Lasso and Elastic Net

On this page…

What Are Lasso and Elastic Net?
Lasso Regularization
Lasso and Elastic Net with Cross Validation
Wide Data via Lasso and Parallel Computing
Lasso and Elastic Net Details
References

What Are Lasso and Elastic Net?

Lasso is a regularization technique. Use `lasso` to:

- Reduce the number of predictors in a regression model.
- Identify important predictors.
- Select among redundant predictors.
- Produce shrinkage estimates with potentially lower predictive errors than ordinary least squares.

Elastic net is a related technique. Use elastic net when you have several highly correlated variables. `lasso` provides elastic net regularization when you set the `Alpha` name-value pair to a number strictly between `0` and `1`.

See Lasso and Elastic Net Details.

For lasso regularization of regression ensembles, see `regularize`.

Lasso Regularization

To see how `lasso` identifies and discards unnecessary predictors:

1. Generate `200` samples of five-dimensional artificial data `X` from exponential distributions with various means:

```
rng(3,'twister') % for reproducibility
X = zeros(200,5);
for ii = 1:5
    X(:,ii) = exprnd(ii,200,1);
end
```

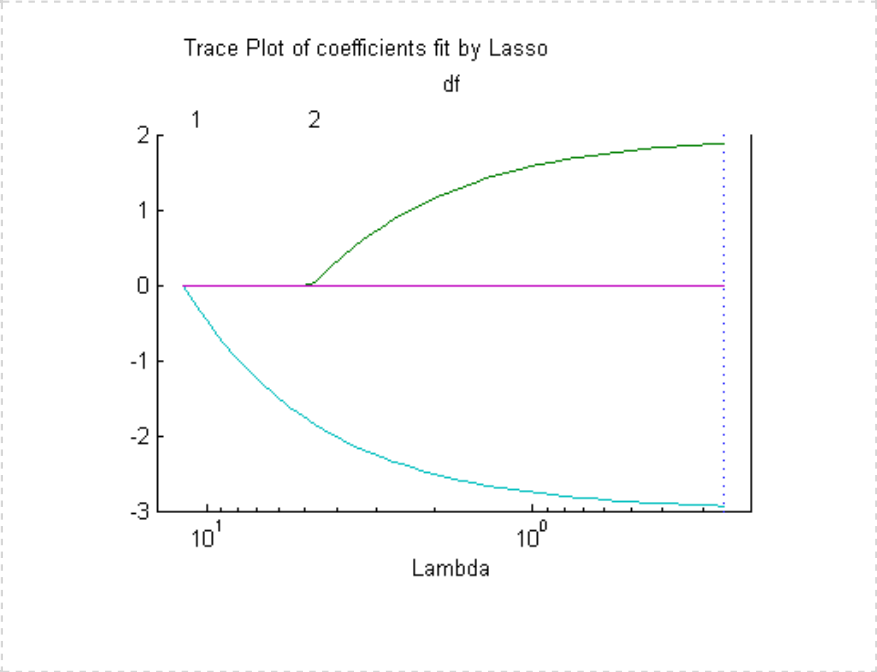
2. Generate response data `Y = X*r + eps` where `r` has just two nonzero components, and the noise `eps` is normal with standard deviation `0.1`:

```
r = [0;2;0;-3;0];
Y = X*r + randn(200,1)*.1;
```

3. Fit a cross-validated sequence of models with `lasso`, and plot the result:

```
[b fitinfo] = lasso(X,Y,'CV',10);
```

```
lassoPlot(b,fitinfo,'PlotType','Lambda','XScale','log');
```



The plot shows the nonzero coefficients in the regression for various values of the `Lambda` regularization parameter. Larger values of `Lambda` appear on the left side of the graph, meaning more regularization, resulting in fewer nonzero regression coefficients.

The dashed vertical lines represent the `Lambda` value with minimal mean squared error (on the right), and the `Lambda` value with minimal mean squared error plus one standard deviation. This latter value is a recommended setting for `Lambda`. These lines appear only when you perform cross validation. Cross validate by setting the `'CV'` name-value pair. This example uses 10-fold cross validation.

The upper part of the plot shows the degrees of freedom (df), meaning the number of nonzero coefficients in the regression, as a function of Lambda. On the left, the large value of Lambda causes all but one coefficient to be 0. On the right all five coefficients are nonzero, though the plot shows only two clearly. The other three coefficients are so small that you cannot visually distinguish them from 0.

For small values of `Lambda` (toward the right in the plot), the coefficient values are close to the least-squares estimate. See step 5.

4. Find the `Lambda` value of the minimal cross-validated mean squared error plus one standard deviation. Examine the MSE and coefficients of the fit at that `Lambda`:

```
lam = fitinfo.Index1SE;
fitinfo.MSE(lam)

ans =
    0.1398

b(:,lam)

ans =
         0
    1.8855
         0
```

```
   -2.9367
         0
```

`lasso` did a good job finding the coefficient vector `r`.

5. For comparison, find the least-squares estimate of `r`:

```
rhat = X\Y

rhat =

   -0.0038
    1.9952
    0.0014
   -2.9993
    0.0031
```

The estimate `b(:,lam)` has slightly more mean squared error than the mean squared error of `rhat`:

```
res = X*rhat - Y; % calculate residuals
MSEmin = res'*res/200 % b(:,lam) value is 0.1398

MSEmin =
    0.0088
```

But `b(:,lam)` has only two nonzero components, and therefore can provide better predictive estimates on new data.

Lasso and Elastic Net with Cross Validation

Consider predicting the mileage (`MPG`) of a car based on its weight, displacement, horsepower, and acceleration. The `carbig` data contains these measurements. The data seem likely to be correlated, making elastic net an attractive choice.

1. Load the data:

```
load carbig
```

2. Extract the continuous (noncategorical) predictors (`lasso` does not handle categorical predictors):

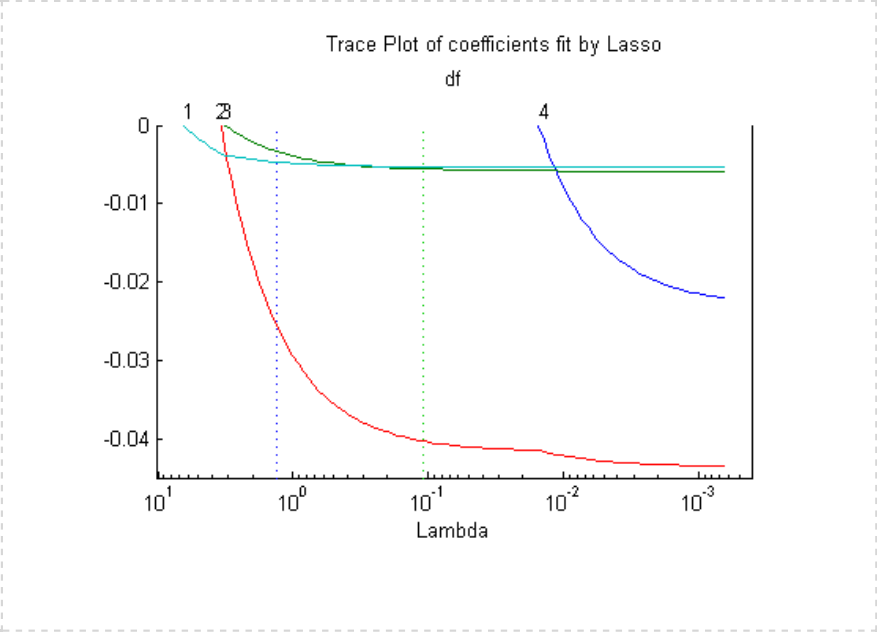
```
X = [Acceleration Displacement Horsepower Weight];
```

3. Perform a lasso fit with 10-fold cross validation:

```
[b fitinfo] = lasso(X,MPG,'CV',10);
```

4. Plot the result:

```
lassoPlot(b,fitinfo,'PlotType','Lambda','XScale','log');
```



5. Calculate the correlation of the predictors:

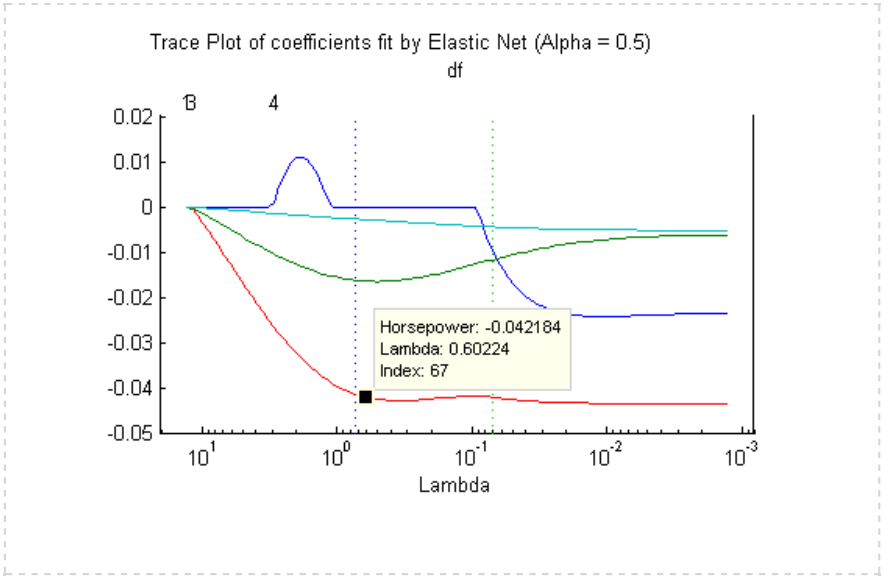
```
% Eliminate NaNs so corr runs
nonan = ~any(isnan([X MPG]),2);
Xnonan = X(nonan,:);
MPGnonan = MPG(nonan,:);
corr(Xnonan)

ans =
    1.0000   -0.5438   -0.6892   -0.4168
   -0.5438    1.0000    0.8973    0.9330
   -0.6892    0.8973    1.0000    0.8645
   -0.4168    0.9330    0.8645    1.0000
```

6. Because some predictors are highly correlated, perform elastic net fitting. Use `Alpha = 0.5`:

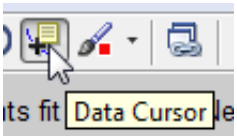
```
[ba fitinfoa] = lasso(X,MPG,'CV',10,'Alpha',.5);
```

7. Plot the result. Name each predictor so you can tell which curve is which:

```
pnames = {'Acceleration','Displacement',...
    'Horsepower','Weight'};
lassoPlot(ba,fitinfoa,'PlotType','Lambda',...
    'XScale','log','PredictorNames',pnames);
```



When you activate the data cursor



and click the plot, you see the name of the predictor, the coefficient, the value of `Lambda`, and the index of that point, meaning the column in `b` associated with that fit.

Here, the elastic net and lasso results are not very similar. Also, the elastic net plot reflects a notable qualitative property of the elastic net technique. The elastic net retains three nonzero coefficients as `Lambda` increases (toward the left of the plot), and these three coefficients reach `0` at about the same `Lambda` value. In contrast, the lasso plot shows two of the three coefficients becoming `0` at the same value of `Lambda`, while another coefficient remains nonzero for higher values of `Lambda`.

This behavior exemplifies a general pattern. In general, elastic net tends to retain or drop groups of highly correlated predictors as `Lambda` increases. In contrast, lasso tends to drop smaller groups, or even individual predictors.

Wide Data via Lasso and Parallel Computing

Lasso and elastic net are especially well suited to *wide* data, meaning data with more predictors than observations. Obviously, there are redundant predictors in this type of data. Use `lasso` along with cross validation to identify important predictors.

Cross validation can be slow. If you have a Parallel Computing Toolbox license, speed the computation using parallel computing.

1. Load the `spectra` data:

```
load spectra
Description

Description =
```

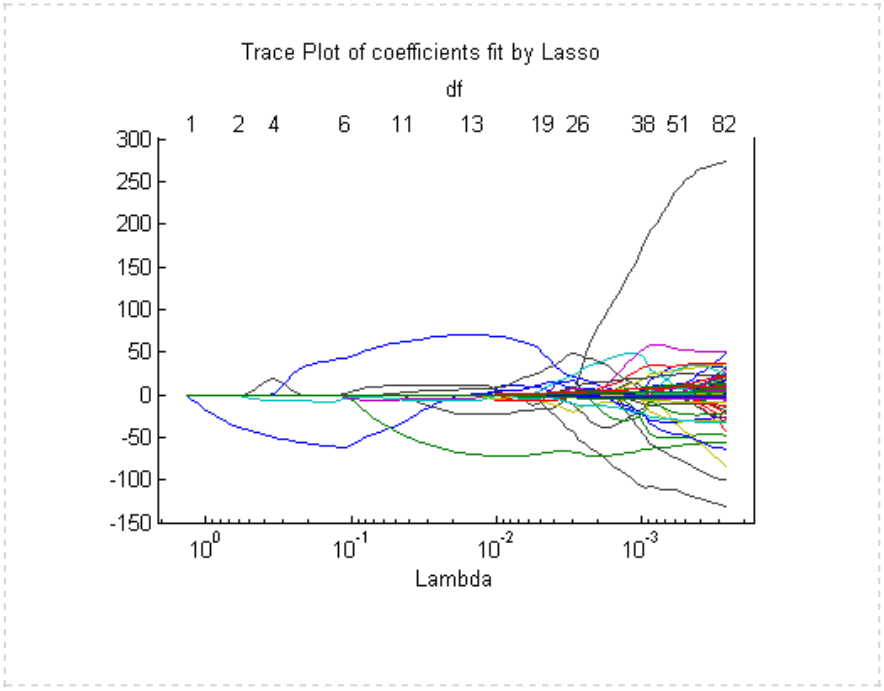
```
== Spectral and octane data of gasoline ==

NIR spectra and octane numbers of 60 gasoline samples

NIR:     NIR spectra, measured in 2 nm intervals from 900 nm to 1700 nm
octane:  octane numbers
spectra: a dataset array containing variables for NIR and octane

Reference:
Kalivas, John H., "Two Data Sets of Near Infrared Spectra," Chemometrics
and Intelligent Laboratory Systems, v.37 (1997) pp.255-259
```

2. Compute the default `lasso` fit:

```
[b fitinfo] = lasso(NIR,octane);
```

3. Plot the number of predictors in the fitted lasso regularization as a function of `Lambda`, using a logarithmic *x*-axis:

```
lassoPlot(b,fitinfo,'PlotType','Lambda','XScale','log');
```

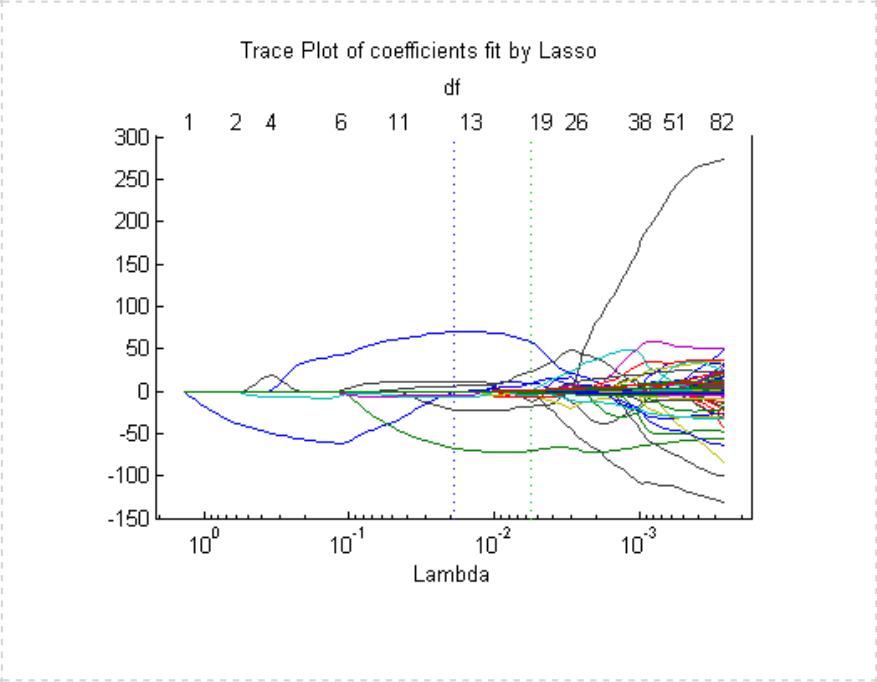


4. It is difficult to tell which value of `Lambda` is appropriate. To determine a good value, try fitting with cross validation:

```
tic
[b fitinfo] = lasso(NIR,octane,'CV',10);
% A time-consuming operation
toc

Elapsed time is 226.876926 seconds.
```

5. Plot the result:

```
lassoPlot(b,fitinfo,'PlotType','Lambda','XScale','log');
```



You can see the suggested value of `Lambda` is over `1e-2`, and the `Lambda` with minimal MSE is under `1e-2`. These values are in the `fitinfo` structure:

```
fitinfo.LambdaMinMSE
ans =
    0.0057

fitinfo.Lambda1SE
ans =
    0.0190
```

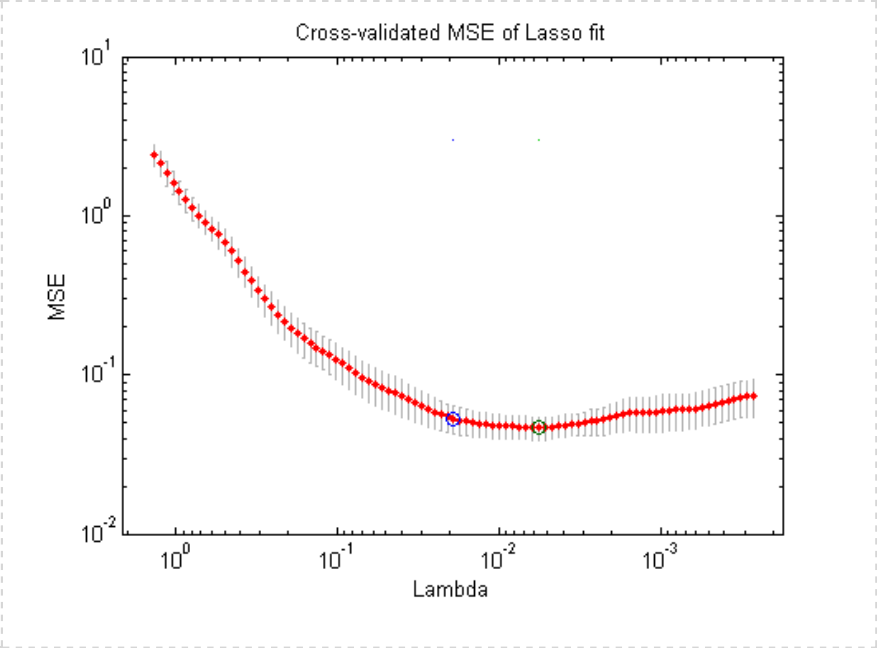
6. Examine the quality of the fit for the suggested value of `Lambda`:

```
lambdaindex = fitinfo.Index1SE;
fitinfo.MSE(lambdaindex)

ans =
    0.0532

fitinfo.DF(lambdaindex)

ans =
    11
```

The fit uses just 11 of the 401 predictors, and achieves a cross-validated MSE of `0.0532`.

7. Examine the plot of cross-validated MSE:

```
lassoPlot(b,fitinfo,'PlotType','CV');
% Use a log scale for MSE to see small MSE values better
set(gca,'YScale','log');
```



As `Lambda` increases (toward the left), MSE increases rapidly. The coefficients are reduced too much and they do not adequately fit the responses.

As `Lambda` decreases, the models are larger (have more nonzero coefficients). The increasing MSE suggests that the models are overfitted.

The default set of `Lambda` values does not include values small enough to include all predictors. In this case, there does not appear to be a reason to look at smaller values. However, if you want smaller values than the default, use the `LambdaRatio` parameter, or supply a sequence of `Lambda` values using the `Lambda` parameter. For details, see the `lasso` reference page.

8. To compute the cross-validated lasso estimate faster, use parallel computing (available with a Parallel Computing Toolbox license):

```
matlabpool open
Starting matlabpool using the 'local' configuration ...
    connected to 4 labs.

opts = statset('UseParallel','always');
tic;
[b fitinfo] = lasso(NIR,octane,'CV',10,'Options',opts);
toc

Elapsed time is 107.539719 seconds.
```

Computing in parallel is more than twice as fast on this problem using a quad-core processor.

Lasso and Elastic Net Details

Overview of Lasso and Elastic Net

Lasso is a regularization technique for performing linear regression. Lasso includes a penalty term that constrains the size of the estimated coefficients. Therefore, it resembles ridge regression. Lasso is a *shrinkage estimator*: it generates coefficient estimates that are biased to be small. Nevertheless, a lasso estimator can have smaller mean squared error than an ordinary least-squares estimator when you apply it to new data.

Unlike ridge regression, as the penalty term increases, lasso sets more coefficients to zero. This means that the lasso estimator is a smaller model, with fewer predictors. As such, lasso is an alternative to stepwise regression and other model selection and dimensionality reduction techniques.

Elastic net is a related technique. Elastic net is a hybrid of ridge regression and lasso regularization. Like lasso, elastic net can generate reduced models by generating zero-valued coefficients. Empirical studies have suggested that the elastic net technique can outperform lasso on data with highly correlated predictors.

Definition of Lasso

The *lasso* technique solves this regularization problem. For a given value of λ, a nonnegative parameter, `lasso` solves the problem

$$\min_{\beta_0,\beta}\left(\frac{1}{2N}\sum_{i=1}^{N}\left(y_i-\beta_0-x_i^T\beta\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|\right),$$

where

- N is the number of observations.
- y_i is the response at observation i.
- x_i is data, a vector of p values at observation i.
- λ is a positive regularization parameter corresponding to one value of `Lambda`.
- The parameters β_0 and β are scalar and p-vector respectively.

As λ increases, the number of nonzero components of β decreases.

The lasso problem involves the L^1 norm of β, as contrasted with the elastic net algorithm.

Definition of Elastic Net

The *elastic net* technique solves this regularization problem. For an α strictly between 0 and 1, and a nonnegative λ, elastic net solves the problem

$$\min_{\beta_0,\beta}\left(\frac{1}{2N}\sum_{i=1}^{N}\left(y_i-\beta_0-x_i^T\beta\right)^2+\lambda P_\alpha(\beta)\right),$$

where

$$P_\alpha(\beta)=\frac{(1-\alpha)}{2}\|\beta\|_2^2+\alpha\|\beta\|_1=\sum_{j=1}^{p}\left(\frac{(1-\alpha)}{2}\beta_j^2+\alpha|\beta_j|\right).$$

Elastic net is the same as lasso when $\alpha = 1$. As α shrinks toward 0, elastic net approaches `ridge` regression. For other values of α, the penalty term $P_\alpha(\beta)$ interpolates between the L^1 norm of β and the squared L^2 norm of β.

References

[1] Tibshirani, R. *Regression shrinkage and selection via the lasso.* Journal of the Royal Statistical Society, Series B, Vol 58, No. 1, pp. 267–288, 1996.

[2] Zou, H. and T. Hastie. *Regularization and variable selection via the elastic net.* Journal of the Royal Statistical Society, Series B, Vol. 67, No. 2, pp. 301–320, 2005.

[3] Friedman, J., R. Tibshirani, and T. Hastie. *Regularization paths for generalized linear models via coordinate descent.* Journal of Statistical Software, Vol 33, No. 1, 2010. http://www.jstatsoft.org/v33/i01

[4] Hastie, T., R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning,* 2nd edition. Springer, New York, 2008.