Lasso and machine learning using Stata

StataCorp LLC

December 5, 2019

Why do we care?

- For the same reason we always care: Extracting signal from noise.

Why do we care?

- For the same reason we always care: Extracting signal from noise.
- Noise is larger. We have access to more data than we have ever had.
- Methods and theory need to adapt to new problems.

Why do we care?

- For the same reason we always care: Extracting signal from noise.
- Noise is larger. We have access to more data than we have ever had.
- Methods and theory need to adapt to new problems.
- Lasso type methods are one answer (popular)
  - Prediction originally
  - Estimation of effects recently

Lasso

- Model selection and parameter estimation simultaneoulsy

Lasso

- Model selection and parameter estimation simultaneoulsy
  - Model selection allows more covariates than observations in data
  - AIC or BIC 2^M. With 10 regressors you have 1,024 candidate models.
  - You obtain coefficients, $\widehat{\beta}$, that can be used for prediction
  - Regularized (penalized) coefficients avoid overfitting (ridge)

Lasso

- Model selection and parameter estimation simultaneoulsy
  - Model selection allows more covariates than observations in data
  - AIC or BIC 2^M. With 10 regressors you have 1,024 candidate models.
  - You obtain coefficients, $\widehat{\beta}$, that can be used for prediction
  - Regularized (penalized) coefficients avoid overfitting (ridge)
- Original Tibshirani (1996). Numerous variations:
  - Elastic net
  - Square-root lasso
  - Adaptive lasso

A brief introduction to Lasso

Mathematically

- Think about linear regression

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - x_i'\beta\right)^2$$

Mathematically

- Think about linear regression

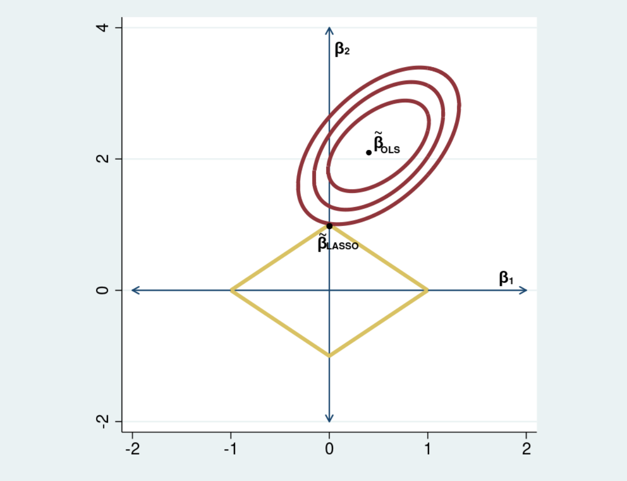
$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - x_i'\beta\right)^2$$

- Lasso minimizes (constrained optimization)

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - x_i'\beta\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

  - $\lambda = 0$ Back to regression (unbiased)
  - $\lambda = \infty$ No coefficients
  - $0 < \lambda < \infty$ biased but avoids overfitting and is good for prediction
  - $|\beta_j|$ penalizes additional coefficients

Graphically



Much more

- Beyond linear
  - Logit
  - Probit
  - Poisson
  - Nonparametric (more on this later)

Much more

- Beyond linear
  - Logit
  - Probit
  - Poisson
  - Nonparametric (more on this later)
- Beyond absolute value penalty (Elastic net)

$$\lambda \sum_{j=1}^{p} \left\{ \alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right\}$$

  - $\alpha = 1$ Lasso
  - $\alpha = 0$ Ridge regression

Selecting λ

- Cross-validation and adaptive lasso (Good for prediction)
  - Tends to overselect
  - Minimizes out of sample prediction error.
- Plugin (Good for inference)
  - Tends to underselect
  - Closed form formula to dominate noise level of problem

A general framework

- The model is given by:

$$\begin{aligned} y_i &= g(x_i) + \varepsilon_i \\ E(\varepsilon_i | x_i) &= 0 \end{aligned}$$

- The function $g(x_i)$ is unknown

A general framework

- The model is given by:

$$\begin{aligned} y_i &= g(x_i) + \varepsilon_i \\ E(\varepsilon_i | x_i) &= 0 \end{aligned}$$

- The function $g(x_i)$ is unknown
- Emphasizes the idea that lasso is an approximation
  - This is even true if the unknown function is linear $g(x_i) = x_i'\beta$
  - If $g(x_i) = x_i'\beta$ you might miss some small coefficients

A general framework

- The model is given by:

$$\begin{aligned} y_i &= g(x_i) + \varepsilon_i \\ E(\varepsilon_i | x_i) &= 0 \end{aligned}$$

- The function $g(x_i)$ is unknown
- Emphasizes the idea that lasso is an approximation
  - This is even true if the unknown function is linear $g(x_i) = x_i'\beta$
  - If $g(x_i) = x_i'\beta$ you might miss some small coefficients
- Embrace model selection error

A general framework: approximating an unknown function

- Belloni, Chernozhukov, and Hansen suggest approximating the unknown function linearly
  - Series estimation: polynomials, natural-splines, B-splines, furier series, etc.

A general framework: approximating an unknown function

- Belloni, Chernozhukov, and Hansen suggest approximating the unknown function linearly
  - Series estimation: polynomials, natural-splines, B-splines, furier series, etc.
- For example:

$$
\begin{aligned}
\widehat{g}(x_i) &= \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots \beta_5 x_i^5 \\
\widehat{g}(x_i) &= f_i' \beta
\end{aligned}
$$

A general framework: Assumptions and workflow

- Assumptions
  - Conjectured f_i can be large, i.e. the dimensions of β are large. Even larger that the sample size n.

A general framework: Assumptions and workflow

- Assumptions
  - Conjectured f_i can be large, i.e. the dimensions of β are large. Even larger that the sample size n.
  - The elements in the best approximating function f_{i0} is smaller than n. Sparsity.
  - You are minimizing approximation error not going after a true model
- Workflow
  - Conjecture a large dimensional approximating model
  - Choose a method to select λ. Cross-validation is the default.
  - Get approximating function for prediction

Lasso for prediction using Stata

Example: Predicting housing value

- Predict the value of a house
- Data from American Housing Survey

Variables

```
. keep if state==20
(871,947 observations deleted)
. tab state
```

State code	Freq.	Percent	Cum.
Kansas/KS	9,652	100.00	100.00
Total	9,652	100.00	

Variables

```
. keep if state==20
(871,947 observations deleted)
. tab state
```

State code	Freq.	Percent	Cum.
Kansas/KS	9,652	100.00	100.00
Total	9,652	100.00	

```
. describe value lotsize bedrooms rooms bage vpperson ptaxes insurance
              storage   display    value
variable name   type    format     label      variable label
---------------------------------------------------------------------------
value           long    %10.0g                Property value in $ (top coded)
lotsize         byte    %36.0g     lsvalues   Lot size
bedrooms        byte    %10.0g                Number of bedrooms
rooms           byte    %10.0g                Number of rooms
bage            float   %9.0g                 Building age
vpperson        float   %9.0g               * Vehicles per person
ptaxes          float   %9.0g                 Property taxes; top coded at
                                                $10,000
insurance       float   %10.0g              * yearly insurance in $1,000
```

Discrete covariates in Stata

```
. local discrete lotsize bedrooms rooms
. quietly mean i.(`discrete´)
. mean i.(lotsize bedrooms)
Mean estimation                              Number of obs    =      9,652
```

	Mean	Std. Err.	[95% Conf.	Interval]
lotsize				
House on less than one acre	.7662661	.0043079	.7578217	.7747104
House on one to less than..	.1422503	.0035557	.1352805	.1492202
House on ten or more acres	.0914836	.0029346	.0857312	.0972361
bedrooms				
0	.0021757	.0004743	.001246	.0031054
1	.0297348	.001729	.0263456	.0331239
2	.2260671	.0042578	.217721	.2344133
3	.4391836	.0050518	.429281	.4490862
4	.2253419	.0042529	.2170052	.2336786
5	.0661003	.0025291	.0611427	.0710579
10	.0113966	.0010805	.0092787	.0135145

Continuous covariates in Stata

```
. local continuous bage vpperson ptaxes insurance
. quietly mean c.(`continuous´)##c.(`continuous´)##c.(`continuous´)
. mean c.(bage vpperson)##c.(bage vpperson)##c.(bage vpperson)
Mean estimation                          Number of obs   =      9,652
```

	Mean	Std. Err.	[95% Conf.	Interval]
bage	46.29351	.237585	45.8278	46.75923
vpperson	.9654589	.0065045	.9527087	.9782091
c.bage#c.bage	2687.856	21.30915	2646.086	2729.627
c.bage#c.vpperson	44.68529	.4281789	43.84597	45.52461
c.vpperson#c.vpperson	1.340433	.0213607	1.298561	1.382304
c.bage#c.bage#c.bage	172171.8	1690.535	168858	175485.6
c.bage#c.bage#c.vpperson	2590.722	32.05076	2527.896	2653.549
c.bage#c.vpperson#c.vpperson	63.86	1.270451	61.36965	66.35035
c.vpperson#c.vpperson#c.vpperson	2.478089	.0897912	2.302079	2.654098

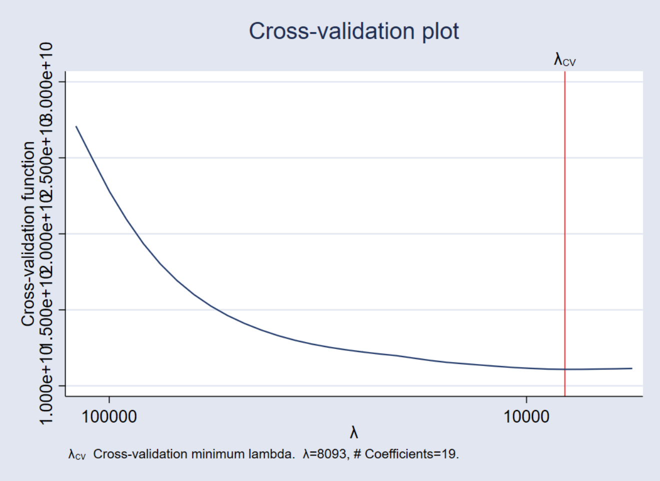
Estimation

```
. local cubic c.(`continuous´)##c.(`continuous´)##c.(`continuous´)
. local dinter i.(`discrete´)#i.(`discrete´)
. set seed 111
. lasso linear value (`discrete´)##(`cubic´) `dinter´
Grid value 1:     lambda = 120184.5   no. of nonzero coef. =       0
Folds: 1...5....10   CVF = 2.71e+10
(output omitted ...)
Grid value 34:    lambda = 5578.472   no. of nonzero coef. =      32
Folds: 1...5....10   CVF = 1.11e+10
... cross-validation complete ... minimum found
Lasso linear model                          No. of obs         =      7,657
                                            No. of covariates  =      1,030
Selection: Cross-validation                 No. of CV folds    =         10
```

ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
1	first lambda	120184.5	0	0.0056	2.71e+10
29	lambda before	8882.505	18	0.5925	1.11e+10
* 30	selected lambda	8093.408	19	0.5931	1.11e+10
31	lambda after	7374.412	21	0.5930	1.11e+10
34	last lambda	5578.472	32	0.5909	1.11e+10

* lambda selected by cross-validation.

cvplot



Cross-validation plot
λCV
Cross-validation function
1.000e+10
1.500e+10
2.000e+10
2.500e+10
3.000e+10
100000
10000
λ
λCV Cross-validation minimum lambda. λ=8093, # Coefficients=19.

lassoknots

```
. lassoknots
─────────────────────────────────────────────────────────────────────────────
     |                  No. of   CV mean |
     |                 nonzero     pred. |   Variables (A)dded, (R)emoved,
  ID |     lambda        coef.     error |        or left (U)nchanged
─────────────────────────────────────────────────────────────────────────────
   2 |  109507.7             2  2.49e+10 | A ptaxes   c.ptaxes#c.ptaxes
   4 |  90915.19             3  2.09e+10 | A c.ptaxes#c.insurance
  11 |  47403.26             4  1.41e+10 | A c.vpperson#c.ptaxes#c.ptaxes
  16 |  29770.63             5  1.25e+10 | A c.ptaxes#c.ptaxes#c.insurance
  20 |  20519.74             6  1.20e+10 | A c.ptaxes#c.ptaxes#c.ptaxes
  21 |  18696.82             7  1.18e+10 | A insurance
     |                                   |   1.lotsize#c.bage
  21 |  18696.82             7  1.18e+10 | R c.ptaxes#c.ptaxes
  22 |  17035.85             9  1.17e+10 | A 11.rooms#c.bage#c.ptaxes#c.ptaxes
     |                                   |   4.bedrooms#c.vpperson#c.ptaxes#
     |                                   |   c.ptaxes
  24 |  14143.46             9  1.15e+10 | A 3.lotsize#c.insurance
  24 |  14143.46             9  1.15e+10 | R c.ptaxes#c.insurance
  (output omitted ...)
  29 |  8882.505            18  1.11e+10 | A 3.lotsize
* 30 |  8093.408            19  1.11e+10 | A c.bage#c.ptaxes#c.ptaxes
  (output omitted...)
  33 |  6122.366            28  1.11e+10 | A 4.bedrooms
     |                                   |   1.lotsize#c.bage#c.ptaxes#c.ptaxes
  34 |  5578.472            32  1.11e+10 | A 10.rooms 1.lotsize#c.vpperson
     |                                   |   13.rooms#c.ptaxes#c.ptaxes#c.ptaxes
     |                                   |   4.rooms#c.insurance#c.insurance#
     |                                   |   c.insurance
─────────────────────────────────────────────────────────────────────────────
* lambda selected by cross-validation.
```

Give me my lambda

```
 . lassoselect id=24
ID = 24  lambda = 14143.46 selected
```

Give me my lambda

```
. lassoselect id=24
ID = 24  lambda = 14143.46 selected

. lasso
Lasso linear model                          No. of obs         =      7,657
                                            No. of covariates  =      1,030
Selection: User                             No. of CV folds    =         10
```

ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
1	first lambda	120184.5	0	0.0056	2.71e+10
23	lambda before	15522.43	9	0.5762	1.15e+10
* 24	selected lambda	14143.46	9	0.5795	1.15e+10
25	lambda after	12886.99	11	0.5827	1.14e+10
34	last lambda	5578.472	32	0.5909	1.11e+10

Give me my lambda

```
. lassoselect id=24
ID = 24  lambda = 14143.46 selected

. lasso
Lasso linear model                          No. of obs         =      7,657
                                            No. of covariates =      1,030
Selection: User                             No. of CV folds    =         10
```

ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
1	first lambda	120184.5	0	0.0056	2.71e+10
23	lambda before	15522.43	9	0.5762	1.15e+10
* 24	selected lambda	14143.46	9	0.5795	1.15e+10
25	lambda after	12886.99	11	0.5827	1.14e+10
34	last lambda	5578.472	32	0.5909	1.11e+10

splitsample

```
. splitsample, generate(sample) split(0.60 0.40)
. label define splits 1 "training" 2 "validation"
. label value sample splits
```

Estimators

- linear lasso using:
  - cross-validation
  - adaptive lasso
  - plugin-method
- ridge regression

Estimation

```
. quietly lasso linear value (`discrete´)##(`cubic´) `dinter´ if sample==1
. estimates store cv
. generate esample = e(sample)
. quietly lasso linear value (`discrete´)##(`cubic´) `dinter´          ///
>         if sample==1 & esample==1, selection(adaptive)
. estimates store adaptive
. quietly lasso linear value (`discrete´)##(`cubic´) `dinter´          ///
>         if sample==1 & esample==1, selection(plugin)
. estimates store plugin
. quietly elasticnet linear value (`discrete´)##(`cubic´) `dinter´     ///
>         if sample==1 & esample==1, alpha(0)
. estimates store ridge
```

Evaluating out-of-sample prediction

```
. lassogof cv adaptive plugin ridge, over(sample)
Penalized coefficients
```

Name	sample	MSE	R-squared	Obs
cv				
	training	1.08e+10	0.6230	4,573
	validation	1.03e+10	0.5917	3,084
adaptive				
	training	1.00e+10	0.6491	4,573
	validation	1.08e+10	0.5758	3,084
plugin				
	training	1.20e+10	0.5798	4,573
	validation	1.08e+10	0.5732	3,084
ridge				
	training	2.84e+10	0.0044	4,573
	validation	2.52e+10	0.0040	3,084

Lasso for inference

Asymptotic metaphor

- Get multiple draws from the population (true model)

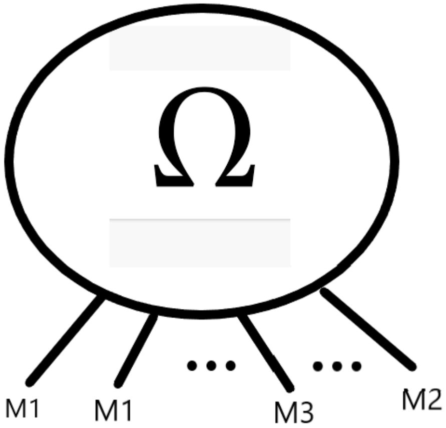
Asymptotic metaphor

- Get multiple draws from the population (true model)
    - Every time you have the same covariates
    - Asymptotically normal and centered around the true value

Asymptotic metaphor

- Get multiple draws from the population (true model)
  - Every time you have the same covariates
  - Asymptotically normal and centered around the true value
- With model selection
  - Covariates are different every time
  - Distribution is not asymptotically normal

Metaphor is broken

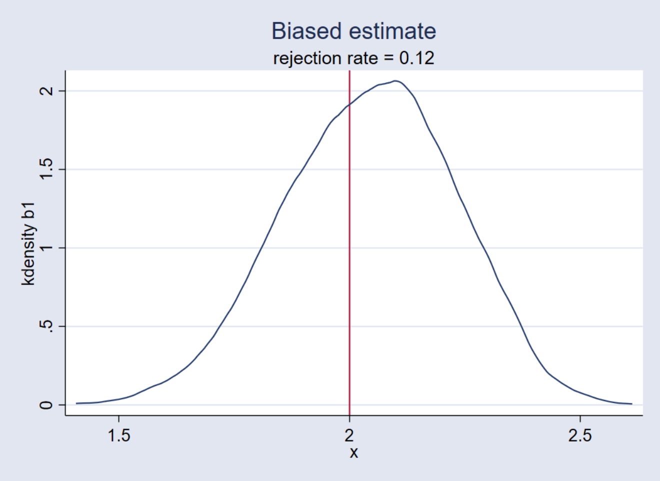


Simulated example

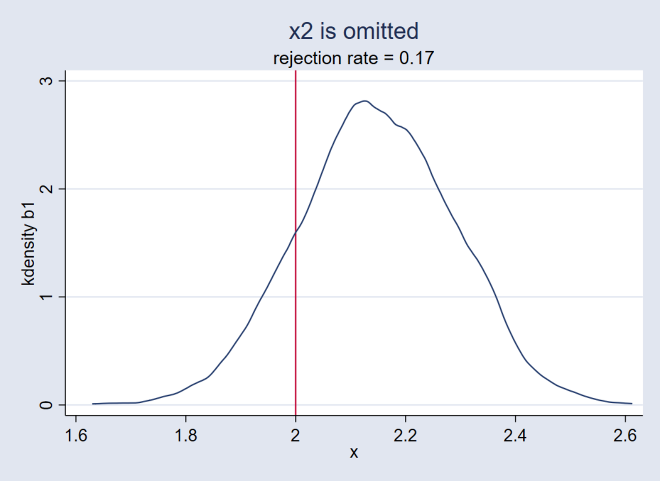
$$y = 1 + 2x_1 + .3x_2 + \varepsilon$$

- ε is a standardized chi-squared
- x_1 and x_2 are correlated
- The coefficient on x_2 is “small”
- x_2 is going to be omitted sometimes

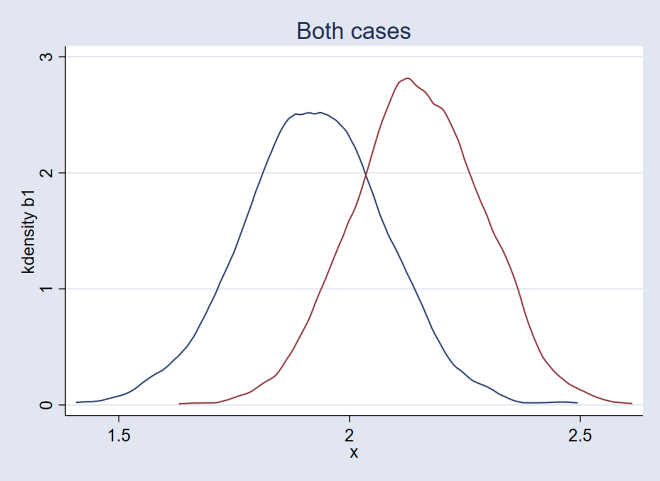
Asymptotic distribution 3000 repetitions



Distribution when x_2 is omitted



Distribution is bimodal



What have we learned

- Standard errors when using model selection are not normal
- Post-model selection (using selected variables and fitting a model) is unjustified
  - Covariates are correlated
  - Some covariates are “small” and belong in the model

What have we learned

- Standard errors when using model selection are not normal
- Post-model selection (using selected variables and fitting a model) is unjustified
    - Covariates are correlated
    - Some covariates are "small" and belong in the model
- We need to account for model selection

Lasso for inference (intuition)

- Want parameters associated with a fixed set of covariates
- All other parameters are controls (nuisance parameter, may be large)
- There is no free lunch:
  - We get reliable inference for the set of fixed covariates
  - We get no inference for the nuisance parameter
- Useful and justified
- You have to have a “reasonable” approximation for the nuisance

Simulated data example

```
. set seed 111
. set obs 3000
number of observations (_N) was 0, now 3,000
. generate a  = (rchi2(5)-5)/sqrt(10)
. generate x1 = (rchi2(5)-5)/sqrt(10) + a
. generate x2 = (rchi2(5)-5)/sqrt(10) + a
. generate x3 = (rchi2(5)-5)/sqrt(10) + a
. generate x4 = (rchi2(5)-5)/sqrt(10) + a
. generate x5 = (rchi2(5)-5)/sqrt(10) + a
. generate b  = 1+ int(runiform()*4 + a)
. generate d  = runiformint(2,5)
. generate e  = (rchi2(5)-5)
. generate y  = 1 + x1 - sin(3*(x2-x3 + x4))*b - b + e
. local cubic c.(x2 x3 x4 x5)##c.(x2 x3 x4 x5)##c.(x2 x3 x4 x5)
```

Simulated data example: Estimation results

```
. poregress y x1, controls(`cubic´##i.b##i.d)
(output omitted ...)
Estimating lasso for y using plugin
(output omitted ...)
Estimating lasso for x1 using plugin
(output omitted ...)
Partialing-out linear model          Number of obs               =      3,000
                                     Number of controls          =      1,749
                                     Number of selected controls =         14
                                     Wald chi2(1)                =     221.03
                                     Prob > chi2                 =     0.0000

------------------------------------------------------------------------------
             |               Robust
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   1.017465   .0684369    14.87   0.000     .8833308    1.151599
------------------------------------------------------------------------------
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.
```

Partialing out regression

1. For each of the covariates of interest (d_j) run lasso on d_j and controls

Partialing out regression

1. For each of the covariates of interest (d_j) run lasso on d_j and controls
2. Regress d_j on selected covariates and get residuals

Partialing out regression

1. For each of the covariates of interest (d_j) run lasso on d_j and controls
2. Regress d_j on selected covariates and get residuals
3. Run lasso of dependent variable on controls

Partialing out regression

1. For each of the covariates of interest (d_j) run lasso on d_j and controls
2. Regress d_j on selected covariates and get residuals
3. Run lasso of dependent variable on controls
4. Regress dependent variable on selected covariates from (3) and get residuals

Partialing out regression

1. For each of the covariates of interest (d_j) run lasso on d_j and controls
2. Regress d_j on selected covariates and get residuals
3. Run lasso of dependent variable on controls
4. Regress dependent variable on selected covariates from (3) and get residuals
5. Run gmm (regression) of residuals from (3) on residuals from (2)

Partialing out regression

1. For each of the covariates of interest (d_j) run lasso on d_j and controls
2. Regress d_j on selected covariates and get residuals
3. Run lasso of dependent variable on controls
4. Regress dependent variable on selected covariates from (3) and get residuals
5. Run gmm (regression) of residuals from (3) on residuals from (2)

- You are familiar with this, it is partialing out

Partialing out regression

1. For each of the covariates of interest (d_j) run lasso on d_j and controls
2. Regress d_j on selected covariates and get residuals
3. Run lasso of dependent variable on controls
4. Regress dependent variable on selected covariates from (3) and get residuals
5. Run gmm (regression) of residuals from (3) on residuals from (2)

- You are familiar with this, it is partialing out
- This is regression partialing as in FWL

Partialing out regression

1. For each of the covariates of interest (d_j) run lasso on d_j and controls
2. Regress d_j on selected covariates and get residuals
3. Run lasso of dependent variable on controls
4. Regress dependent variable on selected covariates from (3) and get residuals
5. Run gmm (regression) of residuals from (3) on residuals from (2)

- You are familiar with this, it is partialing out
- This is regression partialing as in FWL

Alternatives

- Continuous outcome
  - poregress
  - dsregress
  - xporegress
- Binary outcome
  - pologit
  - dslogit
  - xpologit
- Count outcome
  - popoisson
  - dspoisson
  - xpopoisson

Alternatives

- Continuous outcome
  - poregress
  - dsregress
  - xporegress
- Binary outcome
  - pologit
  - dslogit
  - xpologit
- Count outcome
  - popoisson
  - dspoisson
  - xpopoisson
- contrasts, marginal effects, odds ratios, incidence rates

There is more

- Instrumental variable regression (endogeneity)
- Lasso to select exogenous controls and instruments
- Tools
    - poivregress
    - xpoivregress

Labor market data

```
. // Exogenous
. describe exper age husage kidslt6 kidsge6 city

              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
exper           byte    %9.0g                 actual labor mkt exper
age             byte    %9.0g                 woman´s age in yrs
husage          byte    %9.0g                 husband´s age
kidslt6         byte    %9.0g                 # kids < 6 years
kidsge6         byte    %9.0g                 # kids 6-18
city            byte    %9.0g                 =1 if live in SMSA
. // Instruments
. describe motheduc fatheduc huseduc

              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
motheduc        byte    %9.0g                 mother´s years of schooling
fatheduc        byte    %9.0g                 father´s years of schooling
huseduc         byte    %9.0g                 husband´s years of schooling
```

Set up

```
. local exog exper age husage kidslt6 kidsge6 city
. local interex c.(`exog´)##c.(`exog´)
. local ins motheduc fatheduc huseduc
. local insex c.(`ins´)##c.(`ins´)
```

Estimation

```
. xpoivregress lwage (educ = `insex´), controls(`interex´) rseed(12345)
Cross-fit fold 1 of 10 ...
Estimating lasso for lwage using plugin
note: c.city#c.city dropped because of collinearity with another variable
Estimating lasso for educ using plugin
note: c.city#c.city dropped because of collinearity with another variable
Cross-fit fold 2 of 10 ...
(output omitted ...)
Cross-fit partialing-out              Number of obs                =        428
IV linear model                       Number of controls           =         27
                                      Number of instruments        =          9
                                      Number of selected controls  =          4
                                      Number of selected instruments =        3
                                      Number of folds in cross-fit =         10
                                      Number of resamples          =          1
                                      Wald chi2(1)                 =      10.84
                                      Prob > chi2                  =     0.0010

------------------------------------------------------------------------------
             |               Robust
       lwage |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   .0727853   .0221045     3.29   0.001     .0294612    .1161094
------------------------------------------------------------------------------
Endogenous:   educ
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.
```

Parting Remarks

- Explored lasso for prediction in detail
- Looked at the challenges of estimation after model selection
- Explored some of the solutions