feature space, then Vapnik and his colleagues have noticed that if certain conditions are met, the scalar products $\mathbf{z}_i^\top \mathbf{z}_j$ may be computed in the input space by a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{z}_i^\top \mathbf{z}_j$ rather than in the large dimensional feature space. According to Mercer's theorem (see (5)) $K(u, v)$ can be any symmetric function satisfying the following (general) conditions:

$$\int \int K(u, v) g(u) g(v) du dv > 0$$

for all $g \neq 0$ for which

$$\int g^2(u) du < \infty$$

then, $K(u, v)$ has an expansion of the form

$$K(u, v) = \sum_{i=1}^{\infty} \lambda_i \psi_i(u) \psi_i(v)$$

with $\lambda_i > 0$. In other words, $K(u, v)$ describes an inner product in some feature space. For example, the feature space spanning the set of polynomials in d variables, one can easily show that

$$K(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}^\top \mathbf{y})^d.$$

¹ Note that b is the Lagrange multiplier of the equality constraint $\sum \alpha_i y_i = 0$, which arises from setting $\frac{\partial L}{\partial b} = 0$. In Vapnik's derivation the term with b does not appear in the dual functional, and instead the equality constraint is added to the constraints $\alpha_i \geq 0$.

The QP problem that should be solved is exactly like eqn. (24) with the exception that $\mathbf{x}_i^\top \mathbf{x}$ is replaced by $K(\mathbf{x}_i, \mathbf{x})$. The resulting linear classifier has the form:

$$f(\mathbf{x}) = \text{sign}\left(\sum_i y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b\right).$$

B. Fisher's Linear Discriminant

Let $\mathbf{x}_i \in \mathcal{R}^n$, $i = 1, \dots, l$ be divided into m sets (classes) ψ_1, \dots, ψ_m and let $\boldsymbol{\mu}$ be the center of all the data points, and $\boldsymbol{\mu}_j$ be the center of class j . We wish to find a direction vector $\mathbf{w} \in \mathcal{R}^n$ (perpendicular to hyperplanes $\mathbf{w}^\top \mathbf{x} + b = 0$, where b is the distance of the hyperplane from the origin), such that the projection of the centers $\boldsymbol{\mu}_j$ onto the direction \mathbf{w} has the maximum variance (the centers are as separated from each other as possible), and that the projection of the points $\mathbf{x}_i \in \psi_j$ are clustered near the projection of the center $\boldsymbol{\mu}_j$. This desire can be formalized into the following optimization functional.

Recall that the signed distance between a point \mathbf{x} and the hyperplane (\mathbf{w}, b) is

$$\frac{\mathbf{w}^\top \mathbf{x} + b}{|\mathbf{w}|}$$

where $|\mathbf{w}|$ is the norm of \mathbf{w} . The relative (signed) distance between two points $\mathbf{x}_1, \mathbf{x}_2$ from the hyperplane is thus,

$$\frac{\mathbf{w}^\top (\mathbf{x}_1 - \mathbf{x}_2)}{|\mathbf{w}|},$$

which is the distance between the projections of the points onto the direction \mathbf{w} . The sum of square distances between the projected points in class j from the projected center $\boldsymbol{\mu}_j$ is:

$$\frac{1}{|\mathbf{w}|} \sum_{\mathbf{x}_i \in \psi_j} (\mathbf{w}^\top (\mathbf{x}_i - \boldsymbol{\mu}_j))^2 = \frac{1}{|\mathbf{w}|} \mathbf{w}^\top S_j \mathbf{w}$$

where S_j is the scatter matrix of class j defined by:

$$S_j = \sum_{\mathbf{x}_i \in \psi_j} (\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^\top.$$

The sum of variances of each class around its center (in the projected space) is therefore

$$\frac{1}{|\mathbf{w}|} \mathbf{w}^\top S_W \mathbf{w}, \quad (25)$$

where $S_W = \sum_{j=1}^m S_j$. Our desire is to bring the expression above to minimum. The sum of square distances of the projected centers $\boldsymbol{\mu}_j$ from the global center $\boldsymbol{\mu}$ is:

$$\frac{1}{|\mathbf{w}|} \sum_{j=1}^m (\mathbf{w}^\top (\boldsymbol{\mu}_j - \boldsymbol{\mu}))^2 = \frac{1}{|\mathbf{w}|} \mathbf{w}^\top S_B \mathbf{w}, \quad (26)$$

where S_B is the scatter matrix of the centers, defined by:

$$S_B = \sum_{j=1}^m (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^\top.$$

The desire to minimize expression (25) and maximize expression (26) is satisfied by maximizing the quotient

$$J(\mathbf{w}) = \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}},$$

known as the ‘‘Rayleigh’’ quotient. The necessary condition for extremum is

$$0 = \frac{\partial J}{\partial \mathbf{w}} = \frac{(\mathbf{w}^\top S_W \mathbf{w}) S_B \mathbf{w} - (\mathbf{w}^\top S_B \mathbf{w}) S_W \mathbf{w}}{(\mathbf{w}^\top S_W \mathbf{w})^2}$$

therefore \mathbf{w} satisfies

$$S_B \mathbf{w} = \lambda S_W \mathbf{w}$$

that is the generalized eigenvector associated with the maximal generalized eigenvalue ($\lambda = J(\mathbf{w})$). In case S_W is non-singular, then \mathbf{w} is the eigenvector associated with the maximal eigenvalue of $S_W^{-1} S_B$.

In the case of two classes, $m = 2$, the computation is simpler since the squared projected distance between the centers is

$$\frac{1}{|\mathbf{w}|} (\mathbf{w}^\top (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2))^2 = \frac{1}{|\mathbf{w}|} \mathbf{w}^\top S_B \mathbf{w},$$

where S_B is

$$S_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top,$$

and since $S_B \mathbf{w} \cong (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$, we have that,

$$\mathbf{w} \cong S_W^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

and if S_W is singular we take the pseudo-inverse solution which selects the solution with the smallest norm.

Using Fisher’s solution \mathbf{w} as a separating hyperplane for the 2-class problem is optimal when the distributions of both classes are Normal

with equal covariance matrices (for more details, see (6)). Otherwise, there is no guarantee that, even when the classes are linearly separable, that a separable hyperplane would be found.

C. Issues of Implementation

The most popular method for solving QP problems is the “active set” method (see (10), for more details). Let \mathcal{S} be a candidate set of support vectors and let \mathbf{w}, b be a feasible solution (separable hyperplane), i.e., $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$, $i = 1, \dots, l$. In case $n \geq l$ (which is typical, especially when working in the large dimensional feature space), then the solution given by (14), using pseudo-inverse when $n > l$, is feasible and \mathcal{S} contains all the training data. Otherwise, one may use Linear Programming to find a feasible solution, and then select \mathcal{S} to be the margin vectors of the LP solution.

The active set method updates \mathcal{S} by gradually adding and removing points while *maintaining* feasibility of the solution at every step, and while also strictly decreasing the criteria functional at every step. Hence, the optimal solution is guaranteed after a finite number of steps, albeit exponential in the worst case. The iterations proceed as follows: (1). Solve for $\Delta \mathbf{w}$ and Δb , i.e.,

$$S_T \Delta \mathbf{w} = \frac{2s_1 s_2}{s} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) - S_T \mathbf{w},$$

using the pseudo-inverse of S_T (minimum-norm solution).

(2). Consider,

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \Delta \mathbf{w} \quad (27)$$

$$b \leftarrow b + \alpha \Delta b \quad (28)$$

If $\alpha = 1$ is a feasible solution, then we have found an optimal solution over the set \mathcal{S} , goto Step (3). Otherwise, find the largest $0 < \alpha < 1$ that maintains feasibility — add the new Support Vector to \mathcal{S} , goto Step (1).

(3). Calculate \mathbf{c} from eqn. (4), or from the relation $\mathbf{w} = A\mathbf{c}$. If $c_i y_i > 0$ for all $x_i \in \mathcal{S}$, we are done. Otherwise, remove from \mathcal{S} the point associated with the smallest $c_i y_i$. Go to Step (1).

Few comments. In Step (2) one is moving along the direction of the optimal solution of the unconstrained problem. New Support Vectors are added when the hyperplane “hits” upon them during the update process, until the unconstrained problem can no longer be improved. If the Lagrange multipliers $\alpha_i = c_i y_i$ are all positive, then we have

satisfied the Kuhn-Tucker conditions, otherwise the criterion functional can be decreased further by removing the (most) negative $c_i y_i$. Since every step strictly reduces the criterion function, any configuration \mathcal{S} cannot occur twice, thus after a finite number of steps the process must end. More efficient techniques (in terms of memory requirements and average running times) can be found in (13).

References

- P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. In *Proceedings of the European Conference on Computer Vision*, 1996.
- B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifier. In *Proc. 5th Workshop on Computational Learning Theory*, pages 144–152, 1992.
- S. Chen and D. Donoho. Atomic decomposition by basis pursuit. Technical Report Dept. of Statistics, TR-479, Stanford, 1995.
- C. Cortes and V.N. Vapnik. Support vector networks. *Machine Learning*, 20:1–25, 1995.
- R. Courant and D. Hilbert. *Methods of mathematical physics*. Interscience Publishers Inc., 1953.
- R.O. Duda and P.E. Hart. *Pattern classification and scene analysis*. John Wiley, New York, 1973.
- F. Girosi. An equivalence between sparse approximation and support vector machines. Technical Report AI Memo 1606, MIT, 1997.
- G.F. Harpur and R.W. Prager. Development of low entropy coding in a recurrent network. *Network*, 7:277–284, 1996.
- D.W. Jacobs and D. Weinshall. Classifying images using non-metric distances. In *Proceedings of the International Conference on Computer Vision*, January 1998.
- D.G. Luenberger. *Linear and nonlinear programming*. Addison-Wesley, 1937.
- P. Meer, D. Mintz, D. Kim, and A. Rosenfeld. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, 1991.
- B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(13), 1996.
- E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997.
- A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.
- V.N. Vapnik. *The nature of statistical learning*. Springer, 1995.