Using lasso and related estimators for prediction

Di Liu

StataCorp

July 12, 2019

Prediction

What is a prediction?

- Prediction is to predict an outcome variable on new (unseen) data
- Good prediction minimizes mean-squared error (or other loss function) on new data

Examples:

- Given some characteristics, what would be the value of a house?
- Given an application of credit card, what would be probability of default for a customer?

Question:

Suppose I have many covariates, then which one should I include in my prediction model?

Using penalized regression to avoid overfitting

Why not include all potential covariates?

- It may not be feasible if $p > N$
- Even if it is feasible, too many covariates may cause overfitting
- Overfitting is the inclusion of extra parameters that reduce the in-sample loss but increase the out-of-sample loss

Penalized regression

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

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

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

Example: Predicting housing value

Goal: Given some characteristics, what would be the value of a house?

data: Extract from American Housing Survey

characteristics: The number of bedrooms, the number of rooms, building age, insurance, access to internet, lot size, time in house, and cars per person

variables: Raw characteristics and interactions (more than 100 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 ----*/
.
. local vlcont bedrooms rooms bag insurance internet tinhouse vpperson
. local vlfv lotsize bath tenure
. local covars `vlcont´ i.(`vlfv´)                                      ///
>         (c.(`vlcont´) i.(`vlfv´))##(c.(`vlcont´) i.(`vlfv´))
```

Step 1: Split data into training and hold-out sample

Firewall principle

The training dataset used to train the model should not contain information from hold-out sample used to evaluate prediction performance

```
. /*---------- Step 1: split data --------------*/
.
. splitsample, generate(sample) split(0.70 0.30)
. label define lbsample 1 "traning" 2 "hold-out"
. label value sample lbsample
```

Step 2: Choose tuning parameter using training data

```
. /*---------- Step 2: run in traing sample ----*/
.
. quietly regress lnvalue `covars´ if sample == 1
. estimates store ols
.
. quietly lasso linear lnvalue `covars´ if sample == 1
. estimates store lasso
.
. quietly elasticnet linear lnvalue `covars´ if sample == 1, alpha(0.2 0.5 0.75
>  0.9)
. estimates store enet
.
. quietly elasticnet linear lnvalue `covars´ if sample == 1, alpha(0)
. estimates store ridge
```

- **if sample == 1**, restricts estimator to use training data only
- By default, we choose the tuning parameter by cross-validation
- We use **estimates store** to store lasso results
- In **elasticnet**, option **alpha()** specifies α in penalty term $\alpha||\beta||_1 + [(1-\alpha)/2]||\beta||_2^2$
- Specifying **alpha(0)** is ridge regression

Step 3: Evaluate prediction performance using hold-out sample

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

Name	sample	MSE	R-squared	Obs
ols				
	traning	1.104663	0.2256	4,425
	hold-out	1.184776	0.1813	1,884
lasso				
	traning	1.127425	0.2129	4,396
	hold-out	1.183058	0.1849	1,865
enet				
	traning	1.124424	0.2150	4,396
	hold-out	1.180599	0.1866	1,865
ridge				
	traning	1.119678	0.2183	4,396
	hold-out	1.187979	0.1815	1,865

- We choose elastic-net as the best prediction because it has the smallest MSE in hold-out sample

Step 4: Predict housing value using chosen estimator

```
. /*---------- Step 4: Predict housing value using chosen estimator -*/
.
. use housing_new, clear

. estimates restore enet
(results enet are active now)

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

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

- By default, **predict** uses the penalized coefficients to compute $x_i\beta'$
- Specifying option **postselection** makes **predict** use post-selection coefficients, which are from OLS on variables selected by **elasticnet**
- In the **linear** model, post-selection coefficients tend to be less biased and may have better out-of-sample prediction performance than the penalized coefficients

A closer look at lasso

Lasso 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
- We solve the optimzation for a set of λ's
- 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
- Given a dataset, there exists a λ_{max} that shrink all the coefficients to zero
- As λ decreases, more variables will be selected

lasso output

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

. lasso

Lasso linear model                          No. of obs         =      4,396
                                            No. of covariates  =        102
Selection: Cross-validation                 No. of CV folds    =         10
```

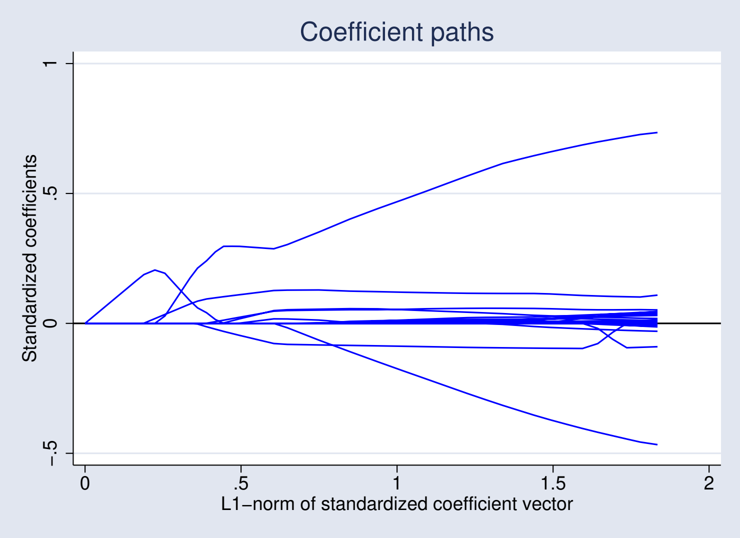
ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
1	first lambda	.4396153	0	0.0004	1.431814
39	lambda before	.012815	21	0.2041	1.139951
* 40	selected lambda	.0116766	22	0.2043	1.139704
41	lambda after	.0106393	23	0.2041	1.140044
44	last lambda	.0080482	28	0.2011	1.144342

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

- We see the number of nonzero coefficients increases as λ decreases
- By default, **lasso** uses 10-fold cross-validation to choose λ

coefpath: Coefficients path plot

```
. coefpath
```



lassoknots: Display knot table

```
. lassoknots
```

ID	lambda	No. of nonzero coef.	CV mean pred. error	Variables (A)dded, (R)emoved, or left (U)nchanged
2	.4005611	1	1.399934	A 1.bath#c.insurance
7	.251564	2	1.301968	A 1.bath#c.rooms
9	.2088529	3	1.27254	A insurance
13	.1439542	4	1.235793	A internet
(output omitted ...)				
35	.0185924	19	1.143928	A c.insurance#c.tinhouse
37	.0154357	20	1.141594	A 2.lotsize#c.insurance
39	.012815	21	1.139951	A c.bage#c.bage 2.bath#c.bedrooms
39	.012815	21	1.139951	R 1.tenure#c.bage
* 40	.0116766	22	1.139704	A 1.bath#c.internet
41	.0106393	23	1.140044	A c.internet#c.vpperson
42	.0096941	23	1.141343	A 2.lotsize#1.tenure
42	.0096941	23	1.141343	R internet
43	.0088329	25	1.143217	A 2.bath#2.tenure 2.tenure#c.insurance
44	.0080482	28	1.144342	A c.rooms#c.rooms 2.tenure#c.bedrooms 1.lotsize#c.internet

* lambda selected by cross-validation.

- One λ is a knot if a new variable is added or removed from the model
- We can use **lassoselect** to choose a different λ. See lassoselect

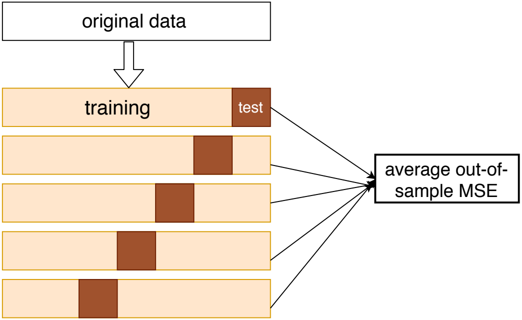
How to choose λ?

For **lasso**, we can choose λ by cross-valiation, adaptive lasso, plugin, and customized 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 is an iterative procedure of cross-validated lasso. It puts more penalty weights on small coefficients than a regular lasso. Covariates with large coefficients are more likely to be selected, and covariates with small coefficients are more likely to be dropped
- Plugin method finds λ that is large enough to dominate the estimation noise

How does cross-validation work?

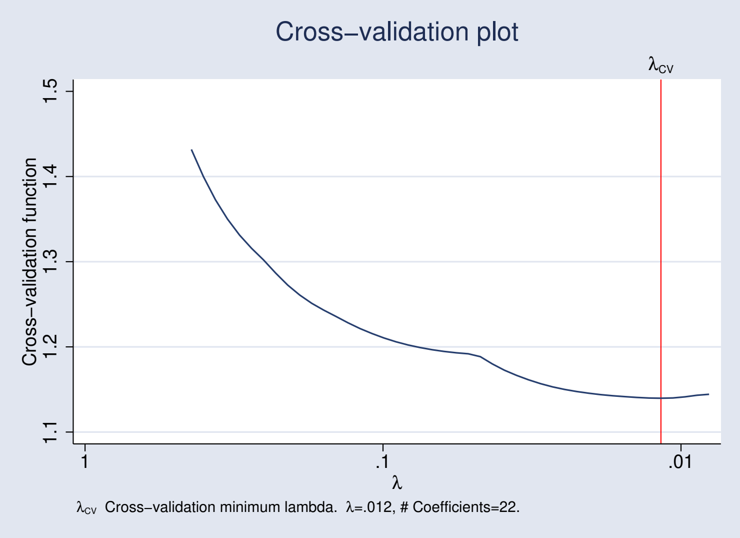
1. Based on data, compute a sequence of λ's as $\lambda_1 > \lambda_2 > \cdots > \lambda_k$. λ_1 set all the coefficients to zero (no variables are selected)
2. For each λ_j, do K-fold cross-validation to get an estimate of out-of-sample MSE



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

cvplot: Cross-validation plot

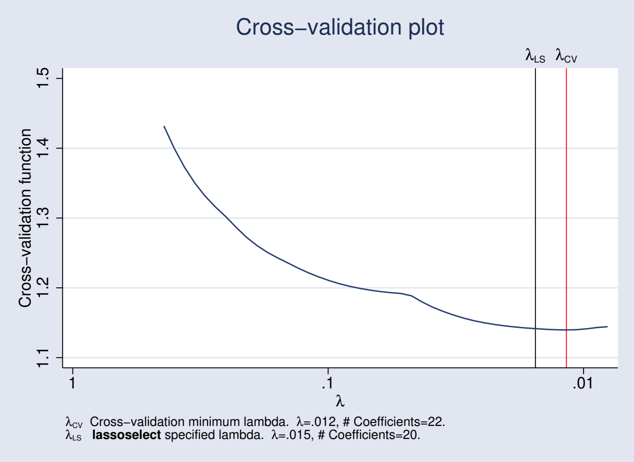
```
. cvplot
```



lassoselect: Manually choose a λ

- First, let's look at output from **lassoknots** lassoknots

```
. estimates restore lasso
(results lasso are active now)
. lassoselect id = 37
ID = 37  lambda = .0154357 selected
.
. cvplot
```



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

```
. quietly lasso linear lnvalue `covars´
. estimates store cv
.
. quietly lasso linear lnvalue `covars´ , selection(adaptive)
. estimates store adaptive
.
. quietly lasso linear lnvalue `covars´ , selection(plugin)
. estimates store plugin
```

lassoinfo: lasso information summary

```
. lassoinfo cv adaptive plugin

    Estimate:  cv
     Command:  lasso
```

Depvar	Model	Selection method	Selection criterion	lambda	No. of selected variables
lnvalue	linear	cv	CV min.	.0034279	36

```
    Estimate:  adaptive
     Command:  lasso
```

Depvar	Model	Selection method	Selection criterion	lambda	No. of selected variables
lnvalue	linear	adaptive	CV min.	.0183654	16

```
    Estimate:  plugin
     Command:  lasso
```

Depvar	Model	Selection method	lambda	No. of selected variables
lnvalue	linear	plugin	.0537642	10

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

Lasso toolbox summary

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