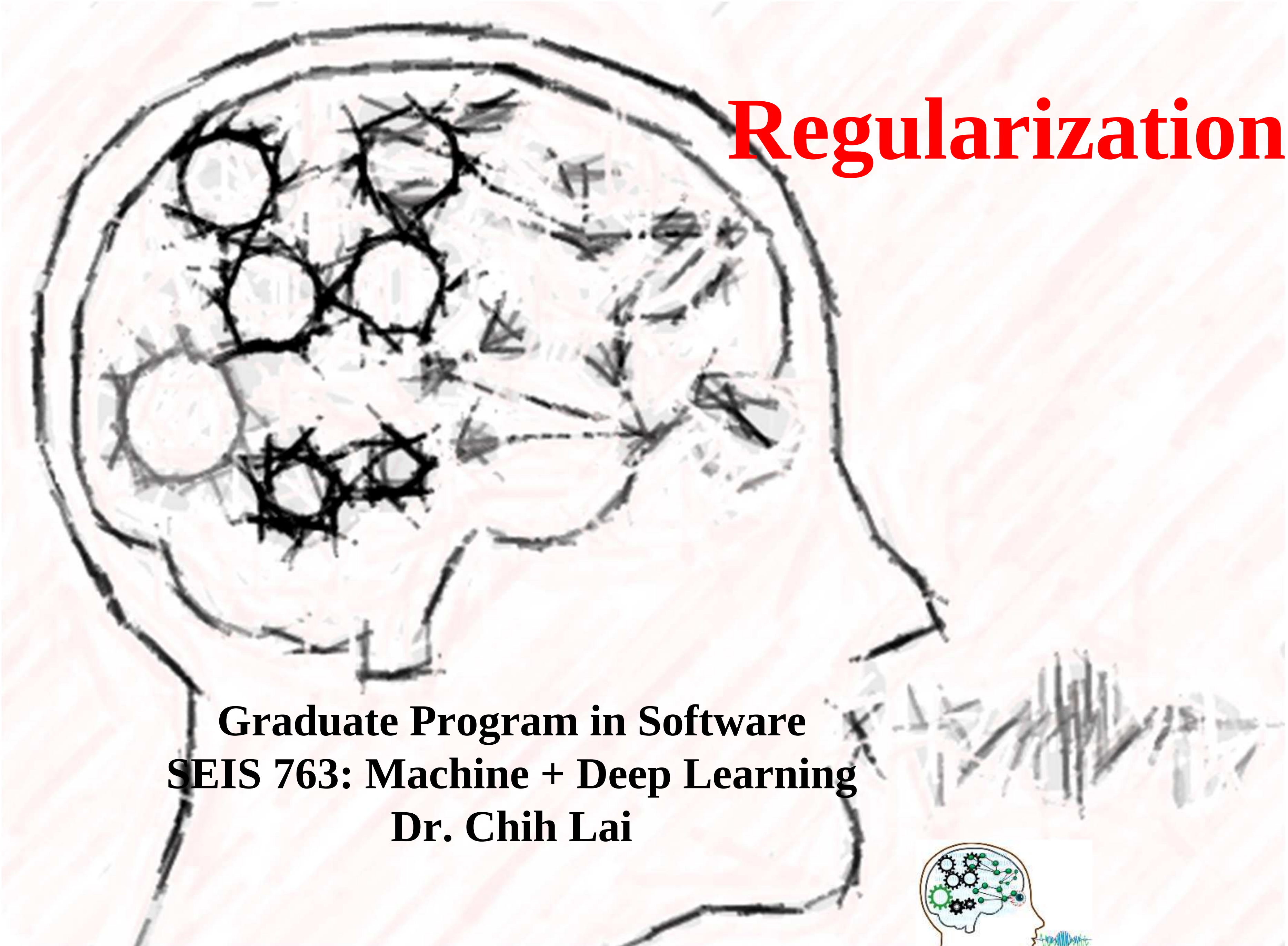
Regularization



Graduate Program in Software
SEIS 763: Machine + Deep Learning
Dr. Chih Lai

References to Matlab Regularization

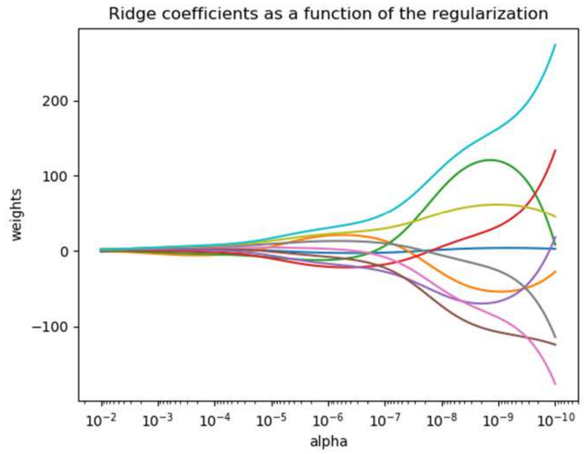
- Matlab Lasso regularization
  - http://www.mathworks.com/help/stats/lasso.html

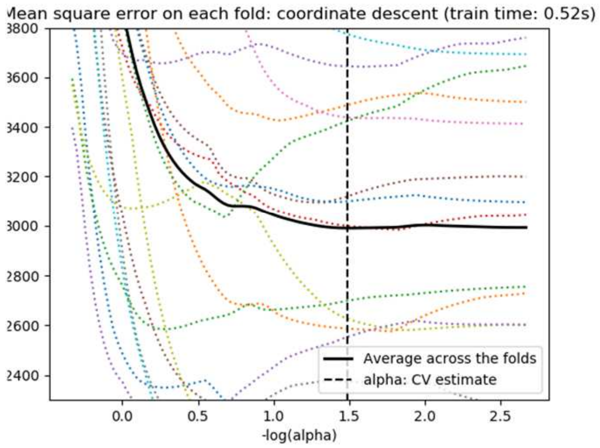
- Matlab Ridge regularization
  - http://www.mathworks.com/help/stats/ridge.html

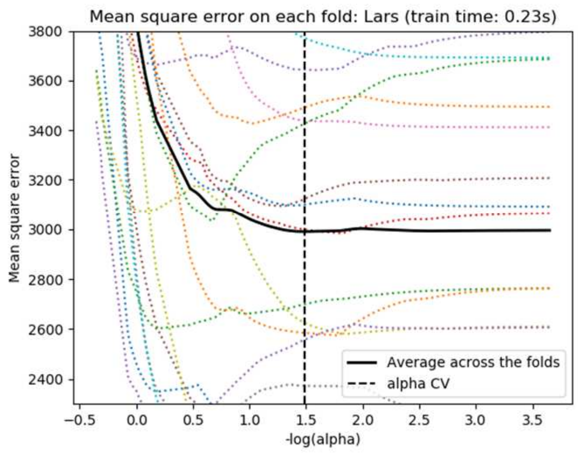
- Matlab Elastic Net regularization
  - http://www.mathworks.com/help/stats/lasso-and-elastic-net.html

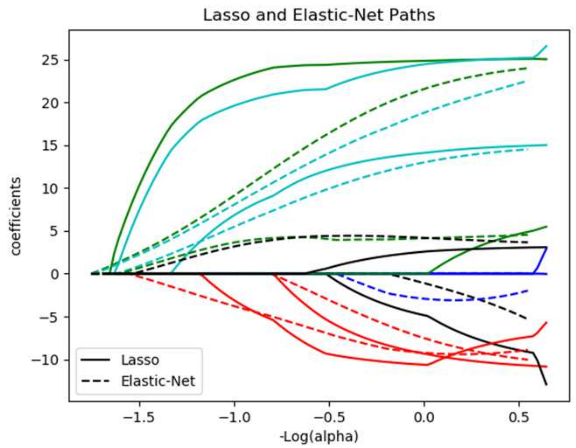
References to Python scikit-learn Regularization

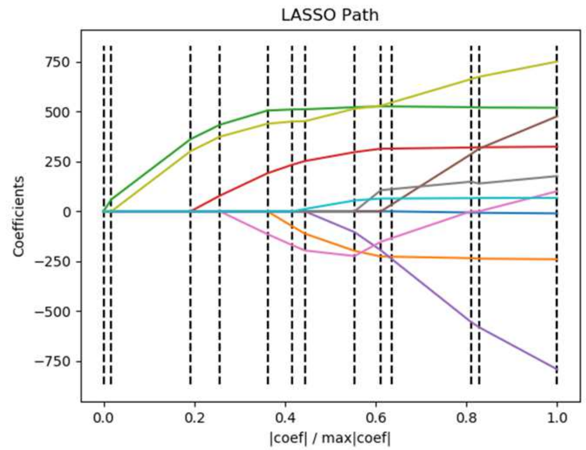
- http://scikit-learn.org/stable/modules/linear_model.html











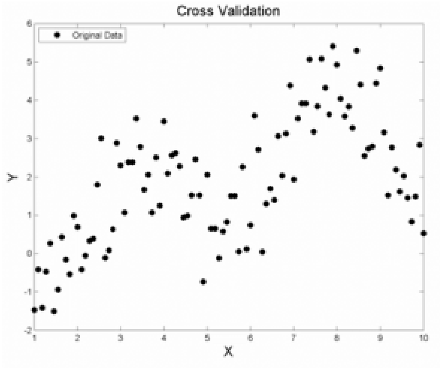
Evaluating Accuracy

- A predictive model tries to explain data as much as possible w/o capturing too much noise (outliers).

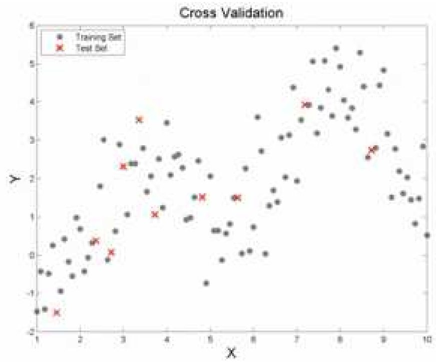
- How do we know our model fits to data or noise?

- Cross validation.
  1. Prepare original data.
  2. Divide data into training & test sets. (or **multiple combinations** of such sets)
  3. Build a model using training data.
  4. Validate the model using test set.

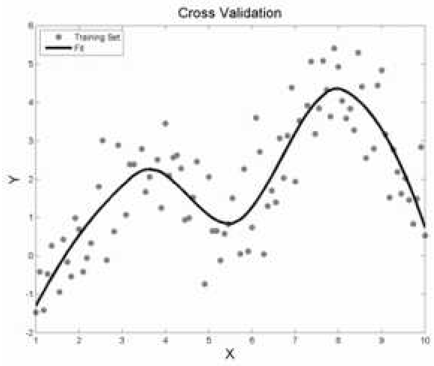
1. Original Data



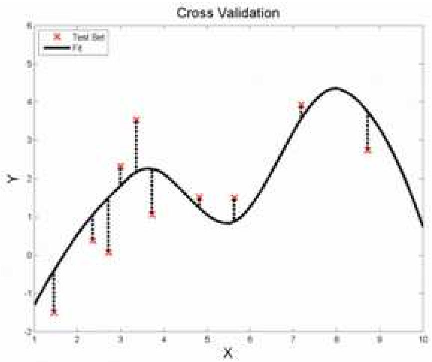
2. Dividing Data



3. Training



4. Testing



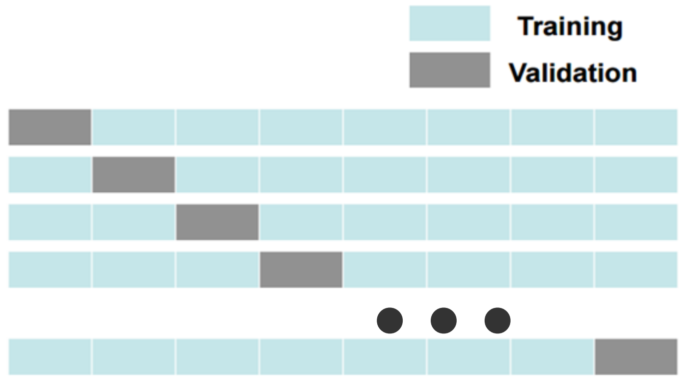
Testing Methods

1. Holdout → 2/3 for training & 1/3 for testing
   - Smaller training set
   - A class may be over-represented in training set
     - Or, it may be under-represented in testing

Outlook	Temp.	Humid	Windy	Play
S	W	H	F	N
S	W	H	T	N
O	W	H	F	Y
R	M	H	F	Y
R	C	L	F	Y
R	C	L	T	N
O	C	L	T	Y
S	M	H	F	N
S	C	L	F	Y
R	M	L	F	Y
S	M	L	T	Y
O	M	H	T	Y
O	W	L	F	Y
R	M	H	T	N

2) Random subsampling
   - Few records may always be in one set

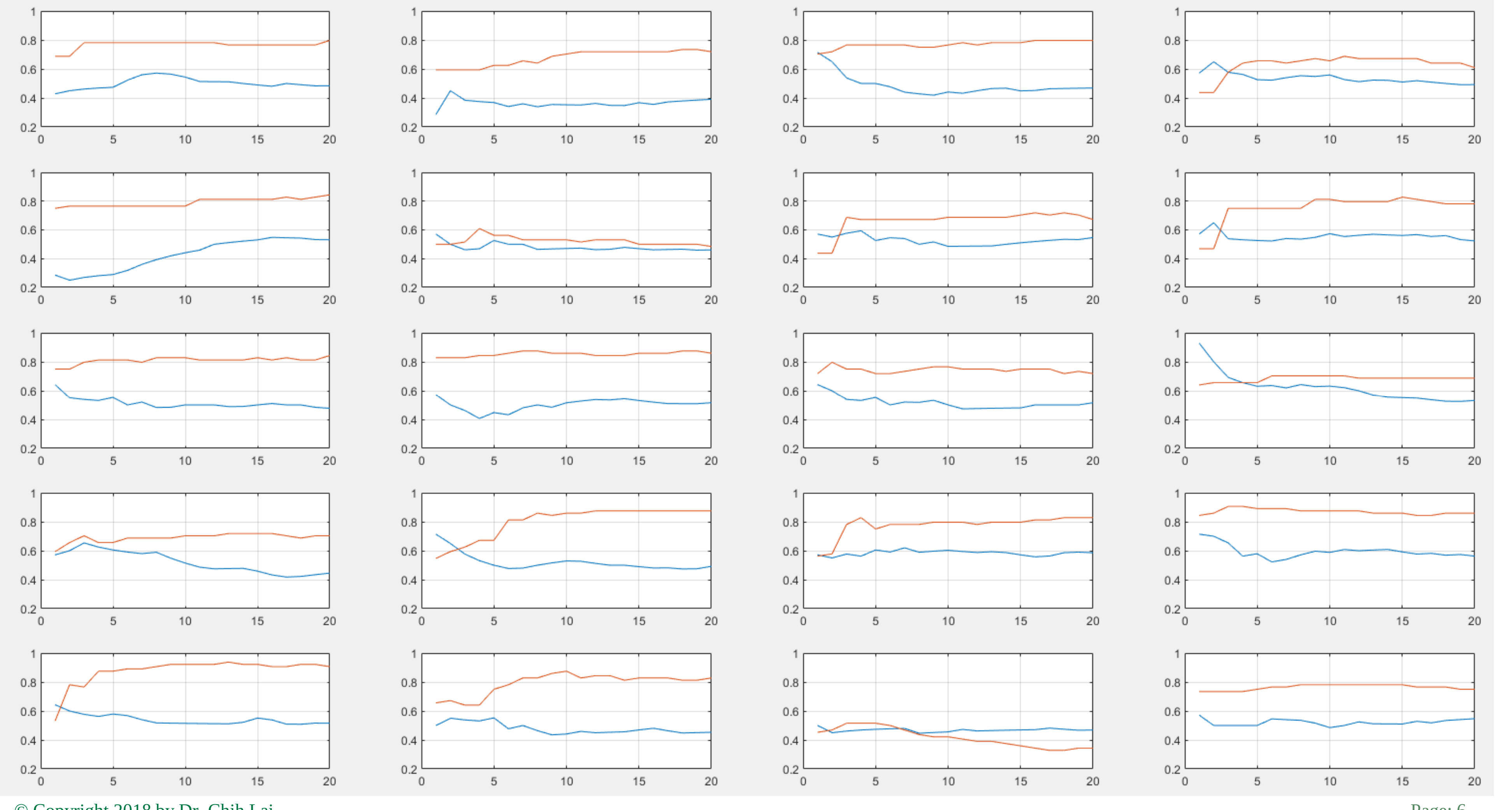
3) Cross validation **(*k*-Fold Cross-Validation)**
   - Partition data into ***k***-fold (disjoint sets).
   - Repeat training on *k*-1 subsets, test on **1** subset.
   - Each record trains *k*-1 times, tests **1** time.
   - Accuracy is then averaged from ***k***-tests.



- After cross-validation, which model has the best performance?
  1. That is not our purpose (or concern) here.
  2. Different models → difference performance on different data.

Really Need Cross-Validation?

- Isn't random selection "random enough" in selecting really random samples?
  - Red line = accuracy of training dat. Blue line = accuracy of testing data
  - So, we randomly partition data 20 times in the following plots. Each partition has 20 / 20 data.



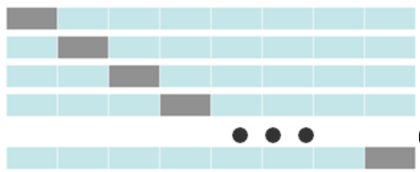
Various Ways To Perform Cross Validation in Matlab

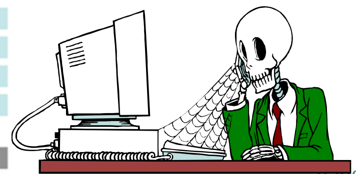
- Matlab functions for general purpose cross validation.
  - crossval()→ Apply cross validation to Matlab functions.
  - **cvpartition**()→ Split data to test sets & training sets.

sklearn.utils.shuffle
sklearn.model_selection.train_test_split
sklearn.model_selection.Kfold
sklearn.model_selection.StratifiedShuffleSplit

- Several Matlab functions have built-in support for cross validation.
  - i.e. cross-validation can be used as input parameters to other Matlab ML functions.
  - Generate a Lasso model using 5-fold cross validation.
    - [B Stats] = lasso(X, Y, **'CV', 5**);

- Matlab cross validation functions are **parallel-computing enabled**.





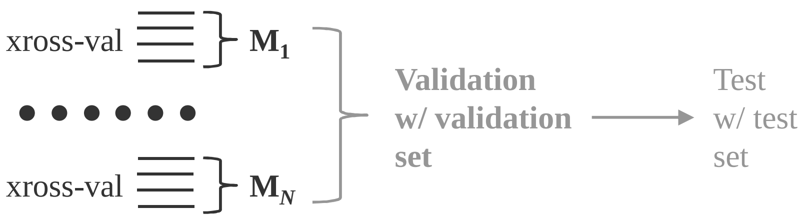
Training, Validation, and Testing Data

F.Y.I.

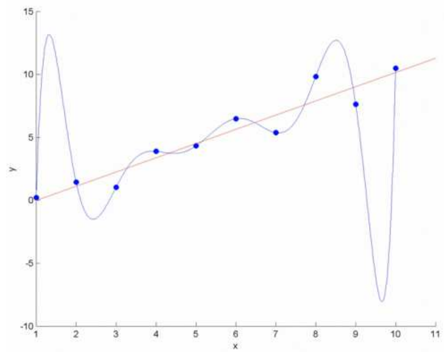
- Training = learning model weights.

- Validation = check / minimize overfitting
  - Training accuracy ↑ Validation accuracy ↑
  - Training accuracy ↑ Validation accuracy ↓ **Overfitting & should stop training**
  - Training accuracy ↓ Validation accuracy ↑
  - Use validation data to pick a model or pick model parameters.

- Test = confirm actual predictive power of your model.
  - Validation = best possible scenario.
  - Testing = most likely scenario.

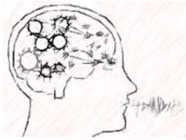


Improving Test Accuracy



- If test accuracy is not good, how do we improve it?
  - Remove outliers… or Avoid overfitting

- Traditional feature Selection.
  - ***Entropy***, ***p-Value***, ***Sequential Covering method*** (matlab **sequentialfs**).
  - Test **ALL** variables **one at a time**, & gradually add/remove to/from final model.
  - Not consider interactive effect. (i.e. XOR problem)
  - ➢ Variables are either included or excluded, **NO** ***partial inclusion***.

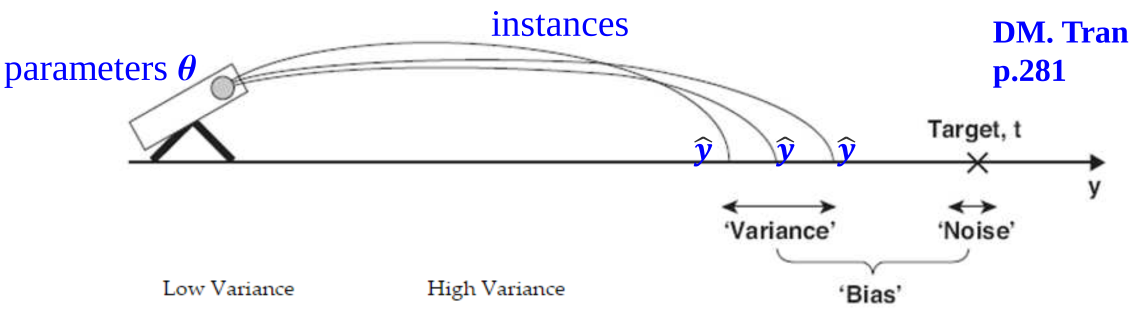


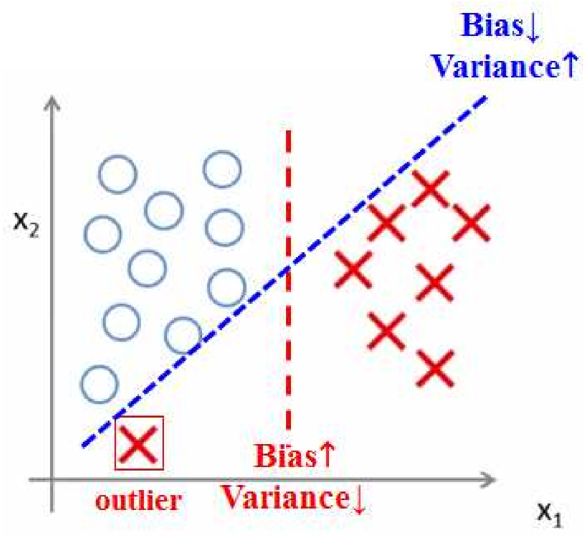
- Regularization is to improve "test" accuracy (or at least not degrade it) by…
  - Retain **only** important predictors to avoid overfitting… ***Bias*** & ***Variance***!!

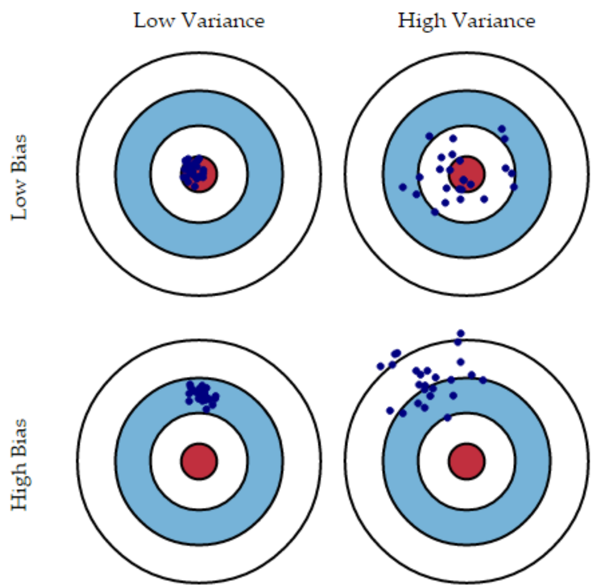
- **Bias & Variance.**

Bias and Variance

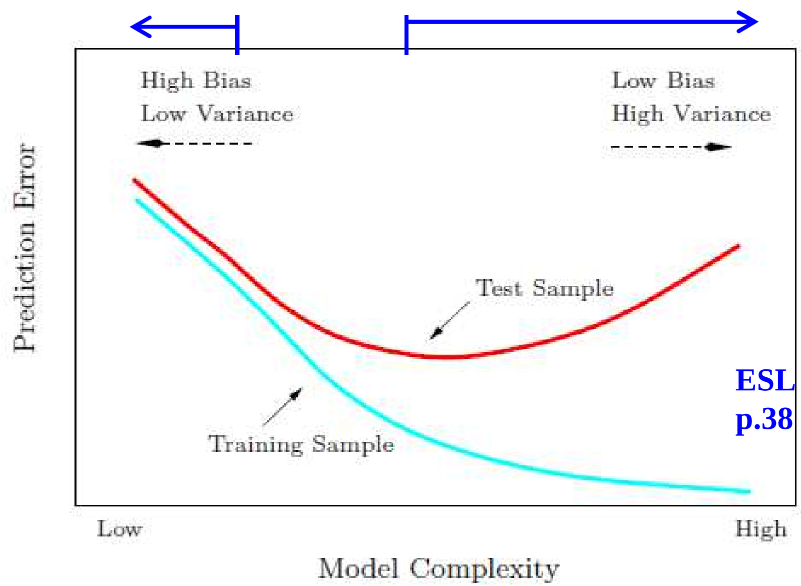
- Bias➔ how far off average prediction (from all $\hat{y}$) is from the correct value y or $\bar{y}$.
- Variance➔ deviation between all predictions $\hat{y}$.



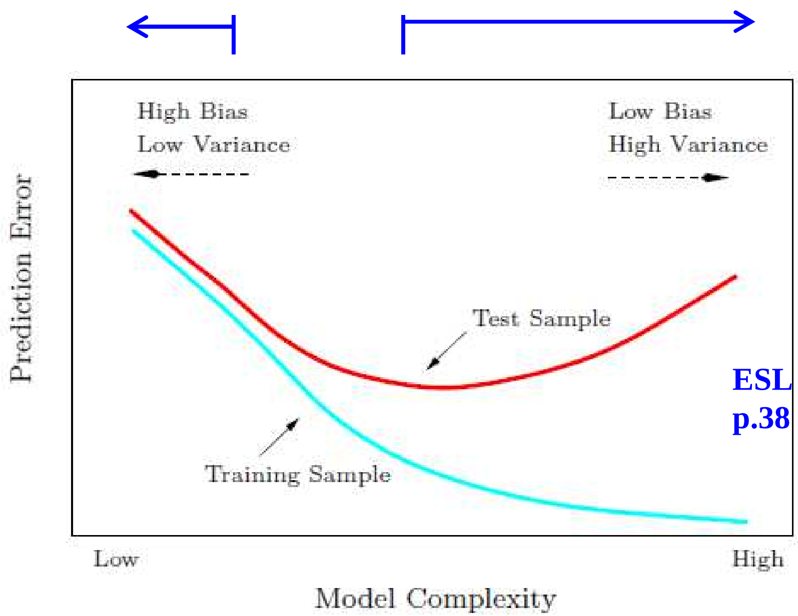


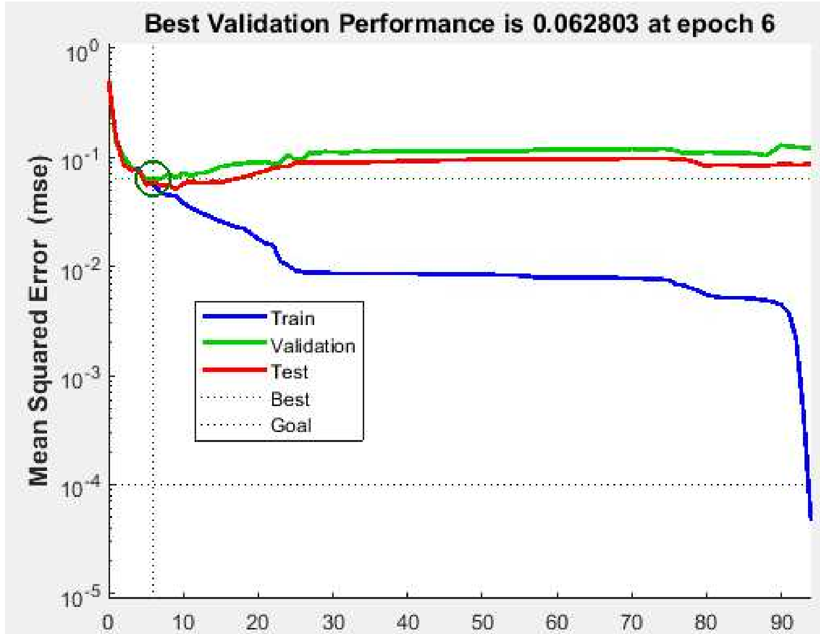


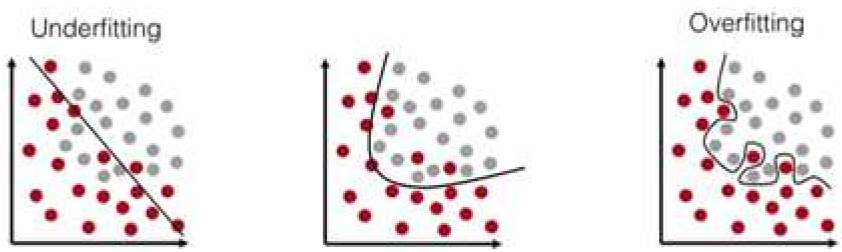
http://scott.fortmann-roe.com/docs/BiasVariance.html

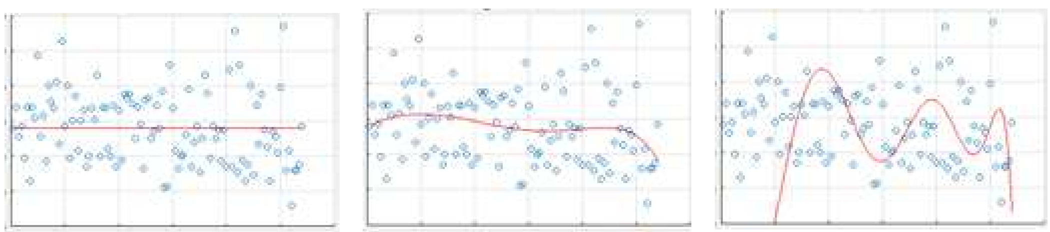


Just About Right !!??





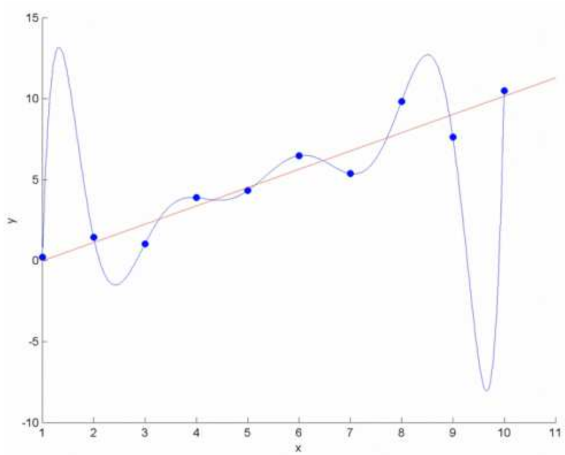




Why Regularization???

- Simplify predictive models by
  - **Decreasing** # of predictors,
  - ➔ **Shrinking** some parameters (coefficients, θ) toward 0, **(but may ≠ 0)**
  - Combine both.

- Why?
  - Design consideration➔ simpler model increases speed & reduces memory.
  - Interpretability.
  - Test accuracy➔ simpler model avoids overfitting & may generalize better **later**.
    - LR may not work well on data w/ many correlated vars (sales & tax).
    - LR model repeatedly use correlated vars to explain data.

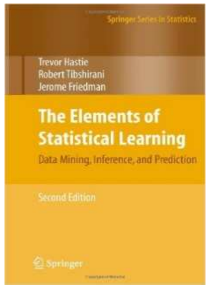


Defining Regularization

- Regularization is to simplify models & prevent overfitting in predictive models.
  - Similar purpose as to feature selection.
  - But, an overall **parametric** (i.e. $\boldsymbol{\theta}$) method.
  - Regularization may perform better than feature selection.
    - Consider multiple features at same time & allow **partial inclusion** of features.

- HOW?? Add penalty into the model to remove/shrink redundant predictors.
  - **Balance** between **MES** & **model complexity**. $J(\boldsymbol{\theta}) \approx \boldsymbol{MSE} + \textbf{ModelComplexity}$
  - More specific, $J(\boldsymbol{\theta}) = \boldsymbol{MSE} + \lambda \times \textbf{ModelComplexity}$
  - By making the model more ***sparse*** (parsimonious) and potentially more accurate (for future).

- Regularization algorithms include:
  - Ridge regression (also known as Tikhonov regularization). **L2**
  - Lasso regression. (***L***east ***A***bsolute ***S***hrinkage and ***S***election ***O***perator). **L1**
  - Elastic net algorithms.



Ridge, Lasso, Elastic Net

- When to use which regularization algorithm? More discussion later.
  - Ridge regression.
    - Shrink θ but **NOT** completely 0, compensate for correlated features w/o eliminating them.
  - Lasso regression. (***L**east **A**bsolute **S**hrinkage and **S**election **O**perator*)
    - Drives some θ **toward 0** relatively quick, great for feature selection w/ ***wide data*** sets.
    - ***Wide*** data sets are data with large # of attributes but small # of records.
  - Elastic net algorithms.
    - Relative weighted average (α) of lasso and ridge.
    - ***L1 ratio.***

Defining **Complexity** in Regularization Formula

- Cost function $\boldsymbol{MSE} = \boldsymbol{J}(\boldsymbol{\theta}) = \frac{1}{2m}(\boldsymbol{Y} - \widehat{\boldsymbol{y}})^2$
  - $\widehat{\boldsymbol{y}} = \boldsymbol{h}(\theta) = \theta^{\mathrm{T}}\boldsymbol{X} = \theta_0 \times \boldsymbol{x}_0 + \theta_1 \times \boldsymbol{x}_1 + \ldots + \theta_n \times \boldsymbol{x}_n$

- **Add regularization**➔ A flexible and tunable way to control overfitting.
  - $J(\boldsymbol{\theta}) = \boldsymbol{MSE} + \lambda \times \textbf{ModelComplexity}$

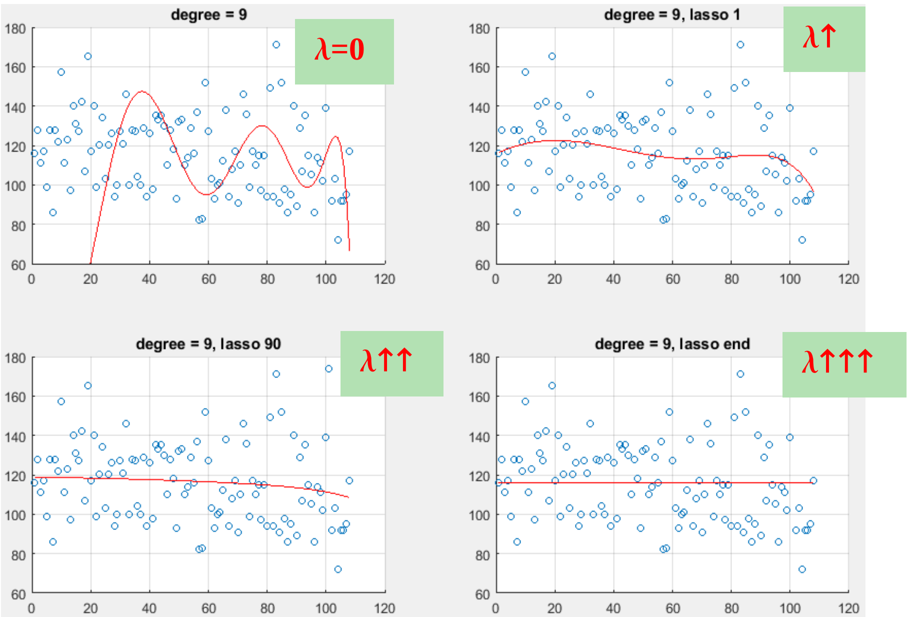
- Choose $\boldsymbol{\theta}$**s** that fit data (i.e. smallest $\boldsymbol{MSE}$) **AND** have smaller ***magnitude*** $\boldsymbol{\theta}$.
  - Magnitude = $\boldsymbol{\Sigma|\theta|^1}$ (Lasso regression)
  - Magnitude = $\boldsymbol{\Sigma|\theta|^2} = \boldsymbol{\Sigma(\theta^{\mathrm{T}}\theta)}$ (Ridge regression)
  - $J(\boldsymbol{\theta}) = \frac{1}{2m}(\boldsymbol{Y} - \widehat{\boldsymbol{y}})^2 + \lambda \boldsymbol{L_p} \approx \frac{1}{2m}(\boldsymbol{Y} - \widehat{\boldsymbol{y}})^2 + \lambda \Sigma \boldsymbol{\theta^P} \approx \frac{1}{2m}(\boldsymbol{Y} - \widehat{\boldsymbol{y}})^2 + \lambda \sum_{j=1}^{n} \boldsymbol{\theta_j^P}$.

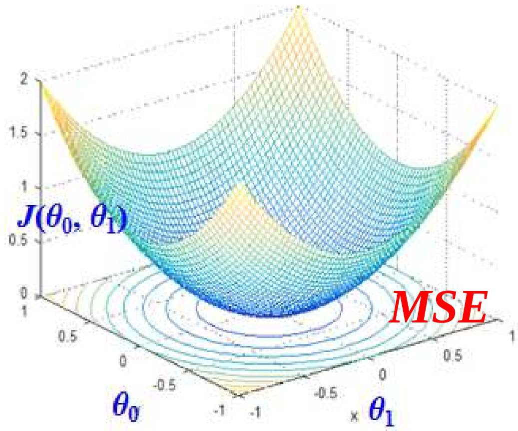
- $J(\boldsymbol{\theta}) \approx \boldsymbol{MSE} + \lambda\boldsymbol{\theta}$
  - λ **is the amount of regularization**.
  - To minimize $J(\boldsymbol{\theta})$, either reduce $\boldsymbol{MSE}$ or model complexity.

$$\boxed{\min_{w} \frac{1}{2n_{samples}}||Xw - y||_2^2 + \alpha||w||_1}$$

Amount of Regularization λ and MSE

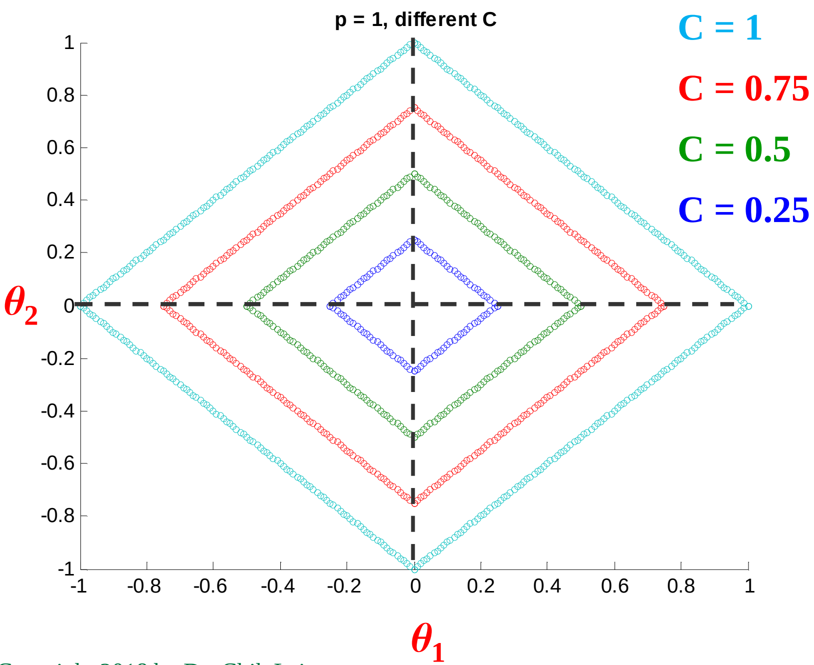
- Choose parameters that fit data (i.e. smallest cost) and have smallest magnitude.
  - $J(\boldsymbol{\theta}) = \frac{1}{2m}(\boldsymbol{Y} - \widehat{\boldsymbol{y}})^2 + \boldsymbol{\lambda L_p} \approx \frac{1}{2m}(\boldsymbol{Y} - \widehat{\boldsymbol{y}})^2 + \boldsymbol{\lambda \Sigma \theta^P}$
  - $J(\boldsymbol{\theta}) \approx \boldsymbol{MSE} + \boldsymbol{\lambda\theta}$
  - **λ is the amount of regularization**.
  - $\lambda\uparrow$ ➔ $\theta\downarrow$ ➔ **MSE**$\uparrow$ less complex model ➔ may **underfit**.
  - $\lambda\downarrow$ ➔ $\theta\uparrow$ ➔ **MSE**$\downarrow$ more complex model ➔ may **overfit**.

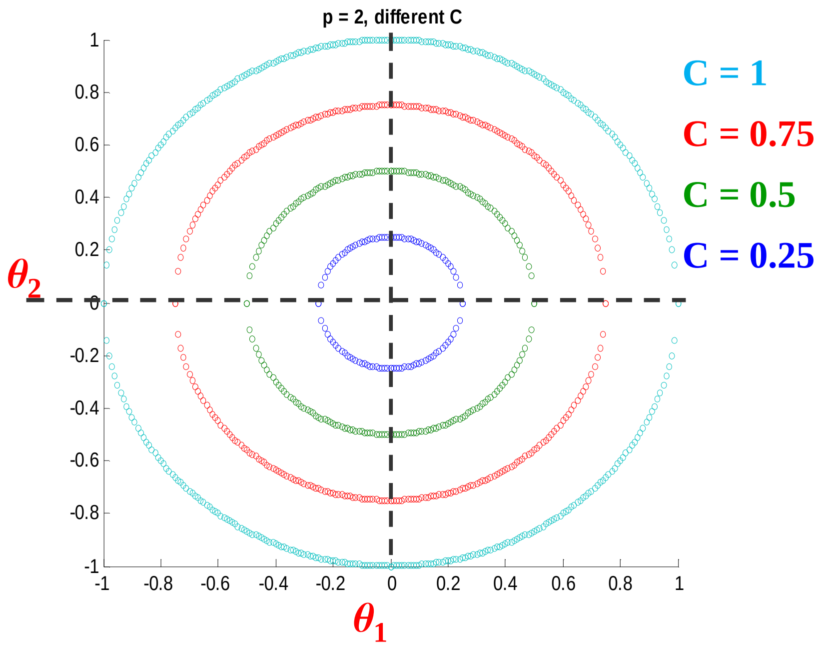




Different $\boldsymbol{\theta}$s ➔ Create Different *MSE* Contours & Isosurfaces

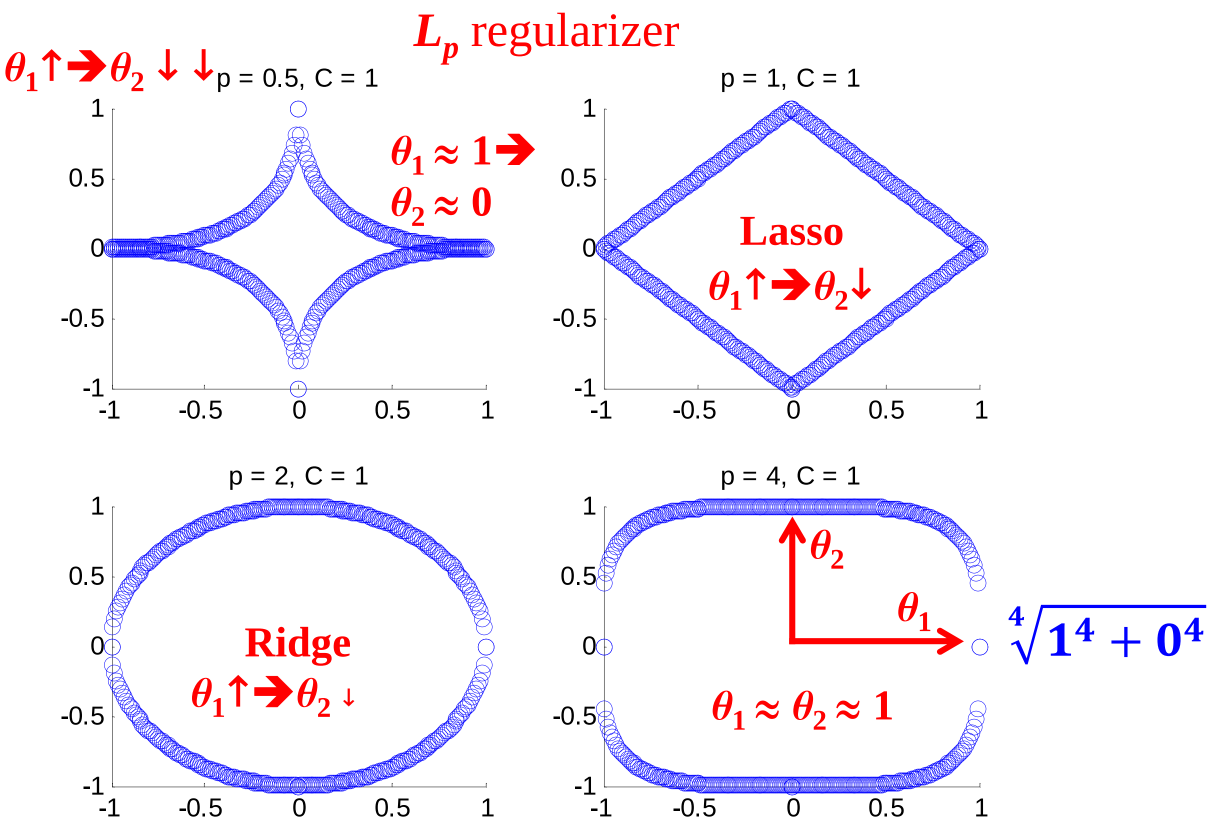
- $J(\boldsymbol{\theta}) = \frac{1}{2m}(\boldsymbol{Y} - \hat{\boldsymbol{y}})^2 + \lambda \boldsymbol{L_p} \approx \frac{1}{2m}(\boldsymbol{Y} - \hat{\boldsymbol{y}})^2 + \lambda\Sigma\theta^P$
- Isosurface created by the $\boldsymbol{L_p}$ ***regularizer*** is $\boldsymbol{C} = (\sum_{j=1}^{n} |\theta_j|^p)^{1/p} = (\sum_{j=1}^{n} ||\theta||_p)^{1/p}$
  - i.e., $\boldsymbol{L_1}$ with θ_1, θ_2, isosurface $\boldsymbol{C} = |\theta_1| + |\theta_2|$
  - i.e., $\boldsymbol{L_2}$ with θ_1, θ_2, isosurface $\boldsymbol{C} = \sqrt[2]{\theta_1^2 + \theta_2^2}$
  - A 2-D **isosurface** is a set of isolines; each contains points of the same complexity ***C***.





Different Isosurface = Different Regularizers

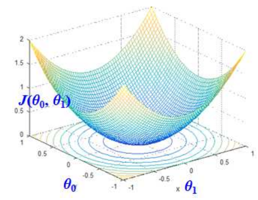
- So the isosurface of $\boldsymbol{L_p}$ $\boldsymbol{C} = (\sum_{j=1}^{n} |\theta_j|^p)^{1/\boldsymbol{p}} = (\sum_{j=1}^{n} ||\boldsymbol{\theta}||_{\boldsymbol{p}})^{1/\boldsymbol{p}}$
  - Example, let $\boldsymbol{p} = 4$ with θ_1, θ_2, so $\boldsymbol{L_4} = \sqrt[4]{\theta_1^4 + \theta_2^4}$



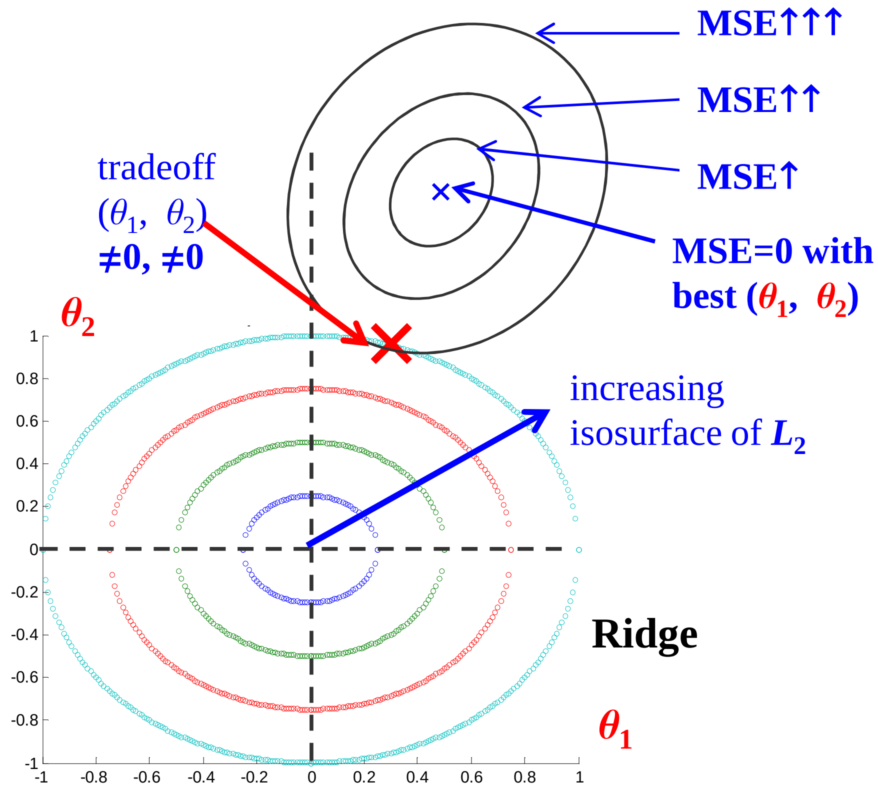

```
figure, i = 0;
for p = [0.5, 1, 2,  4]
   i = i + 1;
   C = 1;  t=[];
   % isosurface in one quad
   for t1 = 0: 0.01: C
      t=[t; [t1 ((C^p)-(t1^p))^(1/p)]; ];
   end
   T = t;
   % isoline in all 4 quads
   T = [T; [t(:, 1) * -1, t(:, 2)]];
   T = [T; [t(:, 1), t(:, 2) * -1]];
   T = [T; [t(:, 1) * -1, t(:, 2) * -1]];
   subplot(2,2,i),
   scatter(T(:,1), T(:,2)),
   title(['p = ' num2str(p) ', C = 1'])
   xlim([-C C]); ylim([-C C])
end
```

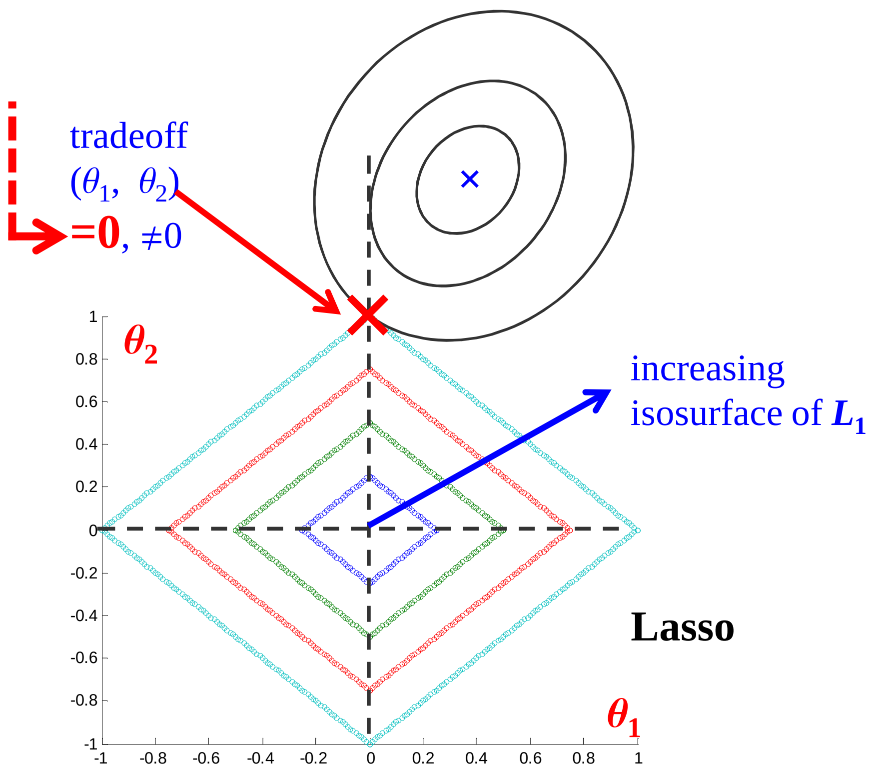
See more Matlab code for plotting *Lp* in Appendix

Interpreting Lasso and Ridge



- $J(\boldsymbol{\theta}) = = \frac{1}{2m}(\boldsymbol{Y} - \hat{\boldsymbol{y}})^2 + \lambda \boldsymbol{L_p} \quad \approx \quad \boldsymbol{MSE} + \boldsymbol{\lambda\theta}$
- Need to balance out between increasing MSE and amount of regularization.
- Lasso
  - More parameters θ will be exactly 0 in L_1 since its isosurface is more protruding.
  - More likely to lead to sparser (simpler) model. Balance between **sparsity** & **convexity**.

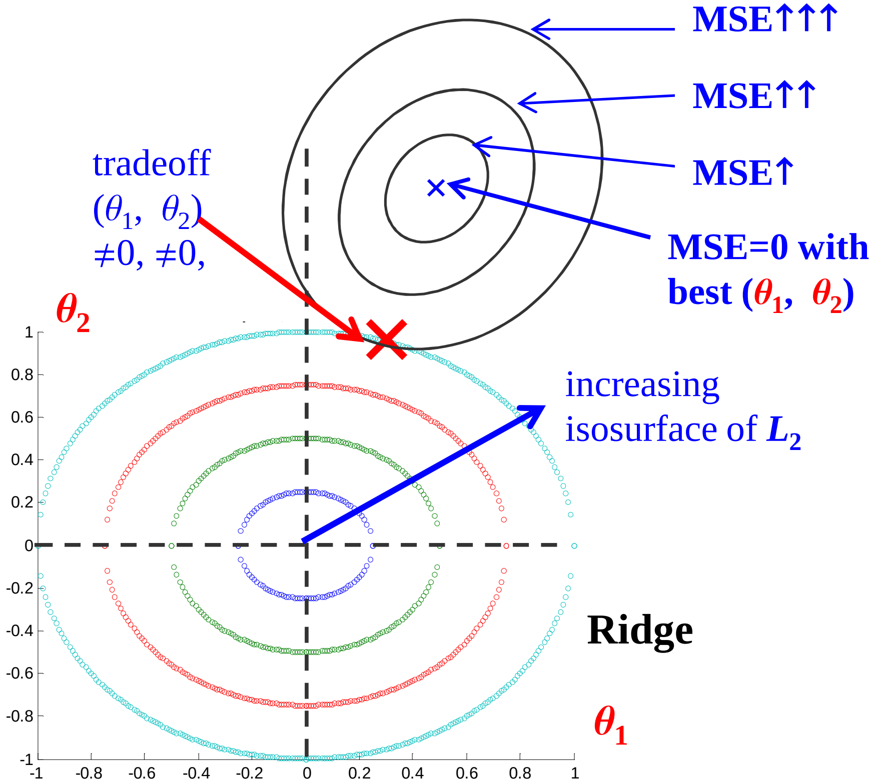


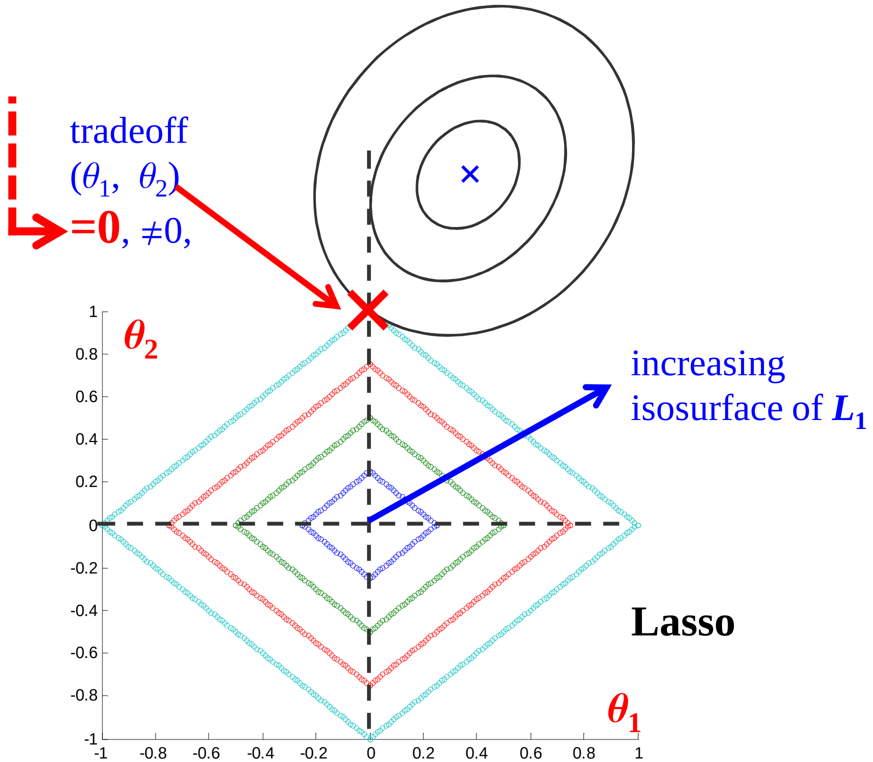


Ridge and Lasso Summary

- Ridge
  - Sparser solutions happen only if smallest **MSE** is **exactly on the θ_1 or θ_2 axis**.
  - Because of the sphere shape of the regularizer.

- Lasso method tends to generate sparser solutions.
  - Sparser solution: have more $\theta = 0$ even when the MMSE is **NOT** on the θ_1 or θ_2 axis.
  - Because of the contour shape of the regularizer, resulting **simpler** predictive models.



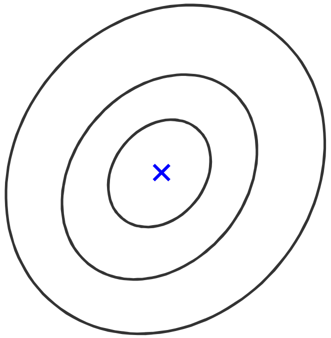


Sparsity and Convexity

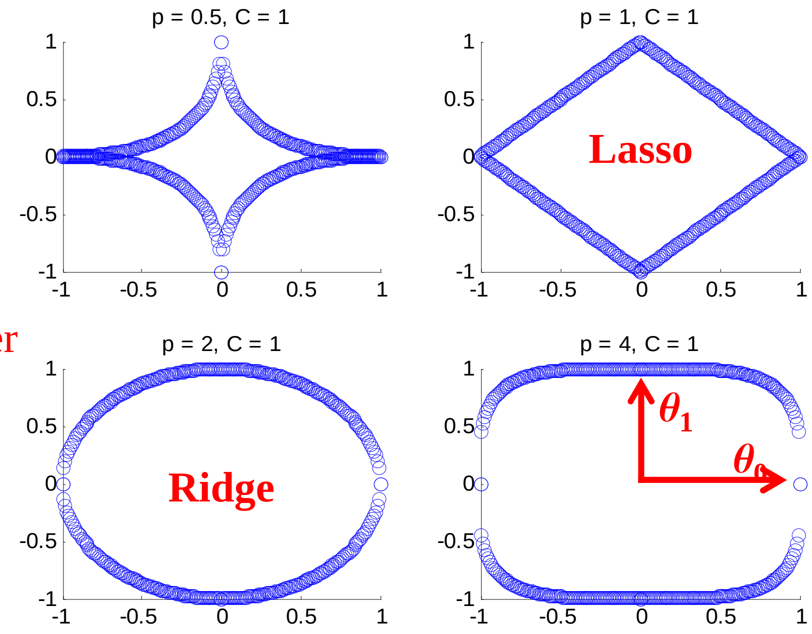
L_p regularizer = $(\sum_{j=1}^{n} |\theta_j|^p)^{1/p}$ or $\|\theta\|_p$

- Balance between sparsity and convexity.
  - p < 1 ➔ regularizer more convex, **easier** to optimize.
    - MSE contour easier to touch protruding isosurface.
  - p = 1 is the only case of sparsity and convexity.
  - p > 1 ➔ regularizer less convex, **harder** to optimize.
    - MSE contour harder to touch protruding isosurface.

MSE contour



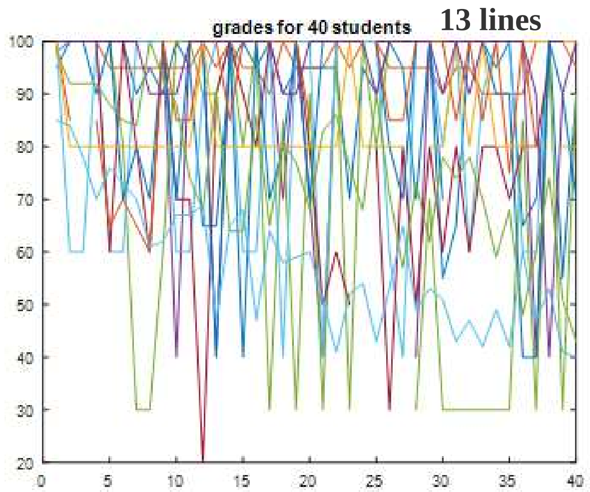
L_p regularizer



Example– Predicting Class Grades

- Grades of 40 students w/ **13** predictors➔ **large # of predictors w.r.t. # records**.
  - 10 homework assignments. 10%
  - 1 semester project. 25%
  - 1 midterm exam. 30%
  - 1 final exam. 35%
  - 1 overall weighted score. dependent variable (i.e. response).
  - 2^{13} = 8192 possible models, each using a subset of variables. **Try all subsets???**

- Purposes:
  - We know the true regression coefficients (see above).
  - To see how well LR can uncover this info.
  - To see how well regularization can identity truly useful info based on the known info.



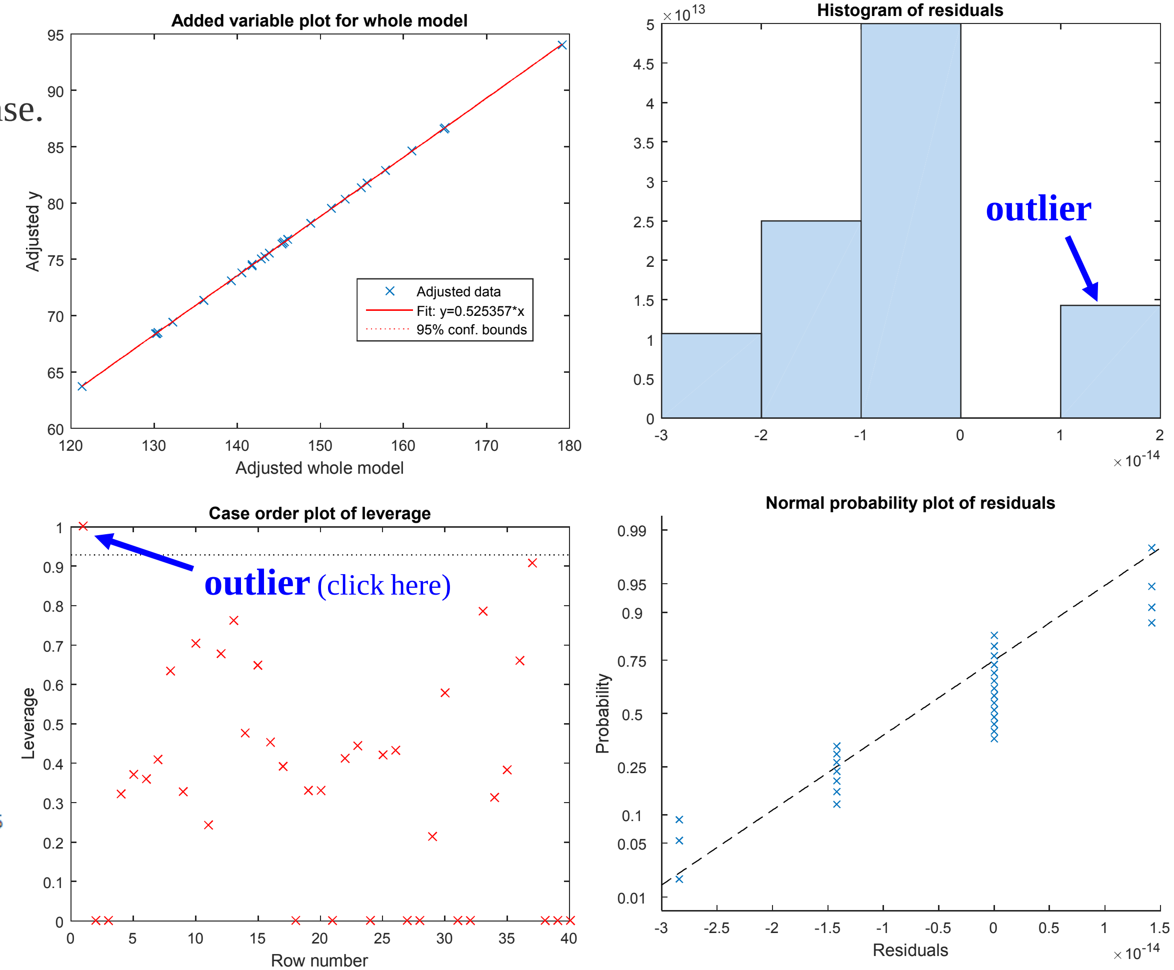
Linear Regression Results

- Grades from 40 students.
  - 10 homework assignments. 10%
  - 1 semester project. 25%
  - 1 midterm exam. 30%
  - 1 final exam. 35%
  - 1 overall weighted score. Response.

Good prediction w/ a **<u>complex</u>** model.

	Estimate	SE	tStat	pValue
(Intercept)	0	0	NaN	NaN
x1	0.01	0	Inf	0
x2	0.01	0	Inf	0
x3	0.01	0	Inf	0
x4	0.01	0	Inf	0
x5	0.01	0	Inf	0
x6	0.01	0	Inf	0
x7	0.01	0	Inf	0
x8	0.01	0	Inf	0
x9	0.01	0	Inf	0
x10	0.01	0	Inf	0
x11	0.25	0	Inf	0
x12	0.3	0	Inf	0
x13	0.35	0	Inf	0

```
Number of observations: 28, Error degrees of freedom: 15
Root Mean Squared Error: 0
R-squared: 1,  Adjusted R-Squared 1
F-statistic vs. constant model: Inf, p-value = 0
```



p-Value

- Reliability ≠ Importance
  - http://blog.minitab.com/blog/adventures-in-statistics-2/how-to-identify-the-most-important-predictor-variables-in-regression-models

Don’t Compare P-values to Determine Variable Importance

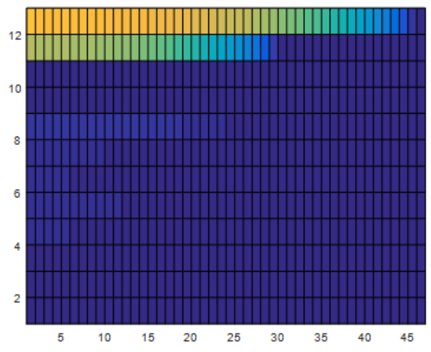
The coefficient value doesn’t indicate the importance a variable, but what about the variable’s p-value?
to help determine whether the variable should be included in the model in the first place.

P-value calculations incorporate a variety of properties, but a measure of importance is not among ther
properties other than importance, such as a very precise estimate and a large sample size.

Effects that are trivial in the real world can have very low p-values. A statistically significant result may n

Takeaway: Low p-values don’t necessarily identify predictor variables that are practically important.

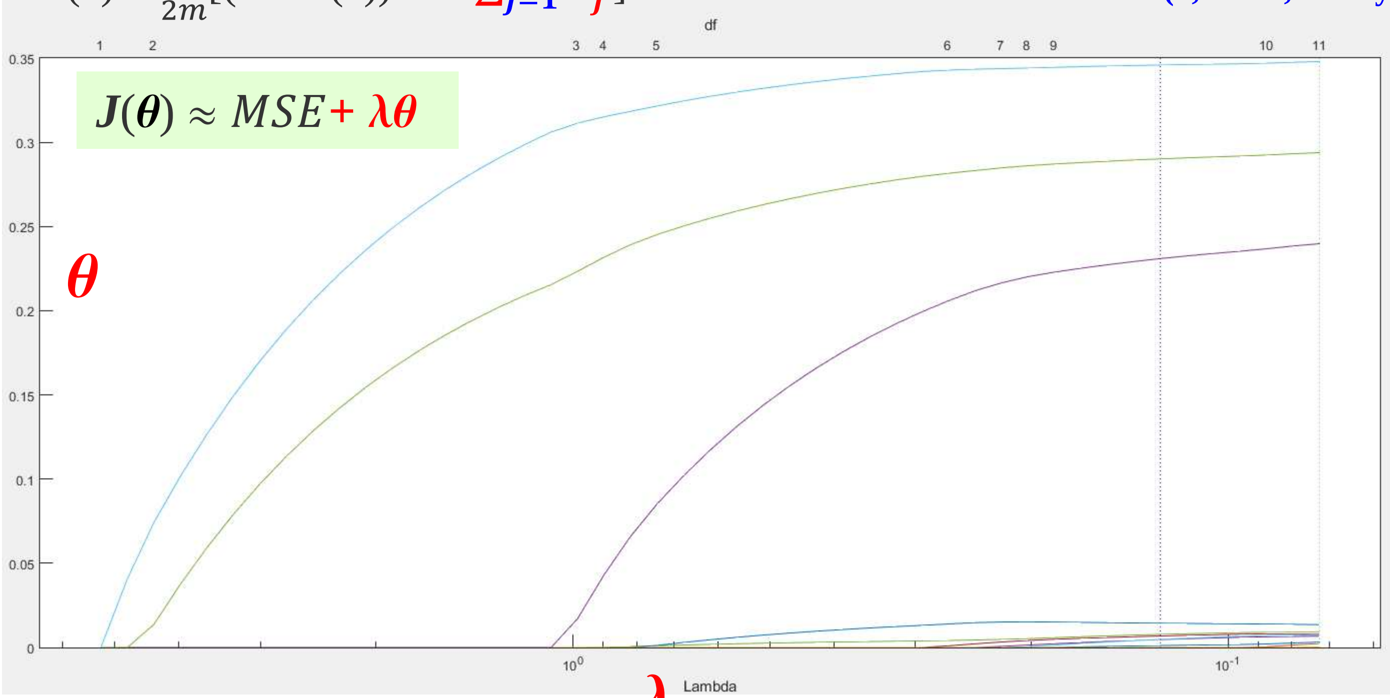
Lasso Identifies Unnecessary Predictors

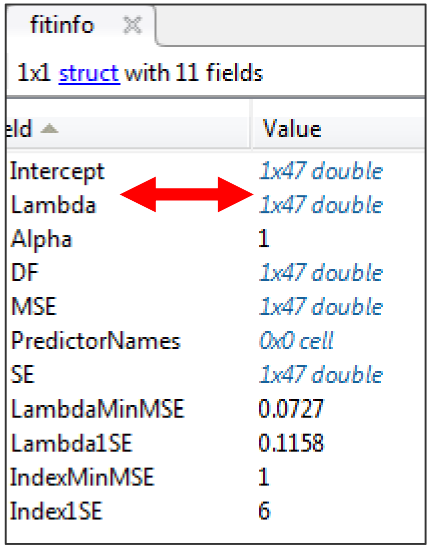


- lassoplot() shows **<u>all</u>** coefficients in regression w/ various λ.
  - Larger λ (regularization parameter) on the left side of the graph.
  - Larger λ has more regularization ➔ more zero coefficients $\boldsymbol{\theta}$.
  - More zero coefficients $\boldsymbol{\theta}$ ➔ less complex model.
  - Lasso tells us Project, Midterm, and Final exams are more important to final grade.

```
allX = xlsread('SEIS01.xlsx'); X = allX(:,1:13); Y = allX(:,14);
[b fitinfo] = lasso(X, Y, 'CV',10, 'Alpha', 1);
lassoPlot(b,fitinfo,'PlotType', 'Lambda', 'XScale', 'log');
```

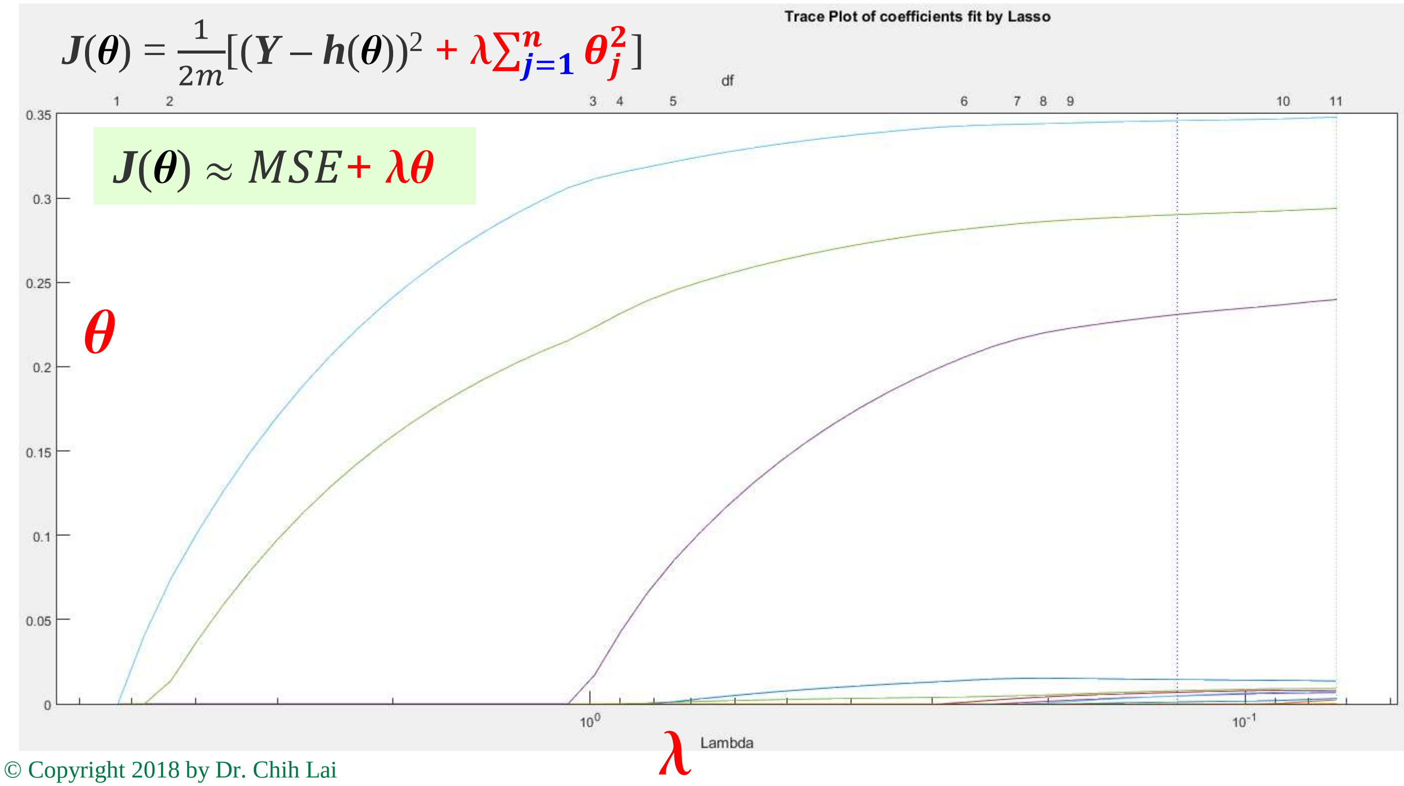
$$J(\boldsymbol{\theta}) = \frac{1}{2m}[(\boldsymbol{Y} - \boldsymbol{h}(\boldsymbol{\theta}))^2 + \lambda\sum_{j=1}^{n} \boldsymbol{\theta}_j^2]$$





Interpreting Lasso Plot

- Dashed lines
  - Green = λ with minimum MSE (MMSE), blue = MMSE + 1σ (recommended λ).
  - Dashed lines appear **<u>only</u>** if you perform ***cross validation*** by setting 'CV' parameters.
- Degrees of freedom (df) = # of nonzero coefficients in regression, w.r.t. λ.
- Larger λ has more regularization ➔ more zero coefficients.
- Small λ, coefficients are close to the least-squares estimate. **(see next slide)**



lasso(X, Y, 'CV',10)

fitinfo

1x1 struct with 11 fields

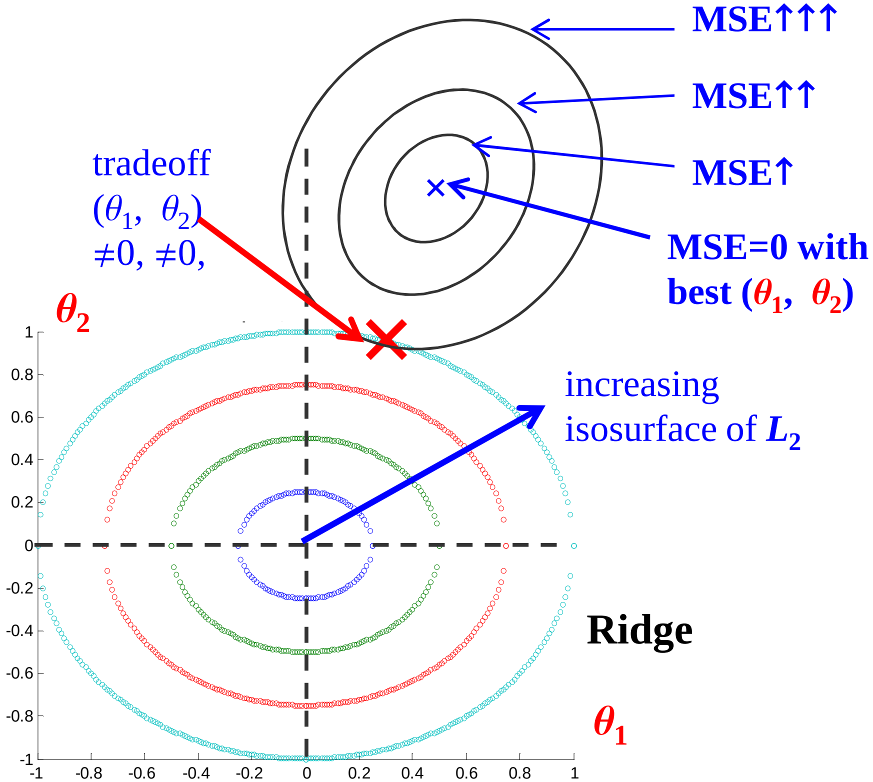
Field	Value
Intercept	1x47 double
Lambda	1x47 double
Alpha	1
DF	1x47 double
MSE	1x47 double
PredictorNames	0x0 cell
SE	1x47 double
LambdaMinMSE	0.0727
Lambda1SE	0.1158
IndexMinMSE	1
Index1SE	6

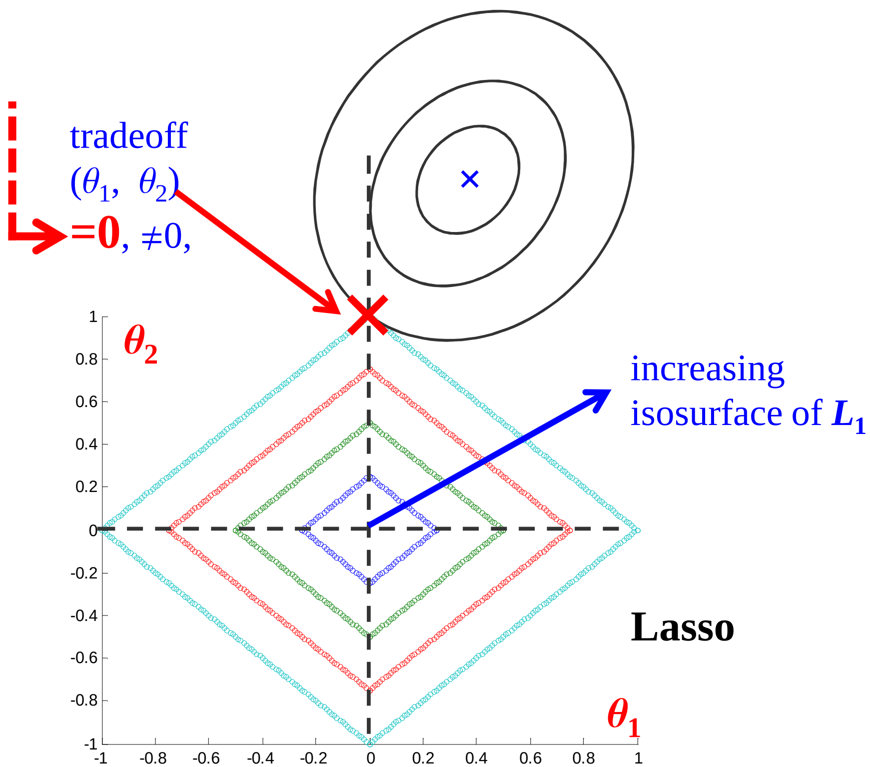
What If $\lambda = 0$??

- $\boldsymbol{J}(\boldsymbol{\theta}) = \frac{1}{2m}[(\boldsymbol{Y} - \boldsymbol{h}(\boldsymbol{\theta}))^2 + \lambda\sum_{j=1}^{n}\boldsymbol{\theta}_j^2] = \frac{1}{2m}[(Y - X\theta)^{\mathrm{T}}(Y - X\theta) + \lambda\sum_{j=1}^{n}\boldsymbol{\theta}_j^2]$.

$\boldsymbol{J}(\boldsymbol{\theta}) \approx MSE + \lambda\boldsymbol{\theta}$

- isosurface of $\boldsymbol{L_p}$ regularizer = $(\sum_{j=1}^{n}|\theta_i|^p)^{1/p}$ or $\|\theta\|_p$
- Need to balance out between increasing MSE and increasing regularizer (penalty).





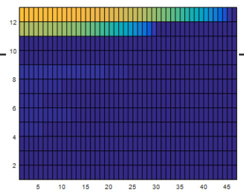
Python scikit-learn Lasso

$J(\boldsymbol{\theta}) \approx MSE + \lambda\theta$

$J(\theta) \approx MSE + \alpha\theta$

```
from sklearn.linear_model import Lasso
import numpy as np

X = [[0,0], [1, 1], [2, 2]];
Y = [0, 1, 2]

for aa in np.arange(0.0000001, 1.1, 0.1):
    clf = Lasso(alpha = aa)
    clf.fit(X, Y)
    print(aa, clf.coef_, clf.intercept_)
```

```
1e-07 [ 0.99999985  0.        ] 1.49999999977e-07
0.1000001 [ 0.84999985  0.        ] 0.15000015
0.2000001 [ 0.69999985  0.        ] 0.30000015
0.3000001 [ 0.54999985  0.        ] 0.45000015
0.4000001 [ 0.39999985  0.        ] 0.60000015
0.5000001 [ 0.24999985  0.        ] 0.75000015
0.6000001 [ 0.09999985  0.        ] 0.90000015
0.7000001 [ 0.  0.] 1.0
0.8000001 [ 0.  0.] 1.0
0.9000001 [ 0.  0.] 1.0
1.0000001 [ 0.  0.] 1.0
```



```
from sklearn import linear_model

X = [[0, 0], [0, 0], [1, 1]]; Y = [0, .1, 1]
default_alpha = 0.1          ### alpha = lambda

########################################
reg = linear_model.Lasso(alpha = default_alpha)
reg.fit(X, Y)
print(reg.coef_, reg.intercept_)
reg.predict([[1, 1]])


########################################
reg = linear_model.Ridge(alpha = default_alpha)
reg.fit (X, Y)
print(reg.coef_, reg.intercept_)

########################################
from sklearn.linear_model import ElasticNet
reg = ElasticNet(alpha = default_alpha, l1_ratio = 0.5)
reg.fit (X, Y)
print(reg.coef_, reg.intercept_)
```

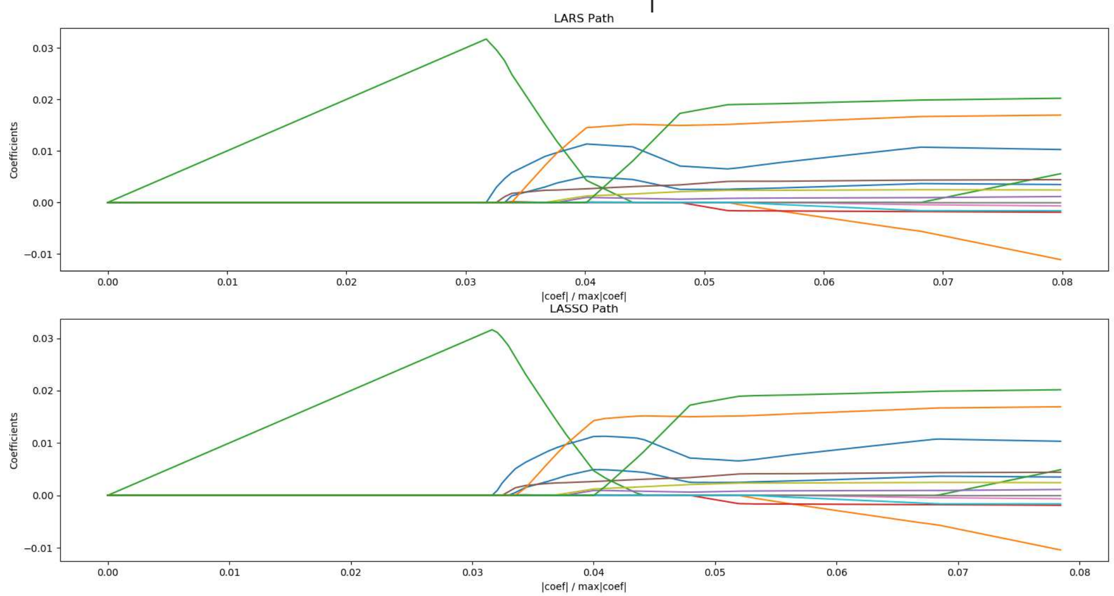
Python scikit-learn Various Lasso Functions

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model
from sklearn.linear_model import lasso_path, lars_path, Lasso, enet_path

dataset = np.genfromtxt("D:/GradeExample.csv", delimiter = ',')

# split into input (X) and output (Y) variables
Y = dataset[:,14];   X = dataset[:,0:13];      X[np.isnan(X)] = 0     # replace all NANs to 0

print("Regularization path using lars_path")       # least angle regression
alphas1, active1, coefs1 = lars_path(X, Y, method='lasso', verbose=True)

print("Regularization path using lasso_path")
eps = 5e-6    # the smaller it is the longer is the path  ← you have control
alphas2, coefs2, _ = lasso_path(X, Y, eps)          # don't know why need the 3rd output

print("ONE regularization using Lasso")
clf = Lasso(fit_intercept=False, alpha=1.3128)
clf.fit(X, Y)
print(clf.intercept_, clf.coef_)
```

$$J(\boldsymbol{\theta}) \approx MSE + \lambda\theta$$

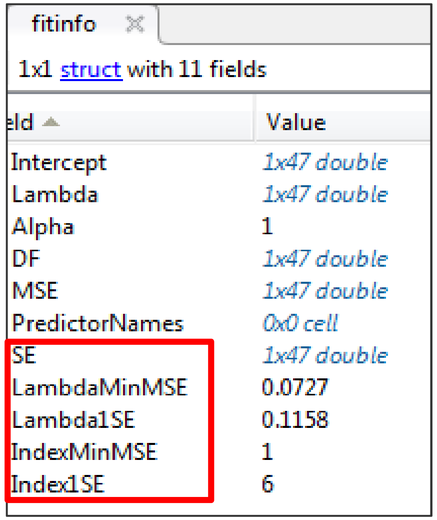
$$J(\theta) \approx MSE + \alpha\theta$$

```
plt.subplot(211)
xx = np.sum(np.abs(coefs1.T), axis=1)
plt.plot(xx, coefs1.T)
ymin, ymax = plt.ylim()
#plt.vlines(xx, ymin, ymax, linestyle='dashed')
plt.xlabel('|coef| / max|coef|')
plt.ylabel('Coefficients')
plt.title('LARS Path')
```



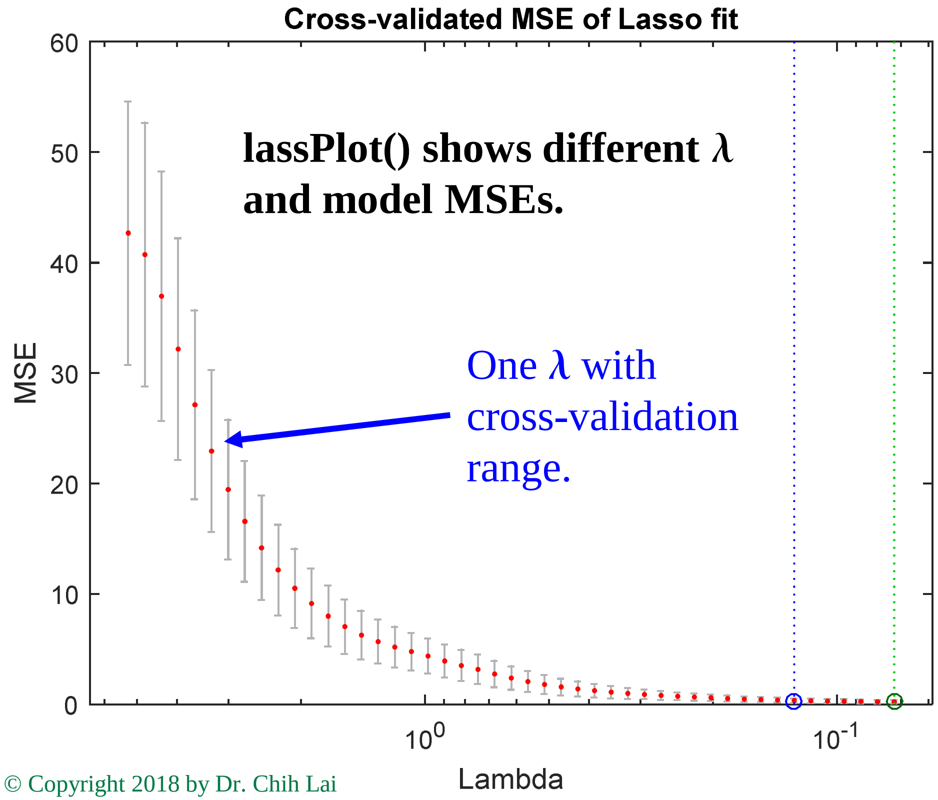
MSE, λ and Cross Validation

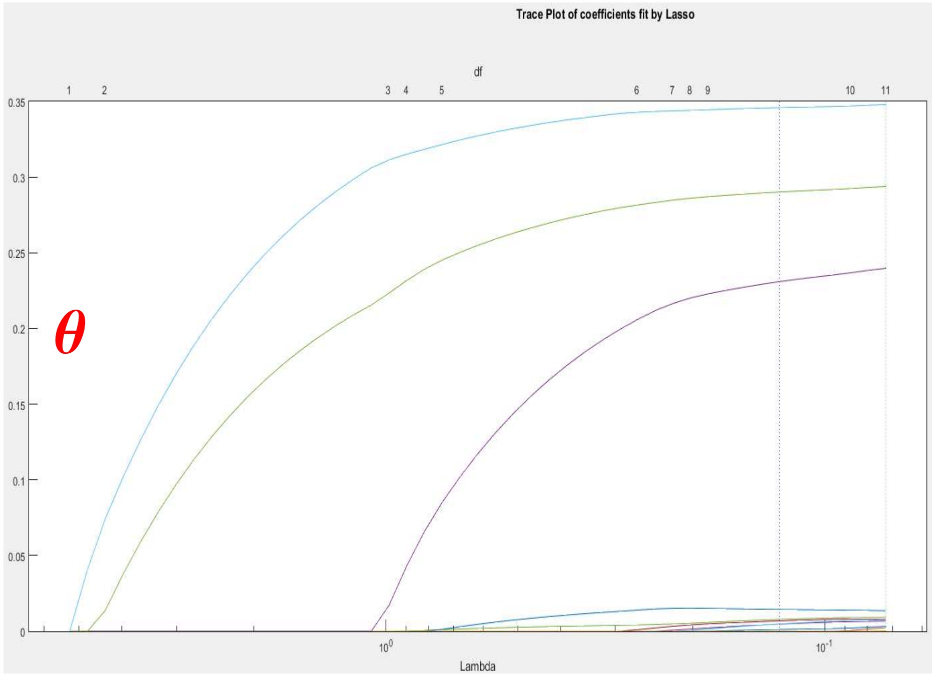
$J(\theta) \approx MSE + \lambda\theta$



Field	Value
Intercept	1x47 double
Lambda	1x47 double
Alpha	1
DF	1x47 double
MSE	1x47 double
PredictorNames	0x0 cell
SE	1x47 double
LambdaMinMSE	0.0727
Lambda1SE	0.1158
IndexMinMSE	1
Index1SE	6

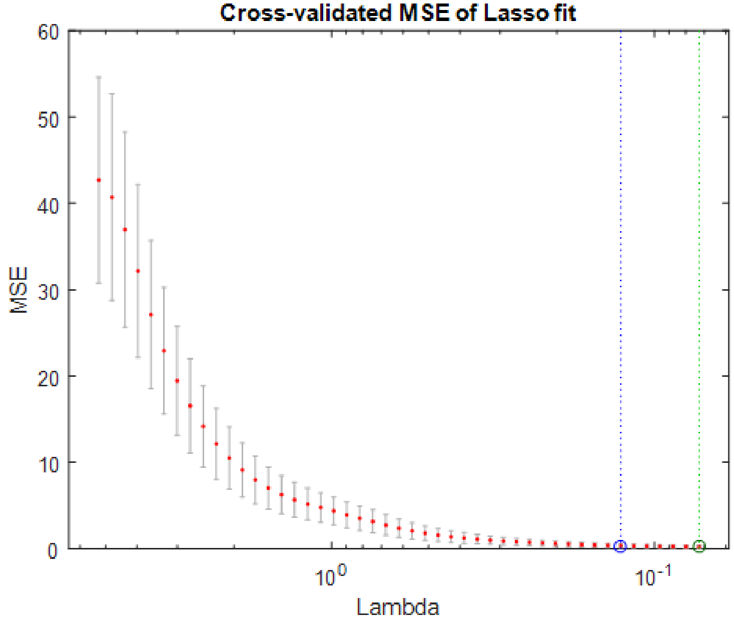
- $\lambda\uparrow$ ➔ MSE↑. $\lambda\uparrow$ ➔ more zero coefficients.
  - MSE range during **cross-validation**.
  - [b fitinfo] = lasso(X, Y, 'CV', **10**, ~~'Alpha', 1~~);
  - lassoPlot(b,fitinfo,'PlotType', 'Lambda', 'XScale', 'log');
  - **lassoPlot(b,fitinfo,'PlotType','CV');** **% lassPlot with Cross Validation.**

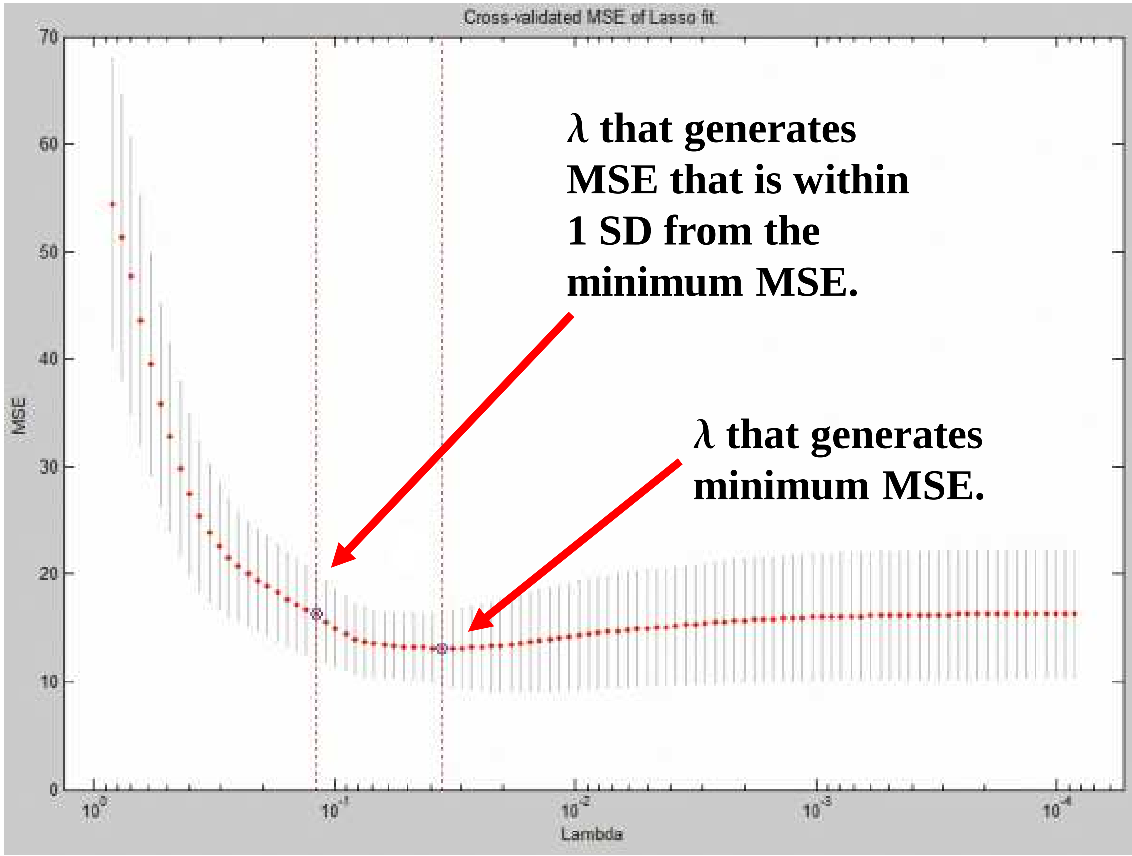




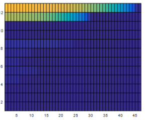
Lasso Plot with Cross Validation

- It shows different λ used in different models and different MSEs.
- Choose λ between the two vertical bars.





http://www.mathworks.com/discovery/regularization.html



Relation of θ and λ

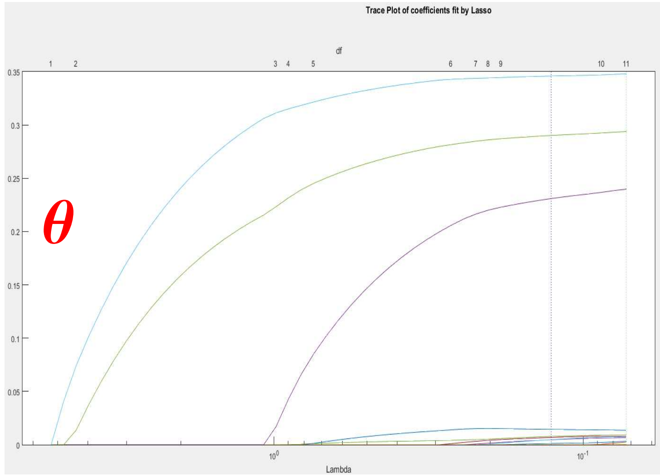
$J(\boldsymbol{\theta}) \approx MSE + \lambda\boldsymbol{\theta}$

- [***b*** fitinfo] = lasso(X, Y, 'CV',10, 'Alpha', 1);
  - ***b*** stores θ (or β), containing model parameters.
  - To visualize how $\boldsymbol{\theta}$ changes w.r.t. $\boldsymbol{\lambda}$.

Name ▲	Value
b	*13x47 double*

Intercept	*1x47 double*
Lambda	*1x47 double*

How to create this plot??

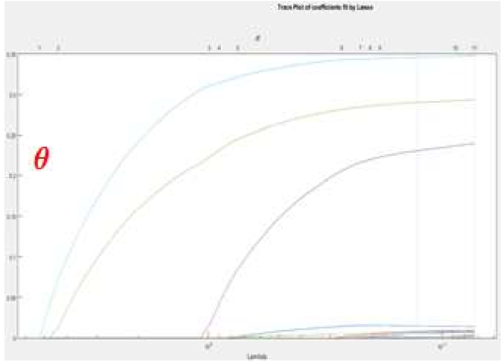


smaller λ → larger λ

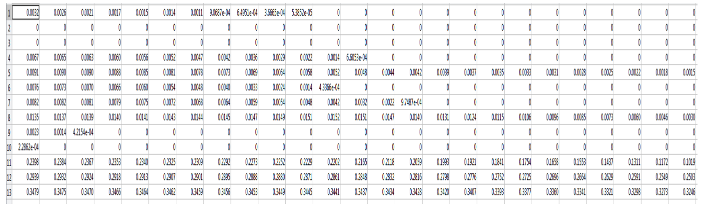
b_i* or *θ_i

1	0.0032	0.0026	0.0021	0.0017	0.0015	0.0014	0.0011	9.0687e-04	6.4951e-04	3.6665e-04	5.3852e-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0.0067	0.0065	0.0063	0.0060	0.0056	0.0052	0.0047	0.0042	0.0036	0.0029	0.0022	0.0014	6.6053e-04	0	0	0	0	0	0	0	0	0	0	0	0
5	0.0091	0.0090	0.0090	0.0088	0.0085	0.0081	0.0078	0.0073	0.0069	0.0064	0.0058	0.0052	0.0048	0.0044	0.0042	0.0039	0.0037	0.0035	0.0033	0.0031	0.0028	0.0025	0.0022	0.0018	0.0015
6	0.0076	0.0073	0.0070	0.0066	0.0060	0.0054	0.0048	0.0040	0.0033	0.0024	0.0014	4.3366e-04	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0.0082	0.0082	0.0081	0.0079	0.0075	0.0072	0.0068	0.0064	0.0059	0.0054	0.0048	0.0042	0.0032	0.0022	9.7487e-04	0	0	0	0	0	0	0	0	0	0
8	0.0135	0.0137	0.0139	0.0140	0.0141	0.0143	0.0144	0.0145	0.0147	0.0149	0.0151	0.0152	0.0151	0.0147	0.0140	0.0131	0.0124	0.0115	0.0106	0.0096	0.0085	0.0073	0.0060	0.0046	0.0030
9	0.0023	0.0014	4.2154e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	2.2862e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0.2398	0.2384	0.2367	0.2353	0.2340	0.2325	0.2309	0.2292	0.2273	0.2252	0.2229	0.2202	0.2165	0.2118	0.2059	0.1993	0.1921	0.1841	0.1754	0.1658	0.1553	0.1437	0.1311	0.1172	0.1019
12	0.2939	0.2932	0.2924	0.2918	0.2913	0.2907	0.2901	0.2895	0.2888	0.2880	0.2871	0.2861	0.2848	0.2832	0.2816	0.2798	0.2776	0.2752	0.2725	0.2696	0.2664	0.2629	0.2591	0.2549	0.2503
13	0.3479	0.3475	0.3470	0.3466	0.3464	0.3462	0.3459	0.3456	0.3453	0.3449	0.3445	0.3441	0.3437	0.3434	0.3428	0.3420	0.3407	0.3393	0.3377	0.3360	0.3341	0.3321	0.3298	0.3273	0.3246

Relation of θ and λ − 2D and 3-D View

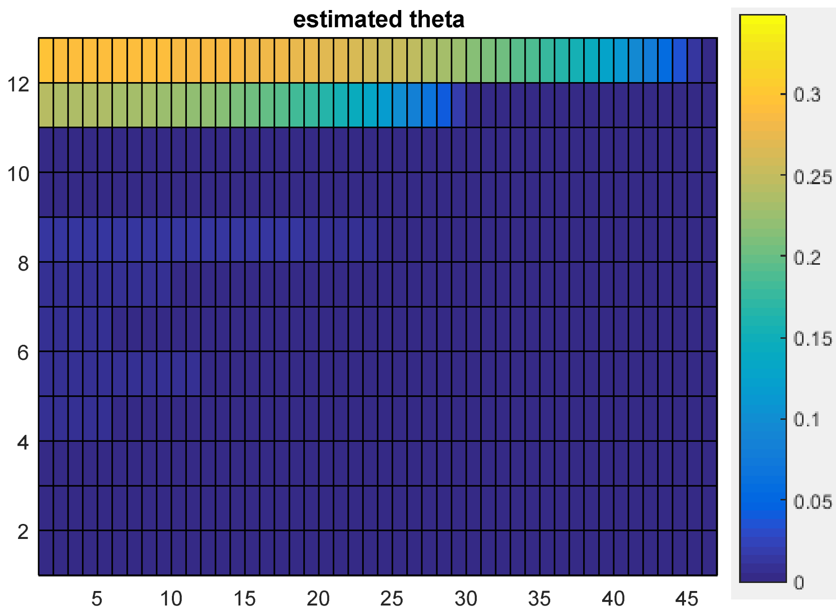


- [***b*** fitinfo] = lasso(X, Y, 'CV',10, 'Alpha', 1);
  - ***b*** represents θ (or β), containing model parameters.
  - To visualize how θ changes w.r.t. λ.
  - figure, **pcolor(*b*),** **colorbar** **% imagesc(*b*)**
  - figure, **mesh(*b*)**

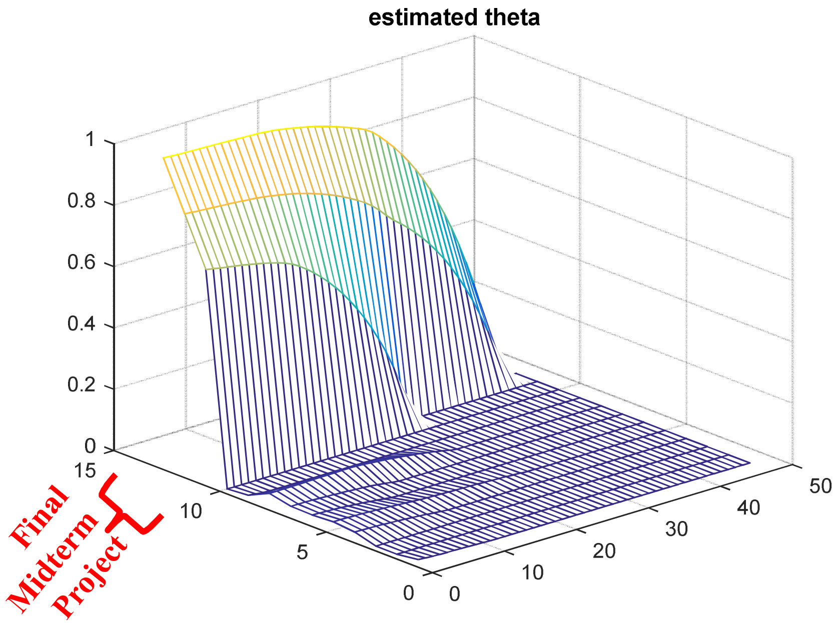


Final Midterm Project

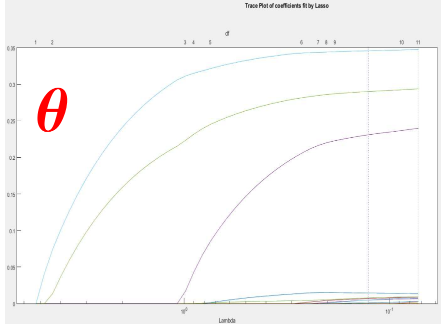
b$_i$* or *θ_i



smaller λ → larger λ



Prediction Using Lasso Results, HOW??



- [***b*** fitinfo] = lasso(X, Y, 'CV',10, 'Alpha', 1);
  - ***b*** represents θ (or β), containing model parameters.

$$J(\boldsymbol{\theta}) \approx MSE + \lambda\boldsymbol{\theta}$$

selected λ

[Lb, Lfitinfo] = lasso(X, Y, 'Alpha', 1, **'Lambda', 0.2674**);

smaller λ → larger λ

b_i or θ_i

1	0.0032	0.0026	0.0021	0.0017	0.0015	0.0014	0.0011	9.0687e-04	6.4951e-04	3.6665e-04	5.3852e-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0.0067	0.0065	0.0063	0.0060	0.0056	0.0052	0.0047	0.0042	0.0036	0.0029	0.0022	0.0014	6.6053e-04	0	0	0	0	0	0	0	0	0	0	0	0
5	0.0091	0.0090	0.0090	0.0088	0.0085	0.0081	0.0078	0.0073	0.0069	0.0064	0.0058	0.0052	0.0048	0.0044	0.0042	0.0039	0.0037	0.0035	0.0033	0.0031	0.0028	0.0025	0.0022	0.0018	0.0015
6	0.0076	0.0073	0.0070	0.0066	0.0060	0.0054	0.0048	0.0040	0.0033	0.0024	0.0014	4.3366e-04	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0.0082	0.0082	0.0081	0.0079	0.0075	0.0072	0.0068	0.0064	0.0059	0.0054	0.0048	0.0042	0.0032	0.0022	[illegible].7487e-04	0	0	0	0	0	0	0	0	0	0
8	0.0135	0.0137	0.0139	0.0140	0.0141	0.0143	0.0144	0.0145	0.0147	0.0149	0.0151	0.0152	0.0151	0.0147	0.0140	0.0131	0.0124	0.0115	0.0106	0.0096	0.0085	0.0073	0.0060	0.0046	0.0030
9	0.0023	0.0014	4.2154e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	2.2862e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0.2398	0.2384	0.2367	0.2353	0.2340	0.2325	0.2309	0.2292	0.2273	0.2252	0.2229	0.2202	0.2165	0.2118	0.2059	0.1993	0.1921	0.1841	0.1754	0.1658	0.1553	0.1437	0.1311	0.1172	0.1019
12	0.2939	0.2932	0.2924	0.2918	0.2913	0.2907	0.2901	0.2895	0.2888	0.2880	0.2871	0.2861	0.2848	0.2832	0.2816	0.2798	0.2776	0.2752	0.2725	0.2696	0.2664	0.2629	0.2591	0.2549	0.2503
13	0.3479	0.3475	0.3470	0.3466	0.3464	0.3462	0.3459	0.3456	0.3453	0.3449	0.3445	0.3441	0.3437	0.3434	0.3428	0.3420	0.3407	0.3393	0.3377	0.3360	0.3341	0.3321	0.3298	0.3273	0.3246

$$\theta = \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \quad X = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 1 & 2 & 3 \end{bmatrix},$$

$$\hat{y} = \theta^{\mathrm{T}}\boldsymbol{X} = [0 \;\; 4] \times \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 1 & 2 & 3 \end{bmatrix}$$

$$= [4 \;\; 8 \;\; 12].$$

$$\theta = \begin{bmatrix} \boldsymbol{\theta_0} \\ \cdots \\ \theta_p \end{bmatrix}, \qquad X = \begin{bmatrix} \mathbf{1} & \cdots & \mathbf{1} \\ \vdots & \ddots & \vdots \\ X1_P & \cdots & Xm_P \end{bmatrix}$$

% $\hat{y} = \theta^{\mathrm{T}}X$

Yhat = [fitinfo.Intercept(**Ite**); (b(:, **Ite**))]**'** * [**ones(size(X, 1)**, 1) X]**'**;

= X * coef + coef0;

Lasso, Important Predictors– Hospital Example

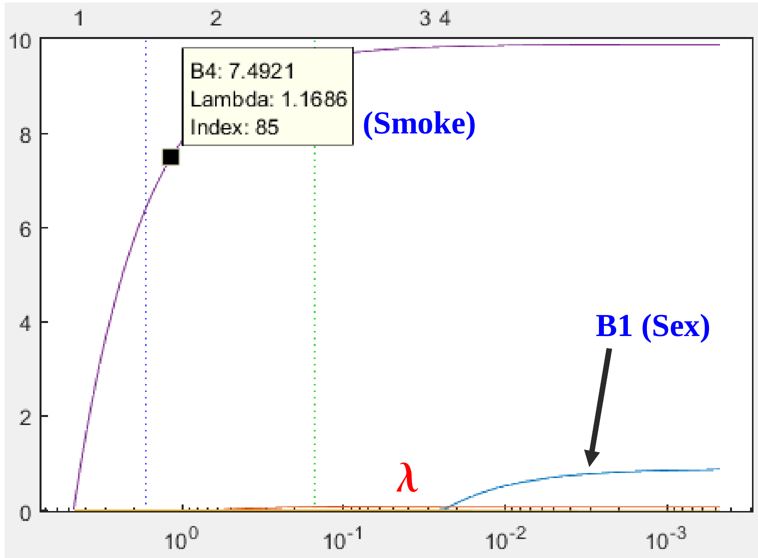
$J(\theta) \approx MSE + \lambda\theta$

- Before regularization (after removing outlier 84).

(Intercept)		115	2.3258e-27
x1	**sex**	0.22181	0.93846
x2	**age**	0.10678	0.10721
x3	**wgt**	0.00036854	0.9946
x4	**smoke**	10.002	2.8087e-16

RMSE = 4.66, R^2 = 0.536, Adj R^2 = 0.516
F-Stats = 27.1

[b fitinfo] = lasso(X, Y, 'CV',10, 'Alpha', 1);
lassoPlot(b,fitinfo,'PlotType', 'Lambda', 'XScale', 'log');



- After lasso, use "sex" & "smoke" predictors.

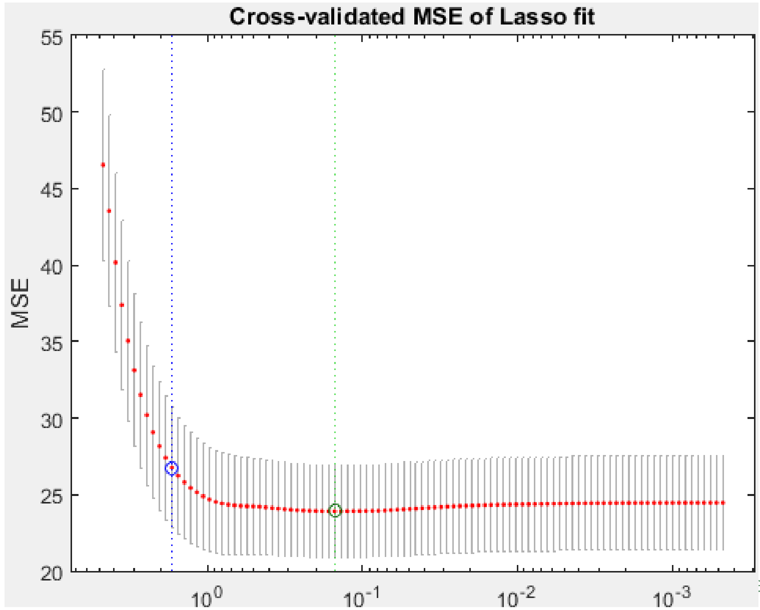
(Intercept)		119.08	3.7353e-121
x1	**sex**	0.33806	0.72584
x2	**smoke**	10.064	1.8505e-16

RMSE = 4.67, R^2 = 0.523, Adj R^2 = 0.513
F-Stats = 52.6

- Use lasso suggestion w/ "smoke" only.

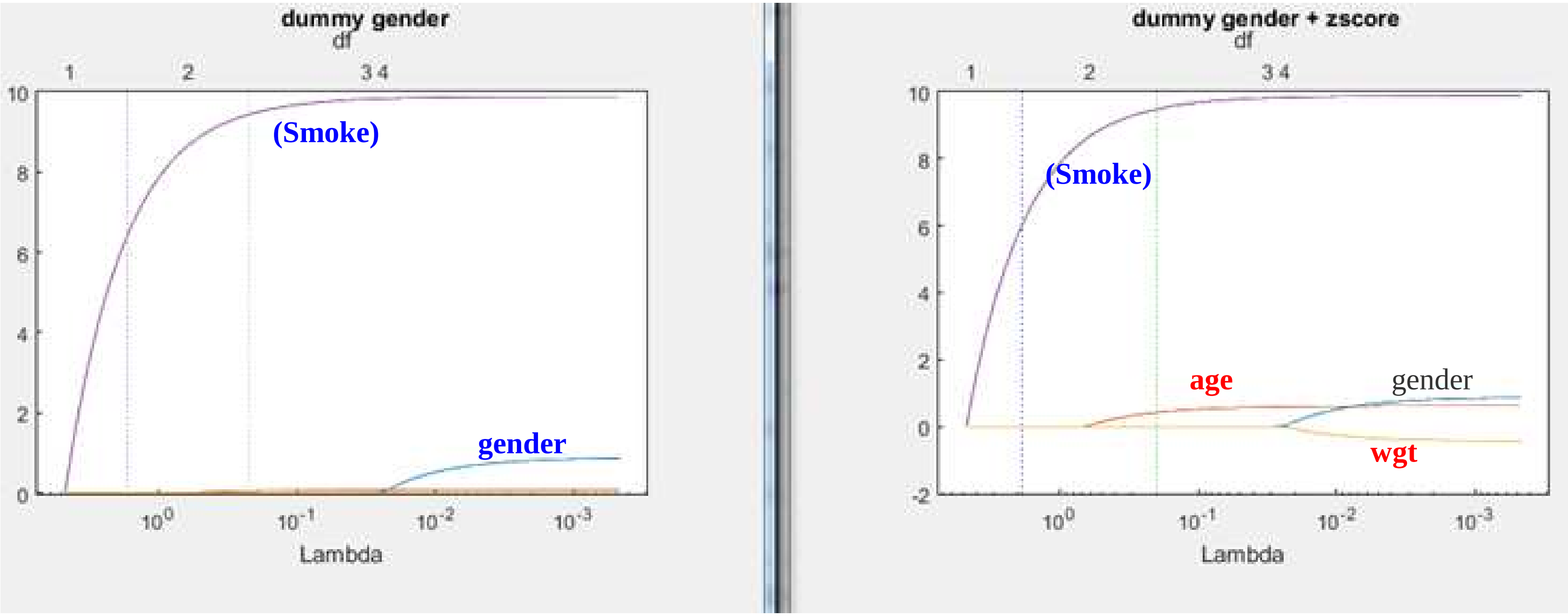
(Intercept)		119.22	4.4124e-130
x1	**smoke**	10.138	3.0632e-17

RMSE = 4.65, R^2 = 0.52, Adj R^2 = 0.517
F-Stats = 106



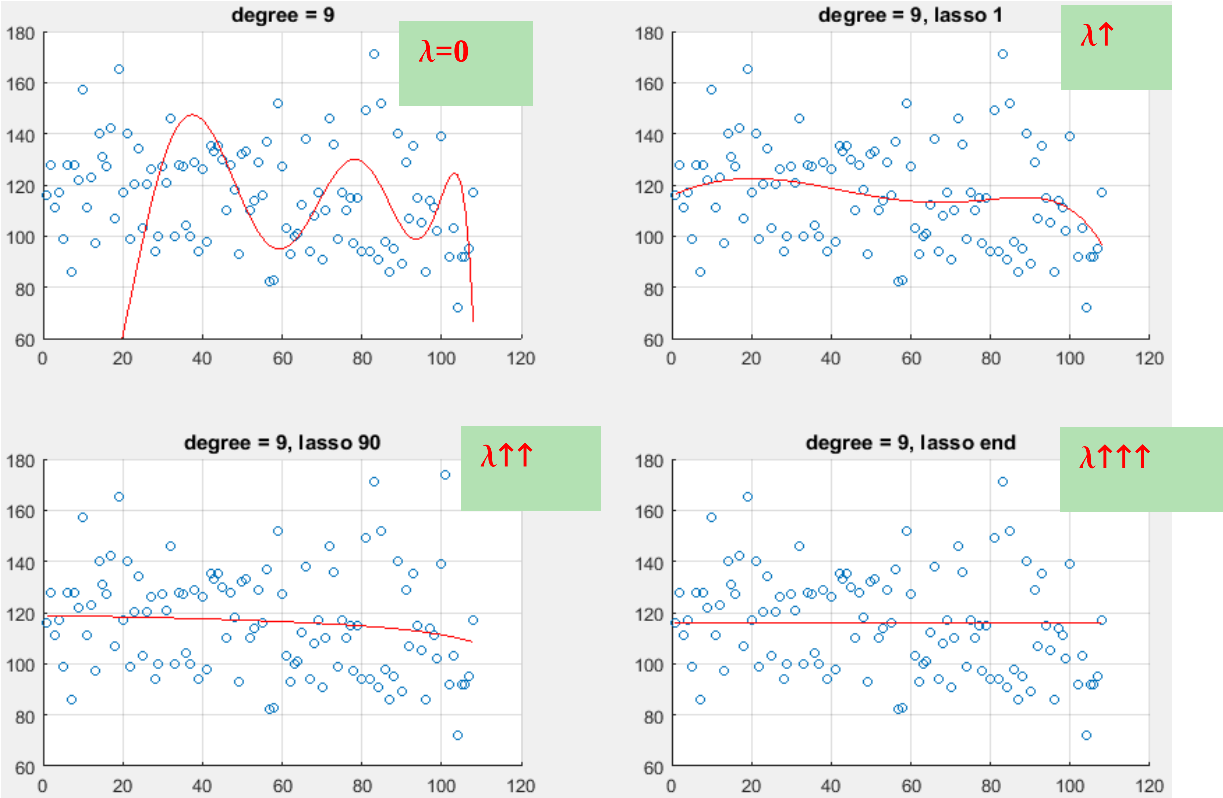
Impact of Standardization

$J(\boldsymbol{\theta}) \approx MSE + \lambda\boldsymbol{\theta}$



Lasso On Blood Sugar Data

$$J(\boldsymbol{\theta}) \approx MSE + \lambda\boldsymbol{\theta}$$



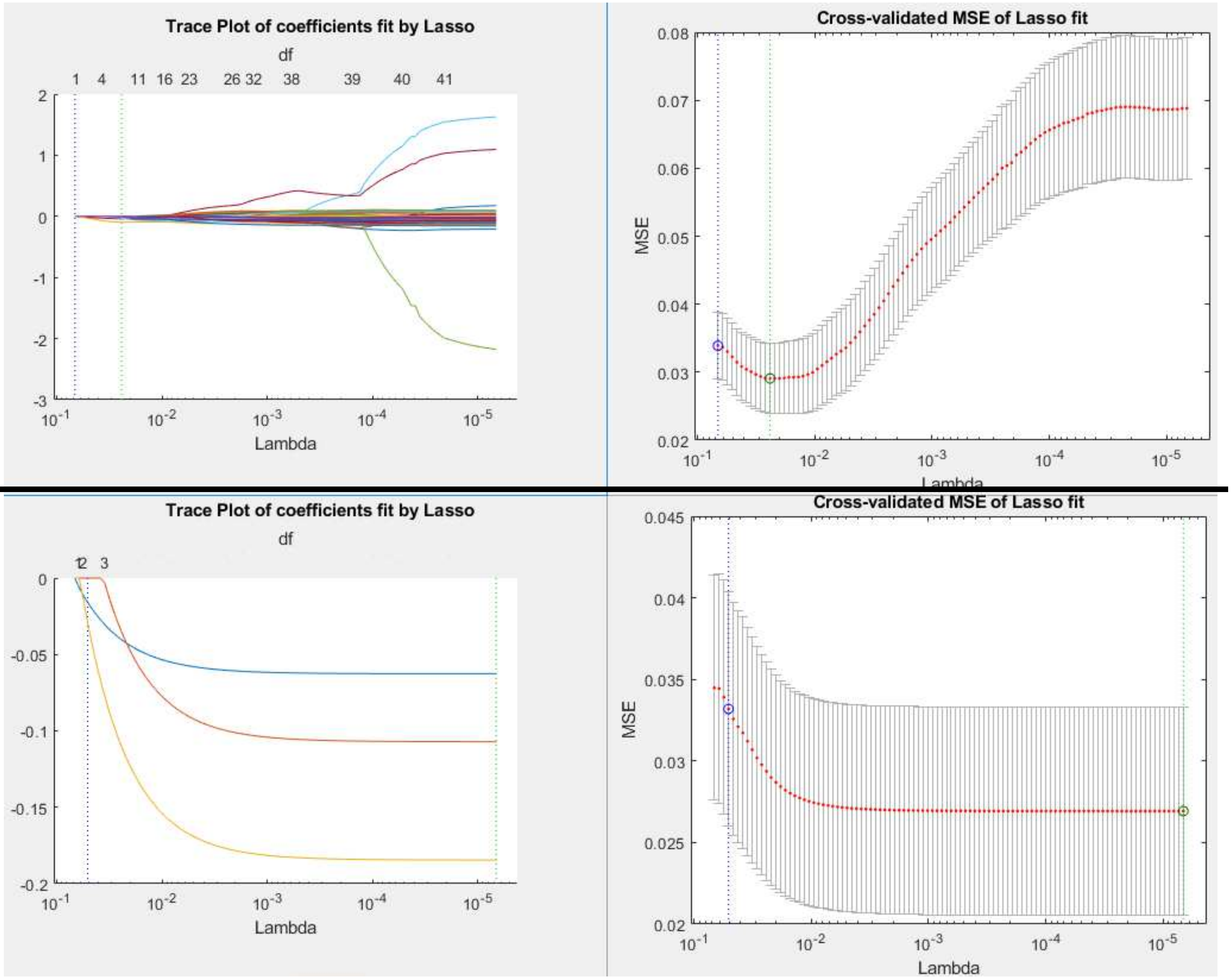
$J(\theta) \approx MSE + \lambda\theta$

Interpreting This Result??

- 91 records of 53 predictors to predict software readability.

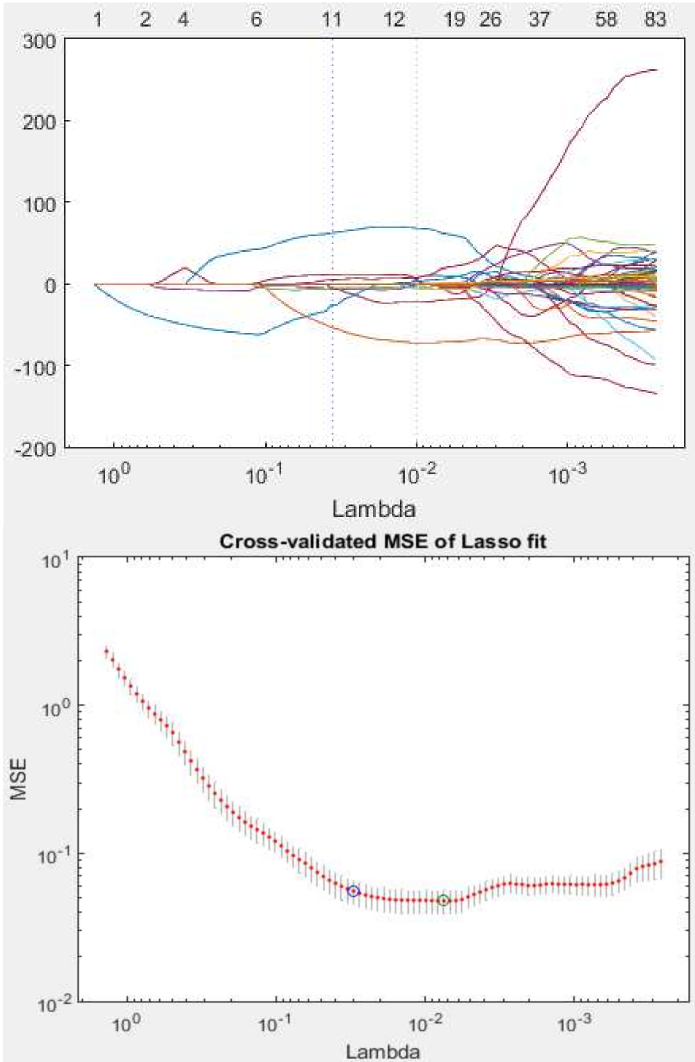
Before Removing Predictors

After Removing Predictors



Parallel Computing for Lasso

```
load spectra                    % 60 gasoline samples, 401 vars

% time-consuming operation
tic,   [b fitinfo] = lasso(NIR,octane,'CV',10);    toc

lassoPlot(b,fitinfo,'PlotType','Lambda','XScale','log');   % plot log x-axis
fitinfo.LambdaMinMSE
fitinfo.Lambda1SE
lambdaindex = fitinfo.Index1SE;        % get the index where 1SE occurs
fitinfo.MSE(lambdaindex)               % MSE at where 1SE occurs
lassoPlot(b,fitinfo,'PlotType','CV');
set(gca,'YScale','log');               % log scale for better view

%% try lasso regression w/ CV using parallel processing here!!

opt = statset('UseParallel', true);

tic,   [b fitinfo] = lasso(NIR,octane, 'CV', 10,  'Options',  opt);    toc
```



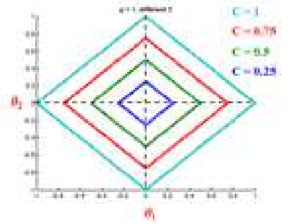
Lasso Summary

- LASSO (***L***east ***A***bsolute ***S***hrinkage and ***S***election ***O***perator)
  - Lasso is a regularization technique for performing linear regression (and **otherSSSS**).
  - Lasso includes a penalty term that constrains magnitude of the estimated coefficients.
  - Reduce # of useless/redundant predictors & identify important predictors.
  - $\boldsymbol{J}(\boldsymbol{\theta}) \approx MSE + \boldsymbol{\lambda\theta}$

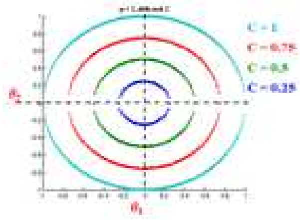
- As λ increases, lasso sets more coefficients to zero.
  - Hence, lasso produces simpler model, with fewer predictors.
  - Hence, lasso $\approx$ to **dimensionality reduction** techniques.
  - ➔ So just use PCA for dimensionality reduction???

See Appendix for Ridge regression.

Lasso and Elastic-Net



- Lasso is a regularization technique.
  - Reduce # of useless/redundant predictors & identify important ones.



- Elastic net is a hybrid of ***ridge*** & ***lasso*** regularization.
  - Relative weighted $\boldsymbol{\alpha}$ of lasso (α = 1) and ridge (α = 0).

- Use elastic net when you have several highly correlated variables.
  - Lasso tends to drop **individual** vars from groups of highly correlated vars.
  - Elastic net tends to retain or drop groups of highly correlated vars w.r.t. $\boldsymbol{\alpha}$

$$\boldsymbol{J(\theta) = \Sigma_i(Y - \theta^T X)^2} \quad + \lambda C \quad \approx MSE + \lambda C, \qquad C = \frac{(1-\alpha)}{2} \times R + \alpha \times \underset{\text{L1 ratio}}{L}$$

Elastic Net Formula

$$\min_{\beta_0, \beta} \left(\frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \beta_0 - x_i^T \beta \right)^2 + \lambda P_\alpha(\beta) \right),$$

where

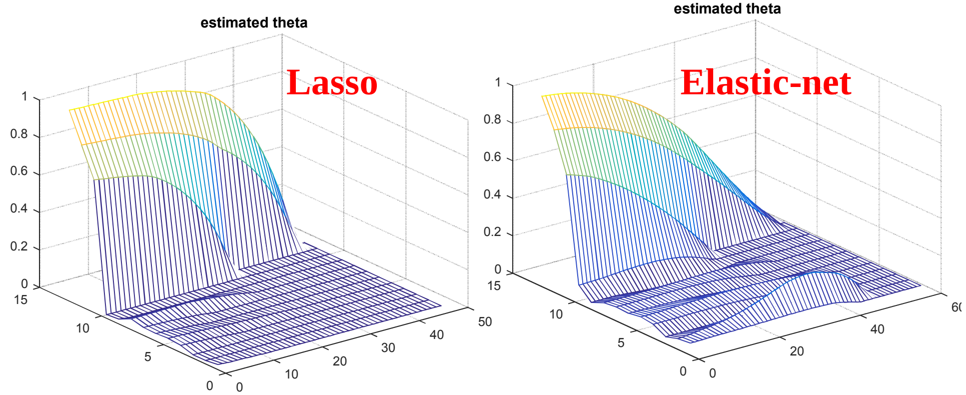
$$P_\alpha(\beta) = \frac{(1-\alpha)}{2} \underbrace{\|\beta\|_2^2}_{\text{Ridge}} + \alpha \underbrace{\|\beta\|_1}_{\text{Lasso}} = \sum_{j=1}^{p} \left(\frac{(1-\alpha)}{2} \beta_j^2 + \alpha |\beta_j| \right).$$

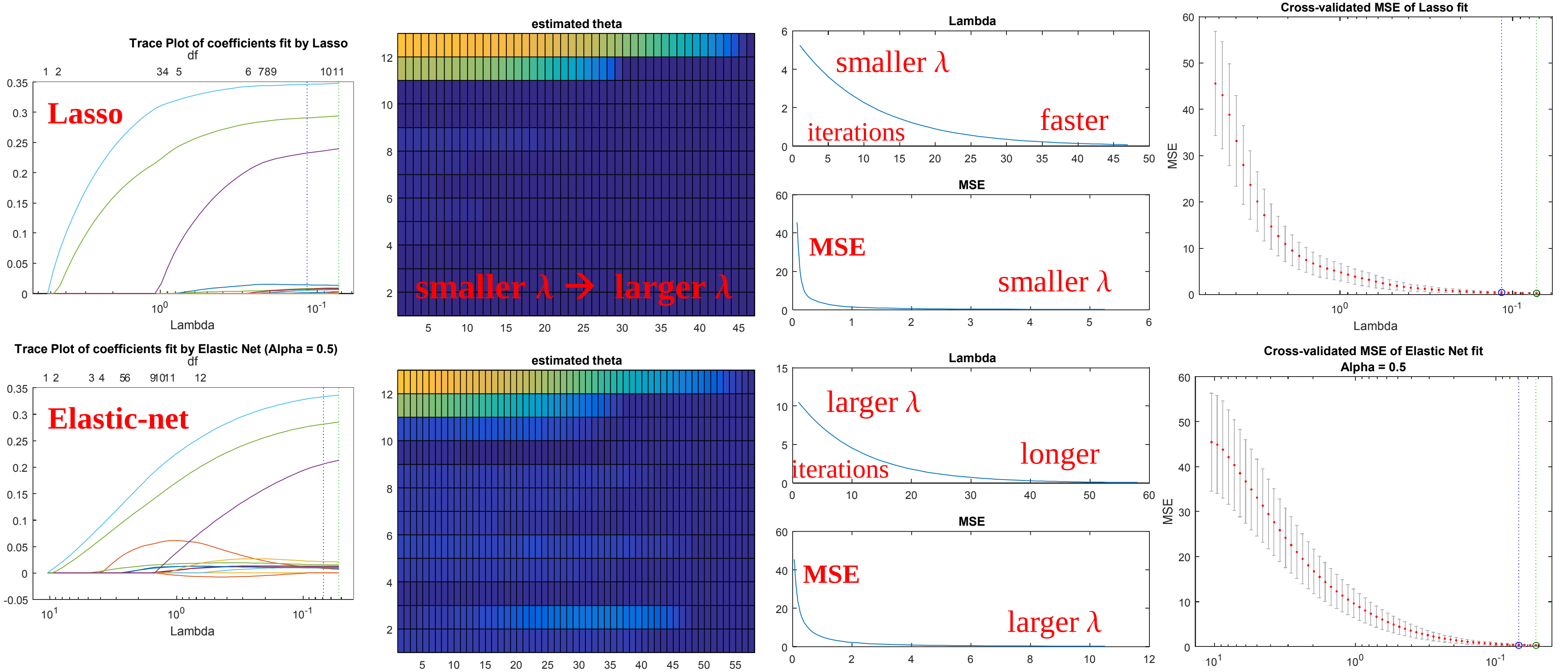
http://www.mathworks.com/help/stats/lasso-and-elastic-net.html

Python Elastic-Net page
1 / (2 * n_samples) * ||y - Xw||^2_2
+ alpha * l1_ratio * ||w||_1
+ 0.5 * alpha * (1 - l1_ratio) * ||w||^2_2

Example– Lasso vs. Elastic-Net

- Lasso vs. Elastic-net
  - On the class grade example.
  - Top row Lasso.
  - Bottom row Elastic-net.
    - More non-0 coefficients.

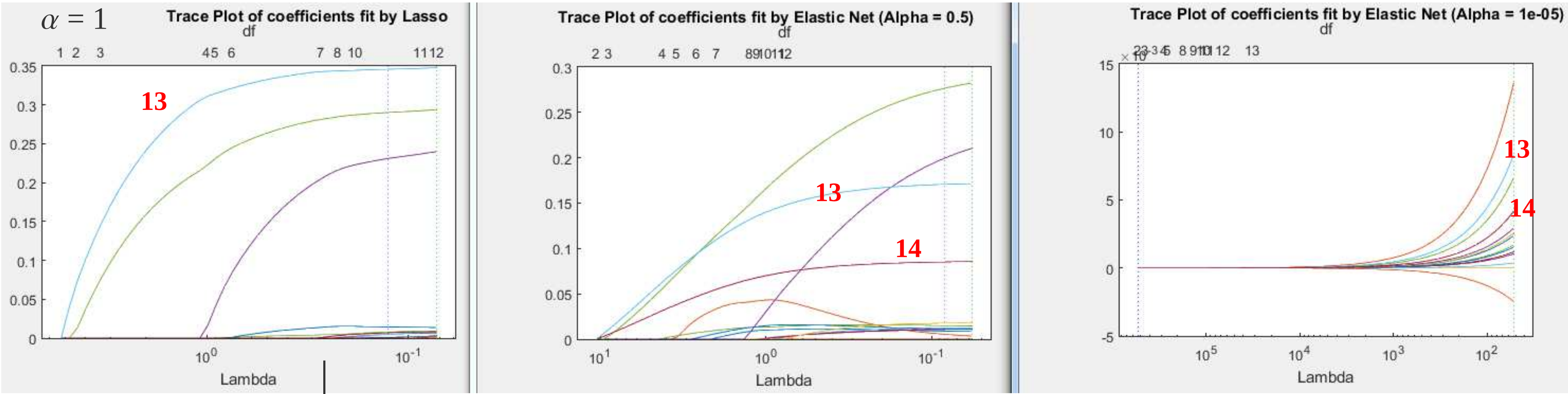




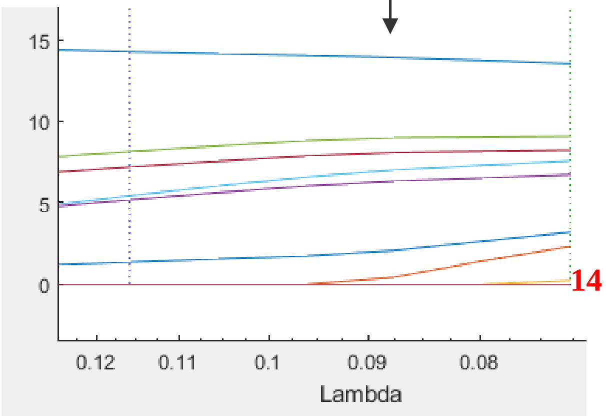
Elastic-Net, Impact of α

- Create 14th attribute that is correlated to 13th attribute (final exam).
  - 14th attribute = (13th attribute) / 2.

$$MSE + \lambda C, \quad C = \frac{(1-\alpha)}{2} \times R + \alpha \times L$$



zoom in




```
from sklearn import linear_model
X = [[0, 0], [0, 0], [1, 1]];                    Y = [0, .1, 1]
default_alpha = 0.1                              ### alpha = lambda
########################################
reg = linear_model.Lasso(alpha = default_alpha)
reg.fit(X, Y)
print(reg.coef_, reg.intercept_)
reg.predict([[1, 1]])
########################################
reg = linear_model.ElasticNet(alpha = default_alpha, l1_ratio = 0.5)
reg.fit (X, Y)
print(reg.coef_, reg.intercept_)
```

When to Use Which Method To Improve Accuracy?

- Feature selection.
  - When smaller # of vars➔ speed outperform regularization.

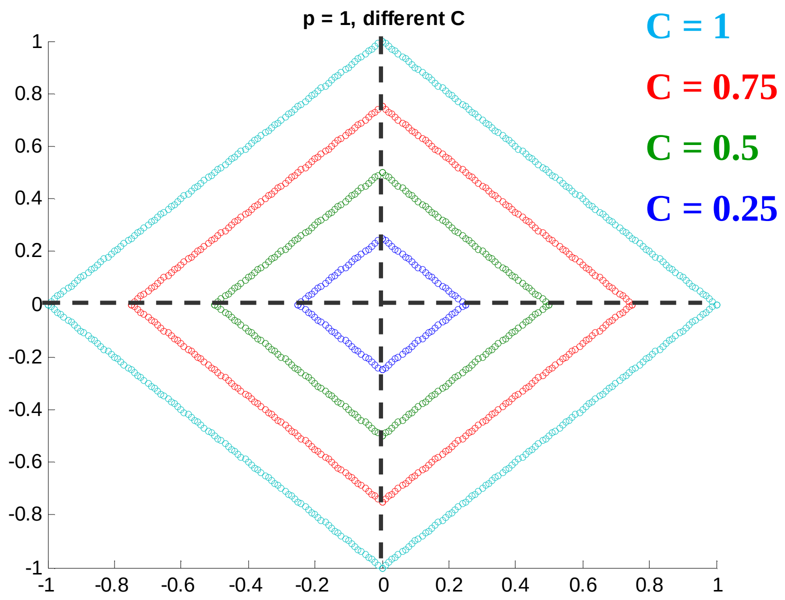
- Ridge.
  - When you have large # of vars and they all have similar importance.

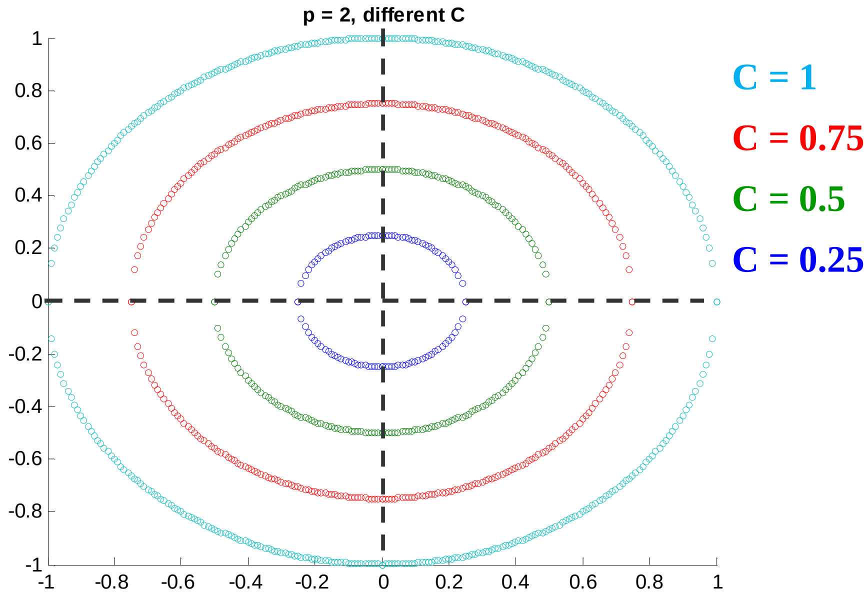
- Lasso.
  - When you have intermediate # of vars and moderate importance.
  - Lasso selects only **one** var from each group of variables.
  - Lasso cannot identify more vars than # of records. (i.e. 500 patients each 10,000 genes)
  - Wide data➔ Data with more predictors than records, resulting redundant predictors in of data.

- Elastic net.
  - Combine lasso and ridge.
  - Outperform lasso on data with highly correlated predictors. (F.Y.I.)

Appendix

Isosurface of L1 and L2





- Isosurface of $\boldsymbol{L_p}$ regularizer = $(\sum_{j=1}^{n}|\theta_j|^p)^{1/p}$ or $\|\theta\|_p$
  - Example, let $\boldsymbol{p}$ = 4 with θ_1, θ_2, so $\boldsymbol{L_4} = \sqrt[4]{\theta_1^4 + \theta_2^4}$

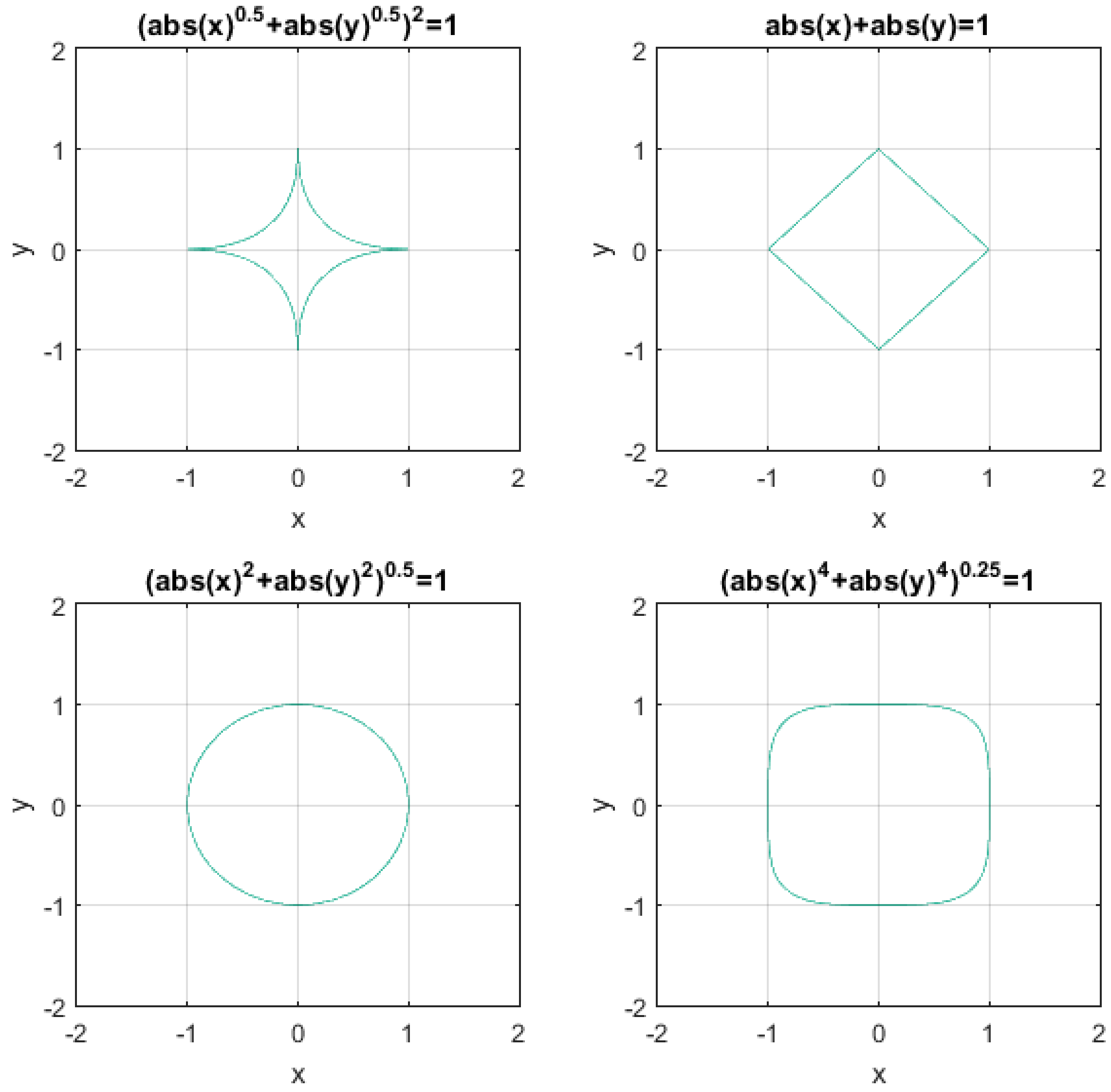
```
for p = [1, 2]
    figure,
    hold on;
    for C = [0.25, 0.5, 0.75, 1];   % multiple C
        t=[];
        % isosurface in one quad
        for t1 = 0: 0.01: C
            t=[t; [t1 (C^p-t1^p)^(1/p)]; ];
        end
        T = t;
        % isoline in all 4 quads
        T = [T; [t(:, 1) * -1, t(:, 2)]];
        T = [T; [t(:, 1), t(:, 2) * -1]];
        T = [T; [t(:, 1) * -1, t(:, 2) * -1]];
        scatter(T(:,1), T(:,2), 5),
    end
    hold off
    title(['\bf p = ' num2str(p) ', different C'])
    xlim([-C(end) C(end)]);
    ylim([-C(end) C(end)])
end
```

Isosurface of L1 to L4 with Constant $C = 1$

- **Simpler Matlab program**
  - $L_2 = \sqrt[2]{\theta_1^2 + \theta_2^2}$

```
syms x y z
figure, subplot(221),


ezplot('(abs(x)^0.5+abs(y)^0.5)^2=1', [-2 2 -2 2])

grid on


subplot(222),
ezplot('abs(x)+abs(y)=1', [-2 2 -2 2])
grid on


subplot(223),
ezplot('(abs(x)^2+abs(y)^2)^0.5=1', [-2 2 -2 2])
grid on


subplot(224),
ezplot('(abs(x)^4+abs(y)^4)^0.25=1', [-2 2 -2 2])
grid on
```

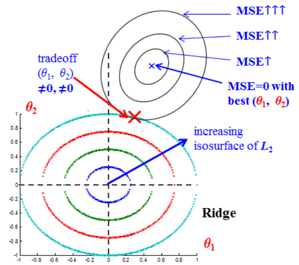
Isosurface of L_p regularizer = $(\sum_{j=1}^{n} |\theta_j|^p)^{1/p}$ or $\|\theta\|_p$



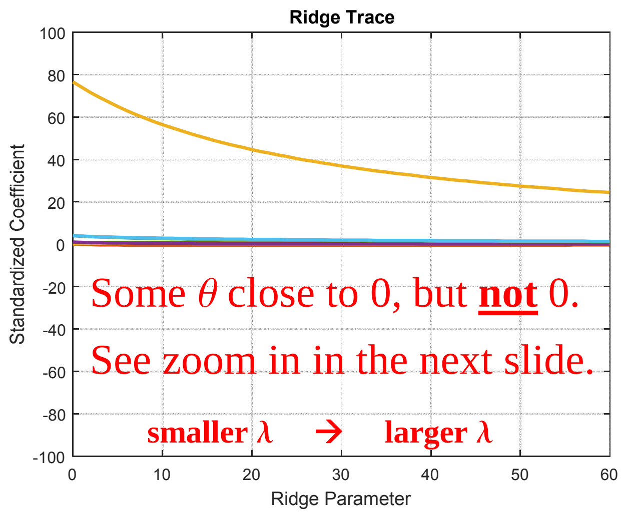
Ridge Regression… :-<<

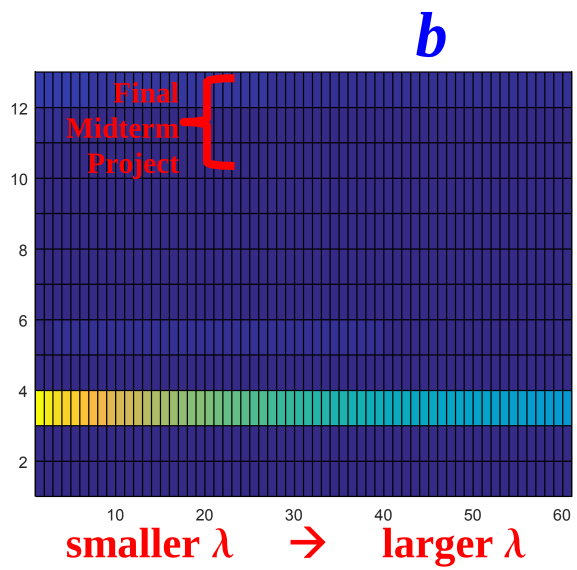
$J(\boldsymbol{\theta}) \approx MSE + \boldsymbol{\lambda\theta}$

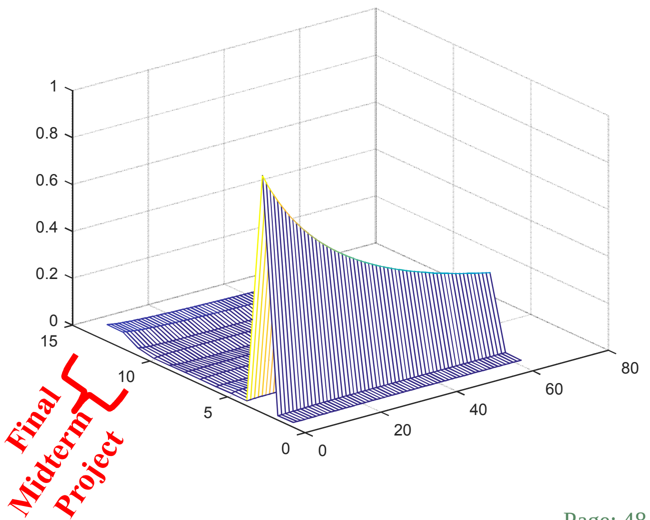
- Ridge is not as effective as Lasso in eliminating coefficients.
  - Some θ close to 0, but not 0.
  - But, I do not expect this weird result in this case.
  - Homework assignment 3 is more important than others…???
  - Matlab ridge() function does not return enough info. You have to build it yourself.
  - Also for ridge in Matlab, **you have to try your own λ range**.




```
k = 0:1:60;      % try your luck here!!
b_ridge = ridge(Y, X, k);
```

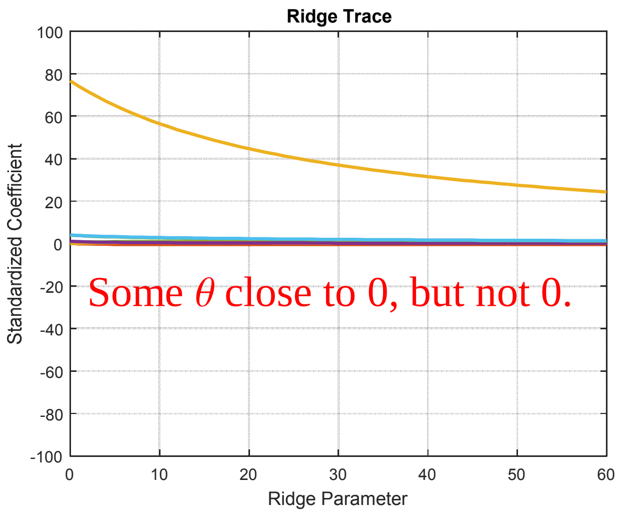


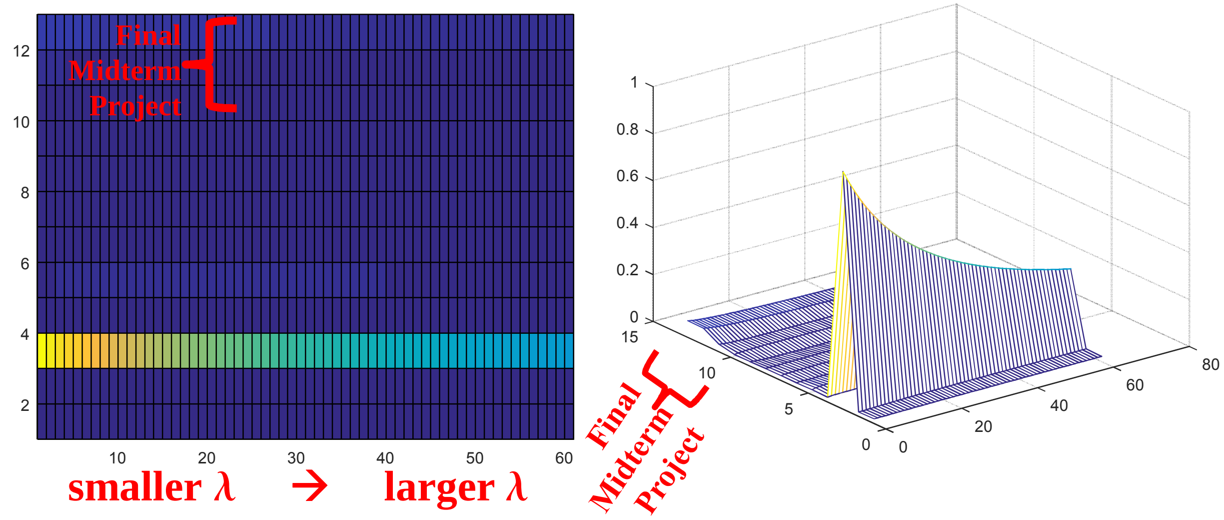


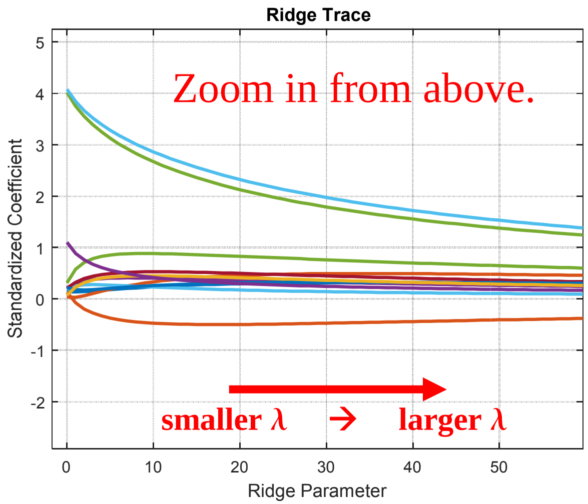


Zoom in to Visualize the **Non-Zero** Coefficients.

- θ may close to 0, but **NOT** 0.
  - Program in the ridge_grade.m file.





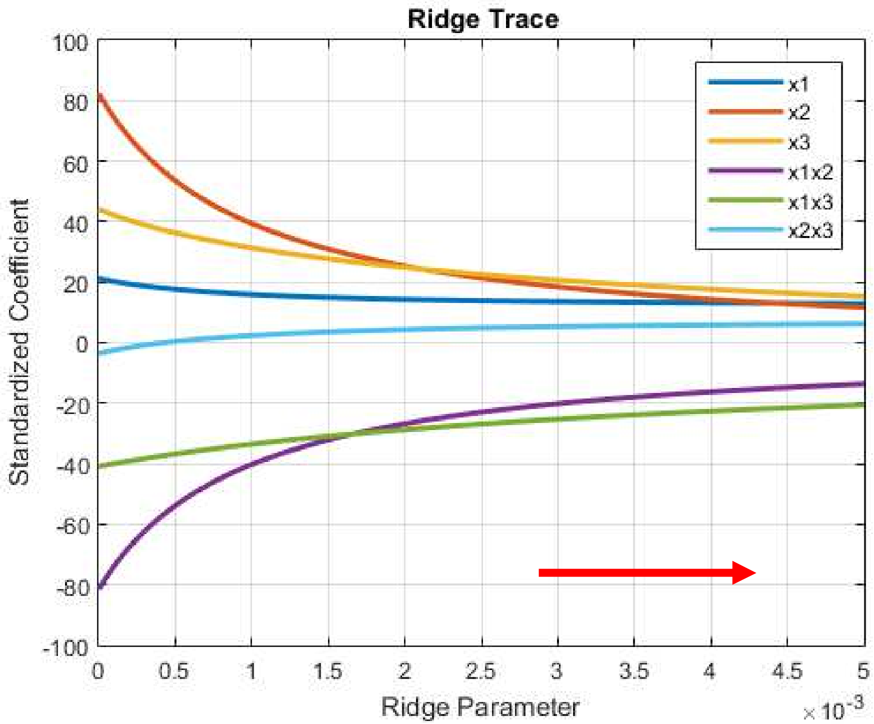


Better Example for Ridge Regression

- Ridge is not as effective as Lasso in eliminating coefficients.
  - Some θ close to 0, but not 0.
  - Example from http://www.mathworks.com/help/stats/ridge.html

```
load acetylene
X = [x1 x2 x3];                  % 3 attributes
D = x2fx(X,'interaction');       % create interaction of attributes
D(:,1) = []; % No constant term
k = 0:1e-5:5e-3;
b = ridge(y,D,k);
```

```
%% Plot Ridge Regression
figure,
plot(k,b,'LineWidth',2)
ylim([-100 100])
grid on
xlabel('Ridge Parameter')
ylabel('Standardized Coefficient')
title('{\bf Ridge Trace}')
legend('x1','x2','x3','x1x2','x1x3','x2x3')
```



Compare to Lasso

- Lasso identifies var 3, 6, 2 are important
  - Yellow, blue, red.

```
load acetylene
X = [x1 x2 x3];
D = x2fx(X,'interaction');
D(:,1) = []; % No constant term
[b fitinfo] = lasso(D, y, 'CV',10, 'Alpha', 1);
lassoPlot(b,fitinfo,'PlotType', 'Lambda', 'XScale', 'log');
```