Generalized Regression

Nonlinear Algorithms Including Ridge, Lasso and Elastic Net

JONATHAN KINLAY

Global Equities

UPDATED: MAY 23, 2011

Ordinary Least Squares Regression

OLS Model Framework

The problem framework is one in which we have observations $\{y_1, y_2, \ldots, y_n\}$ from a random variable Y which we are interested in predicting, based on the observed values $\{\{x_{11}, x_{21}, \ldots, x_{n,1}\}, \{x_{12}, x_{22}, \ldots, x_{n,2}\}, \ldots, \{x_{1\,p}, x_{2\,p}, \ldots, x_{n,p}\}$, from p independent explanatory random variates $\{X_1, X_2, \ldots, X_p\}$. Our initial assumption is that the $X_i$are independent, but relax that condition and consider the case of collinear (i.e. correlated) explanatory variables in the general case.

We frame the problem in matrix form as:

$\mathrm{Y} = \mathrm{X}\beta + \epsilon$,

where ϵ is assumed $\sim \mathrm{No}[0, \sigma]$

Y is the $[\mathrm{n} \times 1]$ matrix of observations $\mathrm{y_i}$

X is the $[\mathrm{n} \times \mathrm{p}]$ matrix of observations $\mathrm{x_{ij}}$

and β is an unknown $[\mathrm{p} \times 1]$ vector of coefficients to be estimated

The aim is to find a solution which minimizes the mean square error:

$$\frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \epsilon_{\mathrm{i}}^2 = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{y_i} - \mathrm{X_i}\, \beta)^2 = \frac{1}{\mathrm{n}} (\mathrm{Y} - \beta \mathrm{X})^{\mathrm{T}}.(\mathrm{Y} - \beta \mathrm{X})$$

which has the well known solution :

$$\hat{\beta} = \left(\mathrm{X}^{\mathrm{T}}\, \mathrm{X}\right)^{-1} \left(\mathrm{X}^{\mathrm{T}}\, \mathrm{Y}\right)$$

In *Mathematica* there are several ways to accomplish this, for example:

```
OLSRegression[Y_, X_]:=Module[
{XT=Transpose[X]},
Z=LinearSolve[XT.X,XT.Y]
]
```

The Problem of Collinearity

Difficulties begin to emerge when the assumption of independence no longer applies. While it is still possible to conclude whether the system of explanatory variables X has explanatory power as a whole, the correlation between the variables means that it no longer possible clearly distinguish the significance of individual variables. Estimates of the beta coefficients may be biased, understating the importance of some of the variables and overstating the importance of others.

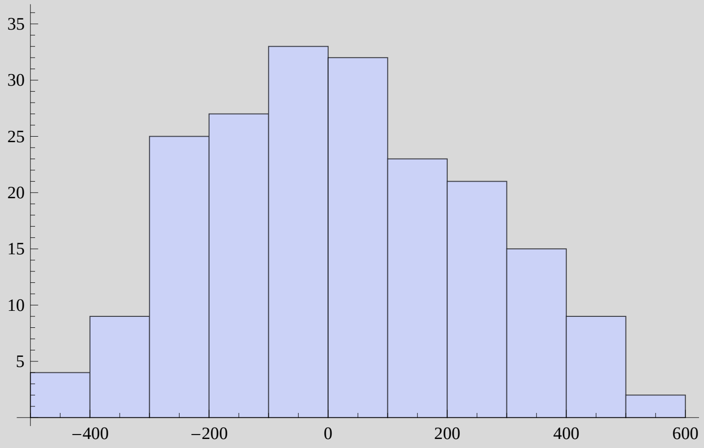
Example of Collinear Variables

This example is based on the original paper on lasso regression (Tibshirani, 1996) and a subsequent paper by Zou and Hastie (2004). We have 220 observations on eight predictors. We use the first 20 observations for model estimation and the remaining 200 for out-of-sample testing. We fix the parameters β and σ and set the correlation between variables X_1, and X_2 to be $0.5^{|i-j|}$ as follows:

```
nObs = 200; σ = 3; β = {3, 1.5, 0, 0, 2, 0, 0, 0};
Σ = Table[0.5^Abs[i-j], {i, 1, 8}, {j, 1, 8}] // MatrixForm
```

```
    1.         0.5       0.25     0.125   0.0625  0.03125  0.015625  0.0078125
    0.5        1.        0.5      0.25    0.125   0.0625   0.03125   0.015625
    0.25       0.5       1.       0.5     0.25    0.125    0.0625    0.03125
    0.125      0.25      0.5      1.      0.5     0.25     0.125     0.0625
    0.0625     0.125     0.25     0.5     1.      0.5      0.25      0.125
    0.03125    0.0625    0.125    0.25    0.5     1.       0.5       0.25
    0.015625   0.03125   0.0625   0.125   0.25    0.5      1.        0.5
    0.0078125  0.015625  0.03125  0.0625  0.125   0.25     0.5       1.
```

```
ϵ = RandomVariate[d = NormalDistribution[0, σ], nObs];
X = RandomReal[{-100, 100}, {nObs, 8}];
Y = X.β + ϵ;
Histogram[Y]
```



In the context of collinear explanatory variables, our standard OLS estimates will typically be biased. In this examples, notice how the relatively large correlations between variables 1-4 induces upward bias in the estimates of the parameters β_3 and β_4, (and downward bias in the estimates of the parameters β_6 and β_7.

```
β̂0 = OLSRegression[Take[Y, 20], Take[X, 20]]
```

```
{3.00439, 1.50906, 0.0145471, 0.0383589, 1.9809, -0.0106091, -0.00378074, -0.00259506}
```

Generalized Regression

Penalized OLS Objective Function

We attempt to deal with the problems of correlated explanatory variables by introducing a penalty component to the OLS objective function. The idea is to penalize the regression for using too many correlated explanatory variables, as follows:

$$\mathrm{WLS} = \frac{1}{n}\sum_{i=1}^{n}(y_i - X_i\,\boldsymbol{\beta})^2 + \lambda P_\alpha,$$

with

$$P_\alpha = \sum_{j=1}^{p}\left[\frac{1}{2}(1-\alpha)\,\boldsymbol{\beta}_j^2 + \alpha\left|\boldsymbol{\beta}_j\right|\right.$$

In[819]:=

```
WLS[Y_, X_, λ_, α_, β_] := Module[
  {n = Length[Y], m = Length[β], W, P},
  Z = Y - X.β;
  W = 1/n Z.Z;
  P = Sum[(1/2 (1 - α) β[[i]]^2 + α Abs[β[[i]]]), {i, 1, m}];
  W + λ P
 ]
```

Nonstandard Regression Types

In the above framework:

- α=0 ridge regression
- α=1 lasso regression
- $\alpha \in (0, 1)$ elastic net regression

We use the *Mathematica* NMinimze function to find a global minimum of the WLS objective function, within a specified range for α, as follows:

```
GeneralRegression[Y_, X_, λ_, αRange_] := Module[
  {nIndep = Last[Dimensions[X]], b},
  b = Table[Unique[b], {1 + nIndep}];
  Reg = NMinimize[
    {WLS[Y, X, λ, b[[1]], Drop[b, 1]], αRange[[1]] ≤ b[[1]] ≤ αRange[[2]]}, b];
  coeff = b /. Last[Reg];
  {Reg[[1]], coeff[[1]], Drop[coeff, 1]}
 ]
```

NMinimze employs a variety of algorithms for constrained global optimization, including Nelder-Mead, Differential Evolution, Simulated Annealing and Random Search. Details can be found here.

The GeneralRegression function returns the minimized WLS value, the optimal α parameter (within the constraints set by the user), and the optimal weight vector β.

Reproducing OLS Regression

Here we simply set λ=0 and obtain the same estimates $\hat{\beta}_0$ as before (note that the optimal α value is negative):

```
GeneralRegression[Take[Y, 20], Take[X, 20], 0, {-20, 20}]
```

```
{4.52034, -5.47031, {3.00439, 1.50906, 0.0145471,
  0.0383589, 1.9809, -0.0106091, -0.00378074, -0.00259506}}
```

Ridge Regression

Here we set λ=1 and constrain $\alpha = 0$. Note that the value of the penalized WLS is significantly larger than in the OLS case, due to the penalty term P_α

```
GeneralRegression[Take[Y, 20], Take[X, 20], 1, {0, 0}]
```

```
{12.134, 0., {3.00387, 1.50879, 0.0145963,
  0.0385133, 1.98083, -0.0105296, -0.00378282, -0.00267926}}
```

Lasso Regression

Here we set λ=1 and constrain $\alpha = 1$. Note that the value of the penalized WLS is lower than Ridge regression:

```
GeneralRegression[Take[Y, 20], Take[X, 20], 1, {1, 1}]
```

```
{11.0841, 1., {3.00424, 1.50904, 0.0143872,
  0.0381963, 1.98082, -0.010514, -0.00360641, -0.00248904}}
```

Elastic Net Regression

Here we set λ=1 and constrain α to lie in the range (0, 1). In this case the optimal value of $\alpha = 1$, the same as for lasso regression:

```
GeneralRegression[Take[Y, 20], Take[X, 20], 1, {0, 1}]
```

```
{11.0841, 1., {3.00424, 1.50904, 0.0143872,
  0.0381963, 1.98082, -0.010514, -0.00360641, -0.00248904}}
```

Generalized Regression

Here we set λ=1 and constrain α to lie in a wider subset of $\mathbb{R}$, for example (-5, 5):

```
GeneralRegression[Take[Y, 20], Take[X, 20], 1, {-20, 20}]
```

```
{-9.42129, 20., {3.01909, 1.51991, -8.01608 × 10^-9,
  -1.28294 × 10^-8, 2.01397, 0.0370202, -6.15894 × 10^-9, -8.15124 × 10^-9}}
```

Empirical Test

We conduct an empirical test of the accuracy of the various regression methods, by simulating 100 data sets consisting of 220 observations (20 in-sample, 200 out-of-sample), with regressors and parameters as before. For each of the regression methods we calculate the MSE from the out-of-sample data, using coefficients estimated using the in-sample data.

First, create a function to calculate the Mean Square Error:

```
MSE[Y_, X_, b_] := Module[
   {nObs = Length[Y], Z = Y − b.Transpose[X]},
   Z.Z/nObs
  ]
```

Now create a test program to run multiple samples:

In[837]:=

```
i = 1; nEpochs = 100; MSS0 = Table[∅, {nEpochs}]; MSS1 = Table[∅, {nEpochs}];
MSS2 = Table[∅, {nEpochs}]; MSS3 = Table[∅, {nEpochs}]; MSS4 = Table[∅, {nEpochs}];
While [i ⩽ nEpochs,
  ϵ = RandomVariate[𝒹 = NormalDistribution[0, σ], nObs];
  X = RandomReal[{-100, 100}, {nObs, 8}];
  Y = X.β + ϵ;
  Parallelize[
   b0 = Last[GeneralRegression[Take[Y, 20], Take[X, 20], 0, {-20, 20}]];
   b1 = Last[GeneralRegression[Take[Y, 20], Take[X, 20], 1, {0, 0}]];
   b2 = Last[GeneralRegression[Take[Y, 20], Take[X, 20], 1, {1, 1}]];
   b3 = Last[GeneralRegression[Take[Y, 20], Take[X, 20], 1, {0, 1}]];
   b4 = Last[GeneralRegression[Take[Y, 20], Take[X, 20], 1, {-20, 20}]];
   MSS0[[i]] = MSE[Drop[Y, 20], Drop[X, 20], b0];
   MSS1[[i]] = MSE[Drop[Y, 20], Drop[X, 20], b1];
   MSS2[[i]] = MSE[Drop[Y, 20], Drop[X, 20], b2];
   MSS3[[i]] = MSE[Drop[Y, 20], Drop[X, 20], b3];
   MSS4[[i]] = MSE[Drop[Y, 20], Drop[X, 20], b4]]; i++];
MSS = {MSS0, MSS1, MSS2, MSS3, MSS4};
```

The average out-of-sample MSE for each regression method is shown in the cell below. The average MSE for the Generalized regression is significantly lower than for other regression techniques.

In[842]:=

```
NumberForm[{Mean[MSS0], Mean[MSS1], Mean[MSS2], Mean[MSS3], Mean[MSS4]}, {4, 2}]
```

Out[842]//NumberForm=

```
{14.69, 14.66, 14.51, 14.51, 13.23}
```

The lower MSE is achieved by lower estimated values for the zero β coefficients:

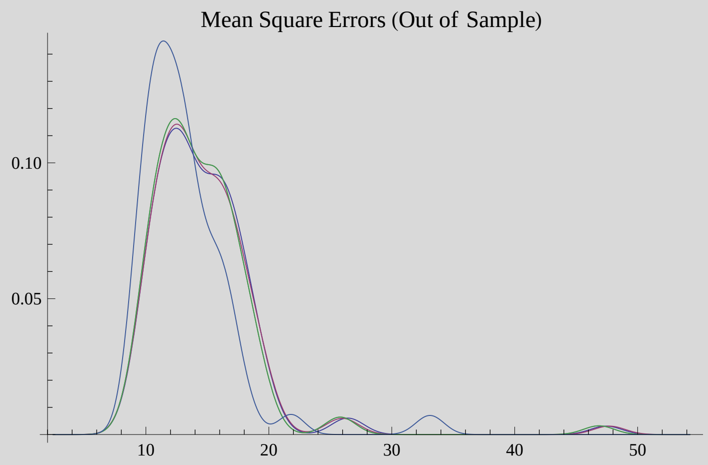
```
In[849]:= {b0, b1, b2, b3, b4} // MatrixForm
```

```
Out[849]//MatrixForm=
 3.01483  1.51507   -0.00115002        0.00488936  2.01666   0.00341592        -0.0162198  0.01785
 3.01352  1.51359   -0.000842353       0.00413787  2.01553   0.00262603        -0.0161391  0.0176
 3.01417  1.51359   -0.000769123       0.00413357  2.01575   0.00240096        -0.0160634  0.0174
 3.01417  1.51359   -0.000769123       0.00413357  2.01575   0.00240096        -0.0160634  0.0174
 3.02687  1.51527   -5.25723 × 10^-10  0.00477497  2.02069   1.28321 × 10^-7   -0.0145281  0.0146
```

A comparison of the histograms of the MSE’s for each of the regression methods underscores the superiority of the Generalized Regression technique:

```
In[844]:= SmoothHistogram[MSS, PlotLabel → "Mean Square Errors (Out of Sample)"]
```

Out[844]=



References

Tong Zhang, Multi - Stage Convex relaxation for Feature Selection,
Neural Information Processing Systems - NIPS, pp.1929 - 1936, 2010

Hui Zou and Trevor Hastie, Regularization and Variable Selection via the Elastic Net,
J.R. Statist. Soc. B (2005) 67, Part 2, pp301 - 320