Using lasso and related estimators for prediction

Di Liu@StataCorp

Motivation: Prediction

What is a prediction?

- Predict an outcome in new data using information from existing data
- Good prediction minimizes mean-squared error (or another loss function) in new data

Examples:

- We have data on housing prices with hundreds of predictors. What would be the value of a new house?
- Given a new application for a credit card, what would be the probability of default?

Questions

- Suppose you have many covariates, what belongs to the prediction model?

- What if there are more variables than number of observations?

Assumption

- We assume that there are only a few variables that matter for good predictions (sparsity assumption)

Why not just run OLS regression using all covariates?

- It may not be feasible if there are more variables than observations (the matrix $X'X$ is not invertible)
- Even if it is feasible, too many covariates may cause **overfitting**
- **Overfitting** is the inclusion of extra parameters that improve the in-sample fit but increase the out-of-sample prediction errors
- These extra parameters capture the in-sample noise, but they perform poorly in the out-of-sample prediction

Using penalized regression to avoid overfitting

$$\widehat{\beta} = argmin_{\beta} \left\{ \sum_{i=1}^{N} L(x_i'\beta, y_i) + P(\beta) \right\}$$

where $L()$ is the loss function and $P(\beta)$ is the penalization.

- For linear model, $L(x_i'\beta, y_i) = (y_i - x_i'\beta)^2$. For nonlinear model, it is the negative log-likelihood function
- The penalty term $P(\beta)$ penalizes including many or large coefficients
- $\widehat{\beta}$ are the penalized coefficients (prediction example)

Penalization

$$\widehat{\beta} = argmin_{\beta} \left\{ \sum_{i=1}^{N} L(x_i'\beta, y_i) + P(\beta) \right\}$$

estimator	$P(\beta)$
lasso	$\lambda \sum_{j=1}^{p} \lvert\beta_j\rvert$
ridge	$\frac{\lambda}{2} \sum_{j=1}^{p} \beta_j^2$
elastic net	$\lambda \left[\alpha \sum_{j=1}^{p} \lvert\beta_j\rvert + \frac{(1-\alpha)}{2} \sum_{j=1}^{p} \beta_j^2 \right]$

- The elastic-net estimator is a mixture of lasso and ridge regression (elastic-net example)
- We solve this optimization problem by searching over a grid of λ's (and α's)

Overview of Stata 16's lasso features

- Lasso and elastic net can select variables from a lot of variables
- You can use these selected variables to
  - predict an outcome using lasso toolbox (today's talk)
  - estimate the effect of other variables of interest on the outcome using the selected variables as controls (next webinar)

Lasso toolbox overview

- Estimation
  - **lasso**
  - **elasticnet**
  - **sqrtlasso**
- Graph
  - **cvplot**
  - **coefpath**
- Exploratory tools
  - **lassoinfo**
  - **lassoknots**
  - **lassocoef**
  - **lassoselect**
- Prediction
  - **splitsample**
  - **predict**
  - **lassogof**

Example: Predicting housing value

Goal: We have data on housing prices with hundreds of predictors. What would be the value of a new house?

Data: Extract from American Housing Survey

Features: The number of bedrooms, the number of rooms, building age, insurance, access to Internet, lot size, time in house, cars per person, ...

Variables: Raw features and interactions (more than 300 variables)

Question: Among **OLS**, **lasso**, **elastic net**, and **ridge** regression, which estimator should be used to predict the house value?

Load data and define potential covariates

```
. /*---------- load data ------------------------*/
.
. use housing, clear

.
.
. /*----------- define potential covariates ----*/
.
. global vlcont bedrooms rooms bag insurance internet tinhouse    ///
>        vpperson serialno crhincome children npersons hincome
. global vlfv lotsize bath tenure state
. global rawvars c.($vlcont) i.($vlfv)
. global covars ($rawvars)##($rawvars)
```

Workflow for prediction

1. **Split** the data into training sample and testing sample
2. **Obtain** $\widehat{\beta}$ for each prediction technique using training sample only
3. **Evaluate** the prediction model performance of each technique using the testing sample and choose the best one
4. **Predict** outcome variable in a new dataset using the chosen model

Step 1: Split data into a training and testing sample

Firewall principle

The training sample should separate from the testing sample.

```
. /*---------- Step 1: split data --------------*/
.
. splitsample, generate(sample) split(0.7 0.3)
. label define lbsample 1 "Training" 2 "Testing"
. label value sample lbsample
.
. tabulate sample
```

sample	Freq.	Percent	Cum.
Training	1,820	70.00	70.00
Testing	780	30.00	100.00
Total	2,600	100.00	

Step 2: Obtain $\widehat{\beta}$ using training sample

```
. /*---------- Step 2: run in training sample ----*/
.
. //---------- OLS -------------//
. regress lnvalue $covars if sample == 1
. estimates store ols
.
. //---------- Lasso -----------//
. lasso linear lnvalue $covars if sample == 1
. estimates store lasso
.
. //---------- Elastic net -----//
. elasticnet linear lnvalue $covars if sample == 1, alpha(0.2 0.5 0.75 0.9)
. estimates store enet
.
. //---------- ridge ----------//
. elasticnet linear lnvalue $covars if sample == 1, alpha(0)
. estimates store ridge
```

- **if sample == 1** restricts the estimator to the training sample only
- In **elasticnet**, option **alpha()** specifies α's to search in penalty term $\alpha||\beta||_1 + [(1-\alpha)/2]||\beta||_2^2$ (penalized regression)
- Specifying **alpha(0)** is ridge regression

The first look at **lasso** output

```
. estimates restore lasso
(results lasso are active now)

. lasso

Lasso linear model                          No. of obs         =      1,820
                                            No. of covariates =        338
Selection: Cross-validation                 No. of CV folds    =         10
```

ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
1	first lambda	.5541667	0	0.0014	1.142842
38	lambda before	.0177293	39	0.4210	.662574
* 39	selected lambda	.0161543	43	0.4211	.662532
40	lambda after	.0147192	45	0.4206	.6630723
43	last lambda	.0111345	62	0.4185	.6654689

* lambda selected by cross-validation.

- Lasso **selects** only 43 variables among 338 potential covariates

  post-selection

- Where is $\widehat{\beta}$? Why there are 43 $\lambda's$? What is the λ^* selected by cross-validation? A closer look at lasso

elasticnet output

```
. estimates restore enet
(results enet are active now)

. elasticnet

Elastic net linear model                    No. of obs         =      1,820
                                            No. of covariates  =        337
Selection: Cross-validation                 No. of CV folds    =         10
```

alpha	ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
0.900						
	1	first lambda	2.770833	0	0.0008	1.145315
	54	lambda before	.0216198	35	0.4237	.6595478
	* 55	selected lambda	.0196992	41	0.4239	.659291
	56	lambda after	.0179492	45	0.4237	.6595447
	59	last lambda	.0135779	58	0.4214	.6621048
0.750						
	60	first lambda	2.770833	0	0.0008	1.145315
	117	last lambda	.0149017	67	0.4202	.6635033
0.500						
	118	first lambda	2.770833	0	0.0008	1.145315
	171	last lambda	.0216198	68	0.4190	.6649168
0.200						
	172	first lambda	2.770833	0	0.0004	1.14397
	219	last lambda	.0377813	102	0.4130	.6717475

* alpha and lambda selected by cross-validation.

- Elastic-net **selects** only 41 variables among 337 potential covariates

Ridge regression output

```
. estimates restore ridge
(results ridge are active now)

. elasticnet

Elastic net linear model                    No. of obs        =      1,820
                                            No. of covariates =        337
Selection: Cross-validation                 No. of CV folds   =         10

--------------------------------------------------------------------------
             |                               No. of   Out-of-     CV mean
             |                              nonzero    sample  prediction
alpha     ID |    Description      lambda     coef.  R-squared       error
-------------+------------------------------------------------------------
0.000        |
           1 |    first lambda   554.1667       337     0.0008    1.145315
          88 |   lambda before   .1692345       337     0.3940     .693461
        * 89 | selected lambda   .1542002       337     0.3942     .693273
          90 |    lambda after   .1405014       337     0.3942    .6932859
         100 |     last lambda   .0554167       337     0.3851    .7036694
--------------------------------------------------------------------------
* alpha and lambda selected by cross-validation.
```

- Ridge regression **selects** all variables
- But different λ leads to a different estimate of β

Step 3: Evaluate prediction performance using testing sample

```
. /*---------- Step 3: Evaluate prediciton in testing sample ----*/
.
. lassogof ols lasso enet ridge, over(sample)
```

Penalized coefficients

Name	sample	MSE	R-squared	Obs
ols				
	Training	.550408	0.5190	1,820
	Testing	.6536993	0.3483	780
lasso				
	Training	.6270008	0.4521	1,820
	Testing	.5566048	0.4451	780
enet				
	Training	.6303752	0.4492	1,820
	Testing	.5575001	0.4442	780
ridge				
	Training	.599504	0.4761	1,820
	Testing	.5730015	0.4287	780

- We choose lasso as the best prediction because it has the smallest MSE in the testing sample

Step 4: Predict housing value (1)

```
. /*---------- Step 4: Predict housing value using chosen estimator -*/
.
. //------ chose lasso result ---------//
. estimates restore lasso
(results lasso are active now)

.
. //------ load new data where housing value is not observed ---//
. use housing_new, clear

.
. //-------- penalized coefficients ----------//
. predict y_pen
(options xb penalized assumed; linear prediction with penalized coefficients)
```

- Default option **xb**: in the linear model, we compute $x_i'\widehat{\beta}$
- Default option **penalized**: we use the $\widehat{\beta}$ from the lasso regression (See penalized regression)

Step 4: Predict housing value (2)

```
. //-------- post-selection coefficients ----------//
. predict y_postsel, postselection
(option xb assumed; linear prediction with postselection coefficients)
```

- Option **postselection**: **OLS y on** X^* gives post-selection $\tilde{\beta}$, where X^* are variables selected by lasso
- Post-selection coefficients are less biased. In the linear model, they may have better out-of-sample prediction performance than the penalized coefficients (Belloni et al., 2013)
- For the nonlinear models, there is no theory

A closer look at lasso (1)

Lasso (Tibshirani, 1996) is

$$\widehat{\beta} = argmin_{\beta} \left\{ \sum_{i=1}^{N} L(x_i'\beta, y_i) + \lambda \sum_{j=1}^{p} \omega_j |\beta_j| \right\}$$

where

- λ is the lasso penalty parameter and ω_j is the penalty loading
- The kink in the absolute value function causes some elements in $\widehat{\beta}$ to be zero given some value of λ
- Lasso is also a variable-selection technique
  - covariates with $\widehat{\beta}_j = 0$ are excluded
  - covariates with $\widehat{\beta}_j \neq 0$ are included

A closer look at lasso (2)

$$\widehat{\beta} = argmin_{\beta} \left\{ \sum_{i=1}^{N} L(x_i'\beta, y_i) + \lambda \sum_{j=1}^{p} \omega_j |\beta_j| \right\}$$

- **lasso** searches over a grid of $\lambda's$, and each λ corresponds to a different β estimate (a different model)
- There is a λ_{max} that shrinks all the coefficients to zero
- As λ decreases, more variables will be selected
- How to choose λ? (choose λ)

The second look at **lasso** output

```
. estimates restore lasso
(results lasso are active now)

. lasso

Lasso linear model                          No. of obs        =      1,820
                                            No. of covariates =        338
Selection: Cross-validation                 No. of CV folds   =         10
```

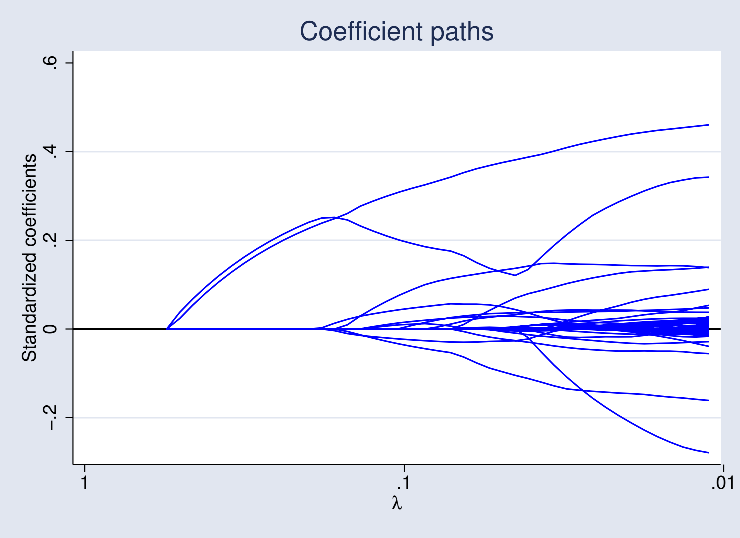
ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
1	first lambda	.5541667	0	0.0014	1.142842
38	lambda before	.0177293	39	0.4210	.662574
* 39	selected lambda	.0161543	43	0.4211	.662532
40	lambda after	.0147192	45	0.4206	.6630723
43	last lambda	.0111345	62	0.4185	.6654689

```
* lambda selected by cross-validation.
```

- The number of nonzero coefficients increases as λ decreases

coefpath: Coefficients path plot

```
. coefpath, xunits(rlnlambda)
```



Dynamic of coefficient path

lassoknots: Display knot table

```
. lassoknots
─────────────────────────────────────────────────────────────────────────────
     |                  No. of    CV mean |
     |                 nonzero      pred. |   Variables (A)dded, (R)emoved,
  ID |     lambda        coef.      error |        or left (U)nchanged
─────────────────────────────────────────────────────────────────────────────
   2 |    .504936            2   1.083387 | A insurance
     |                                    |   c.crhincome#c.hincome
  13 |   .1814646            3   .7871774 | A c.insurance#c.vpperson
  14 |   .1653438            4   .7785965 | A c.bage#c.internet

(output omitted ...)

  41 |   .0134115           51    .663886 | A 22.state#c.tinhouse
     |                                    |   47.state#c.tinhouse
     |                                    |   2.lotsize#c.bage
     |                                    |   3.lotsize#22.state
     |                                    |   3.lotsize#2.tenure
     |                                    |   1.lotsize#c.internet
     |                                    |   1.lotsize#c.serialno
  41 |   .0134115           51    .663886 | R 1.bath#c.internet
  42 |   .0122201           55    .664712 | A 48.state#c.insurance
     |                                    |   5.state#c.crhincome
     |                                    |   3.lotsize#c.bage
     |                                    |   2.lotsize#48.state
     |                                    |   c.insurance#c.hincome
  42 |   .0122201           55    .664712 | R 3.lotsize#1.tenure
  43 |   .0111345           62   .6654689 | A 1.state#c.crhincome
     |                                    |   22.state#c.crhincome
     |                                    |   2.tenure#40.state
     |                                    |   1.lotsize#47.state
     |                                    |   2.lotsize#5.state
     |                                    |   2.lotsize#c.npersons
     |                                    |   c.children#c.npersons
─────────────────────────────────────────────────────────────────────────────
* lambda selected by cross-validation.
```

- A λ is a knot if a variable is added or removed from the model

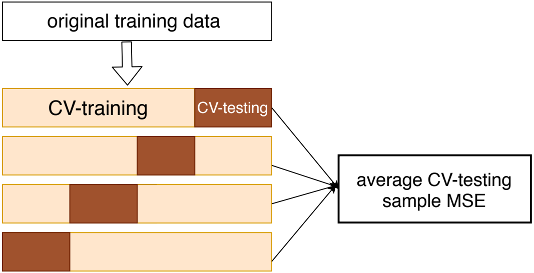
How to choose λ?

For **lasso**, we can choose λ by cross-validation, adaptive lasso, plugin, and manual choice.

- **Cross-validation** mimics the process of doing out-of-sample prediction. It produces estimates of out-of-sample MSE and selects λ with minimum MSE
- **Adaptive lasso** performs multiple lassos, each with CV. After each lasso, variables with zero coefficients are removed and remaining variables are given penalty weights ω_j designed to drive small coefficients to zero. Thus, adaptive lasso typically selects fewer covariates than CV (lasso formula)
- The **Plugin** method is designed to dominate the estimation noise. It tends to selects fewer variables than CV or adaptive

How does cross-validation work?

1. Based on data, compute a sequence of λ's as $\lambda_1 > \lambda_2 > \cdots > \lambda_k$. λ_1 makes all coefficients zero (no variables are selected)
2. For each λ_j, do K-fold cross-validation to get an estimate of out-of-sample MSE



3. Select the λ^* with the smallest estimate of out-of-sample MSE, and refit lasso using λ^* and original training sample

The third look at **lasso** output

```
. estimates restore lasso
(results lasso are active now)

. lasso

Lasso linear model                          No. of obs        =      1,820
                                            No. of covariates =        338
Selection: Cross-validation                 No. of CV folds   =         10
```

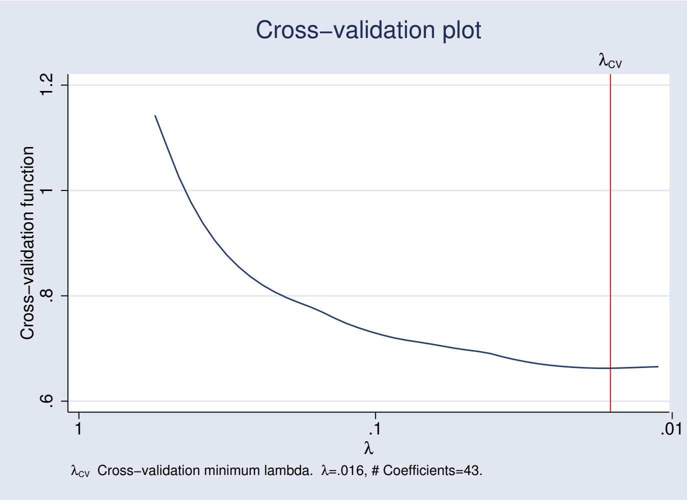
ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
1	first lambda	.5541667	0	0.0014	1.142842
38	lambda before	.0177293	39	0.4210	.662574
* 39	selected lambda	.0161543	43	0.4211	.662532
40	lambda after	.0147192	45	0.4206	.6630723
43	last lambda	.0111345	62	0.4185	.6654689

```
* lambda selected by cross-validation.
```

- The selected λ^* has the smallest CV mean prediction error and largest out-of-sample R-squared estimate
- By default **lasso** searches over 100 $\lambda's$, but there are only 43 λ's here. Why?

cvplot: Cross-validation plot

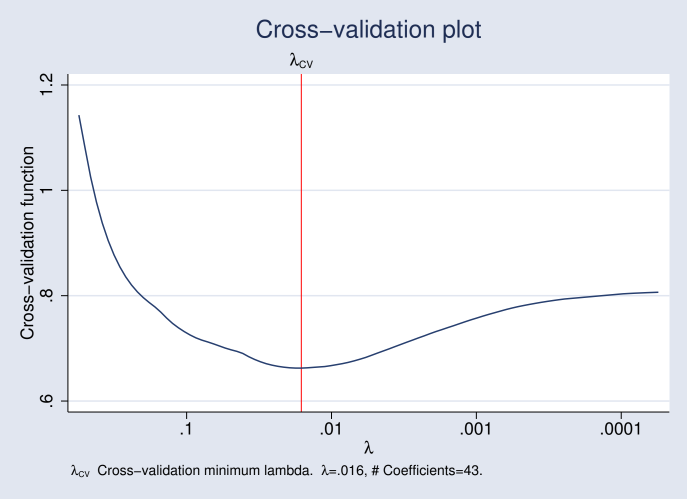
```
. cvplot
```



- **lasso** stops searching for λ once it finds a valid CV minimum

cvplot: Full picture

```
. lasso linear lnvalue $covars if sample == 1, stop(0) selection(cv, alllambdas)
. cvplot
```



- It may take a long time to search all the λ's

Use option **selection()** to choose λ

```
. lasso linear lnvalue $covars if sample == 1
. estimates store cv
.
. lasso linear lnvalue $covars if sample == 1, selection(adaptive)
. estimates store adaptive
.
. lasso linear lnvalue $covars if sample == 1, selection(plugin)
. estimates store plugin
```

lassoinfo: Lasso information summary

```
. lassoinfo cv adaptive plugin

    Estimate:  cv
     Command:  lasso

------------------------------------------------------------------------
             |                                                  No. of
             |             Selection  Selection                selected
      Depvar |      Model     method  criterion     lambda    variables
-------------+----------------------------------------------------------
     lnvalue |     linear         cv    CV min.   .0177293          39
------------------------------------------------------------------------

    Estimate:  adaptive
     Command:  lasso

------------------------------------------------------------------------
             |                                                  No. of
             |             Selection  Selection                selected
      Depvar |      Model     method  criterion     lambda    variables
-------------+----------------------------------------------------------
     lnvalue |     linear   adaptive    CV min.   .2314885          17
------------------------------------------------------------------------

    Estimate:  plugin
     Command:  lasso

-------------------------------------------------------------
             |                                       No. of
             |             Selection                selected
      Depvar |      Model     method     lambda    variables
-------------+-----------------------------------------------
     lnvalue |     linear     plugin   .1060145           12
-------------------------------------------------------------
```

- Adaptive lasso selects fewer variables than regular lasso
- Plugin selects even fewer variables than adaptive lasso

lassocoef: Display lasso coefficients

```
. lassocoef cv adaptive plugin, display(coef)

─────────────────────────────────────────────────────────────────────────────────
                                            |        cv   adaptive     plugin
────────────────────────────────────────────┼────────────────────────────────────
                                      rooms |  .0036953               .0117244
                                  insurance |  .3114183   .4373481    .1495797
                                   vpperson |  .0052322
                                            |
                         c.bedrooms#c.rooms |  .0107225               .0111987
                                            |
(output omitted ...)
                                            |
                                bath#tenure |
          no#Owned with mortgage or loan    |                         .0008039
                                            |
                                      _cons |         0          0           0
─────────────────────────────────────────────────────────────────────────────────
Legend:
  b - base level
  e - empty cell
  o - omitted
```

lassoselect: Manually choose a λ (1)

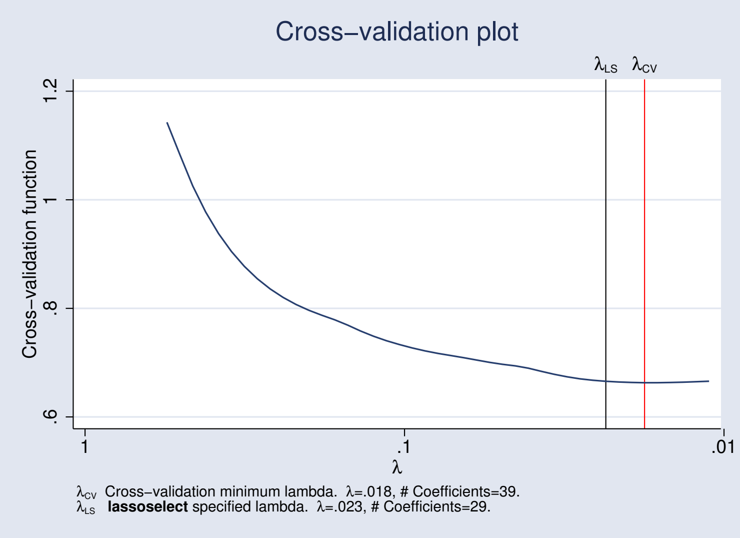
- Suppose you want to choose λ with the minimum BIC, there is no need to rerun **lasso**
- First, let's look at output from **lassoknots** for BIC

```
. estimates restore cv
(results cv are active now)

. lassoknots, display(nonzero bic)
─────────────────────────────────────────────────
         |                   No. of
         |                  nonzero
      ID |     lambda         coef.           BIC
─────────┼───────────────────────────────────────
       2 |    .504936             2      5327.691
      13 |   .1814646             3       4753.82
     (output omitted ...)
      34 |   .0257221            28      4580.049
      34 |   .0257221            28      4580.049
      35 |   .0234371            29      4577.228
      36 |    .021355            33      4597.566
      36 |    .021355            33      4597.566
      37 |   .0194579            34      4595.408
      37 |   .0194579            34      4595.408
    * 38 |   .0177293            39      4624.164
      39 |   .0161543            43      4645.637
      39 |   .0161543            43      4645.637
      40 |   .0147192            45      4652.893
      40 |   .0147192            45      4652.893
      41 |   .0134115            51      4689.776
      41 |   .0134115            51      4689.776
      42 |   .0122201            55      4711.432
      42 |   .0122201            55      4711.432
      43 |   .0111345            62      4755.442
─────────────────────────────────────────────────
* lambda selected by cross-validation.
```

lassoselect: Manually choose a λ (2)

```
. lassoselect id = 35
ID = 35  lambda = .0234371 selected
. estimates store bic
. cvplot
```



Comparing CV, adaptive, plugin, and BIC

```
. lassogof cv bic adaptive plugin if sample == 2
```

Penalized coefficients

Name	MSE	R-squared	Obs
cv	.5571567	0.4445	780
bic	.5613097	0.4404	780
adaptive	.5567655	0.4449	780
plugin	.6087777	0.3931	780

```
.
. lassogof cv bic adaptive plugin if sample == 2, postselection
```

Postselection coefficients

Name	MSE	R-squared	Obs
cv	.5713665	0.4304	780
bic	.5622546	0.4394	780
adaptive	.5626561	0.4390	780
plugin	.5915617	0.4102	780

Lasso toolbox summary

- Estimation:
  - **lasso** and **elasticnet** for linear, binary, and count data
  - **sqrtlasso** for linear data
  - cross-validation, adaptive lasso, plugin, and manual selection
- Graph:
  - **cvplot**: cross-validation plot
  - **coefpath**: coefficient path
- Exploratory tools:
  - **lassoinfo**: summary of lasso fitting
  - **lassoknots**: table of knots
  - **lassocoef**: display lasso coefficients
  - **lassoselect**: manually select λ (or α)
- Prediction
  - **splitsample**: randomly divide data into different samples
  - **predict**: prediction
  - **lassogof**: evaluate in-sample and out-of-sample prediction

References

Belloni, A., V. Chernozhukov, et al. 2013. Least squares after model selection in high-dimensional sparse models. *Bernoulli* 19(2): 521–547.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1): 267–288.