COS 424: Interacting with Data

Lecturer: Dave Blei
Scribe: Xu Da Tan

Lecture #6
February 21, 2008

1 Review of Support Vector Machines

We are given $\{(x_n, y_n)\}_{n=1}^{N}$, where x_n are data points $\in R^P$ and y_n are class labels $\in \{-1, 1\}$

- We assume that the data is linearly separable
- Recall from the previous lecture that $f(x) = \beta^\top x$ where β is orthorgonal to $\beta^\top x = 0$
- For any point x, $\frac{\beta^\top x}{\|\beta\|}$ is the signed distance to $\beta^\top x = 0$
- The margin is given by $C = min_n \frac{y_n x_n^\top \beta}{\|\beta\|}$
- We want to maximize the margin: $max_\beta C$ s.t. $\frac{y_n x_n^\top \beta}{\|\beta\|} \geq C$
- This setup is equivalent to $min_\beta \frac{1}{2}\|\beta\|^2$ s.t. $y_n x_n^\top \beta \geq 1$, which is a convex optimization problem with linear constraints and therefore has a unique optimal solution
- The solution satisfies the **Karush-Kuhn-Tucker** conditions:
  1. $\beta = \sum_{n=1}^{N} \alpha_n y_n x_n$
  2. $\alpha_n > 0$
  3. $\alpha_n \left(y_n x_n^\top \beta - 1\right) = 0$
- We note that $\alpha > 0$ **iff** x_n is on the margin. These x_n are the support vectors - they "hold up" the hyperplane in the Euclidian space. All other points have Lagrange multipliers, $\alpha_n = 0$
- Given a new data point, we can classify it by: $y_{new} = sign\left(\beta^\top x_{new}\right)$

2 The Kernel Trick

2.1 Definition

A kernel is a simple function that corresponds to a dot product in a higher dimension space

2.2 An example

What do we do if the data is not linearly separable?

- Consider the transformation: $(x_1, x_2) \rightarrow^{\phi} \left(x_1^2, x_2^2, \sqrt{2}x_1 x_2\right)$
- We can therefore map the data to a higher dimension, fit the SVM, and then project the classified data back down

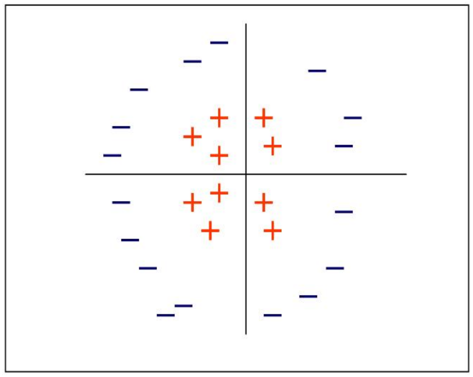

Figure 1: How do we place a line in the right way?

- However, this approach is expensive!
- From the previous lecture, we have $L_D = \sum_{n=1}^{N} \alpha_n - \frac{1}{n} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j x_i^\top x_j$
- Replacing the x's with $\phi(\cdot)$, we get $L_D = \sum_{n=1}^{N} \alpha_n - \frac{1}{n} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \phi(x_i)^\top \phi(x_j)$
- $\phi(x_i)^\top \phi(x_j) = \left(x_i^\top x_j\right)^2 = K(x_i, x_j)$, note that this does not work for all $\phi(\cdot)$
- Then, $L_D = \sum_{n=1}^{N} \alpha_n - \frac{1}{n} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j K(x_i, x_j)$

2.3 Example of kernels

- Polynomial kernel: $\left(1 + x_i^\top x_j\right)^d$
- Radio basic kernel: $exp\left\{\frac{-\|x_i - x_j\|^2}{C}\right\}$
- Note that any algorithm that relies only on a dot product between vectors can be kernelized
- To classify a new data point, observe that $\beta = \sum_{i=1}^{N} \alpha_n y_n \phi(x_n)$
- Then, $sign\left(\beta^\top \phi(x_{new})\right) = sign\left(\sum_{n=1}^{N} \alpha_n y_n \phi(x_n)^\top \phi(x_{new})\right)$
- Thus we can classify a new data point without going to a higher dimensional space!

2.4 Properties of the kernel method

The kernel method combines notions of:

1. Robustness - because of support vectors
2. Complexity - can look at complicated decision boundaries
3. Convex optimization

3 Boosting

3.1 Introduction

Consider the SPAM/HAM classification problem. We can come up with rough rules of thumb (*r.o.t*) to classify the data, such as:

- If it is only an image $\Rightarrow$ SPAM, otherwise $\Rightarrow$ HAM
- If it is from someone who has never emailed you $\Rightarrow$ SPAM, otherwise $\Rightarrow$ HAM
- More than 80% misspelt words $\Rightarrow$ SPAM, otherwise $\Rightarrow$ HAM
- Main idea - Boosting converts many *r.o.t* into a highly accurate predictor. The only requirement of dumb classifiers is that they do better than random

3.2 Sketch of algorithm

1. Devise a way to find a *r.o.t*
2. Run on a subset of your data
3. Obtain first *r.o.t*
4. Run the same procedure on a second subset of data
5. Repeat T times of this algorithm
6. Combine T *r.o.t* to obtain classifier

3.3 Terminology

- "Weak hypothesis" = *r.o.t*
- "Weak learner" = procedure for finding *r.o.t*
- "Weak learning assumption" = We can find a weak hypothesis with error $\frac{1}{2} - \gamma, \gamma \in (0, 1)$

3.4 Theorem

Boosting can drive the training error down to ϵ for any $\epsilon > 0$

3.5 Intuition behind boosting

- Empirically, boosting does very well in test error too
- The algorithm imposes a distribution over the data after the first *r.o.t* such that the data that is 'correct' is weighed less and the data that is 'wrong' is weighed more i.e. focus on the errors

3.6 An illustration of boosting

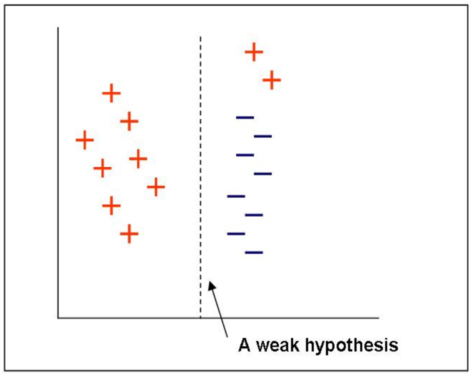

Figure 2: First rule of thumb

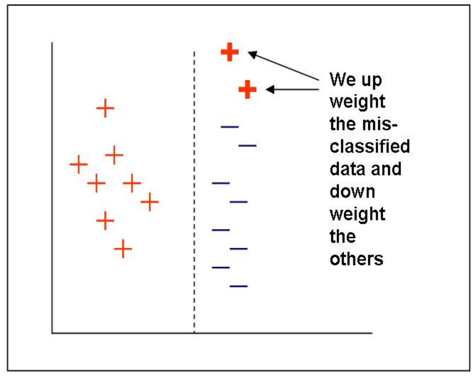

Figure 3: Reweighting the data points

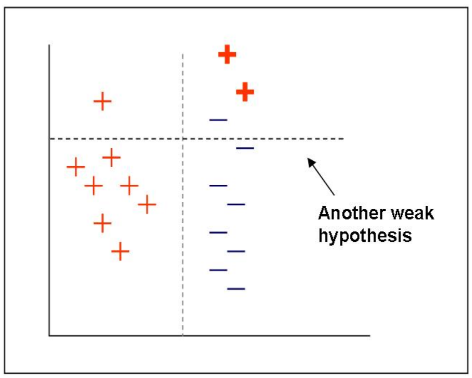

Figure 4: Second rule of thumb

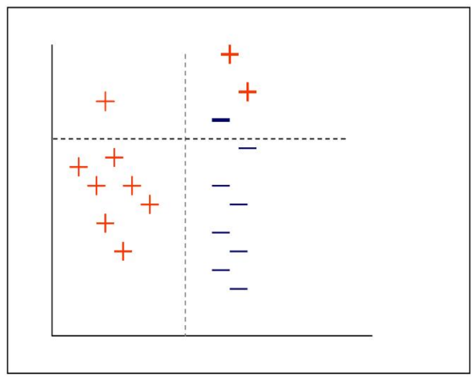

Figure 5: Reweighting the data points again