Title

stata.com

dslogit — Double-selection lasso logistic regression

Description

`dslogit` fits a lasso logistic regression model and reports odds ratios along with standard errors, test statistics, and confidence intervals for specified covariates of interest. The double-selection method is used to estimate effects for these variables and to select from potential control variables to be included in the model.

Quick start

Report an odds ratio from a logistic regression of `y` on `d1`, and include `x1` to `x100` as potential control variables to be selected by lassos

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

Same as above, and estimate odds ratios for the levels of categorical `d2`

```
dslogit 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

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

Same as above, and set a random-number seed for reproducibility

```
dslogit 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

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

Same as above, but apply the option to the lassos for `y`, `d1`, and `i.d2`

```
dslogit 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.

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

Menu

Statistics > Lasso > Lasso inferential models > Binary outcomes > Double-selection logit model

Syntax

```
dslogit 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 for *varsofinterest*
`missingok`	after fitting lassos, ignore missing values in any *othervars* not selected, and include these observations in the final model
`offset(`*varname*`)`	include *varname* in the lasso and model for *depvar* with its coefficient constrained to be 1
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `robust` (the default), `cluster` *clustvar*, or `oim`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`or`	report odds ratios; the default
`coef`	report estimated coefficients
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. `dslogit` 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 for the *varsofinterest*. This option does not apply to *depvar*. Square-root lassos are linear models, and the lasso for *depvar* is always a logit lasso. 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**.

`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).

offset(*varname*) specifies that *varname* be included in the lasso and model for *depvar* with its coefficient constrained to be 1.

SE/Robust

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

When vce(cluster *clustvar*) is specified, all lassos also account for clustering. For each lasso, this affects how the log-likelihood function is computed and how 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**.

or reports the estimated coefficients transformed to odds ratios, that is, e^{α}. Standard errors and confidence intervals are similarly transformed. or is the default.

coef reports the estimated coefficients α rather than the odds ratios (e^{α}). This option affects how results are displayed, not how they are estimated. coef may be specified at estimation or when replaying previously estimated results.

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 dslogit. 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 *varsofinterest*. Square-root lassos are linear models, and this option cannot be used with *depvar*. 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 `dslogit` but are not shown in the dialog box:

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

`dslogit` performs double-selection lasso logistic regression. This command estimates odds ratios, standard errors, and confidence intervals and performs tests for variables of interest while using lassos to select from among potential control variables.

The logistic regression model is

$$\Pr(y = 1|\mathbf{d}, \mathbf{x}) = \frac{\exp(\mathbf{d}\boldsymbol{\alpha}' + \mathbf{x}\boldsymbol{\beta}')}{1 + \exp(\mathbf{d}\boldsymbol{\alpha}' + \mathbf{x}\boldsymbol{\beta}')}$$

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. `dslogit` estimates the $\boldsymbol{\alpha}$ coefficients and reports the corresponding odds ratios, e^{α}. However, double selection 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 double-selection lasso method for inference, as well as the partialing-out and cross-fit partialing-out methods, see [LASSO] **Lasso inference intro**.

Examples that demonstrate how to use `dslogit` 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 lassos methods. See *2 Fitting and interpreting inferential models* for comparisons of the different methods of fitting inferential models that are

available in Stata. Everything we say there about methods of selection is applicable to both linear and nonlinear models. See *3 Fitting logit inferential models to binary outcomes. What is different?* for examples and discussion specific to logistic regression models. The primary difference from linear models involves interpreting the results.

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

`dslogit` 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)`	`dslogit`
`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)`	`logit`
`e(title)`	title in estimation output
`e(offset)`	linear offset variable
`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

`dslogit` implements double-selection lasso logit regression (DSLLR) as described in Belloni, Chernozhukov, and Wei (2016, table 2 and sec. 2.1). The regression model is

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

where $G(a) = \exp(a)/\{1 + \exp(a)\}$, $\mathbf{d}$ 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.

DSLLR algorithm

1. Perform a logit lasso of y on $\mathbf{d}$ and $\mathbf{x}$, and denote the selected controls by $\widetilde{\mathbf{x}}$.

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

2. Fit a logit regression of y on $\mathbf{d}$ and $\widetilde{\mathbf{x}}$, denoting the estimated coefficient vectors by $\widetilde{\boldsymbol{\alpha}}$ and $\widetilde{\boldsymbol{\beta}}$, respectively.

3. Let $w_i = G'(\mathbf{d}_i\widetilde{\boldsymbol{\alpha}}' + \widetilde{\mathbf{x}}_i\widetilde{\boldsymbol{\beta}}')$ be the ith observation of the predicted value of the derivative of $G(\cdot)$.

4. For $j = 1, \ldots, J$, perform a linear lasso of d_j on $\mathbf{x}$ using observation-level weights w_i, and denote the selected controls by $\check{\mathbf{x}}_j$.

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

5. Let $\widehat{\mathbf{x}}$ be the distinct variables from the union of the variables in $\check{\mathbf{x}}_1, \ldots, \check{\mathbf{x}}_J$, and $\widetilde{\mathbf{x}}$.

6. Fit a logit regression of y on $\mathbf{d}$ and $\widehat{\mathbf{x}}$, denoting the estimated coefficient vectors by $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\beta}}$, respectively.

7. Store the point estimates $\widehat{\boldsymbol{\alpha}}$ in `e(b)` and their variance estimates (VCE) in `e(V)`.

   Option `vce(robust)`, the robust estimator of the VCE for a logistic regression, is the default. Specify option `vce(oim)` to get the OIM estimator of the VCE.

See *Methods and formulas* in [LASSO] **lasso** for details on how the lassos in steps 1 and 4 choose their penalty parameter (λ^*).

Reference

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

Also see

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

[LASSO] **pologit** — Partialing-out lasso logistic regression

[LASSO] **xpologit** — Cross-fit partialing-out lasso logistic regression

[R] **logit** — Logistic regression, reporting coefficients

[R] **logistic** — Logistic regression, reporting odds ratios

[U] **20 Estimation and postestimation commands**