4-27-2011

A Review of Cross Validation and Adaptive Model Selection

Ali R. Syed
Georgia State University

A REVIEW OF CROSS VALIDATION AND ADAPTIVE MODEL SELECTION

by

ALI RAZA SYED

Under the Direction of Yixin Fang

ABSTRACT

We perform a review of model selection procedures, in particular various cross validation procedures and adaptive model selection. We cover important results for these procedures and explore the connections between different procedures and information criteria.

INDEX WORDS: Model selection, Adaptive model selection, Cross validation, Information Criteria

A REVIEW OF CROSS VALIDATION AND ADAPTIVE MODEL SELECTION

by

ALI RAZA SYED

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

in the College of Arts and Sciences

Georgia State University

2011

A REVIEW OF CROSS VALIDATION AND ADAPTIVE MODEL SELECTION

by

ALI RAZA SYED

Committee Chair: Yixin Fang

Committee: Yichuan Zhao

Jiawei Liu

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

May 2011

DEDICATION

This is dedicated to my family whose love and support makes my learning endeavors possible.

ACKNOWLEDGEMENTS

I am grateful to the Department of Mathematics and Statistics at Georgia State University for the instruction I received during my Masters in Statistics. In particular, I thank Dr. Fang for some great conversations concerning research topics and for inspiring me to delve further into some exciting areas I may otherwise have never learned about. I also wish to thank the excellent teachers I have had in many classes which includes, but is not limited to, Dr. Zhao, Dr. Han, Dr. Qin and Dr. Zhang. I entered the program knowing very little about discipline of Statistics, and while I have much further to go before proclaiming mastery, I leave the program possessing the tools and confidence to continue onwards.

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

Model selection in statistics is the procedure of selecting the "best" model among a set of competing models. A model is judged to be "best" according to some criteria. A common and prevailing approach is to balance goodness of fit with parsimony. Goodness of fit determines how well the model describes the data. However, increasingly complex models, with increasing number of parameters, are bound to provide better fits at the expense of fitting to the noise as well as the data. This leads to the phenomenon of over-fitting: the model describes the trained data well, but fails to take into generalize to new data. The principle of parsimony, related to Occam's razor, advocates choosing simpler models, with fewer parameters. By balancing model complexity with goodness of fit, we can develop models with lower generalization error.

Cross validation is a method of measuring generalization error through the use of holdout data. There are many cross validation techniques and one of the most common is leave-one-out cross validation (LOO). Adaptive model selection uses a generalization of penalized criteria for model selection where the penalty is based on the data (X. Shen & Ye, 2002). The adaptive selection procedure has the advantage of performing well across a number of different modeling procedures.

This thesis reviews selected important theory concerning these model selection procedures. In this chapter, we continue with a brief sketch of important concepts. In the remaining chapters, we review cross validation procedures and adaptive model selection in greater detail and the connections between various procedures and information criteria.

1.1 The "Best" Model

Faced with competing models, we must first consider criteria for choosing between competing models. A common refrain in statistical modeling is "all models are wrong; some are useful" and there is

much philosophy of science surrounding this topic. In a statistical modeling framework, we are usually concerned with one of two goals for model selection: model estimation or model identification. Model identification has the goal of minimizing a loss function and the desire here is for a statistically efficient model selection procedure. This is commonly the goal in predictive modeling. Model identification has the goal of finding the smallest optimal model describing the data, or the "true" model in this sense, and the desire here is for a statistically consistent model selection procedure. This is commonly the goal in descriptive modeling where we seek to explain a natural or social phenomenon.

1.2 Information Criteria

Given that the true model is unknown to us, we try to quantify the loss of information from the approximate model in consideration M over the available data D. We seek to minimize an information criterion GIC with the general formulation:

$$GIC = -2L(M, D) + \lambda \mid M \mid .$$

$L(M)$ is the log-likelihood of the data D given the model M and λ is factor controlling the penalty exacted for the model's complexity. The first term measures the goodness of fit (GOF) while the second term controls for model complexity. In fitting the model, the likelihood of D given M is found by using the maximum likelihood estimates of the model parameters in M and represents an averaged maximized log-likelihood rather than the expected maximized log-likelihood. Thus, the second term may also be interpreted as a bias estimation to correct for this fact (Sima, 2006). This combination of trading between goodness of fit and complexity may also be seen as the trade-off between bias and variance.

In a classical least squares model, ignoring additive constants and multiplying through by σ^2 results in an equivalent formulation for minimization:

$$GIC = RSS + \lambda \mid M \mid \sigma^2 .$$

RSS is the residual sum of squares. There are a number information criteria, derived from different theoretical considerations, each using a different penalty factor λ. Two common criteria are the AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion). In AIC, $\lambda = 2$ and in BIC, $\lambda = \log n$ where n is the number of observations in our data. AIC arises from information theoretic foundations by considering the expected Kullback-Liebler divergence between the true and approximated models. BIC arises from Bayesian decision-theoretic foundations using the Bayes factor. The BIC exacts a heavier penalty than AIC for more complex models. Very generally, the AIC is preferred for model estimation in predictive modeling due to its asymptotic efficiency property, while the BIC is preferred for model identification in descriptive modeling due to its asymptotic consistency property (Clarke, Fokoue, & H. H. Zhang, 2009).

1.3 Adaptive Model Selection

We consider adaptive model selection (X. Shen & Ye, 2002) because of its connection to information criteria and important properties in controlling for model complexity. In adaptive model selection, the λ factor is a data-adaptive penalty derived using the generalized degrees of freedom for a given modeling procedure. Ye defines a model selection procedure as having two parts: *selection* followed by *fitting*. Consider the variable selection problem in linear regression, where a set of variables must be chosen from a pool of candidate variables. The selection process results in the subset of variables in consideration, while the fitting process determines the goodness of fit for the given subset. Applying the traditional information criteria, such as AIC or BIC, does not correct for the bias induced by the variable selection process. By considering the modeling procedure in totality, the adaptive model selection adjusts for the selection bias. Specifically, it is found that the optimal $\hat{\lambda}$ is obtained by minimizing:

$$RSS + 2G(\lambda).$$

$G(\lambda)$ are the *generalized degrees of freedom* for the model selection procedure.

1.4 Cross Validation

In addition to Information Criteria, cross validation is another popular set of techniques used in model selection. The general procedure is to partition the data into subsets for training and testing. Training is the process of fitting a model while testing is the process of validating the fitted model through measuring the prediction error. The training and test sets are disjoint so the testing data for model evaluation are not used in model fitting. Cross validation is used across a range of areas such as parameter selection, density estimation, classification and stopping criteria in neural networks. Cross validation is not an information criterion in the sense that it does not penalize a goodness of fit measure. However, there exist asymptotic equivalences between cross-validation techniques and some information criteria.

1.5 Overview of the Remainder

There are a variety of model selection procedures in the statistical literature and the foregoing highlights some of the commonly used procedures. In the subsequent chapters, we review cross validation and adaptive model selection in greater detail. Information criteria are not considered in great depth since these are covered in various statistics classes; the introduction above serves as a summary for exploring connections between information criteria and the procedures we will review. In general, we will examine the procedures and their properties in the context of linear models of the form:

$$Y = X'\beta + \epsilon .$$

It is understood that Y is a $n \times 1$ vector for n observations, X is a $n \times p$ vector of observed values for p variables, ϵ is a $n \times 1$ vector with $\epsilon \sim \mathcal{N}\left(0, \sigma^2 I\right)$ and β is an unknown $p \times 1$ vector of coefficients to be estimated. In some cases, we denote f as the function to be estimated through the regression coefficients. In this context, the modeling selection problem is to select a k -subset of va-

riables and estimate an optimal model $\hat{M}_k$ with cardinality $|M| = k$. However, many of the selection procedures reviewed are more generally applicable.

2 CROSS VALIDATION

Suppose a dataset $D = \{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$ and we wish to assess a regression model M to arrive at a corresponding set of predicted values $\hat{y}_i, i = 1, \ldots, n$. We can partition the dataset D into two sets: $D = D_1 \cup D_2$, with k data in D_1 and $n-k$ data in D_2. We fit the model M using the dataset D_2; this is also known as training the model and D_2 as the *training set*. We then use the trained model M to obtain predictions for observations $\hat{Y}_{D_1}$ given X_{D_1}; this is also known as testing the model and D_1 as the *test set*. There are $\binom{n}{k}$ possible partitions of the data and this process can be repeated multiple times. The CV estimate of error is the average prediction error over test sets used and this is estimation of the average generalization error from applying our fitted function to an independent test sample (Clarke, Fokoue, & H. H. Zhang, 2009).

2.1 Leave-One-Out Cross Validation

When $k = 1$ is used in the above formulation, the process is called leave-one-out cross validation (LOOCV), which we review in some depth because of a number of important connections. In this case, our test set always has cardinality 1, and each of the $i = 1, \ldots, n$ possible partitions are used to train and test the model. For each i, let $\hat{y}^{-i}$ denote the predicted value of left-out observation. The leave-one-out cross validation estimate of error is:

$$LOOCV = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^{-i})^2 .$$

The general idea of leave-one-out cross validation appears to have been known for a while, though the earliest "official" statement by Mosteller and Tukey who mentioned that this process ex-

tracts all information from given data without further justification. In linear regression, the LOOCV estimate is known as the PRESS (prediction sum of squares) statistic.

2.1.1 Leaving-one-out lemma

For large samples, the LOOCV estimate seems to have a heavy computational cost requiring n model fits. An interesting result obviates this requirement for certain modeling procedures which are linear in the observations, with $\hat{Y} = \mathbf{H}Y$, $\mathbf{H}$ being the influence or hat matrix, and requires only one fit over the entire dataset D. In this case, let h_{ii} denote the diagonal entries of $\mathbf{H}$, then the statistic is calculated as:

$$LOOCV = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_{ii}}\right)^2.$$

This result holds for linear regression models and cubic smoothing splines, among other procedures, which satisfy the *leaving-one-out lemma* (Wahba, 1990) stated below.

Denote $\tilde{y}^i = (y_1, \ldots, y_{i-1}, \hat{y}_i^{-i}, y_{i+1}, \ldots, y_n)$ as the consequence of replacing i th component of Y with $\hat{y}_i^{-i}$, and denote $\tilde{f}^{-i}$ as the estimate of our fitting function f given data $\tilde{y}^i$. The *leaving-one-lemma* states (Wahba, 1990):

$$\tilde{f}^{-i}(x_i) = \hat{f}^{-i}(x_i), i = 1, \ldots, n.$$

The geometric interpretation here is that adding a new point exactly on the surface of $\hat{f}^{-i}$ leaves the fitted regression unaltered in the given system (Clarke, Fokoue, & H. H. Zhang, 2009). Since our system is linear in the observations with $\mathbf{H} = \mathbf{H}(X)$, we have:

$$\hat{f}(x_i) - \tilde{f}^{-i}(x_i) = \sum_{j=1}^{n} h_{ij} y_j + \left(\sum_{j \neq i}^{n} h_{ij} y_j + h_{ii}\hat{y}_i^{-i}\right) = h_{ii}(y_i - \hat{y}_i^{-i}).$$

Applying the leave-one-out lemma, we obtain:

$$\hat{f}(x_i) - \widetilde{f}^{-i}(x_i) = \hat{f}(x_i) - \widehat{f}^{-i}(x_i) = h_{ii}(y_i - \hat{y}_i^{-i})$$
$$\Rightarrow \frac{y_i - \hat{y}_i}{y_i - \hat{y}_i^{-i}} = 1 - h_{ii}$$

.

Using this expression in the original LOOCV equation leads to the revised equation. Since the revised form requires only one model fit over the entire data, the computational savings are considerable. We will exploit this connection further when we consider an analogous criterion for adaptive model selection.

2.2 Generalized Cross Validation

The generalized cross validation criterion (GCV) (Wahba, 1990) is an approximation to the LOOCV and follows from noting that $\text{tr}(\mathbf{H}) = \sum_{i=1}^{n} h_{ii}$ followed by the approximation: $h_{ii} \approx \frac{1}{n} tr(\mathbf{H})$. This is generally applicable when fitting linear methods with quadratic loss function and is a good approximation provided $h_{ii}, i = 1, \ldots, n$ are not very different (Wahba, 1990). The generalized cross validation statistic becomes:

$$GCV = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - \frac{1}{n} tr(\mathbf{H})} \right)^2 .$$

If the leaving-one-out lemma holds, then the generalized cross validation criterion may provide further computational savings since it requires finding the trace rather than the individual diagonal entries of the hat matrix.

2.2.1 Asymptotic Equivalence with AIC

The generalized cross validation statistic as an approximation to the leave-one-out statistic is also useful for examining analytic properties. Here we use the approximation to show the asymptotic equivalence of leave-one-out cross validation to the AIC (Clarke, Fokoue, & H. H. Zhang, 2009). Consider variable selection in a linear regression mode M where we evaluate a fixed subset of variables of size $|M|$. Since $tr(\mathbf{H}) = |M|$ in linear regression, and noting that $RSS(M) = \sum (y_i - \hat{y}_i)^2$, we have:

$$GCV(M) = \frac{1}{n} \frac{RSS(M)}{(1 - |M|/n)^2}.$$

For large n, when $|M|/n$ is small, we can apply a Taylor expansion, $(1 - |M|/n)^{-2} \approx 1 + 2|M|/n$ to arrive at:

$$GCV(M) \approx \frac{RSS(M)}{n} + 2\frac{RSS(M)}{n}\frac{|M|}{n}.$$

Further note that as $n \to \infty$, $RSS(M)/n \to \sigma^2$ and rewriting the above expression as:

$$GCV(M) \approx \frac{RSS(M)}{n} + 2\hat{\sigma}^2\frac{|M|}{n}.$$

We see that minimizing the GCV is equivalent to minimizing AIC. Note that the connection to Mallow's C_p in linear regression is more apparent here and, in fact, AIC is equivalent to Mallow's C_p in linear regression. Another point we can note is that although the estimate of the variance in linear regression, and consequently Mallow's C_p is based on $(n - p)$ degrees of freedom, our substitution above is valid asymptotically.

2.2.2 Further Note on the AIC/LOO Connection

The above is not a real proof of equivalence but simply suggestive of the connection between LOOCV and AIC and the real proof is due to (Stone, 1977). Stone's proof relies on a likelihood analysis

and is not limited to linear models. Consequently, the log-likelihood of the data can be approximated through the likelihood based LOOCV, or comparisons can be made based on LOOCV error statistic. This becomes useful when faced with models where the likelihood is analytically difficult to compute. Stone's proof also holds for the Takeuchi Information Criterion (TIC), a general form of the AIC, which involves computing the trace of a product involving the Fisher information matrix and the score function (Lee, 2007). This trace reduces to the number of parameters in the case of exponential family of distributions, but not generally. The asymptotic connection to the LOOCV may then be useful as an alternative to analytically difficult computations. However, the LOOCV carries its own computational cost and can be an expensive procedure when the GCV approximation does not hold.

We can also try to gain an intuitive understanding of the asymptotic equivalence by noting that the AIC minimizes the KL divergence between the approximate model and the true model. The KL divergence is not a distance measure between distributions, but really a measure of the information loss when the approximate model is used to model the ground reality. Leave-one-out cross validation uses a maximal amount of data for training to make a prediction for one observation. That is, $n-1$ observations as stand-ins for the approximate model relative to the single observation representing "reality". We can think of this as learning the maximal amount of information that can be gained from the data in estimating loss. Given independent and identically distributed observations, performing this over n possible validation sets leads to an asymptotically unbiased estimate.

The LOOCV method shares similar statistical properties with AIC: it provides asymptotically unbiased result for the true prediction error by trading off with variance.

2.3 Leave-K-Out Cross Validation

In the formulation at the start of this chapter, leave-k-out cross validation (LKOCV) is the general case where the size of the test set $|D_1| = k$. As mentioned earlier, this procedure carries considerable

computational expense due to the $\binom{n}{k}$ possible partitions that must be left-out and is rarely used in practice (Sylvain & Celisse, 2010).

2.4 K-Fold Cross Validation

An alternative procedure is K-fold cross validation and this procedure was motivated by computational expense of the leave-one-out procedure (Geisser, 1975). The K-fold procedure is attractive because it balances computational cost with an increase in the estimation bias. In this procedure, the dataset D is divided into K partitions of roughly equal size, $D = \bigcup_{k=1}^{K} D_k$, and each partition is termed a "fold" of the dataset (thus there are K folds). The procedure may be understood as a leave-one-fold-out procedure in analogy to the leave-one-out procedure. The model is trained on $K-1$ folds and the K th fold is used for testing (Clarke, Fokoue, & H. H. Zhang, 2009). This is repeated K times such that each fold is used for testing exactly once. Setting $K = n$ leads to leave-one-out cross validation.

Define an index function $\kappa : \{1,\ldots,n\} \to \{1,\ldots,K\}$ as a scheme to randomly assign the i th datum to a fold. Leaving out the k th fold for testing and fitting the model on the remaining $k-1$ folds, results in estimated model function $\hat{f}^{-\kappa(i)}$. The cross validation statistic for prediction error is then (Hastie, Tibshirani, & Friedman, 2001):

$$KCV = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}^{-\kappa(i)})^2 .$$

The k-fold cross validation procedure is often applied to choose a model specific parameter. Suppose that the models are indexed by parameter $\lambda \in \Lambda$ with corresponding estimated model function $\hat{f}_{\lambda}^{-\kappa(i)}$ to be evaluated on the k th fold. The optimal $\hat{\lambda}$ is chosen as (Hastie, Tibshirani, & Friedman, 2001):

$$\hat{\lambda} = \arg\min_{\lambda} KCV(\lambda) = \arg\min_{\lambda} \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_{\lambda}^{-\kappa(i)})^2 .$$

The final model is trained with the optimal parameter over the entire data, with the KCV statistic reported as the cross validation prediction error. In general, the recommended values for K are 5 or 10 .

2.5 Choosing a Cross Validation Method

As with any model selection procedure, the bias and variance tradeoff influences the choice of cross validation procedure. LOOCV is asymptotically unbiased as mentioned earlier. The choice of training set size influences our decision due to some important results in (Shao, 1993).

In general, choosing LOOCV or KCV with small size for the testing set results in an overestimation of the variance, but relatively small bias. When the goal is model estimation, and the sample has low variance, a smaller bias is preferred. Thus the recommendation is to opt for LOOCV or KCV with relatively small folds. The smallest bias is obtained when the training set size approaches the sample size, i.e. $(n-k) \to n$ as in LOOCV. The goal of model identification requires that a larger bias is induced since consistency results require the training set size to be much smaller than the sample size (Sylvain & Celisse, 2010).

It is found that small variance in the CV statistic generally leads to optimal model selection performance. In the cross validation cases covered here, the number of folds K is linked to the size of the training set. In this case, it has been found that the variance of the CV statistic must be quantified precisely and this varies with the specific procedure being used (Sylvain & Celisse, 2010).

3 ADAPTIVE MODEL SELECTION

Minimizing information criteria are a popular choice of model selection. As reviewed above, for model M these take the general form:

$$-2l(M) + \lambda |M|.$$

$l(M)$ is the maximum log-likelihood of the model M ; $|M|$ is the cardinality of the model; and λ is a fixed penalty factor depending on the criterion. In linear regression, this is equivalent to minimizing the form:

$$RSS(M) + \lambda |M|.$$

$RSS(M)$ is the residual sum of squares from fitting the model.

One problem with such criteria is none of them perform uniformly well across a variety of situations. AIC, with penalty $\lambda = 2$, performs well when the size of the true model is large but yields substantial bias when the model size is small. BIC, with a heavier penalty $\lambda = \log(n)$, performs well when the size of the true model is small, but yields substantial selection bias with large models (B. Zhang, 2010). The motivation for an adaptive model selection procedure (X. Shen & Ye, 2002) is to produce a data-adaptive penalty that reduces the selection bias across the range of situations. The basis for the data-adaptive penalty is the *generalized degrees of freedom* (Ye, 1998).

3.1 Degrees of Freedom

In linear regression, the degrees of freedom are best understood through geometric consideration and may be defined as the dimension of the estimation subspace (Walker, 1940). In the simplest case of a linear system with p independent variables per observation, we seek the least squares projection of the n-dimensional observation vector Y in a p-dimensional estimation subspace. There are p degrees of freedom available for estimation of the p-coordinate vector of Y and this resolves to

the number of independent parameters in our system. The system is linear in the observations through the projection matrix $\mathbf{H} = \mathbf{H}(X)$, also known as the hat matrix. Increasing the dimension of our estimation subspace reduces the least squares distance between Y and its projection $\hat{Y} = \mathbf{H}Y$ at the cost of introducing a more complex structure for our estimation subspace. Thus the degrees of freedom are a measure of model complexity and this is the term penalized by the various information criteria.

A similar notion extends to systems outside the least squares class as long as they are linear in the observations, i.e. with $\hat{Y} = \mathbf{H}Y$. In such cases, a variety of definitions exist for different systems as the *effective degrees of freedom*, usually the trace of some function d of the hat matrix: $tr(d(H))$. In linear regression, the degrees of freedom are equivalent to $tr(H) = \sum h_{ii} = \sum \frac{\partial \hat{y}_i}{\partial y_i}$, "the sum of the sensitivities of the fitted values with respect to the observed response values" (Ye, 1998).

3.2 Generalized Degrees of Freedom

The *generalized degrees of freedom* (GDF) is a generalization of the concepts of degrees of freedom mentioned above. In addition, the observation, given above, of degrees of freedom as the sum of sensitivities of fitted to response values motivates Ye's definition of GDF in linear models (Ye, 1998):

$$GDF(M) = \sum_{i=1}^{n} \frac{\partial \mathrm{E}\hat{\mu}_i(\mathbf{Y})}{\partial \mu_i} = \sum_{i=1}^{n} \lim_{\delta \to 0} \mathrm{E}\left[\frac{\hat{\mu}_i(\mathbf{Y} + \delta \mathbf{e}_i) - \hat{\mu}_i(\mathbf{Y})}{\delta} \right]$$
$$= \sum_{i=1}^{n} \frac{\mathrm{E}\hat{\mu}_i(\mathbf{Y})(y_i - \mu_i)}{\sigma^2} = \sum_{i=1}^{n} \frac{\mathrm{cov}(\hat{\mu}_i(\mathbf{Y}), y_i - \mu_i)}{\sigma^2} .$$

The GDF is defined as the "sum of *average* sensitivities of the fitted value $\hat{\mu}_i(\mathbf{Y})$ to a small change in y_i". It is a measure of the flexibility of the modeling procedure \$M\$. In the same that the degrees of freedom enable us to consider the complexity of the modeling space and the tendency to

overfit, the GDF enables us to consider the complexity of the modeling procedure and its tendency to overfit. In this way, the GDF depend on the "true" model and modeling procedure.

The notion of GDF was extended to a general class of modeling procedures using an optimal loss formulation which is consistent with the GDF definition given above (Xiaotong Shen & Huang, 2006).

3.3 Adaptive Model Selection Procedure

The adaptive selection criterion takes the form:

$$RSS(M) + \hat{\lambda} \mid M \mid .$$

$\hat{\lambda}$ is the data-adaptive penalty in the sense that it grows when the size of the true model is small, and shrinks when the size of the true model is large. It also adapts in the sense of approximating optimal performance over the class of information criteria. It is found that the optimal penalty is obtained by when the following expression is minimized, which coincides with the GDF as previously stated:

$$\hat{\lambda} = \arg\min_{\lambda} RSS + 2\widehat{G}(\lambda), \lambda \in (0, \infty) .$$

For variable selection in linear models over a class of models $M(\lambda)$, the procedure is as follows. $\hat{\lambda}$ is determined by minimizing the above expression. For each fixed λ, we determine the least squares fit for $M(\lambda)$. $\widehat{G}(\lambda)$ is found by using a Monte Carlo regression procedure (X. Shen & Ye, 2002):

- Sample δ_j from n-dimensional $N(0, \tau^2 \mathbf{I})$ where $\tau = 0.5\sigma$.
- Compute the $\hat{\mu}_{\hat{M}(\lambda)}(y + \delta_j), j = 1, \ldots, T$.
- Compute the regression slope $\hat{\mu}_{\hat{M}(\lambda)}(y + \delta_j) = a + \hat{\lambda}_i \delta_{ji}, j = 1, \ldots, T$.
    - $\widehat{G}(\lambda) = \sum_{i=1}^{n} \hat{\lambda}_i$.

3.4 GCV Analog

Ye defines a GDF analog (Ye, 1998) to the GCV criterion for model M as

$$GCV(M) = \frac{RSS(M)}{(n - GDF(M))^2}.$$

Given the GCV derivation from the previous chapter, we show how this is derived. Recall that after applying the leaving-one-out lemma is satisfied, we have:

$$\hat{f}(x_i) - \tilde{f}^{-i}(x_i) = \hat{f}(x_i) - \hat{f}^{-i}(x_i) = h_{ii}(y_i - \hat{y}_i^{-i})$$
$$\Rightarrow h_{ii} = \frac{\hat{y}_i - \hat{y}_i^{-i}}{y_i - \hat{y}_i^{-i}} = \frac{\Delta \hat{y}_i}{\Delta y_i} \approx \frac{\partial \hat{\mu}_i}{\partial y_i}.$$

Using this in the expression for the LOOCV, with the approximation $\sum h_{ii} = \sum \frac{\partial \hat{\mu}_i}{\partial y_i} = GDF(M)$, we arrive at:

$$GCV(M) = \frac{RSS(M)}{(1 - GDF(M)/n)^2}.$$

Minimizing this expression is equivalent to minimizing the one given by Ye.

4 SIMULATION

4.1 Adaptive Model Selection

We reproduce one of the simulations performed by Ye to better understand the adaptive model selection procedure through implementation. The code in the R language is provided in the appendix.

The chart below shows the MSE values for different model selection procedures against the number of true variables in the model. Our results are comparable to those of the paper. Note that in the legend, “AMS” refers to “Adaptive Model Selection” and K refers to the number of non-zero variables in the true model.

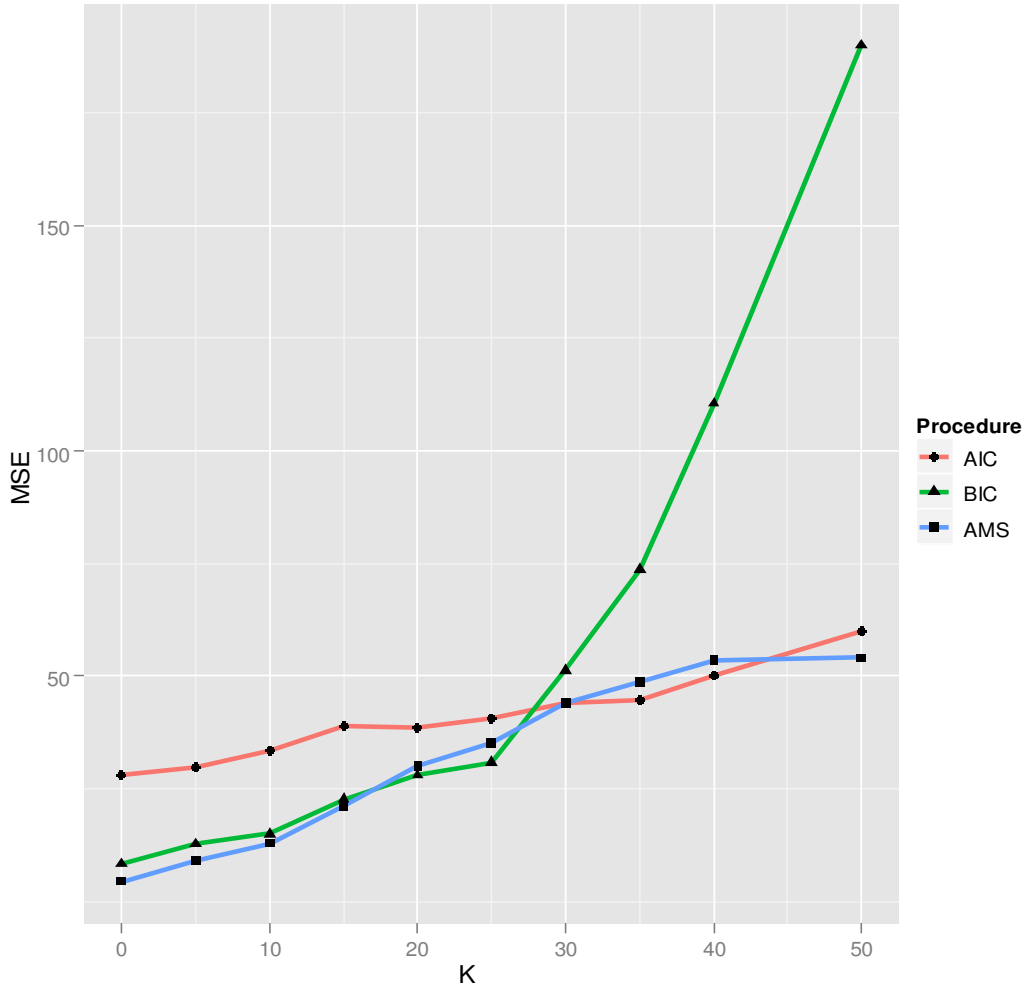

Figure 4.1 MSE from simulations for 3 procedures

The next figure shows the average number of variables selected by the different procedures for given number of non-zero variables in the true model.

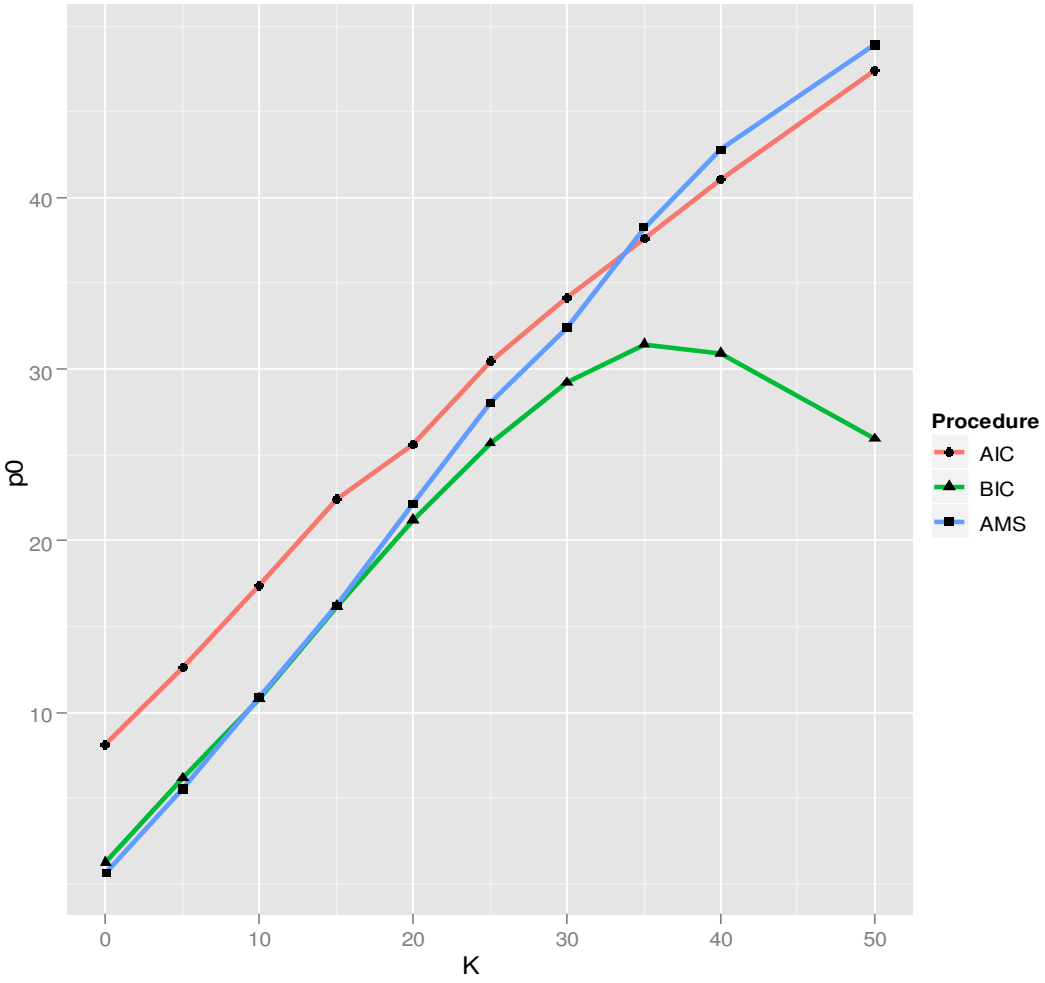

Figure 4.2 Number of variables selected by the different procedures

5 REFERENCES

Clarke, B., Fokoue, E., & Zhang, H. H. (2009). *Principles and Theory for Data Mining and Machine Learning (Springer Series in Statistics)* (p. 798). Springer. Retrieved from http://www.amazon.com/Principles-Machine-Learning-Springer-Statistics/dp/0387981349.

Geisser, S. (1975). The Predictive Sample Reuse Method with Applications. *Journal of the American Statistical Association*, *70*(350), 320-328. Retrieved from http://www.jstor.org/stable/2285815.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning*. *Book* (Vol. 2, p. 764). Springer. doi: 10.1007/978-0-387-84858-7.

Lee, H. (2007). Cross-validation for model selection. Retrieved from http://groundtruth.info/AstroStat/slog/2007/cross-validation-for-model-selection/.

Shao, J. (1993). Linear Model Selection by Cross-Validation. *Journal of the American Statistical Association*, *88*(422), 486. doi: 10.2307/2290328.

Shen, X., & Ye, J. (2002). Adaptive model selection. *Journal of the American Statistical Association*, *97*(457), 210–221. ASA. Retrieved March 18, 2011, from http://pubs.amstat.org/doi/pdf/10.1198/016214502753479356.

Shen, Xiaotong, & Huang, H.-C. (2006). Optimal Model Assessment, Selection, and Combination. *Journal of the American Statistical Association*, *101*(474), 554-568. doi: 10.1198/016214505000001078.

Sima, D. M. (2006). *Regularization Techniques in Model Fitting and Parameter Estimation*. Retrieved from ftp://ftp.esat.kuleuven.ac.be/pub/SISTA/dsima/reports/thesisDianaSima.html.

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*(1), 44–47. JSTOR. Retrieved March 18, 2011, from http://www.jstor.org/stable/2984877.

Sylvain, A., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, *4*, 40-79. doi: 10.1214/09-SS054.

Wahba, G. (1990). *Spline Models for Observational Data*. Society for industrial and applied mathematics.

Walker, H. M. (1940). Degrees of freedom. *Journal of Educational Psychology*, *31*(4), 253-269. doi: 10.1037/h0054588.

Ye, J. (1998). On Measuring and Correcting the Effects of Data Mining and Model Selection. *Journal of the American Statistical Association*, *93*(441), 120- 131. American Statistical Association. Retrieved March 18, 2011, from http://www.questia.com/PM.qst?a=o&se=gglsc&d=5002287558.

Zhang, B. (2010). *Adaptive model selection in linear mixed models*. University of Minnesota. Retrieved April 27, 2011, from http://gradworks.umi.com/33/71/3371928.html.

6 APPENDICES

Appendix A

```
rm(list=ls())

if (.Platform$OS.type == 'unix') {
    require(doMC)
    registerDoMC()
}
else if (.Platform$OS.type == 'windows') {
    require(doSNOW)
    .clusters <- makeCluster(2, type='SOCK')
    registerDoSNOW(.clusters)
}

require(MASS)
require(DAAG)
require(leaps)
require(foreach)

cat('getDoParWorkers', getDoParWorkers(), '\n')
cat('--- Start script:', date(), '\n')
start <- Sys.time()

#------------------------------------------
# utils
#------------------------------------------

# return px1 B such that theoretical R.sq = r.sq
# p must be multiple of 5
# k: num non-zero coefs per 5-vector subset
find.beta <- function(x, n, p, k, r.sq) {
       k.max <- 10
       if (k == 0) {
              B <- rep(0, p)
       } else {
         bk <- rep(c(1, 0), c(k, k.max-k))
         b <- rep(bk, p/k.max)
         xx <- t(x) %*% x
         B.k.sq <- (r.sq * n) / ( (1 - r.sq) * (t(b) %*% xx %*% b) )
         B <- sqrt(B.k.sq[1]) * b
    }
    array(B, dim=c(p, 1), dimnames=list(paste('b', 1:p, sep='')))
}

gen.x <- function(n, p, x.Sigma) {
    require(mvtnorm)
    x <- rmvnorm(n, mean=rep(0, p), sigma=x.Sigma)
       colnames(x) <- paste('x', 1:p, sep='')
       x
}
```

```
	gen.vars <- function(n, p, k, r.sq, x.Sigma, y.sigma) {
		x <- gen.x(n, p, x.Sigma)

	    b <- find.beta(x, n, p, k, r.sq)
	    mu <- x %*% b

	    y <- mu + array(rnorm(n, mean=0, sd=y.sigma), dim=c(n, 1))
	    colnames(y) <- 'y'

	    fo.full <- formula(paste('y~0+', paste('x', 1:p, sep='', collapse='+')))

	    list(y=y, x=x, b=b, mu=mu, fo.full=fo.full)
	}

	fit.step <- function(lambda, data) {
		fit.lower <- lm(y ~ 0, data=data)
		fit <- stepAIC(fit.lower, scope=form.upper, k=lambda, direc-
tion='forward', trace=FALSE)
		fit
	}

	adaptive <- function(y, x, p) {
		data <- as.data.frame(cbind(y, x))
		pert <- replicate(pert.T, rnorm(n, mean=0, sd=pert.tau))
		pert.y <- pert + matrix(y, nrow=n, ncol=pert.T, byrow=FALSE)

		pert.subs <- foreach(j=1:pert.T, .packages=c('leaps')) %dopar% {
			subs <- regsubsets(x, pert.y[ , j], method='forward', nvmax=p,
                                            intercept=FALSE, really.big=TRUE)
			summary(subs)
		}

		cache.g <- cache.G <- cache.lam <- c()
		f.g <- function(lambda) {
			lam <- round(lambda, 2)

			pkg <- c('leaps')
			expt <- c('pert.subs', 'x', 'p')
			pert.mu <- foreach(j=1:pert.T, .combine=cbind, .export=expt,
.packages=pkg) %dopar% {
				ye.ic <- pert.subs[[j]]$rss + lam * (1:p)
				model.idx <- which.min(ye.ic)
				coef.idx <- pert.subs[[j]]$which[model.idx, ]

				b <- rep(0, p)
				b[coef.idx] <- coef(pert.subs[[j]]$obj, model.idx)
				bb <- matrix(b, nrow=50, ncol=1)
				fit <- x %*% b
				fit
			}

			g0 <- foreach(i=1:n, .combine=sum, .export=c('pert')) %dopar% {
				fit.sens <- lm(pert.mu[i, ] ~ pert[i, ])
				coef(fit.sens)[2]
			}
```

```
		return( g0 )
	}

	f.G <- function(lambda) {
		lam <- round(lambda, 2)
		if (any(cache.lam == lam)) {
			return( cache.G[which(cache.lam == lam)] )
		}

		g0 <- f.g(lam)
		fit <- fit.step(lam, data)
		G <- sum( resid(fit)^2 ) + g0

		cache.lam <<- c(cache.lam, lam)
		cache.g <<- c(cache.g, g0)
		cache.G <<- c(cache.G, G)

		return( G )
	}

	res <- optimize(f.G, interval=c(0, 20))
	lambda.hat <- round(res$minimum, 2)
	g0 <- cache.g[which(cache.lam == lambda.hat)]
	return( list(lam=lambda.hat, gdf=g0/2) )
}


#-------------------------------------------
# constants
#-------------------------------------------
p <- 50
n <- 200
ks <- c(0, 3, 7, 10)
#ks <- c(1, 2, 4, 5, 6, 8, 9)
nsims <- 50

r.sq <- 0.75
y.sigma <- 1
x.Sigma <- diag(p)

pert.T <- n + 20
pert.tau <- 0.5

# full model as upper bound for stepwise selection
form.upper <- formula(paste('~0+', paste('x', 1:p, sep='', collapse='+')))

cat('p:', p, 'n:', n, 'nsims:', nsims, 'pert.T:', pert.T, '\n')


k.results <- list()
for (k in ks) {
	cat('  k', k, date(), '\n')
	s1 <- Sys.time()
```

```
            coln <- c('mse.aic', 'mse.bic', 'mse.lam', 'p0.aic', 'p0.bic', 'p0.lam',
                        'gdf', 'lambda', 'k', 'p', 'n')
            sim.results <- array(NA, dim=c(nsims, 11), dimnames=list(sim=1:nsims,
coln))

            for (isim in 1:nsims) {
                  cat('    isim', isim, date(), '\n')
                  s2 <- Sys.time()

                  vars <- gen.vars(n, p, k, r.sq, x.Sigma, y.sigma)

                  res <- adaptive(vars$y, vars$x, p)
                  cat('      lambda:', res$lam, ', gdf:', res$gdf, '\n')

                  data <- as.data.frame(cbind(vars$y, vars$x))
                  pkg <- c('MASS')
                  penalties <- c(aic=2, bic=log(n), lam=res$lam)
                  res.fits <- foreach(pen=penalties, .combine=cbind, .packages=pkg)
%dopar% {
                        fit <- fit.step(pen, data)
                        p0 <- length(coef(fit))
                        mse <- sum( (vars$mu - fitted(fit))^2 )
                        c(mse=mse, p0=p0)
                  }
                  colnames(res.fits) <- names(res.fits)
                  print(res.fits)

                  sim.results[isim, ] <- c(res.fits[1, ], res.fits[2, ],
gdf=res$gdf, lambda=res$lam,
                                                      k=k, p=p, n=n)
                  fname <- paste('sim-k', k, 's', nsims, 'Rd', sep='.')
                  save(sim.results, file=fname)

                  f2 <- Sys.time()
                  runt <- as.numeric(difftime(f2, s2, units='secs'))
                  cat('    time:', runt, 'secs,', round(runt/60, 1), 'mins',
                        round(runt/60/60, 2), 'hours', '\n')
            }

            k.results[[as.character(k)]] <- sim.results
            fname <- paste('k-k', k, 'ns', nsims, 'Rd', sep='.')
            save(k.results, file=fname)

            f1 <- Sys.time()
            runt <- as.numeric(difftime(f1, s1, units='secs'))
            cat('  time:', runt, 'secs,', round(runt/60, 1), 'mins',
                  round(runt/60/60, 2), 'hours', '\n')
      }



      cat('--- End script:', date(), '\n')
      end <- Sys.time()
      runtime <- as.numeric(difftime(end, start, units='secs'))
```

```
cat('Run time:', runtime, 'secs,', round(runtime/60, 1), 'mins',
       round(runtime/60/60, 2), 'hours', '\n')

if (.Platform$OS.type == 'windows') {
    stopCluster(.clusters)
}
```