Transductive inference and the Rebalancing approach

Shobhit Verma

November 19, 2008

Abstract

The justification of using parametric regression techniques (like Linear, Polynomial, Neural networks etc.) comes from the close relationship between the regression estimates and the maximum likelihood estimates. However, it is common to use regression.

1 Rebalancing the regression in the Transductive setting

The justification of using parametric regression techniques (like Linear, Polynomial, Neural networks etc.) comes from the close relationship between the regression estimates and the maximum likelihood estimates. However, it is common to use regression for choosing the "best" function approximation from a set of points related by an unknown (deterministic) function.

In the canonical setting when the regression is equivalent to the maximum likelihood estimation, we shall see that the scatter of the independent variables does not effect the estimates. However we show that this does not remain true in the latter case and introduces a kind of "lobbying effect". We will note that depending on our interests (derived from the domain of applictaion), the "lobbying effect" may or may not be desirable. In particular, such a lobbying is undesirable in the general transductive setting. Consequently, we describe the "rebalancing approach", which tunes the lobbying so that it works in our best interest.

2 Function approximation

In this section, we describe an alternate setting where regression is commonly used. Suppose X is a k dimentional vector of real values and Y is a real value, which is an unknown (deterministic) function of X. Note that unlike the previous case, when we had access to a family of candidate functions, here we have no information on the form of the fuction. Formally, let

$$X \in \mathbb{R}^k$$

$$Y = f(X)$$

$$f() : \mathbb{R}^k \to \mathbb{R}$$

where the form of $f()$ is unknown. We are given the values of pairs X_i and Y_i for $i = 1 \cdots n$ which are n arbitrary instances of the above relation. We want to estimate $f()$.

To find $f()$ in practice, it is common to take a parametric family $h(\gamma, x)$ of functions and choose the parameter γ which minimizes the expression

$$\sum_{i=1}^{n} (y_i - h(\gamma, x_i))^2$$

In the appendix we give intuitions supporting the application of regression in the setting of function approximation and express the need to introduce new formalities which make the problem well-posed. In the following section we demonstrate the "lobbying effect" which is expressed in function approximation when the parametric family of regression functions $h(\gamma, x)$ is not sufficiently complex to approximate the actual function.

3 The lobbying effect

Suppose the scatter of the learning set X_i is such that some regions from the domain D have more representatives than others, it is easy to see that the "best fit" functional is "pulled" more strongly towards the (X, Y) pairs in the denser region than towards the other pairs. This effect can be best understood by artificially pronouncing it as follows..

Given a set of pairs (X_i, Y_i) for $i = 1 \cdots n$, define $(X_i, Y_i) = (X_1, Y_1)$ for $i = n + 1 \cdots 2n$.

This defines a "new" dataset , however minimizing the sum of squared errors in this dataset is equivalent to minimizing a weighted sum of squared errors for the old dataset with the weight of the first error equal to the sum of weights of all other errors ! Thus, due to over representation in the new sample, (X_1, Y_1) pulls the "best fit" estimate with greater force than it ought to. We call this effect, the "lobbying effect". In actual practice, as f is "simple" and continuos, the values of Y for points in the neighbourhood of X_1 would be close to each other. Therefore we could have observed a similar "lobbying effect", had we defined the new dataset as $(X_i, Y_i) = (Z_i, f(Z_i))$ for $i = n + 1 \cdots 2n$, where Z_i s are some points in the neighbourhood of X_1. Thus over-representaion of the region around X_1 gives rise to a similar "lobbying effect".

Note that depending on our "interest" (The pdf $w(x)$: see Appendix B), such a lobbying may or may not be good. In particular, if X actually comes from the distribution with pdf $w(x)$, lobbying is good. However X may indeed come from a different distribution, or worse, X may simply be non-deterministic, but not necessarily random. In the simulated example, the interest was "flat" on all values of X, therefore the lobbying had a bad effect on the estimate, which was corrected by the "rebalancing approach".

4 Rebalancing approach

We saw that when we give equal weight to approximation errors at all examples, due to the lobbying effect, the approximating function is pulled nearer to regions having more representation in the set of examples. The "rebalancing" approach reduces this effect by instead minimising a weighted loss, where the weight of an approximation error at a point from a dense region is lesser than the weight of approximation error at a point which is away from them. This naturally induces a notion we call "loneliness" of a point, which will be the weight of the approximation error at that point. Intuitively, points belonging to a denser region in the scatter of the example are less lonely than points away from them.

A criterion for loneliness

There can be many defintions capturing the notion of "loneliness". One such approach is to treat X as if it came from a distribution with pdf $p(x)$, intuitively we could define $loneliness(x_i)$ as some decreasing function of $p(x_i)$. For the example simulated later in the article, we define $loneliness(x_i) = \frac{1}{\hat{p}(x_i)}$, where $\hat{p}(x)$ is the kernel density estimate of the scatter of X, using gaussian kernels with σ^2 variance. The paramter σ captures how far two points have to be so that the values of the function at them are very different.

In general, we minimize the weighted sum of losses with the weight of a loss at x_i, defined as $weight(x_i) = loneliness(x_i) * w(x_i)$, so that we retain the "good" effect of lobbying when X in the training set actually comes from the distribution with pdf $w(x)$.

Example 1

We simulated an experiment by taking 1000 observations from a deterministic function defined on $\mathbb{R}^2$

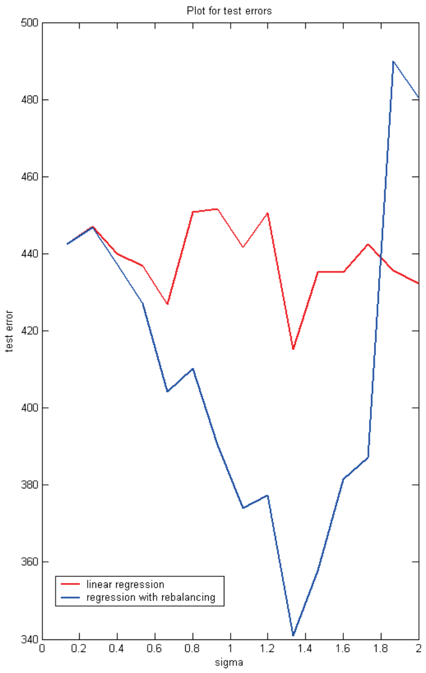
$$Y = f(x) = x_1 + x_2 + x_1x_2 + 2sin(\frac{x_1x_2}{5}) + 5$$

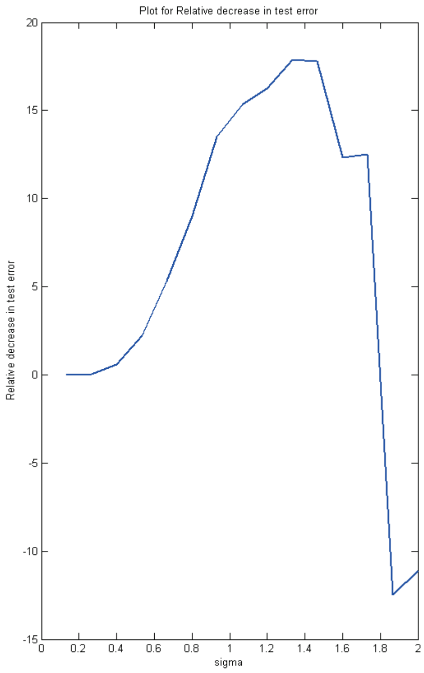
From the observed data, we tried to get a close approximation of f using a polynomial of degree at most two. Formally

$$h(\gamma, x) = \gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 x_1 x_2 + \gamma_4 x_1^2 + \gamma_5 x_2^2 + \gamma_6$$

We calculated the sum of squared approximation errors on a grid of 900 points uniformly scattered on the region $[0, 5] \times [0, 5]$ as a measure of goodness of fit, lets call this quantity the "test error". We re-did the above approximation on the same data by the "rebalancing approach" as proposed in the article and compared the "test error" of both the methods for various values of σ. We repeated this simulation 75 times and the average value of "test error" with and without using the rebalancing approach is plotted in the left figure.

The figure to the right plots the percentage reduction in the test error by using the rebalancing approach for different values of σ. Note that the percentage reduction in test error becomes negative for $\sigma > 1.7$. This is due to the





fact that for high values of σ, the kernel is not suitable for the kernel density estimate (KDE) to be close to the actual density of X. This is a limitation of the KDE scheme itself and should not be treated as a failure of the rebalancing approach.

Appendix A: The canonical setting

Suppose X is a k dimentional vector of real values and Y is a real value, which is a function of X upto some error coming from a known family of distributions. If we happen to know a family of functions relating X and Y, we can choose the best function by appealing to the principle of maximum likelihood estimation. Formally, let

$$X \in \mathbb{R}^k$$

$$Y = f(\Theta, X) + \epsilon$$

where $f(\Theta,)$ is a class of functions parameterised by Θ, which is a $q-$dimensional real vector.

$$f() : \mathbb{R}^q \times \mathbb{R}^k \rightarrow \mathbb{R}$$

and ϵ is a random variable with a known parametric family of pdfs.

$$\epsilon \sim g(\beta, x)$$

We are given the values of pairs X_i and Y_i for $i = 1 \cdots n$ which are n independent instances of the above relation. We want to find the "best" value of Θ from this information. We can calculate the likelihood $L(\Theta, \beta)$ of the given sample as

$$L(\Theta, \beta) = \prod_{i=1}^{i=n} g(\beta, (Y_i - f(\Theta, X_i)))$$

Following the maximum likelihood approach, we would like to estimate Θ and β as the values which maximize the likelihood $L(\Theta, \beta)$. It is possible that $L(\Theta, \beta)$ is unbounded in Θ and β, in which case the maximization is not well defined. However if it so happens that the value of Θ which maximizes $L(\Theta, \beta)$ for a fixed value of β does not depend on β, we would like to consider this value of θ, a valid maximum likelihood estimate.

In particular, when the error, ϵ are assumed to be gaussian with mean zero, the MLE of θ is same as the estimate minimizing the sum of sqared prediction errors (SSE).

$$\hat{\Theta} = argmin(\sum_{i=1}^{n}(Y_i - f(\Theta, X_i))^2)$$

Note that in this procedure, the pdf $g()$ dictates the loss function which needs to be minimized on the sample, for example, if the error ϵ happens to come from a zero centered family of double exponential distributions, we must minimize the sum of absolute prediction errors to find the MLE of β (as opposed to minimising the SSE).

Note that if we prefer to make less error around some specific regions of interest at the cost of making more error at other values of X, there is no way to induce such a preference into the MLE estimate with homoscedastic errors. This is a consequence of the fact that all "prediction errors" come from the same distribution, hence contribute similarily to the likelihood. In other words, deviations of equal magnitude at different X values have the same contribution to the likelihood. Thus, the scatter of X has no extra effect on the value of the estimate.

Appendix B: Regression for Function Approximation

If we follow the problem of function approximation as stated in the earlier, there doesn't seem to any reasonable way of finding $f()$, as clearly there are uncountably many such functions and theoretically, we can output any one of them.

This requires us to define a notion of "utility" of the estimated function, so that we can redefine our objective as finding the most useful estimate of $f()$. To make this objective attainable, we might need to make resonable assumptions on the form of $f()$.

In the canonical setting, our objective was very well defined as we knew that the actual function belonged to the given parametric family of functions and we could state the objective as that of finding the “actual” value of the parameters. However, in the setting of function approximation, we can try to find the most useful value of the parameters, based on some definetion of “utility” for a value of parameters.

One possible definition for utility is that the difference between $\hat{f}(x)$ and $f(x)$ must be “small”. That is, we should be able to approximately compute the value of $f(x)$ at values of x not necessarily in the training sample. To make this an attainable objective, we assume that the form of $f()$ is not too “complex”, i.e. it is possible to approximate f upto some error using “simple” functions. There is no universal notion of complexity, however most parametrized classes of functions (polynomials, splines, neural nets) are considered “simple”.

To disambiguate this criterion further, we need to quantify the notion of “small”, which introduces the “Loss function”, $Loss(x_1, x_2)$, which is a real number capturing the significance of the error, when the actual value of x_1 is predicted as x_2. A common choice for this is the squared error loss function, $Loss(x_1, x_2) = (x_1 - x_2)^2$, as it is symmetric and gives rise to analytically tractable optimization problems.

Still, we may not be able to simultaneously minimize the loss at every unknown x, by choosing one single function from the family. Therefore in this setting it is required to define one single quantity which needs to be minimized. This quantity may depend on the value of losses at all values of X, in the domain D. Generally it would look like

$$Total\ Loss = \int_{x \in D} Loss(x, \hat{f}(x)) w(x) dx$$

Where $w(x)$ is the weight of loss at x. When the domain D of x is finite, we can make $w(x)$ same for all values of x, however this may make the objective infinite for cases when D is of unbounded measure, so generally we would assume $w(x)$ to be a probability distribution function(pdf).

$$\int_{x \in D} w(x) dx = 1$$

With this assumption, it is possible to “interprete” the objective as though we want to minimize the expected loss when the test x is randomly chosen from D based on the pdf $w(x)$.

The loss function to be used in function approximation really comes from the domain of application and captures the “loss” as percieved by the user, whereas in the canonical setting, the loss function has to come from the randomness inherent in the data. Also, in the canonical setting, the objective is unambiguously well defined as that of finding an estimate for the “actual” value of the parameters, whereas in function approximation, even to make sure that the problem is well-posed, we need to define a preference over losses based on the value of x at which it occurs, in form of specifying the pdf $w(x)$. Unfortunately, in real world applications as in the case of the simulated example, the need for specifying $w(x)$ is often overlooked. In such cases, we need to have a "default" $w(x)$ to use. We propose to use $w(x)$ as a constant over a large $k-$ dimensional rectangle supposedly containing all "interesting" values of x. Let D_r be the intersection of this rectangle with the domain D of X. The proposed objective is then to minimize

$$Total\ Loss = \int_{x \in D_r} Loss(x, \hat{f}(x))w(x)dx$$

In the simulated example, the quantity defined as "test error" approximates the above with D_r as the two dimensional rectangle $[0,5] \times [0,5]$. We saw that the "re-balancing" approach did seem to yeild better results.