Prediction, model selection, and causal inference with regularized regression

Introducing two Stata packages: LASSOPACK and PDSLASSO

Achim Ahrens (ESRI, Dublin),

Mark E Schaffer (Heriot-Watt University, CEPR & IZA),

with Christian B Hansen (University of Chicago)

https://statalasso.github.io/

London Stata Conference 2018
at the Cass Business School,
6-7 September 2018.

Background

The on-going revolution in data science and machine learning (ML) has not gone unnoticed in economics & social science.

See surveys by Mullainathan and Spiess, 2017; Athey, 2017; Varian, 2014.

Background

The on-going revolution in data science and machine learning (ML) has not gone unnoticed in economics & social science.

See surveys by Mullainathan and Spiess, 2017; Athey, 2017; Varian, 2014.

(Supervised) Machine learning

- Focus on prediction & classification.
- Wide set of methods: support vector machines, random forests, neural networks, penalized regression, etc.
- *Typical problems:* predict user-rating of films (Netflix), classify email as spam or not, Genome-wide association studies

Background

The on-going revolution in data science and machine learning (ML) has not gone unnoticed in economics & social science.

See surveys by Mullainathan and Spiess, 2017; Athey, 2017; Varian, 2014.

(Supervised) Machine learning

- Focus on prediction & classification.
- Wide set of methods: support vector machines, random forests, neural networks, penalized regression, etc.
- *Typical problems:* predict user-rating of films (Netflix), classify email as spam or not, Genome-wide association studies

Econometrics and allied fields

- Focus on causal inference using OLS, IV/GMM, Maximum Likelihood.
- *Typical question:* Does x have a causal effect on y?

Background

The on-going revolution in data science and machine learning (ML) has not gone unnoticed in economics & social science.

See surveys by Mullainathan and Spiess, 2017; Athey, 2017; Varian, 2014.

(Supervised) Machine learning

- Focus on prediction & classification.
- Wide set of methods: support vector machines, random forests, neural networks, penalized regression, etc.
- *Typical problems:* predict user-rating of films (Netflix), classify email as spam or not, Genome-wide association studies

Econometrics and allied fields

- Focus on causal inference using OLS, IV/GMM, Maximum Likelihood.
- *Typical question:* Does x have a causal effect on y?

Central question:

How can econometricians+allies learn from machine learning?

Motivation I: Model selection

The standard linear model

$$y_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \varepsilon_i.$$

Why would we use a fitting procedure other than OLS?

Motivation I: Model selection

The standard linear model

$$y_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \varepsilon_i.$$

Why would we use a fitting procedure other than OLS?

Model selection.

We don't know the true model. Which regressors are important?

Including too many regressors leads to **overfitting**: good in-sample fit (high R^2), but bad *out-of-sample* prediction.

Including too few regressors leads to **omitted variable bias**.

Motivation I: Model selection

The standard linear model

$$y_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \varepsilon_i.$$

Why would we use a fitting procedure other than OLS?

Model selection.

Model selection becomes even more challenging when the data is high-dimensional.

If p is close to or larger than n, we say that the data is high-dimensional.

- If $p > n$, the model is not identified.
- If $p = n$, perfect fit. Meaningless.
- If $p < n$ but large, overfitting is likely: Some of the predictors are only significant by chance (false positives), but perform poorly on new (unseen) data.

Motivation I: Model selection

The standard linear model

$$y_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \varepsilon_i.$$

Why would we use a fitting procedure other than OLS?

High-dimensional data.

Large p is often not acknowledged in applied work:

- The true model is unknown *ex ante*. Unless a researcher runs one and only one specification, the low-dimensional model paradigm is likely to fail.
- The number of regressors increases if we account for non-linearity, interaction effects, parameter heterogeneity, spatial & temporal effects.

Example: Cross-country regressions, where we have only small number of countries, but thousands of macro variables.

Motivation I: Model selection

The standard approach for model selection in econometrics is (arguably) hypothesis testing.

Problems:

- Pre-test biases in multi-step procedures. This also applies to model building using, e.g., the *general-to-specific* approach (Dave Giles).
- Especially if p is large, inference is problematic. Need for false discovery control (multiple testing procedures)—rarely done.
- 'Researcher degrees of freedom' and 'p-hacking': researchers try many combinations of regressors, looking for statistical significance (Simmons et al., 2011).

Motivation I: Model selection

The standard approach for model selection in econometrics is (arguably) hypothesis testing.

Problems:

- Pre-test biases in multi-step procedures. This also applies to model building using, e.g., the *general-to-specific* approach (Dave Giles).
- Especially if p is large, inference is problematic. Need for false discovery control (multiple testing procedures)—rarely done.
- 'Researcher degrees of freedom' and 'p-hacking': researchers try many combinations of regressors, looking for statistical significance (Simmons et al., 2011).

Researcher degrees of freedom

"it is common (and accepted practice) for researchers to explore various analytic alternatives, to search for a combination that yields 'statistical significance,' and to then report only what 'worked."' Simmons et al., 2011

Motivation II: Prediction

The standard linear model

$$y_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \varepsilon_i.$$

Why would we use a fitting procedure other than OLS?

Bias-variance-tradeoff.

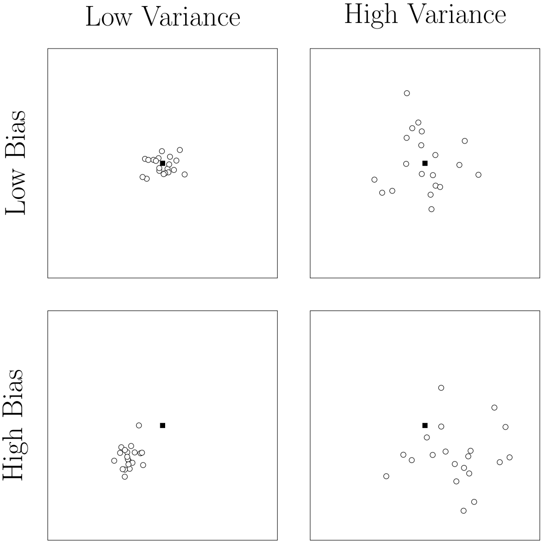
OLS estimator has zero bias, but not necessarily the best *out-of-sample* predictive accuracy.

Suppose we fit the model using the data $i = 1, \ldots, n$. The prediction error for y_0 given x_0 can be decomposed into

$$PE_0 = E[(y_0 - \hat{y}_0)^2] = \sigma_\varepsilon^2 + Bias(\hat{y}_0)^2 + Var(\hat{y}_0).$$

In order to minimize the expected prediction error, we need to select low variance and low bias, but not necessarily zero bias!

Motivation II: Prediction



The squared points ('■') indicate the true value and round points ('○') represent estimates. The diagrams illustrate that a high bias/low variance estimator may yield predictions that are on average closer to the truth than predictions from a low bias/high variance estimator.

Motivation II: Prediction

There are cases where ML methods can be applied ‘off-the-shelf’ to policy questions.

Kleinberg et al. (2015) and Athey (2017) provide some examples:

- Predict patient’s life expectancy to decide whether hip replacement surgery is beneficial.
- Predict whether accused would show up for trial to decide who can be let out of prison while awaiting trial.
- Predict loan repayment probability.

Motivation II: Prediction

There are cases where ML methods can be applied ‘off-the-shelf’ to policy questions.

Kleinberg et al. (2015) and Athey (2017) provide some examples:

- Predict patient’s life expectancy to decide whether hip replacement surgery is beneficial.
- Predict whether accused would show up for trial to decide who can be let out of prison while awaiting trial.
- Predict loan repayment probability.

But: in other cases, ML methods are not directly applicable for research questions in econometrics and allied fields, especially when it comes to causal inference.

Motivation III: Causal inference

Machine learning offers a set of methods that **outperform OLS** in terms of *out-of-sample* prediction.

But economists are in general more interested in **causal inference**.

Recent theoretical work by Belloni, Chernozhukov, Hansen and their collaborators has shown that these methods can also be used in estimation of structural models.

Motivation III: Causal inference

Machine learning offers a set of methods that **outperform OLS** in terms of *out-of-sample* prediction.

But economists are in general more interested in **causal inference**.

Recent theoretical work by Belloni, Chernozhukov, Hansen and their collaborators has shown that these methods can also be used in estimation of structural models.

Two very common problems in applied work:

- **Selecting controls** to address omitted variable bias when many potential controls are available
- **Selecting instruments** when many potential instruments are available.

Background

Today, we introduce two Stata packages:

`LASSOPACK` (including `lasso2`, `cvlasso` & `rlasso`)

- implements penalized regression methods: LASSO, elastic net, ridge, square-root LASSO, adaptive LASSO.
- uses fast path-wise coordinate descent algorithms (Friedman et al., 2007).
- three commands for three different penalization approaches: cross-validation (`cvlasso`), information criteria (`lasso2`) and 'rigorous' (theory-driven) penalization (`rlasso`).
- focus is on **prediction & model selection**.

`PDSLASSO` (including `pdslasso` and `ivlasso`):

- relies on the estimators implemented in `LASSOPACK`
- intended for **estimation of structural models**.
- allows for many controls and/or many instruments.

High-dimensional data

The general model is:

$$y_i = \boldsymbol{x}_i' \boldsymbol{\beta} + \varepsilon_i$$

We index observations by i and regressors by j. We have up to $p = \dim(\beta)$ potential regressors. p can be very large, potentially even larger than the number of observations n.

High-dimensional data

The general model is:

$$y_i = \boldsymbol{x}_i' \boldsymbol{\beta} + \varepsilon_i$$

We index observations by i and regressors by j. We have up to $p = \dim(\boldsymbol{\beta})$ potential regressors. p can be very large, potentially even larger than the number of observations n.

The high-dimensional model accommodates situations where we only observe a few explanatory variables, but the number of potential regressors is large when accounting for model uncertainty, non-linearity, temporal & spatial effects, etc.

High-dimensional data

The general model is:

$$y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + \varepsilon_i$$

We index observations by i and regressors by j. We have up to $p = \dim(\boldsymbol{\beta})$ potential regressors. p can be very large, potentially even larger than the number of observations n.

The high-dimensional model accommodates situations where we only observe a few explanatory variables, but the number of potential regressors is large when accounting for model uncertainty, non-linearity, temporal & spatial effects, etc.

OLS leads to disaster: If p is large, we overfit badly and classical hypothesis testing leads to many false positives. If $p > n$, OLS is not identified.

High-dimensional data

The general model is:

$$y_i = \boldsymbol{x}_i' \boldsymbol{\beta} + \varepsilon_i$$

This becomes manageable if we assume **(exact) sparsity**: of the p potential regressors, **only s regressors belong in the model**, where

$$s := \sum_{j=1}^{p} \mathbb{1}\{\beta_j \neq 0\} \ll n.$$

In other words: most of the true coefficients β_j are actually zero. But we don't know which ones are zeros and which ones aren't.

High-dimensional data

The general model is:

$$y_i = \boldsymbol{x}_i' \boldsymbol{\beta} + \varepsilon_i$$

This becomes manageable if we assume **(exact) sparsity**: of the p potential regressors, **only *s* regressors belong in the model**, where

$$s := \sum_{j=1}^{p} \mathbb{1}\{\beta_j \neq 0\} \ll n.$$

In other words: most of the true coefficients β_j are actually zero. But we don't know which ones are zeros and which ones aren't.

We can also use the weaker assumption of **approximate sparsity**: some of the β_j coefficients are well-approximated by zero, and the approximation error is sufficiently 'small'.

The LASSO

The LASSO (Least Absolute Shrinkage and Selection Operator, Tibshirani, 1996), "ℓ_1 norm".

Minimize: $$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \boldsymbol{x}_i'\boldsymbol{\beta}\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

There's a cost to including lots of regressors, and we can reduce the objective function by throwing out the ones that contribute little to the fit.

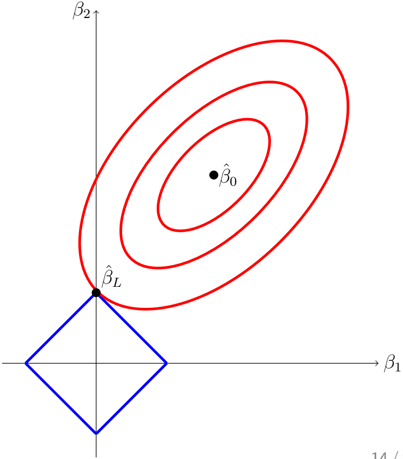
The effect of the penalization is that LASSO sets the $\hat{\beta}_j$s for some variables to zero. In other words, it does the **model selection** for us.

In contrast to ℓ_0-norm penalization (AIC, BIC) computationally feasible. Path-wise coordinate descent ('shooting') algorithm allows for fast estimation.

The LASSO

The LASSO estimator can also be written as

$$\hat{\beta}_L = \arg\min \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i' \beta)^2 \quad \text{s.t.} \quad \sum_{j=1}^{p} |\beta_j| < \tau.$$



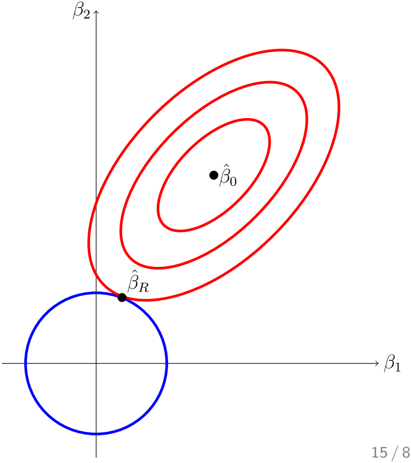
Example:

- $p = 2$.
- Blue diamond is the constraint region $|\beta_1| + |\beta_2| < \tau$.
- $\hat{\beta}_0$ is the OLS estimate.
- $\hat{\beta}_L$ is the LASSO estimate.
- Red lines are RSS contour lines.
- $\hat{\beta}_{1,L} = 0$ implying that the LASSO omits regressor 1 from the model.

LASSO vs Ridge

For comparison, the Ridge estimator is

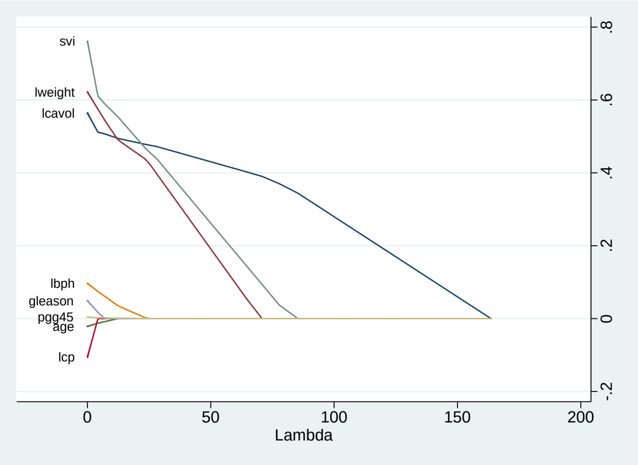
$$\hat{\boldsymbol{\beta}}_R = \arg\min \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i' \boldsymbol{\beta})^2 \quad \text{s.t.} \quad \sum_{j=1}^{p} {\beta_j}^2 < \tau.$$



Example:

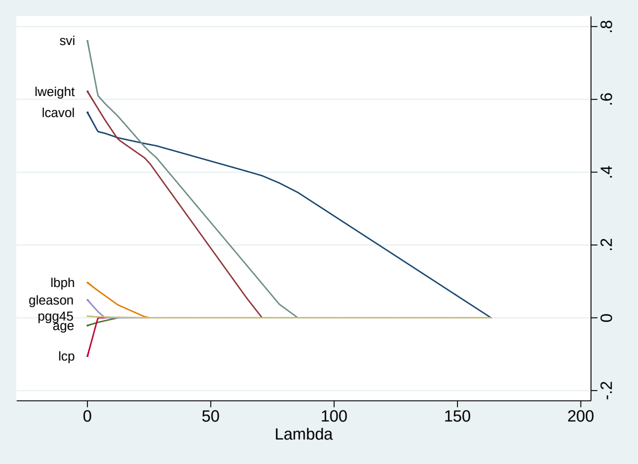
- $p = 2$.
- Blue circle is the constraint region ${\beta_1}^2 + {\beta_2}^2 < \tau$.
- $\hat{\beta}_0$ is the OLS estimate.
- $\hat{\beta}_R$ is the Ridge estimate.
- Red lines are RSS contour lines.
- $\hat{\beta}_{1,L} \neq 0$ and $\hat{\beta}_{2,L} \neq 0$. Both regressors are included.

The LASSO: The solution path



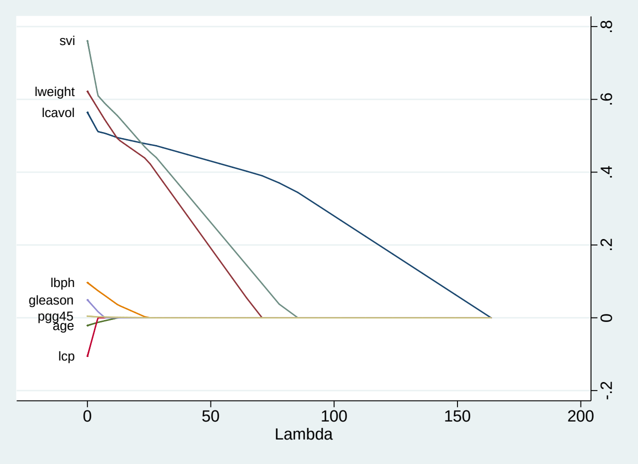
The LASSO coefficient path is a continuous and piecewise linear function of λ, with changes in slope where variables enter/leave the active set.

The LASSO: The solution path



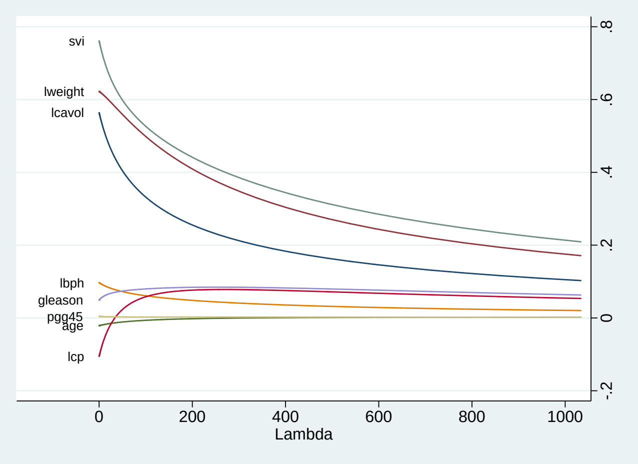
The LASSO yields sparse solutions. As λ increases, variables are being removed from the model. Thus, the LASSO can be used for model selection.

The LASSO: The solution path



We have reduced a complex model selection problem into a one-dimensional problem. We ‘only’ need to choose the ‘right’ penalty level, i.e., λ.

LASSO vs Ridge solution path



Ridge regression: No sparse solutions. The Ridge is not a model selection technique.

The LASSO: Choice of the penalty level

The penalization approach allows us to simplify the model selection problem to a one-dimensional problem.

But how do we select λ? — Three approaches:

- **Data-driven:** re-sample the data and find the λ that optimizes out-of-sample prediction. This approach is referred to as *cross-validation*.

  → Implemented in `cvlasso`.

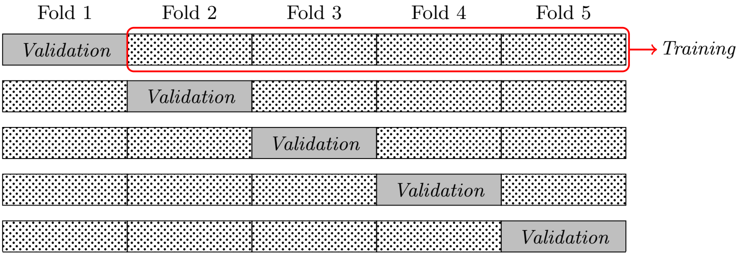
- **'Rigorous' penalization:** Belloni et al. (2012, *Econometrica*) develop theory and feasible algorithms for the optimal λ under heteroskedastic and non-Gaussian errors. Feasible algorithms are available for LASSO and square-root LASSO.

  → Implemented in `rlasso`.

- **Information criteria:** select the value of λ that minimizes information criterion (AIC, AICc, BIC or EBIC$_\gamma$).

  → Implemented in `lasso2`.

The LASSO: *K*-fold cross-validation



Fold 1 Fold 2 Fold 3 Fold 4 Fold 5

Validation → *Training*

Validation

Validation

Validation

Validation

Step 1 Divide data set into 5 groups (folds) of approximately equal size.

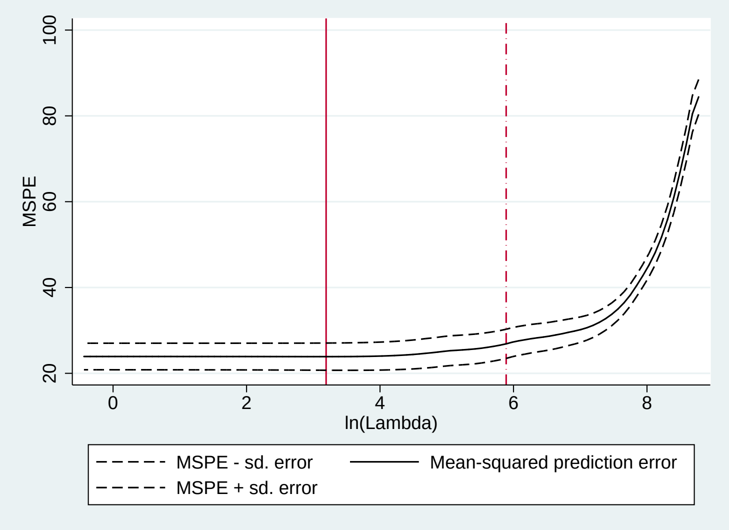
Step 2 Treat fold 1 as the validation data set. Fold 2-5 constitute the training data.

Step 3 Estimate the model using the training data. Assess predictive performance for a range of λ using the validation data.

Step 4 Repeat Step 2-3 using folds $2, \ldots, 5$ as validation data.

Step 5 Identify the λ that shows best out-of-sample predictive performance.

The LASSO: *K*-fold cross-validation



The solid vertical line corresponds to the lambda value that minimizes the mean-squared prediction error (λ_{LOPT}). The dashed line marks the largest lambda at which the MSPE is within one standard error of the minimal MSPE (λ_{LSE}).

The LASSO: *h*-step ahead cross-validation*

Cross-validation can also be applied in the time-series context.

Let T denote an observation in the training data set, and V an observation in the validation data set. ‘ . ’ indicates that an observation is not being used.

We can divide the data set as follows:

$$
\begin{array}{ccc}
 & & \textit{Step} \\
 & & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
t & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} &
\begin{bmatrix}
T & T & T & T & T \\
T & T & T & T & T \\
T & T & T & T & T \\
V & T & T & T & T \\
. & V & T & T & T \\
. & . & V & T & T \\
. & . & . & V & T \\
. & . & . & . & V
\end{bmatrix}
\end{array}
$$

1-step ahead cross-validation

See Hyndman, RJ, *Hyndsight blog*.

The LASSO: *h*-step ahead cross-validation*

Cross-validation can also be applied in the time-series context.

Let T denote an observation in the training data set, and V an observation in the validation data set. ' . ' indicates that an observation is not being used.

We can divide the data set as follows:

$$\begin{array}{ccc} & & \textit{Step} \\ & & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ t & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} & \begin{bmatrix} T & T & T & T & T \\ T & T & T & T & T \\ T & T & T & T & T \\ V & T & T & T & T \\ . & V & T & T & T \\ . & . & V & T & T \\ . & . & . & V & T \\ . & . & . & . & V \end{bmatrix} \end{array}$$

1-step ahead cross-validation

$$\begin{array}{ccc} & & \textit{Step} \\ & & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ t & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} & \begin{bmatrix} T & T & T & T & T \\ T & T & T & T & T \\ T & T & T & T & T \\ . & T & T & T & T \\ V & . & T & T & T \\ . & V & . & T & T \\ . & . & V & . & T \\ . & . & . & V & . \\ . & . & . & . & V \end{bmatrix} \end{array}$$

2-step ahead cross-validation

See Hyndman, RJ, *Hyndsight blog*.

The LASSO: Theory-driven penalty

While cross-validation is a popular & powerful method for predictive purposes, it is often said to lack theoretical justification.

The LASSO: Theory-driven penalty

While cross-validation is a popular & powerful method for predictive purposes, it is often said to lack theoretical justification.

The theory of the 'rigorous' LASSO has two main ingredients:

- **Restricted eigenvalue condition (REC):** OLS requires full rank condition, which is too strong in the high-dimensional context. REC is much weaker.
- **Penalization level:** We need λ to be large enough to 'control' the noise in the data. At the same time, we want the penalty to be as small as possible (due to shrinkage bias).

The LASSO: Theory-driven penalty

While cross-validation is a popular & powerful method for predictive purposes, it is often said to lack theoretical justification.

The theory of the 'rigorous' LASSO has two main ingredients:

- **Restricted eigenvalue condition (REC):** OLS requires full rank condition, which is too strong in the high-dimensional context. REC is much weaker.
- **Penalization level:** We need λ to be large enough to 'control' the noise in the data. At the same time, we want the penalty to be as small as possible (due to shrinkage bias).

This allows to derive theoretical results for the LASSO:

→ consistent prediction and parameter estimation.

The theory of Belloni et al. (2012) allows for non-Gaussian & heteroskedastic errors, and has been extended to panel data (Belloni et al., 2016).

The LASSO: Information criteria

We have implemented the following information criteria:

$$\mathrm{AIC}(\lambda,\alpha) = N \log\left(\hat{\sigma}^2(\lambda,\alpha)\right) + 2df(\lambda,\alpha)$$

$$\mathrm{BIC}(\lambda,\alpha) = N \log\left(\hat{\sigma}^2(\lambda,\alpha)\right) + df(\lambda,\alpha)\log(N)$$

$$\mathrm{AICc}(\lambda,\alpha) = N \log\left(\hat{\sigma}^2(\lambda,\alpha)\right) + 2df(\lambda,\alpha)\frac{N}{N - df(\lambda,\alpha)}$$

$$\mathrm{EBIC}_\gamma(\lambda,\alpha) = N \log\left(\hat{\sigma}^2(\lambda,\alpha)\right) + df(\lambda,\alpha)\log(N) + 2\gamma df(\lambda,\alpha)\log(p)$$

df is the degrees of freedom. For the LASSO, *df* is equal to the number of non-zero coefficients (Zou et al., 2007).

The LASSO: Information criteria

Both AIC and BIC are less suitable in the large-p-small-N setting where they tend to select too many variables.

AIC$_c$ addresses the small sample bias of AIC and should be favoured over AIC if n is small (Sugiura, 1978; Hurvich and Tsai, 1989).

The BIC underlies the assumption that each model has the same probability. While this assumption seems reasonable if the researcher has no prior knowledge, it causes the BIC to over-select in the high-dimensional context.

Chen and Chen (2008) introduce the Extended BIC, which imposes an additional penalty on the number of parameters. The prior distribution is chosen such that dense models are less likely.

LASSO-type estimators

Various **alternative estimators** have been inspired by the LASSO; to name a few (all implemented in `LASSOPACK`):

Square-root LASSO (Belloni et al., 2011, 2014a)

$$\hat{\boldsymbol{\beta}}_{\sqrt{\text{lasso}}} = \arg\min \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \boldsymbol{x}_i'\boldsymbol{\beta})^2} + \frac{\lambda}{N}\sum_{j=1}^{p}\phi_j|\beta_j|,$$

The main advantage of the square-root LASSO comes into effect when rigorous penalization is employed: the optimal λ is independent of the unknown error under homoskedasticity, implying a practical advantage.

LASSO-type estimators

Various **alternative estimators** have been inspired by the LASSO; to name a few (all implemented in `LASSOPACK`):

Elastic net (Zou and Hastie, 2005)

The elastic net introduced by Zou and Hastie (2005) applies a mix of ℓ_1 (LASSO-type) and ℓ_2 (ridge-type) penalization:

$$\hat{\boldsymbol{\beta}}_{\text{elastic}} = \arg\min \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \boldsymbol{x}_i' \boldsymbol{\beta}\right)^2 + \frac{\lambda}{N} \left[\alpha \sum_{j=1}^{p} \psi_j |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \psi_j \beta_j^2 \right]$$

where $\alpha \in [0,1]$ controls the degree of ℓ_1 (LASSO-type) to ℓ_2 (ridge-type) penalization. $\alpha = 1$ corresponds to the LASSO, and $\alpha = 0$ to ridge regression.

LASSO-type estimators

Various **alternative estimators** have been inspired by the LASSO; to name a few (all implemented in `LASSOPACK`):

Post-estimation OLS (Belloni et al, 2012, 2013)

Penalized regression methods induce an attenuation bias that can be alleviated by post-estimation OLS, which applies OLS to the variables selected by the first-stage variable selection method, i.e.,

$$\hat{\boldsymbol{\beta}}_{\text{post}} = \arg\min \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \boldsymbol{x}_i' \boldsymbol{\beta}\right)^2 \quad \text{subject to} \quad \beta_j = 0 \ \text{ if } \ \tilde{\beta}_j = 0, \quad (1)$$

where $\tilde{\beta}_j$ is a sparse first-step estimator such as the LASSO. Thus, post-estimation OLS treats the first-step estimator as a genuine model selection technique.

LASSO-type estimators

Model selection is a much more difficult problem than prediction. The LASSO is only model selection consistent under the rather strong *irrepresentable condition* (Zhao and Yu, 2006; Meinshausen and Bühlmann, 2006).

LASSO-type estimators

Model selection is a much more difficult problem than prediction. The LASSO is only model selection consistent under the rather strong *irrepresentable condition* (Zhao and Yu, 2006; Meinshausen and Bühlmann, 2006).

This shortcoming motivated the **Adaptive LASSO (Zou, 2006)**:

$$\hat{\boldsymbol{\beta}}_{\text{alasso}} = \arg\min \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \boldsymbol{x}_i' \boldsymbol{\beta} \right)^2 + \frac{\lambda}{N} \sum_{j=1}^{p} \hat{\phi}_j |\beta_j|,$$

with $\hat{\phi}_j = 1/|\hat{\beta}_{0,j}|^{\theta}$. $\hat{\beta}_{0,j}$ is an initial estimator, such OLS, univariate OLS or the LASSO.

The Adaptive LASSO is variable-selection consistent for fixed p under weaker assumptions than the standard LASSO.

LASSOPACK

`LASSOPACK` includes three commands: `lasso2` implements LASSO and related estimators. `cvlasso` supports cross-validation, and `rlasso` offers the ‘rigorous’ (theory-driven) approach to penalization.

Basic syntax

`lasso2` *depvar indepvars* [*if*] [*in*] [, ...]

`cvlasso` *depvar indepvars* [*if*] [*in*] [, ...]

`rlasso` *depvar indepvars* [*if*] [*in*] [, ...]

All three commands support replay syntax and come with plenty of options. See the help files on SSC or `https://statalasso.github.io/` for the full syntax and list of options.

Application: Predicting Boston house prices

For demonstration, we use house price data available on the StatLib archive.

Number of observations: 506 census tracts
Number of variables: 14

Dependent variable: median value of owner-occupied homes (`medv`)

Predictors: crime rate, environmental measures, age of housing stock, tax rates, social variables. (See Descriptions.)

LASSOPACK: the lasso2 command

Estimate LASSO (default estimator) for a range of lambda values.

```
. lasso2 medv crim-lstat
  Knot|  ID     Lambda    s      L1-Norm        EBIC     R-sq   | Entered/removed
------+---------------------------------------------------------+----------------
     1|   1 6858.98553    1      0.00000  2255.87077   0.0000   | Added _cons.
     2|   2 6249.65216    2      0.08440  2218.17727   0.0924   | Added lstat.
     3|   3 5694.45029    3      0.28098  2182.00996   0.1737   | Added rm.
     4|  10 2969.09110    4      2.90443  1923.18586   0.5156   | Added ptratio.
     5|  20 1171.07071    5      4.79923  1763.74425   0.6544   | Added b.
     6|  22  972.24348    6      5.15524  1758.73342   0.6654   | Added chas.
     7|  26  670.12972    7      6.46233  1745.05577   0.6815   | Added crim.
     8|  28  556.35346    8      6.94988  1746.77384   0.6875   | Added dis.
     9|  30  461.89442    9      8.10548  1744.82696   0.6956   | Added nox.
    10|  34  318.36591   10     13.72934  1730.58682   0.7106   | Added zn.
    11|  39  199.94307   12     18.33494  1733.17551   0.7219   | Added indus rad.
    12|  41  165.99625   13     20.10743  1736.45725   0.7263   | Added tax.
    13|  47   94.98916   12     23.30144  1707.00224   0.7359   | Removed indus.
    14|  67   14.77724   13     26.71618  1709.60624   0.7405   | Added indus.
    15|  82    3.66043   14     27.44510  1720.65484   0.7406   | Added age.
Use ´long´ option for full output. Type e.g. ´lasso2, lic(ebic)´ to run the model selected by EBIC.
```

LASSOPACK: the lasso2 command

Estimate LASSO (default estimator) for a range of lambda values.

```
. lasso2 medv crim-lstat
  Knot|  ID     Lambda    s      L1-Norm        EBIC     R-sq   | Entered/removed
------+---------------------------------------------------------+----------------
     1|   1 6858.98553    1      0.00000  2255.87077   0.0000   | Added _cons.
     2|   2 6249.65216    2      0.08440  2218.17727   0.0924   | Added lstat.
     3|   3 5694.45029    3      0.28098  2182.00996   0.1737   | Added rm.
     4|  10 2969.09110    4      2.90443  1923.18586   0.5156   | Added ptratio.
     5|  20 1171.07071    5      4.79923  1763.74425   0.6544   | Added b.
     6|  22  972.24348    6      5.15524  1758.73342   0.6654   | Added chas.
     7|  26  670.12972    7      6.46233  1745.05577   0.6815   | Added crim.
     8|  28  556.35346    8      6.94988  1746.77384   0.6875   | Added dis.
     9|  30  461.89442    9      8.10548  1744.82696   0.6956   | Added nox.
    10|  34  318.36591   10     13.72934  1730.58682   0.7106   | Added zn.
    11|  39  199.94307   12     18.33494  1733.17551   0.7219   | Added indus rad.
    12|  41  165.99625   13     20.10743  1736.45725   0.7263   | Added tax.
    13|  47   94.98916   12     23.30144  1707.00224   0.7359   | Removed indus.
    14|  67   14.77724   13     26.71618  1709.60624   0.7405   | Added indus.
    15|  82    3.66043   14     27.44510  1720.65484   0.7406   | Added age.
Use ´long´ option for full output. Type e.g. ´lasso2, lic(ebic)´ to run the model selected by EBIC.
```

Columns in output show:

- `Knot` – points at which predictors enter or leave the active set (i.e., set of selected variables)
- `ID` – Index of lambda values
- `Lambda` – lambda values (default is to use 100 lambdas)
- `s` – number of selected predictors (including the constant)
- `L1-Norm` – L1-norm of coefficient vector
- `EBIC` – Extended BIC. Note: use `ic(`*string*`)` to display AIC, BIC or AIC$_c$
- `R-sq` – R-squared
- `Entered/removed` – indicates which predictors enter or leave the active set at knot

LASSOPACK: the lasso2 command

Estimate LASSO (default estimator) for a range of lambda values.

```
. lasso2 medv crim-lstat
  Knot|  ID     Lambda    s      L1-Norm        EBIC     R-sq   | Entered/removed
------+-----------------------------------------------------------+----------------
     1|   1 6858.98553    1      0.00000  2255.87077   0.0000   | Added _cons.
     2|   2 6249.65216    2      0.08440  2218.17727   0.0924   | Added lstat.
     3|   3 5694.45029    3      0.28098  2182.00996   0.1737   | Added rm.
     4|  10 2969.09110    4      2.90443  1923.18586   0.5156   | Added ptratio.
     5|  20 1171.07071    5      4.79923  1763.74425   0.6544   | Added b.
     6|  22  972.24348    6      5.15524  1758.73342   0.6654   | Added chas.
     7|  26  670.12972    7      6.46233  1745.05577   0.6815   | Added crim.
     8|  28  556.35346    8      6.94988  1746.77384   0.6875   | Added dis.
     9|  30  461.89442    9      8.10548  1744.82696   0.6956   | Added nox.
    10|  34  318.36591   10     13.72934  1730.58682   0.7106   | Added zn.
    11|  39  199.94307   12     18.33494  1733.17551   0.7219   | Added indus rad.
    12|  41  165.99625   13     20.10743  1736.45725   0.7263   | Added tax.
    13|  47   94.98916   12     23.30144  1707.00224   0.7359   | Removed indus.
    14|  67   14.77724   13     26.71618  1709.60624   0.7405   | Added indus.
    15|  82    3.66043   14     27.44510  1720.65484   0.7406   | Added age.
Use ´long´ option for full output. Type e.g. ´lasso2, lic(ebic)´ to run the model selected by EBIC.
```

Selected `lasso2` options:

- `sqrt`: use square-root LASSO.
- `alpha(`*real*`)`: use elastic net. *real* must lie in the interval [0,1]. `alpha(1)` is the LASSO (the default) and `alpha(0)` corresponds to ridge.
- `adaptive`: use adaptive LASSO.
- `ols`: use post-estimation OLS.
- `plotpath(`*string*`)`, `plotvar(`*varlist*`)`, `plotopt(`*string*`)` and `plotlabel` are for plotting.

See `help lasso2` or `https://statalasso.github.io/` for full syntax and list of options.

LASSOPACK: the lasso2 command

Run model selected by EBIC (using replay syntax):

```
. lasso2, lic(ebic)
Use lambda=16.21799867742649 (selected by EBIC).
------------------------------------------------------
         Selected |           Lasso   Post-est OLS
------------------+-----------------------------------
             crim |      -0.1028391     -0.1084133
               zn |       0.0433716      0.0458449
             chas |       2.6983218      2.7187164
              nox |     -16.7712529    -17.3760262
               rm |       3.8375779      3.8015786
              dis |      -1.4380341     -1.4927114
              rad |       0.2736598      0.2996085
              tax |      -0.0106973     -0.0117780
          ptratio |      -0.9373015     -0.9465246
                b |       0.0091412      0.0092908
            lstat |      -0.5225124     -0.5225535
------------------+-----------------------------------
   Partialled-out*|
------------------+-----------------------------------
            _cons |      35.2705812     36.3411478
------------------------------------------------------
```

- The lic(ebic) option can either be specified using the replay syntax or in the first lasso2 call.
- lic(ebic) can be replaced by lic(aicc), lic(aic) or lic(bic).
- Both LASSO and post-estimation OLS estimates are shown.

LASSOPACK: the cvlasso command

K-fold cross-validation with 10 folds using the LASSO (default behaviour).

```
. cvlasso medv crim-lstat, seed(123)
K-fold cross-validation with 10 folds. Elastic net with alpha=1.
Fold 1 2 3 4 5 6 7 8 9 10
          |              Lambda           MSPE       st. dev.
──────────┼──────────────────────────────────────────────
         1|       6858.9855       84.302552       5.7124688
..
        32|       383.47286       26.365176       3.5552884  ^
..
        64|       19.534637       23.418936       3.1298343  *
..
       100|       .68589855       23.441481       3.1133575
* lopt = the lambda that minimizes MSPE.
  Run model: cvlasso, lopt
^ lse = largest lambda for which MSPE is within one standard error of the minimal MSPE.
  Run model: cvlasso, lse
```

LASSOPACK: the cvlasso command

K-fold cross-validation with 10 folds using the LASSO (default behaviour).

```
. cvlasso medv crim-lstat, seed(123)
K-fold cross-validation with 10 folds. Elastic net with alpha=1.
Fold 1 2 3 4 5 6 7 8 9 10
          |           Lambda          MSPE       st. dev.
----------+------------------------------------------------

         1|        6858.9855      84.302552      5.7124688
..
        32|        383.47286      26.365176      3.5552884  ^
..
        64|        19.534637      23.418936      3.1298343  *
..
       100|        .68589855      23.441481      3.1133575
* lopt = the lambda that minimizes MSPE.
  Run model: cvlasso, lopt
^ lse = largest lambda for which MSPE is within one standard error of the minimal MSPE.
  Run model: cvlasso, lse
```

Selected `cvlasso` options:

- `sqrt`, `alpha(`*real*`)`, `adaptive`, etc. to control choice of estimation method.
- `rolling`: triggers rolling *h*-step ahead cross-validation (various options available).
- `plotcv(`*string*`)` and `plotopt(`*string*`)` for plotting.

See `help cvlasso` or `https://statalasso.github.io/` for full syntax and list of options.

LASSOPACK: the cvlasso command

Run model using value of λ that minimizes MSPE (using replay syntax):

```
. cvlasso, lopt
Estimate lasso with lambda=19.535 (lopt).
---------------------------------------------------------
         Selected |           Lasso   Post-est OLS
------------------+--------------------------------------
             crim |      -0.1016991     -0.1084133
               zn |       0.0428658      0.0458449
             chas |       2.6941511      2.7187164
              nox |     -16.6475746    -17.3760262
               rm |       3.8449399      3.8015786
              dis |      -1.4268524     -1.4927114
              rad |       0.2683532      0.2996085
              tax |      -0.0104763     -0.0117780
          ptratio |      -0.9354154     -0.9465246
                b |       0.0091106      0.0092908
            lstat |      -0.5225040     -0.5225535
------------------+--------------------------------------
   Partialled-out*|
------------------+--------------------------------------
            _cons |      35.0516465     36.3411478
---------------------------------------------------------
```

- `lopt` can be replaced by `lse`, which leads to a more parsimonious specification.
- `lopt`/`lse` can either be specified using the replay syntax (as above) or added to the first `cvlasso` call.

LASSOPACK: the rlasso command

Estimate ‘rigorous’ LASSO:

```
. rlasso medv crim-lstat
─────────────────────────────────────────────────────
        Selected │            Lasso   Post-est OLS
─────────────────┼───────────────────────────────────
            chas │        0.7844330      3.3200252
              rm │        4.0515800      4.6522735
         ptratio │       -0.6773194     -0.8582707
               b │        0.0039067      0.0101119
           lstat │       -0.5017705     -0.5180622
           _cons │ *     14.4716482     11.8535884
─────────────────────────────────────────────────────
*Not penalized
```

- rlasso uses feasible algorithms to estimate the optimal penalty level & loadings, and allows for non-Gaussian, heteroskedastic and cluster-dependence errors.
- In contrast to lasso2 and cvlasso, rlasso reports the selected model at the first call.

LASSOPACK: the rlasso command

Estimate 'rigorous' LASSO:

```
. rlasso medv crim-lstat
───────────────────┬──────────────────────────────────
           Selected │             Lasso   Post-est OLS
───────────────────┼──────────────────────────────────
               chas │         0.7844330      3.3200252
                 rm │         4.0515800      4.6522735
            ptratio │        -0.6773194     -0.8582707
                  b │         0.0039067      0.0101119
              lstat │        -0.5017705     -0.5180622
              _cons │*       14.4716482     11.8535884
───────────────────┴──────────────────────────────────
*Not penalized
```

Selected options:

- `sqrt`: use rigorous square-root LASSO
- `robust`: penalty level and penalty loadings account for heteroskedasticity
- `cluster(`*varname*`)`: penalty level and penalty loadings account for clustering on variable *varname*

See `help rlasso` or `https://statalasso.github.io/` for full syntax and list of options.

Application: Predicting Boston house prices

We divide the sample in half (253/253). Use first half for estimation, and second half for assessing prediction performance.

Estimation methods:

- 'Kitchen sink' OLS: include all regressors
- Stepwise OLS: begin with general model and drop if p-value > 0.05
- 'Rigorous' LASSO with theory-driven penalty
- LASSO with 10-fold cross-validation
- LASSO with penalty level selected by information criteria

Application: Predicting Boston house prices

We divide the sample in half (253/253). Use first half for estimation, and second half for assessing prediction performance.

	OLS	Stepwise	rlasso	cvlasso	lasso2	lasso2
					AIC/AICc	BIC/EBIC$_1$
crim	1.201*	1.062*		0.985	1.053	
zn	0.0245			0.0201	0.0214	
indus	0.01000					
chas	0.425			0.396	0.408	
nox	-8.443	-8.619*		-6.560	-7.067	
rm	8.878***	9.685***	8.681	8.925	8.909	9.086
age	-0.0485***	-0.0585***	-0.00608	-0.0470	-0.0475	-0.0335
dis	-1.120***	-0.956***		-1.025	-1.057	-0.463
rad	0.204			0.158	0.171	
tax	-0.0160***	-0.0121***	-0.00267	-0.0148	-0.0151	-0.00925
ptratio	-0.660***	-0.766***	-0.417	-0.660	-0.659	-0.659
b	0.0178***	0.0175***	0.000192	0.0169	0.0172	0.0110
lstat	-0.115*		-0.124	-0.113	-0.113	-0.109
Selected predictors	13	8	6	12	12	7
in-sample RMSE	3.160	3.211	3.656	3.164	3.162	3.279
out-of-sample RMSE	17.42	15.01	7.512	14.78	15.60	7.252

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$. Constant omitted.

Application: Predicting Boston house prices

- OLS exhibits lowest in-sample RMSE, but worst out-of-sample prediction performance. Classical example of overfitting.
- Stepwise regression performs slightly better than OLS, but is known to have many problems: biased (over-sized) coefficients, inflated R^2, invalid p-values.
- In this example, AIC & AICc and BIC & EBIC$_1$ yield the same results, but AICc and EBIC are generally preferable for large-p-small-n problems.
- LASSO with 'rigorous' penalization and LASSO with BIC/EBIC$_1$ exhibit best out-of-sample prediction performance.

Interlude: Stata/Mata coding issues

Parameter vectors may start out large and end up large, or start out large and end up sparse. How to store and report?

Stata's factor variables and operators: extremely powerful, very useful. Specify multiple interactions and model quickly becomes high-dimensional. But can be hard to work with subsets of factor variables (e.g. Stata extended macro function `: colnames b` will rebase the selected subset of factor variables extracted from `b`). Our solution: create temp vars and maintain a dictionary relating them to a clean list of factor vars.

Cross-validation means repeatedly creating many temp vars when vars are standardized (scaled). Can be slow.

- Trick #1: Use uninitialized temp vars created in Mata rather than temp vars intialized to missing in Stata.
- Trick #2: Optionally avoid temp vars completely by standardizing on-the-fly (i.e., when estimating) instead of repeatedly creating new standardized vars ex ante.

The LASSO and Causal Inference

The main strength of the LASSO is prediction (rather than model selection). But the LASSO's strength as a prediction technique can also be used to aid causal inference.

The LASSO and Causal Inference

The main strength of the LASSO is prediction (rather than model selection). But the LASSO's strength as a prediction technique can also be used to aid causal inference.

Basic setup: we **already know** the causal variable of interest. No variable selection needed for this. But the LASSO can be used to **select other variables or instruments** used in the estimation.

The LASSO and Causal Inference

The main strength of the LASSO is prediction (rather than model selection). But the LASSO's strength as a prediction technique can also be used to aid causal inference.

Basic setup: we **already know** the causal variable of interest. No variable selection needed for this. But the LASSO can be used to **select other variables or instruments** used in the estimation.

Two cases:

(1) Selection of controls, to address omitted variable bias.

(2) Selection of instruments, to address endogeneity via IV estimation.

We look at selection of controls first (implemented in `pdslasso`) and then selection of IVs (implemented in `ivlasso`).

NB: the package can be used for problems involving selection of both controls and instruments.

Choosing controls: Post-Double-Selection LASSO

Our model is

$$y_i = \underbrace{\boxed{\alpha d_i}}_{\text{aim}} + \underbrace{\boxed{\beta_1 x_{i,1} + \ldots + \beta_p x_{i,p}}}_{\text{nuisance}} + \varepsilon_i.$$

The causal variable of interest or "treatment" is d_i. The xs are the set of potential controls and not directly of interest. We want to obtain an estimate of the parameter α.

The problem is the controls. We want to include controls because we are worried about omitted variable bias – the usual reason for including controls.

But which ones do we use?

Choosing controls: Post-Double-Selection LASSO

But which controls do we use?

If we use too many, we run into a version of the overfitting problem. We could even have $p > n$, so using them all is just impossible.

If we use too few, or use the wrong ones, then OLS gives us a biased estimate of α because of omitted variable bias.

And to make matters worse: "researcher degrees of freedom" and "*p*-hacking". Researchers may consciously or unconsciously choose controls to generate the results they want.

Theory-driven choice of controls can not only generate good performance in estimation, it can also reduce the "researcher degrees of freedom" and restrain *p*-hacking.

Choosing controls: Post-Double-Selection LASSO

Our model is

$$y_i = \underbrace{\alpha d_i}_{\text{aim}} + \underbrace{\beta_1 x_{i,1} + \ldots + \beta_p x_{i,p}}_{\text{nuisance}} + \varepsilon_i .$$

Naive approach: estimate the model using the LASSO (imposing that d_i is not subject to selection), and use the controls selected by the LASSO.

Choosing controls: Post-Double-Selection LASSO

Our model is

$$y_i = \underbrace{\alpha d_i}_{\text{aim}} + \underbrace{\beta_1 x_{i,1} + \ldots + \beta_p x_{i,p}}_{\text{nuisance}} + \varepsilon_i.$$

Naive approach: estimate the model using the LASSO (imposing that d_i is not subject to selection), and use the controls selected by the LASSO.

Badly biased. Reason: we might miss controls that have a strong predictive power for d_i, but only small effect on y_i.

Similarly, if we only consider the regression of d_i against the controls, we might miss controls that have a strong predictive power for y_i, but only a moderately sized effect on d_i. See Belloni et al. (2014b).

Choosing controls: Post-Double-Selection LASSO

Post-Double-Selection (PDS) LASSO (Belloni et al., 2014c, *ReStud*):

- Step 1: Use the LASSO to estimate

  $$y_i = \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_j x_{i,j} + \ldots + \beta_p x_{i,p} + \varepsilon_i,$$

  i.e., without d_i as a regressor. Denote the set of LASSO-selected controls by A.
- Step 2: Use the LASSO to estimate

  $$d_i = \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_j x_{i,j} + \ldots + \beta_p x_{i,p} + \varepsilon_i,$$

  i.e., where the causal variable of interest is the dependent variable. Denote the set of LASSO-selected controls by B.
- Step 3: Estimate using OLS

  $$y_i = \alpha d_i + \mathbf{w}_i' \beta + \varepsilon_i$$

  where $\mathbf{w}_i = A \cup B$, i.e., the union of the selected controls from Steps 1 and 2.

Choosing controls: "Double-Orthogonalization"

An alternative to PDS: "Double-Orthogonalization", proposed by Chernozhukov-Hansen-Spindler 2015 (CHS).

The **PDS method** is equivalent to Frisch-Waugh-Lovell partialling-out all selected controls from both y_i and d_i.

The **CHS method** essentially partials out from y_i only the controls in set A (selected in Step 1, using the LASSO with y_i on the LHS), and partials out from d_i only the controls in set B (selected in Step 2, using the LASSO with d_i on the LHS).

CHS partialling-out can use either the LASSO or Post-LASSO coefficients.

Both methods are supported by `pdslasso`.

Important PDS caveat: we can do inference on the causal variable(s), but not on the selected high-dimensional controls. (The CHS method partials them out, so the temptation is not there!)

Using the LASSO to choose controls

Why can we use the LASSO to select controls even though the LASSO is (in most scenarios) not model selection consistent?

Using the LASSO to choose controls

Why can we use the LASSO to select controls even though the LASSO is (in most scenarios) not model selection consistent?

Two ways to look at this:

- **Immunization property:** moderate model selection mistakes of the LASSO do not affect the asymptotic distribution of the estimator of the low-dimensional parameters of interest (Belloni et al., 2012, 2014c). We can treat modelling the the nuisance component of our structural model as a prediction problem.
- The **irrepresentable condition** states that the LASSO will fail to distinguish between two variables (one in the active set, the other not) if they are highly correlated. These type of variable selection mistakes are not a problem if the aim is to control for confounding factors or estimate ("predict") instruments.

PDSLASSO: the pdslasso command

The PDSLASSO package has two commands, pdslasso and ivlasso. In fact they are the same command, and the only difference is that pdslasso has a more restrictive syntax.

Basic syntax

pdslasso *depvar d_varlist (hd_controls_varlist)* [*if*] [*in*] [, ...]

with many options and features, including:

- heteroskedastic- and cluster-robust penalty loadings.
- LASSO or Sqrt-LASSO
- support for Stata time-series and factor-variables
- pweights and aweights
- fixed effects and partialling-out unpenalized regressors
- saving intermediate rlasso output
- ... and all the rlasso options

Example: Donohue & Levitt (2001) (via BCH 2014)

Example: Donohue & Levitt (2001) on the effects of abortion on crime rates using state-level data (via Belloni-Chernozhukov-Hansen JEP 2014). 50 states, data cover 1985-97.

Did legalization of abortion in the US around 1970 lead to lower crime rates 20 years later? (Idea: woman more likely to terminate in difficult circumstances; prevent this and the consequences are visible in the child's behavior when they grow up.)

Controversial paper, mostly hasn't stood up to later scrutiny. But a good example here because the PDS application is discussed in BCH (2014) and because it illustrates the ease of use of factor variables to create interactions.

Example: Donohue & Levitt (2001) (via BCH 2014)

Donohue & Levitt look at different categories of crime; we look at the property crime example. Estimation is in first differences.

y_{it} is the growth rate in the property crime rate in state i, year t

d_{it} is the growth rate in the abortion rate in state i, year $t - 20$ (appx)

And the controls come from a very long list:

Controls in the Donohue & Levitt (2001) example

Controls (all state-level):

- initial level and growth rate of property crime
- growth in prisoners per capita, police per capita, unemployment rate, per capita income, poverty rate, spending on welfare program at time $t - 15$, gun law dummy, beer consumption per capita (original Donohue-Levitt list of controls)
- plus quadratic in lagged levels in all the above
- plus quadratic state-level means in all the above
- plus quadratic in initial state-level values in all the above
- plus quadratic in initial state-level growth rates in all the above
- plus all the above interacted with a quadratic time trend
- year dummies (unpenalized)

In all, 336 high-dimensional controls and 12 unpenalized year dummies.

We use cluster-robust penalty loadings in the LASSOs and cluster-robust SEs in the final OLS estimations of the structural equation.

pdslasso command syntax

Usage in the Donohue-Levitt example:

```
pdslasso dep_var d_varlist (hd_controls_varlist),
    partial(unpenalized_controls)
    cluster(state_id)
    rlasso
```

The unpenalized variables in `partial(.)` must be in the main `hd_controls_varlist`.

`cluster(.)` implies cluster-robust penalty loadings and cluster-robust SEs in the final OLS estimation. (These options can also be controlled separately.)

The `rlasso` option of `pdslasso` displays the intermediate `rlasso` results and also stores them for later replay and inspection.

Levitt-Donohue example: pdslasso command line

```
pdslasso D.lpc_prop D.efaprop
    (
    c.prop0##c.prop0
    c.Dprop0##c.Dprop0
    c.(D.(xxprison-xxbeer))##c.(D.(xxprison-xxbeer))
    c.(L.xxprison)##c.(L.xxprison)
    c.(L.xxpolice)##c.(L.xxpolice)
    ...
    (c.Dxxafdc150##c.Dxxafdc150)#(c.trend##c.trend)
    (c.Dxxgunlaw0##c.Dxxgunlaw0)#(c.trend##c.trend)
    (c.Dxxbeer0##c.Dxxbeer0)#(c.trend##c.trend)
    i.year
    )
    , partial(i.year) cluster(statenum) rlasso
```

Levitt-Donohue example: `pdslasso` output

```
Partialling out unpenalized controls...
1.  (PDS/CHS) Selecting HD controls for dep var D.lpc_prop...
Selected: xxincome0 xxafdc150 c.Mxxincome#c.trend
2.  (PDS/CHS) Selecting HD controls for exog regressor D.efaprop...
Selected: prop0 cD.xxprison#cD.xxbeer L.xxincome

Estimation results:

Specification:
Regularization method:                 lasso
Penalty loadings:                      cluster-lasso
Number of observations:                600
Number of clusters:                     50
Exogenous (1):                         D.efaprop
High-dim controls (336):               prop0 c.prop0#c.prop0 Dprop0 c.Dprop0#c.Dprop0
                                       D.xxprison D.xxpolice D.xxunemp D.xxincome D.xxpover
                                       D.xxafdc15 D.xxgunlaw D.xxbeer
                                       cD.xxprison#cD.xxprison cD.xxprison#cD.xxpolice
                                       cD.xxprison#cD.xxunemp cD.xxprison#cD.xxincome
                                       cD.xxprison#cD.xxpover cD.xxprison#cD.xxafdc15
                                       cD.xxprison#cD.xxgunlaw cD.xxprison#cD.xxbeer
                                       ...
                                       c.Dxxbeer0#c.Dxxbeer0#c.trend
                                       c.Dxxbeer0#c.Dxxbeer0#c.trend#c.trend
Selected controls (6):                 prop0 cD.xxprison#cD.xxbeer L.xxincome xxincome0
                                       xxafdc150 c.Mxxincome#c.trend
Partialled-out controls (12):          86b.year 87.year 88.year 89.year 90.year 91.year
                                       92.year 93.year 94.year 95.year 96.year 97.year
```

Levitt-Donohue example: `pdslasso` output

Note at the beginning of the output the following message:

```
Partialling out unpenalized controls...
1.  (PDS/CHS) Selecting HD controls for dep var D.lpc_prop...
Selected: xxincome0 xxafdc150 c.Mxxincome#c.trend
2.  (PDS/CHS) Selecting HD controls for exog regressor D.efaprop...
Selected: prop0 cD.xxprison#cD.xxbeer L.xxincome
```

Specifying the `rlasso` option means you get to see the “rigorous” LASSO results for Step 1 (selecting controls for the dependent variable y) and Step 2 (selecting controls for the causal variable d):

Levitt-Donohue example: pdslasso output

```
lasso estimation(s):

_pdslasso_step1
---------------------------------------------------
         Selected |           Lasso   Post-est OLS
------------------+--------------------------------
        xxincome0 |      -0.0010708     -0.8691891
        xxafdc150 |      -0.0027622     -0.0147806
                  |
      c.Mxxincome#|
          c.trend |      -5.4258229     -7.2534845
---------------------------------------------------

_pdslasso_step2
---------------------------------------------------
         Selected |           Lasso   Post-est OLS
------------------+--------------------------------
            prop0 |       0.2953010      0.3044819
                  |
      cD.xxprison#|
        cD.xxbeer |      -1.4925825     -6.8662863
                  |
         xxincome |
              L1. |      16.3769883     26.0105200
---------------------------------------------------
```

Levitt-Donohue example: pdslasso output

pdslasso reports 3 sets of estimations of the structural equation:

- CHS using LASSO-orthogonalized variables
- CHS using Post-LASSO-OLS-orthogonalized variables
- PDS using all selected variables as controls

```
OLS using CHS lasso-orthogonalized vars
                              (Std. Err. adjusted for 50 clusters in statenum)
------------------------------------------------------------------------------
             |               Robust
  D.lpc_prop |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     efaprop |
         D1. |  -.0645541    .044142    -1.46   0.144    -.1510708    .0219626
------------------------------------------------------------------------------

OLS using CHS post-lasso-orthogonalized vars
                              (Std. Err. adjusted for 50 clusters in statenum)
------------------------------------------------------------------------------
             |               Robust
  D.lpc_prop |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     efaprop |
         D1. |  -.0628553   .0481347    -1.31   0.192    -.1571975     .031487
------------------------------------------------------------------------------
```

Levitt-Donohue example: `pdslasso` output I

Reminder: we can do inference on the causal variable *d* (here, `D.efaprop`) but **not** on the selected controls.

```
OLS with PDS-selected variables and full regressor set
                                       (Std. Err. adjusted for 50 clusters in statenum)
--------------------------------------------------------------------------------------
                      |               Robust
           D.lpc_prop |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
----------------------+---------------------------------------------------------------
              efaprop |
                  D1. |  -.0897886    .056477    -1.59   0.112    -.2004815    .0209043
                      |
                prop0 |   .0088669   .0253529     0.35   0.727    -.0408239    .0585577
                      |
cD.xxprison#cD.xxbeer |  -.1947112   2.542185    -0.08   0.939    -5.177302     4.78788
                      |
             xxincome |
                  L1. |   21.28066   4.650744     4.58   0.000     12.16537    30.39595
                      |
            xxincome0 |  -15.71353   4.354251    -3.61   0.000    -24.24771   -7.179358
            xxafdc150 |  -.0264625   .0074138    -3.57   0.000    -.0409932   -.0119318
                      |
  c.Mxxincome#c.trend |  -9.449333    4.21689    -2.24   0.025    -17.71429    -1.18438
                      |
                 year |
                  87  |   .0551684   .0357699     1.54   0.123    -.0149394    .1252762
                  88  |   .1144515   .0698399     1.64   0.101    -.0224323    .2513353
                  89  |   .2042385   .1017077     2.01   0.045      .004895     .403582
```

Levitt-Donohue example: pdslasso output II

```
                  90  |   .2827328   .1363043     2.07   0.038     .0155812    .5498844
                  91  |   .3645207   .1675923     2.18   0.030     .0360458    .6929955
                  92  |   .3915994   .2067296     1.89   0.058    -.0135831    .7967819
                  93  |   .4761361   .2398321     1.99   0.047     .0060738    .9461985
                  94  |     .58132   .2744475     2.12   0.034     .0434128    1.119227
                  95  |   .6640497   .3108557     2.14   0.033     .0547837    1.273316
                  96  |    .689488   .3448339     2.00   0.046     .0136261     1.36535
                  97  |   .7730275   .3812726     2.03   0.043      .025747    1.520308
                      |
                _cons |  -.4087512   .2016963    -2.03   0.043    -.8040687   -.0134338
---------------------------------------------------------------------------------------
Standard errors and test statistics valid for the following variables only:
    D.efaprop
------------------------------------------------------------------------------
```

pdslasso with rlasso option

The rlasso option stores the PDS LASSO estimations for later replay or restore (NB: pdslasso calls rlasso to do this. The variables may be temp vars, as here, in which case rlasso is also given the dictionary mapping temp names to display names.)

```
. est dir

-----------------------------------------------------------
        name | command      depvar       npar  title 
-------------+---------------------------------------------
_pdslasso_~1 | rlasso       D.lpc_prop      3  lasso step 1
_pdslasso_~2 | rlasso       D.efaprop       3  lasso step 2
-----------------------------------------------------------

. estimates replay _pdslasso_step1

. estimates replay _pdslasso_step2
```

Choosing instruments: IV LASSO

Our model is:

$$y_i = \alpha d_i + \varepsilon_i$$

As above, the causal variable of interest or "treatment" is d_i. We want to obtain an estimate of the parameter α.

But we cannot use OLS because d_i is endogenous: $E(d_i \varepsilon_i) \neq 0$.

IV estimation is possible: we have available instruments $z_{i,j}$ that are valid (orthogonal to the error term): $E(z_{ij} \varepsilon_i) \neq 0$.

Choosing instruments: IV LASSO

Our model is:

$$y_i = \alpha d_i + \varepsilon_i$$

As above, the causal variable of interest or "treatment" is d_i. We want to obtain an estimate of the parameter α.

But we cannot use OLS because d_i is endogenous: $E(d_i \varepsilon_i) \neq 0$.

IV estimation is possible: we have available instruments $z_{i,j}$ that are valid (orthogonal to the error term): $E(z_{ij} \varepsilon_i) \neq 0$.

The problem is we have many instruments. The IV estimator is badly biased when the number of instruments is large and/or the instruments are only weakly correlated with the endogneous regressor(s).

Choosing instruments: IV LASSO

Our model is:

$$y_i = \alpha d_i + \varepsilon_i$$

As above, the causal variable of interest or "treatment" is d_i. We want to obtain an estimate of the parameter α.

But we cannot use OLS because d_i is endogenous: $E(d_i \varepsilon_i) \neq 0$.

IV estimation is possible: we have available instruments $z_{i,j}$ that are valid (orthogonal to the error term): $E(z_{ij} \varepsilon_i) \neq 0$.

The problem is we have many instruments. The IV estimator is badly biased when the number of instruments is large and/or the instruments are only weakly correlated with the endogneous regressor(s).

Choosing instruments: IV LASSO

EXAMPLES:

- Uncertainty about the correct choice/specification of instruments. Various alternatives available but theory provides no guidance.
- Unknown non-linear relationship between the endogenous regressor and instruments,

  $$d_i = f(\boldsymbol{z}_i) + \nu_i.$$

  Use large set of transformation of $\boldsymbol{z}_i$ to approximate the non-linear form.

Choosing instruments: IV LASSO

EXAMPLES:

- Uncertainty about the correct choice/specification of instruments. Various alternatives available but theory provides no guidance.
- Unknown non-linear relationship between the endogenous regressor and instruments,

  $$d_i = f(\boldsymbol{z}_i) + \nu_i.$$

  Use large set of transformation of $\boldsymbol{z}_i$ to approximate the non-linear form.

Idea: The first stage of 2SLS is a prediction problem. So we can use LASSO-type methods.

Choosing instruments: IV LASSO

Choose the instruments by using the LASSO on the first-stage regression (d_i on LHS, IVs on RHS) and then two possible approaches, analogous to PDS vs CHS in the exogenous case covered above:

PDS-type approach: Assemble instruments for each endogenous regressor, and use the union of selected IVs in a standard IV estimation. Extends straightforwardly to selecting from high-dimensional controls (as in basic PDS). Also extends straightforwardly to models with both exogenous and endogenous causal variables d.

CHS-type approach (Belloni et all 2012, CHS 2015): Use predicted value $\hat{d}_i$ from first-stage LASSO/Post-LASSO as an optimal instrument in a standard IV estimation. Extends not-so-straightforwardly (multiple steps involved) to selecting from high-dimensional controls and to models with both exogenous and endogenous d (see the CHS paper).

Example: Angrist-Kruger 1991 Quarter-of-birth IVs

Model is a standard Mincer-type wage equation

$$log(wage)_i = \alpha educ_i + <controls> + \varepsilon_i$$

And we have the usual endogeneity (omitted variables bias) with $educ_i$ (years of education).

Angrist-Kruger (1991): compulsory school age laws vary from state to state, so amount of education varies exogenously by state according to when you were born and when the cutoff kicked in.

They estimated the above with various controls in the main equation (year dummies, place-of-birth state dummies), and using as instrument the quarter of birth plus interactions of QOB with YOB and POB dummies.

Example: Angrist-Kruger 1991 Quarter-of-birth IVs

Problem: These interaction instruments in some specification were very numerous (could number several hundred) and were weakly correlated with years of education.

Paper is now very widely used and cited as examples of the "weak instruments problem" and the "many weak instruments problem" in particular.

LASSO solution: use the LASSO to select instruments.

Perfectly possible that the LASSO will select no instruments at all. This is good! Means that there is evidence that the model is unidentified, or not identified strongly enough to be able to do reliable evidence using standard IV methods. Better to avoid using standard IV methods in this case.

ivlasso command syntax

Basic syntax:

Basic syntax

`ivlasso` *depvar d_varlist (hd_controls_varlist) (endog_d_varlist = high_dimensional_IVs)* [*if*] [*in*] [, ...]

Usage in the Angrist-Kruger example:

```
ivlasso dep_var (hd_controls_varlist)
    (endog_d_varlist = high_dimensional_IVs),
    partial(unpenalized_controls)
    fe
    rlasso
```

where we illustrate the usage of state fixed effects.

Angrist-Kruger example: ivlasso command line

Fixed effects (data are xtset by state), year dummies are unpenalized controls, IVs are QOB and QOB interacted with year dummies, save the rlasso results:

```
ivlasso lnwage (i.yob)
    (educ=i.qob i.yob#i.qob), fe partial(i.yob) rlasso
```

Fixed effects, year dummies penalized, IVs are QOB and QOB interacted with year dummies and state dummies:

```
ivlasso lnwage (ibn.yob)
    (educ=ibn.qob ibn.yob#ibn.qob ibn.pob#ibn.qob), fe
```

Note the use of base factor variables. In effect we let the LASSO choose the base categories.

Angrist-Kruger example: `ivlasso` output

```
Fixed effects transformation...
1.  (PDS/CHS) Selecting HD controls for dep var lnwage...
Selected:
3.  (PDS) Selecting HD controls for endog regressor educ...
Selected: 30bn.yob 31.yob 32.yob 33.yob 36.yob 37.yob 38.yob 39.yob
5.  (PDS/CHS) Selecting HD controls/IVs for endog regressor educ...
Selected: 30bn.yob 31.yob 32.yob 37.yob 38.yob 39.yob 1bn.qob 4.qob
          30bn.yob#1bn.qob 47.pob#4.qob
6a. (CHS) Selecting lasso HD controls and creating optimal IV for endog regressor educ...
Selected: 30bn.yob 31.yob 32.yob 37.yob 38.yob 39.yob
6b. (CHS) Selecting post-lasso HD controls and creating optimal IV for endog regressor educ...
Selected: 30bn.yob 31.yob 32.yob 37.yob 38.yob 39.yob
7.  (CHS) Creating orthogonalized endogenous regressor educ...
```

Angrist-Kruger example: `ivlasso` output

```
Estimation results:

Specification:
Regularization method:                 lasso
Penalty loadings:                      homoskedastic
Number of observations:                329,509
Number of fixed effects:                    51
Endogenous (1):                        educ
High-dim controls (10):                30bn.yob 31.yob 32.yob 33.yob 34.yob
                                       35.yob 36.yob 37.yob 38.yob 39.yob
Selected controls, PDS (8):            30bn.yob 31.yob 32.yob 33.yob 36.yob
                                       37.yob 38.yob 39.yob
Selected controls, CHS-L (6):          30bn.yob 31.yob 32.yob 37.yob 38.yob 39.yob
Selected controls, CHS-PL (6):         30bn.yob 31.yob 32.yob 37.yob 38.yob 39.yob
High-dim instruments (248):            1bn.qob 2.qob 3.qob 4.qob 30bn.yob#1bn.qob
                                       30bn.yob#2.qob
                                       ...
                                       56.pob#1bn.qob 56.pob#2.qob 56.pob#3.qob
                                       56.pob#4.qob
Selected instruments (4):              1bn.qob 4.qob 30bn.yob#1bn.qob
                                       47.pob#4.qob
```

Note that out of 248 instruments, only 4 were selected. Also note how the LASSO chose the base categories.

Angrist-Kruger example: `ivlasso` output

Results using the optimal instruments (LASSO and Post-LASSO) methods:

```
Structural equation (fixed effects, #groups=51):

IV using CHS lasso-orthogonalized vars
------------------------------------------------------------------------------
      lnwage |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   .0880653   .0191934     4.59   0.000     .0504469    .1256837
------------------------------------------------------------------------------

IV using CHS post-lasso-orthogonalized vars
------------------------------------------------------------------------------
      lnwage |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   .0873329   .0182045     4.80   0.000     .0516527     .123013
------------------------------------------------------------------------------
```

Angrist-Kruger example: `ivlasso` output

Results using the PDS methodology: only the 4 variables selected as instruments (`1bn.qob`, `4.qob`, `30bn.yob#1bn.qob` and `47.pob#4.qob`); note also that nearly all the year dummies were selected by the LASSO as controls,

```
IV with PDS-selected variables and full regressor set
------------------------------------------------------------------------------
      lnwage |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   .0872734   .0181917     4.80   0.000     .0516183    .1229285
             |
         yob |
         30  |   .0287962    .007576     3.80   0.000     .0139474    .0436449
         31  |    .020713   .0057296     3.62   0.000     .0094832    .0319427
         32  |   .0139227   .0049638     2.80   0.005     .0041939    .0236515
         33  |    .010831   .0046016     2.35   0.019      .001812      .01985
         36  |  -.0067316   .0045436    -1.48   0.138    -.0156368    .0021737
         37  |  -.0131574   .0049521    -2.66   0.008    -.0228634   -.0034513
         38  |  -.0155679   .0058099    -2.68   0.007    -.0269552   -.0041806
         39  |  -.0271007   .0063086    -4.30   0.000    -.0394653   -.0147361
------------------------------------------------------------------------------
Standard errors and test statistics valid for the following variables only:
    educ
------------------------------------------------------------------------------
```

Installation

Both **LASSOPACK** and **PDSLASSO** are available through SSC:

```
ssc install lassopack
ssc install pdslasso
```

To get the latest stable version from our website, check the installation instructions at `https://statalasso.github.io/installation/`.

Summary I

Machine learning/Penalized regression

- ML provides wide set of flexible methods focused on prediction and classification problems.
- Penalized regression outperforms OLS in terms of prediction due to *bias-variance-tradeoff*.
- LASSO is just one ML method, but has some advantages: closely related to OLS, sparsity, well-developed theory, etc.

Summary I

Machine learning/Penalized regression

- ML provides wide set of flexible methods focused on prediction and classification problems.
- Penalized regression outperforms OLS in terms of prediction due to *bias-variance-tradeoff*.
- LASSO is just one ML method, but has some advantages: closely related to OLS, sparsity, well-developed theory, etc.

The package `LASSOPACK`

- implements penalized regression methods: LASSO, elastic net, ridge, square-root LASSO, adaptive LASSO.
- uses fast path-wise coordinate descent algorithms
- three commands for three different penalization approaches: cross-validation (`cvlasso`), information criteria (`lasso2`) and ‘rigorous’ (theory-driven) penalization (`rlasso`).

Summary II

Causal inference

- Distinction between *parameters of interest* and *high-dimensional set of controls/instruments*.
- General framework allows for causal inference with low-dimensional parameters robust to misspecification; and avoids problems associated with model selection using significance testing.
- But there's a price: the framework is designed for inference on low-dim parameters only.

The package `PDSLASSO`

- includes `pdslasso` and `ivlasso` for selection of controls and/instruments using 'rigorous' LASSO and Sqrt-LASSO.
- supports weak-identification robust inference using *sup-score* test.

Recommended resources

- *NBER* Summer Institute 2013: Econometric Methods for High-Dimensional Data, with video lectures by Victor Chernozhukov and Christian Hansen, among others
- Two free textbooks: *An Introduction to Statistical Learning* (non-technical) and *Elements of Statistical Learning* (more advanced).
- Video lectures on Statistical Learning by Trevor Hastie & Rob Tibshirani (based on *An Introduction to Statistical Learning*)
- See References and our website `https://statalasso.github.io/` for more material.

Appendix: Boston house prices

Variable descriptions

Name	*Description*
`crim`	per capita crime rate by town
`zn`	proportion of residential land zoned for lots over 25,000 sq.ft.
`indus`	proportion of non-retail business acres per town
`chas`	Charles River dummy variable ($= 1$ if tract bounds river; 0 otherwise)
`nox`	nitric oxides concentration (parts per 10 million)
`rm`	average number of rooms per dwelling
`age`	proportion of owner-occupied units built prior to 1940
`dis`	weighted distances to five Boston employment centres
`rad`	index of accessibility to radial highways
`tax`	full-value property-tax rate per $10,000
`pratio`	pupil-teacher ratio by town
`b`	$1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town
`lstat`	% lower status of the population
`medv`	Median value of owner-occupied homes in $1000's

Go back.

References I

Sendhil Mullainathan and Jann Spiess. Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives*, 31(2): 87–106, may 2017. doi: 10.1257/jep.31.2.87. URL http://www.aeaweb.org/articles?id=10.1257/jep.31.2.87.

Susan Athey. The Impact of Machine Learning on Economics. *NBER AI Workshop 2017*, (MI):1–27, 2017. ISSN 0895-3309. doi: 10.1257/jep.31.2.87.

Hal R Varian. Big Data: New Tricks for Econometrics. *The Journal of Economic Perspectives*, 28(2):pp. 3–27, 2014. ISSN 08953309. URL http://www.jstor.org/stable/23723482.

References II

Joseph P Simmons, Leif D Nelson, and Uri Simonsohn. False-Positive Psychology: Undisclosed Flexibility in Data Collection and Analysis Allows Presenting Anything as Significant. *Psychological Science*, 22 (11):1359–1366, oct 2011. ISSN 0956-7976. doi: 10.1177/0956797611417632. URL https://doi.org/10.1177/0956797611417632.

Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ziad Obermeyer. Prediction Policy Problems. *American Economic Review*, 105(5): 491–495, may 2015. doi: 10.1257/aer.p20151023. URL http://www.aeaweb.org/articles?id=10.1257/aer.p20151023.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1 (2):302–332, 2007. ISSN 1932-6157. doi: 10.1214/07-AOAS131. URL http://projecteuclid.org/euclid.aoas/1196438020.

References III

Robert Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58 (1):267–288, jan 1996. ISSN 00359246. doi: 10.2307/2346178. URL `http://www.jstor.org/stable/2346178`.

Alexandre Belloni, D Chen, Victor Chernozhukov, and Christian Hansen. Sparse Models and Methods for Optimal Instruments With an Application to Eminent Domain. *Econometrica*, 80(6):2369–2429, 2012. ISSN 1468-0262. doi: 10.3982/ECTA9626. URL `http://dx.doi.org/10.3982/ECTA9626`.

Alexandre Belloni, Victor Chernozhukov, Christian Hansen, and Damian Kozbur. Inference in High Dimensional Panel Models with an Application to Gun Control. *Journal of Business & Economic Statistics*, 34(4):590–605, 2016. URL `http://arxiv.org/abs/1411.6507`.

References IV

Hui Zou, Trevor Hastie, and Robert Tibshirani. On the "degrees of freedom" of the lasso. *Ann. Statist.*, 35(5):2173–2192, 2007. doi: 10.1214/009053607000000127. URL https://doi.org/10.1214/009053607000000127.

Nariaki Sugiura. Further analysts of the data by akaike' s information criterion and the finite corrections. *Communications in Statistics - Theory and Methods*, 7(1):13–26, jan 1978. ISSN 0361-0926. doi: 10.1080/03610927808827599. URL https://doi.org/10.1080/03610927808827599.

Clifford M Hurvich and Chih-Ling Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989. doi: 10.1093/biomet/76.2.297. URL http://dx.doi.org/10.1093/biomet/76.2.297.

References V

Jiahua Chen and Zehua Chen. Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771, 2008. doi: 10.1093/biomet/asn034. URL +http://dx.doi.org/10.1093/biomet/asn034.

Alexandre Belloni, Victor Chernozhukov, and Lie Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98 (4):791–806, 2011. URL http://biomet.oxfordjournals.org/content/98/4/791.abstract.

Alexandre Belloni, Victor Chernozhukov, and Lie Wang. Pivotal estimation via square-root Lasso in nonparametric regression. *The Annals of Statistics*, 42(2):757–788, 2014a. doi: 10.1214/14-AOS1204. URL http://dx.doi.org/10.1214/14-AOS1204.

References VI

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 67(2):301–320, 2005. ISSN 13697412. doi: 10.1111/j.1467-9868.2005.00503.x.

Peng Zhao and Bin Yu. On Model Selection Consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1248547.1248637.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3): 1436–1462, jun 2006. ISSN 0090-5364. doi: 10.1214/009053606000000281. URL http://projecteuclid.org/Dienst/getRecord?id=euclid.aos/1152540754/.

Hui Zou. The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006. ISSN 0162-1459. doi: 10.1198/016214506000000735.

References VII

Alexandre Belloni, Victor Chernozhukov, and Christian Hansen. High-Dimensional Methods and Inference on Structural and Treatment Effects. *Journal of Economic Perspectives*, 28(2):29–50, 2014b.

Alexandre Belloni, Victor Chernozhukov, and Christian Hansen. Inference on treatment effects after selection among high-dimensional controls. *Review of Economic Studies*, 81:608–650, 2014c. ISSN 1467937X.