C# .NET Algorithm for Variable Selection Based on the Mallow's C_p Criterion

Jessie Chen, MEng.
Massachusetts Institute of Technology, Cambridge, MA
jic@mit.edu

Abstract: *Variable selection techniques are important in statistical modeling because they seek to simultaneously reduce the chances of data overfitting and to minimize the effects of omission bias. The Linear or Ordinary Least Squared regression model is particularly useful in variable selection because of its association with certain optimality criterions. One of these is the Mallow's C_p Criterion which evaluates the fit of a regression model by the squared distance between its predictions and the true values. The first part of this project seeks to implement an algorithm in C# .NET for variable selection using the Mallow's C_p Criterion and also to test the viability of using a greedy version of such an algorithm in reducing computational costs. The second half aims to verify the results of the algorithm through logistic regression. The results affirmed the use of a greedy algorithm, and the logistic regression models also confirmed the Mallow's C_p results. However, further studies on the details of the Mallow's C_p algorithm, a calibrated logistic regression modeling process, and perhaps incorporation of techniques such as cross-validation may also be useful before drawing final conclusions concerning the reliability of the algorithm implemented.* *Keywords: variable selection; overfitting; omission bias; linear least squared regression; Mallow's C_p; logistic regression; C-Index*

Background

Variable Selection

Variable selection is an area of study concerned with the strategies for selecting one subset out of a pool of independent variables that is able to explain or predict the dependent variable well enough, such that all contributions from the variables that remain unselected may be neglected or considered pure error [13]. Explanation and prediction are the two main goals of variable selection; But while the two are distinct-- a regression equation which gives a good prediction might not be very plausible from a theoretical viewpoint-- the techniques used for variable selection are generally identical in both cases [13]. Because the predictor variables are almost always intercorrelated, the values of parameter estimates will likely change whenever a predictor is either included or eliminated [8]. Therefore, it is crucial to monitor the parameters closely in the variable selection process.

Parameters play a crucial role in understanding *overfitting*, a term for fitting a regression model with more variables than actually needed. Given a matrix X containing the values of all predictor variables, and vector Y containing the dependent variable values, matrix algebra will allow us to find the optimal set of coefficients for the system of equations by multiply the pseudoinverse by Y as follows [4]:

$$\beta = (X^T X)^{-1} X^T Y$$

Thus, the optimal parameters for a subset X_A of X is $\beta_A = (X_A^T X_A)^{-1} X_A^T Y$ (the estimator for the true β_{ATrue} value). Since it can also be shown that $var(x^T \beta) \geq var(x_A^T \beta_A)$, one can conclude that the variability of the predicted value $Y_A = x_A^T\beta_A$ is generally reduced when the prediction is based on a subset of all available predictors [13]. On the other hand, selecting too few variables can result in what is known as *omission bias*. Supposing that X_B now contains all variables in X not included in X_A and that at least one predictor in set B is nonredundant. The expected value $E(\beta_A)$ can now be calculated as

$$E(\beta_A) = \beta_{ATrue} + (X_A^T X_A)^T X_A^T X_B \beta_{BTrue}$$

with the second term representing the shift between the true value of β_{ATrue} and the expected value of its estimator β_A. The bias of the prediction is then,

$$bias(Y_A) = bias(x_A^T\beta_A) = x_A^T - x_A^T(X_A^T X_A)^T X_A^T X_B \beta_B.$$

In summary, the aim of variable selection is to select just enough variables so that such an omission bias is small, but at the same time to refrain from increasing the variance of the prediction more than necessary and thus resulting in overfitting [13]. In addition, variable selection techniques can generally be divided into two groups: Stepwise and Best-Subset. The first enters or removes variables only one at a time, and hence can be performed with large numbers of independent variables, but may often overlook good subsets of predictors. The second almost guarantees to find the best subset for each number of predictors but can only be performed when the number of predictors is small [2] [7] [13].

Linear Least Squared Regression

Linear Least Squared Regression, or *Ordinary Least Squared Regression* (OLS) is one method of variable selection that can be used to perform variable selection when working with binary, or indicator, dependent variables [6]. A regression model assumes the following two statistical relations about the data in question: 1) That for each value of X, there exists a probability distribution of Y, and 2) that the expected values of these probability distributions of Y vary systematically with X [10]. In OLS, this relationship is assumed to be linear.

OLS can be a valuable model for variable selection because associated with OLS are certain *optimality criterions*, used to determine how well a regression model fits the data, that can be employed in performing Best-Subset types of variable selection. One of these which makes use of the *residual sum of squares* value obtained from an OLS Regression model is the Mallow's C_p Criterion.

Mallow's C_p

In Mallow's C_p, it is first assumed that the model with all the predictors is the correct model and thus estimates the true residual variance σ_{True}^2 by

$$\sigma^2 = RSS(k) / (n-k)$$

where *k* is the number of available predictors, *n* denotes the number of observations, and RSS(k) is the *residual sum of squares* (the sum of the square of the difference between the observed value of the dependent variable and the value predicted by the model for all data points) with all the predictors in the model, or of the true model by assumption [1] [13]. Then, letting RSS(*p*) be the residual sum of squares with only *p* of the *k* predictors, Mallow's C_p is given as

$$C_p = RSS(p) / \sigma^2 \ - \ (n-2p).$$

From a statistical viewpoint, Mallow's C_p aims to minimize the expression

$$(1/\sigma^2)\ E(\hat{y}(p) - \mu)^T (\hat{y}(p) - \mu)$$

where σ^2 is used as a scale parameter, $\hat{y}(p)$ is the prediction using only *p* predictors, and μ is the true but unknown mean response [13]. Thus, it evaluates the fit of a regression model by the squared distance between the true value and the model's prediction. And because this formula is an expected value of quadratic form involving population parameters typically unknown, Mallow's C_p is meant to be an estimator of this expression [13]. It follows then, that if *p* predictors are sufficient to provide a good description of the data, then Mallow's C_p will have the same scale of magnitude as the distance between $\hat{y}(p)$ and μ. Also, if a subset of *p* predictors can explain the dependent variable well, then the expected value of C_p can be shown to be $E(C_p) = p\ [2(k-p) / (n-k-2)]$, implying that $E(C_p)$ approaches *p* when $n >> k$ and *p*. Putting these facts together, we can derive that a good model will yield a C_p value that is 1) small and 2) near *p*.

Problems with Using OLS on Binary Dependent Variables

While criterions like Mallow's C_p make the OLS model valuable to the variable selection process, there are, nonetheless, a few shortcomings associated with using a Linear Least Squared model when handling dependent variables that are binary. The first of these is that in a linear model, the predicted values will become greater than 1 or less than zero far enough down the extremes of the x-axis and such values are theoretically inadmissible [3]. The second is that, homoscedasticity, the assumption that the variance of Y is constant across all values of X cannot be the case with a binary dependent variable since the variance of Y is, in fact, equal to the product of the probabilities of getting a 1 or 0 [3]. Finally, when performing significance testing on the β parameters, OLS makes the assumption that all errors of prediction are normally distributed [3]. This can hardly be the case when Y takes on only the values 0 and 1 [3].

As a result, other models of regression have been proposed to address these concerns. These include the Weighted Least Squares Regression model which takes into account heteroscedasticity. However, the candidate that has been the most successful in handling all three drawbacks is the Logistic Regression model.

Logistic Regression

The Logistic Regression equation

$$\text{Logit}(p_i) = \ln (p_i / (1-p_i) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_n x_{in}$$

relates p_i, the probability of getting a dependent variable value of 1, to a linear combination of the predictor variables [6]. Associated with Logistic Regression is a different set of significance tests used to determine inclusion or elimination of each β coefficient from the model including the Wald Test, the Likelihood-Ratio Test, and the Hosmer-Lemeshow Goodness of Fit Test [5]. One method for determining how accurately a particular Logistic Regression model fits a set of data, however, is to calculate and examine the *C-Index* value of the model [12]. This number is equivalent to the area under the Receiver Operating Characteristics (ROC) curve, and thus will suggest that a model is a good fit if its value approaches 1 [12].

Overview of Project

In light of the above, this project seeks to make use of the Mallow's C_p Criterion in an algorithm for variable selection, and thus employs the Linear Least Squared (OLS) model of regression. However, though Mallow's C_p is considered a Best-Subset type algorithm for variable selection, there is a possibility of lowering computational cost by implementing such an algorithm in a greedy manner [15]. This hypothesis is tested by running Mallow's C_p on all possible subsets of predictor variables, and comparing the results to those of subsets selected through a greedy version of the algorithm. The aforementioned shortcomings of OLS are also taken into account by verifying the results of the Mallow's C_p algorithms using logistic regression modeling. Instead of regenerating all possible subsets, or even using a stepwise method for calibrating the model, logistic regression is simply run on all the subsets already returned by Mallow's C_p as the optimal subset for each possible subset size. The C-Indices of each of these models are then calculated and used to verify the original Mallow's C_p results.

Material and Methods

For algorithm testing, the Pima Indian Diabetes dataset from the UCI Machine Learning Repository was used. This data set consisted of values for eight different predictor variables and a binary dependent variable indicating whether the individual had been diagnosed with diabetes. The eight independent variables are 1. number of times pregnant, 2. Plasma glucose concentration after 2 hours in an oral glucose tolerance test, 3. Diastolic blood pressure (mm Hg), 4. Triceps skin fold thickness (mm), 5. 2-Hour serum insulin (mu U/ml), 6. Body mass index (weight in kg/(height in m)^2), 7. Diabetes pedigree function, and 8. Age (years) [14].

The algorithm itself was implemented using the .NET C# library within the Microsoft Visual Studios environment. For verification, C# was used to generate data files of the right data format, and logistic regression was then performed on the data sets through the "Logistic Regression Calculating Page" [9].

Table 1 contains a summary of the functions implemented as part of the overall algorithm.

Table 1: Function Summaries

Function Name	Input Parameter(s)	Return Value	Function Calls	Description
1. ReadData	string filename			Reads data contained in "data.txt" into ArrayList data.
2. FindVariance			FullParameters	Calculates Residual Sum of Squares and Variance of full set of predictor variables
3. FullParameters		ArrayList X	ComputeParameters	Prepares data and returns computed parameter for full set of predictor variables
4. ComputeParameters	ArrayList A, ArrayList Y	ArrayList X	MatrixTranspose, MatrixMultiply, MatrixInverse	Calcualte and return value of $(A^TA)^{-1}A^TY$
5. MatrixInverse	ArrayList AO	ArrayList inverse	PrintMatrix PrintRow	Perform Gauss-Jordan Elimination (including row swaps) to find and return inverse of AO
6. MatrixTranspose	ArrayList A	ArrayList AT	PrintMatrix	Generates and returns the transpose of matrix A
7. MatrixMultiply	ArrayList A ArrayList B	ArrayList product	PrintMatrix	Multiplies matrices A and B and returns product
8. PrintMatrix	ArrayList M			Prints matrix M to console for debugging purposes
9. PrintRow	ArrayList R			Prints row R of a matrix to console for debugging purposes
10. MallowsCpGreedy			ComputeParameters, Quicksort	Performs the greedy version of Mallow's C_p and prints result to text file "CpTable.txt" (See below for details)
11. MallowsCpAll			CalculateCps	Performs Mallow's C_p on all possible subsets of predictor variables and prints result to text file "CpTableAll.txt" (See below for details)
12. ComputeCps	ArrayList subsetList, int size		ComputeParameters, Quicksort	Helper function to MallowsCpAll that does the bulk of the

				computation (See below for details)
13. GenerateFiles				Generate data files of the proper format for all subsets of predictor variables that are to be used in the logistic regression validation process
14. GetPValues	string FileName, int lines, int start	ArrayList pValues		Read in a logistic regression output file and the necessary line and character locator and returns the predicted values for the dependent variable calculated by the system
15. FindCIndex	ArrayList P			Takes a list of predicted values for the dependent variable, calculates the C-Index for the system, and writes the result to text file “CIndices.txt” (See below for details)
16. Main	string[] args			Instantiates a regression system and makes all the necessary function calls to perform required tasks

MallowsCpGreedy (Algorithm Details)

The MallowsCpGreedy function maintains an “existingSet” and a “remainingSet” of variables. ExistingSet is initialized to an empty ArrayList, while remainingSet is initialized to contain all possible predictor variables. While there is still at least one item remaining in remainingSet, MallowsCpGreedy performs a *for* loop over all variables remaining in remainingSet, adding each one to the current existingSet one by one (The temporary variable used to store this subset of existingSet plus one variable from the remainingSet is called “currentSet.”) In each iteration of the *for* loop, MallowsCpGreedy

1. Generates a new data set from the full data ArrayList, including only entries from variables in the current currentSet.
2. Computes the optimal parameters for the current set of predictors
3. Calculates the residual sum of squares for the current OLS model
4. Computes the C_p value and adds it to a running list called CpList along with the index and number of the particular predictor variable added to the existingSet during this iteration of the *for* loop

When the for loop has completed, the CpList is run through a Quicksort algorithm that sorts the list entries by their C_p values. Data from the topmost, or minimum C_p, entry is then written to a text file called “CpTable.txt.” This predictor variable is then removed from the remainingSet and added to the existingSet. MallowsCpGreedy exits when there are no more variables remaining in the remainingSet.

MallowsCpAll (Algorithm Details)

The MallowsCpAll algorithm utilizes a system involving an integer counter "count" and bitwise comparators to generate all possible subsets of all possible sizes out of the set of all available predictor variables [4]. Each of the eight predictor variables are mapped to one bit of the integer counter, and inclusion and exclusion is indicated by the bit being set to 1 or 0 in each integer representation. The variable count is looped through the values 1 and 255 and the number of 1's in the least significant 8 bits are counted in each round. The predictor variables corresponding to the 1's are then added to a running list called "members." At the end of each round, a case statement parses the subset denoted by count into the appropriate bin according to the number of variables contained in members. After all subsets have been added, each list of subsets, grouped by size, is passed to the CalculateCps function for further processing.

CalculateCps (Algorithm Details)

The details of CalculateCps is very similar to those of MallowsCpGreedy. However, instead of looping through each subset of the existingSet plus one variable from the remaining set until the remainingSet is empty, CalculateCps loops through all the subsets in the ArrayList subsetList passed to it as a parameter. For each of these subsets, CalculateCps goes through the four steps outlined in MallowsCpGreedy: Generate new data set, compute optimal parameters, calculate residual sum of squares, and compute C_p value-- this time, adding the C_p value itself and the entire subset in question to the CpList. The algorithm then sorts the CpList by C_p values using Quicksort as in MallowsCpGreedy. And finally, CalculateCps writes the results for all the subsets contained in CpList to a text file "CpTableAll.txt."

FindCIndex (Algorithm Details)

The GetPValues function is first used to generate an ArrayList of probabilities calculated by a particular logistic regression system for the dependent variable for each of the data points. The FindCIndex algorithm then takes this ArrayList of P values and sorts them into "healthy" and "sick" bins based on the observed value for each data point. A double *for* loop is then used to tally the number of concordant, discordant, and tied pairs among the results. These counts are then used to calculate the C-Index for the particular logistic regression model in question.

Procedure

The Pima Indian Diabetes data from the UCI Machine Learning Repository was run through both the MallowsCpGreedy and MallowsCpAll algorithms. Because the Mallow's C_p is intrinsically a Best-Subset algorithm, the results were compared to verify whether a greedy algorithm can be used to lower computational cost without compromising accuracy. The GenerateFiles function was then used to generate datasets for all subsets returned by the MallowsCpGreedy algorithm. Logistic regression was performed on these generated datasets using the Logistic Regression Calculating Page [9]. The C-Index for each of these logistic regression models are then calculated using the GetPValues and FindCIndex functions. The Mallow's C_p and logistic regression results are then compared.

Results

After running the MallowsCpGreedy algorithm, "CpTable.txt" contained the following results:

Table 2: CpTable Output

Value of p (Including Intercept)	Number of Predictor Variables	Variables in Subset	C_p Value
2	1	1	86.199071926954

3	2	1, 5	47.0498293176931
4	3	1, 5, 0	19.365631341847
5	4	1, 5, 0, 6	10.945482286911
6	5	1, 5, 0, 6, 2	**5.91402469860009**
7	6	1, 5, 0, 6, 2, 7	4.59629230089502
8	7	1, 5, 0, 6, 2, 7, 4	5.01930150012004
9	8	1, 5, 0, 6, 2, 7, 4, 3	7.00000000000068

Table 3: Variable Mapping

Number	Corresponding Variable
0	Number of times pregnant
1	Glucose concentration after 2 hours in an oral glucose tolerance test
2	Diastolic blood pressure (mm Hg)
3	Triceps skin fold thickness (mm)
4	2-Hour serum insulin (mu U/ml)
5	Body mass index (weight in kg/(height in m)^2
6	Diabetes pedigree function
7	Age (years)

The C_p value that was the closest to its *p* value is the one corresponding to a subset of 5 selected predictors: Variables 1, 5, 0, 6, and 2. There was an error margin here of only 1.4% between the C_p value 5.91402469860009 and its *p* value of 6, while the error margin was 119% for the subset of size 5 (one less predictor), and 20% for the subset of size 7 (one more predictor). When the MallowsCpAll algorithm was performed on the same dataset, the results were identical (See Appendix D). That is, the same minimum C_p subsets for each subset size were chosen by the greedy and all-subsets algorithms, and moreover, the calculated C_p values were also identical for the two algorithms as expected.

When the subsets from Table 2 were used to generate logistic regression models, the C-Indices calculated were as follows:

Table 4: C-Indices from Logistic Regression Models

Number of Variables	Subset	C-Index
1	1	0.960820895522388
2	1, 5	0.962686567164179
3	1, 5, 0	0.98134328358209
4	1, 5, 0, 6	0.98134328358209
5	1, 5, 0, 6, 2	**0.988805970149254**
6	1, 5, 0, 6, 2, 7	**0.988805970149254**
7	1, 5, 0, 6, 2, 7, 4	0.985074626865672
8	1, 5, 0, 6, 2, 7, 4, 3	0.985074626865672

The subset with 5 predictor variables remain the subset with the highest C-Index and hence the model that most closely fits the given data. It also resulted in a tie with the subset of 6 variables when their C-Indices were compared.

Discussion

From the above results, it can be seen that the greedy version of the Mallow's C_p algorithm did indeed produce identical results as the version that took into account all possible subsets. By using the greedy algorithm, computational costs will be significantly reduced.

The test using C-Indices from logistic regression also seems to confirm that the subset with variables 1, 5, 0, 6, and 2 is the best subset of predictors that will created the best balance between the variance of the dependent variable and the omission bias. First, the subset of 5 variables chosen by Mallow's C_p resulted in the maximum C-Index value of 0.988805970149254. Then, even though the subset of size 6 resulted in the same C-Index value, we can still conclude from our data that the subset of 5 is our preferred set of predictors since adding variable 7 to the subset neither increased nor decreased the fit of our model. In other words, including variable 7 neither provided more useful information nor took away from the existing model. Therefore, all things being equal, it is generally preferable to go with the smaller subset (Reasons may include considerations such as reduced cost in data collection.)

Nonetheless, if we were to maintain our assumption that the logistic regression model is more error free and hence more accurate than the OLS, then it may still be wise to take into consideration an apparent discrepancy between the Mallow's C_p and logistic regression results: that while the logistic regression model indicated no difference between the inclusion and exclusion of Variable 7, the Mallow's C_p algorithm did indicate a distinction. This may be due to Mallow's C_p's being intrinsically aware of the advantages behind the exclusion of redundant variables. However, further study of the Mallow's C_p criterion will be necessary before such conclusions can be drawn. Running a calibrated logistic regression on the Pima Indians Diabetes dataset, independent from OLS, may also be useful for verifying whether the same subset would be chosen by logistic regression alone. Finally, cross-validation techniques can also be used increase the accuracy and to verify the results of both models [11].

Conclusion

As a result of this study, we can conclude that Mallow's C_p is a useful criterion to employ when generating models for variable selection based on Linear Least Squared regression. In addition, a greedy version of the algorithm not only allowed the number of computations to decrease significantly, but also appear to produce identical results as when all possible subsets were generated and taken into account. Finally, the Mallow's C_p results were verified by C-Index calculations in conjunction with logistic regression modeling. However, further research on the handling of seemingly redundant variables by Mallow's C_p, the use of an independent and calibrated logistic regression test, and the incorporation of techniques such as cross-validation for increased accuracy and verification of results would be useful before further conclusion as drawn.

References

[1] Anderson, David R., Dennis J. Sweeney, Thomas A. Williams. *Statistics: Concepts and Applications*. New York: West Publishing Company, 1986.

[2] Black Hill State University. "SPSS Logistic Regression Algorithm", at http://www.bhsu.edu/instres/logistic_regression.pdf, 2005.

[3] Brannick, Michael T. Personal website at http://luna.cas.usf.edu/~mbrannic/files/regression/Logistic.html, 2005.

[4] Chong, Hamilton. Personal Communications, 12/2005.

[5] Conner, Edward F. Personal website at http://userwww.sfsu.edu/~efc/classes/biol710/logistic/logisticreg.htm, 2005.

[6] Frees, Edward W. *Data Analysis Using Regression Models: The Business Perspective*. Upper Saddle River, NJ: Prentice Hall, 1996.

[7] Garson, David. Personal Website at http://www2.chass.ncsu.edu/garson/pa765/logistic.htm, 2005.

[8] Halekoh, Ulrich. "Module 5: Logistic Regression", at http://genetics.agrsci.dk/biometry/courses/statmaster/course/module05/index.html, 2005.

[9] "Logistic Regression Calculating Page" at http://members.aol.com/johnp71/logistic.html, 2005.

[10] Moskowitz, Herbert, Gordon P. Wright. *Statistics for Management and Economics*. London: Charles E. Publishing Co. and A Bell & Howell Company, 1985.

[11] Ohno-Machado, Lucila. "Cross-validation and Bootstrap Ensembles, Bagging, Boosting," 6.873/HST.951 Medical Decision Support Lecture Notes, Fall 2005.

[12] Ohno-Machado. "Evaluation," 6.873/HST.951 Medical Decision Support Lecture Notes, Fall 2005.

[13] Schuster, Christof. *Regression Analysis for Social Sciences*. New York: Academic Press, 1998.

[14] UCI Machine Learning Repository. "Pima-Indians-Diabetes." http://www.ics.uci.edu/~mlearn/databases/pima-indians-diabetes/, 2005.

[15] Vinterbo, Staal. In-Class Communications, 12/2005.

Appendices
(Found on Website at http://web.mit.edu/~jic/www/mds.htm)

A. C# Code Implementation
B. Pima Indian Dataset
C. CpTable.txt
D. CpTableAll.txt
E. Logistic Regression Results