Parametric Models

(PDF)

In this lecture, we discussed Parametric Models. We focused on determining the functional form of a model. Functional forms of models can be any polynomial to a degree of M. If M = 10, this would be a 10th order polynomial. What we want to do is learn the parameters or weights of the model using the training data, and once we have a feel for the relationship, we can determine what functional form of the model leads to the lowest prediction error.

We are able to able to create a generalized basis function that will convert a polynomial of any M to a basis function using a dummy variabel for for each parameter. For a linear model, the basis function expansion with a dummy variable will not look much different from the linear equation. However, for a polynomial, we are able to convert a paramter of x^i to $\phi_i(x)$.

For example:

The equation: y = $\beta_0 + \beta_1 x + \beta_2 x^2$

would be converted into the basis function: y = $\beta_0 + \beta_1\phi_1(x) + \beta_2\phi_2(x)$

Assumptions

Now that we have standardized the model to a basis function, there are some assumptions we can make about the model.

Assumptions: 1. We have an expected value of T for the prediction given the basis function vector ϕ is linear in ϕ. 2. All distributions around the expected values are assumed to be independent and identically distrubuted zero mean Gaussian with constant variance.

When we say that the expected values are independent and identically distributed, this means that the line of fit that we get from the model is the mean value of T with respect to each ϕ. Under this assumption, minimizing the Mean Squared Error on the training data yields the Maximum Likelihood Estimate (MLE) solution of the assumed generative model.

To visualize this, refer to IMAGE 1.

IMAGE 1

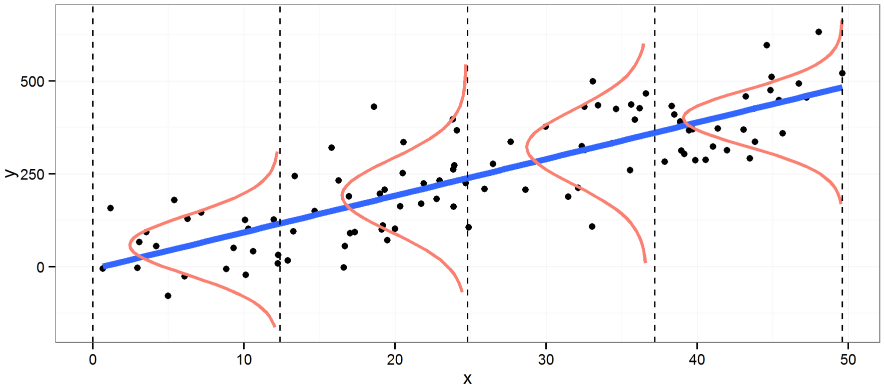

Figure 1: IID.png

Interpreting the Weights

The weights of a basis function are the same as βs of linear regressions in that they are coefficients of each predictor $\phi(x)$ when calculating an output t.

The partial derivative of each predictor helps to observe the minimum residual error. By minimizing the minimal error for each weight, we should in turn be able to obtain the model with the residual error.

In regressions, the error is minimized by looking at the RMSE(root mean square error).

When predictors are collinear, the weights for those predictors have more importance than they should because one variable is highly correlated with another rather than the actual target y.

Linear Regressions When changing to the basis form of a linear regression model, the weights of the function will be the βs of the linear regression. That is, they are the coefficients of each predictor that is trying to predict the output.

$w = \beta$

where β is the coefficient of the predictor X

from $Y = \beta_0 + \beta_1(X)_1 + \ldots + \beta_n(X)_n$

Polynomial: (with scalar x) When using the basis functions as a polynomial of a scalar x, the basis functions look like:

$\phi_i(x) = x^i$

With polynomials as the basis functions, the weights w are now coefficients of the polynomial basis function rather than the individual predictors from linear regression.

$y = w_0 + w_1(x)^1 + ... + w_M(x)^M$

Over Fitting and the Bias-Variance Tradeoff

A problem that arises in predictive modeling is the issue of selecting the complexity of a certiain model. For example, a general polynomial of the form:

$y(x, \mathbf{w}) = \beta_0 + \sum_{i=1}^{n} \beta_i x^i$

When we fit a predictive model of this form onto our data, we will find that the model becomes increasingly more complex as n increases and will have the ability to capture more details of our dataset. However, this becomes a problem when n becomes sufficiently large (i.e. when n is exactly equal to the number of data points).

As the model becomes increasingly complex, the model begins to "overfit" on the data, or becomes too dependent on the training dataset in question. When this occurs, the model essentially loses predictive power, as any additional observation would likely be considered an outlier to the model. The calculated RSS would likely be higher as the model becomes more complicated.

However, increasing the value of n is necessary in order to capture the variance of y to some extent. Consider the model:

$y(x, \mathbf{w}) = w_0$

This model will be optimized exactly where $w_0 = E[y|X_1, X_2...]$, and will result in an underfit.

This could also lead to some problems, as the model becomes too simple to capture the variances in the data. The resulting RSS will also be large, but for a different reason, as the prediction will always be the mean of the y values.

This is what is known as the bias-variance tradeoff. The optimal number of parameters can be found by plotting the RSS against the number of parameters, which will be lowest at the point where $bias^2$ and $variance$ intersect.

It is also important to note that if n (the number of parameters) is exactly the number of data points, the training error will be exactly 0. The polynomial will be flexible enough to go through each data point and predict each value with 100% accuracy. The testing error, as before, will be abnormally high (if not the highest).