Introduction
Traditional Nonpara-metrics
HDLM
Penalized Estimation
λ - CV
λ - Theory
Penalized Estimation Examples
Comments on Penalized Estimation

Notes 1: Introduction and Penalized Estimation

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

1. Introduction

Introduction

"Big Data":

1. Large data sets: e.g. n observations, p variables, np too big to fit on a computer
   - e.g. eBay, Google, Amazon
2. High-dimensional models: n observations, p parameters, $n \approx p$ or $n \ll p$
   - traditional nonparametrics (easy to see with series/sieves)
   - text data, big survey data sets
   - flexible "parametric models", semiparametric models
3. Statistical Learning/Data-mining: Exploit data to get good forecasting rules
   - want model that is flexible enough to accommodate important patterns but not so flexible it overspecializes to specific data set (goal of nonparametrics)

Predictive Models

Useful to think about relation between target (Y) and input (X) as

$$\underbrace{Y_i}_{\text{target}} = \underbrace{f(X_i)}_{\text{signal}} + \underbrace{\varepsilon_i}_{\text{noise}}$$

Goal: Learn $f(\cdot)$ from the data in a way that yields "generalizable" forecasts

OR

Get a forecast rule that minimizes expected forecast loss

- Focus on squared error loss minimization - $f(X) = \mathrm{E}[Y|X]$

Nonparametrics

Nonparametric estimation/inference:

- fundamentally about fitting flexible models to data
- allows data to guide researcher in learning by leveraging weak structure imposed by researcher (e.g. continuity and differentiability)
  - Structure provides dimension reduction by not requiring the researcher to try to learn the value of a function at point x^* by looking only at observations with $x_i = x^*$
- tries to trade off bias and variance in estimation by adapting model complexity to the data at hand
- fundamentally about description/prediction but underlies learning about more "structural" parameters (e.g. treatment effects)

Nonparametrics

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Traditional nonparametric approaches (e.g. kernels, series, ...) perform poorly when size of the input space is large (curse of dimensionality)

- if unwilling to assume much, need LOTS of data before informative conclusions can be drawn

More recent nonparametric approaches - high-dimensional models

- drawn largely from machine learning, data-mining
- tend impose more structure - more dimension reduction - than traditional nonparametrics
- scale better computationally and as the dimension of the problem increases

3. Traditional Nonparametric Methods

Coarsely Discrete Regressors

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Suppose that X can take on R values, $\{x_1, ..., x_R\}$. E.g.

- Gender, R = 2.
- Years of Schooling, R = 20ish.
- Gender x Years of Schooling, R = 2*20ish.

Estimation of $\mathrm{E}[Y|X = x_r]$ is easy!!

- Find all observations with $x_i = x_r$ and calculate sample mean with this subsample
- No assumptions about $\mathrm{E}[Y|X]$ - completely flexible
- Will have usual properties as long as R finite (just learning about R expectations)

Example: Conditional Wages

Data:

- 329,505 men from 1980 U.S. Census
- aged 40-49
- 0-20 years of schooling
- race (black, white)
- married (married, non-married)

Condition on schooling or age:

- Schooling:
  - 21 categories
  - average of 15,691 observations per category
  - Range: 215-122,934
- Age:
  - 40 categories
  - average of 8238 observations per category
  - Range: 7327-9683

Example: Conditional Wages

Intro to HD Estimation

Introduction

Traditional Nonparametrics

HDLM

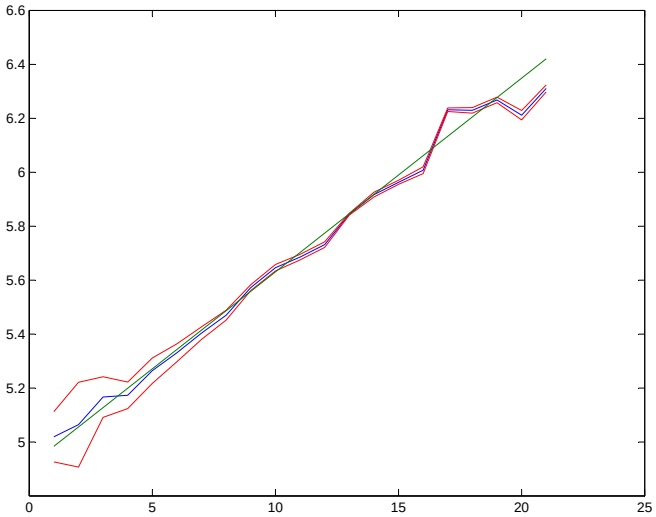
Penalized Estimation

λ - CV

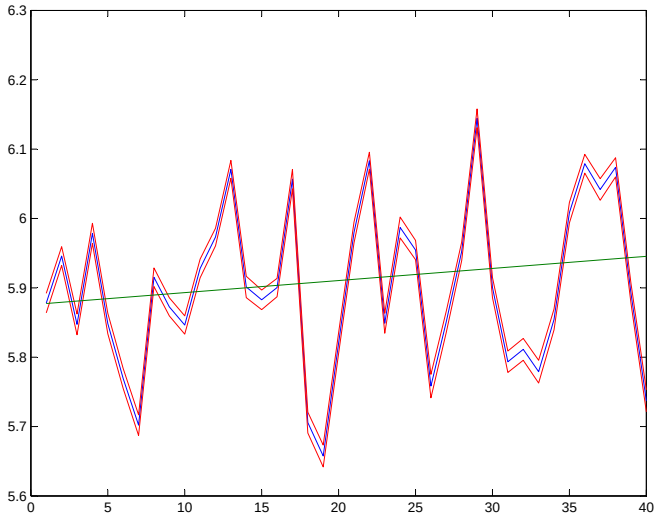
λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation



(a) Schooling



(b) Age

Example: Conditional Wages

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

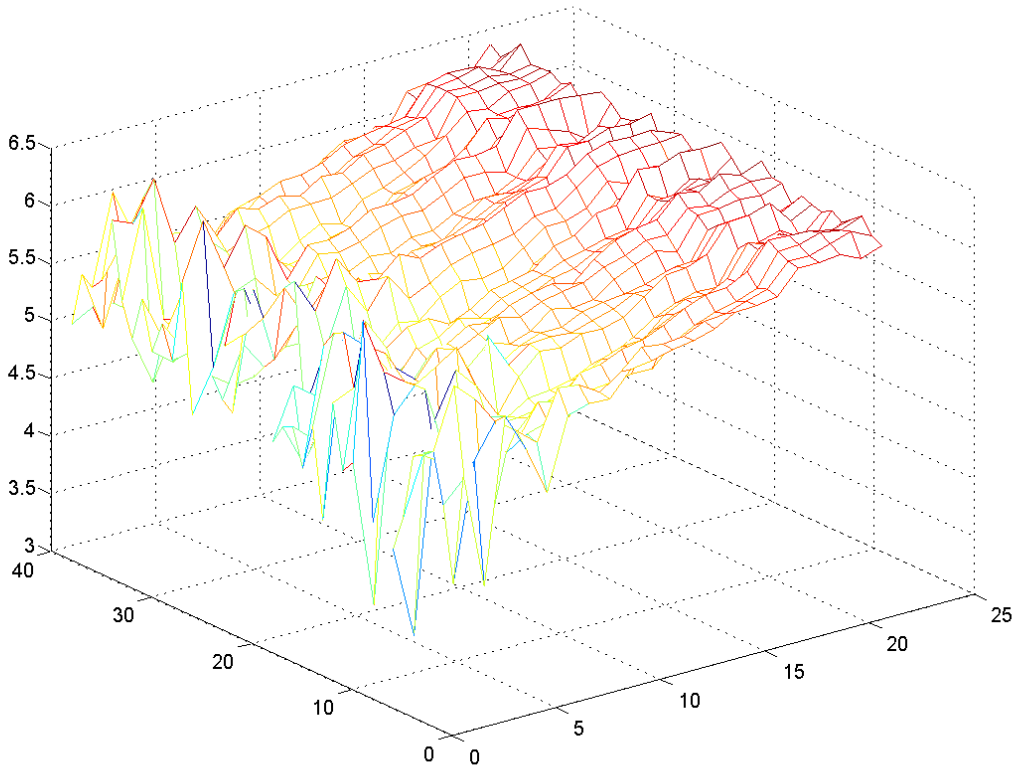
λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Condition on schooling and age:

- 840 categories
- average number of observations per category: 392 (large range: 0 - 4181)



Example: Conditional Wages

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Things only get worse as we condition on Race and Marital Status

- 3360 categories
- average of 98 observations per category
- Range: 0 - 3635
- 259 empty categories (7.7%)
- 670 categories with 0-2 observations (19.9%)

Example: Conditional Wages

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

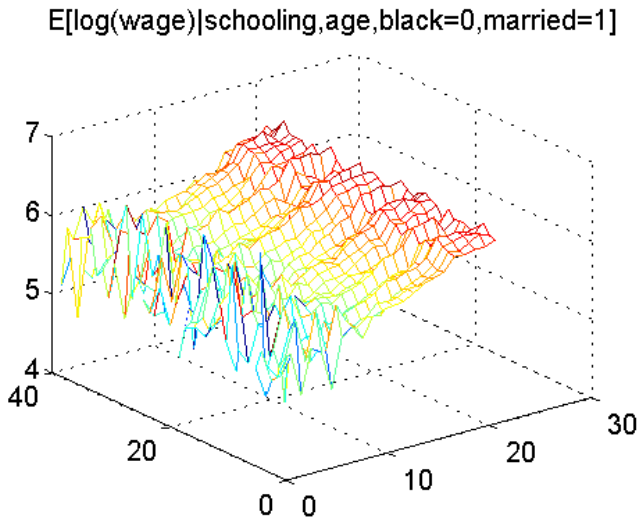
Penalized Estimation

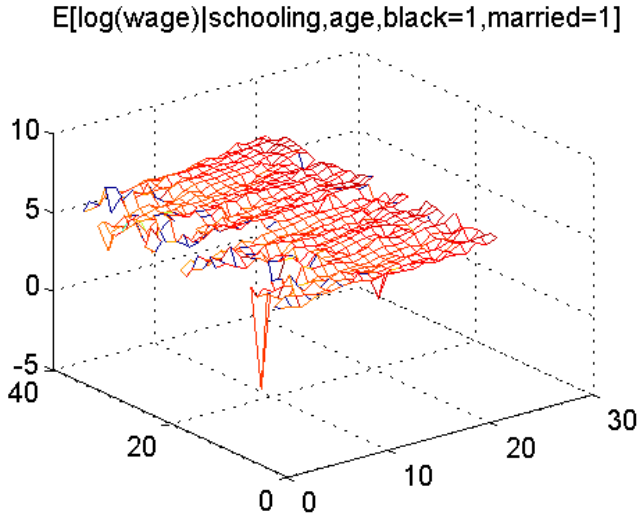
λ - CV

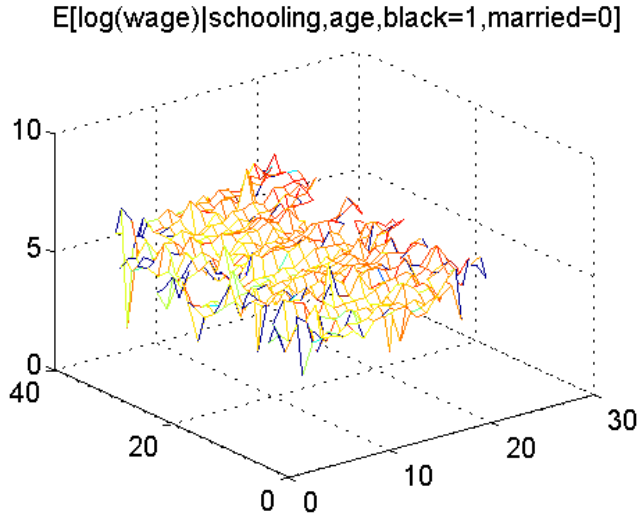
λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation







Example: Conditional Wages

Some questions:

1. Do we really think conditional expectation function is that bumpy?
2. What do we do about categories with 0 observations?
3. Estimates cell by cell are unbiased. What happens to variance as the number of cells increases?
4. Suppose we conditioned on State of Birth too? (171,360 categories) [Curse of dimensionality]

Fundamental statistical learning problem - Need for Regularization:

Structure and estimators that trade off bias and variance to produce reasonable forecasts/models

Traditional Nonparametrics - Kernels

Literally averaging for each separate x value is only feasible in cases where X is coarsely discrete - need beliefs/regularization

- Smoothness: E.g. $\mathrm{E}[Y|X]$ is a smooth (e.g. continuous, differentiable, etc.) function of X
  - Function shouldn't change much across values of X that are close
  - Estimate $\mathrm{E}[Y|X = x^*]$ by averaging y's over values of x close to x^*

Kernel Regression:

$$\widehat{\mathrm{E}[Y|X = x^*]} \equiv \widehat{g}(x^*) = \frac{\sum_{i=1}^{n} y_i K_h(x_i - x^*)}{\sum_{i=1}^{n} K_h(x_i - x^*)}$$

where $K_h(\cdot)$ is a kernel function and h is a bandwidth.

Some kernel details

Common Univariate Kernels:

- Uniform: $K_h(u) = \frac{1}{2h} 1(|u| < h)$
- Gaussian: $K_h(u) = \frac{1}{\sqrt{2\pi h^2}} \exp\{\frac{-u^2}{2h}\}$
- Epanechnikov: $K_h(u) = \frac{3}{4h}\left(1 - \left(\frac{u}{h}\right)^2\right)_+$
- Triangular: $K_h(u) = \frac{1}{h}\left(1 - \frac{|u|}{h}\right)_+$

Multivariate kernels:

- Most common to just take product of univariate kernels ("product kernel")
- Any multivariate density would also work
  - E.g. $q-$dimensional multivariate normal with $q \times q$ bandwidth matrix H

Intuition using uniform kernel

With uniform kernel,

$$\begin{aligned}\widehat{g}(x^*) &= \frac{\sum_{i=1}^n y_i 1(|x_i - x^*| < h)}{\sum_{i=1}^n 1(|x_i - x^*| < h)} \\ &= \frac{1}{n_{x^*,h}} \sum_{i \,:\, |x_i - x^*| < h} y_i\end{aligned}$$

- $n_{x^*,h}$ is the number of observations such that $|x_i - x^*| < h$
- I.e. estimator is just sample average of the y_i across all points where $|x_i - x^*| < h$

Local averaging picture

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

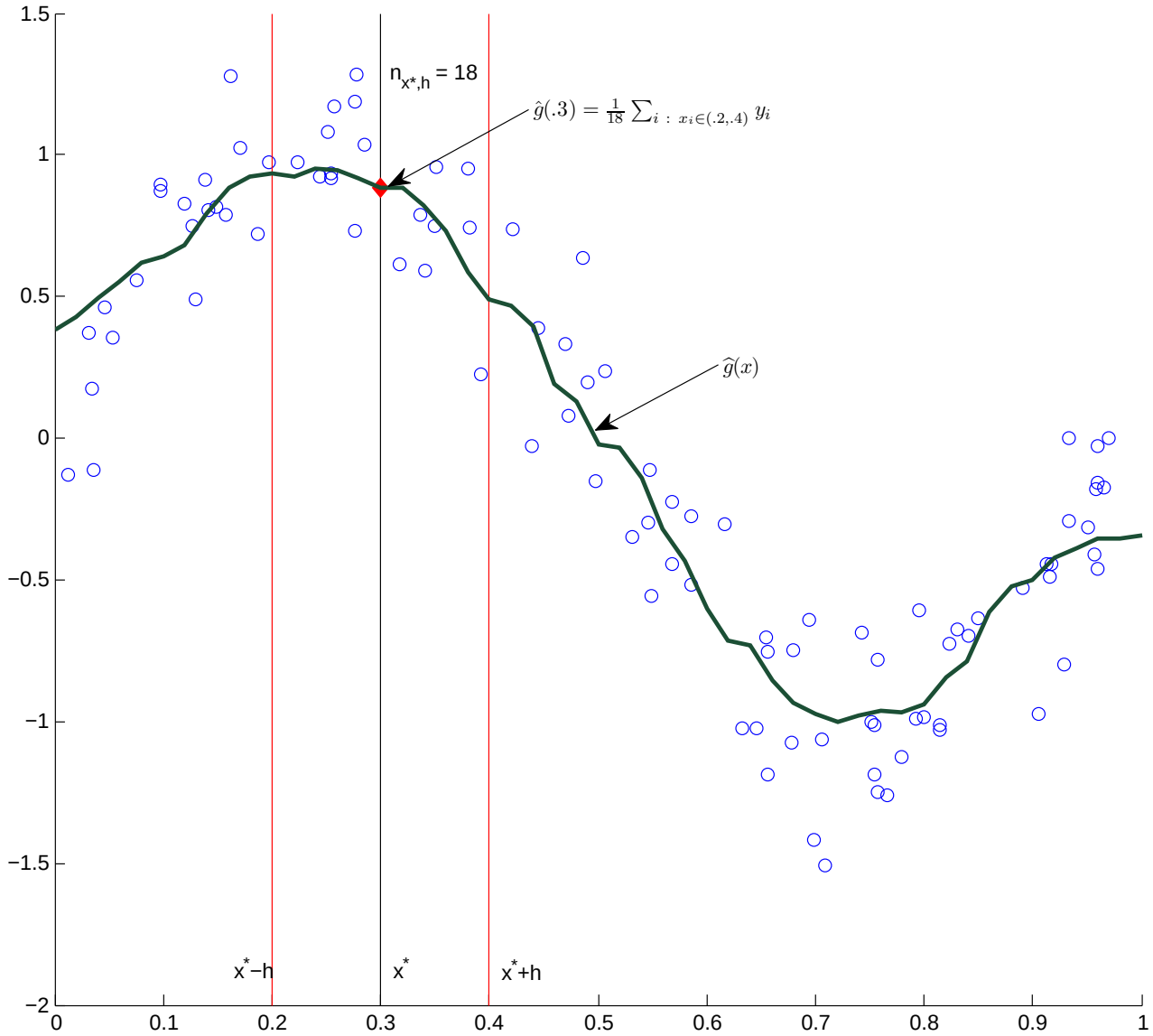
Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation



Traditional Nonparametrics - K-NN

Intro to HD Estimation

Introduction

Traditional Nonparametrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

K-NN:

- $\widehat{f}(x) = \frac{1}{K} \sum_{i:d(x_i,x) \leq d(x_{(K)},x)} y_i$
  - K: number of neighbors to use
  - $d(x_1, x_2)$: distance from point x_1 to point x_2, usually Euclidean
  - x_K the observation ranked K^{th} in distance from target point x
- Can be viewed as kernel with varying bandwidth

Kernels and K-NN are local methods

Aside: Scaling

Some data-mining/nonparametric methods depend heavily on scale of observations

E.g. KNN - very common to use Euclidean distance:

$$\|x_i - x_k\| = \sqrt{\sum_{j=1}^{p}(x_{j,i} - x_{j,k})^2}$$

- very different answer depending on scale of x's (e.g. meters versus centimeters)

Very common to scale x's before analysis:

- E.g. standardize: $\frac{x_{j,i} - \bar{x}_j}{s_j}$
- Scale to $[0, 1]$: $\frac{x_{j,i} - \min x_{j,i}}{\max x_{j,i}}$

Traditional Nonparametrics - Series

Series:

- Model $f(x) = \sum_{j=0}^{p} \beta_j p_j(x) + r(x)$
- $p_j(x)$ are series/basis terms
  - E.g. $\{p_j(x)\} = 1, x, x^2, x^3, ...$ (or orthogonal polynomials)
  - E.g. $\{p_j(x)\} = 1, x, x^2, x^3, (x - k_4)_+^3, (x - k_5)_+^3, ...$ where $k_4, k_5, ...$ are "knots" (cubic spline)
- Obtain $\widehat{f}(x)$ by LS regression of Y on $\{p_j(X)\}_{j=1}^{p}$
- Global method

Regularization comes in through the choice of p.

- Higher p means less bias since we are leaving out less terms from the infinite sum
- Higher p means higher variance since we are estimating more regression coefficients from the same amount of data

Series Estimation

Operationally, series are extremely easy

- Define $\varphi^n(x) = (\varphi_{n,1}(x), ..., \varphi_{n,p}(x))'$
- Define $Z_n = [\varphi^n(x_1), ..., \varphi^n(x_n)]'$ (an $n \times p$ matrix)

The series estimator of $\mathrm{E}[Y|X = x]$:

$$\widehat{g}(x) = \varphi^n(x)'\widehat{\beta} \text{ where } \widehat{\beta} = (Z_n'Z_n)^{-1}(Z_n'Y)$$

I.e. estimate coefficients by OLS of Y on Z_n. Can also do inference using conventional OLS output (e.g. Newey (1997))

The Curse of Dimensionality

Traditional nonparametric approaches perform poorly when the dimension of the covariate space increases.

For example, how does data "fill-in" as dimensionality increases?

- Think about data uniformly distributed on the p-dimensional unit cube
  - To get fraction b of the observations, need b of the volume
  - On average, will need a cube with edge length $b^{1/p}$
  - Neighbors aren't so local! (e.g. $p = 10$, $b = .01$, need $.01^{.1} \approx .63$ - cover 63% of the support of each input

Traditional nonparametric estimators impose too little structure to produce useful models in high-dimensions.

3. High-Dimensional Linear Model

HDLM: OLS Estimator

High-dimensional linear model (HDLM):

$$Y_i = X_i'\beta + \varepsilon_i = \sum_{j=1}^{p} \beta_j X_{j,i} + \varepsilon_i, \quad \mathrm{E}[\varepsilon_i | X_{1,i}, ..., X_{p,i}] = 0, \quad p \gg n$$

Suppose observed data matrix X is full (row) rank

Least squares estimator solves $X'X\beta = X'Y$ and yields family of solutions:

$$\widehat{\beta}^w = (X'X)^- X'Y + [I - (X'X)^- X'X]w$$

for A^- the Moore-Penrose generalized inverse and w an arbitrary conformable vector

No unique coefficient estimator.

HDLM: OLS Prediction

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Least squares fitted values in HDLM satisfy

$$\begin{aligned}\widehat{Y} &= X\widehat{\beta}^{w} \\ &= X(X'X)^{-}X'Y + [X - X(X'X)^{-}X'X]w \\ &= XX'(XX')^{-1}(XX')^{-1}XX'Y + [X - XX'(XX')^{-1}(XX')^{-1}XX'X]w \\ &= Y\end{aligned}$$

using $(X'X)^{-} = X'(XX')^{-1}(XX')^{-1}X$ (property of Moore-Penrose inverse)

I.e. any solution perfectly fits Y within sample.

Linearity insufficient structure to allow informative estimation and inference when dimensionality gets high enough. Need further regularization/dimension reduction.

HDLM Prediction

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Goal: Find a linear combination of X that provides a good forecast of Y

Need more structure:

- Popular additional structure is sparsity
  - Of the $p \gg n$ available predictors, only $s \ll n$ are needed to obtain a high-quality prediction of Y
- Want to find which (if any) elements of X we can drop and still get good forecasts of Y
- Note that dropping a variable $\Leftrightarrow$ setting coefficient on that variable to 0

Series/Sieves

Note that nonparametric series/sieve estimation falls within this framework:

$$\begin{aligned} y_i &= g(z_i) + u_i \\ &= x_i'\beta + r_i + u_i = x_i'\beta + \varepsilon_i \end{aligned}$$

- z_i some low-dimensional set of observed variables
- $x_i = \{p_k(z_i)\}_{k=1}^{p}$; e.g. $x_i = \{1, z_i, z_i^2, z_i^3, ..., z_i^p\}$
- Believe decent forecast available using fewer than n series terms
- Allows approximation errors (just absorbed in ε in these notes) - needs to be dealt with explicitly in the theory
  - E.g. Belloni, Chernozhukov, and Hansen (2014) “Inference on Treatment Effects after Selection amongst High-Dimensional Controls”

Bias-Variance Tradeoff and Variable Selection

How do we decide which variables to drop?

- Leaving out variable with strong correlation to signal $\Rightarrow$ model too simplistic $\Rightarrow$ bias
- Putting in many variables $\Rightarrow$ hard to learn about all corresponding coefficients $\Rightarrow$ variance
- As with any nonparametrics, want enough variables to capture predictability without over-complicating the model

In principle, want to try all possible combinations and choose the one that does the best job forecasting out-of-sample

- Underlies best subsets selection methods
- Best subsets computationally infeasible when the dimension of X is not small
  - Forward selection and stepwise selection meant to approximate best subsets solution

Penalized estimation offers another computationally tractable approach to address the selection problem.

4. Penalized Estimation

Regularization

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Key idea in much of contemporary statistics is regularization:

At a high-level, *regularization* is just introducing additional information to solve an ill-posed inverse problem

- a very long history of use in mathematics
- any useful statistical method is doing some kind of regularization (e.g. mean is not allowed to vary arbitrarily for each observation)
- Many modern statistical methods *explicitly* introduce regularization by directly penalizing model complexity
  - E.g. information criteria (such as AIC, BIC) for choosing variables

Penalized Estimation

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Penalized estimator solves

$$\widehat{f} = \arg\min_{f} L(\text{data}, f) + \lambda C(f)$$

- $L(\text{data}, f)$ is a loss function that decreases as model fits data better (e.g. sum of squared residuals)
  - In principle, works for essentially any loss function and (theoretical) results available for common ones
- $C(f)$ is a penalty (cost) function that increases in model complexity
- λ is penalty parameter (price) that controls how fit versus complexity are balanced

Squared Error Loss and Linear Models

Canonical examples:

$$\widehat{\beta}_P = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda p(\beta)$$

- Ridge: $p(\beta) = \sum_{j=1}^{p} \psi_j \beta_j^2$
- LASSO: $p(\beta) = \sum_{j=1}^{p} |\psi_j \beta_j|$
- "ℓ_q": $p(\beta) = \sum_{j=1}^{p} |\psi_j \beta_j|^q$

Squared Error Loss and Linear Models

More elaborate examples:

- Elastic Net:

$$p(\beta) = \sum_{j=1}^{p} |\psi_{1,j}\beta_j| + \lambda_2 \sum_{j=1}^{p} \psi_{2,j}\beta_j^2$$

- SCAD: For $a > 2$,

$$p(\beta) = \sum_{j=1}^{p} \begin{cases} |\psi_j\beta_j| \text{ if } |\psi_j\beta_j| \leq \lambda \\ -\frac{|\psi_j\beta_j|^2 - 2a\lambda|\psi_j\beta_j| + \lambda^2}{2\lambda(a-1)} \text{ if } \lambda < |\psi_j\beta_j| \leq a\lambda \\ \frac{(a+1)\lambda}{2} \text{ if } |\psi_j\beta_j| > a\lambda \end{cases}$$

  (a quadratic spline with knots at λ and $a\lambda$)

- Lava: Decompose $\beta = \delta + \gamma$,

$$p(\beta) = \sum_{j=1}^{p} |\psi_{1,j}\delta_j| + \lambda_2 \sum_{j=1}^{p} \psi_{2,j}\gamma_j^2$$

High-Level Comments on Penalty Functions

- Impose regularization by shrinking coefficients towards 0 (in principle could shrink to other value)
- Complexity measured by size of estimated coefficients
- Builds in a belief that model is not too complicated in that coefficients should not be "too big"
- Scale of variables (and thus coefficients) very important for this belief
  - Standard implementations assume homoscedasticity and standardize data *ex ante*
  - Take all the ψ's = 1 in this case
  - More generally, proper choice of ψ can allow for heteroscedasticity, non-Guassian errors, dependence
    - To my knowledge, only formally worked out for LASSO under heteroscedasticity (BCH 2012) and clustering (BCHK 2016)
- Key tradeoffs are type of shrinkage and computational complexity

Penalty Functions

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

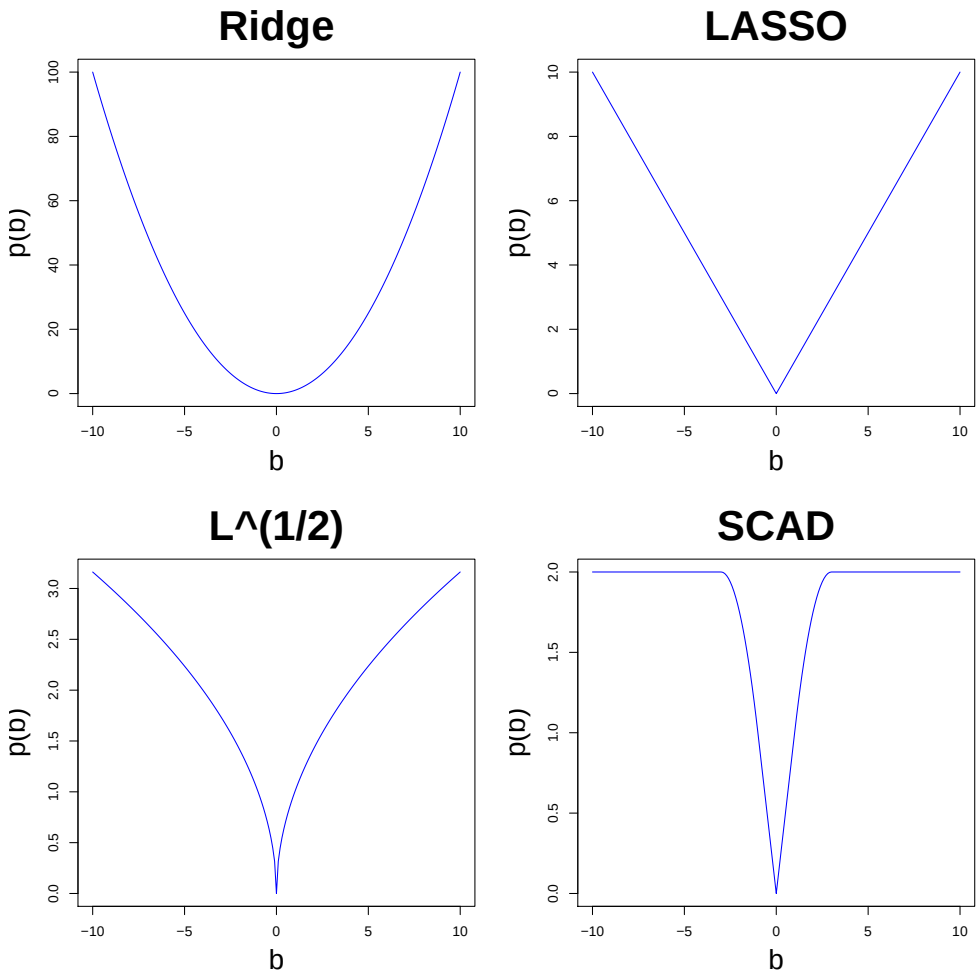
Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation



All have $\lambda = 1$; $a = 3$ for SCAD

Shrinkage Functions

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Shrinkage functions offer another device for thinking about what different penalties do

Idea: Consider a simple case where problems have explicit solution in terms of MLE

Shrinkage function captures what penalty function does to MLE in that setting

E.g. consider least squares estimation with orthonormal input matrix so each $\widehat{\beta}_{MLE}$ obtained by marginal regression

Shrinkage Functions

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

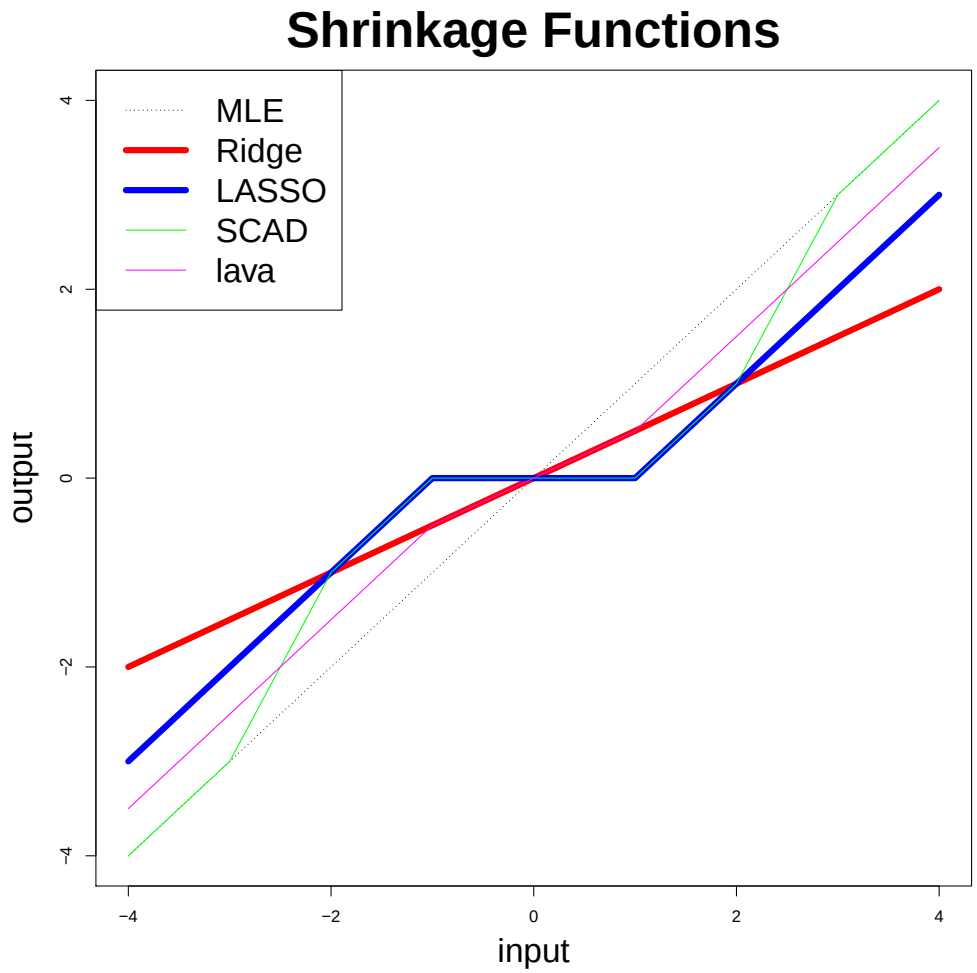
Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation



$\lambda = 1$ for all cases, $a = 3$ for SCAD, $\lambda_2 = 1$ for lava

Constrained Optimization Problem

Intro to HD Estimation

Introduction

Traditional Nonparametrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Penalized estimators have equivalent formulation in terms of a constrained optimization problem:

E.g. LS criterion:

$$\widehat{\beta}_P = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i'\beta)^2$$
$$\text{subject to} \quad p(\beta) \leq M$$

Equivalent to penalized formulation when M and λ set appropriately

Easy to see how kinked penalties lead to variable selection in this framework

Constrained Optimization Problem Contours

Intro to HD Estimation

Introduction

Traditional Nonparametrics

HDLM

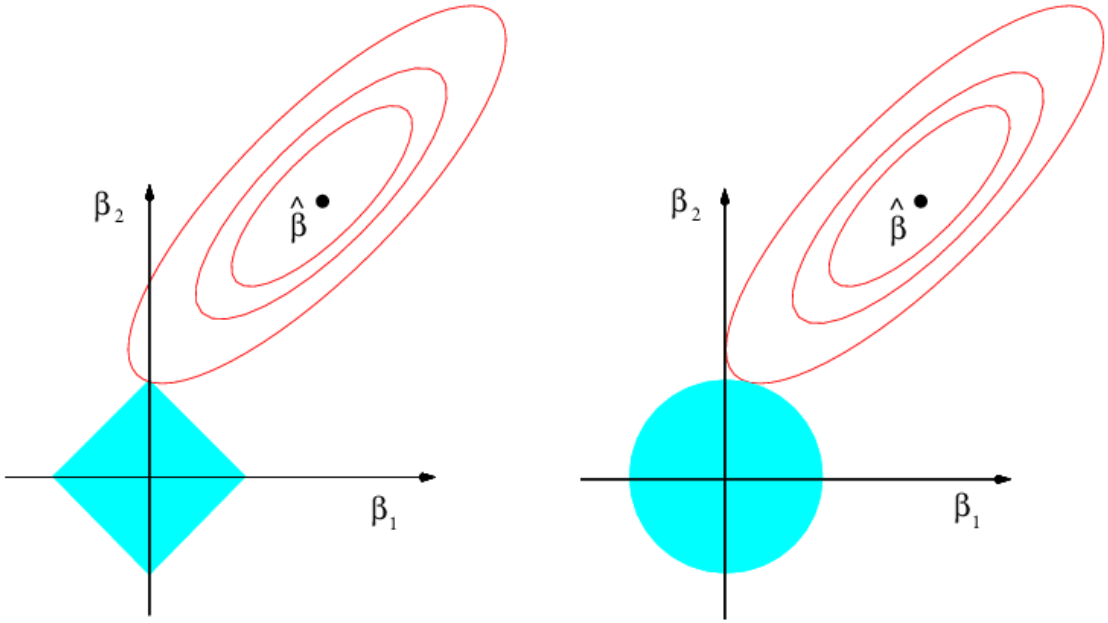
Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation



Contours of LS criterion in 2-D case and shape of constraint region (LASSO - diamond, Ridge - circle). (Figure 6.7 from ISL.)

Shrinkage Bias

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Penalized estimators with convex penalty functions computationally convenient (coupled with convex loss)

Shrink all coefficients towards 0

- Great for coefficients that are really zero (or vanishingly small)
- May lead to substantial biases for non-zero coefficients

Non-convex penalties (e.g. SCAD, ℓ_p with $p < 1$) motivated by desire to mitigate this bias

Adaptive LASSO alters penalty loadings based on first step estimate to alleviate bias

Post-Penalized Estimation (Belloni and Chernozhukov (2013))

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Computationally simple and intuitive idea to undo shrinkage bias by applying unpenalized estimator using only variables selected to have non-zero cofficient

E.g.

$$\widehat{\beta}_L = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \|\beta\|_1$$

Post-LASSO estimator

$$\widehat{\beta}_{PL} = \arg \min_{\beta : \beta_j = 0 \ \forall \ j \text{ such that } \widehat{\beta}_{L,j} = 0} \sum_{i=1}^{n} (y_i - x_i'\beta)^2$$

Post-LASSO Estimation

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Results in Belloni and Chernozhukov (2013) suggest Post-LASSO works at least as well as LASSO and sometimes much better

Seems to work well with theoretically driven plug-in penalty given in Belloni and Chernozhukov (2013)

Be careful with cross-validation

5. Penalty Parameter Choice: Cross-Validation

Choice of penalty parameter

With penalty selected to produce desired shrinkage/regularization, still need to choose appropriate tuning parameters (e.g. λ)

Cross-validation (CV) standard in statistics

Convex penalties are computationally very convenient

- E.g. fast/efficient algorithms for LASSO produce solution path for ALL λ

Can also do some theory in this case to get some guidance (e.g. BCH 2012, BCHK 2016)

- Probably especially useful in non-iid/dependent data settings

Model Validation

We don't really care about explaining what we've seen.

What matters is accuracy out-of-sample

If we got a bunch of new data NOT USED TO ESTIMATE MODEL (say $\{x_i^o, y_i^o\}_{i=1}^m$), could use this data to evaluate models:

$$\text{Out-of-Sample MSE}(\lambda) = \frac{1}{m}\sum_{i=1}^{m}(y_i^o - \widehat{y}_i^o)^2 = \frac{1}{m}\sum_{i=1}^{m}(y_i^o - \widehat{f}(x_i^o;\lambda))^2$$

Model Validation

Don't really have data we haven't seen...

Key Idea: Use sample to replicate forecasting environment by splitting data to estimate out-of-sample predictive ability

Split the data:

- Training Sample - data used to estimate prediction rule(s)
- Testing Sample - data used to test estimated rule(s) on NEW data
  - AKA "validation" or "hold-out" sample

Model Validation

Training data: $\{y_i^T, x_i^T\}_{i=1}^{n_T} \to \widehat{f}_T(\cdot; \lambda)$

Testing data: $\{y_i^o, x_i^o\}_{i=1}^{m}$, $m + n_T = n$

Estimate of out-of-sample MSE:

$$\widehat{MSE}(\lambda) = \frac{1}{m} \sum_{i=1}^{m} (y_i^o - \widehat{f}_T(x_i^o; \lambda))^2$$

Cross-Validation

Idea of splitting sample and using a training and validation sample good but

1. Depends on choice of training and validation samples
   - Which observations in which sample?
   - How many observations in each?
2. Not using all observations for estimation AND testing

Cross-validation (CV) refinement of sample splitting idea that helps address these concerns

Leave-one-out CV

Leave-one-out CV (LOOCV):

1. For $i = 1$ to n
   - Estimate model using $(Y_{-i}, X_{-i}) \rightarrow \widehat{f}_{-i}(\cdot; \lambda)$
   - Forecast $\widehat{y}_i(\lambda) = \widehat{f}_{-i}(x_i; \lambda)$
2. Estimate generalization error as $\widehat{MSE}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i(\lambda))^2$
   - Or use any other criterion for comparing forecast to realization as appropriate

- Pretty close to solving 2 issues noted previously

Do this for each model/tuning parameter under consideration

Choose the model that does best according to forecasting criterion (or 1SE rule)

K-Fold CV

LOOCV has lots of nice features but not most common procedure used.

- Can be computationally intensive, especially in problems where "big data" is interesting
- All training samples essentially identical (highly correlated)
  - less variability in fitted models than under true repeated sampling
  - underestimate prediction risk

Current standard is K-Fold CV (with $K = 5$ or 10)

K-Fold CV

1. Divide sample into K approximately equal sized groups
2. For $g = 1$ to K:
   - Use subsample g as validation sample and use remaining groups as training sample
   - Estimate model using $(Y_{-g}, X_{-g}) \rightarrow \widehat{f}_{-g}(\cdot; \lambda)$
   - Forecast $\widehat{y}_i(\lambda) = \widehat{f}_{-g}(x_i; \lambda)$ for all $i \in \mathcal{I}_g$ where $\mathcal{I}_g$ is the set of all indices belonging to group g
3. Estimate generalization error as $\widehat{MSE}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i(\lambda))^2$
   - Or use any other criterion for comparing forecast to realization as appropriate

Do this for each model/tuning parameter under consideration

Choose the model that does best according to forecasting criterion (or 1SE rule)

6. Penalty Parameter Choice: Theory

Lasso Problem

Recall that lasso estimates parameters by solving

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

Lasso problem is convex (has a unique solution) but is not differentiable

Can find solution by looking at subdifferential

- subderivative of function $f(\cdot)$ at point x_0 is a set of vectors v such that $f(x) - f(x_0) \geq v'(x - x_0)$
- at a point where a function is differentiable, subdifferential is the conventional gradient
- a convex function is minimized at the point where 0 is included in the subdifferential

Scalar Lasso problem

Specialize to case where $\dim(x_i) = 1$, $\frac{1}{n}\sum_i x_i^2 = 1$, $\psi_1 = 1$, so lasso solves

$$\widehat{\beta}_P = \arg\min_\beta Q(\beta) = \arg\min_\beta \sum_{i=1}^n (y_i - \beta x_i)^2 + \lambda|\beta|$$

Subdifferential:

$$\begin{aligned}\partial Q(\beta) &= -2x'y + 2n\beta + \lambda \text{ if } \beta > 0 \\ &= -2x'y + 2n\beta - \lambda \text{ if } \beta < 0 \\ &\in -2x'y + 2n\beta + s\lambda \text{ for } s \in [-1,1] \text{ if } \beta = 0\end{aligned}$$

Estimator $\widehat{\beta}_P$ found at point where 0 is in the subdifferential at that point:

$$\begin{aligned}\widehat{\beta}_P &= \frac{1}{n}x'y - \frac{1}{2n}\lambda && \text{if } \frac{1}{n}x'y - \frac{1}{2n}\lambda > 0 \\ &= \frac{1}{n}x'y + \frac{1}{2n}\lambda && \text{if } \frac{1}{n}x'y + \frac{1}{2n}\lambda < 0 \\ &= 0 && \text{if } |\frac{1}{n}x'y| \leq \frac{1}{2n}\lambda\end{aligned}$$

Intuition for penalty parameter choice

Estimate β to be exactly 0 whenever $|\frac{1}{n}x'y| \leq \frac{1}{2n}\lambda$

A desirable property would be that we get $\widehat{\beta} = 0$ when β really is 0 with high-probability

- get this by choosing any λ "big enough"
- but big λ implies more shrinkage on non-zero coefficients

Implies choosing λ such that

$$\Pr(|\frac{1}{\sqrt{n}}x'\varepsilon| \leq \frac{1}{2\sqrt{n}}\lambda) \to 1$$

- $\frac{1}{\sqrt{n}}x'\varepsilon \overset{a}{\sim} N(0, \sigma^2)$ [assuming, e.g., iid sampling, $\varepsilon \perp x$, and $\mathrm{E}[\varepsilon^2] = \sigma^2$]
- Suggests choosing $\lambda = 2\sqrt{n}\sigma\Phi^{-1}(1 - \gamma_n/2)$ for $\gamma_n \to 0$

High-p, non-iid case

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Look back at general problem

$$\widehat{\beta}_P \in \arg\min_b \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i'b)^2 + \frac{\lambda}{n}\sum_{j=1}^{p}\widehat{\psi}_j|b_j|.$$

Need to choose λ and $\widehat{\psi}_j$, $1 \leq j \leq p$.

Key to good selection properties of Lasso is choosing these so that

$$\frac{\lambda\widehat{\psi}_j}{n} \geq 2c\left|\frac{1}{n}\sum_{i=1}^{n}x_{j,i}\epsilon_i\right| \quad \text{for each } 1 \leq j \leq p$$

occurs with high probability.

General Intuition for Choice of λ

1. Previous inequality holding $\Leftrightarrow \lambda/\sqrt{n} \geq 2c \left| \frac{1}{\sqrt{n}\widehat{\psi}_j} \sum_{i=1}^{n} x_{j,i}\epsilon_i \right|$ for each $1 \leq j \leq p$.
   - Setting $\lambda/\sqrt{n}$ large enough to dominate p standard normals would work if $\frac{1}{\sqrt{n}\widehat{\psi}_j} \sum_{i=1}^{n} x_{j,i}\epsilon_i$ were standard normal.
   - $\lambda = 2c\sqrt{n}\Phi^{-1}(1 - \gamma_n/2p)$ with $\gamma_n = o(1)$ will implement this

General Intuition for Choice of ψ_j

2. Need ψ_j to be an appropriate measure of the variability of $\frac{1}{\sqrt{n}} \sum_{i=1}^{n} x_{j,i}\epsilon_i$
   - "Ideally": $\widehat{\psi}_j = \psi_j$ where

$$\psi_j^2 = \text{Var}\left(\frac{1}{\sqrt{n}} \sum_{i=1}^{n} x_{j,i}\epsilon_i\right)$$

   - Suggests using $\widehat{\psi}_j$ a consistent estimator of Var $\left(\frac{1}{\sqrt{n}} \sum_{i=1}^{n} x_{j,i}\epsilon_i\right)$
   - Results for independent, heteroskedastic case given in Belloni, Chernozhukov, and Hansen (2012) using Huber-Eicker-White variance estimator
   - Results for clustered case given in Belloni, Chernozhukov, Hansen, and Kozbur (2016) using clustered variance estimator
     - Both rely on application/extension of moderate deviation theory of Jing, Shao, and Wang (2003)

Implementation

To implement in practice, need to form $\widehat{\psi}_j$.

Feasible iterative procedure:

1. Form initial guess about $\{\varepsilon_i\}_{i=1}^n$, $\{\widehat{\varepsilon}_i\}_{i=1}^n$
   - Simple choice is to set $\widehat{\varepsilon}_i = y_i$
   - Another choice is to set $\widehat{\varepsilon}_i = y_i - (x_i^0)'\widehat{\beta}^0$ where x_i^0 is a (small) set of initial variables thought likely to be important and $\widehat{\beta}^0$ are the associated least squares regression coefficients
2. Form $\widehat{\psi}_j = \widehat{\text{Var}}\left(\frac{1}{\sqrt{n}}\sum_{i=1}^n x_{j,i}\epsilon_i\right)$ using $\widehat{\varepsilon}_i$ in place of ε_i
3. Estimate lasso coefficients with λ given above and $\widehat{\psi}_j \to \widehat{\beta}_P$
4. Update $\widehat{\varepsilon}_i = y_i - x_i'\widehat{\beta}_P$
5. Repeat 2-4 a small number of times.

Some Formal Properties

- Let $s = \|\beta\|_0$
- $a_n s$ for $a_n \to \infty$ dimensional submatrices of $\frac{1}{n}\sum_i x_i x_i'$ have minimum and maximum eigenvalue bounded with probability approaching one
- lots of bounded moments
- $\frac{s^2 \log^2(p)}{n} \to 0$ and $\frac{\log^3(p)}{n} \to 0$

Results:

- $\frac{1}{n}\sum_i (x_i'\beta - x_i'\widehat{\beta}_P)^2 = O_p(s\log(p)/n)$
  - Best possible forecast rate
- $\widehat{s} = O(s)$
  - Selected model has similar size to true model - note that no guarantee you get the right variables

7. Penalized Estimation Examples

Example: Baseball Data

Let's look at ridge and LASSO in building a model to predict a baseball player's salary given performance metrics

Code: Hitters_Example2.R

- Y (*Salary*): Salary in \$1000 in 1987
- X_1 (*AtBat*): Number of at bats in 1986

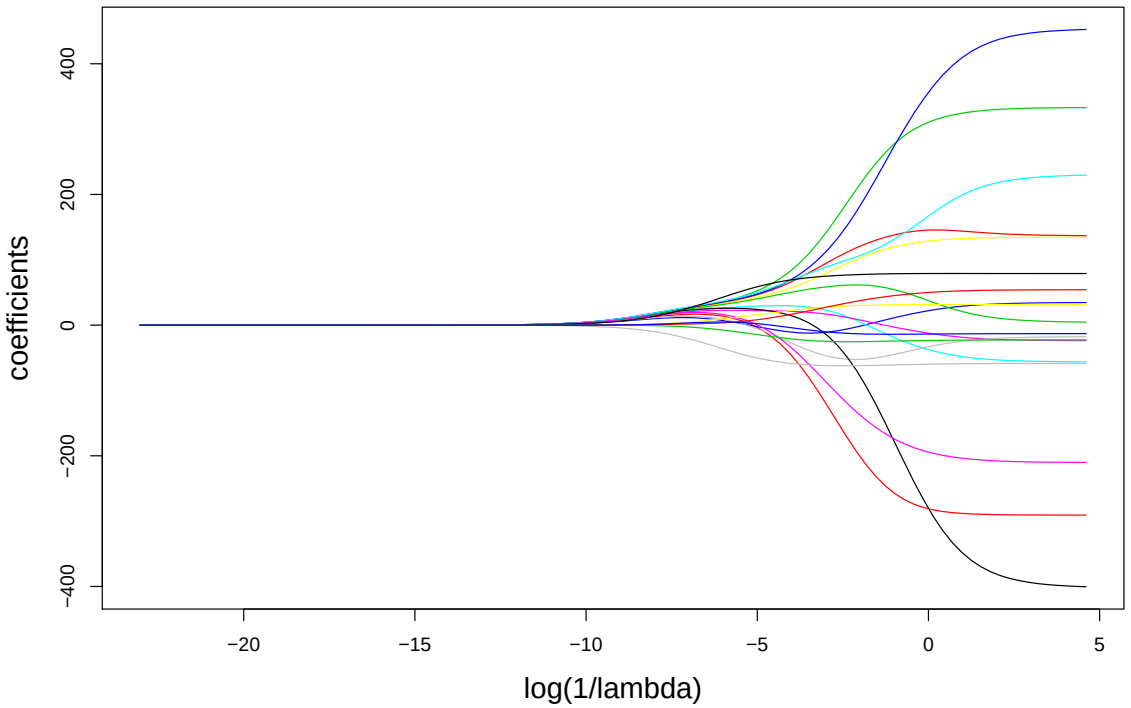
...

- X_{19} (*NewLeague*): Player's league at start of 1987 (American or National)

All x-variables standardized

Example: Baseball Data

Plot Ridge coefficients against $\log(1/\lambda)$



Move smoothly between (essentially) 0 and (essentially) the unpenalized values

Example: Baseball Data

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

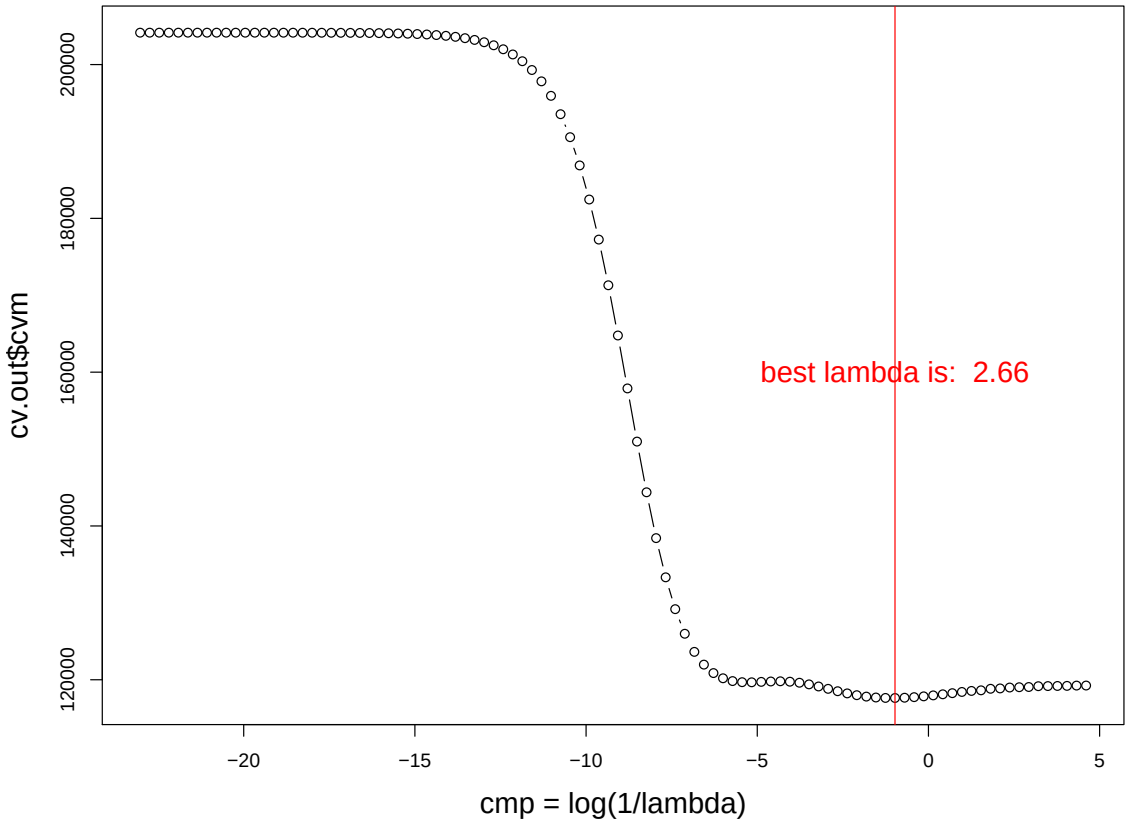
λ - CV

λ - Theory

Penalized Estimation Examples

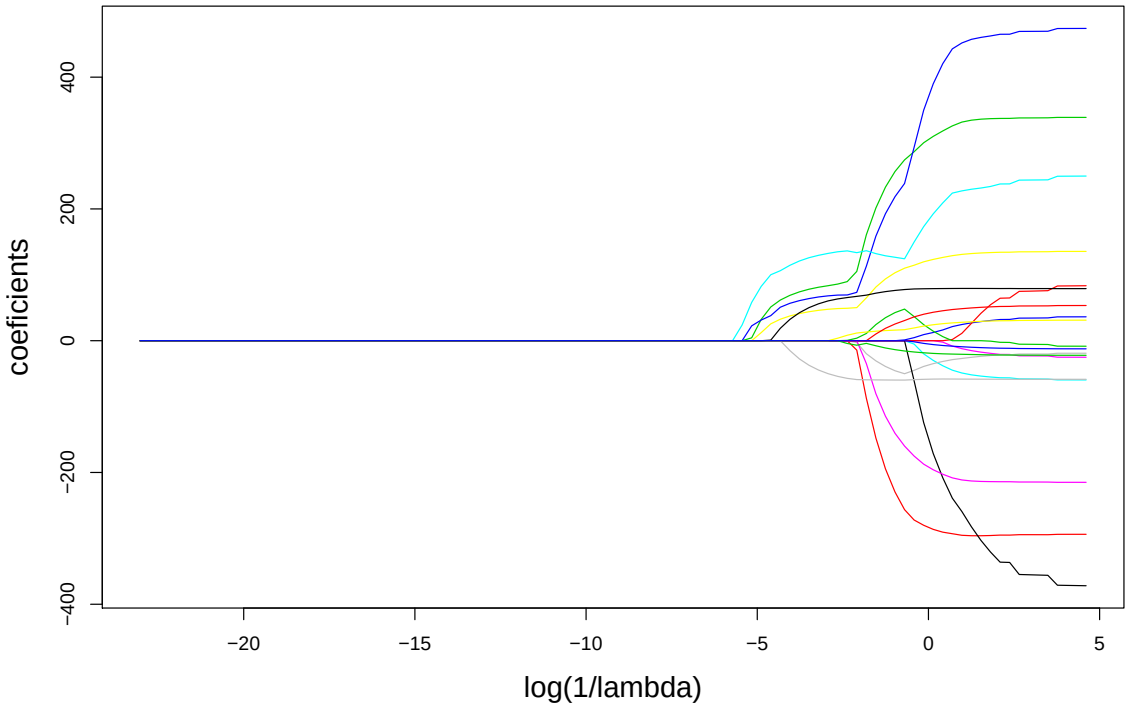
Comments on Penalized Estimation

Can choose which value of λ to use based on CV:



Example: Baseball Data

Plot LASSO coefficients against $\log(1/\lambda)$



Move (fairly) smoothly between (exactly) 0 and (essentially) the unpenalized values

Coefficients get zeroed out along the path

Example: Baseball Data

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

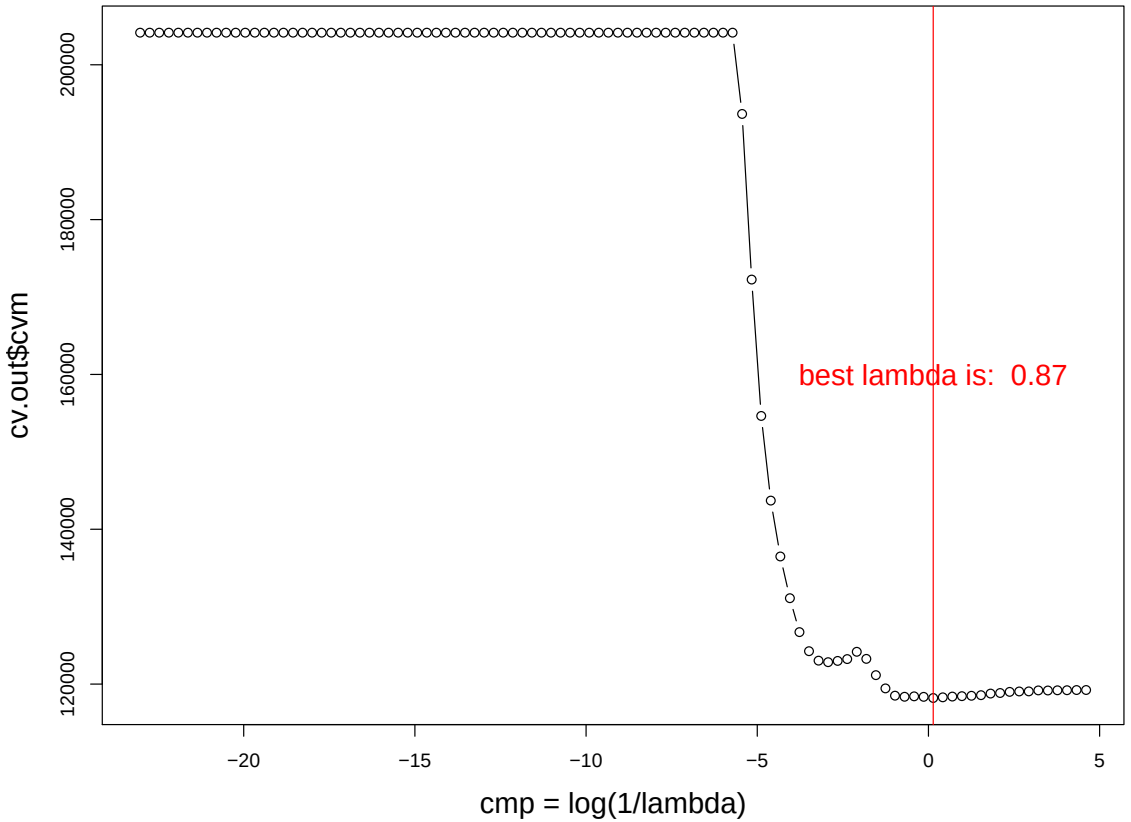
λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Can choose which value of λ to use based on CV:



Example: Baseball Data

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

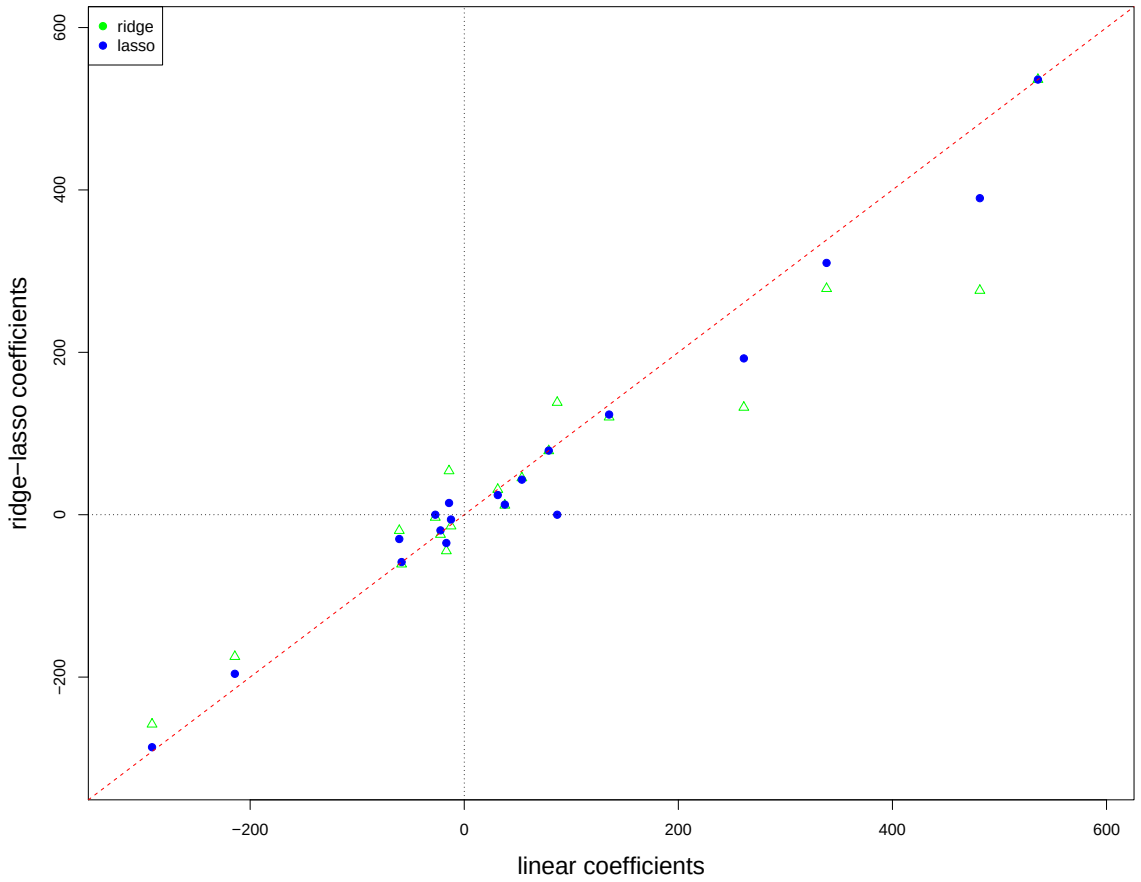
λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

How do CV-min Ridge and LASSO coefficients compare to unpenalized?



Example: Baseball Data

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

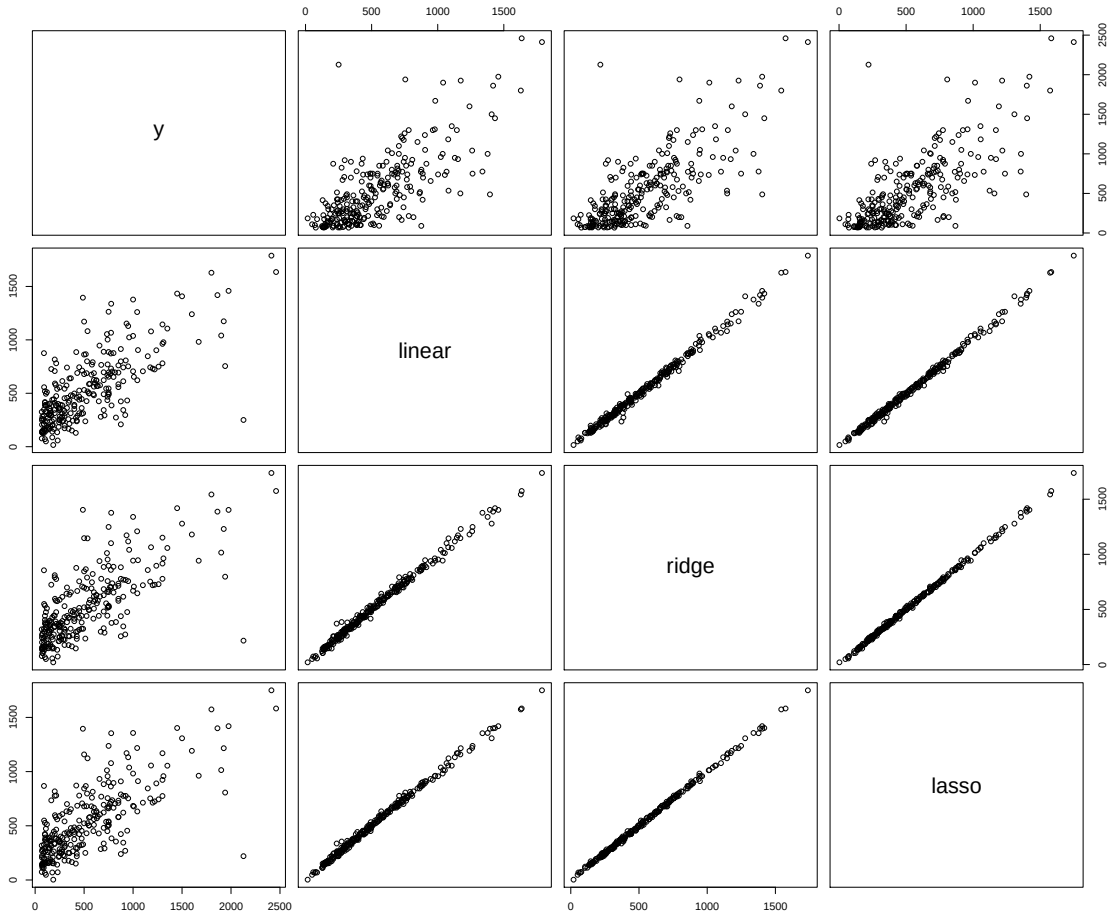
λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

How do CV-min Ridge and LASSO fitted values compare to unpenalized?



Example: Baseball Data

Pretty close agreement in this example

LASSO model is slightly more parsimonious than ridge or full

	Int.	AtBat	Hits	HmRun	Runs	RBI	Walks	Years
OLS	535.9	-291.6	338.5	37.9	-60.7	-27.0	135.3	-16.7
Ridge	535.9	-258.0	278.4	11.6	-19.6	-3.3	120.2	-44.6
LASSO	535.9	-286.4	310.1	12.5	-29.9	0	123.4	-34.8
	CAtBat	CHits	CHmRun	CRuns	CRBI	CWalks	Lea	DivW
OLS	-391.8	86.9	-14.2	481.7	261.2	-214.3	31.3	-58.5
Ridge	-180.8	138.21	53.95	276.2	132.2	-174.8	31.0	-60.9
LASSO	-171.2	0	14.5	389.9	192.5	-195.9	24.2	-58.3
	PutOuts	Assists	Err	NewLea				
OLS	78.9	53.8	-22.2	-12.4				
Ridge	78.6	45.3	-24.4	-14.0				
LASSO	79.0	43.2	-19.3	-5.9				

Example: Boston Housing Data

Code: LassoHousing_Example3.R

Boston housing data from Harrison and Rubinfield (1978) - conveniently included in the R library "MASS":

- Outcome: *medv* - median home value
- Predictors: 13 raw predictors in data
  - *crim* (per capita crime rate)
  - *zn* (proportion residential land zoned for lots over 25,000 ft^2)
  - *indus* (proportion of non-retail business acres)
  - *chas* (Charles River dummy)
  - *nox* (nitrogen oxide concentration)
  - *rm* (average rooms per dwelling)
  - *age* (proportion of owner-occupied units build pre 1940)
  - *dis* (weighted mean of distances to 5 employment centers)
  - *rad* (index of highway accessibility)
  - *tax* (property tax rate per $10,000)
  - *ptratio* (pupil-teacher ratio by town)
  - *black* (1000*(proportion black - .63)^2)
  - *lstat* (percent lower SES)

Example: Boston Housing Data

Want to have a flexible model that captures predictive power in observed explanatory variables.

Consider:

- Charles River dummy
- Cubic polynomial in all continuous variables including all first and second order interactions
- dummy variables formed by splitting all continuous variables at deciles
  - where possible - otherwise form deciles, take unique elements and form dummies
- interact *lstat* and *dis* dummies with sixth order polynomials in *lstat* and *dis* and third order polynomials in *crim*, *nox*, *tax*, and *ptratio*
- interact other dummies with cubic in base variable (e.g. *crim* dummies with cubic in *crim*)
- 1240 variables (including some redundant ones)

Example: Boston Housing Data

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Consider Ridge and LASSO

Choose penalty parameter using 10-fold CV

Example: Boston Housing Data

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

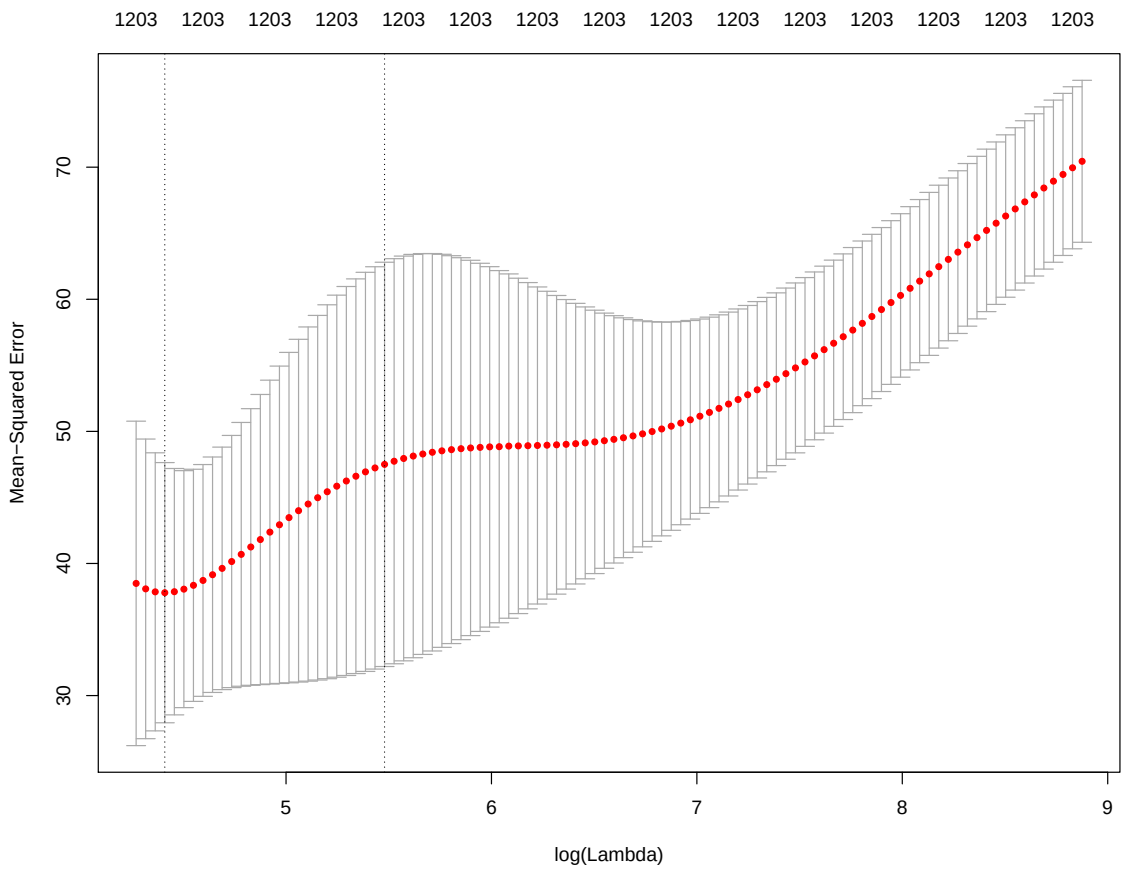
λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

CV function for Ridge:



Lines give CV-minimizing value and largest value within 1 estimated standard error

Example: Boston Housing Data

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

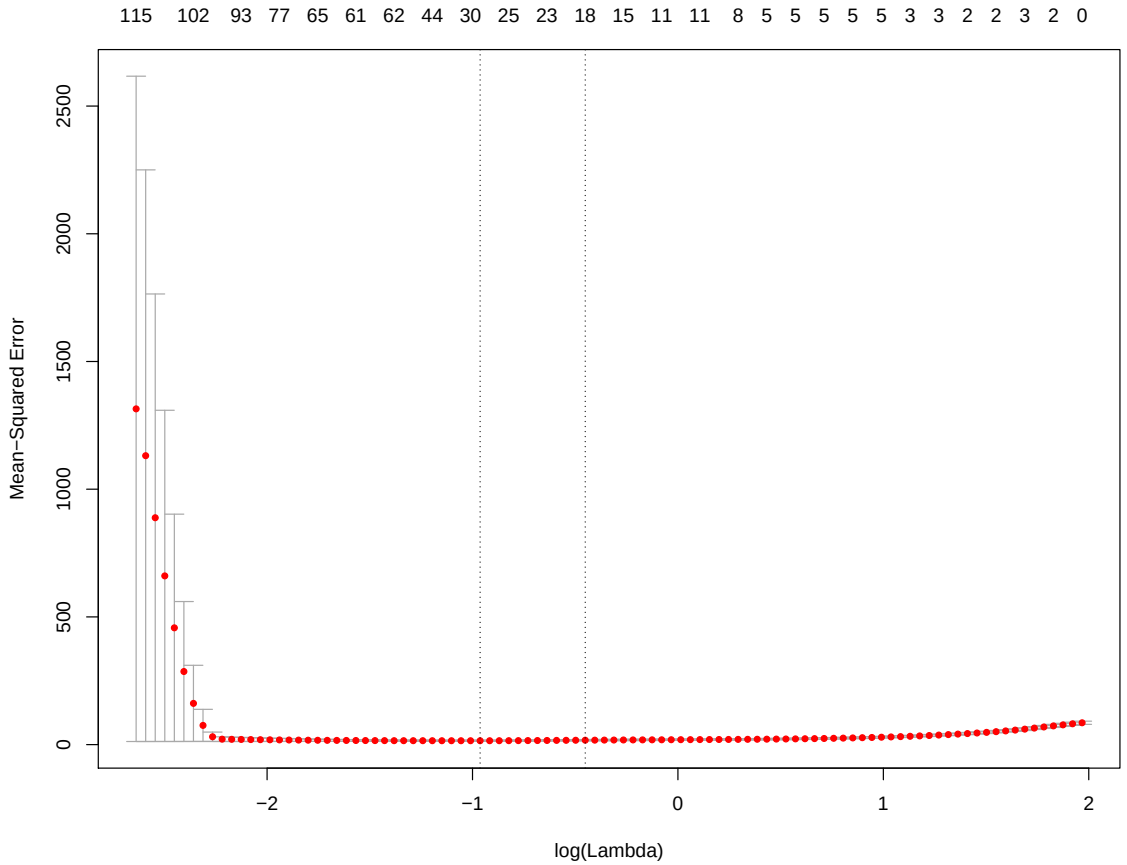
λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

CV function for LASSO:



Lines give CV-minimizing value and largest value within 1 estimated standard error

Example: Boston Housing Data

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Performance on validation data set:

- OLS with raw variables: Validation MSE: 29.96
- Ridge(CV-Min): Validation MSE = 29.23
- Ridge(1SE): Validation MSE = 35.83
- LASSO(CV-Min): Validation MSE = 19.43
- LASSO(1SE): Validation MSE = 20.89

Example: Riboflavin Production

Code: LassoRiboflavin_Example4.R

Classic high-dimensional example

$n = 71$ observations, $p = 4088$ right-hand-side variables

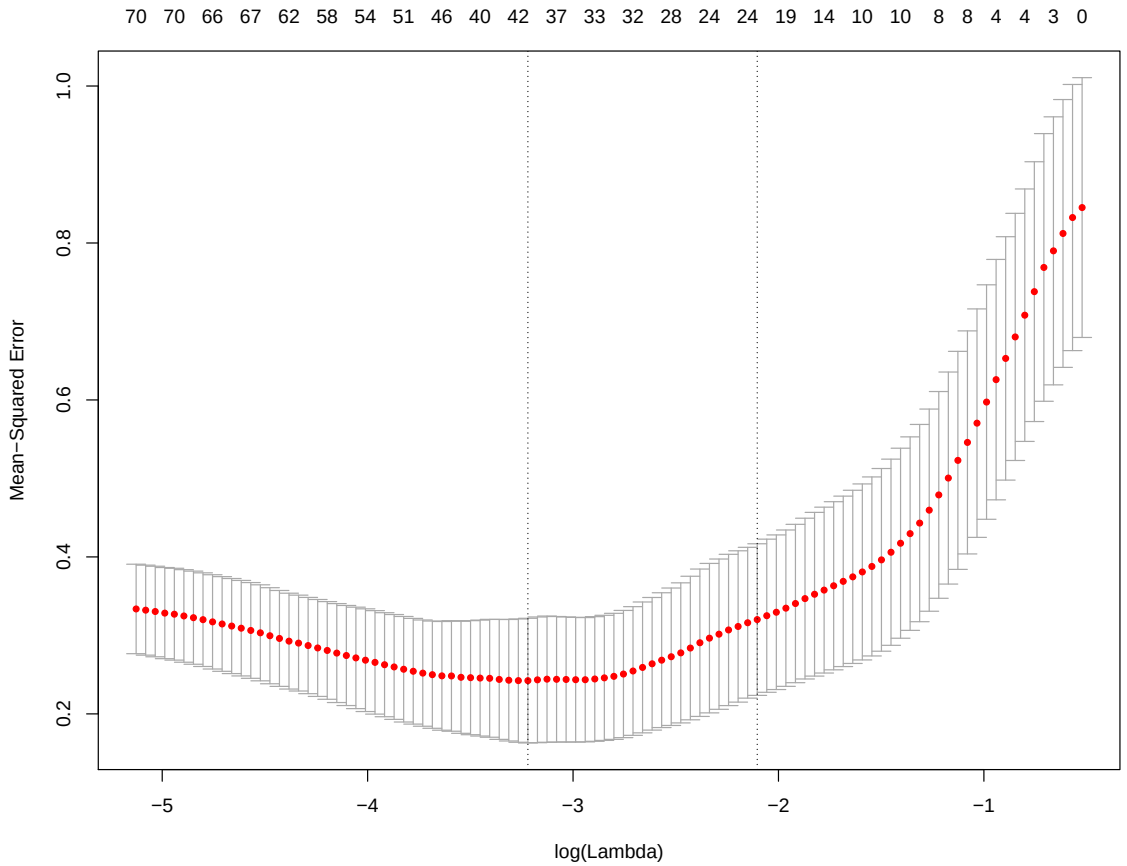
Data:

- Y - log(riboflavin production)
- X_j - log(expression level of gene$_j$), $j = 1, ..., 4088$

Goal: Uncover genes likely associated with riboflavin production

Example: Riboflavin Production

5-Fold CV function for LASSO:



Example: Riboflavin Production

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

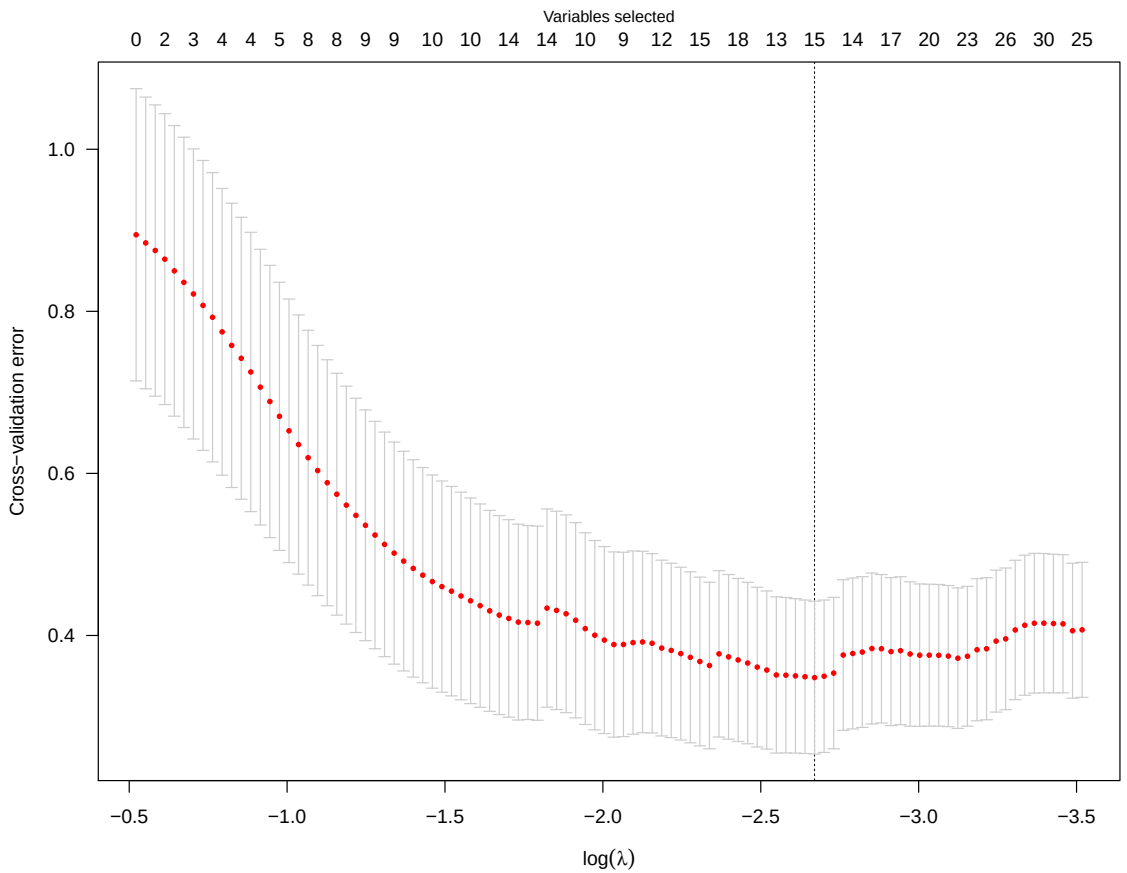
λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

5-Fold CV function for SCAD:



Example: Riboflavin Production

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

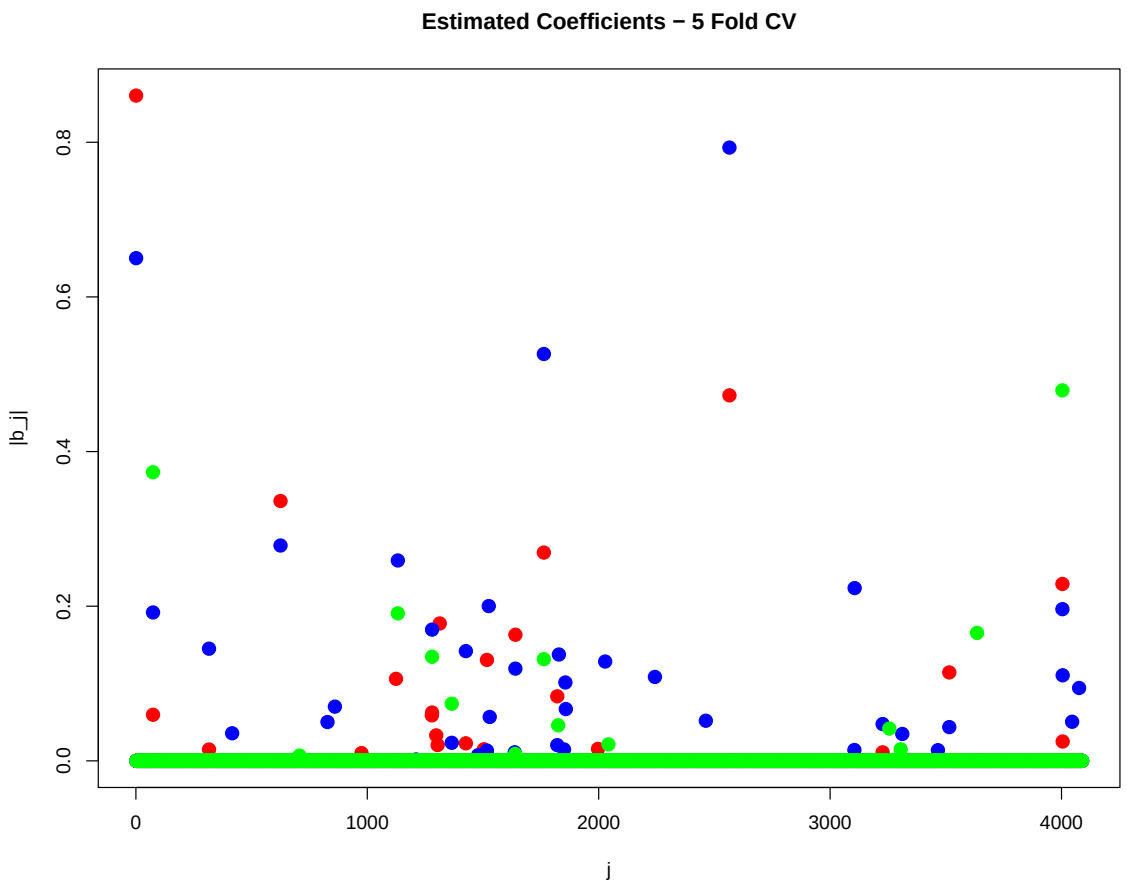
λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Estimated (absolute value of) coefficients (blue - lasso-cv-min, red - lasso-cv-1se, green - scad-cv-min):



23, 40, and 16 variables selected by LASSO (1SE), LASSO (MIN), SCAD (MIN)

Increasing p Simulation

Look at (simulated) LASSO performance in a couple of simple settings:

Simulation Design 1:

- $Y = X_1 + .25\varepsilon$
- $(X_1, ...X_p, \varepsilon)' \sim N(0, I_{p+1})$
- $n = 100$, $p \in \{1, 5, 20, 50, 90, 100, 200\}$

Simulation Design 2:

- $Y = \exp(-X_1/2) + .75X_1 1(X_1 > 0) + .1(\sum_{j=2}^{5} X_j) + .1(X_2 - X_3 + X_4 - X_5)^2 + .1\varepsilon$
- $(X_1, ...X_p, \varepsilon)' \sim N(0, I_{p+1})$
- $n \in \{100, 200\}$, $p \in \{1, 5, 20, 50, 100, 200\}$

Increasing p Simulation

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Consider (i) lasso using only linear terms and (ii) lasso using nonlinear terms formed by "basis" expansion

Note that there's a huge computational bottleneck in large p cases.

- Allowing for nonlinearity and interactions in all the terms blows up the computational overhead quickly
- E.g. with p variables and only allowing second order effects there are $p + p(p+1)/2$ terms - (20300 terms with $p = 200$) - That's a really big design matrix
- Allow all first and second order terms + dummies for quartiles interacted with linear term

Increasing p Simulation

Intro to HD Estimation

Introduction

Traditional Nonparametrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Let's look at results in linear model first.

LASSO penalty parameter allowing nonlinear terms selected by 5-fold CV

	$p = 1$		$p = 5$		$p = 20$		$p = 50$	
	CV	MSE	CV	MSE	CV	MSE	CV	MSE
OLS	0.056	0.064	0.058	0.069	0.076	0.074	0.134	0.119
LASSO	0.056	0.064	0.057	0.066	0.058	0.066	0.059	0.070
LASSO.NL	0.051	0.068	0.055	0.070	0.060	0.071	0.060	0.076

Increasing p Simulation

Nonlinear case with $n = 100$

LASSO penalty parameter allowing nonlinear terms selected by 5-fold CV

	$p = 1$		$p = 5$		$p = 20$		$p = 50$	
	CV	MSE	CV	MSE	CV	MSE	CV	MSE
OLS	0.439	0.552	0.518	0.531	0.657	0.601	0.895	0.955
LASSO	0.420	0.577	0.416	0.577	0.424	0.577	0.379	0.591
LASSO.NL	0.253	0.407	0.038	0.016	0.169	0.072	0.339	0.339

Increasing p Simulation

Nonlinear case with $n = 200$

LASSO penalty parameter allowing nonlinear terms selected by 5-fold CV

	$p = 1$		$p = 5$		$p = 20$		$p = 50$	
	CV	MSE	CV	MSE	CV	MSE	CV	MSE
OLS	0.270	0.534	0.330	0.503	0.341	0.541	0.445	0.607
LASSO	0.286	0.535	0.321	0.503	0.307	0.526	0.331	0.524
LASSO.NL	0.230	0.369	0.015	0.017	0.031	0.032	0.086	0.060

For $p = 100$ and $p = 200$, computation and storage of matrices is a headache even in these tiny examples

Some options:

- Consider *ex ante* screening to eliminate very unlikely candidates before analysis (e.g. Fan and Lv (2008))
- Consider a different set of tools that don't require precomputation of large candidate design matrix

8. Comments on Penalized Estimation

Parting Thoughts

Intro to HD Estimation

Introduction

Traditional Nonpara-metrics

HDLM

Penalized Estimation

λ - CV

λ - Theory

Penalized Estimation Examples

Comments on Penalized Estimation

Some remarks on penalized estimation/subset selection:

- Penalized/selection estimators attractive for a variety of reasons
  - Produce results interpretable within familiar modeling frameworks
  - Can perform remarkably well in forecasting (and other contexts) with well-chosen variables and methods
  - Reasonably easy to build in functional restrictions (e.g. monotonicity, shape constraints, etc.)
  - Readily extended to the usual models (e.g. penalized logistic regression)
  - Many extensions, related methods (e.g. fused LASSO and smoothing splines for functional data, group LASSO, etc.)
  - Structure amenable to theoretical analysis
- Have some unappealing feature
  - Bookkeeping to deal with nonlinearities, interactions, etc. with even a few X's gets annoying
  - Need to construct all the relevant terms - memory/computation intensive
  - Not automatic - e.g. if you didn't think of the interaction and include it in the set of candidate variables, you won't find it (other methods - e.g. trees and random forests do this)