The University of Texas at Austin
Department of Electrical and Computer Engineering

EE380L: Data Mining — Spring 2016

Problem Set Two, Part II

Constantine Caramanis Due: Wednesday, March 9, 2016.

This is Part II of Problem Set 2. This part focuses on probability and on machine learning/statistics.

Statistics and Machine Learning

In these next exercises, we explore the important question of how much we can trust the output of regression, and go over the *standard error*, which we discussed in the last class. We also investigate another concept we discussed in class: changing the regression problem by not only trying to minimize the squared losses (the squared errors of the fit), but also adding a penalty. We discussed two forms of penalties. When the penalty is a multiple of the sum of squares of β_i, then we have what is called *Ridge Regression*. When we penalize the sum of the absolute values of the β_i, then we have what is called *LASSO*. In general, adding such a penalty is called *regularization*.

1. **Linear Regression and Ordinary Least Squares**.

   Download and load into Python the data in `cars2010`, `cars2011`, `cars2012` – you'll find these in canvas zipped in the file `FuelEfficiency.zip` in the Problem Sets folder.

   These data sets contain information about fuel efficiency obtained from the U.S. Department of Energy?s Office of Energy Efficiency and Renewable Energy and the U.S. Environmental Protection Agency, by way of Kuhn and Johnson's book, Applied Predictive Modeling.

   (a) Which one is a good candidate for the *dependent* variable? And which are the continuous variables, and which are the categorical variables?

   (b) Using the data in the `cars2010` data set, find the best (in the the OLS (ordinary least squares) sense) linear fit for `FE` on `EngDispl`, and plot your solution. Report the R^2 value.

   (c) Plot the residual errors. Now sum the residual errors. What do you find?

   (d) Now do the same where you fit a linear regression for `FE` on both `EngDispl` and `NumCyl`. Report the R^2 value. Sum the residual errors. What do you find?

   (e) Now solve the OLS regression problem for `FE` against all the variables. Report the R^2 value. Sum the residual errors. What do you find?

2. **Regression and Standard Errors**

   In class we spent a while discussing how much we can trust the output of a particular regression. That is, in the simple regression case, we solve for $\hat{\beta}_0$ and $\hat{\beta}_1$ and propose:

   $$y = \hat{\beta}_0 + x\hat{\beta}_1.$$

   This means that we believe that a 1 unit increase in x is associated with a $\hat{\beta}_1$-unit increase in y. But how true is this?

If we were to repeat the experiment with a different (but statistically similar) population, we would expect to get somewhat different results.

(a) Assuming that 2010, 2011 and 2012 cars are "statistically similar," let's check this: compare the intercept and slope values you found in your regression of `FE` on `EngDispl` using 2010 data, with the intercept and slope values you find in a regression of `FE` on `EngDispl` using 2010 data. What do you find?

The burning question is... if $\hat{\beta}_0$ and $\hat{\beta}_1$ vary from one population to the next even when we believe we are fitting the same thing, when (and how) can we trust the relationships we find? Remember that two key goals of regression are explaining a relationship, and *prediction*. That is, we want to understand how one variable is related to another, and also we may want to use this relationship to predict the fuel efficiency of some car we have not tested, based on certain measurements. And in order to have a sense of how good our explanation or prediction is, we need to know how much we would expect our solutions $\hat{\beta}_0$ and $\hat{\beta}_1$ (and other $\hat{\beta}_i$ in the multiple regression setting) to change if we were to solve the problem again on a different population.

We want to understand this variation. We can do this using the *normal equation*, and an assumption about *normally distributed errors*. There is an explanation of this that we gave in class in the `Understanding Regression.ipynb` IPython notebook. But let's see it again here in a little more detail.

The Normal Errors Assumption: The true model is

$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + \cdots + x_p\beta_p + w,$$

where $w \sim N(0, \sigma^2)$, i.e., w is a Gaussian random variable with mean 0 and variance σ^2, i.e., standard deviation equal to σ (recall that the standard deviation is the square root of the variance). This means that the i^{th} data point that we see is generated by the relationship

$$y_i = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + \cdots + x_{ip}\beta_p + w_i,$$

where $w_i \sim N(0, \sigma^2)$.

In matrix notation, this reads:

$$\boldsymbol{y} = \mathbf{1}\beta_0 + X\beta + \boldsymbol{w},$$

where $\mathbf{1}$ is the all ones vector, and $\boldsymbol{w}$ is a random vector, whose component are *independent and identically distributed* according to the normal distribution given above.

It makes the notation (and the following discussion) a little more simple if we put the all ones vector as an additional column in X, and then write β to denote the old β as well as β_0, so that the above then becomes:

$$\boldsymbol{y} = X\beta + \boldsymbol{w}.$$

The fact that $\boldsymbol{w}$ is a random vector, whose component are *independent and identically distributed* according to $N(0, \sigma^2)$, means that the components of $\boldsymbol{w}$ are all *mutually independent*, and

$$\mathbb{E}[w_i] = 0, \qquad \mathrm{Var}(w_i) = \mathbb{E}[w_i^2] - \mathbb{E}[w_i]^2 = \mathbb{E}[w_i^2] = \sigma^2.$$

(b) Use the exercise from the probability section below, to show that

$$\mathrm{Cov}(\boldsymbol{w}) = \mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\top] = \sigma^2 I,$$

where I denotes the identity matrix:

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{bmatrix}.$$

The Normal Equation: Recall that in class we gave what is called the *normal equation* for $\hat{\beta}$. The OLS solution $\hat{\beta}$ is given by the solution to the optimization problem

$$\min_{\beta} : \ \|X\beta - \boldsymbol{y}\|_2^2,$$

and the solution can be given in closed form:

$$\hat{\beta} = (X^\top X)^{-1} X^\top \boldsymbol{y}.$$

Remark: deriving this is not too difficult – it just takes a little vector derivatives and we did it in class. Expanding the expression $\|X\beta - \boldsymbol{y}\|_2^2$ we have $\beta^\top X^\top X\beta + 2\beta^\top X^\top \boldsymbol{y} + \|\boldsymbol{y}\|_2^2$. Taking the derivative and setting equal to zero we have: $X^\top X\beta + \beta^\top X^\top \boldsymbol{y} = 0$, and solving gives us the normal equation above.

Now, we have from above that $\boldsymbol{y} = X\beta + \boldsymbol{w}$. Note that while we do not know β, we do know $\boldsymbol{y}$, so we can plug this in to our expression for $\hat{\beta}$ above. This gives:

$$\begin{aligned} \hat{\beta} &= (X^\top X)^{-1} X^\top \boldsymbol{y} \\ &= (X^\top X)^{-1} X^\top (X\beta + \boldsymbol{w}) \\ &= (X^\top X)^{-1} X^\top X\beta + (X^\top X)^{-1} X^\top \boldsymbol{w} \\ &= \beta + (X^\top X)^{-1} X^\top \boldsymbol{w}. \end{aligned}$$

As we noted in class, this means that $\hat{\beta}$ is an *unbiased estimator* of β, since it is equal to the true β, plus a multiple of zero mean noise. That is, if we consider $\hat{\beta}$ as a random variable, since it depends on the outcome of the noise which is random, we can compute its expectation:

$$\mathbb{E}[\hat{\beta}] = \mathbb{E}[\beta + (X^\top X)^{-1} X^\top \boldsymbol{w}] = \beta + (X^\top X)^{-1} X^\top \mathbb{E}[\boldsymbol{w}] = \beta.$$

And therefore $\hat{\beta}$ is an *unbiased estimate of* β.

While it is good that $\hat{\beta}$ is unbiased, we would also like to know how much variation we should expect. That is, if we perform the experiment again but with a different subset of the population (i.e., a different data set that we believe is statistically similar), how much should we expect $\hat{\beta}$ to change? This is a measure of how much we trust our calculation of $\hat{\beta}$, and hence it tries to answer our main question above.

Let's compute the standard deviation of each coefficient of $\hat{\beta}$. The standard deviation is the square root of the variance. Hence we want to compute the variance of each coefficient $\hat{\beta}_i$. That is, we want to compute

$$\mathrm{Var}(\hat{\beta}_1) = \mathbb{E}[(\hat{\beta}_i^2 - \mathbb{E}[\hat{\beta}i])^2] = \mathbb{E}[\hat{\beta}_i^2] - (\mathbb{E}[\hat{\beta}_i])^2.$$

Showing that the last equality holds is an exercise in the probability section below. Also in the exercise below, you show that if we have a vector of random variables, as we do here ($\hat{\beta}_1$ is random, $\hat{\beta}_2$ is random, and so on – however many dimensions $\hat{\beta}$ has), then the variance of each component of $\hat{\beta}$ is given by the diagonal of the covariance matrix.

That means that if we can compute the $p \times p$ matrix $\mathrm{Cov}(\hat{\beta})$, then $\mathrm{Var}(\hat{\beta}_i) = [\mathrm{Cov}(\hat{\beta})]_{ii}$. That is, the variance of the i^{th} component of $\hat{\beta}$ is the i^{th} element of the diagonal of the matrix $\mathrm{Cov}(\hat{\beta})$.

To compute $\mathrm{Cov}(\hat{\beta})$ we have:

$$\begin{aligned}
\mathrm{Cov}(\hat{\beta}) &= \mathbb{E}[\hat{\beta}\hat{\beta}^\top] - \mathbb{E}[\hat{\beta}]\mathbb{E}[\hat{\beta}]^\top \\
&= \mathbb{E}[\hat{\beta}\hat{\beta}^\top] - \beta\beta^\top \\
&= \mathbb{E}[((X^\top X)^{-1}X^\top \boldsymbol{w})((X^\top X)^{-1}X^\top \boldsymbol{w})^\top] \\
&= \mathbb{E}[(X^\top X)^{-1}X^\top \boldsymbol{w}\boldsymbol{w}^\top X(X^\top X)^{-1}] \\
&\overset{(a)}{=} (X^\top X)^{-1}X^\top \mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\top]X(X^\top X)^{-1} \\
&\overset{(b)}{=} \sigma^2(X^\top X)^{-1}X^\top X(X^\top X)^{-1} \\
&= \sigma^2(X^\top X)^{-1},
\end{aligned}$$

where (a) follows because by the same principle that $\mathbb{E}[4Z] = 4\mathbb{E}[Z]$, the expectation also moves through the constant (non-random) matrices $(X^\top X)^{-1}X^\top$; and (b) follows because $\mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\top] = \sigma^2 I$, and we can pull that out to the left (σ^2 is a constant and commutes with matrix multiplication, and the identity is just the identity).

Computing σ^2. The above says that the standard errors of the coefficients of $\hat{\beta}$ that we compute, are given by the square root of the diagonals of $\sigma^2(X^\top X)^{-1}$. We have a problem, however: *we do not know σ*. We must therefore estimate it from the data.

(c) If we knew β and our proposed model

$$\texttt{FE} = \beta_0 + \texttt{EngDispl}\beta_1 + \texttt{NumCyl}\beta_2 + w,$$

or in terms of $\boldsymbol{x}_i$ and y_i,

$$y_i = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + w_i,$$

were perfect, then we could compute

$$w_i = y_i - (\beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2),$$

and thereby estimate σ^2, the variance of $\boldsymbol{w}$, via

$$\hat{\sigma} = \frac{1}{n-2}\sum_{i=1}^{n} w_i^2 = \frac{1}{n-2}\sum_{i=1}^{n}(y_i - (\beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2))^2$$

Remark: Note that we divide by $(n-2)$ instead of n. It turns out that this is the best way to form an unbiased estimate of the variance, from data.

We do not know β, however. Hence we must estimate it using our estimate for β.

Using the `cars2010` data set, and again regressing `FE` against `EngDispl` and `NumCyl`, compute an estimate for σ. Do the same for the data sets `cars2011` and `cars2012`.

(d) Compute the covariance of $\hat{\beta}$ for the `cars2010` data set when you regress `FE` against `EngDispl` and `NumCyl`, and report the standard deviation of $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\beta}_2$.

(e) How do you explain the fact that the coefficients from year to year seem to differ by several standard deviations?

3. **Ridge Regression and ℓ^2 Regularization.**

In this exercise, we will explore a modification of the least squares objective. That is, rather than computing $\hat{\beta}$ by solving:

$$\min: \ \frac{1}{n}\|X\beta - \boldsymbol{y}\|_2^2 = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i\beta - y_i)^2,$$

we will solve the so-called ℓ^2-regularized regression, also known as *Ridge Regression*:

$$\min: \ \frac{1}{n}\|X\beta - \boldsymbol{y}\|_2^2 + \lambda\|\beta\|_2^2 = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i\beta - y_i)^2 + \lambda\sum_{i=1}^{n}\beta_i^2.$$

(a) Since we are penalizing each coefficient of β by the same multiple of λ, it makes sense to *standardize the data*. To do this: For each column X_j of the X matrix, form the standardized version by

$$\tilde{X}_{ij} = \frac{X_{ij} - \bar{X}_j}{\hat{\sigma}_j},$$

where $\bar{X}_j$ is the average of the elements in the j^{th} column, and $\hat{\sigma}_j$ is the standard deviation of the elements of the j^{th} column. Similarly, center the output:

$$\tilde{y}_i = y_i - \bar{y},$$

where again, $\bar{y}$ is the mean of the output variables y_i.

(b) Use `sklearn.linear_model.Ridge` to solve the ridge regression function with your centered data for three different values of λ, from small to large: $\lambda = 0.01, 5, 10000$. What do you observe?[1] This should be illustrating a similar concept as what we saw in class, which is also illustrated here: `http://scikit-learn.org/stable/auto_examples/linear_model/plot_ridge_path.html`

(c) (Choosing λ). You will use `cars2011` as a *cross validation set*, to help you choose a good value of λ. Choose a reasonable range of λ, and solve for the ridge regression solution using that value of λ – note that for this part you are using `cars2010`. Then, compute the prediction error using the data set `cars2011`. So, the procedure is: for a fixed λ, solve ridge regression on `cars2010` and get the solution, W$\hat{\beta}_\lambda$; then evaluate the prediction error of this $\hat{\beta}_\lambda$ on the data set `cars2011`. Now move to your next value of λ in the range that you are sweeping out, and repeat.

Note: you are standardizing the data in `cars2010`. Therefore, you must also standardize the data in your validation set, `cars2011`. You must standardize *using the same mean and standard deviation values that you computed from the `cars2010` data set, not using the mean and standard deviation of the `cars2011` data set.*

[1] Note that if you want, you can have this normalize the data for you, but we do not need to select this option because we normalized manually above. We will see in the next exercises why we did this.

(d) Now use `cars2012` as the test set, to compute the prediction error for the optimal λ that you computed above.

(e) In this exercise, we learned about the *training set*, the *cross validation* (or just *validation*) set, and finally, the *testing set*. Why did we need three different sets?

4. **LASSO and ℓ^1-Regularized Regression**

In this exercise we will play with a different regularizer, and instead of the sum of squares, we will penalize with the sum of absolute values. Thus, we solve:

$$\min : \frac{1}{n}\|X\beta - \boldsymbol{y}\|_2^2 + \lambda\|\beta\|_1 = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i\beta - y_i)^2 + \lambda\sum_{i=1}^{n}|\beta_i|.$$

(a) Download the synthetic data sets `lasso.zip` from Canvas. These have a training, testing and validation set, as in the previous exercise: $(X_{\text{train}}, \boldsymbol{y}_{\text{train}})$, $(X_{\text{cv}}, \boldsymbol{y}_{\text{cv}})$, and $(X_{\text{test}}, \boldsymbol{y}_{\text{test}})$.

(b) Find the ridge regression solution for $\hat{\beta}$ with a very small λ, e.g., $\lambda = 0.001$. This is helpful, compared to OLS, because the problem is under-determined. Compute the training error and compare it to the testing error.

(c) Pick a range of λ and find the best λ in this range using the validation data set (hint: the optimal value will be quite small, so start from $\lambda = 0$). Then compute its testing error on the testing set. Compare to what you found above. Also, report the $\hat{\beta}$ that you found.

(d) Repeat the above, except use the *training set* to do cross validation. That is, sweep the same range of regularization parameter, λ, and then choose the one that gives the best error on the training set. Compare this λ to the one you found above. Compare the predictive quality of the solution to the one you found above. Why did this approach not work?

(e) (Bonus) Scikit-learn has several packages for cross validation, as well as some examples. Have a look at `http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.cross_val_score.html` and the examples, e.g., `http://scikit-learn.org/stable/auto_examples/exercises/plot_cv_diabetes.html`. Note that this has a different approach for cross validation. You can use it to quickly recompute the best values for λ (which they refer to as α). What do you get?

Probability

1. Let Z be a random variable that takes values z_i with probability p_i, $i = 1, \dots, n$.

(a) Write down the expression for $\mathbb{E}[Z]$ and for $\mathbb{E}[Z^2]$. Show that if $p_i = p_j$ for all i and j, then $\mathbb{E}[Z]$ is just the average of the values $\{z_1, \dots, z_n\}$.

(b) The variance of Z is a measure of how spread out Z is, i.e., how much it deviates from its expected value. Precisely, variance is defined as $\text{Var}(Z) = \mathbb{E}[(Z - \mathbb{E}[Z])^2]$. Show that

$$\text{Var}(Z) = \mathbb{E}[(Z - \mathbb{E}[Z])^2] = \mathbb{E}[Z^2] - (\mathbb{E}[Z])^2.$$

2. Suppose X and Y are random variables. The *Covariance* of X and Y s defined as

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])].$$

(a) Show that

$$\mathrm{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y].$$

(b) Find examples of random variables X and Y where (a) $\mathrm{Cov}(X, Y) > 0$, (b) $\mathrm{Cov}(X, Y) < 0$, and (c) $\mathrm{Cov}(X, Y) = 0$. When $\mathrm{Cov}(X, Y) = 0$ we say that X and Y are uncorrelated.

3. If X and Y are random variables, they are called *independent* if

$$\mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x) \cdot \mathbb{P}(Y = y),$$

for all values of x and all values of y. If X, Y and Z are random variables, they are called *pairwise independent* if all pairs are independent. If X, Y and Z are random variables, they are called *mutually independent* if

$$\mathbb{P}(X = x, Y = y, Z = z) = \mathbb{P}(X = x) \cdot \mathbb{P}(Y = y) \cdot \mathbb{P}(Z = z),$$

for all values of x, y and z. And so on, for collections of more than three random variables.

Show that if X and Y are independent, then

$$\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y].$$

Note that this implies that if X and Y are independent, then they are uncorrelated.

4. For X, Y, and Z random variables, the *Covariance Matrix* of the random vector (X, Y, Z) is defined as:

$$\mathbb{E}\left[\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \begin{pmatrix} X & Y & Z \end{pmatrix}\right] = \mathbb{E}\begin{bmatrix} X^2 & XY & XZ \\ XY & Y^2 & YZ \\ XZ & YZ & Z^2 \end{bmatrix}.$$

Suppose X, Y and Z represent the result of rolling three independent dice. What is the covariance matrix of (X, Y, Z)?

5. Show by example that while independent implies uncorrelated, the converse does not hold. That is, find an example of random variables X and Y that are uncorrelated but not independent.[2]

6. Find an example of three random variables X, Y and Z that are pairwise independent, but not mutually independent.

7. Consider flipping a fair coin. Let Z denote the random variable that is the number of Heads that come up in a row. Thus, if the first flip comes up tails, $Z = 0$. If the flip sequence is

$$HHTHHHHT....$$

then $Z = 2$, and so on.

- Write the probability mass function of Z.
- Compute the mean and variance of Z.

[2] For jointly Gaussian (or Normal) random variables, in fact this does not hold: independent implies uncorrelated, and uncorrelated implies independent.