**STAT 383C: Statistical Modeling I** **Fall 2015**

# Lecture 9 — September 24

*Lecturer: Purnamrita Sarkar* *Scribe: Elaina Babayan*

**Disclaimer:** These scribe notes have been slightly proofread and may have typos etc.

**Note:** The latex template was borrowed from EECS, U.C. Berkeley.

## 9.1 Cross-Validation (cont.)

Cross-validation can be used to estimate the prediction error of a model (on unseen data) with k features. Then by varying k and comparing the error the number of features can be selected to minimize the predictive error. Cross-validation can also be used to compare different model classes (e.g. linear model, quadratic model, etc.).

### 9.1.1 How to Perform Cross-Validation

1. Select the number of folds to be used, b, the number of model features, k (do not select which k features, just the number k), and model type to be investigated
2. Perform random shuffling on data to divide it into b folds
3. For each fold, i, where i= 1,...,b, leave out the $i^{th}$ fold and use the remaining b-1 folds as training data
4. Using the best subset method, fit all possible combinations of models with k features to the training data
5. Calculate the RSS training error of each of the models and select the model with the lowest training error
6. Calculate and store the test error of the best model (using test data from ith fold), $err_i$. Average over the test points.
7. Repeat for all b folds and then calculate the mean and standard deviation of the test error

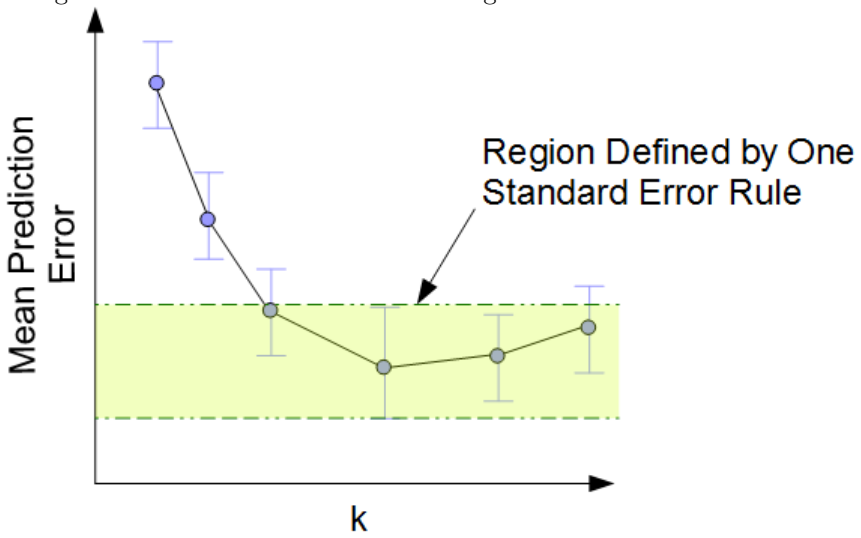
$$m = \frac{\sum_{i=1}^{b} err_i}{b} \tag{9.1}$$

$$se = \sqrt{\frac{var(err)}{b}} \tag{9.2}$$

8. Repeat for different values of k

**One Standard Error Rule:** The One Standard Error Rule can be used to compare models with different numbers of parameters in order to select the most parsimonious model with low error. To use, find model with minimum error, then select the simplest model whose mean falls within 1 standard deviation of the minimum (Fig. 9.1).

**Figure 9.1.** Selecting Number of Model Parameters Using Cross-Validation and One Standard Error Rule




### 9.1.2 Comparison of Cross-Validation with AIC and BIC

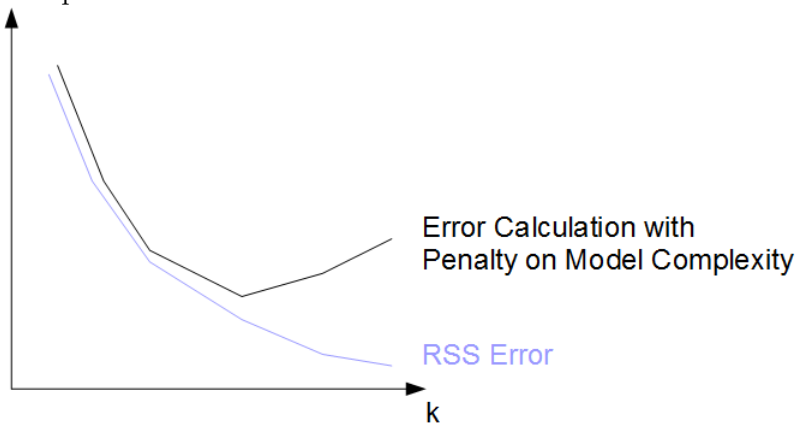
AIC and BIC are alternative methods of estimating the model predictive error. Both penalize model complexity, so as k becomes large both AIC and BIC will estimate larger errors than an RSS error which considers training error only (Fig. 9.2). This helps prevent overfitting.

As compared to cross-validation, AIC and BIC are computationally cheaper but are limited–may not be valid for some nonlinear problems Cross-validation is more computationally expensive, but it is more broadly applicable.

## 9.2 Regularization

Regularization methods such as Ridge Regression and LASSO introduce a penalty term on model complexity to prevent overfitting.

**Figure 9.2.** A Comparison of RSS Model Error to Methods That Penalize Model Complexity




### 9.2.1 Ridge Regression

For least squares regression, the model coefficients are selected by

$$\hat{\beta}_{LS} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \tag{9.3}$$

For ridge regression, an additional term is added which penalizes all $\beta_j$ for $j > 0$

$$\hat{\beta}_{ridge} = \arg\min_{\beta} \sum_{i=1}^{n} ((y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{9.4}$$

$\lambda$ is a positive constant that we pick using Cross Validation. It is clear from this equation that if the variables are on different scales the Ridge Regression model will penalize them differently. Thus, the standard errors must be normalized to 1. Moreover, note that we do not penalize the intercept term, since that would lead to a different fit (modulo scaling) if one added a constant to all features.

**Reparameterization** Use redefinition of model parameters to center $X$ and $Y$ about their mean values. Replace $x_{ij}$ with $x_{ij} - \bar{x_j}$ and estimate $\beta_0$ with $\bar{y}$

$$\beta_0' = \beta_0 + \sum_{j=1}^{k} x_{ij}\beta_j \tag{9.5}$$

$$\beta_j' = \beta_j, \forall j > 0 \tag{9.6}$$

Now the new objective function is given by:

$$\sum_{i=1}^{n} (y_i - \beta_0' - \sum_{j=1}^{p} (x_{ij} - \bar{x}_j)\beta_j')^2 + \lambda \sum_{j=1}^{p} (\beta_j')^2 \tag{9.7}$$

Lets see what the MLE of $\beta_0'$ is. Remember, in order to get that we derive the above w.r.t $\beta_0'$ and set it to zero.

$$\sum_{i=1}^{n}(y_i - \hat{\beta}_0' - \sum_{j=1}^{p}(x_{ij} - \bar{x}_j)\hat{\beta}_j') = 0$$

Now, we have:

$$n(\bar{y} - \hat{\beta}_0') - \sum_{i=1}^{n}\sum_{j=1}^{p}(x_{ij} - \bar{x}_j)\hat{\beta}_j' = n(\bar{y} - \hat{\beta}_0') - \sum_{j=1}^{p}(\sum_{i} x_{ij} - n\bar{x}_j)\hat{\beta}_j' = n(\bar{y} - \hat{\beta}_0') = 0$$
$$\Rightarrow \hat{\beta}_0' = \bar{y}$$

Now we have:

$$\hat{\boldsymbol{\beta}}' = \arg\min_{\boldsymbol{\beta}'} \sum_{i=1}^{n}((y_i - \bar{y} - \sum_{j=1}^{p}(x_{ij} - \bar{x}_j)\beta_j')^2 + \lambda\sum_{j=1}^{p}(\beta_j')^2$$

Since Equation 9.5 gives us a one one correspondence between the $\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$s, its not hard to argue that the optimization functions are equivalent, i.e. the argmin of one, after suitable transformation gives the argmin of the other. So from now on we will standardize our X's and center our Y's and learn a ridge regression through the origin. Unlike in OLS $\boldsymbol{\beta}$ will have $p$ dimensions instead of $p + 1$.

Thus the reparameterized equation for $\hat{\beta}_{ridge}$ is

$$\hat{\boldsymbol{\beta}}_{ridge} = \min_{\beta}(\boldsymbol{y} - \boldsymbol{X}^T\boldsymbol{\beta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta} \tag{9.8}$$

with solution–

$$\hat{\boldsymbol{\beta}}_{ridge} = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{9.9}$$

**Numerical stability:** Note that the ridge regression is identical to OLS except the fact that we are adding $\lambda$ to the diagonals of $\boldsymbol{X}^T\boldsymbol{X}$. Essentially this stabilizes the problem, if $\boldsymbol{X}^T\boldsymbol{X}$ has small eigenvalues. Why? Because for any square symmetric matrix, adding the same $\lambda$ to all diagonal terms simply adds $\lambda$ to all the eigenvalues and as a result makes the matrix non-singular. In fact, ridge regression was introduced for the first time in Statistics (Hoerl and Kennard, 1970) with this motivarion.

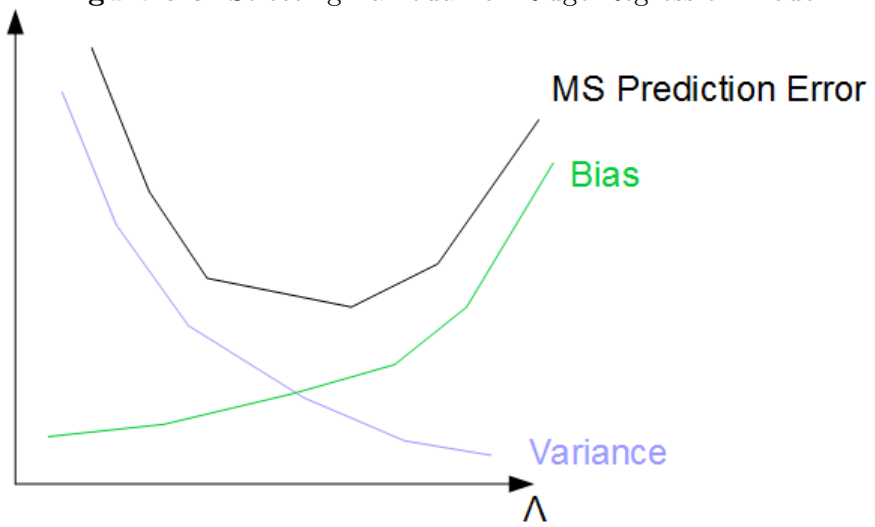
it basically adds $\lambda$ to that.

**How to Choose $\lambda$:** Selection of $\lambda$ is a tradeoff between bias, variance and mean square error (Fig. 9.3).

1. Perform a grid search over $\lambda$ or $log(\lambda)$

2. For each $\lambda$ perform a cross-validation and calculate the mean and standard error on the estimated model prediction error
3. Identify the $\lambda$ with the lowest mean predictive error
4. Apply One Standard Error Rule to select most parsimonious model whose mean lies within one standard error (in this case more parsimonious means more regularization, so a larger $\lambda$)

**Figure 9.3.** Selecting Lambda For Ridge Regression Model




**Equivalent Bayesian Interpretation** Assume that $y$ is normally distributed, and apply a Gaussian prior to $\beta$

$$\boldsymbol{y} \sim N(\boldsymbol{X}\beta, \sigma^2 I) \tag{9.10}$$
$$\boldsymbol{\beta} \sim N(0, \tau^2 I) \tag{9.11}$$

Then the posterior distribution of $\beta$ given $y$ can be calculated as

$$f(\beta|y) \propto \exp\left(\frac{-(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})}{2\sigma^2}\right)\exp\left(\frac{-\boldsymbol{\beta}^\top\boldsymbol{\beta}}{2\tau^2}\right) \tag{9.12}$$

Taking a logarithm we see that the MAP estimate (in this case also posterior mean) is none other that–

$$\boldsymbol{\beta}_{MAP} = \arg\min_{\boldsymbol{\beta}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T + \frac{\sigma^2}{\tau^2}\boldsymbol{\beta}^T\boldsymbol{\beta}$$

From this form it can be observed that the relationship between $\sigma$ and $\tau$ defines $\lambda$. For example, if $\tau$ is much smaller than $\sigma$, then we penalize more for larger magnitude of $\boldsymbol{\beta}$.

Ridge regression generally has smaller MSE than linear regression, and if some of the true parameter values were zero, Ridge Regression will make the corresponding coefficients small. However, Ridge Regression will not drive coefficients to zero (unless $\lambda = \infty$, in which case it drives all coefficients to zero), so it cannot be used for variable selection.

### 9.2.2 LASSO

$$\hat{\boldsymbol{\beta}}_{LASSO} = \min_{\boldsymbol{\beta}} (\boldsymbol{y} - \boldsymbol{X}\beta)^\top (\boldsymbol{y} - \boldsymbol{X}\beta) + \lambda \sum_{j=1}^{p} |\beta_j| \tag{9.13}$$

The MSE of LASSO is comparable to Ridge Regression, and LASSO will drive some coefficients to zero with a large $\lambda$. This is a convex optimization problem, for which there are efficient optimization algorithms.

Like in the ridge regression setting, we standardize the $X$'s, center the $Y$'s and train a model through the origin.

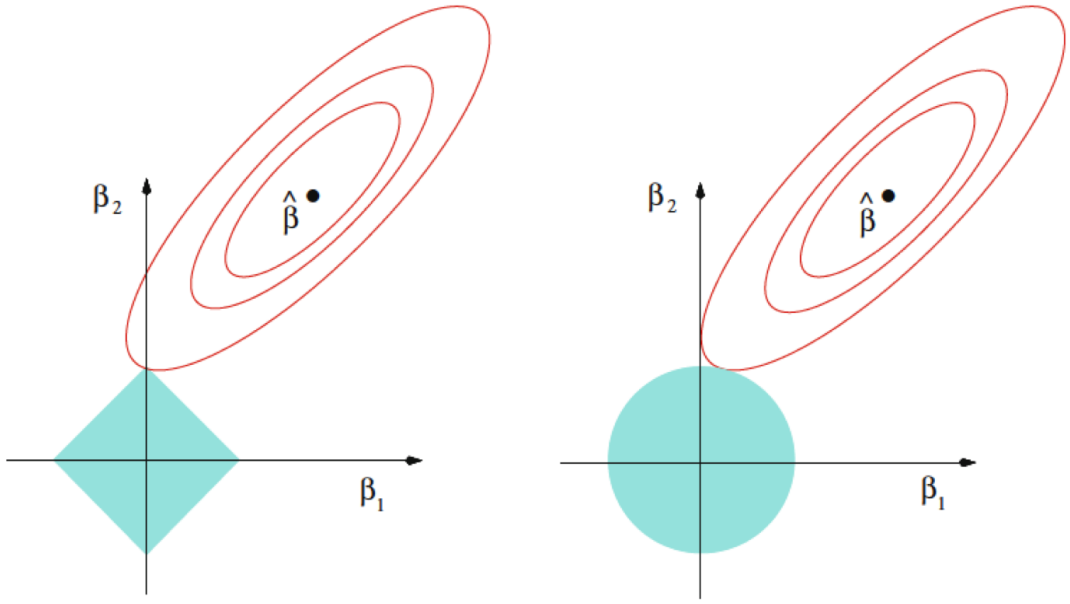



**FIGURE 3.11.** *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

**Alternative Objective Function** It can be shown that both the Ridge and Lasso regression problems can also be reformulated as follows:

$$\hat{\boldsymbol{\beta}}_{ridge} = \min_{\boldsymbol{\beta}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \qquad \text{Subject to } \boldsymbol{\beta}^\top \boldsymbol{\beta} \leq \tau^2 \tag{9.14}$$

$$\hat{\boldsymbol{\beta}}_{lasso} = \min_{\boldsymbol{\beta}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \qquad \text{Subject to } \|\boldsymbol{\beta}\|_1 \leq \tau \tag{9.15}$$

In fact, the alternative objective functions are equivalent to the corresponding ones with added penalty terms. For every $\lambda$ there is a $\tau > 0$ for which the same $\hat{\beta}$ minimizes the two objective functions. Pictorially (Taken from H-T-F[1]) this essentially is telling us that an unconstrained optimization tries to minimize the RSS in Figure 9.2.2. However the constraints essentially forces one to pick the $\boldsymbol{\beta}$ for which the contours of the unconstrained objective function first hits the constraint area.

The lasso constraint region has corners unlike the ridge regression constraint region. As we go to higher dimensions there will be more faces, corners etc, which zero out some coefficients. Thus Lasso does variable selection, unlike ridge regression.

# References

[1] T. Hastie, R. Tibshirani, J. Friedman. The elements of statistical learning. *Springer*, 2013.