Perceptron Learning Algorithm

Jia Li

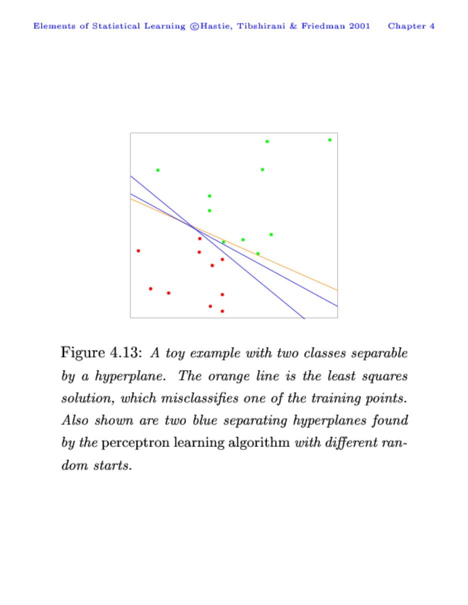
Department of Statistics
The Pennsylvania State University

Email: jiali@stat.psu.edu
http://www.stat.psu.edu/∼jiali

Separating Hyperplanes

- Construct linear decision boundaries that explicitly try to separate the data into different classes as well as possible.
- Good separation is defined in a certain form mathematically.
- Even when the training data can be perfectly separated by hyperplanes, LDA or other linear methods developed under a statistical framework may not achieve perfect separation.

Perceptron Learning Algorithm



Elements of Statistical Learning ©Hastie, Tibshirani & Friedman 2001 Chapter 4

Figure 4.13: *A toy example with two classes separable by a hyperplane. The orange line is the least squares solution, which misclassifies one of the training points. Also shown are two blue separating hyperplanes found by the* perceptron learning algorithm *with different random starts.*

Review of Vector Algebra

- A hyperplane or *affine set* L is defined by the linear equation:

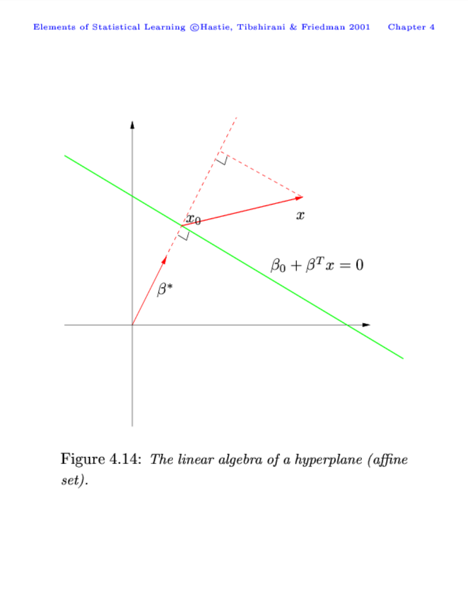
$$L = \{x : f(x) = \beta_0 + \beta^T x = 0\} .$$

- For any two points x_1 and x_2 lying in L, $\beta^T(x_1 - x_2) = 0$, and hence $\beta^* = \beta / \parallel \beta \parallel$ is the vector normal to the surface of L.
- For any point x_0 in L, $\beta^T x_0 = -\beta_0$.
- The signed distance of any point x to L is given by

$$\begin{aligned} \beta^{*T}(x - x_0) &= \frac{1}{\parallel \beta \parallel}(\beta^T x + \beta_0) \\ &= \frac{1}{\parallel f'(x) \parallel} f(x) . \end{aligned}$$

- Hence $f(x)$ is proportional to the signed distance from x to the hyperplane defined by $f(x) = 0$.

Perceptron Learning Algorithm



Elements of Statistical Learning ©Hastie, Tibshirani & Friedman 2001 Chapter 4

Figure 4.14: *The linear algebra of a hyperplane (affine set).*

Rosenblatt's Perceptron Learning

- Goal: find a separating hyperplane by minimizing the distance of misclassified points to the decision boundary.
- Code the two classes by $y_i = 1, -1$.
- If $y_i = 1$ is misclassified, $\beta^T x_i + \beta_0 < 0$. If $y_i = -1$ is misclassified, $\beta^T x_i + \beta_0 > 0$.
- Since the signed distance from x_i to the decision boundary is $\frac{\beta^T x_i + \beta_0}{\|\beta\|}$, the distance from a misclassified x_i to the decision boundary is $\frac{-y_i(\beta^T x_i + \beta_0)}{\|\beta\|}$.
- Denote the set of misclassified points by $\mathcal{M}$.
- The goal is to minimize:

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

Stochastic Gradient Descent

- To minimize $D(\beta, \beta_0)$, compute the gradient (assuming $\mathcal{M}$ is fixed):

$$\begin{aligned} \frac{\partial D(\beta, \beta_0)}{\partial \beta} &= -\sum_{i \in \mathcal{M}} y_i x_i \ , \\ \frac{\partial D(\beta, \beta_0)}{\partial \beta_0} &= -\sum_{i \in \mathcal{M}} y_i \ . \end{aligned}$$

- Stochastic gradient descent is used to minimize the piecewise linear criterion.
- Adjustment on β, β_0 is done after each misclassified point is visited.

- The update is:

$$\begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} \leftarrow \begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} + \rho \begin{pmatrix} y_i x_i \\ y_i \end{pmatrix} .$$

  Here ρ is the learning rate, which in this case can be taken to be 1 without loss of generality. (Note: if $\beta^T x + \beta_0 = 0$ is the decision boundary, $\lambda \beta^T x + \lambda \beta_0 = 0$ is also the boundary.)

Issues

- If the classes are linearly separable, the algorithm converges to a separating hyperplane in a finite number of steps.
- A number of problems with the algorithm:
  - When the data are separable, there are many solutions, and which one is found depends on the starting values.
  - The number of steps can be very large. The smaller the gap, the longer it takes to find it.
  - When the data are not separable, the algorithm will not converge, and cycles develop. The cycles can be long and therefore hard to detect.

Optimal Separating Hyperplanes

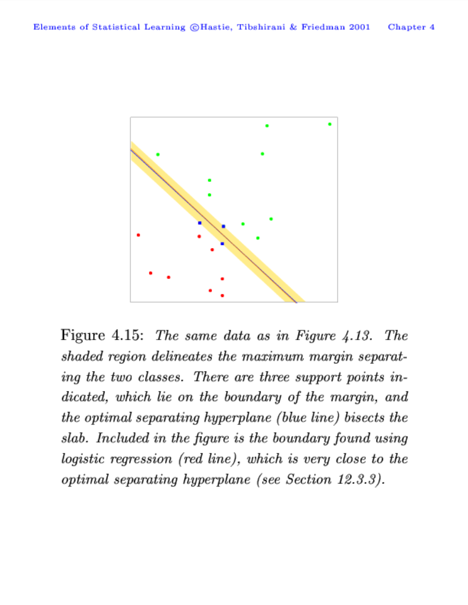
- Suppose the two classes can be linearly separated.
- The *optimal separating hyperplane* separates the two classes and maximizes the distance to the closest point from either class.
- There is a unique solution.
- Tend to have better classification performance on test data.
- The optimization problem:

$$\max_{\beta,\beta_0} C$$

$$\text{subject to } \frac{1}{\parallel \beta \parallel} y_i(\beta^T x_i + \beta_0) \geq C, i = 1, ..., N$$

- Every point is at least C away from the decision boundary $\beta^T x + \beta_0 = 0$.

Perceptron Learning Algorithm



Elements of Statistical Learning ©Hastie, Tibshirani & Friedman 2001 Chapter 4

Figure 4.15: *The same data as in Figure 4.13. The shaded region delineates the maximum margin separating the two classes. There are three support points indicated, which lie on the boundary of the margin, and the optimal separating hyperplane (blue line) bisects the slab. Included in the figure is the boundary found using logistic regression (red line), which is very close to the optimal separating hyperplane (see Section 12.3.3).*

- For any solution of the optimization problem, any positively scaled multiple is a solution as well. We can set $\parallel \beta \parallel = 1/C$. The optimization problem is equivalent to:

$$\min_{\beta,\beta_0} \frac{1}{2} \parallel \beta \parallel^2$$

$$\text{subject to } \; y_i(\beta^T x_i + \beta_0) \geq 1, i = 1, ..., N$$

- This is a convex optimization problem.

- The Lagrange sum is:

$$L_P = \min_{\beta,\beta_0} \frac{1}{2} \parallel \beta \parallel^2 - \sum_{i=1}^{N} a_i [y_i(\beta^T x_i + \beta_0) - 1] \ .$$

- Setting the derivatives to zero, we obtain:

$$\begin{aligned} \beta &= \sum_{i=1}^{N} a_i y_i x_i \ , \\ 0 &= \sum_{i=1}^{N} a_i y_i \ . \end{aligned}$$

- Substitute into L_P, we obtain the Wolfe dual

$$L_D = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} a_i a_k y_i y_k x_i^T x_k$$

subject to $a_i \geq 0$.

This is a simpler convex optimization problem.

- The Karush-Kuhn-Tucker conditions require:

$$a_i[y_i(\beta^T x_i + \beta_0) - 1] = 0 \ \forall i \ .$$

  - If $a_i > 0$, then $y_i(\beta^T x_i + \beta_0) = 1$, that is, x_i is on the boundary of the slab.
  - If $y_i(\beta^T x_i + \beta_0) > 1$, that is, x_i is not on the boundary of the slab, $a_i = 0$.
- The points x_i on the boundary of the slab are called *support points*.
- The solution vector β is a linear combination of the support points:

$$\beta = \sum_{i:a_i>0} a_i y_i x_i \ .$$