Using lasso and related estimators for prediction

07/09/2013

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

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

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
- $\hat{\beta}$ are the penalized coefficients (prediction example)

Penalization

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

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

- 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** $\hat{\beta}$ 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 $\hat{\beta}$ 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 $\alpha||\beta||_1 + [(1-\alpha)/2]||\beta||_2^2(\textit{penalizedregression})$
- Specifying **alpha(0)** is ridge regression

The first look at **lasso** output

```
. estimates restore lasso
```

- Lasso **selects** only 43 variables among 338 potential covariates post-selection
- Where is $\hat{\beta}$? Why there are 43 λ s? What is the λ^* selected by cross-validation? A closer look at lasso

elasticnet output

```
. estimates restore lasso
```

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

Ridge regression output

```
. estimates restore lasso
```

- 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 prediction in testing
sample ----*/
```

- 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 -*/
```

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

Step 4: Predict housing value (2)

```
. //------post-selection coefficients -------//
```

- Option postselection: OLS y on X^* gives post-selection $\hat{\beta}$, 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

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

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 $\hat{\beta}$ to be zero given some value of λ
- Lasso is also a variable-selection technique
  - covariates with $\hat{\beta}_j = 0$ are excluded
  - covariates with $\hat{\beta}_j \neq 0$ are included

A closer look at lasso (2)

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

where

- **lasso** searches over a grid of λ s, and each   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
```

- 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
```

- 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 fewercovariatesthan 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
```

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

cvplot: Cross-validation plot

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

cvplot: Full picture

- 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
```

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

lassocoef: Display lasso coefficients

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

lassoselect: Manually choose a λ (1)

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

```
. estimates restore cv
```

lassoselect: Manually choose a λ (2)

```
. lassoselect id = 35
```

Comparing CV, adaptive, plugin, and BIC

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

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.