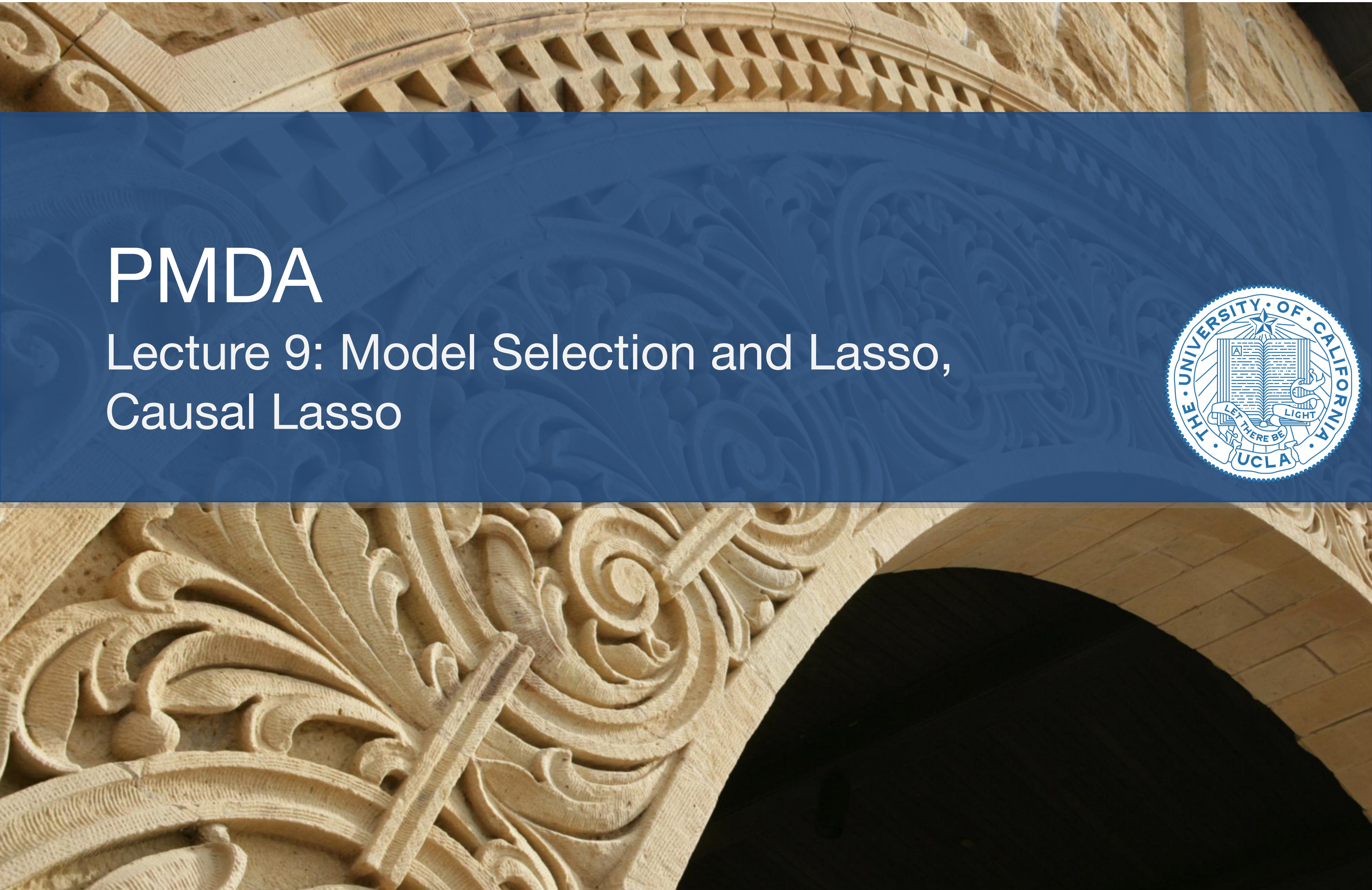
UCLA Anderson
SCHOOL of MANAGEMENT



PMDA

Lecture 9: Model Selection and Lasso, Causal Lasso

Housekeeping

- Overview / where we are
  - Weeks 2-8: causal inference (A/B testing, controls, panel data, diffs-in-diffs, IV, RDD)
- Last two weeks:
  - Today: Model selection and Lasso, Causal Lasso
  - Class 10:
    - Regression Trees, Random Forest, Causal Forest (other methods if time permits)
    - Recap of course: Overview of methods & how to relate to specific business problems
  - Exam preparation : email me if you want specific topics covered
- **Important:**
  - **Assignment #4** if available today and **due** on **Mar 16**
  - Final Exam: Mar 21st at 8:30am

Causal inference vs prediction

- So far: Causal inference
    - We were trying to estimate the causal effect of one specific variable (e.g. treatment, ad, price)
    - We considered a multivariate regression but didn't care about the coefficients on the other variables or about model fit
- This and part of next class: Predictive methods and their relationship with causal inference
    - Different question: which variables have most predictive power?
    - i.e. we now care about coefficients of all variables and about model fit
    - Examples:
        - Which demographics predict credit default?
        - Which ad characteristics drive CTR?
        - Which customer demographics predict responsiveness to ad (required for targeting strategy)?

Variable selection for causal inference vs. prediction

Q: How did we decide which variables to include in the regression when we care about causality?

A: Two reasons:

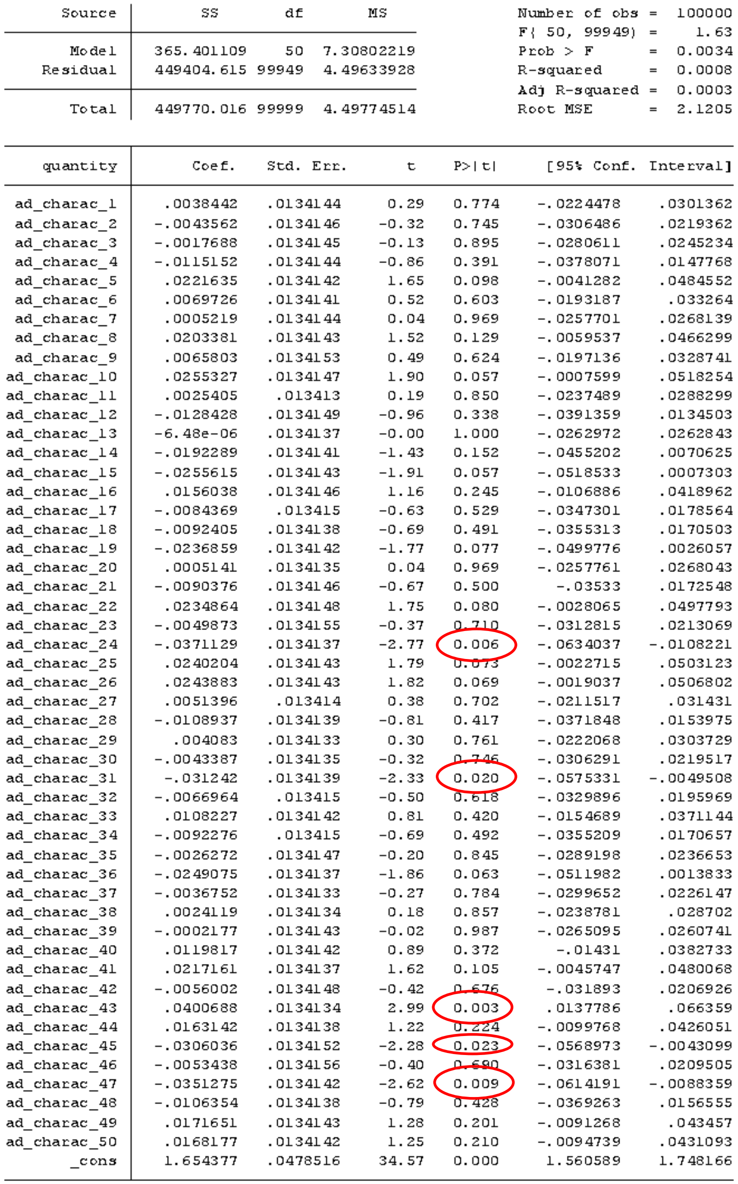
- Main goal was to avoid omitted variable bias, i.e. variables that correlate with both X and Y should be included as controls

- If X was random, we included controls to improve precision

Q: How do we decide which variables to include for prediction?
A:

- All variables we have access to?
- Put all variables, then drop insignificant ones ?
- … neither of those is a good idea !!!

Example: Predicting sales from online ads



```
      Source |       SS       df       MS              Number of obs =  100000
-------------+------------------------------           F( 50, 99949) =    1.63
       Model |  365.401109    50  7.30802219           Prob > F      =  0.0034
    Residual |  449404.615 99949  4.49633928           R-squared     =  0.0008
-------------+------------------------------           Adj R-squared =  0.0003
       Total |  449770.016 99999  4.49774514           Root MSE      =  2.1205

------------------------------------------------------------------------------
    quantity |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 ad_charac_1 |   .0038442   .0134144     0.29   0.774    -.0224478    .0301362
 ad_charac_2 |  -.0043562   .0134146    -0.32   0.745    -.0306486    .0219362
 ad_charac_3 |  -.0017688   .0134145    -0.13   0.895    -.0280611    .0245234
 ad_charac_4 |  -.0115152   .0134144    -0.86   0.391    -.0378071    .0147768
 ad_charac_5 |   .0221635   .0134142     1.65   0.098    -.0041282    .0484552
 ad_charac_6 |   .0069726   .0134141     0.52   0.603    -.0193187     .033264
 ad_charac_7 |   .0005219   .0134144     0.04   0.969    -.0257701    .0268139
 ad_charac_8 |   .0203381   .0134143     1.52   0.129    -.0059537    .0466299
 ad_charac_9 |   .0065803   .0134153     0.49   0.624    -.0197136    .0328741
ad_charac_10 |   .0255327   .0134147     1.90   0.057    -.0007599    .0518254
ad_charac_11 |   .0025405    .013413     0.19   0.850    -.0237489    .0288299
ad_charac_12 |  -.0128428   .0134149    -0.96   0.338    -.0391359    .0134503
ad_charac_13 |  -6.48e-06   .0134137    -0.00   1.000    -.0262972    .0262843
ad_charac_14 |  -.0192289   .0134141    -1.43   0.152    -.0455202    .0070625
ad_charac_15 |  -.0255615   .0134143    -1.91   0.057    -.0518533    .0007303
ad_charac_16 |   .0156038   .0134146     1.16   0.245    -.0106886    .0418962
ad_charac_17 |  -.0084369    .013415    -0.63   0.529    -.0347301    .0178564
ad_charac_18 |  -.0092405   .0134138    -0.69   0.491    -.0355313    .0170503
ad_charac_19 |  -.0236859   .0134142    -1.77   0.077    -.0499776    .0026057
ad_charac_20 |   .0005141   .0134135     0.04   0.969    -.0257761    .0268043
ad_charac_21 |  -.0090376   .0134146    -0.67   0.500      -.03533    .0172548
ad_charac_22 |   .0234864   .0134148     1.75   0.080    -.0028065    .0497793
ad_charac_23 |  -.0049873   .0134155    -0.37   0.710    -.0312815    .0213069
ad_charac_24 |  -.0371129   .0134137    -2.77   0.006    -.0634037   -.0108221
ad_charac_25 |   .0240204   .0134143     1.79   0.073    -.0022715    .0503123
ad_charac_26 |   .0243883   .0134143     1.82   0.069    -.0019037    .0506802
ad_charac_27 |   .0051396    .013414     0.38   0.702    -.0211517     .031431
ad_charac_28 |  -.0108937   .0134139    -0.81   0.417    -.0371848    .0153975
ad_charac_29 |    .004083   .0134133     0.30   0.761    -.0222068    .0303729
ad_charac_30 |  -.0043387   .0134135    -0.32   0.746    -.0306291    .0219517
ad_charac_31 |   -.031242   .0134139    -2.33   0.020    -.0575331   -.0049508
ad_charac_32 |  -.0066964    .013415    -0.50   0.618    -.0329896    .0195969
ad_charac_33 |   .0108227   .0134142     0.81   0.420    -.0154689    .0371144
ad_charac_34 |  -.0092276    .013415    -0.69   0.492    -.0355209    .0170657
ad_charac_35 |  -.0026272   .0134147    -0.20   0.845    -.0289198    .0236653
ad_charac_36 |  -.0249075   .0134137    -1.86   0.063    -.0511982    .0013833
ad_charac_37 |  -.0036752   .0134133    -0.27   0.784    -.0299652    .0226147
ad_charac_38 |   .0024119   .0134134     0.18   0.857    -.0238781     .028702
ad_charac_39 |  -.0002177   .0134143    -0.02   0.987    -.0265095    .0260741
ad_charac_40 |   .0119817   .0134142     0.89   0.372      -.01431    .0382733
ad_charac_41 |   .0217161   .0134137     1.62   0.105    -.0045747    .0480068
ad_charac_42 |  -.0056002   .0134148    -0.42   0.676     -.031893    .0206926
ad_charac_43 |   .0400688   .0134134     2.99   0.003     .0137786     .066359
ad_charac_44 |   .0163142   .0134138     1.22   0.224    -.0099768    .0426051
ad_charac_45 |  -.0306036   .0134152    -2.28   0.023    -.0568973   -.0043099
ad_charac_46 |  -.0053438   .0134156    -0.40   0.690    -.0316381    .0209505
ad_charac_47 |  -.0351275   .0134142    -2.62   0.009    -.0614191   -.0088359
ad_charac_48 |  -.0106354   .0134138    -0.79   0.428    -.0369263    .0156555
ad_charac_49 |   .0171651   .0134143     1.28   0.201    -.0091268     .043457
ad_charac_50 |   .0168177   .0134142     1.25   0.210    -.0094739    .0431093
       _cons |   1.654377   .0478516    34.57   0.000     1.560589    1.748166
------------------------------------------------------------------------------
```

- Suppose you are told that this output shows the effect of ad characteristics (has picture or not, color is red, color is blue …)

Q: Based on the regression, which characteristics should you use for future ad designs?

A: All the variables were actually created by a random number generator…

- All the significant results in this regression occurred by pure chance!

Problem with hypothesis testing: false discoveries

Q: Say a true regression coefficient = 0. What is the probability of incorrectly concluding that it is significant (false discovery)?

A: Pr(rejecting true null), which is also known as Type I error. This is our threshold for the p-value, commonly 0.05

Q: Probability of no false discovery for one zero coefficient?

A: Pr(no false discovery)= Pr(test not rejecting) = 0.95

Q: Probability of no false discovery when we have two zero coefficients (assuming t-tests are independent)?

A: Pr(no false discovery)=Pr(test 1 not rejecting) * Pr(test 2 not rejecting) = 0.95^2

Q: Probability that we have *at least one* false discovery when we have 20 zero coefficients?

A:

- 1 – Pr(no false discovery) = 1 - 0.95^20 = 0.64
- With 50 variables the probability is 92 percent !!

Problem with R-squared: overfitting

- Our goal now:
  - Find the set of X variables that are best predictors
  - How to measure fit for prediction?
- **Q:** Should we use the R-squared of the regression?

$$R^2 = 1 - \frac{Var(e)}{Var(Y)}$$

- **A:**
  - Simple R-squared does not help with the issue of false discoveries
  - R-squared always increases when adding variables
  - R-squared would suggest that you should include as many X-variables as possible in the regression

→ **OVERFITTING**

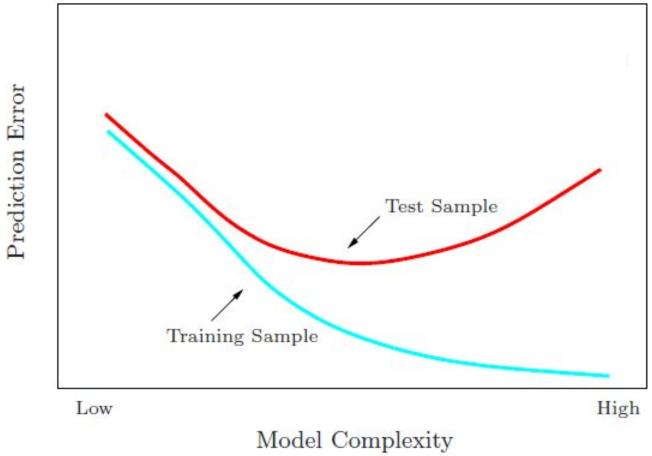
Solution: out-of-sample fit

- Better measure of fit for prediction: Out-of-sample fit/cross validation
  - We use data in one sample to make predictions for a new sample (e.g. future customers)
  - If we have false discoveries (statistically significant by chance) in the first sample, they will not predict well the new sample

- Implementation
  - Split the data into a training and a test sample
  - Estimate the regression on the training sample
  - Use estimated regression to make predictions for the test sample
  - Measure the R-squared in the test sample → large only if the predictions are accurate (i.e., the impact of X is "real", not a statistical fluke)

Model complexity vs. error

- Var(e) in the training sample and the test samples as a function of the complexity of a model



- Out-of-sample error variance first goes down, then up as complexity increases (not the case for in-sample variance, which always goes down)

→Suggests there is an optimal degree of complexity

Summary of model selection for prediction

Problem is overfitting:

- When there are many X’s to choose from, some of them will have large coefficients just by chance
- When we take the model to a new data sample, “wrong” coefficients add noise and the model makes bad predictions

How do we avoid this problem?

- Use **cross validation**: we validate the model on a sample of data that was not used for estimation (a hold-out or test sample)
- The model that provides the best fit in the test sample is the best predictive model
- The best predictive model is not always the most complex model

Agenda

- Variable Selection: Overfitting and Cross-validation
- Regularization & Lasso regression: method that implements model selection automatically
- Causal Lasso
- Workshop

Regularization

- Remember the least-squares estimator of a multivariate regression

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \cdots + \beta_k X_{ki} + \varepsilon_i$$

$$\min_{\beta_0,\beta_1,\ldots,\beta_k} \sum_i (Y_i - (\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + +\beta_k X_{ki}))^2$$

- We choose coefficients that minimize (in-sample) variance of prediction error

- Regularization adds a penalty term on the magnitude of coefficients:

$$\min_{\beta_0,\beta_1,\ldots,\beta_k} \sum_i \left(\varepsilon_i^2\right) + \sum_k c(\beta_k)$$

Regularization: Lasso penalty

- One popular example is the Lasso:

You have to choose this

$$\min_{\beta_0,\beta_1,\ldots,\beta_k} \sum_i \varepsilon_i^2 + \boxed{\lambda} \times \boxed{\sum_k |\beta_k|}$$

This is called L1 norm

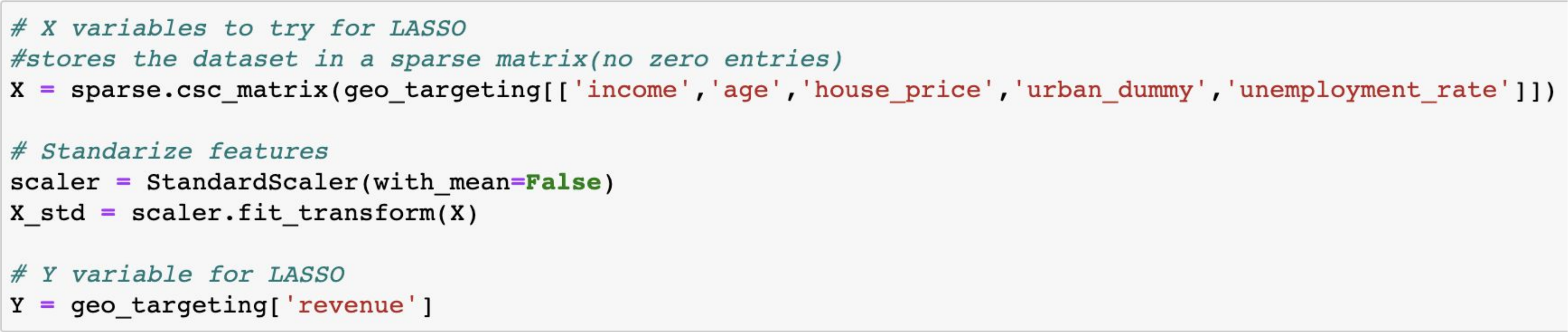
- Properties:
  - More coefficients give lower variance of errors, but higher penalty
  - As λ → 0 you include all variables (= the least squares estimator)
  - As λ → ∞ you include no variables

Data for today’s lecture

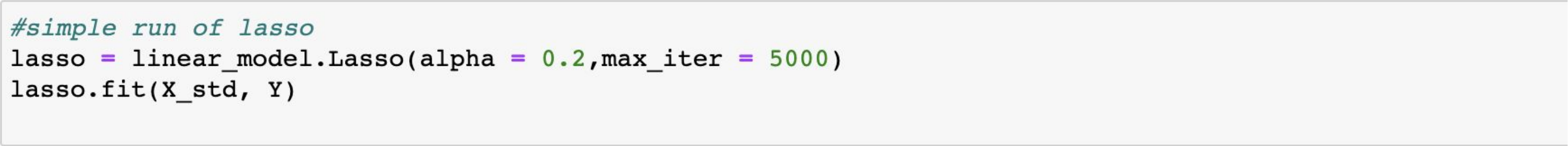
- We will use a simpler data-set called “geo_targeting” for now
- It contains data at the census-tract level (small geographic area)
  - Outcome variable: revenue retailer makes in this census tract
  - Demographic variables for the census tract: (average) income, house prices, age; urban dummy, unemployment rate
- We want to select the best predictors for revenue

Lasso on geo-targeting data

- Need to import “linear_model” from sklearn
- Assign X matrix of variables to select from (need to standardize, but we will talk about this later)
- Assign outcome variable Y
- Do a simple run of Lasso



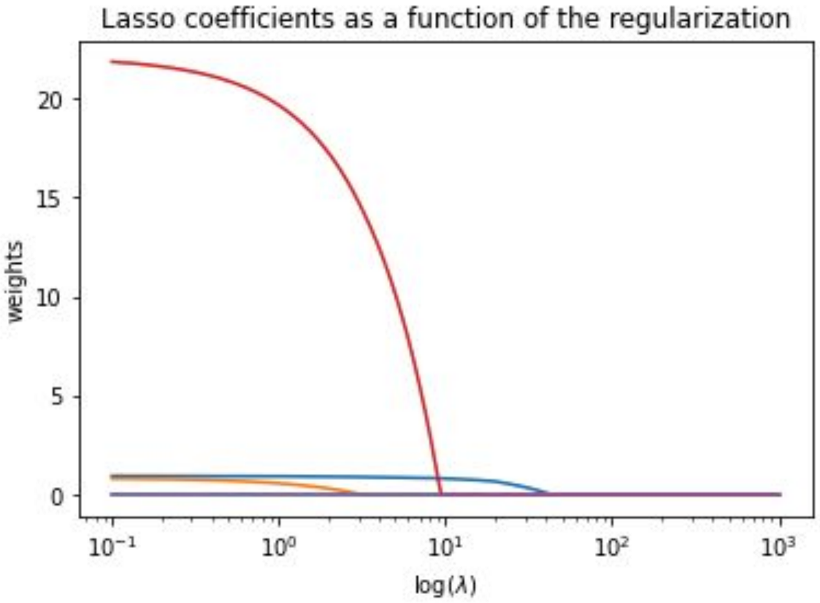
```
# X variables to try for LASSO
#stores the dataset in a sparse matrix(no zero entries)
X = sparse.csc_matrix(geo_targeting[['income','age','house_price','urban_dummy','unemployment_rate']])

# Standarize features
scaler = StandardScaler(with_mean=False)
X_std = scaler.fit_transform(X)

# Y variable for LASSO
Y = geo_targeting['revenue']
```



```
#simple run of lasso
lasso = linear_model.Lasso(alpha = 0.2,max_iter = 5000)
lasso.fit(X_std, Y)
```

First step: plot coefficients for decreasing penalty λ

- Run of Lasso for different values of λ and plot each coefficient



- In the right corner, lambda is so high that all coefficients are zero
- Moving left, lambda is gradually lowered and you retain more variables
- Each line is a coefficient. Graph shows how many/which variables are retained first (they are somewhat "important" as you keep them even with high penalty)

Second step: choose optimal λ

- Next: choose λ that gives the best out-of-sample fit

- LassoCV command already has k-fold cross-validation built-in (more robust than one-fold cv). For each λ:
  - Split sample into k partitions (folds)
  - Estimate model on data except 1st fold, evaluate fit on 1st fold
  - Do the same for all other folds
  - Compute average fit on the k out-of-sample folds
  - This gives you the performance of the model parametrized by λ

Choose optimal λ

- Cross-validated Lasso:
  - For each lambda value, we run k-fold cross-validation
  - We get k out-of-sample MSE measures & compute the average
  - Then pick lambda with lowest average out-of-sample MSE

- **Q:** What is the set of coefficients associated with the model at the optimal lambda value?
  - Problem: we estimated k models at each lambda value, each with different coefficient values

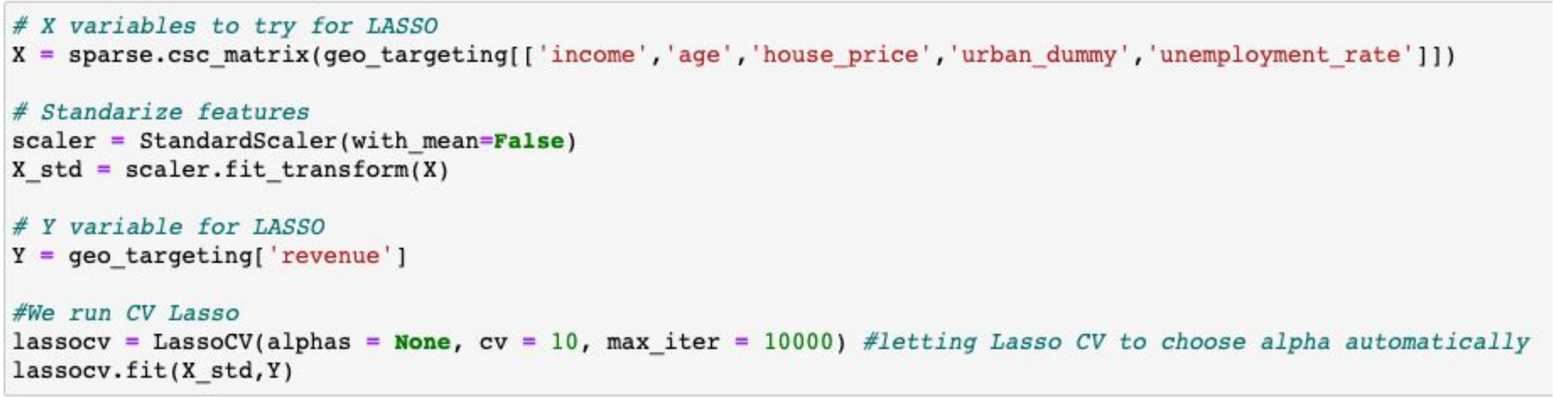
- **A:** Coefficients are derived by estimating the model on the full sample using the optimal lambda

1se rule

- While we typically would use the lambda that minimizes the average out of sample performance (min rule), we sometimes use the 1se rule
- The min rule is best if the focus is on out of sample predictive performance
- The 1se rule is more conservative: it chooses a less complex model. While reasonable, it is still very ad-hoc -> use it if you put a lot of weight on interpreting the coefficients
- 1se rule:
  - Is the biggest lambda with average out of sample MSE that is no more than 1se away from the MSE for the optimal lambda

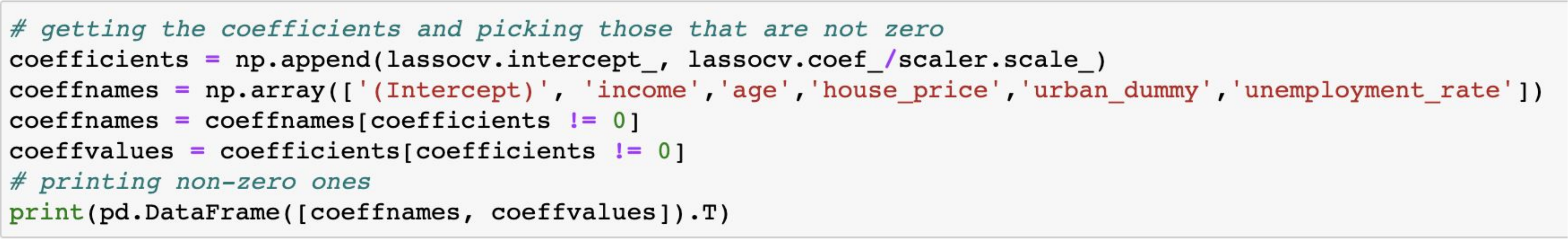
Optimal λ for geo-targeting data

- Code: use LassoCV; “cv” stands for cross-validation



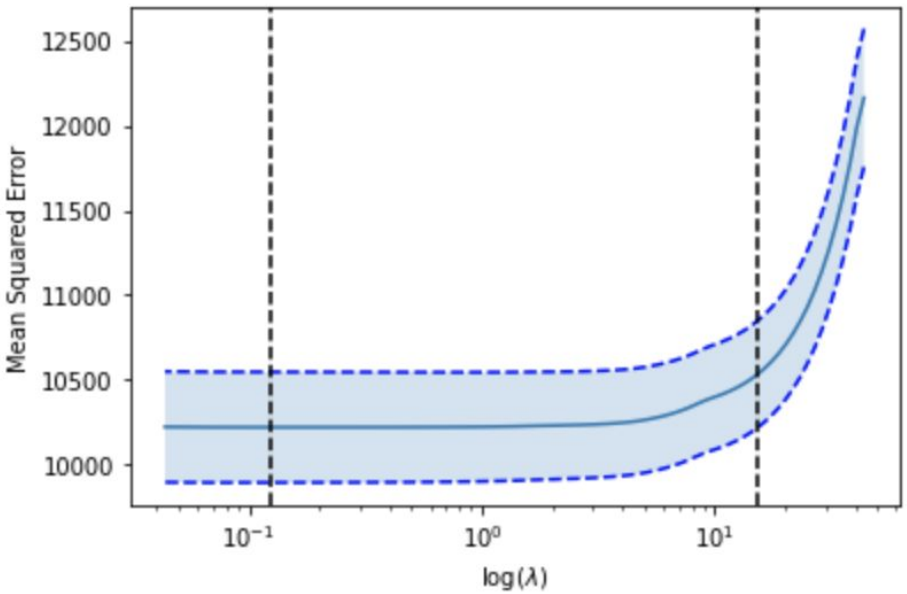
```
# X variables to try for LASSO
X = sparse.csc_matrix(geo_targeting[['income','age','house_price','urban_dummy','unemployment_rate']])

# Standarize features
scaler = StandardScaler(with_mean=False)
X_std = scaler.fit_transform(X)

# Y variable for LASSO
Y = geo_targeting['revenue']

#We run CV Lasso
lassocv = LassoCV(alphas = None, cv = 10, max_iter = 10000) #letting Lasso CV to choose alpha automatically
lassocv.fit(X_std,Y)
```

- You then retrieve and print coefficients retained in the best model



```
# getting the coefficients and picking those that are not zero
coefficients = np.append(lassocv.intercept_, lassocv.coef_/scaler.scale_)
coeffnames = np.array(['(Intercept)', 'income','age','house_price','urban_dummy','unemployment_rate'])
coeffnames = coeffnames[coefficients != 0]
coeffvalues = coefficients[coefficients != 0]
# printing non-zero ones
print(pd.DataFrame([coeffnames, coeffvalues]).T)
```

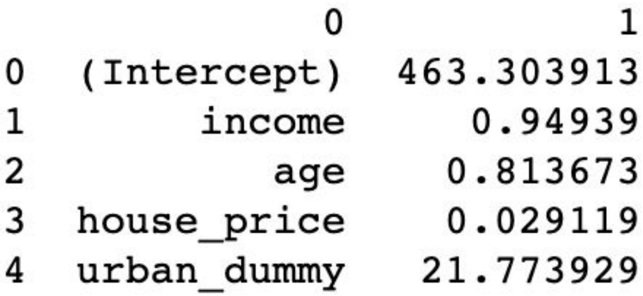
Optimal λ for geo-targeting data



- Start on the right with all coefficients equal to zero (see top of the graph)
- Going left, we lower λ ⇒ include more coefficients. Graph traces out the Mean Squared Error (MSE = Var(e) in the out-of-sample R-squared formula)
- Here: optimum (smallest MSE) is on the left end of the graph.

Results for Lasso with cross-validation

- The best model (= model with best OOS fit) has the following four variables:



```
               0           1
0  (Intercept)  463.303913
1       income     0.94939
2          age    0.813673
3  house_price    0.029119
4  urban_dummy   21.773929
```

Other ways to regularize

- Different methods use different penalty functions

$$\min_{\beta_0,\beta_1,\ldots,\beta_k} \sum_i (\varepsilon_i^2) + \sum_k c(\beta_k)$$

- Lasso: L1 norm: c(β_k) = λ∑_k |β_k |
- Ridge: L2 norm: c(β_k) = λ∑_k (β_k)^2
  - Smaller penalty near zero
- Elastic net: fancy name, but simply a combination of Lasso & Ridge
- Lasso tends to be preferred option in many setting

Technicality: standardizing variables

- Standardizing makes sure all coefficients have same order of magnitude

- Implemented by default in R (via cv.glmnet) but must do manually in python using StandardScaler(with_mean=False)
- The option with_mean=False means that we don't center the data (columns) just scale it. This is needed to keep the sparsity of the matrix

- In R it can be switched off using “standardize=FALSE” option

- When all variables are dummies (as sometimes happens in marketing applications), we don't want to standardize (in all other settings we do)

Agenda

- Variable Selection: Overfitting and Cross-validation
- Regularization & Lasso regression: method that implements model selection automatically
- Causal Lasso: Use Lasso for causal inference (in setting without A/B test)
- Workshop

Starting point: OVB in observational data

- Remember golf ad example: run ads on the golf channel (not randomly assigned)

- Say true relationship is

$$Purchase = \beta_0 + \beta_1 GolfAd + \beta_2 GolfChannel + \beta_3 Income + u$$

- Estimating a univariate regression

$$Purchase = \beta_0 + \beta_1 GolfAd + e$$

yields the following ad effect coefficient:

$$\boxed{E(\hat{\beta}_1) = \beta_1 + \beta_2 \frac{Cov(GolfAd, GolfChannel)}{Var(GolfAd)} + \beta_3 \frac{Cov(GolfAd, Income)}{Var(GolfAd)}}$$

Starting point: omitted variable bias

$$\boxed{E(\hat{\beta}_1) = \beta_1 + \beta_2 \frac{Cov(GolfAd, GolfChannel)}{Var(GolfAd)} + \beta_3 \frac{Cov(GolfAd, Income)}{Var(GolfAd)}}$$

Q: Does omitting either golf channel or income (or both) cause omitted variable bias?

A:

- Depends on the ad targeting!
- Golf channel is used for targeting and hence it causes OVB
- If no targeting based on income, Cov(GolfAd,Income)=0 and hence we do not need to control for income

What should we do?

- In general:
  - Need to control for variables that correlate with the non-randomized treatment variable (GolfAd in this case) and with the outcome (Purchase)
  - So we can keep adding variables
  - **Q:** Any potential issues with adding an exhaustive list covariates?
  - **A:** Yes, we can saturate the model (dimensions close to or more than #observations). This will lead to large std errors. OLS is a good low dimensional method
  - We can use Lasso for this …

Simple Lasso #1

- Simple approach #1:
  - Consider the equation

$$outcome = \beta_0 + \beta_{treat} treatment + \beta_1 X_1 + \beta_2 X_2 + \cdots + e$$

  - Use Lasso to select from control variables X_1,X_2,…
  - Make sure to always retain treatment (can be excluded from Lasso penalty)
  - Run OLS of the outcome on treatment and the selected control variables

Problem with using Lasso to select controls

- **Q:** Can you think of a possible problem with letting Lasso select control variables in approach #1?

- **A:**
  - Lasso's selection criterion is to choose controls that predict outcome
  - Then any variable that is highly correlated to the treatment will be dropped (we are not penalizing the treatment) because including those variables will not add much more predictive power for the outcome (the treatment is already in the model)
  - Excluding such variables that are highly correlated with treatment can lead to OVB if they are correlated with the outcome

Looking deeper into the problems

- Approach #1 ignores the relationship between the controls and the treatment
- We could then account for this by considering the reduced form

$$treatment = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \cdots + v$$

- Approach #1 is based on a structural relation in which we aim to learn the treatment effect given controls
    - That is not an equation that represents a forecasting rule for the outcome (given the treatment and covariates)
    - We can thus also consider the following predictive equation (obtained after replacing the above reduced form in the structural equation)

$$outcome = \gamma_0 + \gamma_1 X_1 + \gamma_2 X_2 + \cdots + e$$

Simple Lasso #2

- We have the following predictive relationships
- We may estimated them using high-dimensional methods such as Lasso

$$treatment = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \cdots + v$$

$$outcome = \gamma_0 + \gamma_1 X_1 + \gamma_2 X_2 + \cdots + e$$

- Simple approach 2:
    - Use Lasso to select from control variables X_1,X_2,… in only one of the relationships above
    - Run OLS of the outcome on treatment and the selected control variables

Problem with using Lasso to select controls

- **Q:** Can you think of a possible problem with letting Lasso select control variables in approach #2?

- **A:**
  - Lasso's selection criterion is to choose controls that predict outcome
  - Using the second equation: could end up dropping controls that are only mildly correlated with outcome, but still correlated with treatment → still have OVB
  - Using the first equation: could end up dropping controls that are only mildly correlated with treatment, but still correlated with outcome → still have OVB
  - Better approach: causal Lasso

Causal Lasso

- Causal Lasso uses both predictive relationships
    - **First Stage:** Use Lasso to select from control variables X_1,X_2,… that predict the treatment
    - **Second Stage:** Use Lasso to select from control variables X_1,X_2,… that predict the outcome (leave out the treatment)
    - **Third Stage:** Run OLS with the treatment and the union of the controls variables from the first and second stage

Causal Lasso additional comments

- We select variables that are useful for predicting the treatment

- We select variables that are useful for predicting the outcome

- Hence we are using variables that are important for either of the two predictive relationships to guard against omitted-variables bias

- We leave out variables that are only mildly correlated with treatment and outcome. Variables with large effects are included!
- This minimizes the impact of OVB **Q:** why?
  - **A:** the coefficient of a variable that is not included and its correlation with treatment will be small thus reducing the impact of OVB

Causal Lasso Final Thoughts

- This approach is also known as “Double Selection”
- Logic:
    - First stage: select variables that predict treatment
    - Second stage: Select variables that predict the outcome

- **Q:** Does this remove OVB? Is it conservative?
- **A:**
    - Yes, but conservative approach to removing OVB: OVB caused by variables that correlate with both X and Y, but here we control for correlation with either
    - But can improve precision
- **Q:** What would happen if we did the intersection?
- **A:** We would leave out, for example, a variable that is highly correlated with treatment but mildly correlated with the outcome thereby leading to OVB

Agenda

- Variable Selection: Overfitting and Cross-validation
- Regularization & Lasso regression: method that implements model selection automatically
- Causal Lasso: Use Lasso for causal inference (in setting without A/B test)
- Workshop