Title

poregress — Partialing-out lasso linear regression

Description

poregress fits a lasso linear regression model and reports coefficients along with standard errors, test statistics, and confidence intervals for specified covariates of interest. The partialing-out method is used to estimate effects for these variables and to select from potential control variables to be included in the model.

Quick start

Estimate a coefficient for d1 in a linear regression of y on d1, and include x1–x100 as potential control variables to be selected by lassos

```
poregress y d1, controls(x1-x100)
```

As above, and estimate coefficients for the levels of categorical d2

```
poregress y d1 i.d2, controls(x1-x100)
```

Use cross-validation (CV) instead of a plugin iterative formula to select the optimal λ^* in each lasso

```
poregress y d1 i.d2, controls(x1-x100) selection(cv)
```

As above, and set a random-number seed for reproducibility

```
poregress y d1 i.d2, controls(x1-x100) selection(cv) rseed(28)
```

Specify CV for the lasso for y only, with the stopping rule criterion turned off

```
poregress y d1 i.d2, controls(x1-x100) lasso(y, selection(cv), stop(0))
```

As above, but apply the option to the lassos for y, d1, and i.d2

```
poregress y d1 i.d2, controls(x1-x100) lasso(*, selection(cv), stop(0))
```

Compute lassos beyond the CV minimum to get full coefficient paths, knots, etc.

```
poregress y d1 i.d2, controls(x1-x100) ///
     lasso(*, selection(cv, alllambdas))
```

Menu

Statistics > Lasso > Lasso inferential models > Continuous outcomes > Partialing-out model

Syntax

`poregress` *depvar varsofinterest* [*if*] [*in*] ,

`controls(`[(*alwaysvars*)] *othervars*`)` [*options*]

varsofinterest are variables for which coefficients and their standard errors are estimated.

options	Description
Model	
* `controls(`[(*alwaysvars*)] *othervars*`)`	*alwaysvars* and *othervars* make up the set of control variables; *alwaysvars* are always included; lassos choose whether to include or exclude *othervars*
`selection(plugin)`	use a plugin iterative formula to select an optimal value of the lasso penalty parameter λ^* for each lasso; the default
`selection(cv)`	use CV to select an optimal value of the lasso penalty parameter λ^* for each lasso
`selection(adaptive)`	use adaptive lasso to select an optimal value of the lasso penalty parameter λ^* for each lasso
`selection(bic)`	use BIC to select an optimal value of the lasso penalty parameter λ^* for each lasso
`sqrtlasso`	use square-root lassos
`semi`	use semipartialing-out lasso regression estimator
`missingok`	after fitting lassos, ignore missing values in any *othervars* not selected, and include these observations in the final model
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `robust` (the default) or `cluster` *clustvar*
Reporting	
`level(`*#*`)`	set confidence level; default is `level(95)`
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
[`no`]`log`	display or suppress an iteration log
`verbose`	display a verbose iteration log
`rseed(`*#*`)`	set random-number seed
Advanced	
`lasso(`*varlist*, *lasso_options*`)`	specify options for the lassos for variables in *varlist*; may be repeated
`sqrtlasso(`*varlist*, *lasso_options*`)`	specify options for square-root lassos for variables in *varlist*; may be repeated
`reestimate`	refit the model after using `lassoselect` to select a different λ^*
`noheader`	do not display the header on the coefficient table
`coeflegend`	display legend instead of statistics

*`controls()` is required.

varsofinterest, *alwaysvars*, and *othervars* may contain factor variables. Base levels of factor variables cannot be set for *alwaysvars* and *othervars*. See [U] **11.4.3 Factor variables**.

`collect` is allowed; see [U] **11.1.10 Prefix commands**.

`reestimate`, `noheader`, and `coeflegend` do not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

`controls(`[`(`*alwaysvars*`)`] *othervars*`)` specifies the set of control variables, which control for omitted variables. Control variables are also known as confounding variables. `poregress` fits lassos for *depvar* and each of the *varsofinterest*. *alwaysvars* are variables that are always to be included in these lassos. *alwaysvars* are optional. *othervars* are variables that each lasso will choose to include or exclude. That is, each lasso will select a subset of *othervars*. The selected subset of *othervars* may differ across lassos. `controls()` is required.

`selection(plugin | cv | adaptive | bic)` specifies the selection method for choosing an optimal value of the lasso penalty parameter λ^* for each lasso or square-root lasso estimation. Separate lassos are estimated for *depvar* and each variable in *varsofinterest*. Specifying `selection()` changes the selection method for all of these lassos. You can specify different selection methods for different lassos using the option `lasso()` or `sqrtlasso()`. When `lasso()` or `sqrtlasso()` is used to specify a different selection method for the lassos of some variables, they override the global setting made using `selection()` for the specified variables.

`selection(plugin)` is the default. It selects λ^* based on a "plugin" iterative formula dependent on the data. See [LASSO] **lasso options**.

`selection(cv)` selects the λ^* that gives the minimum of the CV function. See [LASSO] **lasso options**.

`selection(adaptive)` selects λ^* using the adaptive lasso selection method. It cannot be specified when `sqrtlasso` is specified. See [LASSO] **lasso options**.

`selection(bic)` selects the λ^* that gives the minimum of the BIC function. See [LASSO] **lasso options**.

`sqrtlasso` specifies that square-root lassos be done rather than regular lassos. The option `lasso()` can be used with `sqrtlasso` to specify that regular lasso be done for some variables, overriding the global `sqrtlasso` setting for these variables. See [LASSO] **lasso options**.

`semi` specifies that the semipartialing-out lasso regression estimator be used instead of the fully partialing-out lasso estimator, which is the default. See *Methods and formulas* in [LASSO] **poregress**.

`missingok` specifies that, after fitting lassos, the estimation sample be redefined based on only the nonmissing observations of variables in the final model. In all cases, any observation with missing values for *depvar*, *varsofinterest*, *alwaysvars*, and *othervars* is omitted from the estimation sample for the lassos. By default, the same sample is used for calculation of the coefficients of the *varsofinterest* and their standard errors.

When `missingok` is specified, the initial estimation sample is the same as the default, but the sample used for the calculation of the coefficients of the *varsofinterest* can be larger. Now observations with missing values for any *othervars* not selected will be added to the estimation sample (provided there are no missing values for any of the variables in the final model).

missingok may produce more efficient estimates when data are missing completely at random. It does, however, have the consequence that estimation samples can change when selected variables differ in models fit using different selection methods. That is, when *othervars* contain missing values, the estimation sample for a model fit using the default selection(plugin) will likely differ from the estimation sample for a model fit using, for example, selection(cv).

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (robust) and that allow for intragroup correlation (cluster *clustvar*); see [R] ***vce_option***.

When vce(cluster *clustvar*) is specifed, all lassos also account for clustering. For each lasso, this affects how the log-likelihood function is computed and how the the sample is split in cross-validation; see *Methods and formulas* in [LASSO] **lasso**. Specifying vce(cluster *clustvar*) may lead to different selected controls and therefore to different point estimates for your variable of interest when compared to the estimation that ignores clustering.

Reporting

level(*#*); see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(*#*), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Optimization

[no]log displays or suppresses a log showing the progress of the estimation. By default, one-line messages indicating when each lasso estimation begins are shown. Specify verbose to see a more detailed log.

verbose displays a verbose log showing the iterations of each lasso estimation. This option is useful when doing selection(cv) or selection(adaptive). It allows you to monitor the progress of the lasso estimations for these selection methods, which can be time consuming when there are many *othervars* specified in controls().

rseed(*#*) sets the random-number seed. This option can be used to reproduce results for selection(cv) and selection(adaptive). The default selection method selection(plugin) does not use random numbers. rseed(*#*) is equivalent to typing set seed *#* prior to running poregress. See [R] **set seed**.

Advanced

lasso(*varlist*, *lasso_options*) lets you set different options for different lassos, or advanced options for all lassos. You specify a *varlist* followed by the options you want to apply to the lassos for these variables. *varlist* consists of one or more variables from *depvar* or *varsofinterest*. _all or * may be used to specify *depvar* and all *varsofinterest*. This option is repeatable as long as different variables are given in each specification. *lasso_options* are selection(...), grid(...), stop(*#*), tolerance(*#*), dtolerance(*#*), and cvtolerance(*#*). When lasso(*varlist*, selection(...)) is specified, it overrides any global selection() option for the variables in *varlist*. It also overrides the global sqrtlasso option for these variables. See [LASSO] **lasso options**.

sqrtlasso(*varlist*, *lasso_options*) works like the option lasso(), except square-root lassos for the variables in *varlist* are done rather than regular lassos. *varlist* consists of one or more variables from *depvar* or *varsofinterest*. This option is repeatable as long as different variables are given in each specification. *lasso_options* are selection(...), grid(...), stop(#), tolerance(#), dtolerance(#), and cvtolerance(#). When sqrtlasso(*varlist*, selection(...)) is specified, it overrides any global selection() option for the variables in *varlist*. See [LASSO] **lasso options**.

The following options are available with poregress but are not shown in the dialog box:

reestimate is an advanced option that refits the poregress model based on changes made to the underlying lassos using lassoselect. After running poregress, you can select a different λ^* for one or more of the lassos estimated by poregress. After selecting λ^*, you type poregress, reestimate to refit the poregress model based on the newly selected λ's.

reestimate may be combined only with reporting options.

noheader prevents the coefficient table header from being displayed.

coeflegend; see [R] **Estimation options**.

Remarks and examples

stata.com

poregress performs partialing-out lasso linear regression. This command estimates coefficients, standard errors, and confidence intervals and performs tests for variables of interest while using lassos to select from among potential control variables.

The linear regression model is

$$\mathbf{E}[y|\mathbf{d}, \mathbf{x}] = \mathbf{d}\boldsymbol{\alpha}' + \mathbf{x}\boldsymbol{\beta}'$$

where $\mathbf{d}$ are the variables for which we wish to make inferences and $\mathbf{x}$ are the potential control variables from which the lassos select. poregress reports estimated coefficients for $\boldsymbol{\alpha}$. However, partialing-out does not provide estimates of the coefficients on the control variables ($\boldsymbol{\beta}$) or their standard errors. No estimation results can be reported for $\boldsymbol{\beta}$.

For an introduction to the partialing-out lasso method for inference, as well as the double-selection and cross-fit partialing-out methods, see [LASSO] **Lasso inference intro**.

Examples that demonstrate how to use poregress and the other lasso inference commands are presented in [LASSO] **Inference examples**. In particular, we recommend reading *1 Overview* for an introduction to the examples and to the vl command, which provides tools for working with the large lists of variables that are often included when using lasso methods. See *2 Fitting and interpreting inferential models* for examples of fitting inferential lasso linear models and comparisons of the different methods available in Stata.

If you are interested in digging deeper into the lassos that are used to select controls, see *5 Exploring inferential model lassos* in [LASSO] **Inference examples**.

Stored results

poregress stores the following in e():

Scalars	
e(N)	number of observations
e(N_clust)	number of clusters
e(k_varsofinterest)	number of variables of interest
e(k_controls)	number of potential control variables
e(k_controls_sel)	number of selected control variables
e(df)	degrees of freedom for test of variables of interest
e(chi2)	χ^2
e(p)	p-value for test of variables of interest
e(rank)	rank of e(V)
Macros	
e(cmd)	poregress
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(lasso_depvars)	names of dependent variables for all lassos
e(varsofinterest)	variables of interest
e(controls)	potential control variables
e(controls_sel)	selected control variables
e(model)	linear
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(chi2type)	Wald; type of χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(rngstate)	random-number state used
e(properties)	b V
e(predict)	program used to implement predict
e(select_cmd)	program used to implement lassoselect
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved
Matrices	
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

poregress implements two methods for the partialing-out lasso regression. We call the default method partialing-out lasso regression (POLR). We call the optional method, obtained by specifying option semi, a semipartialing-out lasso regression (SPOLR). We refer to these methods as versions of partialing-out regression because they reduce to the classic method of partialing-out regression in a special case discussed below.

The POLR was derived by Belloni et al. (2012) and Chernozhukov, Hansen, and Spindler (2015a, 2015b). The SPOLR was derived by Belloni et al. (2012), Belloni, Chernozhukov, and Hansen (2014), Belloni, Chernozhukov, and Kato (2015), and Belloni, Chernozhukov, and Wei (2016).

The authors referred to their methods as "instrumental-variable methods". We refer to these methods as partialing-out regression methods because the idea of partialing-out regression is more cross-disciplinary and because these methods do not need outside instrumental variables when the covariates are exogenous.

Mechanically, the POLR and the SPOLR methods are method of moments estimators in which the moment conditions are the score equations from an ordinary least-squares (OLS) estimator of a partial outcome on one or more partial covariates. The partial outcome is the residual from a regression of the outcome on the controls selected by a lasso. Each of the partial covariates is a residual from a regression of the covariate on the controls selected by a lasso.

The POLR method is limited to a linear model for the outcome. This method follows from Chernozhukov, Hansen, and Spindler (2015a; 2015b, sec. 5) and Chernozhukov et al. (2018, C18). The algorithm described in Chernozhukov, Hansen, and Spindler (2015a, 2015b) is for endogenous variables with many outside instruments and many controls. As they note, imposing an exogeneity assumption and assuming that there are no outside instruments reduces their algorithm to one for exogenous covariates with many controls.

The SPOLR method extends naturally to nonlinear models for the outcome and has two sources. It is implied by Belloni, Chernozhukov, and Kato (2015, algorithm 1), which is for a median regression of y on $\mathbf{x}$. Replacing median regression with mean regression in their step (i) and replacing the median moment condition with the mean moment condition in step (iii) produces the SPOLR algorithm detailed below. This algorithm is also implied by Belloni, Chernozhukov, and Wei (2016, table 1 and sec. 2.1) for a linear model.

The regression model is

$$\mathbf{E}[y|\mathbf{d}, \mathbf{x}] = \mathbf{d}\boldsymbol{\alpha}' + \beta_0 + \mathbf{x}\boldsymbol{\beta}'$$

where $\mathbf{d}$ contains the J covariates of interest and $\mathbf{x}$ contains the p controls. The number of covariates in $\mathbf{d}$ must be small and fixed. The number of controls in $\mathbf{x}$ can be large and, in theory, can grow with the sample size; however, the number of nonzero elements in $\boldsymbol{\beta}$ must not be too large, which is to say that the model must be sparse.

POLR algorithm

1. For $j = 1, \dots, J$, perform a linear lasso of d_j on $\mathbf{x}$, and denote the selected controls by $\widetilde{\mathbf{x}}_j$.

   Each of these lassos can choose the lasso penalty parameter (λ_j^*) using the plugin estimator, adaptive lasso, or CV. The heteroskedastic plugin estimator for the linear lasso is the default.

2. For $j = 1, \dots, J$, fit a linear regression of d_j on $\widetilde{\mathbf{x}}_j$, denote the estimated coefficients by $\widehat{\boldsymbol{\gamma}}_j$, and define the partial-covariate variable $z_j = d_j - \widetilde{\mathbf{x}}_j\widehat{\boldsymbol{\gamma}}_j$, with its ith observation denoted by $z_{j,i}$.

   Collect the J partial covariates for the ith observation into the vector $\mathbf{z}_i = (z_{1,i}, \dots, z_{J,i})$.

3. Perform a linear lasso of y on $\mathbf{x}$ to select covariates $\widetilde{\mathbf{x}}_y$.

   This lasso can choose the lasso penalty parameter (λ^*) using the plugin estimator, adaptive lasso, or CV. The heteroskedastic plugin estimator for the linear lasso is the default.

4. Fit a linear regression of y on $\widetilde{\mathbf{x}}_y$, denote the estimated coefficients by $\widetilde{\boldsymbol{\beta}}_y$, and define the partial outcome for the ith observation by $\widetilde{y}_i = y_i - \widetilde{\mathbf{x}}_{y,i}\widetilde{\boldsymbol{\beta}}_y$.

5. Compute $\widehat{\alpha}$ by solving the following J sample-moment equations.

$$\frac{1}{n}\sum_{i=1}^{n}(\widetilde{y}_i - \mathbf{z}_i\boldsymbol{\alpha}')\mathbf{z}_i' = \mathbf{0}$$

The VCE is estimated by the robust estimator for method of moments.

SPOLR algorithm

1. For $j = 1, \ldots, J$, perform a linear lasso of d_j on $\mathbf{x}$, and denote the selected controls by $\widetilde{\mathbf{x}}_j$.

   Each of these lassos can choose the lasso penalty parameter (λ_j^*) using the plugin estimator, adaptive lasso, or CV. The heteroskedastic plugin estimator for the linear lasso is the default.
2. For $j = 1, \ldots, J$, fit a linear regression of d_j on $\widetilde{\mathbf{x}}_j$, denote the estimated coefficients by $\widehat{\boldsymbol{\gamma}}_j$, and define the partial-covariate variable $z_j = d_j - \widetilde{\mathbf{x}}_j\widehat{\boldsymbol{\gamma}}_j$, with its ith observation denoted by $z_{j,i}$.

   Collect the J partial covariates for the ith observation into the vector $\mathbf{z}_i = (z_{1,i}, \ldots, z_{J,i})$.
3. Perform a linear lasso of y on $\mathbf{d}$ and $\mathbf{x}$ to select covariates $\check{\mathbf{x}}_y$.

   This lasso can choose the lasso penalty parameter (λ^*) using the plugin estimator, adaptive lasso, or CV. The heteroskedastic plugin estimator for the linear lasso is the default.
4. Fit a linear regression of y on $\mathbf{d}$ and $\check{\mathbf{x}}_y$, let $\check{\boldsymbol{\beta}}$ be the estimated coefficients on $\check{\mathbf{x}}_y$, and define the partial outcome for the ith observation by $\check{y}_i = y_i - \check{\mathbf{x}}_{y,i}\check{\boldsymbol{\beta}}'$.
5. Compute $\widehat{\alpha}$ by solving the following J sample-moment equations.

$$\frac{1}{n}\sum_{i=1}^{n}(\check{y}_i - \mathbf{d}_i\boldsymbol{\alpha}')\mathbf{z}_i' = \mathbf{0}$$

   The VCE is estimated by the robust estimator for method of moments.

See *Methods and formulas* in [LASSO] **lasso** for details on how the lassos in steps 1 and 3 of both algorithms choose their penalty parameter (λ^*).

References

Belloni, A., D. Chen, V. Chernozhukov, and C. B. Hansen. 2012. Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica* 80: 2369–2429. https://doi.org/10.3982/ECTA9626.

Belloni, A., V. Chernozhukov, and C. B. Hansen. 2014. Inference on treatment effects after selection among high-dimensional controls. *Review of Economic Studies* 81: 608–650. https://doi.org/10.1093/restud/rdt044.

Belloni, A., V. Chernozhukov, and K. Kato. 2015. Uniform post-selection inference for least absolute deviation regression and other Z-estimation problems. *Biometrika* 102: 77–94. https://doi.org/10.1093/biomet/asu056.

Belloni, A., V. Chernozhukov, and Y. Wei. 2016. Post-selection inference for generalized linear models with many controls. *Journal of Business & Economic Statistics* 34: 606–619. https://doi.org/10.1080/07350015.2016.1166116.

Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. B. Hansen, W. K. Newey, and J. M. Robins. 2018. Double/debiased machine learning for treatment and structural parameters. *Econometrics Journal* 21: C1–C68. https://doi.org/10.1111/ectj.12097.

Chernozhukov, V., C. B. Hansen, and M. Spindler. 2015a. Post-selection and post-regularization inference in linear models with many controls and instruments. *American Economic Review* 105: 486–490. https://doi.org/10.1257/aer.p20151022.

——. 2015b. Valid post-selection and post-regularization inference: An elementary, general approach. *Annual Review of Economics* 7: 649–688. https://doi.org/10.1146/annurev-economics-012315-015826.

Also see

[LASSO] **lasso inference postestimation** — Postestimation tools for lasso inferential models

[LASSO] **dsregress** — Double-selection lasso linear regression

[LASSO] **xporegress** — Cross-fit partialing-out lasso linear regression

[R] **regress** — Linear regression

[U] 20 Estimation and postestimation commands