COS 424: Interacting with Data

Lecturer: Dave Blei
Scribes: Wyatt Lloyd and Jeff Terrace

Lecture #5
February 19, 2008

1 Naive Bayes

In the last lecture, we discussed Naive Bayes, and we were left with this equation:

$$P(c, w_{1:N}|\pi, \theta) = P(c|\pi) \prod_{n=1}^{N} P(w_n|\theta_c) \tag{1}$$

and the following questions:

1. What's weird about this?
   This seems weird because it does not take into account the order of the words. It completely ignores the structure of the text (ie. bag of words).

2. What is the effect of (1)?
   The effect this has is that seeing one word many times increases its influence to the classifier. Because of this, spam e-mails that have non-spam words in them reduces the chance of the e-mail being classified as spam.

3. Can we adapt NB to other kinds of data?
   Yes, we can generalize this into a generic model. Naive Bayes only requires that you have a set of data with corresponding categories. It can then predict which category future data belongs to.

   For other kinds of data,

   - $\hat{\pi} \propto$ # instances of each class
   - $\hat{\theta_c}$: MLE of the data restricted to class c

1.1 General Probabalistic Classifier

The generic version of the classifier is as follows:

$$p(x, c|\pi, \theta) = p(c|\pi)p(x|\theta_c) \tag{2}$$

This allows us to predict the category of given data.

Suppose we have the data set $\{x_d, c_d\}_{d=1}^{D}$. In the previous example, x_d was the e-mail, and c_d was the category. For the generic model, x_d is the data, and c_d is still the category.

By using the chain rule, we get:

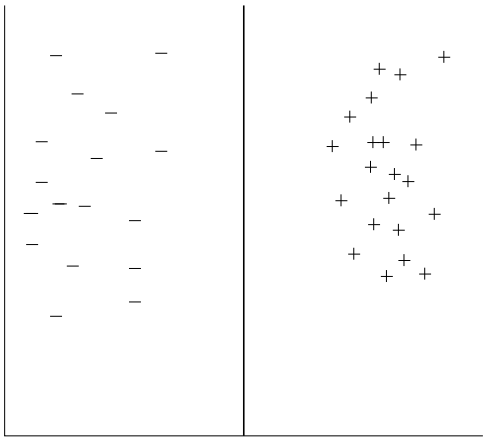
$$p(x_d, c_d|\pi, \theta) = p(c_d|\pi)p(x_d|c_d, \theta_c) \tag{3}$$

By computing the MLE for each partition (category), we use probability as a language to express uncertainty.

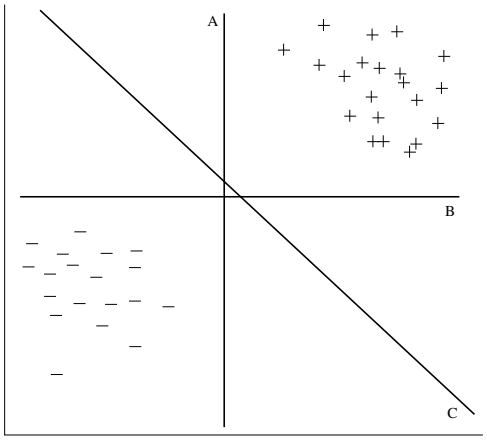
2 Support Vector Machines (SVM)

2.1 Introduction

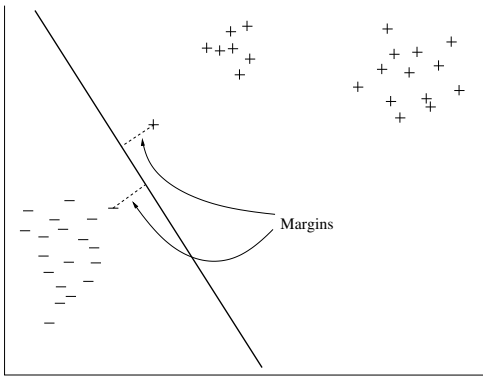
To introduce SVMs, we will first discuss the problem of placing a hyperplane in a linearly separable set of points that belong to two classes. The two classes are plus (+) and minus (-), and the classifier draws a line to separate the two classes. Below is a figure showing a set of points that belong to a class, and a line equidistant from the centers of the two clusters. This is one way to separate the clusters, but it becomes more complicated in other



situations. Consider the two clusters in the figure below. If all you care about is that the test points are separated, the three lines, A, B, and C are just three of the many valid lines. However, if you assume that the test points are in their locations for a reason, and



given that our goal is the classify new points, we can do better. Instead of choosing an arbitrary line, we will maximize the *margin* which is the minimum distance over all data to the boundary. The line we draw using this method is called the boundary, and is a type of *optimal hyperplane*. Note that the distance to the boundary from the nearest (+) and nearest (-) is equal, because the margin is maximized. In this figure, we show the margin from the nearest (+) and nearest (-).



2.2 Formalization

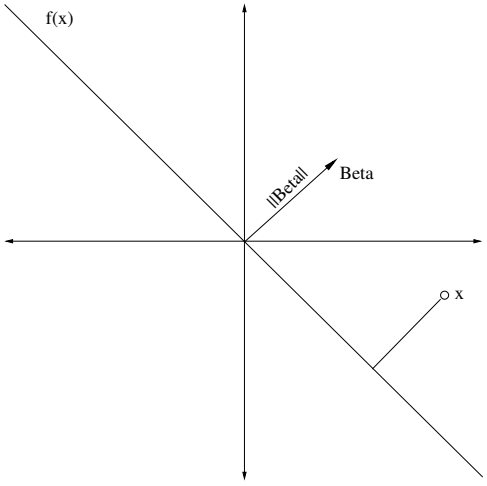
To formalize placing the hyperplane, we will first define our data as $\{(x_n, y_n)\}_{n=1}^{N}$ where $x_n \in REALS^p$ and $y_n \in -1, 1$.

A more formal definition of the margin, C, is

$$C = \min_n \frac{y_n x_n^T \beta}{\|\beta\|}$$

Problem: Maximize C s.t. $\frac{y_i x_i^T \beta}{\|\beta\|} \geq C$. This will find a line such that the distance to every point is greater than C and which mazimizes C.

As a review of linear algebra, the figure below shows $f(x) = \beta^T X$ which goes through the origin. Note that if the optimal hyperplane does not go through the origin, the data can be shifted so that it does. The distance from the point x_0 to the hyperplane is $\frac{\beta^T x_0}{\|\beta\|}$.



Problem: $\|\beta\|$ is free, so this is not a one-to-one relationship.

Solution: Set $C = \frac{1}{\|\beta\|}$

New Optimization: $\max_\beta \frac{1}{\|\beta\|}$ s.t. $y_i x_i^T \beta \geq 1$
This is the same as $\min_\beta \frac{1}{2}\|\beta\|^2$ s.t. $y_i x_i^T \beta \geq 1; i = 1, \ldots, N$
We do this because we can use the easy method for solving quadratic equations to solve our problem.

2.3 Optimization

We can use the Karush-Kuhn-Tucker (KKT) conditions, a generalization of Lagrange Multipliers, to maximize the margin. Let us examine the Lagrange Multiplier:

$$L_p = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} \alpha_i (y_i x_i^T \beta - 1) \tag{4}$$

We then take the derivative of this with respect to β_d:

$$\frac{\partial L_p}{\partial \beta_d} = \beta_d - \sum_{i=1}^{N} \alpha_i y_i x_{id} \tag{5}$$

and set the derivative to 0 which results in the formula for finding the optimal coefficient:

$$\beta = \sum_{i=1} N \alpha_i y_i x_i d \tag{6}$$

This is the first KKT condition. We then take the duality function and we get (with some algebra not shown here):

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{7}$$

The second KKT condition is to maximize L_D such that $\alpha \geq 0$.
The third KKT condition is $\alpha_i (y_i x_i^T \beta - 1) = 0$. If $\alpha_i \geq 0$ then the second term has to be 0, so x_i lies on the margin. If $y_i x_i^T \beta - 1 = 0$ then $\frac{y_i x_i^T \beta}{\|\beta\|} = \frac{1}{\|\beta\|}$.

2.4 Meaning Behind the Name

The effect of the α_is is that the only vectors that count are those which lie on the margin because α_i is 0 if x_i is not at the margin. These vectors that count are called the *support vectors*. Therefore, our intuition turns into formalization — only points at the margin matter.