Inference for parameters of interest after lasso model selection

David M. Drukker

Executive Director of Econometrics
Stata

Canadian Stata Users Group meeting
25 May 2019
High-dimensional models include too many potential covariates for a given sample size.

I have an extract of the data Sunyer et al. (2017) used to estimate the effect air pollution on the response time of primary school children.

\[ htime_i = \text{no2}_i \gamma + \mathbf{x}_i \beta + \epsilon_i \]

*htime* measure of the response time on test of child *i* (hit time)

*no2* measure of the pollution level in the school of child *i*

*\mathbf{x}_i* vector of control variables that might need to be included

There are 252 controls in \( \mathbf{x} \), but I only have 1,084 observations.

I cannot reliably estimate \( \gamma \) if I include all 252 controls.
Potential solutions

\[ htime_i = no2_i \gamma + x_i \beta + \epsilon_i \]

- I am willing to believe that the number of controls that I need to include is small relative to the sample size
  - This is known as a sparsity assumption
- Suppose that \( \tilde{x} \) contains the subset of \( x \) that must be included to get a good estimate of \( \gamma \) for the sample size that I have
- If I knew \( \tilde{x} \), I could use the model

\[ htime_i = no2_i \gamma + \tilde{x}_i \tilde{\beta} + \epsilon_i \]

So, the problem is that I don’t know which variables belong in \( \tilde{x} \) and which do not
\[ htime_i = no2_i \gamma + \tilde{x}_i \tilde{\beta} + \epsilon_i \]

- Now I have a covariate-selection problem
  - Which of the controls in \( x \) belong in \( \tilde{x} \)?
- Historically, I would use theory to decide which variables go into \( \tilde{x} \)
- Many researchers want to use data-based methods or machine-learning methods to perform the covariate selection
- Some post-covariate-selection estimators provide reliable inference for the few parameters of interest
  - Some do not
A naive approach

The “naive” solution is:

1. Always include the covariates of interest
2. Use covariate-selection to obtain estimate of which covariates are in $\tilde{x}$
   Denote estimate by $\hat{x}$
3. Use estimate $\hat{x}$ as if it contained the covariates in $\tilde{x}$
   regress htime no2 xhat
Unfortunately, naive estimators that use the selected covariates as if they were $\tilde{\mathbf{x}}$ provide unreliable inference in repeated samples.

- Covariate-selection methods make too many mistakes in estimating $\mathbf{x}$ when some of the coefficients are small in magnitude.
- Here is an example of small coefficient:
  - A coefficient with a magnitude between 1 and 2 times the standard error is small.
- If your model only approximates the functional form of the true model, there are approximation terms:
  - The coefficients on some of the approximating terms are most likely small.
Missing small-coefficient covariates matters

- It might seem that not finding covariates with small coefficients does not matter
  - But it does

- When some of the covariates have small coefficients, the distribution of the covariate-selection method is not sufficiently concentrated on the set of covariates that best approximates the process that generated the data
  - Covariate-selection methods will frequently miss the covariates with small coefficients causing omitted variable bias

- The random inclusion or exclusion of these covariates causes the distribution of the naive post-selection estimator to be not normal and makes the usual large-sample theory approximation invalid in theory and unreliable in finite samples
Beta-min condition

- The beta-min condition was invented to rule-out the existence of small coefficients in the model that best approximates the process that generated the data.
- Beta-min conditions are super restrictive and are widely viewed as not defensible.
Partialing-out estimators

\[ htime_i = \text{no}2_i \gamma + \tilde{x}_i \tilde{\beta} + \epsilon_i \]

- A series of seminal papers
  - Belloni, Chen, Chernozhukov, and Hansen (2012);
  - Belloni, Chernozhukov, and Hansen (2014);
  - Belloni, Chernozhukov, and Wei (2016a); and

derived a series of partialing-out estimators that provide reliable inference for \( \gamma \)

- These methods use covariate-selection methods to control for \( \tilde{x} \)
- The cost of using covariate-selection methods is that these partialing-out estimators do not produce estimates for \( \tilde{\beta} \)
I am going to provide lots of details, but here are two take aways

1. If you have time, use the cross-fit partialing-out estimator
   - xporegress, xpologit, xpopoisson, xpoivregress

2. If the cross-fit estimator takes too long, use either the partialing-out estimator
   - poregress, pologit, popoisson, poivregress
   or the double-selection estimator
   - dsregress, dslogit, dspoisson
. use breathe7

. local ccontrols "sev_home sev_sch age ppt age_start_sch oldsibl"
. local ccontrols "`ccontrols´ youngsibl no2_home ndvi_mn noise_sch"

. local fcontrols "grade sex lbweight lbfeed smokep"
. local fcontrols "`fcontrols´ feduc4 meduc4 overwt_who"
. describe htime no2_class `fcontrols´ `ccontrols´

<table>
<thead>
<tr>
<th>variable name</th>
<th>type</th>
<th>format</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>htime</td>
<td>double</td>
<td>%10.0g</td>
<td>ANT: mean hit reaction time (ms)</td>
</tr>
<tr>
<td>no2_class</td>
<td>float</td>
<td>%9.0g</td>
<td>Classroom NO2 levels (g/m3)</td>
</tr>
<tr>
<td>grade</td>
<td>byte</td>
<td>%9.0g</td>
<td>Grade in school</td>
</tr>
<tr>
<td>sex</td>
<td>byte</td>
<td>%9.0g</td>
<td>Sex</td>
</tr>
<tr>
<td>lbweight</td>
<td>float</td>
<td>%9.0g</td>
<td>1 if low birthweight</td>
</tr>
<tr>
<td>lbfeed</td>
<td>byte</td>
<td>%19.0f</td>
<td>duration of breastfeeding</td>
</tr>
<tr>
<td>smokep</td>
<td>byte</td>
<td>%3.0f</td>
<td>1 if smoked during pregnancy</td>
</tr>
<tr>
<td>feduc4</td>
<td>byte</td>
<td>%17.0g</td>
<td>Paternal education</td>
</tr>
<tr>
<td>meduc4</td>
<td>byte</td>
<td>%17.0g</td>
<td>Maternal education</td>
</tr>
<tr>
<td>overwt_who</td>
<td>byte</td>
<td>%32.0g</td>
<td>WHO/CDC-overweight 0:no/1:yes</td>
</tr>
<tr>
<td>sev_home</td>
<td>float</td>
<td>%9.0g</td>
<td>Home vulnerability index</td>
</tr>
<tr>
<td>sev_sch</td>
<td>float</td>
<td>%9.0g</td>
<td>School vulnerability index</td>
</tr>
<tr>
<td>age</td>
<td>float</td>
<td>%9.0g</td>
<td>Child´s age (in years)</td>
</tr>
<tr>
<td>ppt</td>
<td>double</td>
<td>%10.0g</td>
<td>Daily total precipitation</td>
</tr>
<tr>
<td>age_start_sch</td>
<td>double</td>
<td>%4.1f</td>
<td>Age started school</td>
</tr>
<tr>
<td>oldsibl</td>
<td>byte</td>
<td>%1.0f</td>
<td>Older siblings living in house</td>
</tr>
<tr>
<td>youngsibl</td>
<td>byte</td>
<td>%1.0f</td>
<td>Younger siblings living in house</td>
</tr>
<tr>
<td>no2_home</td>
<td>float</td>
<td>%9.0g</td>
<td>Residential NO2 levels (g/m3)</td>
</tr>
<tr>
<td>ndvi_mn</td>
<td>double</td>
<td>%10.0g</td>
<td>Home greenness (NDVI), 300m buffer</td>
</tr>
<tr>
<td>noise_sch</td>
<td>float</td>
<td>%9.0g</td>
<td>Measured school noise (in dB)</td>
</tr>
</tbody>
</table>
. `xpregress` `htime` `no2_class`, `controls(i.(`fcontrols´) c.(`ccontrols´) ///
>       i.(`fcontrols´)#c.(`ccontrols´))`

Cross-fit fold 1 of 10 ...
Estimating lasso for `htime` using plugin
Estimating lasso for `no2_class` using plugin
(output omitted)

Cross-fit fold 10 of 10 ...
Estimating lasso for `htime` using plugin
Estimating lasso for `no2_class` using plugin

Cross-fit partialed-out linear model

<table>
<thead>
<tr>
<th></th>
<th>Number of obs</th>
<th>Number of controls</th>
<th>Number of selected controls</th>
<th>Number of folds in cross-fit</th>
<th>Number of resamples</th>
<th>Wald chi2(1)</th>
<th>Prob &gt; chi2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of obs</td>
<td>1,084</td>
<td>252</td>
<td>15</td>
<td>10</td>
<td>1</td>
<td>25.36</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

| `htime` | `Robust` | `Coef.` | `Std. Err.` | `z`  | `P>|z|` | `[95% Conf. Interval]` |
|----------|-----------|----------|-------------|------|--------|-----------------------|
| `no2_class` |          | 2.353006 | .4672161    | 5.04 | 0.000  | 1.437279  3.268732 |

Note: Chi-squared test is a Wald test of the coefficients of the variables of interest jointly equal to zero.

Another microgram of NO2 per cubic meter increases the mean reaction time by 2.35 milliseconds.
. poregress htime no2_class, controls(i(`fcontrols´) c(`ccontrols´) ///
  i(`fcontrols´)#c(`ccontrols´))

Estimating lasso for htime using plugin
Estimating lasso for no2_class using plugin

Partialed-out linear model

<table>
<thead>
<tr>
<th></th>
<th>Number of obs</th>
<th>Number of controls</th>
<th>Number of selected controls</th>
<th>Wald chi2(1)</th>
<th>Prob &gt; chi2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1,084</td>
<td>252</td>
<td>11</td>
<td>24.45</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

| htime  | Coef.  | Robust Std. Err. | z     | P>|z|   | [95% Conf. Interval] |
|--------|--------|------------------|-------|-------|----------------------|
| no2_class | 2.286149 | .4623136 | 4.95 | 0.000 | 1.380031 3.192267 |

Note: Chi-squared test is a Wald test of the coefficients of the variables of interest jointly equal to zero.

Another microgram of NO$_2$ per cubic meter increases the mean reaction time by 2.29 milliseconds.
Describe estimators implemented in `poregress` and `xporegress`.

Estimators use the least absolute shrinkage and selection operator (lasso) to perform covariate-selection.

I discuss lasso details after describing estimators.

For now just think of lasso as covariate-selection method that works when the number of potential covariates is large.

The number of potential covariates $p$ can be greater than the number of observations $N$. 
Consider model

\[ y = d \gamma + x \beta + \epsilon \]

For simplicity, \( d \) is a single variable, all methods handle multiple variables

I discuss a linear model

Nonlinear models have similar methods that involve more details
PO estimator for linear model (I)

\[ y = d \gamma + x \beta + \epsilon \]

1. Use a lasso of \( y \) on \( x \) to select covariates \( \tilde{x}_y \) that predict \( y \)
2. Regress \( y \) on \( \tilde{x}_y \) and let \( \tilde{y} \) be residuals from this regression
3. Use a lasso of \( d \) on \( x \) to select covariates \( \tilde{x}_d \) that predict \( d \)
4. Regress \( d \) on \( \tilde{x}_d \) and let \( \tilde{d} \) be residuals from this regression
5. Regress \( \tilde{y} \) on \( \tilde{d} \) to get estimate and standard error for \( \gamma \)

- Only the coefficient on \( d \) is estimated
- Not estimating \( \beta \) can be viewed as the cost of getting reliable estimates of \( \gamma \) that are robust to the mistakes that model-selection techniques make
PO estimator for linear model (II)

\[ y = d\gamma + x\beta + \epsilon \]

1. Use a lasso of \( y \) on \( x \) to select covariates \( \tilde{x}_y \) that predict \( y \)
2. Regress \( y \) on \( \tilde{x}_y \) and let \( \tilde{y} \) be residuals from this regression
3. Use a lasso of \( d \) on \( x \) to select covariates \( \tilde{x}_d \) that predict \( d \)
4. Regress \( d \) on \( \tilde{x}_d \) and let \( \tilde{d} \) be residuals from this regression
5. Regress \( \tilde{y} \) on \( \tilde{d} \) to get estimate and standard error for \( \gamma \)

This is an extension of the partialing-out method for obtaining the ordinary least squares (OLS) estimate for the coefficient and standard error on \( d \) (Also known as the result of the Frisch-Waugh-Lovell theorem)
\[ y = d\gamma + x\beta + \epsilon \]

1. Use a lasso of \( y \) on \( x \) to select covariates \( \tilde{x}_y \) that predict \( y \)
2. Regress \( y \) on \( \tilde{x}_y \) and let \( \tilde{y} \) be residuals from this regression
3. Use a lasso of \( d \) on \( x \) to select covariates \( \tilde{x}_d \) that predict \( d \)
4. Regress \( d \) on \( \tilde{x}_d \) and let \( \tilde{d} \) be residuals from this regression
5. Regress \( \tilde{y} \) on \( \tilde{d} \) to get estimate and standard error for \( \gamma \)

- Heuristically, the moment conditions used in step 5 are unrelated to the selected covariates
- Formally, the moments conditions used in step 5 have been orthogonalized, or “immunized” to small mistakes in covariate selection

- Chernozhukov, Hansen, and Spindler (2015a); and Chernozhukov, Hansen, and Spindler (2015b)
Cross-fitting is also known as double machine learning (DML).

It uses split-sample techniques on PO estimators:
- to weaken the sparsity condition
- to get better finite sample performance

Split-sample techniques further reduce the impact of covariate selection on the estimator for $\gamma$. 
Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins (2018) discusses

- Why sample-splitting techniques applied to naive machine-learning/covariate-selection estimators do not provide reliable inference for $\gamma$ in repeated samples

Heuristically, the machine-learning estimators do not converge fast enough to remove the correlation between covariate of interest and the out-of-sample errors in the term predicted by the machine-learning method.
Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins (2018) discusses PO estimators simplify the problem and their distributions depend on the correlation between partialed-out covariate of interest and the errors in the term predicted by the machine-learning method.

- Naive estimator depends correlation between the covariate of interest and the errors in the term predicted by the machine-learning method.

- Sample-splitting gets better properties by depending on the out-of-sample correlation between partialed-out covariate of interest and the errors in the term predicted by the machine-learning method instead of the in-sample correlation.
1. Split data into samples A and B

2. Using the data in sample A
   1. Use a lasso of $y$ on $x$ to select covariates $\tilde{x}_y$ that predict $y$
   2. Regress $y$ on $\tilde{x}_y$ and let $\tilde{\beta}_A$ be the estimated coefficients
   3. Use a lasso of $d$ on $x$ to select covariates $\tilde{x}_d$ that predict $d$
   4. Regress $d$ on $\tilde{x}_d$ and let $\tilde{\delta}_A$ be the estimated coefficients

3. Using the data in sample B
   1. Fill in the residuals for $\tilde{y} = y - \tilde{x}_y\tilde{\beta}_A$
   2. Fill in the residuals for $\tilde{d} = d - \tilde{x}_d\tilde{\delta}_A$

4. Using the data in sample B
   1. Use a lasso of $y$ on $x$ to select covariates $\tilde{x}_y$ that predict $y$
   2. Regress $y$ on $\tilde{x}_y$ and let $\tilde{\beta}_B$ be the estimated coefficients
   3. Use a lasso of $d$ on $x$ to select covariates $\tilde{x}_d$ that predict $d$
   4. Regress $d$ on $\tilde{x}_d$ and let $\tilde{\delta}_B$ be the estimated coefficients

5. Using the data in sample A
   1. Fill in the residuals for $\tilde{y} = y - \tilde{x}_y\tilde{\beta}_B$
   2. Fill in the residuals for $\tilde{d} = d - \tilde{x}_d\tilde{\delta}_B$

6. Regress $\tilde{y}$ on $\tilde{d}$ to get estimates for $\gamma$
What’s a lasso?

- The linear lasso solves

\[
\hat{\beta} = \arg \min_{\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i \beta') + \lambda \sum_{j=1}^{k} \omega_j |\beta_j| \right\}
\]

where

- \( \lambda > 0 \) is the lasso penalty parameter
- \( x \) contains the \( p \) potential covariates
- the \( \omega_j \) are parameter-level weights known as penalty loadings
What’s a lasso?

\[
\hat{\beta} = \arg \min_{\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i \beta') + \lambda \sum_{j=1}^{k} \omega_j |\beta_j| \right\}
\]

- As \( \lambda \) grows, the coefficients get “shrunk” towards zero.
- The kink in the absolute value function causes some of the elements of \( \hat{\beta} \) to be zero at the solution for some values of \( \lambda \).
- There is a finite value of \( \lambda = \lambda_{\text{max}} \) for which all the estimated coefficients are zero.
- As \( \lambda \) decreases from \( \lambda_{\text{max}} \), the number of nonzero coefficients increases.
  - If \( p < n \), you obtain the (unpenalized) OLS estimates at \( \lambda = 0 \).
What’s a lasso?

\[
\hat{\beta} = \arg \min_{\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i' \beta) + \lambda \sum_{j=1}^{k} \omega_j |\beta_j| \right\}
\]

- For \( \lambda \in (0, \lambda_{max}) \) some of the estimated coefficients are exactly zero and some of them are not zero.
- This is how the lasso works as a covariate-selection method
  - Covariates with estimated coefficients of zero are excluded
  - Covariates with estimated coefficients that not zero are included
Choosing $\lambda$

- You must choose $\lambda$ before you use the lasso to perform covariate selection.
- Three methods for selecting $\lambda$ are:
  1. Plug-in estimators
    - These estimators are the default in the PO, DS, and XPO commands.
  2. Cross-validation
  3. The adaptive lasso
Plug-in based lasso

Plug-in estimators find the value of the $\lambda$ that is large enough to dominate the estimation noise.

- see Belloni, Chernozhukov, and Wei (2016b); Belloni, Chen, Chernozhukov, and Hansen (2012); and Bickel et al. (2009).
- Belloni, Chernozhukov, and Wei (2016b) and Belloni, Chen, Chernozhukov, and Hansen (2012) show that a lasso with their plug-in estimator achieves an optimal bound on the number of covariates it will include.
- In practice, their bound means that a plug-in-based lasso will include the important covariates and that it will not include many covariates that do not belong in the model.
Cross-validated lasso

- Cross-validation (CV) finds the $\hat{\beta}$ that minimizes the out-of-sample prediction error.

- CV is widely used, but it is not the best method when using lasso as a covariate-selection method in a PO, XPO, or DS estimator.
  - CV tends to choose a $\lambda$ that causes lasso to include variables whose coefficients are zero in the model that best approximates the true data generating process.
  - This over-selection tendency can cause a CV-based PO, DS, XPO estimator to have poor coverage properties.

(Although the XPO estimators are more robust to this problem that PO and DS estimators)
Cross-validated lasso

- See Hastie, Tibshirani, and Wainwright (2015) for lots about how CV lasso is implemented.
- See Chetverikov, Liao, and Chernozhukov (2017) for some technical results that could explain the tendency of the cross-validated lasso to include many covariates that do not belong in the model.
- See Bühlmann and Van de Geer (2011) for some discussions of the tendency of cross-validated lasso to over select.
Adaptive lasso

- The adaptive lasso tends to include more zero-coefficient covariates than a plug-in based lasso and fewer than a cross-validated lasso.
- The adaptive lasso is a multistep version of CV.
  - The first step is CV.
    - The second step does CV among the covariates selected in the first step.
    - In the second step, the penalty loadings are set to the inverse of the first-step estimates coefficients.
  - Covariate with larger coefficients are more likely to be included in the second step.
Conclusion

- If you have a model like

$$E[y|d, x] = G(d\gamma + x\beta)]$$

where
- $G()$ is the functional form implied by a linear regression, a logit regression, a Poisson regression
- $d$ contains a few known covariates
- $x$ contains many potential controls

You can use
- `xporegress`, `xpologit`, `xpopoisson`, `poregress`, `pologit`, or `popoisson`, to estimate $\gamma$

- `xpoivregress` and `poivregress` estimate $\gamma$ for linear models with endogenous covariates when there are many potential instruments and many potential controls.


