CS Lecture Notes

Jeremy Nixon

February 5, 2015

Linear Regression Lecture 3

Machine learning is about continuous math, not discrete math, and so there is a lot of calculus.

1. Linear Regression

   Data: xn, tnNn=1

   Our data are tuples of x and t, a training value ad a target value.

   xnis in RD tn is in R

   X in RNxD

   x is a column vector of all of these features., from x1T to xNT. This is the desigin matrix. We also have a t vector, which is a column that represents our targets.

   Since one of these dimensions is one, we dont have to worry about an explicit representation of the bias.

   Well have a regression function

   y(w,x) = wTx. w and x are column vectors.

   ERROR

   ED(w) = sum from n = 1 to N of (tn-y(w, x-n))2

   The is convenient for after we take the derivative, it cancels the square.

   (t-Xw)T(t-Xw)

   So our t vector is Nx1, our x matrix is NxD, and our weight vector is Dx1

   We have a quantity t-Xw that is Nx1, and so were going to multiply.

2. Gradients

Weve defined this loss function and want to minimize the loss function as a function of W.

The gradient is the vector of derivatives in terms of each dimension. The gradient of F is the partial derivatives in terms of the weights.

Most people aren't completely comfortable with gradients.

We're going to review what we did last lecture, but more slowly.

$$\Delta_w E_D(w) = \tfrac{1}{2}\Delta w(t^T t - 2w^T x^T t + w^T x^T x_w)$$
$$= \tfrac{1}{2}\Delta w t^T t - \delta_N N^T x^T t + \tfrac{1}{2}\delta_w w x^T x_w$$

Rather than take the derivative with respect to w, we can take the derivative wirth respect to beta and solve for the distribution of the points around our regression line.

The w that we found didn't depend on beta. So we can just plug back in the w that's the MLE and then solve for Beta.

In general the MLE method and the least squres do not give you the same weights. It only works here because of log of our distribution happens to be the same.

So why do least squares? Why not always to MLE? MLE gives us a probability distribution. But it can be simpler to use least squares. You may not get any mileage through this beta stuff, and so it is easier and simpler to just use least wquares.

If you want to make a new prediction, take your new x and get the mean by plugging it in, and then use your beta function to determine the variance.

Basis Functions

Everything has the form $y(x, w) = w_0 + w_1 x$. But with linear regression we only get linear values on the weights. So what if we used a different basis function?

J Basis Functions

$y(x, w) = w_- + w_1 x + w_2 x^2 + w_2 x^3$

These basis functions are where engineering is done in machine learning - we're creating features that are important representations of the data.

Usually we refer to basis functions with ϕ.

$\phi x -> R$ $\phi(x) -> R$

This is the location of the kernel trick in SVMs, and the advantages we get with neural networks.

$x- > \phi$