36-724 Spring 2006: Linear Discriminant Methods

Brian Junker

April 18, 2006

- Separating Hyperplanes
- Perceptrons (Rosenblatt, 1958)
- Optimal Separating Hyperplanes (Vapnik, 1996)
- Support Vector Methods...
Separating Hyperplanes

Consider a two-class classification problem. Both LDA and logistic regression separate the classes with an (affine) hyperplane

\[ L = \{ x : \beta_0 + \beta_1^T x = 0 \} \]

and provide a discriminant function

\[ f(x) = \beta_0 + \beta_1^T x = \begin{cases} < 0 & \Rightarrow \text{Class 1} \\ > 0 & \Rightarrow \text{Class 2} \end{cases} \]

Note

- For any \( x_0 \in L, \beta^T x = -\beta_0 \).
- For any \( x_1, x_2 \in L, \beta^T (x_1 - x_2) = 0 \); hence \( \beta^* = \beta/||\beta|| \) is a unit vector normal to \( L \);
- Let \( x \notin L \), let \( x_0 \in L \); the signed distance from \( x \) to \( L \) can be obtained by projecting the difference vector \( x - x_0 \) onto \( \beta^* \):

\[ \beta^{*T} (x - x_0) = (1/||\beta||)(\beta^T x + \beta_0) = f(x)/||f'(x)|| \]
Perceptrons (Rosenblatt, 1958)

Suppose we code \( i \in \text{Class 1} \) as \( y_i = 1 \) and \( i \in \text{Class 2} \) as \( y_i = -1 \). We could try to find a good hyperplane by minimizing the distance of misclassified points to the hyperplane.

- If a response \( y_i = 1 \) is misclassified then \( x_i^T \beta + \beta_0 < 0 \)
- If a response \( y_i = -1 \) is misclassified then \( x_i^T \beta + \beta_0 > 0 \)

So we want to minimize

\[
D(\beta, \beta_0) = -\sum_{i \in M} y_i (x_i^T \beta + \beta_0)
\]

where \( M \) is the set of mis-classified points. (Note: \( e^D \) is the “exponential” loss function in boosting…)

The gradient of \( D \) is given by

\[
\frac{\partial D}{\partial \beta} = -\sum_{i \in M} y_i x_i \\
\frac{\partial D}{\partial \beta_0} = -\sum_{i \in M} y_i
\]
The perceptron algorithm visits each misclassified point in some order and updates using a gradient-descent-like method:

\[
\begin{pmatrix}
\beta \\
\beta_0
\end{pmatrix}
\leftarrow
\begin{pmatrix}
\beta \\
\beta_0
\end{pmatrix}
+ \rho
\begin{pmatrix}
y_i x_i \\
y_i
\end{pmatrix}
\] (typically \( \rho = 1 \) but this is not central).

- When the data are separable by a hyperplane this algorithm will eventually converge; however the solution depends on the order in which you visit elements of \( \mathcal{M} \).
- The number of steps to convergence increases as the separation between the sets decreases.
- When the data are not separable, the algorithm cycles, often in large loops that are hard to detect.
Optimal Separating Hyperplanes (Vapnik, 1996)

We can get a unique solution to the problem if we “thicken out” the separating hyperplane, as much as possible.

Thus we want to change the perceptron criterion to:

\[
\max_{\beta, \beta_0 : \|\beta\|=1} C, \text{ such that } y_i(x_i^T\beta + \beta_0) \geq C \forall i = 1, \ldots, N
\]

or equivalently

\[
\max_{\beta, \beta_0} C, \text{ such that } y_i(x_i^T\beta + \beta_0) \geq C\|\beta\| \forall i = 1, \ldots, N
\]

Taking \(\|\beta\| = 1/C\), we get the equivalent problem

\[
\min_{\beta, \beta_0} \frac{1}{2}\|\beta\|^2, \text{ such that } y_i(x_i^T\beta + \beta_0) \geq 1 \forall i = 1, \ldots, N
\]

This is a convex optimization problem (quadratic criterion, linear constraints) that can be solved with Lagrange-multiplier methods.
The Convex Optimization Problem

We want to minimize

\[ L_P = \frac{1}{2} \| \beta \|^2 + \sum_{i=1}^{N} \alpha_i [y_i (x_i^T \beta + \beta_0) - 1] \]

subject to the additional constraint that the Lagrange multipliers \( \alpha_i \) all satisfy \( \alpha_i \geq 0 \); note that \( L_p \) is basically a penalized loss function again.

Setting derivatives with respect to \( \beta \) and \( \beta_0 \) equal to zero we discover

\[
\beta = \sum_{i=1}^{N} \alpha_i y_i x_i \\
0 = \sum_{i=1}^{N} \alpha_i y_i
\]
substituting back, we obtain the simpler problem of minimizing

\[ L_D = \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k x_i^T x_k - \sum_{i=1}^{N} \alpha_i = \frac{1}{2} \alpha^T A \alpha - \alpha^T 1 \]

subject to \( \alpha_i \geq 0 \forall i \), and \( \sum_{i=1}^{N} \alpha_i y_i = 0 \) [where \( A = (\text{diag}(y)X)(X^T \text{diag}(y)) \)].

This can be solved with standard software (e.g. \texttt{optim()} in R); and then

\[ \beta = \sum_{i=1}^{N} \alpha_i y_i x_i \]

\[ \beta_0 = -\frac{1}{2} \beta^T (x_r + x_s) \]

where \( x_r \) and \( x_s \) are any two support points (\( \alpha_r, \alpha_s > 0 \)) from opposite classes.

\textit{Some comments}

- The constraints above define an empty slab around \( L \) of thickness \( 1/\|\beta\| \);
- The algorithm essentially choose among the many separating hyperplanes to maximize the thickness \( 1/\|\beta\| \).
- The slab, and the hyperplane, are defined by \textit{support points}—that is, those points in either class on the boundary of the slab.
Support Vector Methods...

When the classes are not linearly separable, we are in trouble again. There are (at least) two approaches to “fixing” the problem:

- Define “slack variables” $\xi_1, \ldots, \xi_N$, with $\xi_i \geq 0$ and $\sum_i \xi_i \leq K$ for some $K$. Then we can solve

$$\max_{\beta, \beta_0 : \|\beta\| = 1} C, \text{ such that } y_i(x_i^T \beta + \beta_0) \geq C(1 - \xi_i) \forall i = 1, \ldots, N$$

which is equivalent to solving

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2, \text{ such that } y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \forall i = 1, \ldots, N$$

This is the support vector classifier.

- Another attractive alternative is the method of support vector machines; more on this next time!