Master of Space Big Data Analytics
Course MSDZ06

Class 2
Regression Analysis

Min Ding 丁忞
Tuesday, Jan. 26, 2021

Intended Learning Outcomes

At the end of this class, you will be able to understand and apply the following method in Matlab:

1. Ordinary Linear Regression
2. Regularized Linear Regression (ridge & lasso)
3. ~~Principal Component Analysis~~

And become familiarized with:

1. Bias-Variance Tradeoff
2. Constrained regression

Regression Analysis

Data Set

- n cases $i = 1, 2, \ldots, n$
- 1 Response (dependent) variable

$$y_i,\ i = 1, 2, \ldots, n$$

- p Explanatory (independent) variables

$$\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,p})^T,\ i = 1, 2, \ldots, n$$

Goal of Regression Analysis:

- Extract/exploit relationship between y_i and $\mathbf{x}_i$.

Examples

- Prediction
- Causal Inference
- Approximation
- Functional Relationships

Slides from MIT OCW 18.S096, Fall 2013

General Linear Model: For each case i, the conditional distribution $[y_i \mid x_i]$ is given by

$$y_i = \hat{y}_i + \epsilon_i$$

where

- $\hat{y}_i = \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_{i,p} x_{i,p}$
- $\beta = (\beta_1, \beta_2, \ldots, \beta_p)^T$ are p regression parameters (constant over all cases)
- ϵ_i Residual (error) variable (varies over all cases)

Extensive breadth of possible models

- Polynomial approximation ($x_{i,j} = (x_i)^j$, explanatory variables are different powers of the same variable $x = x_i$)
- Fourier Series: ($x_{i,j} = sin(jx_i)$ or $cos(jx_i)$, explanatory variables are different sin/cos terms of a Fourier series expansion)
- Time series regressions: time indexed by i, and explanatory variables include lagged response values.

Note: *Linearity* of $\hat{y}_i$ (in regression parameters) maintained with non-linear x.

Steps for Fitting a Model

(1) Propose a model in terms of
   - Response variable Y (specify the scale)
   - Explanatory variables $X_1, X_2, \ldots X_p$ (include different functions of explanatory variables if appropriate)
   - Assumptions about the distribution of ϵ over the cases

(2) Specify/define a criterion for judging different estimators.

(3) Characterize the best estimator and apply it to the given data.

(4) Check the assumptions in (1).

(5) If necessary modify model and/or assumptions and go to (1).

Judging Criteria (R^2 and SSE)

Coefficient of Determination (R-Squared)

Purpose

Coefficient of determination (R-squared) indicates the proportionate amount of variation in the response variable *y* explained by the independent variables *X* in the linear regression model. The larger the R-squared is, the more variability is explained by the linear regression model.

Definition

R-squared is the proportion of the total sum of squares explained by the model. `Rsquared`, a property of the fitted model, is a structure with two fields:

- `Ordinary` — Ordinary (unadjusted) R-squared

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}.$$

- `Adjusted` — R-squared adjusted for the number of coefficients

$$R^2_{adj} = 1 - \left(\frac{n-1}{n-p}\right)\frac{SSE}{SST}.$$

SSE is the sum of squared error, *SSR* is the sum of squared regression, *SST* is the sum of squared total, *n* is the number of observations, and *p* is the number of regression coefficients (including the intercept). Because R-squared increases with added predictor variables in the regression model, the adjusted R-squared adjusts for the number of predictor variables in the model. This makes it more useful for comparing models with a different number of predictors.

From Matlab Documentation

Judging Criteria

Information Criteria

Model comparison tests—such as the likelihood ratio, Lagrange multiplier, or Wald test—are only appropriate for comparing nested models. In contrast, information criteria are model selection tools that you can use to compare any models fit to the same data. That is, the models being compared do not need to be nested.

Basically, information criteria are likelihood-based measures of model fit that include a penalty for complexity (specifically, the number of parameters). Different information criteria are distinguished by the form of the penalty, and can prefer different models.

Let $\log L(\hat{\theta})$ denote the value of the maximized loglikelihood objective function for a model with *k* parameters fit to *N* data points. Two commonly used information criteria are:

- **Akaike information criterion (AIC).** The AIC compares models from the perspective of information entropy, as measured by Kullback-Leibler divergence. The AIC for a given model is

  $$-2 \log L(\hat{\theta}) + 2k.$$

  When comparing AIC values for multiple models, smaller values of the criterion are better.
- **Bayesian information criterion (BIC)**. The BIC, also known as Schwarz information criterion, compares models from the perspective of decision theory, as measured by expected loss. The BIC for a given model is

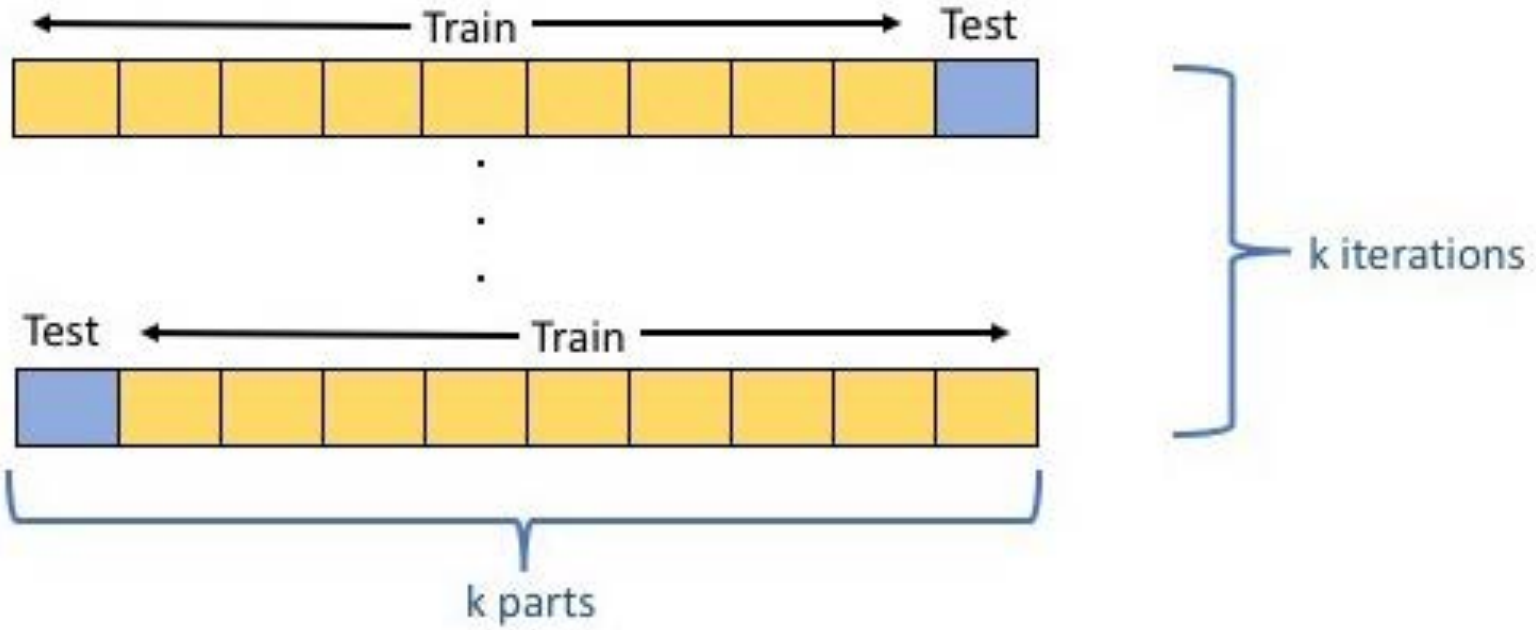
  $$-2 \log L(\hat{\theta}) + k \log(N).$$

  When comparing BIC values for multiple models, smaller values of the criterion are better.

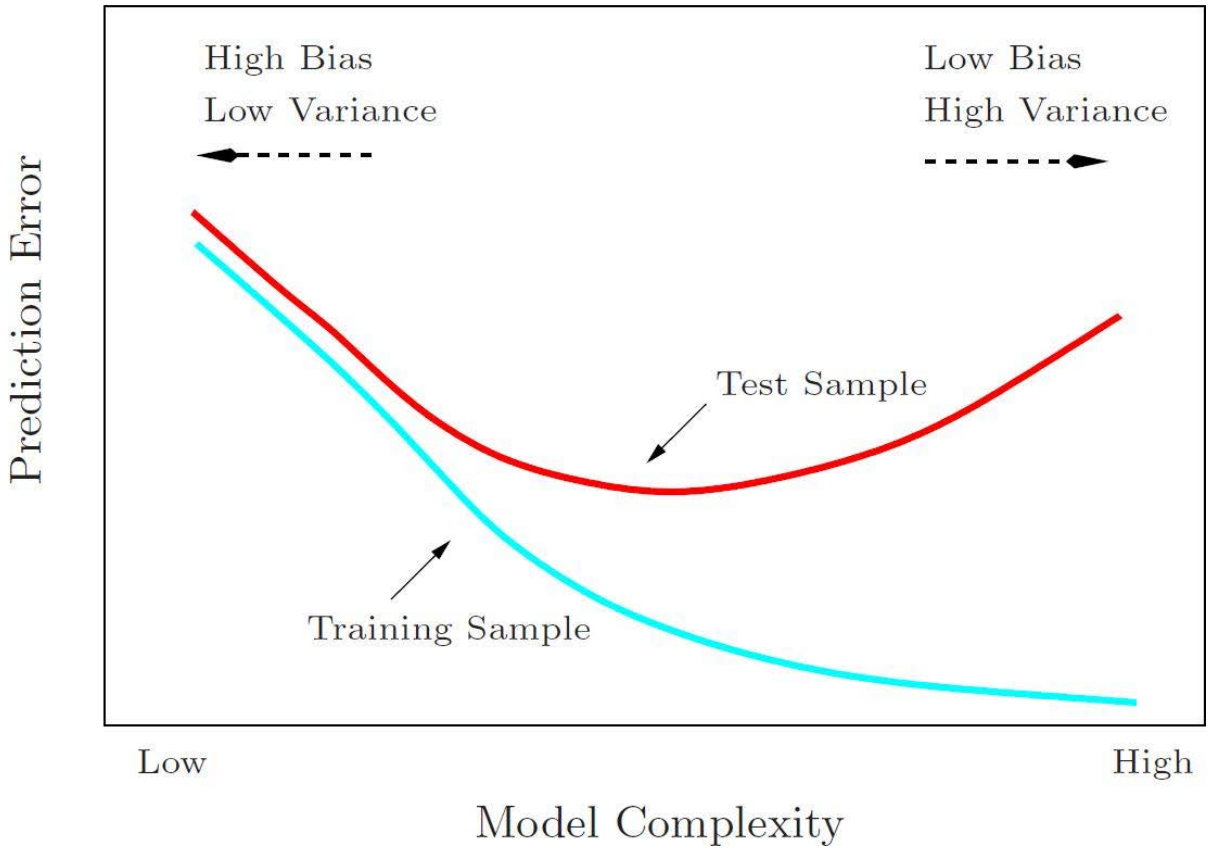
Judging Criteria

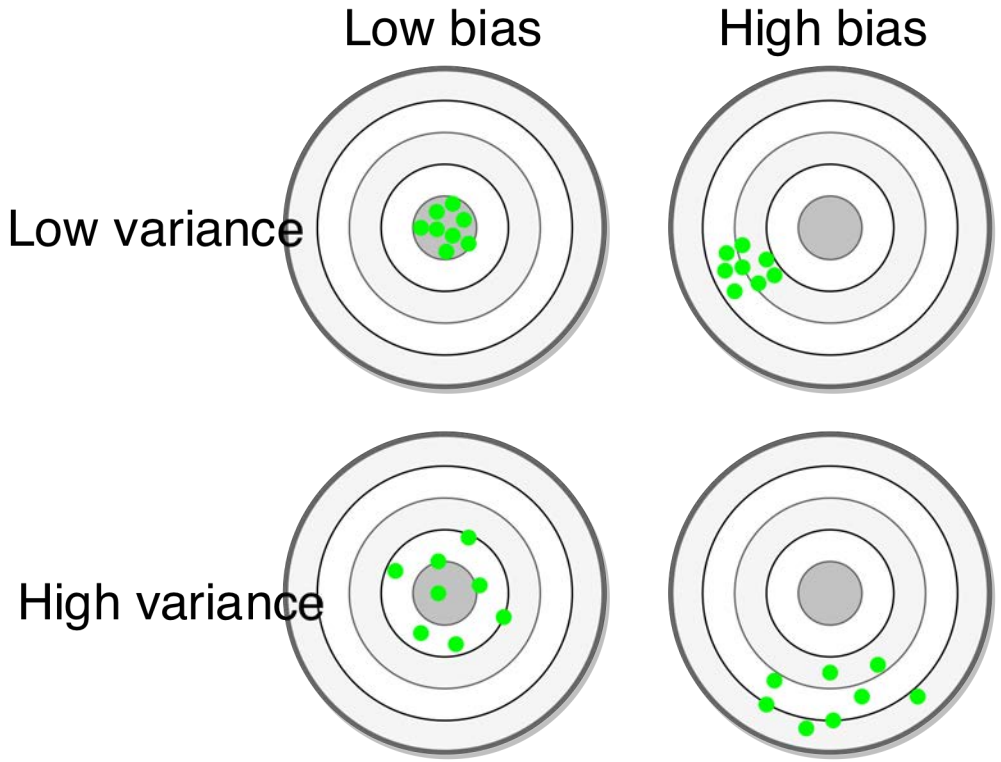
K Folds Cross Validation Method

1. Divide the sample data into k parts.
2. Use k-1 of the parts for training, and 1 for testing.
3. Repeat the procedure k times, rotating the test set.
4. Determine an expected performance metric (mean square error, misclassification error rate, confidence interval, or other appropriate metric) based on the results across the iterations



Bias-Variance Tradeoff





Ordinary Least Squares Estimates

Least Squares Criterion: For $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^T$, define

$$\begin{aligned} Q(\boldsymbol{\beta}) &= \sum_{i=1}^{N} [y_i - \hat{y}_i]^2 \\ &= \sum_{i=1}^{N} [y_i - (\beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_{i,p} x_{i,p})]^2 \end{aligned}$$

Ordinary Least-Squares (OLS) estimate $\hat{\boldsymbol{\beta}}$: minimizes $Q(\boldsymbol{\beta})$.

Matrix Notation

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad \mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{p,n} \end{bmatrix} \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

Solving for OLS Estimate $\hat{\beta}$

$$\hat{\mathbf{y}} = \begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{pmatrix} = \mathbf{X}\beta \text{ and}$$

$$\begin{aligned} Q(\beta) &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = (\mathbf{y} - \hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}}) \\ &= (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) \end{aligned}$$

OLS $\hat{\beta}$ solves $\frac{\partial Q(\beta)}{\partial \beta_j} = 0, \quad j = 1, 2, \ldots, p$

$$\begin{aligned} \frac{\partial Q(\beta)}{\partial \beta_j} &= \frac{\partial}{\partial \beta_j}\left(\sum_{i=1}^{n}[y_i - (x_{i,1}\beta_1 + x_{i,2}\beta_2 + \cdots x_{i,p}\beta_p)]^2\right) \\ &= \sum_{i=1}^{n} 2(-x_{i,j})[y_i - (x_{i,1}\beta_1 + x_{i,2}\beta_2 + \cdots x_{i,p}\beta_p)] \\ &= -2(\mathbf{X}_{[j]})^T(\mathbf{y} - \mathbf{X}\beta) \end{aligned}$$

where $\mathbf{X}_{[j]}$ is the jth column of $\mathbf{X}$

Solving for OLS Estimate $\hat{\boldsymbol{\beta}}$

$$\frac{\partial Q}{\partial \boldsymbol{\beta}} = \begin{bmatrix} \frac{\partial Q}{\partial \beta_1} \\ \frac{\partial Q}{\partial \beta_2} \\ \vdots \\ \frac{\partial Q}{\partial \beta_p} \end{bmatrix} = -2 \begin{bmatrix} \mathbf{X}_{[1]}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ \mathbf{X}_{[2]}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ \vdots \\ \mathbf{X}_{[p]}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \end{bmatrix} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

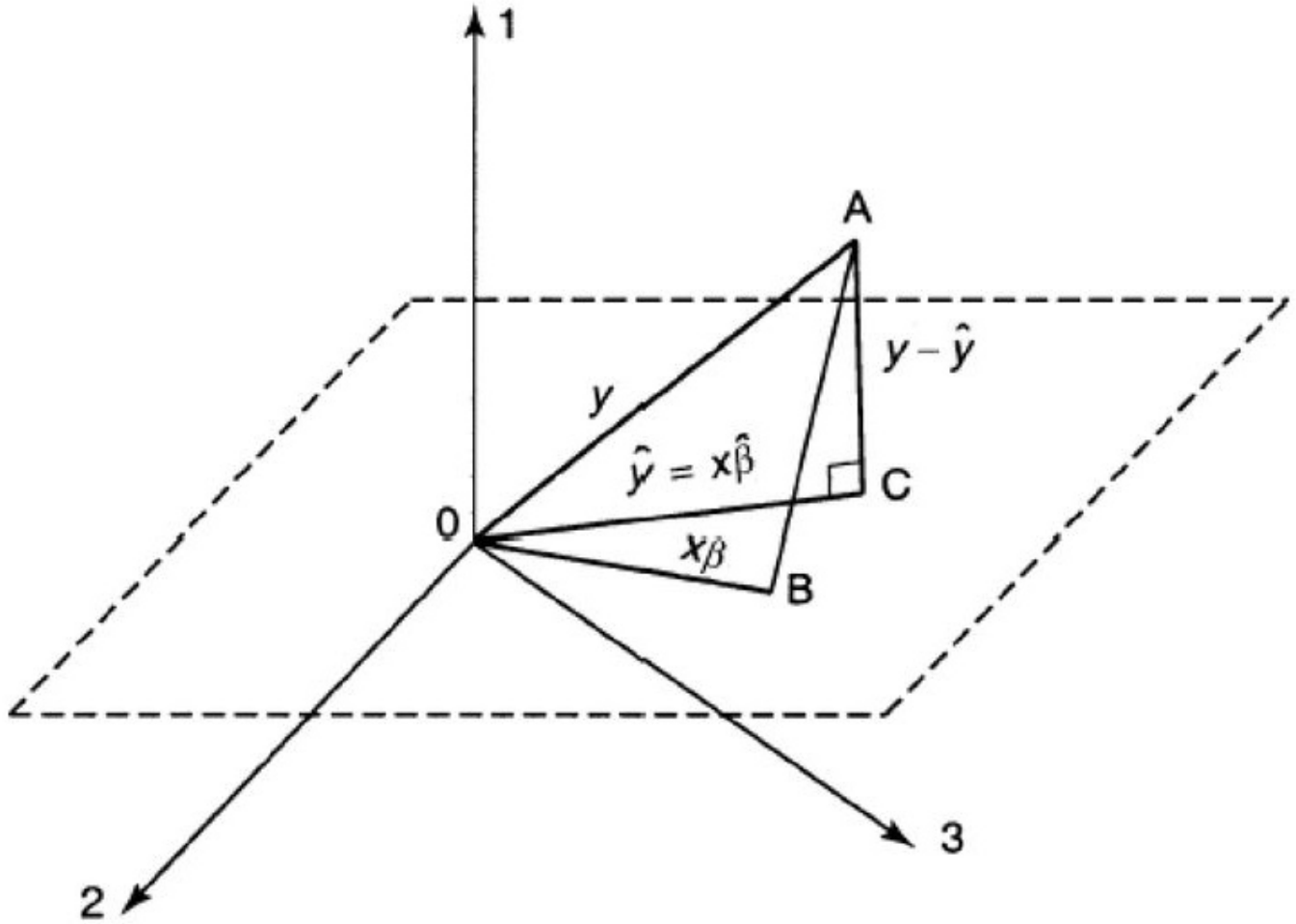
So the OLS Estimate $\hat{\boldsymbol{\beta}}$ solves the **"Normal Equations"**

$$\begin{aligned} & \mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = \mathbf{0} \\ \iff \quad & \mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{y} \\ \implies \quad & \hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \end{aligned}$$

N.B. For $\hat{\boldsymbol{\beta}}$ to exist (uniquely)

$$\begin{aligned} & (\mathbf{X}^T\mathbf{X}) \text{ must be invertible} \\ \iff \quad & \mathbf{X} \text{ must have Full Column Rank} \end{aligned}$$

Geometric Representation of OLS Estimates



Linear Regression Analysis 5E
Montgomery, Peck & Vining

Maximum-Likelihood Estimation

Consider the normal linear regression model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \text{ where } \{\epsilon_i\} \text{ are i.i.d. } N(0, \sigma^2), \text{ i.e.,}$$
$$\boldsymbol{\epsilon} \sim N_n(\mathbf{0}_n, \sigma^2\mathbf{I}_n)$$
$$\text{or } \mathbf{y} \sim N_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n)$$

Definitions:

- The **likelihood function** is
$$L(\boldsymbol{\beta}, \sigma^2) = p(\mathbf{y} \mid \mathbf{X}, \mathbf{B}, \sigma^2)$$
where $p(\mathbf{y} \mid \mathbf{X}, \mathbf{B}, \sigma^2)$ is the joint probability density function (pdf) of the conditional distribution of $\mathbf{y}$ given data $\mathbf{X}$, (known) and parameters $(\boldsymbol{\beta}, \sigma^2)$ (unknown).
- The **maximum likelihood** estimates of $(\boldsymbol{\beta}, \sigma^2)$ are the values maximizing $L(\boldsymbol{\beta}, \sigma^2)$, i.e., those which make the observed data $\mathbf{y}$ most likely in terms of its pdf.

Because the y_i are independent r.v.'s with $y_i \sim N(\mu_i, \sigma^2)$ where $\mu_i = \sum_{j=1}^{p} \beta_j x_{i,j}$,

$$\begin{aligned} L(\boldsymbol{\beta}, \sigma^2) &= \prod_{i=1}^{n} p(y_i \mid \beta, \sigma^2) \\ &= \prod_{i=1}^{n} \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_i - \sum_{j=1} \beta_j x_{i,j})^2} \right] \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T (\sigma^2 \mathbf{I}_n)^{-1} (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})} \end{aligned}$$

The maximum likelihood estimates $(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2)$ maximize the log-likeliood function (dropping constant terms)

$$\begin{aligned} logL(\boldsymbol{\beta}, \sigma^2) &= -\frac{n}{2} log(\sigma^2) - \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\sigma^2 \mathbf{I}_n)^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ &= -\frac{n}{2} log(\sigma^2) - \frac{1}{2\sigma^2} Q(\boldsymbol{\beta}) \end{aligned}$$

where $Q(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ ("Least-Squares Criterion" !)

- The OLS estimate $\hat{\boldsymbol{\beta}}$ is also the ML-estimate.
- The ML estimate of σ^2 solves

$$\frac{\partial log\ L(\hat{\boldsymbol{\beta}}, \sigma^2)}{\partial(\sigma^2)} = 0 \text{ ,i.e., } -\frac{n}{2}\frac{1}{\sigma^2} - \frac{1}{2}(-1)(\sigma^2)^{-2} Q(\hat{\boldsymbol{\beta}}) = 0$$

$$\Longrightarrow \hat{\sigma^2}_{ML} = Q(\hat{\boldsymbol{\beta}})/n = (\textstyle\sum_{i=1}^{n} \hat{\epsilon}_i^2)/n \quad (biased!)$$

Polynomial Regression

Matrix form and calculation of estimates [edit]

The polynomial regression model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_m x_i^m + \varepsilon_i \ (i = 1, 2, \ldots, n)$$

can be expressed in matrix form in terms of a design matrix $\mathbf{X}$, a response vector $\vec{y}$, a parameter vector $\vec{\beta}$, and a vector $\vec{\varepsilon}$ of random errors. The *i*-th row of $\mathbf{X}$ and $\vec{y}$ will contain the *x* and *y* value for the *i*-th data sample. Then the model can be written as a system of linear equations:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ 1 & x_3 & x_3^2 & \cdots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix},$$

which when using pure matrix notation is written as

$$\vec{y} = \mathbf{X}\vec{\beta} + \vec{\varepsilon}.$$

The vector of estimated polynomial regression coefficients (using ordinary least squares estimation) is

$$\hat{\vec{\beta}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\,\mathbf{X}^\mathsf{T}\vec{y},$$

assuming $m < n$ which is required for the matrix to be invertible; then since $\mathbf{X}$ is a Vandermonde matrix, the invertibility condition is guaranteed to hold if all the x_i values are distinct. This is the unique least-squares solution.

From Wikipedia

Multiple Regression

Suppose an experiment involves two independent variables—say, u and v—and one dependent variable, y. A simple equation for predicting y from u and v has the form

$$y = \beta_0 + \beta_1 u + \beta_2 v \tag{4}$$

A more general prediction equation might have the form

$$y = \beta_0 + \beta_1 u + \beta_2 v + \beta_3 u^2 + \beta_4 uv + \beta_5 v^2 \tag{5}$$

This equation is used in geology, for instance, to model erosion surfaces, glacial cirques, soil pH, and other quantities. In such cases, the least-squares fit is called a *trend surface*.

Equations (4) and (5) both lead to a linear model because they are linear in the unknown parameters (even though u and v are multiplied). In general, a linear model will arise whenever y is to be predicted by an equation of the form

$$y = \beta_0 f_0(u, v) + \beta_1 f_1(u, v) + \cdots + \beta_k f_k(u, v)$$

with $f_0, \ldots, f_k$ any sort of known functions and $\beta_0, \ldots, \beta_k$ unknown weights.

Weighted Least Squares Regression

- In some cases the observations may be weighted—for example, they may not be equally reliable.
- In this case, one can minimize the weighted sum of squares:

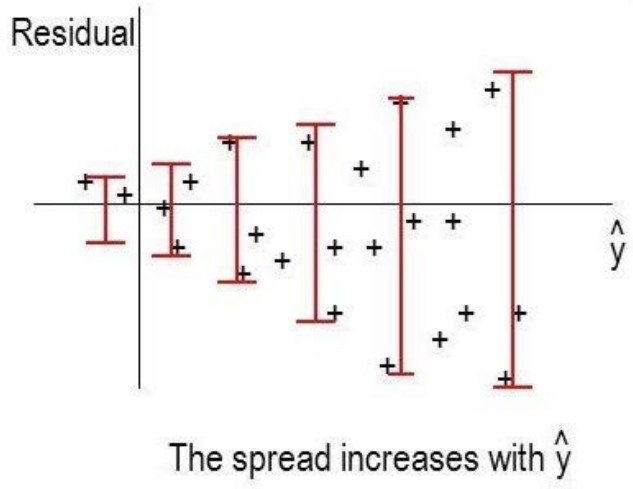
$$\arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{m} w_i \left| y_i - \sum_{j=1}^{n} X_{ij}\beta_j \right|^2 = \arg\min_{\boldsymbol{\beta}} \left\| W^{1/2}(\mathbf{y} - X\boldsymbol{\beta}) \right\|^2.$$

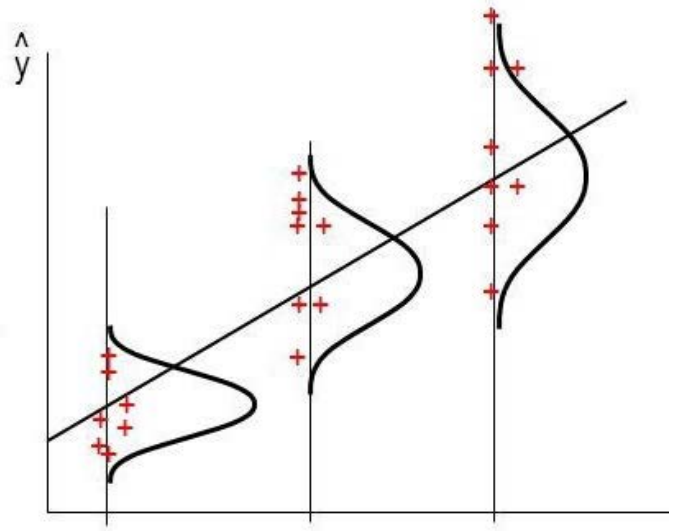
- Now view Wy as y, and WX as X in the ordinary least squares problem.
- Suppose $\hat{\beta}$ satisfies $\mathbf{WX}\,\hat{\boldsymbol{\beta}} = \mathbf{Wy}$
- The **normal equation** for the ordinary least squares problem is:

$$(\mathbf{WX})^{\mathsf{T}}\,\mathbf{WX}\,\hat{\boldsymbol{\beta}} = (\mathbf{WX})^{\mathsf{T}}\,\mathbf{Wy}$$

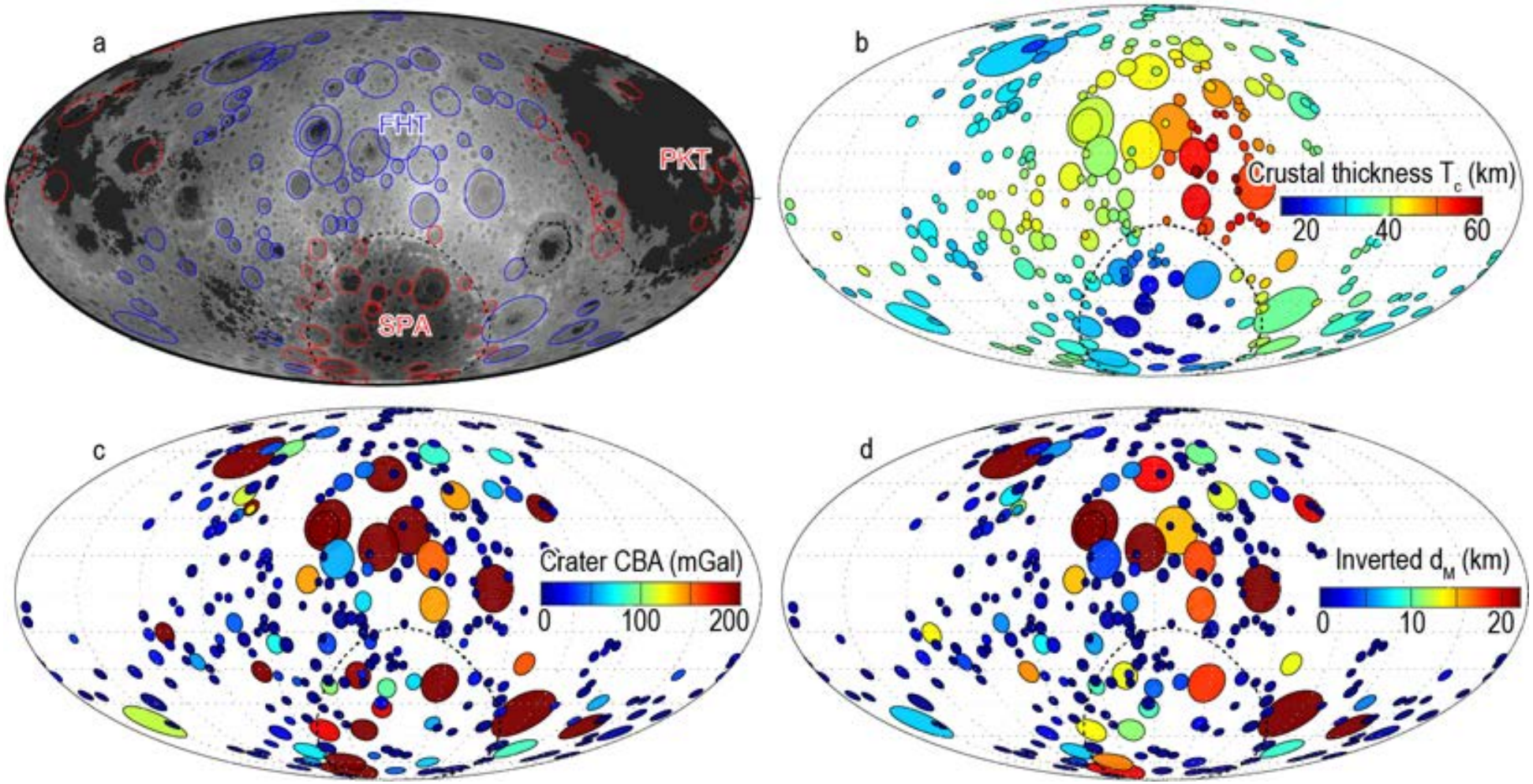
- Solving this matrix equation provides $\hat{\beta}$

i.e., Heteroscedas'ticity – 异方差性





Example 1: Data



From Ding+JGR, 2021, data available in https://zenodo.org/record/3833814

Multivariate Linear Regression Results

```
X = [Dc Tc Rho0 Phic Pmare Hmare];
Y = CBA;
```

Linear regression model:

y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	-214.03	382.67	-0.55929	**0.57636**
x1	0.58473	0.021412	27.308	2.8652e-85
x2	0.35324	0.32694	1.0805	**0.28076**
x3	0.040008	0.1284	0.3116	**0.75555**
x4	128.57	74.963	1.7151	0.087313
x5	4.9339	10.241	0.48176	**0.63031**
x6	-5.0967	2.1785	-2.3396	0.019927

Number of observations: 324, Error degrees of freedom: 317

Root Mean Squared Error: 32.7

R-squared: 0.751, Adjusted R-Squared 0.747

F-statistic vs. constant model: 160, p-value = 1.06e-92

- ***High p-values indicate some of these predictors are unnecessary.***

Interpretation: T-statstic & F-statistic

Inferences About the Slope: t Test

- t test for a population slope
  - Is there a linear relationship between X and Y?
- Null and alternative hypotheses
  - H_0: $\beta_1 = 0$ (no linear relationship)
  - H_1: $\beta_1 \neq 0$ (linear relationship does exist)
- Test statistic

$$t = \frac{b_1 - \beta_1}{S_{b_1}}$$

$$d.f. = n - 2$$

where:

b_1 = regression slope coefficient

β_1 = hypothesized slope

S_{b1} = standard error of the slope

Interpretation: F-statistic and ANOVA

$$F = \frac{\text{variance between treatments}}{\text{variance within treatments}}$$

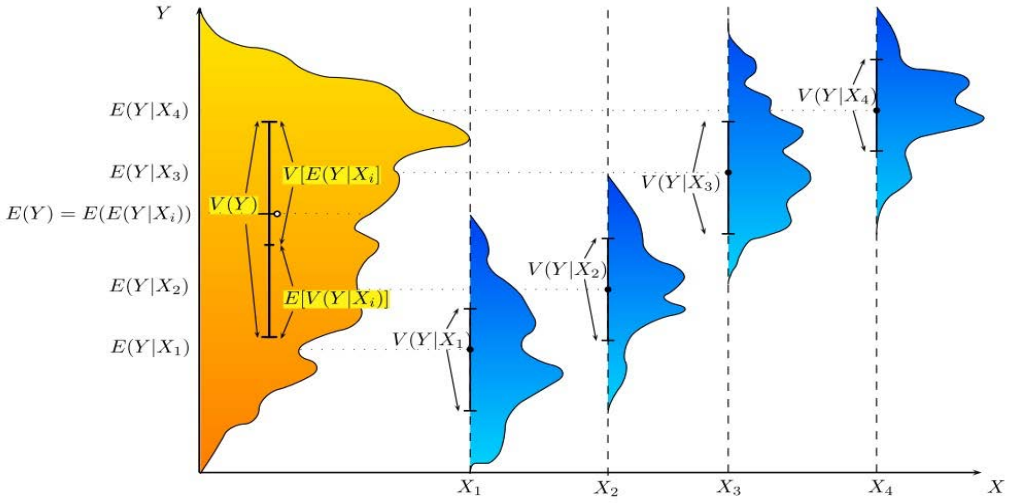


Figure 1: ANOVA : Fair fit

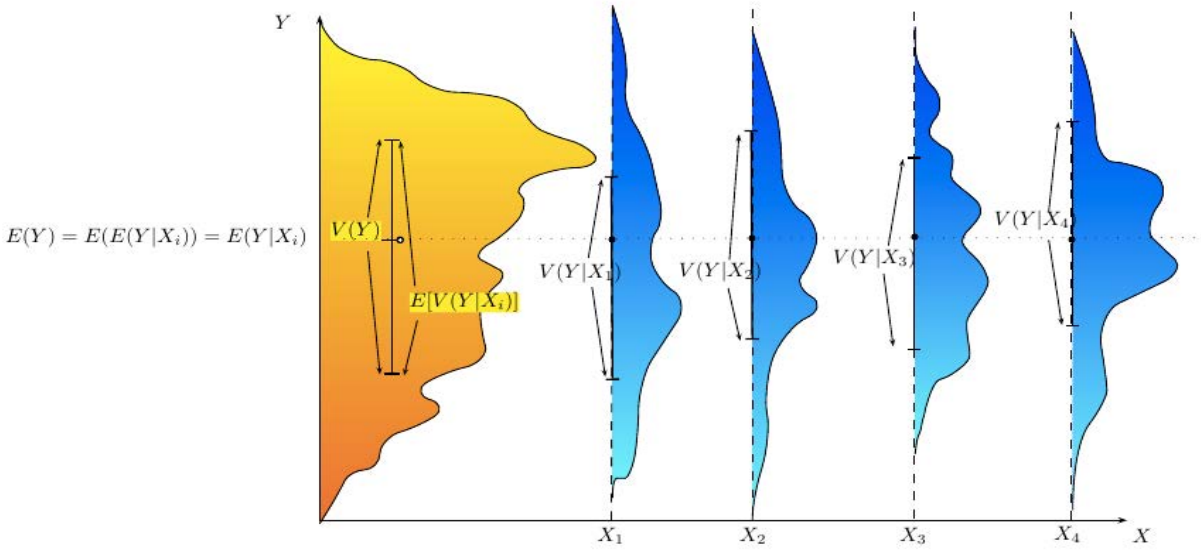


Figure 2: ANOVA : No fit

Best Regression Model After Outlier Removal and Subset Selection

1. Adding x1, FStat = 929.2868, pValue = 6.34095e-97
2. Adding x6, FStat = 6.3488, pValue = 0.012231

Linear regression model:

y ~ 1 + x1 + x6

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	-68.447	4.4214	-15.481	9.1031e-41
x1	0.58434	0.020919	27.934	6.1725e-88
x6	-4.9961	1.9828	-2.5197	0.012231

Number of observations: 324, Error degrees of freedom: 321

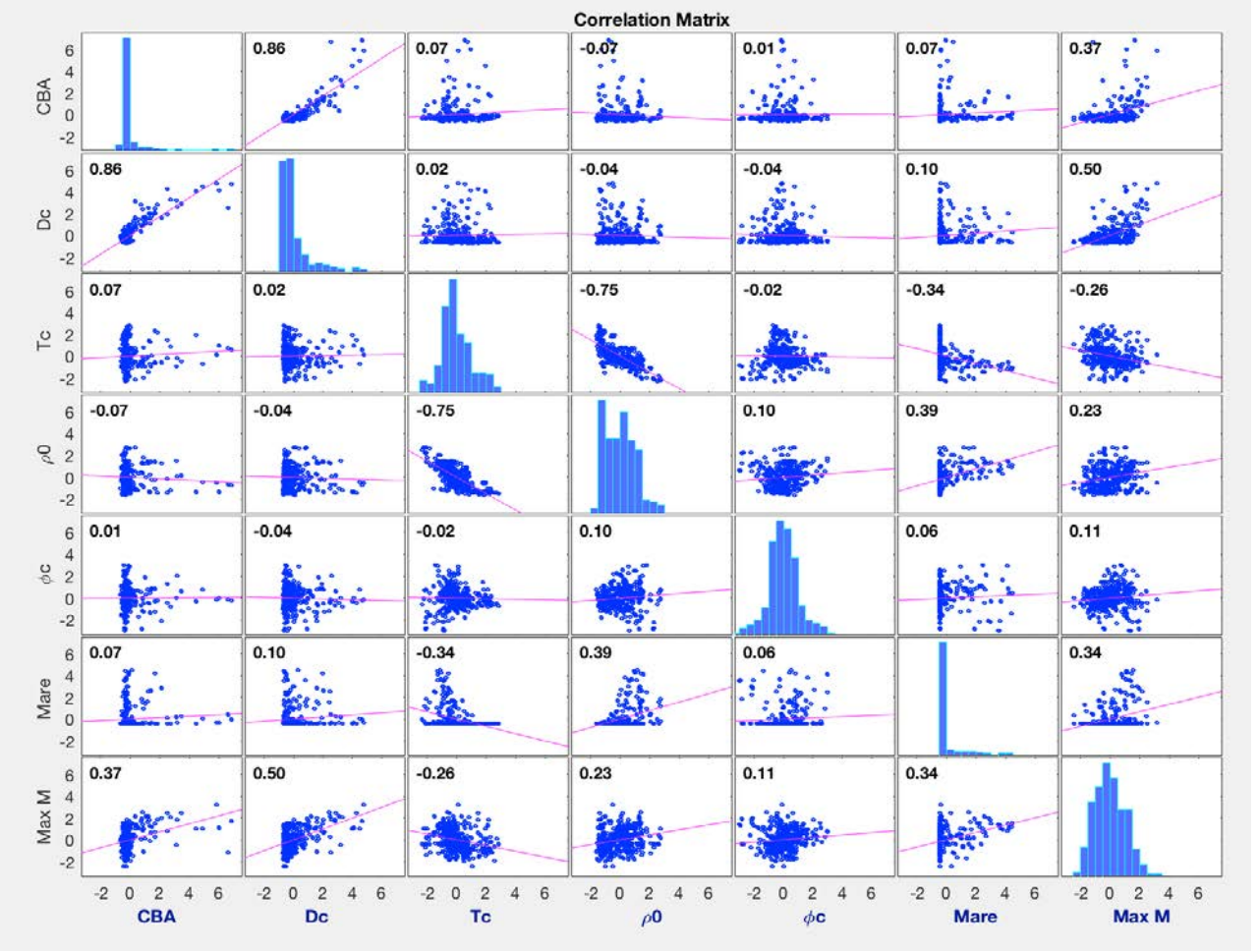
Root Mean Squared Error: 32.8

R-squared: 0.748, Adjusted R-Squared 0.746

F-statistic vs. constant model: 476, p-value = 1.05e-96

- *The adjusted R^2 penalizes you for adding independent variables that do not fit the model.*
- ***Results exclude x2 (Tc) as an effective predictor for explaining y (CBA).***
- ***But this x2 is the variable of interest.***

Collinearity Between Predictors



- Explain why the previous two models fit the data similarly and why x2 is not Included.
- One possible way to solve this “overfitting” is to try regularized linear regression.

Regularization

- Can modify our cost function J to add "preference" for certain parameter values

$$J(\underline{\theta}) = \frac{1}{2}(\underline{y} - \underline{\theta}\,\underline{X}^T) \cdot (\underline{y} - \underline{\theta}\,\underline{X}^T)^T + \alpha\,\theta\theta^T$$

L_2 penalty:
"Ridge regression"

- New solution (derive the same way)

$$\underline{\theta} \quad = \quad \underline{y}\,\underline{X}(\underline{X}^T\,\underline{X} + \alpha I)^{-1}$$

  - Problem is now well-posed for any degree

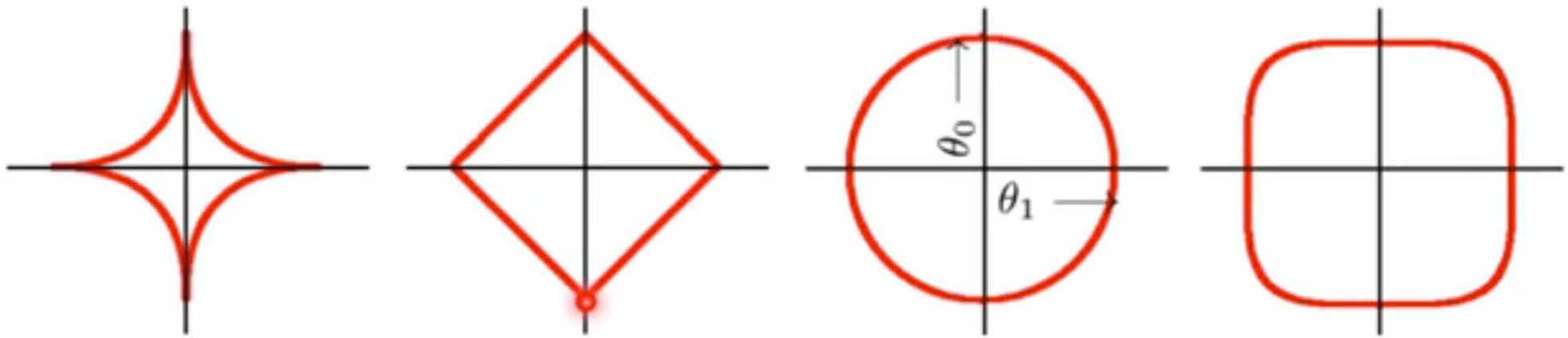
- Notes:
  - "Shrinks" the parameters toward zero
  - Alpha large: we prefer small theta to small MSE
  - Regularization term is independent of the data: paying more attention reduces our variance

https://www.youtube.com/watch?v=sO4ZirJh9ds

Different regularization functions

- More generally, for the L_p regularizer: $\left(\sum_i |\theta_i|^p \right)^{\frac{1}{p}}$

Isosurfaces: $\|\theta\|_p = \text{constant}$

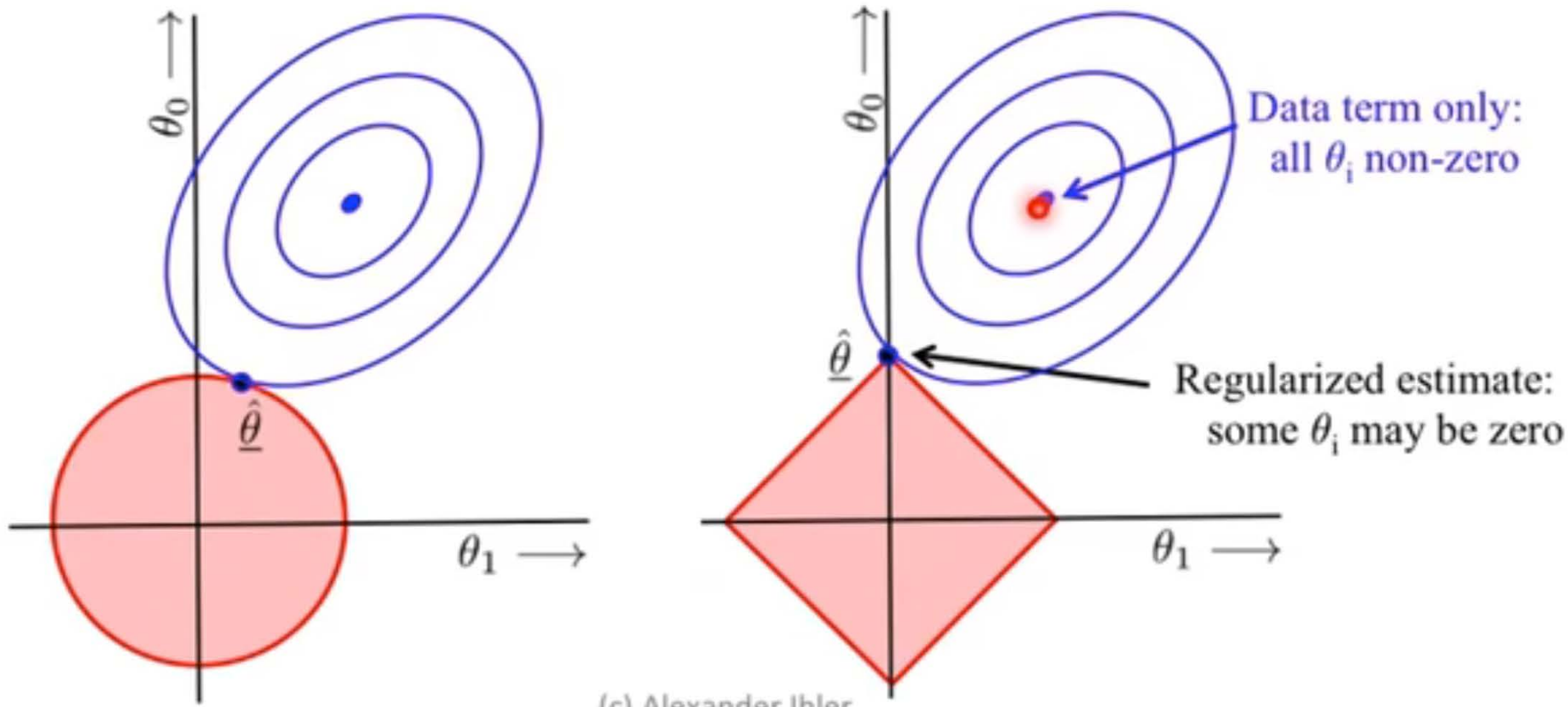


p=0.5 | p=1 Lasso | p=2 Quadratic | p=4

L_0 = limit as $p \to 0$: "number of nonzero weights", a natural notion of complexity

Regularization: L1 vs L2

- Estimate balances data term & regularization term
- Lasso tends to generate sparser solutions than a quadratic regularizer.



Regularization can be used for feature selection

There are other approaches to feature selection:

- **Stepwise regression** sequentially adds or removes features until there is no improvement in prediction; used with linear regression or generalized linear regression algorithms. Similarly, sequential feature selection for any supervised learning sequentially builds up a feature set algorithm until accuracy (or a custom performance measure) stop improving.
- **Automated feature selection** such as neighborhood component analysis (NCA) identifies a subset of features that maximize classification performance based on their predictive power.
- **Boosted and bagged decision trees** are ensemble methods that compute variable importance from out-of-bag estimates.
- **Regularization (lasso and elastic nets)** is a shrinkage estimator used to remove redundant features by reducing their weights (coefficients) to zero.

Comparison of Feature Selection Methods

Best Subset, Forward Stepwise, or Lasso?
Analysis and Recommendations Based on Extensive Comparisons

Trevor Hastie Robert Tibshirani Ryan J. Tibshirani

Abstract

In exciting new work, Bertsimas et al. (2016) showed that the classical best subset selection problem in regression modeling can be formulated as a mixed integer optimization (MIO) problem. Using recent advances in MIO algorithms, they demonstrated that best subset selection can now be solved at much larger problem sizes that what was thought possible in the statistics community. They presented empirical comparisons of best subset selection with other popular variable selection procedures, in particular, the lasso and forward stepwise selection. Surprisingly (to us), their simulations suggested that best subset selection consistently outperformed both methods in terms of prediction accuracy. Here we present an expanded set of simulations to shed more light on these comparisons. The summary is roughly as follows:

- neither best subset selection nor the lasso uniformly dominate the other, with best subset selection generally performing better in high signal-to-noise (SNR) ratio regimes, and the lasso better in low SNR regimes;
- best subset selection and forward stepwise perform quite similarly throughout;
- the relaxed lasso (actually, a simplified version of the original relaxed estimator defined in Meinshausen, 2007) is the overall winner, performing just about as well as the lasso in low SNR scenarios, and as well as best subset selection in high SNR scenarios.

Relaxed Lasso

Definition 1. The relaxed Lasso estimator is defined for $\lambda \in [0, \infty)$ and $\phi \in (0, 1]$ as

$$\hat{\beta}^{\lambda,\phi} = \arg\min_{\beta} n^{-1} \sum_{i=1}^{n} (Y_i - X_i^{\mathrm{T}}\{\beta \cdot \mathbf{1}_{\mathcal{M}_\lambda}\})^2 + \phi\lambda\|\beta\|_1, \tag{4}$$

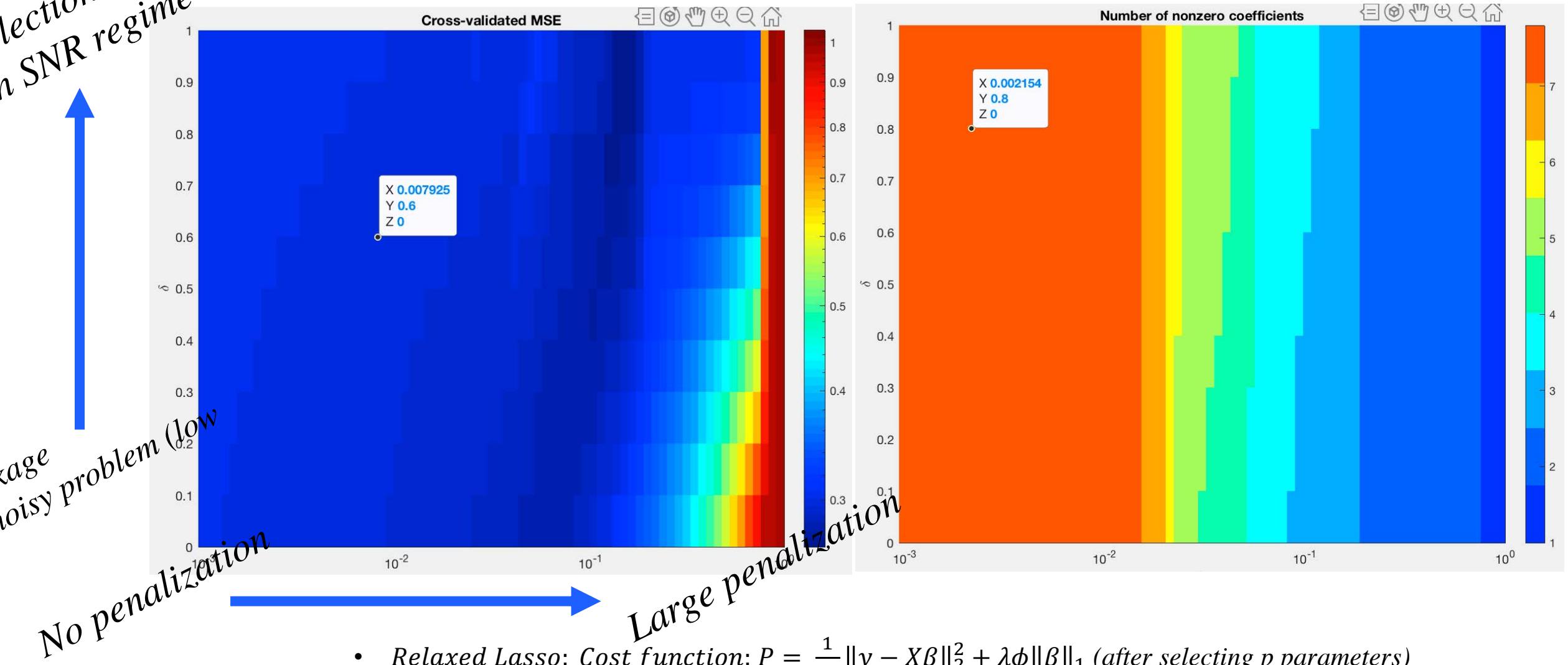
where $\mathbf{1}_{\mathcal{M}_\lambda}$ is the indicator function on the set of variables $\mathcal{M}_\lambda \subseteq \{1, \ldots, p\}$ so that for all $k \in \{1, \ldots, p\}$,

$$\{\beta \cdot \mathbf{1}_{\mathcal{M}_\lambda}\}_k = \begin{cases} 0, & k \notin \mathcal{M}_\lambda, \\ \beta_k, & k \in \mathcal{M}_\lambda. \end{cases}$$

Note that only predictor variables in the set $\mathcal{M}_\lambda$ are considered for the relaxed Lasso estimator. The parameter λ controls thus the variable selection part, as in ordinary Lasso estimation. The relaxation parameter ϕ controls on the other hand the shrinkage of coefficients. If $\phi = 1$, the Lasso and relaxed Lasso estimators are identical. For $\phi < 1$, the shrinkage of coefficients in the selected model is reduced compared to ordinary Lasso estimation. The case of $\phi = 0$ needs special consideration, as the definition above would produce a degenerate solution. In the following, we define the relaxed Lasso estimator for $\phi = 0$ as the limit of the above definition for $\phi \to 0$. In this case, all coefficients in the model $\mathcal{M}_\lambda$ are estimated by the OLS-solution. This estimator (for $\phi = 0$) was already proposed in Efron et al. (2004) as Lars–OLS hybrid, "using Lars to find the model but not to estimate the coefficients" (Efron et al., 2004). The reduction of the sum of squared residuals of this hybrid method over the ordinary Lasso estimator was found to be small for the studied data set, which contained 10 predictor variables only.

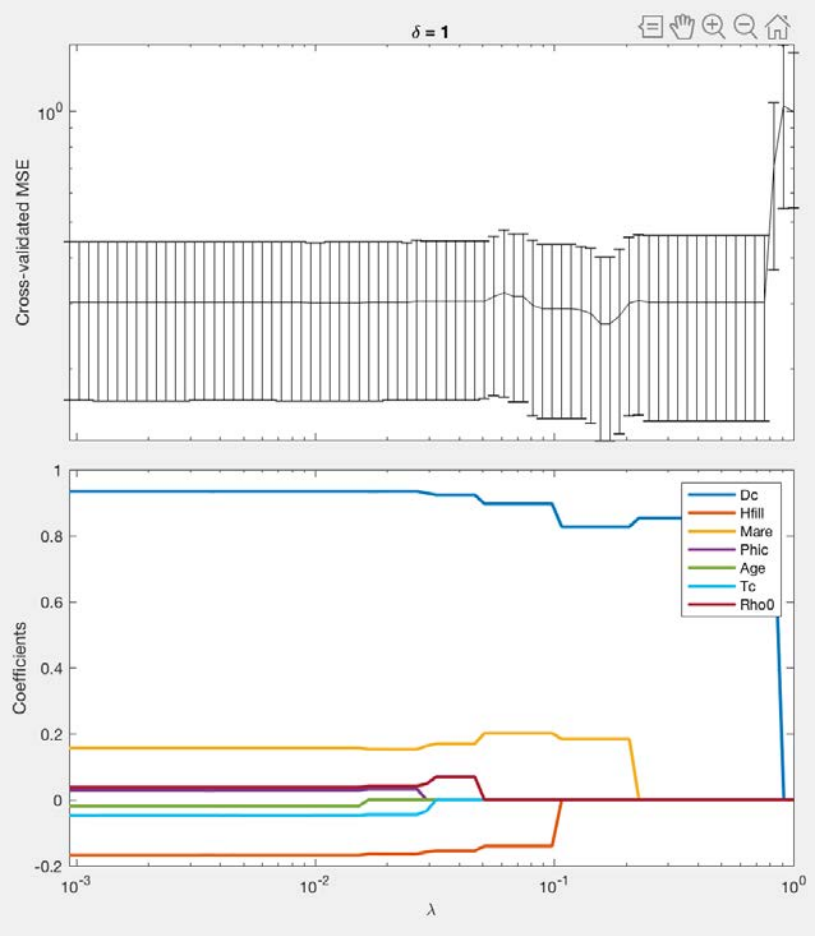
Meinshausen, 2007, CSDA

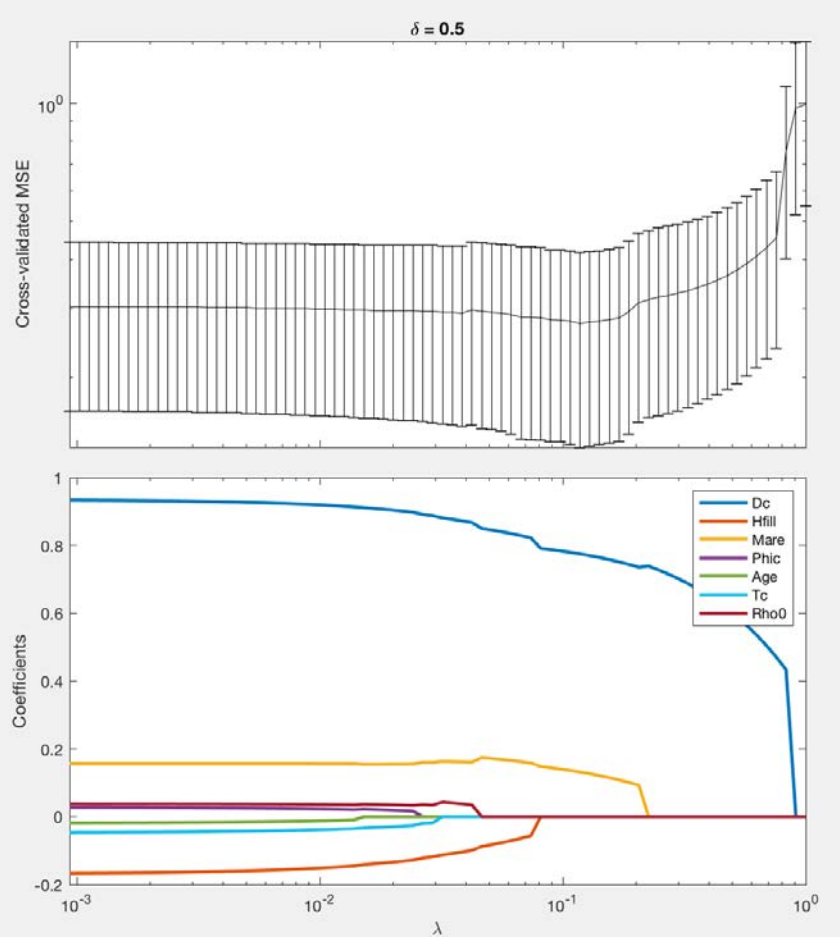
Example 2: Feature Selection Using Relaxed Lasso

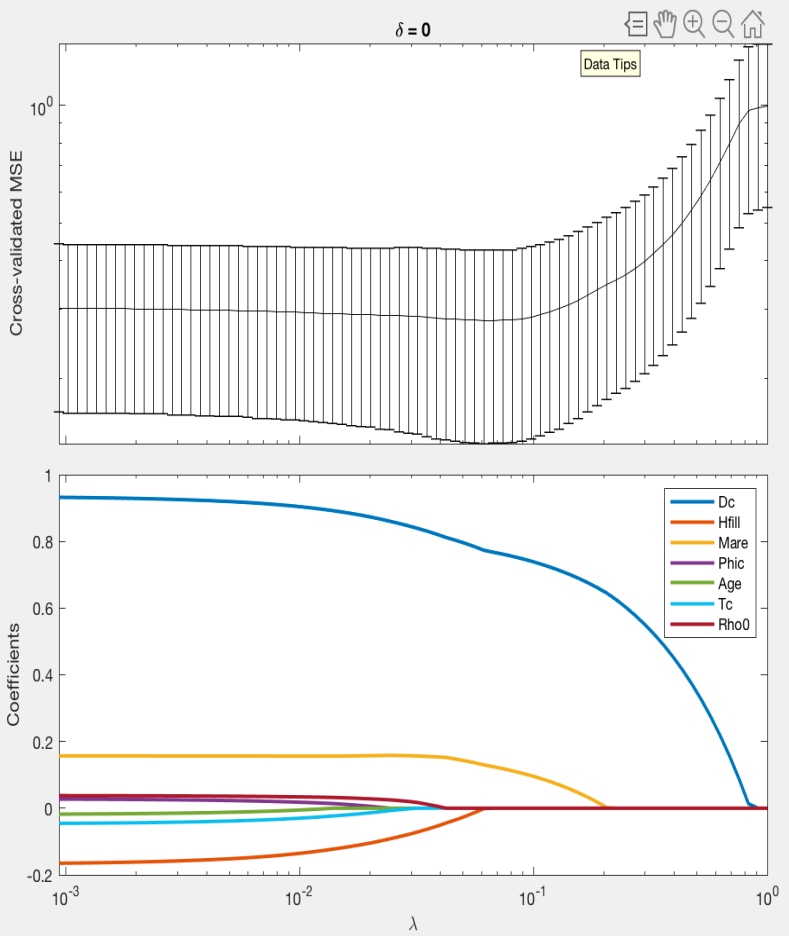


- $Relaxed\ Lasso$: $Cost\ function$: $P = \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda\phi\|\beta\|_1$ (*after selecting p parameters*)
- *Relaxed lasso is equivalent to the "Forward-Lasso Adaptive Shrinkage" (FLASH) algorithm (Radchenko and James, Ann App Stats, 2011): [Shrinkage para] δ=1- [relaxation para] ϕ*
- *The penalization parameter λ controls the variable selection part, while the relaxation parameter ϕ controls the shrinkage of parameters.*

Sensitivity to tuning parameters







- *Relaxed to "Best subset selection"*
- *Better for high SNR regime*

- *No relaxation or full shrinkage"Lasso"*
- *better for noisy problem (low SNR regime)*

- Best solution: p = 2 (Dc, Mare)
- Adding more parameters can also explain the data fairly well.

Feature Transformation Techniques

Another dimensionality reduction approach is to use feature extraction or feature transformation techniques, which transform existing features into new features (predictor variables) with the less descriptive features dropped.

- **Principal component analysis (PCA)**, used to summarize data in fewer dimensions by projection onto a unique orthogonal basis
- **Factor analysis,** used to build explanatory models of data correlations
- **Nonnegative matrix factorization,** used when model terms must represent non-negative quantities, such as physical quantities.

Example 3: PCA

Constrained Regression

lsqlin

Solve constrained linear least-squares problems

Syntax

```
x = lsqlin(C,d,A,b)
x = lsqlin(C,d,A,b,Aeq,beq,lb,ub)
x = lsqlin(C,d,A,b,Aeq,beq,lb,ub,x0,options)
x = lsqlin(problem)
[x,resnorm,residual,exitflag,output,lambda] = lsqlin( __ )
```

Description

Linear least-squares solver with bounds or linear constraints.

Solves least-squares curve fitting problems of the form

$$\min_x \frac{1}{2}\|C \cdot x - d\|_2^2 \text{ such that } \begin{cases} A \cdot x \le b, \\ Aeq \cdot x = beq, \\ lb \le x \le ub. \end{cases}$$

References

- Regression Analysis: https://ocw.mit.edu/courses/mathematics/18-s096-topics-in-mathematics-with-applications-in-finance-fall-2013/video-lectures/lecture-6-regression-analysis/
- Regularization: https://www.youtube.com/watch?v=sO4ZirJh9ds
- Logistic Regression and PCA: https://www.coursera.org/learn/machine-learning
- Matlab Documentation
- Wikipedia

- Data: https://zenodo.org/record/3833814
- Penalized (relaxed lasso regression): https://github.com/w-mcilhagga/penalized