2013

Curds and Whey: Little Miss Muffit's Contribution to Multivariate Linear Regression

John Cameron Kidd

CURDS AND WHEY: LITTLE MISS MUFFIT'S CONTRIBUTION TO MULTIVARIATE LINEAR REGRESSION

by

John Cameron Kidd

Thesis submitted in partial fulfillment
of the requirements for the degree

of

HONORS IN UNIVERSITY STUDIES
WITH DEPARTMENTAL HONORS

in

Statistics
in the Department of Mathematics and Statistics

Approved:

Thesis/Project Advisor Dr. Richard Cutler	**Departmental Honors Advisor** Dr. David Brown

Director of Honors Program
Dr. Nicholas Morrison

UTAH STATE UNIVERSITY
Logan, UT

Spring 2013

Curds and Whey: Little Miss Muffit's Contribution to Multivariate Linear Regression

John Kidd

April 18, 2013

Abstract

A common multivariate statistical problem is the prediction of two or more response variables using two or more predictor variables. The simplest model for this situation is the multivariate linear regression model. The standard least squares estimation for this model involves regressing each response variable separately on all the predictor variables. Breiman and Friedman [1] show how to take advantage of correlations among the response variables to increase the predictive accuracy for each of the response variable with an algorithm they call *Curds and Whey*. In this report, I describe an implementation of the Curds and Whey algorithm in the R language and environment for statistical computing [6], apply the algorithm to some example data sets, and discuss extensions of the algorithm to linear classification methods.

1 Introduction and Background

1.1 The Multiple Regression Model

One of the most widely used statistical methods is *multiple linear regression*, in which a numerical response is modeled as a linear combination of values on two or more numerical *predictor* or *explanatory* variables. Examples include predicting oxygen uptake as using fitness and anthropometric measurements on the subjects, insurance profits as using industry and economic variables, human mortality rates using measurements of socio-economic status and air pollution, and species abundances using ecological and climate measurements. The multiple linear regression model may be written as:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i \quad \text{for } i = 1, 2, \ldots, n \tag{1}$$

where Y_i is the value of the response variable for the i^{th} observation, $x_{i1}, x_{i2}, \ldots, x_{ip}$ are the values on the explanatory variables for the i^{th} observation, ε_i is a random error, and $\beta_0, \beta_1, \ldots, \beta_p$ are unknown parameters that must be estimated. Usually it is assumed that the ε_i are statistically independent, with common mean 0 and variance σ^2, and are approximately normal in distribution. This model in matrix form may be written as:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{2}$$

where

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

1.2 Least Squares Estimation and Prediction

Given a set of parameter estimates, $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots \hat{\beta}_p)^T$, one can compute *fitted* or *predicted* values, $\hat{Y}_i$, by substitution into equation (1) and setting the random error term equal to zero. That is,

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_p x_{ip} \quad \text{for } i = 1, 2, \ldots, n.$$

or, in matrix form, $\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$.

To estimate $\boldsymbol{\beta}$, we may minimize the sum of squared deviations between the observe response variable value, Y_i, and the predicted values, $\hat{Y}_i$. That is, we minimize $\mathrm{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y} = \mathbf{X}\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$. Assuming the columns of the matrix $\mathbf{X}$, the predictor variables, are linearly independent, the least squares estimator of $\boldsymbol{\beta}$ has the elegant form

$$\hat{\boldsymbol{\beta}}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{3}$$

1.3 Shrinkage Estimation

For the multiple linear regression model when the predictor variables are correlated, a different estimation procedure called *ridge regression* [3] yields more stable parameter estimates (the $\hat{\beta}_j$'s) and smaller prediction error. Following the notation of equation (3), the ridge regression estimate of $\boldsymbol{\beta}$ may be written

$$\hat{\boldsymbol{\beta}}_{(\tau)} = (\mathbf{X}^T\mathbf{X} + \tau\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}, \tag{4}$$

where $\mathbf{I}$ is the identity matrix with 1's down the diagonal and 0's in all the off-diagonal entries, and τ is a *shrinkage parameter*. Typically, τ is estimated by minimizing prediction error or by graphical means. Ridge regression is a *shrinkage estimation procedure* in the sense that as τ increases, the $\hat{\beta}_{j(\tau)}$'s decrease in magnitude, sometimes changing sign in the process. As τ gets very large, all the $\hat{\beta}_{j(\tau)}$'s will tend to zero.

Other forms of shrinkage have also been shown to provide more stable parameter estimates as well as smaller prediction errors [2,4,5,9,11].

1.4 Canonical Correlation

Canonical Correlation analysis is a method for characterizing the linear associations among two sets of variables. Let $\mathbf{X}$ be an $n \times p$ matrix with the columns being the measured values of one set of variables, and $\mathbf{Y}$ be an $n \times q$ matrix with the columns being the values on the other set of variables. Assume, without loss of generality, that $q \leq p$.

Let V_1 and W_1 be vectors such that $\mathbf{X}V_1$ and $\mathbf{Y}W_1$ maximizes the correlation among all linear combinations of variables in $\mathbf{X}$ and $\mathbf{Y}$. This maximal correlation, c_1, is the *first canonical correlation.*

Next, V_2 and W_2 are found so that $\mathbf{X}V_2$ and $\mathbf{Y}W_2$ maximizes the correlation among linear combinations of variables in $\mathbf{X}$ and $\mathbf{Y}$, *subject to the constraint that* $V_2 \perp V_1$ *and* $W_2 \perp W_1$. The correlation, c_2, is the *second canonical correlation.* The process is continued, yielding q canonical correlations and vectors $V_1, V_2, ...V_q$, and $W_1, W_2, ..., W_q$.

The vectors $W_1, W_2, ..., W_q$ may be stacked together to create a matrix $\mathbf{T}$ that may be used to transform the variables in $\mathbf{Y}$ into *canonical coordinates*, the coordinate system that yields the canonical correlations.

1.5 The Multivariate Linear Regression Model

The *multivariate* linear regression model extends the *multiple* linear regression model to predicting two or more response variables using the same suite of predictor variables. We may write the model as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{E}, \tag{5}$$

where $\mathbf{Y}$ is an $n \times q$ matrix, the columns of which are q response variables. In this model $\mathbf{X}$ is an $n \times (p+1)$ matrix comprising, as columns, p predictor variables and a column of 1's for the intercept term. $\mathbf{E}$ is an $n \times q$ matrix of residual or random error terms, and $\mathbf{B}$ is a $p \times q$ matrix of coefficients to be estimated. The k^{th} column of $\mathbf{B}$ is the vector of coefficients for the predictor variables for the k^{th} response variable.

The least squares estimate of $\mathbf{B}$ may be expressed as

$$\left[\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2, \ldots, \hat{\boldsymbol{\beta}}_q\right] = \hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} = \left[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}_1, (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}_2, \ldots, (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}_q\right]$$

Thus, the $\hat{\beta}_j$'s corresponding to the k^{th} response variable use only information from the k^{th} response variable. In situations in which the response variables are highly correlated this result seems counter intuitive, and it is this observation which motivated Breiman and Friedman [1] to develop an alternative approach. Their *Curds and Whey* algorithm employs elements of canonical correlation and shrinkage estimation to use the relationships among the response variables to enhance the accuracy of predictions for each of the response variables.

2 The Curds and Whey Procedure Setup

The general idea behind the Curds and Whey algorithm is to take the least squares regressions, and then to modify the predicted values from those regressions by shrinking them using the canonical correlations between the response variables and the predictor variables. Thus, the equation can be thought of as:

$$\tilde{\mathbf{Y}} = \mathbf{M}\hat{\mathbf{Y}} \tag{6}$$

Where $\mathbf{M}$ is the matrix estimated such that $\mathbf{M} = \mathbf{T}^{-1}\mathbf{D}\mathbf{T}$ with $\mathbf{T}$ being a $q \times q$ matrix, where q is the number of response variables one is trying to predict, whose rows are the canonical correlation coordinates of the response variables, and $\mathbf{D}$ is a diagonal matrix where each entry d_i is a function of the canonical correlations and the proportion or predictors to size of the data set (and each d_i is less than one). In order to search for the best $\mathbf{M}$, and to follow an intuitive progression, I will show two methods to find $\mathbf{M}$. One is very general and simple, and does not involve cross-validating the data, where the other produces a general cross validation rather than the full cross validation.

2.1 Standardizing Data

When dealing with multivariate data, often variables are measured on different scales. For many procedures, this can lead to one or more predictor variables having a much larger influence on the response than others simply because of its scale. This can make interpretation difficult, as well as cause extreme observations to influence results. Standardizing data puts all variables on similar scales and prevents variables from exhibiting large influence because of their scale alone.

With the Curds and Whey algorithm, not only do the predictor variables need to be on the same scale, but due to use of canonical correlations of the response variables, the response variables also need to be standardized. If they were not, there is a risk that one response variable that is on a larger scale may throw off some of the predictions for the other response variables.

The standardization used in this project was to subtract off the mean from all observations of a variable and then division by the standard deviation. This transformation is the ”conversion to z-scores” from introductory statistics classes.

2.2 Finding D

There are two was to estimate the optimal shrinking matrix, **D**. In the simplest case given $r = p/N$, with N being equal to the total number of observations, define the d_i's as:

$$d_i = \frac{c_i^2}{c_i^2 + r(1 - c_i^2)}, i = 1, 2, ..., q \tag{7}$$

This gives improved predictions compared to ordinary least squares, but it does not provide enough shrinkage to be optimal. A second approach, based upon generalized cross-validation sets the d_i's as follows:

$$d_i = \frac{(1 - r)(c_i^2 - r)}{(1 - r)^2 c_i^2 + r^2(1 - c_i^2)}, i = 1, 2, ..., q \tag{8}$$

In some instances, this will result in a d_i that is less than 0. In this case, the d_i's are restricted to 0.

3 The Procedure

The Curds and Whey algorithm follows these steps:

1. Standardize response and predictor variables.
2. Transform **Y** to the observed canonical coordinate system, $\mathbf{Y}^* = \mathbf{TY}$.
3. Perform a separate ordinary least squares regression of each of the Y_i*'s on all the predictor variables **X**, obtaining a new variable, we'll call $\hat{Y}_i$*.

4. Separately scale (shrink) each of the $\hat{Y_i}$*'s by the corresponding d_i (8). Or, it can be thought of as $\hat{\mathbf{Y}}$*. This gives a new set, called $\tilde{\mathbf{Y}}$*.
5. Transform back to the original $\mathbf{Y}$ coordinate system, $\tilde{\mathbf{Y}} = \mathbf{T}^{-1}\ \tilde{\mathbf{Y}}$*.

3.1 General Example

In general terms, the process can be described in a more straight forward manner in the language of a statistical package. I accompany this with coded examples from R. Say one has data, and to work with the *cancor* package [7], one has to split into two matrices or predictor and response variables, named accordingly. For the example, I use the following randomly generated data, which has 100 observations. There are 5 response variables on 20 predictor variables. (For reproducibility, I set the seed to 1000)

```
> set.seed(1000)
> predictors <- matrix(rnorm(2000, 50, 4), 100, 20)
> for(i in 1:10){
+   predictors[,i] <- predictors[,i] * i
+ }
> response <- predictors %*% matrix(rep(1 : 5, each = 20) , 20, 5) +
+                          matrix(rnorm(500, 0, 50), 100, 5)
```

First, I will standardize all of the data in two parts: one for the response variables and one for the predictor variables.

```
> for(i in 1:5){
+   mean <- mean(response[, i])
+   sd <- sd(response[, i])
+   response[, i]<-(response[, i] - mean) / sd
+ }
> for(i in 1:20){
+   mean <- mean(predictors[, i])
+   sd <- sd(predictors[, i])
+   predictors[, i]<-(predictors[, i] - mean) / sd
+ }
```

Next, I obtain the canonical correlations of the data, and transform the response variables into the canonical correlation coordinates. In R, this looks like:

```
> cancor.all <- cancor(predictors, response)
> cancor.cor <- cancor.all$cor
> cancor.y <- cancor.all$ycoef
> yPrime <- as.matrix(response) %*% cancor.y
```

Next I perform ordinary least squares regression on the transformed response variables on the original predictor variables, and find the predicted values from the OLS regression. In R:

```
> new.yPrime.data <- data.matrix(cbind(yPrime, predictors))
> yPrime.lm <- lm(new.yPrime.data[, 1:5] ~ new.yPrime.data[, 6:25])
> yhat.Prime <- yPrime.lm$fitted.values
```

Next, I obtain the shrinkage matrix, $\mathbf{D}$, by equation (3). In words, this is a bit difficult to describe. So, I rewrite it with several variables. Lets call $(1 - r)$ A. Then, call the i^{th} correlation squared subtract r B. Then, call $(1-r)$ squared C, just remember r^2 is r squared, and call 1 subtract the i^{th} correlation squared D. Then, (8) could be rewritten as:

$$d_i = \frac{A * B}{C * (c_i^2 + r^2 * D)} \tag{9}$$

Hopefully this formulation is simpler than (8). Continuing the example, just remember that there are 20 predictor variables (p), 100 observations (N), and 5 response variables (q).

```
> r <- 20 / 100
> di <- rep(0, 5)
> di <- {(1 - r) * ({cancor.cor^2} - r)} /
+   {({1 - r}^2) * (cancor.cor^2) + ({r^2} * {1 - cancor.cor^2}) }
> for(i in 1:5){
+   di[i] <- max(di[i], 0)
+ }
> D <- diag(di)
```

At this point, I take the last couple of steps to get the final $\tilde{\mathbf{Y}}$ by shrinking $\hat{\mathbf{Y}}^*$ (*yhat.Prime* in R), to yield $\tilde{\mathbf{Y}}^*$ (*yhatstar* in the following R code). I then transform back into the original $\mathbf{Y}$ coordinate system by multiplying $\tilde{\mathbf{Y}}^*$ by the inverse of the canonical coordinate matrix $\mathbf{T}^{-1}$, (*cancor.y* in R), which yields $\tilde{\mathbf{Y}}$ (*yFinal* in R).

```
> yhatstar <- yhat.Prime %*% D

> yFinal <- yhatstar %*% solve(cancor.y)
```

For some ease, and further usability, one can extract some more information from the data using another least squares regression. If one regresses the new $\tilde{\mathbf{Y}}$ on all of the predictor variables, one can argue that one has the coefficients needed to predict $\tilde{\mathbf{Y}}$ from the original predictor variables.

```
> yFinal.lm <- lm(yFinal~predictors)

> CurdsCoeffs <- yFinal.lm$coefficients
```

It is possible to do some checks to verify that this has worked. To save room, recall that there are 100 observations, and 5 response variables. Therefore, there are 500 measurements. Using R, it can be checked to see if the predictors (with a column of 1's added for the intercept term), multiplied by the coefficients equals the $\tilde{\boldsymbol{Y}}$ (yFinal) matrix. Each test of equivalency will return a 1 if the two values are equal, and a 0 if they are unequal. Thus, summing up all the results will give the number that are equal between the two. Rounding to 8 or so places to account for computer rounding error may also be advisable.

```
> sum( round( cbind( 1, predictors) %*% CurdsCoeffs, 8) ==

+                        round( yFinal, 8))

[1] 500
```

Therefore, it can concluded that the given coefficients will predict the $\tilde{y}$ from the original predictor variables.

3.2 Measuring Improvement

In trying to measure the improvement achieved in Curds and Whey over ordinary least squares regression. In the original paper, Breiman and Friedman [1] suggest using the *predictive error*. This involves finding a model for the data without observation n being included, finding the predicted value for observation n from the model, and then finding the residual between the predicted value for Y_n ($\hat{Y}_n$) and the actual Y_n and squaring the difference ($(\hat{Y}_n - Y_n)^2$). One then finds the average for all N observations. Define $Pred_i$ to be the predictive error for the i^{th} response variable. It may be written as:

$$Pred_i = \frac{1}{N}\sum_{n=1}^{N}(\hat{Y}_n - Y_n)^2, i = 1, 2, ..., q \tag{10}$$

Where q is the number of response variables, and, as before, $\hat{Y}_n$ is the predicted value for observation n with the n^{th} observation dropped. That is, one finds a model with Y_n being excluded from the model, then using the predictor variable values for observation n and the model, finds a predicted value for Y_n, ($\hat{Y}_n$). Finally, square the differences, and find the average.

One then can find the average predictive error by the following:

$$Pred_{ave} = \frac{1}{q}\sum_{i=1}^{q} Pred_i, i = 1, 2, ..., q \tag{11}$$

In ordinary least squares regression, this is very straightforward and simple to compute. There are some diagnostic tools for finding the residual with the n^{th} value omitted. This is done by taking the residual for the n^{th} observation from the OLS model, dividing by one minus the n^{th} diagonal of the hat matrix, and then squaring that quantity. If r is the vector of residuals and $\mathbf{H}$ is the hat matrix, then use:

$$(\frac{r_n}{1 - H_{n,n}})^2 \tag{12}$$

Then, taking the sum of all the values and dividing by N to get the predictive error for the q^{th} response variable.

To do this in R, I will make use of the *lm.influence* call in the MASS package [10]. This package gives influence diagnostics for ordinary least squares models fitted by the lm function in the stats package [7]. I am able to pull the hat matrix easily from the influence function. Then take the average of the residuals divided by the hat matrix diagonals (given from ”hat”) squared. A loop must be used for each response variable.

```
> example.data <- cbind(response, predictors)

> preError.lm <- lm(example.data[, 1:5] ~ example.data[, 6:25])

> preError <- rep(0, 6)

> for(i in 1: 5){

+   preError[i] <- sum(     (preError.lm$residuals[, i] /

+                       (1 - lm.influence(preError.lm)$hat))**2)    /

+                                        100

+ }

> preError[6] <- sum(preError[1:5] / 5)

> preError
```

```
[1] 0.38204778 0.09367968 0.05548279 0.03263117 0.01470052 0.11570839
```

Here, *preError[6]* and *preError.std[6]* are the average of the predictive errors for the 5 response variables. This is called preError in R because this is the predictive error before running Curds and Whey.

With regards to the Predictive Error after Curds and Whey, due to the shrinkage, I have not been able to find a quick method, and have had to resort to brute force by running loops through to drop each observation and to fit a new model. This turns into a process that is easily implemented into a function. This function will allow us to output the error from the ordinary least squares regression as well as the predictive error after Curds and Whey.

The code for the function is omitted, but the values of the ”preError” and ”postError” were saved as example.preError and example.postError. I then put those into a matrix called compare for easy comparison. One should see on the first line that *example.preError* is exactly equal to *preError.std* from above. Then, one can compare the predictive errors for ordinary least squares and for Curds and Whey (*pre and post error*).

```
> preError

[1] 0.38204778 0.09367968 0.05548279 0.03263117 0.01470052 0.11570839

> example.preError

[1] 0.38204778 0.09367968 0.05548279 0.03263117 0.01470052 0.11570839

> compare

     Ordinary Least Squares       Curds and Whey

[1,]             0.38204778           0.27894416
```

```
[2,]             0.09367968           0.08207534

[3,]             0.05548279           0.04317539

[4,]             0.03263117           0.02884861

[5,]             0.01470052           0.01325166

[6,]             0.11570839           0.08925903
```

For some perspective, one can look at the differences and percentages of change:

```
      Difference   Percentage

[1,] 0.103103617    26.987101

[2,] 0.011604345    12.387260

[3,] 0.012307408    22.182386

[4,] 0.003782560    11.591864

[5,] 0.001448858     9.855829

[6,] 0.026449358    22.858635
```

The difference column is the difference of the OLS prediction error and the Curds and Whey prediction error. One can see that every value is positive, showing and improvement for every predictor. The second column is the percentage of improvement from the OLS to Curds and Whey procedure. (This is calculated by finding what percentage the Curds and Whey error is from the OLS error, and subtracting that from 1 to find the improvement, and then displayed as a percentage). The smallest improvement is almost 10% in this scenario, with improvement increasing up to almost 27%.

4 Real World Examples

4.1 Chemometrics

An example given by Breiman and Friedman [1] deals with chemometrics. In this data set, take from [8]. There are 56 observations ($N = 56$), each with 22 predictor variables ($p = 22$), and 6 responses ($q = 6$). The data were taken from a simulation of a low density tubular polyethylene reactor. The predictor variables are all temperatures measured at equal distances along reactor together with the wall temperature of the reactor and feed rate. The responses are:

- y_1: number-average molecular weight
- y_2: weight-average molecular weight
- y_3: frequency of long chain branching
- y_4: frequency of short chain branching
- y_5: content of vinyl groups
- y_6: content of vinylidene groups

A log transformation was applied to all six response variables to correct for right-skewness. The average absolute correlation of response variables is .48, but it can be seen in the following table that y_3 is more weakly correlated with the other variables. In some cases, the correlation between y_3 and the other variables is actually negative.

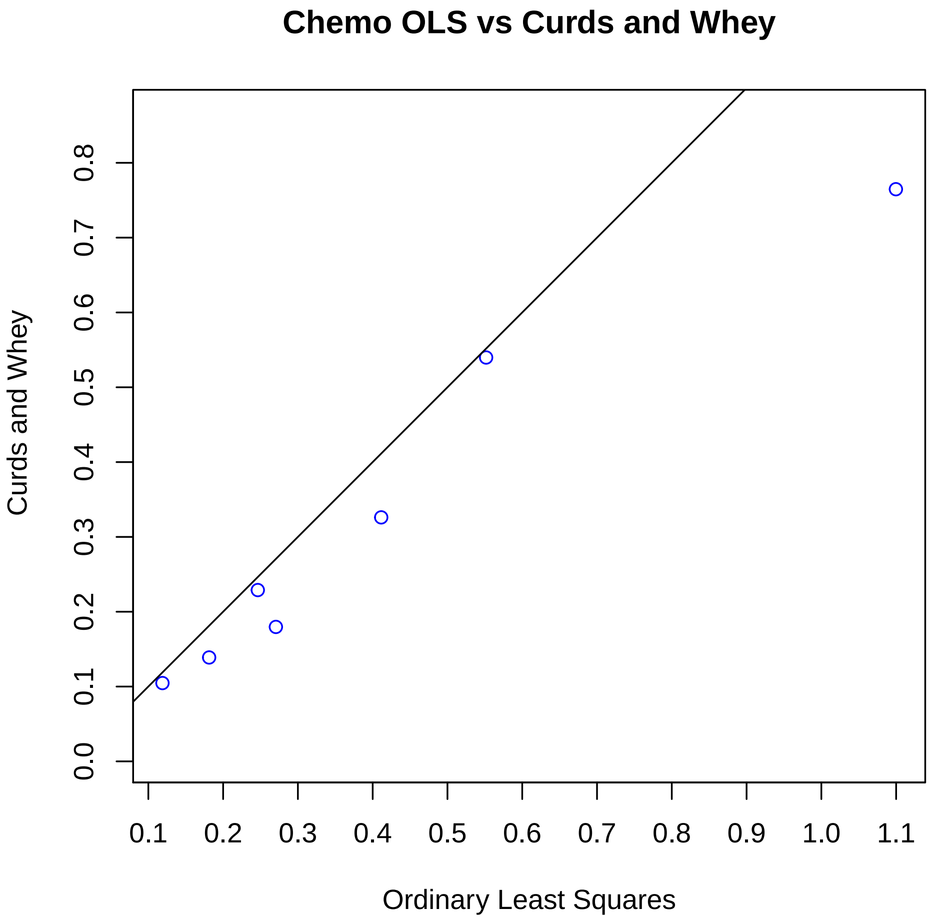
```
           lr1        lr2         lr3        lr4        lr5        lr6
lr1 1.00000000  0.9566719  0.06507871  0.2543081  0.2551151  0.2591868
lr2 0.95667189  1.0000000 -0.12843581  0.2824976  0.2655915  0.2755877
lr3 0.06507871 -0.1284358  1.00000000 -0.4997273 -0.4839793 -0.4787396
lr4 0.25430811  0.2824976 -0.49972730  1.0000000  0.9744166  0.9782465
lr5 0.25511507  0.2655915 -0.48397932  0.9744166  1.0000000  0.9760463
lr6 0.25918676  0.2755877 -0.47873960  0.9782465  0.9760463  1.0000000
```

```
> chemo.compare
    Ordinary Least Squares       Curds and Whey
y1               0.5517121            0.5397667
y2               1.0996896            0.7647241
y3               0.2463759            0.2289689
y4               0.1188889            0.1046603
y5               0.2705596            0.1796938
y6               0.1813059            0.1388877
Ave              0.4114220            0.3261169
> chemo.compare2
    Difference   Percentage
y1  0.01194537     2.165145
y2  0.33496542    30.459998
y3  0.01740700     7.065220
y4  0.01422853    11.967928
```

```
y5  0.09086581    33.584397

y6  0.04241816    23.395909

Ave 0.08530505    20.734198
```

```
> chemo.shrinkage

[1] 0.9934 0.9735 0.8644 0.1621 0.1293 0.0000
```

To visualize the results, I graphed the values of the prediction error along with a line to indicate equal errors.



As every value is below the line, it can be seen that each predictor and the average all saw improvement. One can also see that y_2 and y_5 saw significant improvement. y_3, as might be expected due to its lower correlation did not see quite as high of an improvement, and y_1 did not improve very well either. Overall, though, there was an average improvement of almost 21%. One can also see the shrinkage factors (8) that were applied to the data listed below under chemo.shrinkage.

4.2 Automobile Data

The next data set is one that concerns different attributes of cars. For the predictor variables, there are twenty-one variables, such as make of car, number of doors, type of fuel, engine size, horsepower, cylinders, and a few others. These predictors include factors as well as numeric values.

The response variables are:

- y_1: Miles Per Gallon: City
- y_2: Miles Per Gallon: Highway
- y_3: Price

In this case, it seems intuitive that the two MPG responses will be correlated, but One may want to check the overall correlation between all three variables.

```
           MPG.City   MPG.High      price
MPG.City  1.0000000  0.9723499 -0.7026849
MPG.High  0.9723499  1.0000000 -0.7155898
price    -0.7026849 -0.7155898  1.0000000
```

So, one can see that the two MPG variables are highly correlated with each other, and are highly negatively correlated with price. As bigger cars tend to be use more fuel and are often luxury cars, they would be more expensive.

Running the analysis on this data as I did before yields the following results:

```
> auto.compare

            Ordinary Least Squares       Curds and Whey

City MPG                 0.1571167            0.1557632

Highway MPG              0.1694486            0.1667852

Price                    0.1304048            0.1289536

Ave                      0.1523233            0.1505007

> auto.compare2

             Difference   Percentage

City MPG    0.001353513    0.8614701

Highway MPG 0.002663353    1.5717764

Price       0.001451179    1.1128264

Ave         0.001822681    1.1965871

> auto.shrinkage

[1] 0.9921 0.9154 0.6812
```

In this case, there is not a large improvement over the ordinary least squares regression, but if one notices the shrinkage factors, it can be seen that there is not a lot of shrinkage taking place. Still, there is a small improvement again across all of the response variables. It still can be noted that the prediction error for this data is small even under ordinary least squares regression, so even a small amount of improvement is worthwhile.

I ran a second analysis with a smaller set of predictors to see how much improvement can be achieved. A few of the predictors seem likely to be directly tied to miles per gallon, so they should be included. The new set of predictors then includes make, gas type, engine size, weight, cylinders, fuel system, bore size, stroke, compression, power, and rpm.

```
> auto.compare

            Ordinary Least Squares       Curds and Whey

City MPG                 0.1562651            0.1558749

Highway MPG              0.1578281            0.1573520

Price                    0.1555377            0.1551778

Ave                      0.1565437            0.1561349

> auto.compare2

              Difference   Percentage

City MPG    0.0003901853    0.2496945

Highway MPG 0.0004761433    0.3016847

Price       0.0003599060    0.2313946

Ave         0.0004087449    0.2611060

> auto.shrinkage
```

```
[1] 0.9956 0.9326 0.7544
```

Again, as may be the case with these data only, we do not see an incredible amount of improvement over ordinary least squares regression, but we again do see improvement in for every predictor. It does seem to lend evidence to the idea that the algorithm returns more accurate results with a larger number of predictors. This of course can be at least partially explained by the knowledge that more predictors always leads to a more accurate prediction, though those extra predictors may only be modeling noise in the data.

4.3 Teen Crime Data

The final data set deals with violent crimes committed by teens in all 50 states and Washington D.C. The data was collected between the years of 1985 and 1993. It contains many possible predictor and response variables, so I will list them all as x_i values. All the variables are as follows:

- x_1: Percentage of Seniors that graduate from High School
- x_2: Standardized transformation of Scoring Method used in Survey
- x_3: Number of 1 to 14 year-olds in 1985
- x_4: Number of 1 to 14 year-olds that died in 1985.
- x_5, x_6: x_3 and x_4 repeated but for 1991.
- x_7, x_8: Percentage of Kids living in Poverty in 1985, 1991 respectively.
- x_9 to x_{19}: Percentage of Kids living in Single Parent Families from 1983 through 1993 respectively.

- x_{20} to x_{25}: The Median income in 1987 through 1992 respectively.

- x_{26} to x_{33}: Juvenile Violent Crimes per 100,000 people in 1985 to 1992.

A lot of research has been done on this data, and in a lot of circumstances, the Juvenile Violent Crime rates have been the natural response variables. With this data, I go through the data twice, once with all of the predictor variables to view the change in predictive accuracy, and then once with a more specialized set of predictors.

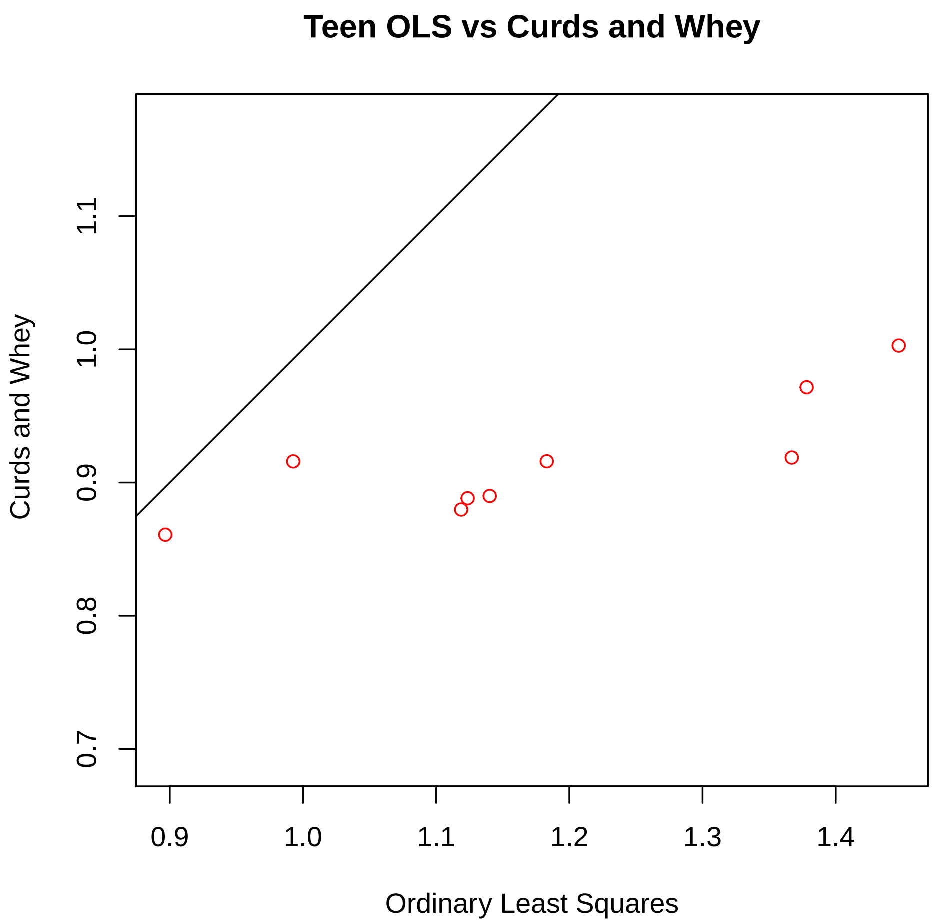
Firstly, I run the Curds and Whey procedure trying to predict the Juvenile Crime Rate using all of the other variables available. This yields:

```
> teen.compare

     Ordinary Least Squares       Curds and Whey

1985              1.3669879            0.9187600

1986              1.3781500            0.9715483

1987              1.4473000            1.0028335

1988              1.1187575            0.8797412

1989              1.1236404            0.8882148

1990              0.9926789            0.9158647

1991              0.8966329            0.8608297

1992              1.1402116            0.8899276

Ave               1.1830449            0.9159650

> teen.compare2

     Difference   Percentage

1985 0.44822787    32.789455
```

```
1986 0.40660178    29.503448
1987 0.44446650    30.710046
1988 0.23901636    21.364447
1989 0.23542552    20.952035
1990 0.07681423     7.738074
1991 0.03580321     3.993073
1992 0.25028395    21.950659
Ave  0.26707993    22.575637
```

```
> teen.shrinkage
[1] 0.7493 0.6070 0.5509 0.0855 0.0000 0.0000 0.0000 0.0000
```

Thus again, there is improvement in every category, and some substantial improvement on some years. It should be noted that 1990 and 1991 saw the least improvement, which improvement is almost an order of magnitude less than the most improved. The following graph shows, as it did with the chemometrics example, that each Curds and Whey prediction has a smaller error than the OLS predictions.

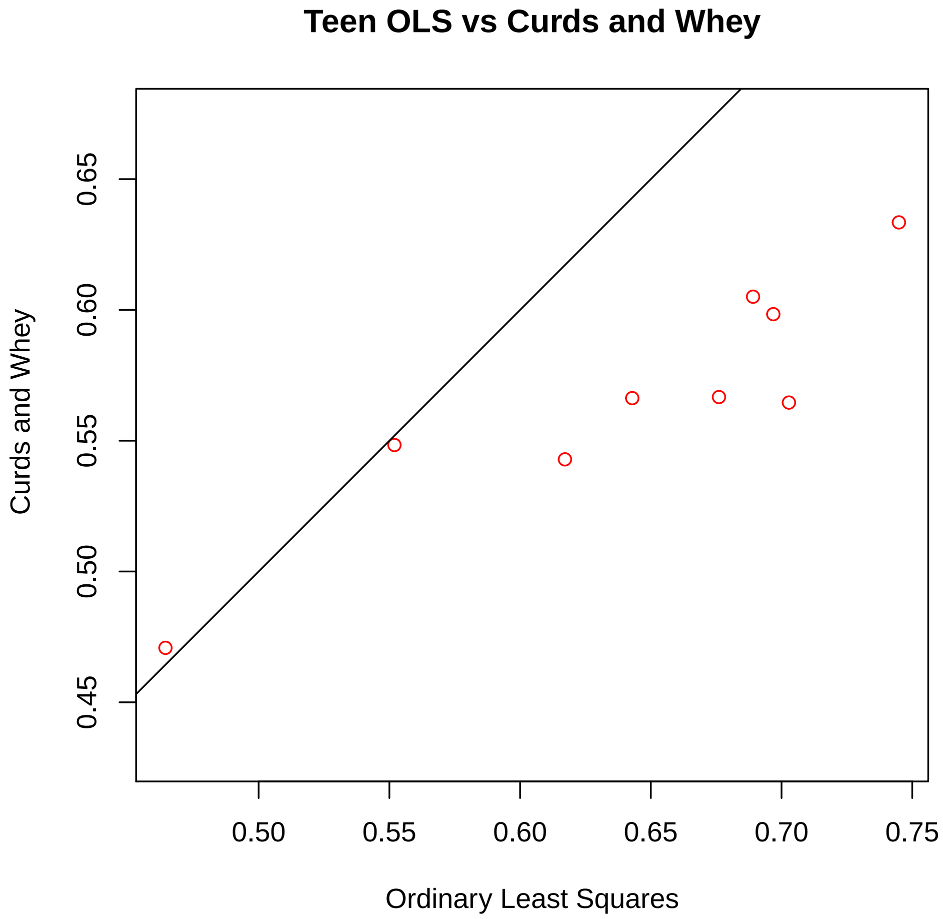


The second set to be tested includes only the predictors involving the percentage of single parent households and median income. This returns the results:

Second Check:

```
> teen.compare

     Ordinary Least Squares       Curds and Whey

1985              0.6891016            0.6050609

1986              0.6968589            0.5983855

1987              0.7448917            0.6334626
```

```
1988               0.6760759            0.5666972

1989               0.6171540            0.5428756

1990               0.5519348            0.5483529

1991               0.4643568            0.4708149

1992               0.7028250            0.5645866

Ave                0.6428999            0.5662795

> teen.compare2

       Difference   Percentage

1985  0.084040759   12.1956987

1986  0.098473403   14.1310394

1987  0.111429116   14.9591020

1988  0.109378696   16.1784633

1989  0.074278426   12.0356387

1990  0.003581946    0.6489799

1991 -0.006458074   -1.3907567

1992  0.138238388   19.6689632

Ave   0.076620332   11.9179266

> teen.shrinkage

[1] 0.8202 0.5936 0.5343 0.0000 0.0000 0.0000 0.0000 0.0000
```



Again, it is seen that 1990 and 1991 do not receive the same improvement from Curds and Whey as do the other years. In fact, in this case, the prediction for 1991 is slightly worse than it was before. To try to understand what might be causing this, I next looked at the correlation between the response variables:

```
          JVCAR85   JVCAR86   JVCAR87   JVCAR88   JVCAR89   JVCAR90   JVCAR91
JVCAR85 1.0000000 0.9655667 0.9511003 0.9107886 0.8981195 0.9049542 0.9126092
JVCAR86 0.9655667 1.0000000 0.9763015 0.9413312 0.9417452 0.9506291 0.9490129
JVCAR87 0.9511003 0.9763015 1.0000000 0.9572283 0.9524495 0.9390007 0.9284937
```

```
JVCAR88 0.9107886 0.9413312 0.9572283 1.0000000 0.9581374 0.9472979 0.9355166
JVCAR89 0.8981195 0.9417452 0.9524495 0.9581374 1.0000000 0.9425560 0.9390979
JVCAR90 0.9049542 0.9506291 0.9390007 0.9472979 0.9425560 1.0000000 0.9762858
JVCAR91 0.9126092 0.9490129 0.9284937 0.9355166 0.9390979 0.9762858 1.0000000
JVCAR92 0.9102105 0.9365655 0.9340547 0.9498841 0.9437584 0.9426121 0.9680579
          JVCAR92
JVCAR85 0.9102105
JVCAR86 0.9365655
JVCAR87 0.9340547
JVCAR88 0.9498841
JVCAR89 0.9437584
JVCAR90 0.9426121
JVCAR91 0.9680579
JVCAR92 1.0000000
```

Some things that can be noticed is that 1990 and 1991 are more highly correlated with each other than any other year.

In the end, the focus largely is on how the average improvement was fairly substantial (over 10%), and conclude that overall, Curds and Whey did provide a good increase in prediction accuracy.

5 Further Work

Being able to improve upon linear regression with multiple response variables opens up possibilities for future work in other related areas in statistics. One possible field is classification with linear or near-linear classifiers. As most linear classifiers perform at a similar level, any amount of improvement may be a significant contribution to the subject. With regards to what has been found thus far, more work can be done to increase the utility of this procedure to include such things such as hypothesis tests.

Acknowledgements

All analyses were performed using the R statistical software R version 3.0.0 (2013-04-03).

Research adviser: Dr. Richard Cutler, Department of Mathematics and Statistics, Utah State University.

References

[1] Leo Breiman and Jerome H Friedman. Predicting multivariate responses in multiple linear regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(1):3–54, 1997.

[2] John B Copas. Regression, prediction and shrinkage. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 311–354, 1983.

[3] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[4] William James and Charles Stein. Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 361–379, 1961.

[5] William F Massy. Principal components regression in exploratory statistical research. *Journal of the American Statistical Association*, 60(309):234–256, 1965.

[6] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2013. ISBN 3-900051-07-0.

[7] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2013. ISBN 3-900051-07-0.

[8] Bert Skagerberg, John F MacGregor, and Costas Kiparissides. Multivariate data analysis applied to low-density polyethylene reactors. *Chemometrics and intelligent laboratory systems*, 14(1):341–356, 1992.

[9] Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 111–147, 1974.

[10] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S.* Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

[11] Herman Wold. Soft modelling by latent variables: the non-linear iterative partial least squares (nipals) approach. *Perspectives in Probability and Statistics, In Honor of MS Bartlett*, pages 117–144, 1975.