TTIC 31020 : Problem Set 1

Maria Hyun

October 15, 2015

Problem1

Show that for any $a \in \mathbb{R}^{d+1}$

$$E_{p(x,y)}[(y - w^{*T}x)a^T x] = 0$$

$w^* = \arg\min_w \sum_{i=1}^{N}(y_i - w^T x_i)^2 \Rightarrow L(w, X, y) = L(w) = \frac{1}{N}\sum_{i=1}^{N}(y_y - w^T x_i)^2$

Since w^* is optimal solution

$$\frac{\partial L(w, X, y)}{\partial w} = 0$$

Derivatices w.r.t $w_0, w_1, ..., w_d$ must all be zero

$L(w) = \frac{1}{N}\sum_{i=1}^{N}(y_i - w^T x_i)^2$

$$\frac{\partial}{\partial w_0}L(w) = \frac{1}{N}\sum_{i=0}^{N}\frac{\partial}{\partial w_0}(y_i - w^T x_i) = 0$$

It apply that $\sum_{i=0}^{N}(y_i - w^T x_i) = 0$

$E_{p(x,y)}[(y - w^{*T}x)a^T x] = \int\int((y - w^{*T}x)a^T x)\, p(y|x)p(x)dydx = \int E_{y|x}[y_i - w^T x_i](a^T x)p(x)dx = 0$

Problem2

Explain how the original statement, regarding zero correlation between prediction errors made by $\hat{w}$ estimated by least squares with any linear function of the training x_i, follows from $E_{p(x,y)}[(y-w^{*T}x)a^Tx] = 0$

Naturally, any model is highly optimized for the data it was trained on. The expected error the model exhibits on new data will always be higher than that it exhibits on the training data. As example, we could go out and sample 100 people and create a regression model to predict an individual's happiness based on their wealth. We can record the squared error for how well our model does on this training set of a hundred people. If we then sampled a different 100 people from the population and applied our model to this new group of people, the squared error will almost always be higher in this second case.

It is helpful to illustrate this fact with an equation. We can develop a relationship between how well a model predicts on new data (its true prediction error and the thing we really care about) and how well it predicts on the training data (which is what many modelers in fact measure).

True Prediction Error=Training Error+Training Optimism

Here, Training Optimism is basically a measure of how much worse our model does on new data compared to the training data. The more optimistic we are, the better our training error will be compared to what the true error is and the worse our training error will be as an approximation of the true error.

Training error almost always UNDER estimates test error, sometimes dramatically.

Training error usually UNDER estimates test error when the model is very complex (compared to the training set size), and is a pretty good estimate when the model is not very complex.

However, it's always possible we just get too few hard-to-predict points in the test set, or too many in the training set. Then the test error can be LESS than training error, when by chance the test set has easier cases than the training set.

Problem3

Let $\hat{w}$ be the least square estimate of the regression parameter from the unscaled X, and let $\hat{\tilde{w}}$ be the solution obtained from the scaled X. Show that the scaling does not change optimality, in the sense that $\hat{w}^T x = \hat{\tilde{w}}^T \tilde{x}$.

First, note that scaling can be represented as a linear operation C composed of the scaling factors along its main diagonal.

$$C = \begin{bmatrix} 1 & & & \\ & c_1 & & \\ & & \ddots & \\ & & & c_d \end{bmatrix}$$

Using the scaling Operator C, we can express the scaled inputs $\tilde{x}$ and design matrix $\tilde{X}$ as function of x and X respectively.

$$\tilde{x} = Cx \qquad \tilde{X} = XC \tag{1}$$

As was demonstrated in class, under the Gaussian noise model, the ML estimate of the regression parameters in given by

$$\hat{x} = (X^T X)^{-1} X^T y$$

Using the expression in Equation 5, we find

$$\hat{\tilde{w}} = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T y = ((XC)^T XC)^{-1} (XC)^T y = (C^X X^T XC)^{-1} C^T X^T y$$

At this point, we can apply the following matrix identity: if the individual inverses A^{-1} and B^{-1} exits, then $(AB)^{-1} = B^{-1}A^{-1}$. Since C is real, symmetric square matrix, C^{-1} must exist. Similarly, $X^T X$ is real, symmetric square matrix so $(X^T XC)^{-1}$ must also exist. As a result, we can apply the matrix identity to the previous expression.

$$\begin{aligned} A &= ((C^T)(X^T XC))^{-1} C^T X^T y \\ &= (X^T XC)^{-1} (C^T)^{-1} C^T X^T y \\ &= C^{-1} (X^T X)^{-1} X^T y \\ &= C^{-1} \hat{w} \end{aligned} \tag{2}$$

To prove that the scaled solution is optimal, we apply Equation 1 and w as follows. Recall that $(A^{-1})^T = (A^T)^{-1}$. As a result, $(C^{-1})^T = (C^T)^{-1}$. Since C is a real, symmetric matrix, $C = C^T$ and, as a result, $(C^{-1})^T C = I$.

$$\hat{w}^T x = \hat{\tilde{x}}^T \tilde{x}$$

Problem4

we can solve for the least squares polynomial regression coefficients $\hat{w}$ by using the extended design matrix $\tilde{X}$ such that

$$\hat{w} = (\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T y, \text{ where } \tilde{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^d \\ 1 & x_2 & x_2^2 & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^d \end{bmatrix}$$

where d is the degree of the polynomial and $x_1, ..., x_N$ and y are the observed points and their associated labels, respectively. To prevent numerical round-off errors, this method was applied to the column-normalized design matrix $\tilde{X}$ using degexpand.m within the main script prob5.m (as discussed in Problem 2). The resulting linear and quadratic polynomials are shown in Figure 2(a). The fitting parameters obtained using all data points and up to a fourth-order polynomial are tabulated below.

Polynomial Degree	Empirical Loss	Log-likelihood	10-fold Cross Validation Score
Linear (d = 1)	1.057	-529.5	1.063
Quadratic (d = 2)	0.930	-506.1	0.943
Cubic (d = 3)	0.930	-506.1	0.947
Quartic (d = 4)	0.925	-505.1	0.950

Note that the empirical loss L_N is defined to be the average sum of squared errors as given by Equation 9. The log-likelihood of the data (under a Gaussian model) was derived in class on 9/11/06 and is given by Equation 7 as $l(Y; w, \sigma) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - f(x_i : w)^2 - N\log(\sigma\sqrt{2\pi})$ where $\sigma \to \hat{\sigma}_{ML}$ is the maximum likelihood estimate of $\frac{3}{4}$ under a Gaussian noise model (as given by Equation 8 in Problem 3). At this point, we turn our attention to the model-order selection task (i.e., deciding what degree of polynomial best-represents the data). As discussed in class of 9/13/06, we will use 10-fold cross validation to select the best model. First, we partition the data into 10 roughly equal parts (see lines 65-70 of prob5.m). Next, we perform 10 sequential trials where we train on all but the ith fold of the data and then measure the empirical error on the remaining samples. In general, we formulate the k-fold cross validation score as

$$\hat{L}_k = \frac{1}{N}\sum_{i=1}^{k}\sum_{j \in fold\, i}(y_i - f(x_j; \hat{w}_i)^2$$

where $\hat{w}_i$ is fit to all samples except those in the i^{th} fold. The resulting 10-fold cross-validation scores are tabulated above (for up to a fourth-order polynomial). Since the lowest cross-validation score is achieved for the quadratic polynomial we select this as the best model.

In conclusion, we ¯nd that the quadratic polynomial has the lowest cross-validation score. However, as shown in Figure 2(b), it is immediately apparent that the Gaussian noise model does not accurately represent the underlying distribution; more specifically, if the underlying distribution was Gaussian, we'd expect half of the data points to be above the model prediction (and the other half below). This is clearly not the case for this example motivating the alternate noise model we'll analyze in Problem 6.

Problem5

Part 1: Consider the noise model $y = f(x; w) + v$ in which v is drawn from $p(v)$ as follows.

$$p(v) = \begin{cases} e^{-v} & if\, v > 0, \\ 0 & otherwise \end{cases}$$

Perform 10-fold cross-validation for linear and quadratic regression under this noise model, using exhaustive numerical search similar to that used in Problem 4. Plot the selected model and report the empirical loss and the log-likelihood under the estimated exponential noise model.

let's begin by defining the distribution of the label y, given the input x.

$$p(y|x, w) = \begin{cases} \exp(-(y - f(x; w)))-v & if\, y > f(x; w), \\ 0 & otherwise \end{cases}$$

Once again, we assume that the observations are i.i.d. such that the likelihood P is given as follows.

$$P(Y; w) = \prod_{i=1}^{N} p(y_i|x_i, w) = \prod_{i=1}^{N} \begin{cases} \exp(f(x_i; w) - y_i) & if\, y_i > f(x; w), \\ 0 & otherwise \end{cases}$$

The log-likelihood l is then given by

$$l(Y; w) = \log P(Y; w) = \sum_{i=1}^{N} \begin{cases} \exp(f(x_i; w) - y_i) & if\, y_i > f(x_i; w), \\ \infty & otherwise \end{cases}$$

since the logarithm is a monotonic function and $\lim_{x\to\infty} \log x = -\infty$.The corresponding ML estimate for w is given by the following expression.

$$\hat{w}_{ML} = \arg\max_{w}(Y; w) = \arg\min_{w} \sum_{i=1}^{N} \begin{cases} \exp(y_i - f(x_i; w)) & if\, y_i > f(x_i; w), \\ \infty & otherwise \end{cases}$$

Note that Equations 10 and 11 prevent any prediction $f(x_i; w)$ from being above the corresponding label y_i. As a result, the exponential noise distribution will effectively lead to a model corresponding to the lower-envelope of the training data. Using the maximum likelihood formulation in Equation 12, we can solve for the optimal regression parameters using an exhaustive search (similar to what was done in Problem 4). This approach was applied to all the data points on lines 19-72 of prob6.m. The resulting best-fit linear and quadratic polynomial models are shown in Figure 3(a). The fitting parameters obtained using all the data points and up to a second-order polynomial are tabulated below.

Polynomial Degree	Empirical Loss	Log-likelihood	10-fold Cross Validation Score
Linear (d = 1)	2.837	-487.5	2.837
Quadratic (d = 2)	1.901	-359.1	1.900

Note that the empirical loss L_N was calculated using Equation 9. The log-likelihood of the data (under the exponential noise model) was determined using Equation 11. Model selection was performed using 10-fold cross-validation as in Problem 5. The resulting

scores are tabulated above. Since the lowest cross-validation score was achieved for the quadratic polynomial we select this as the best model and plot the result in Figure 3(b). Comparing the model in Figure 3(b) with that in Figure 2(b), we conclude that the quadratic polynomial under the exponential noise model better approximates the underlying distribution; more specifically, the quadratic polynomial (under an exponential noise model) achieves a log-likelihood of -359.2, whereas it only achieves a log-likelihood of -506.1 under the Gaussian noise model. It is also important to note that the Gaussian noise model leads to a lower squared loss (i.e., empirical loss), however this is by the construction of the least squares estimator used in Problem 5. This highlights the important observation that a lower empirical loss does not necessarily indicate a better model -this only applies when the choice of noise model appropriately models the actual distribution.

Part 2: Now evaluate the polynomials selected under the Gaussian and exponential noise models for the data in 2005. Report which performs better in terms of likelihood and empirical loss.

Noise Model	Empirical Loss	Log-likelihood
Gaussian (d = 2)	2.837	2.837
Exponential (d = 2)	1.901	1.900

In both Problems 5 and 6.1, the quadratic polynomial was selected as the best model by 10-fold cross validation. In order to gauge the generalization capabilities of these models, the 2004 model parameters we used to predict the 2005 samples. The results for each noise model are tabulated below.

In conclusion, we find that the Gaussian model achieves a lower empirical loss on the 2005 samples. This is expected, since the Gaussian model is equivalent to the least squares estimator which minimizes empirical loss. The exponential model, however, achieves a significantly higher log-likelihood - indicating that it models the underlying data more effectively than the Gaussian noise model. As a result, we reiterate the point made previously: low empirical loss can be achieved even if the noise model does not accurately represent the underlying noise distribution.