About this class

Separating hyperplanes (again)

The perceptron algorithm

The perceptron convergence theorem

Gradient descent

These notes are based on Bishop, Chapter 3, and Mitchell, Chapter 4.

The Perceptron

Weight vector $W =< w_0, w_1, \ldots, w_n >$ such that for an input $x_1, \ldots x_n$, predict $Y = 1$ if $w_0 + \sum_{i=1}^{n} w_i x_i > 0$ and $Y = -1$ (equivalently 0, but it's easier to think about it this way) otherwise

For notational convenience add an additional imaginary $x_0 = 1$ for every input so now the classifier is sgn$(W.X)$

Perceptron update rule:

$$w_i \leftarrow w_i + \Delta w_i$$

where

$$\Delta w_i = \eta(Y - W.X)x_i$$

For the perceptron criterion, η does not matter

Decision surface is a hyperplane. w_0 affects the distance from origin, but not the angle. The rest of w is perpendicular to the separating hyperplane

Similar to logistic regression – historically very important!

The Perceptron Convergence Theorem

Using capital letters for the entire vector, let R be $\max_t ||X_t||$

Suppose we repeatedly iterate through the data and perform the perceptron criterion update

Suppose the data are linearly separable. That is

$$\exists W^* Y_t * (W^*.X_t) = 1$$

The perceptron algorithm will converge to a linear separator

We have $W_{t+1} = W_t + (X_t.Y_t)$

After running the algorithm for some time, let's say that input (X_n, Y_n) has been misclassified τ_n times

$$W = \sum_n \tau_n X_n Y_n$$

$$\Rightarrow W^{*T}.W = \sum_n \tau_n W^{*T} X_n Y_n$$

$$\geq \tau \min_n W^{*T} X_n Y_n$$

where τ is now the total number of mistakes made. So we have established the boundedness below of the projection of W on W^*

Now the next step is to show boundedness above of W after τ errors

$$||W_{\tau+1}||^2 = ||W^\tau + X_n Y_n||^2$$

$$= ||W_n||^2 + 2Y_n(W_n.X) + ||X_n||^2$$

Since we made a mistake, $Y_n(W_n.X) < 0$, and we have $||X_n||^2 < R^2$, so

$$||W_{\tau+1}||^2 < ||W_n||^2 + R^2$$

Therefore, after τ updates

$$||W||^2 \leq \tau R^2$$

So the length of W increases no faster than $\sqrt{\tau}$, but it increases at least as fast as τ. Therefore, it must stop changing for sufficiently large τ

Things to think about:

1. Decrease in error is not monotonic!

2. Basic generalization result – this algorithm will converge after a finite number of errors. Therefore it *must* be able to generalize (assuming concept is in the hypothesis space)

Gradient Descent

Y is the target output. Let o be the unthresholded perceptron output. Define the error function $E(w) = \frac{1}{2}\sum_{d\in D}(Y_d - o_d)^2$.

Then

$$\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{1}{2}\sum_{d\in D}\frac{\partial}{\partial w_i}(Y_d - o_d)^2 \\
&= \frac{1}{2}\sum_{d\in D} 2(Y_d - o_d)\frac{\partial}{\partial w_i}(Y_d - o_d) \\
&= \sum_{d\in D}(Y_d - o_d)\frac{\partial}{\partial w_i}(Y_d - w\cdot x_d) \\
&= \sum_{d\in D}(Y_d - o_d)(-x_{id})
\end{aligned}$$

Then the increment in a step of gradient descent for the ith weight is given by:

$$\Delta w_i = -\eta\frac{\partial E}{\partial w_i} = \eta\sum_{d\in D}(Y_d - o_d)(x_{id})$$

Gradient descent converges asymptotically to the minimum error hypothesis (there is a unique error-minimizing weight vector). However, there is no finite-time guarantee of convergence. But it is guaranteed to converge, unlike training using the perceptron criterion.

Another point to note: gradient descent using this error function uses the unthresholded output, whereas the perceptron criterion uses the thresholded output.