lassogof — Goodness of fit after lasso for prediction

Description

lassogof calculates goodness of fit of predictions after lasso, sqrtlasso, and elasticnet. It also calculates goodness of fit after regress, logit, probit, and poisson estimations for comparison purposes. For linear models, mean squared error of the prediction and $R^2$ are displayed. For logit, probit, and Poisson models, deviance and deviance ratio are shown.

Quick start

See goodness of fit for current lasso result using penalized coefficient estimates

lassogof

See goodness of fit for current lasso result using postselection coefficient estimates

lassogof, postselection

See goodness of fit for four stored estimation results

lassogof mylasso mysqrtlasso myelasticnet myregress

See goodness of fit for all stored estimation results

lassogof *

Randomly split sample into two, fit a lasso on the first sample, and calculate goodness of fit separately for both samples

splitsample, generate(sample) nsplit(2)
lasso linear y x* if sample == 1
lassogof, over(sample)

Menu

Statistics > Postestimation
Syntax

```
lassogof [namelist] [if] [in] [, options]
```

`namelist` is a name of a stored estimation result, a list of names, `._all`, or `*. _all` and `*` mean the same thing. See [R] estimates store.

**options**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>use penalized (shrunken) coefficient estimates; the default</td>
</tr>
<tr>
<td>use postselection coefficient estimates</td>
</tr>
<tr>
<td>display goodness of fit for samples defined by <code>varname</code></td>
</tr>
<tr>
<td>do not use weights when calculating goodness of fit</td>
</tr>
</tbody>
</table>

**Options**

**Main**

- `penalized` specifies that the penalized coefficient estimates be used to calculate goodness of fit. Penalized coefficients are those estimated by lasso with shrinkage. This is the default.
- `postselection` specifies that the postselection coefficient estimates be used to calculate goodness of fit. Postselection coefficients are estimated by taking the covariates selected by lasso and reestimating the coefficients using an unpenalized estimator—namely, an ordinary linear regression, logistic regression, probit model, or Poisson regression as appropriate.
- `over(varname)` specifies that goodness of fit be calculated separately for groups of observations defined by the distinct values of `varname`. Typically, this option would be used when the lasso is fit on one sample and one wishes to compare the fit in that sample with the fit in another sample.
- `noweights` specifies that any weights used to estimate the lasso be ignored in the calculation of goodness of fit.

**Remarks and examples**

`lassogof` is intended for use on out-of-sample data. That is, on data different from the data used to fit the lasso.

There are two ways to do this. One is to randomly split your data into two subsamples before fitting a lasso model. The examples in this entry show how to do this using `splitsample`.

The other way is to load a different dataset in memory and run `lassogof` with the lasso results on it. The steps for doing this are as follows.

1. Load the data on which you are going to fit your model.
   ```
   . use datafile1
   ```

2. Run `lasso` (or `sqrtlasso` or `elasticnet`).
   ```
   . lasso ... 
   ```

3. Save the results in a file.
   ```
   . estimates save filename
   ```
4. Load the data for testing the prediction.
   . use datafile2, clear

5. Load the saved results, making them the current (active) estimation results.
   . estimates use filename

6. Run lassogof.
   . lassogof

Example 1: Comparing fit in linear models

We will show how to use lassogof after lasso linear.

Here is an example using lasso from [LASSO] lasso examples. We load the data and make the vl variable lists active.

   . use https://www.stata-press.com/data/r16/fakesurvey_vl
   (Fictitious Survey Data with vl)
   . vl rebuild

We now use splitsample to generate a variable indicating the two subsamples.

   . set seed 1234
   . splitsample, generate(sample) nsplit(2)
   . label define svalues 1 "Training" 2 "Testing"
   . label values sample svalues

We run lasso on the first subsample and set the random-number seed using the rseed() option so we can reproduce our results.

   . lasso linear q104 ($idemographics) $ifactors $vlcontinuous
   > if sample == 1, rseed(1234)

   Lasso linear model
   No. of obs = 458
   No. of covariates = 273
   Selection: Cross-validation
   No. of CV folds = 10

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>lambda</th>
<th>No. of nonzero coef.</th>
<th>Out-of-sample R-squared</th>
<th>CV mean prediction error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>first lambda</td>
<td>.9109571</td>
<td>4</td>
<td>0.0147</td>
<td>16.93341</td>
</tr>
<tr>
<td>18</td>
<td>lambda before</td>
<td>.1873395</td>
<td>42</td>
<td>0.2953</td>
<td>12.10991</td>
</tr>
<tr>
<td>* 19</td>
<td>selected lambda</td>
<td>.1706967</td>
<td>49</td>
<td>0.2968</td>
<td>12.08516</td>
</tr>
<tr>
<td>20</td>
<td>lambda after</td>
<td>.1555325</td>
<td>55</td>
<td>0.2964</td>
<td>12.09189</td>
</tr>
<tr>
<td>23</td>
<td>last lambda</td>
<td>.1176546</td>
<td>74</td>
<td>0.2913</td>
<td>12.17933</td>
</tr>
</tbody>
</table>

* lambda selected by cross-validation.

. estimates store linearcv

After the command finished, we used estimates store to store the results in memory so we can later compare these results with those from other lassos.

We are now going to run an adaptive lasso, which we do by specifying the option selection(adaptive).
. lasso linear q104 ($idemographics) $ifactors $vlcontinuous
> if sample == 1, rseed(4321) selection(adaptive)
(output omitted)
Lasso linear model
No. of obs = 458
No. of covariates = 273
Selection: Adaptive
No. of lasso steps = 2
Final adaptive step results

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>lambda</th>
<th>No. of nonzero coef.</th>
<th>Out-of-sample R-squared</th>
<th>CV mean prediction error</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>first lambda</td>
<td>51.68486</td>
<td>4</td>
<td>0.0101</td>
<td>17.01083</td>
</tr>
<tr>
<td>77</td>
<td>lambda before</td>
<td>0.4095937</td>
<td>46</td>
<td>0.3985</td>
<td>10.33691</td>
</tr>
<tr>
<td>* 78</td>
<td>selected lambda</td>
<td>0.3732065</td>
<td>46</td>
<td>0.3987</td>
<td>10.33306</td>
</tr>
<tr>
<td>79</td>
<td>lambda after</td>
<td>0.3400519</td>
<td>47</td>
<td>0.3985</td>
<td>10.33653</td>
</tr>
<tr>
<td>124</td>
<td>last lambda</td>
<td>0.0051685</td>
<td>59</td>
<td>0.3677</td>
<td>10.86697</td>
</tr>
</tbody>
</table>

* lambda selected by cross-validation in final adaptive step.
. estimates store linearadaptive

We want to see which performs better for out-of-sample prediction. We specify the over() option with the name of our sample indicator variable, sample. We specify the postselection option because for linear models, postselection coefficients are theoretically slightly better for prediction than the penalized coefficients (which lassogof uses by default). See the discussion in `predict` in [LASSO] lasso postestimation.

. lassogof linearcv linearadaptive, over(sample) postselection
Postselection coefficients

<table>
<thead>
<tr>
<th>Name</th>
<th>sample</th>
<th>MSE</th>
<th>R-squared</th>
<th>Obs</th>
</tr>
</thead>
<tbody>
<tr>
<td>linearcv</td>
<td>Training</td>
<td>8.652771</td>
<td>0.5065</td>
<td>503</td>
</tr>
<tr>
<td></td>
<td>Testing</td>
<td>14.58354</td>
<td>0.2658</td>
<td>493</td>
</tr>
<tr>
<td>linearadaptive</td>
<td>Training</td>
<td>8.637575</td>
<td>0.5057</td>
<td>504</td>
</tr>
<tr>
<td></td>
<td>Testing</td>
<td>14.70756</td>
<td>0.2595</td>
<td>494</td>
</tr>
</tbody>
</table>

The ordinary lasso did a little better in this case than adaptive lasso.
Example 2: Comparing fit in logit and probit models

We fit a logit model on the same data we used in the previous example.

```stata
. lasso logit q106 $idemographics $ifactors $vlcontinuous
> if sample == 1, rseed(1234)
(output omitted)
```

Lasso logit model

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>lambda</th>
<th>No. of nonzero sample CV mean deviance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>first lambda</td>
<td>.1155342</td>
<td>0.0004 1.384878</td>
</tr>
<tr>
<td>22</td>
<td>lambda before</td>
<td>.0163767</td>
<td>65 0.1857 1.127315</td>
</tr>
<tr>
<td>* 23</td>
<td>selected lambda</td>
<td>.0149218</td>
<td>69 0.1871 1.125331</td>
</tr>
<tr>
<td>24</td>
<td>lambda after</td>
<td>.0135962</td>
<td>73 0.1864 1.126333</td>
</tr>
<tr>
<td>27</td>
<td>last lambda</td>
<td>.010285</td>
<td>88 0.1712 1.147343</td>
</tr>
</tbody>
</table>

* lambda selected by cross-validation.

```stata
. estimates store logit
```

Let's now fit a probit model.

```stata
. lasso probit q106 $idemographics $ifactors $vlcontinuous
> if sample == 1, rseed(1234)
(output omitted)
```

Lasso probit model

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>lambda</th>
<th>No. of nonzero sample CV mean deviance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>first lambda</td>
<td>.1844415</td>
<td>0 0.0004 1.384877</td>
</tr>
<tr>
<td>21</td>
<td>lambda before</td>
<td>.0286931</td>
<td>61 0.1820 1.132461</td>
</tr>
<tr>
<td>* 22</td>
<td>selected lambda</td>
<td>.0261441</td>
<td>64 0.1846 1.128895</td>
</tr>
<tr>
<td>23</td>
<td>lambda after</td>
<td>.0238215</td>
<td>70 0.1841 1.129499</td>
</tr>
<tr>
<td>26</td>
<td>last lambda</td>
<td>.0180201</td>
<td>87 0.1677 1.152188</td>
</tr>
</tbody>
</table>

* lambda selected by cross-validation.

```stata
. estimates store probit
```
We look at how they did for out-of-sample prediction.

. lassogof logit probit, over(sample)

Penalized coefficients

<table>
<thead>
<tr>
<th>Name</th>
<th>sample</th>
<th>Deviance</th>
<th>Deviance ratio</th>
<th>Obs</th>
</tr>
</thead>
<tbody>
<tr>
<td>logit</td>
<td>Training</td>
<td>.8768969</td>
<td>0.3674</td>
<td>499</td>
</tr>
<tr>
<td></td>
<td>Testing</td>
<td>1.268346</td>
<td>0.0844</td>
<td>502</td>
</tr>
<tr>
<td>probit</td>
<td>Training</td>
<td>.8833892</td>
<td>0.3627</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>Testing</td>
<td>1.27267</td>
<td>0.0812</td>
<td>503</td>
</tr>
</tbody>
</table>

They both did not do very well. The out-of-sample deviance ratios were notably worse than the in-sample values. The deviance ratio for nonlinear models is analogous to $R^2$ for linear models. See Methods and formulas for the formal definition.

We did not specify the postselection option in this case because there are no theoretical grounds for using postselection coefficients for prediction with nonlinear models.

Stored results

lassogof stores the following in r():

Macros
- r(names) names of estimation results displayed
- r(over_var) name of the over() variable
- r(over_levels) levels of the over() variable

Matrices
- r(table) matrix containing the values displayed

Methods and formulas

lassogof reports the mean squared error (MSE) and the $R^2$ measures of fit for linear models. It reports the deviance and the deviance ratio for logit, probit, and poisson models. The deviance ratio is also known as $D^2$ in the literature.

See Wooldridge (2020, 720) for more about MSE and Wooldridge (2020, 76–77) for more about $R^2$. The deviance measures are described in Hastie, Tibshirani, and Wainwright (2015, 29–33) and McCullagh and Nelder (1989, 33–34).

In the formulas below, we use $x_{bi}$ to denote the linear prediction for the $i$th observation. By default, the lasso penalized coefficients $\hat{\beta}$ are used to compute $x_{bi}$. Specifying the option postselection causes the postselection estimates $\hat{\beta}$ to be used to compute $x_{bi}$. See predict in [LASSO] lasso postestimation for a discussion of penalized estimates and postselection estimates.

We also use the following notation. $y_i$ denotes the $i$th observation of the outcome. $w_i$ is the weight applied to the $i$th observation; $w_i = 1$ if no weights were specified in the estimation command or if option noweights was specified in lassogof. $N$ is the number of observations in the sample over which the goodness-of-fit statistics are computed. If frequency weights were specified at estimation $N_s = \sum_{i=1}^{N} w_i$; otherwise, $N_s = N$. 
The formulas for the measures reported after linear models are

\[ R^2 = 1 - \frac{\text{RSS}}{\text{TSS}} \]

\[ \text{MSE} = \frac{1}{N_s} \frac{\text{RSS}}{} \]

where

\[ \text{RSS} = \sum_{i=1}^{N} w_i (y_i - xb_i)^2 \]

\[ \text{TSS} = \sum_{i=1}^{N} w_i (y_i - \bar{y})^2 \]

\[ \bar{y} = \frac{1}{N_s} \sum_{i=1}^{N} w_i y_i \]

The deviance ratio \( D^2 \) is given by

\[ D^2 = \frac{D_{\text{null}} - D}{D_{\text{null}}} \]

where \( D_{\text{null}} \) is the deviance calculated when only a constant term is included in the model and \( D \) is the deviance of the full model.

The formulas for the deviance and for \( D_{\text{null}} \) vary by model.

For logit, the deviance and the \( D_{\text{null}} \) are

\[ D = -\frac{2}{N_s} \sum_{i=1}^{N} w_i \left[ \tilde{y}_i xb_i + \ln \left\{ 1 + \exp(xb_i) \right\} \right] \]

\[ D_{\text{null}} = -\frac{2}{N_s} \sum_{i=1}^{N} w_i \left\{ \tilde{y}_i \ln \bar{y} + (1 - \tilde{y}_i) \ln(1 - \bar{y}) \right\} \]

\[ \tilde{y}_i = \begin{cases} 1 & y_i > 0 \\ 0 & \text{otherwise} \end{cases} \]

\[ \bar{y} = \frac{1}{N_s} \sum_{i=1}^{N} w_i \tilde{y}_i \]
For probit, the deviance and the $D_{\text{null}}$ are

$$D = -\frac{2}{N_s} \sum_{i=1}^{N} w_i \left[ \tilde{y}_i \ln \{ \Phi(xb_i) \} + (1 - \tilde{y}_i) \ln \{1 - \Phi(xb_i)\} \right]$$

$$D_{\text{null}} = -\frac{2}{N_s} \sum_{i=1}^{N} w_i \left\{ \tilde{y}_i \ln \bar{y} + (1 - \tilde{y}_i) \ln (1 - \bar{y}) \right\}$$

$$\tilde{y}_i = \begin{cases} 1 & y_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\bar{y} = \frac{1}{N_s} \sum_{i=1}^{N} s_i w_i \tilde{y}_i$$

For poisson, the deviance and the $D_{\text{null}}$ are

$$D = -\frac{2}{N_s} \sum_{i=1}^{N} w_i \left\{ y_i xb_i - \exp(xb_i) - v_i \right\}$$

$$v_i = \begin{cases} 0 & \text{if } y_i = 0 \\ y_i \ln y_i - y_i & \text{otherwise} \end{cases}$$

$$D_{\text{null}} = -\frac{2}{N_s} \sum_{i=1}^{N} w_i (y_i \ln \bar{y} - \bar{y} - v_i)$$

$$\bar{y} = \frac{1}{N_s} \sum_{i=1}^{N} w_i y_i$$

References

Also see

[**LASSO**] lasso — Lasso for prediction and model selection

[**LASSO**] lassoknots — Display knot table after lasso estimation

[**LASSO**] lasso postestimation — Postestimation tools for lasso for prediction