S&DS 365 / 565

Data Mining and Machine Learning

Regularization

Yale

Regularization helps control (regulate) parameters.

help robustness, prediction accuracy, often l_2 norm

General form

param you identifed ↓ any data ↓ $\{(x_i, y_i)$ supervised learning; (x_i) no label unsupervised learning

$$\widehat{\beta} \in \arg\min_{\beta} \underbrace{\mathcal{L}(\beta, \{Z_i\}_{i=1}^n)}_{\text{Loss Function}} + \lambda \underbrace{R(\beta)}_{\text{Regularizer}}$$

- Loss Function measures fit to data $\{Z_i\}$ (the param β you identifed)
- Regularizer controls parameters, fits our prior beliefs of the data (complexity of)

  Later talk about Bayes, priors fit this general form too
- Trade-off Loss and Regularization via regularization parameter λ

  λ control how much you emphasize on regularizer $\begin{cases} \lambda = 0 & \widehat{\beta} \in \arg\min \text{Loss} \\ \lambda \uparrow & \widehat{\beta} \in \arg\min \text{ more } R(\beta) \end{cases}$

  λ is a hyper param for your problem, often use cross validation to pick optimal λ

Choices for regularizer: ℓ_p norms

convex for $p \geq 1$

$\beta \in R^d = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_j \\ \vdots \\ \beta_d \end{bmatrix}$

$$\|\beta\|_p = \left(\sum_{j=1}^{d} |\beta_{(j)}|^p \right)^{\frac{1}{p}}$$

Focus on l_2 & l_1 in discussion

ℓ_2 norm squared

$$\frac{1}{2} \|\beta\|_2^2$$

ℓ_1 norm

$$\|\beta\|_1 = \sum_{j=1}^{d} |\beta_{(j)}|$$

Choices for losses : least squares, logistic, any GLM, etc...

any loss function

① Specific example: Ridge Regression

$$\widehat{\beta} \in \arg\min_{\beta} \frac{1}{2n}\|X\beta - y\|_2^2 + \frac{\lambda}{2}\|\beta\|_2^2$$

least square regression — regularizer l_2-norm

Result is that the coefficients get *shrunken* towards 0 and standard errors of coefficients are much lower. Regularization often called **shrinkage** → $\lambda\uparrow$ param are shrinking as $\lambda\uparrow$

② Specific example: Lasso: combines shrinkage and variable selection.

(Least absolute shrinkage and selection operator)

(feature / attribute selection)

$$\widehat{\beta} \in \arg\min_{\beta} \frac{1}{2n}\|X\beta - y\|_2^2 + \lambda\|\beta\|_1$$

l_1-norm

l_1-norm is selecting variables, it set some coefficent of β to be 0

good for interpretability: this 5 params are truly needed for prediction

avoid overfitting to noise if # data << # features

Technically this is Lasso

original version of Lasso, more constrained

$$\widehat{\beta} \in \arg\min_{\beta} \frac{1}{2n}\|X\beta - y\|_2^2 \quad \text{such that } \|\beta\|_1 \le R$$

because convexity → one solver called Laus will give solution path for all choices of R — it will give corresponding $\hat{\beta}$, λ or R

But they are equivalent with respective choices of λ and R.

use cross validation to pick optimal λ and R

eg. 500 patients (n) vs. 30,000 genes

Ridge regression

Ridge regression has a closed form solution

$$\widehat{\beta}^{(\lambda)} \in \arg\min_{\beta} \frac{1}{2n}\|X\beta - y\|_2^2 + \frac{\lambda}{2}\|\beta\|_2^2$$

Closed-form solution: HW! 5

similar to linear regression: take gradient, set to 0

General loss: gradient descent (neural net people call it weight decay)

→ no closed-form solution, but can use

$$\widehat{\beta} \in \arg\min_{\beta} \underbrace{\mathcal{L}(\beta, \{Z_i\}_{i=1}^n)}_{\text{Loss Function}} + \lambda \underbrace{R(\beta)}_{\text{Regularizer}}$$

Take $R(\beta) = \frac{1}{2}\|\beta\|_2^2$ ℓ_2-norm (penalty)

$$\beta_{k+1} = (1 - \eta_k\lambda)\beta_k - \eta_k \nabla \mathcal{L}(\beta_k, \{Z_i\}_{i=1}^n)$$

$= \beta_k - \eta_k \nabla_\beta(\,) = \beta_k - \eta_k(\nabla_\beta(\beta_k) + \lambda\beta_k)$

λ for ridge regression

- Each λ results in a different set of coefficients $\widehat{\beta}^{(\lambda)}$.
- λ controls the amount of shrinkage (bias-variance tradeoff).
  - λ near 0 implies $\widehat{\beta}^{(\lambda)}$ is near least squares estimate $\widehat{\beta}$.
  - as $\lambda \to \infty$, slopes $\widehat{\beta}_j^{(\lambda)} \to 0$ for $j = 1, \ldots, p$.
  - Use cross-validation to select λ.

Bayes use Prior to pick λ

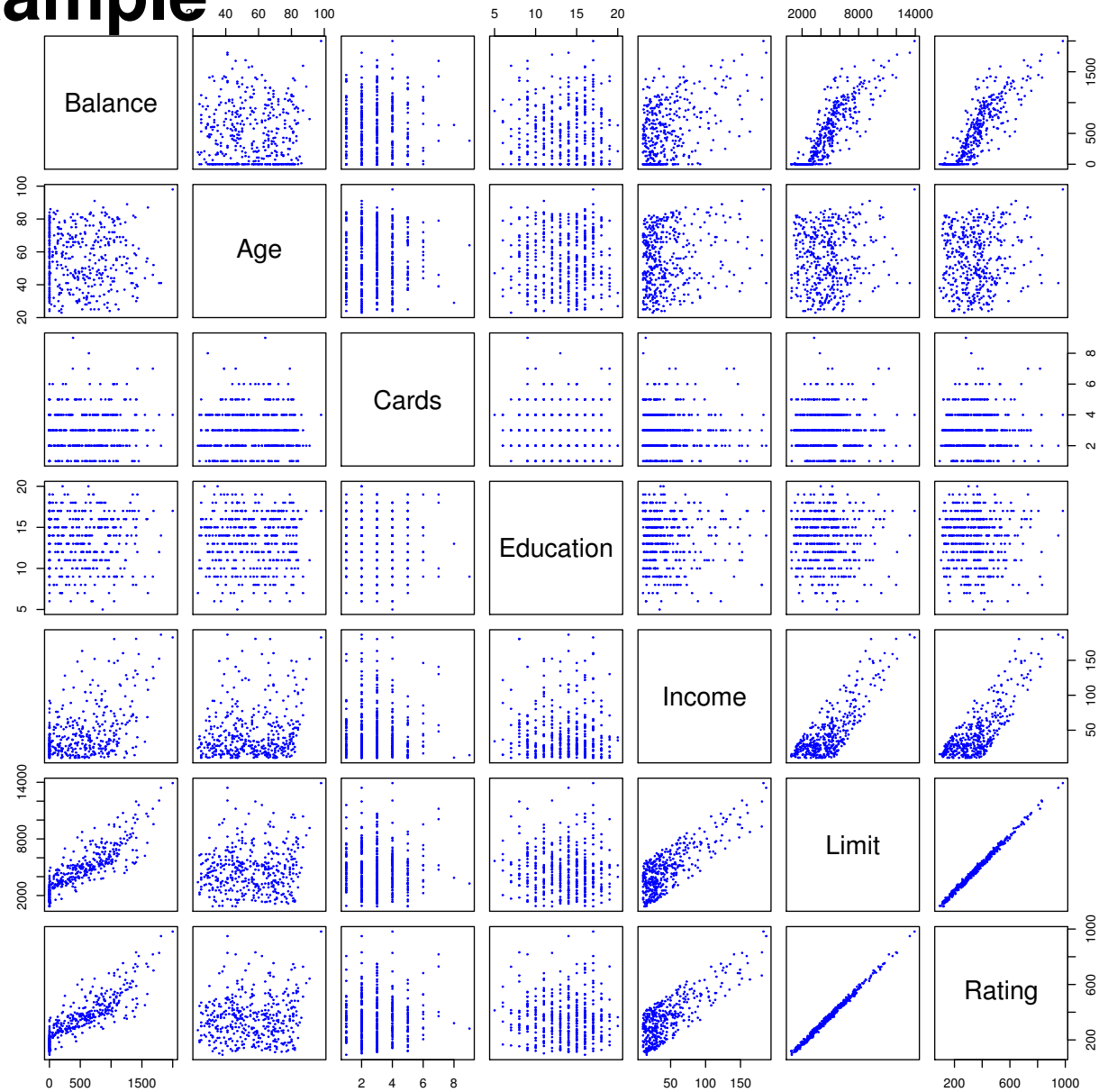
Prior is similar to cross-validation

$\mathrm{Var}(\hat{\beta}^{(\lambda)})$ vs $\mathrm{Bias}(\hat{\beta}^{(\lambda)})$ Trade-off

as $\lambda \to \infty$ we are $\mathrm{Bias}\ \hat{\beta}^{(\lambda)}$ to be 0, so $\mathrm{Var}(\hat{\beta}^{(\lambda)})$ is small

Credit example

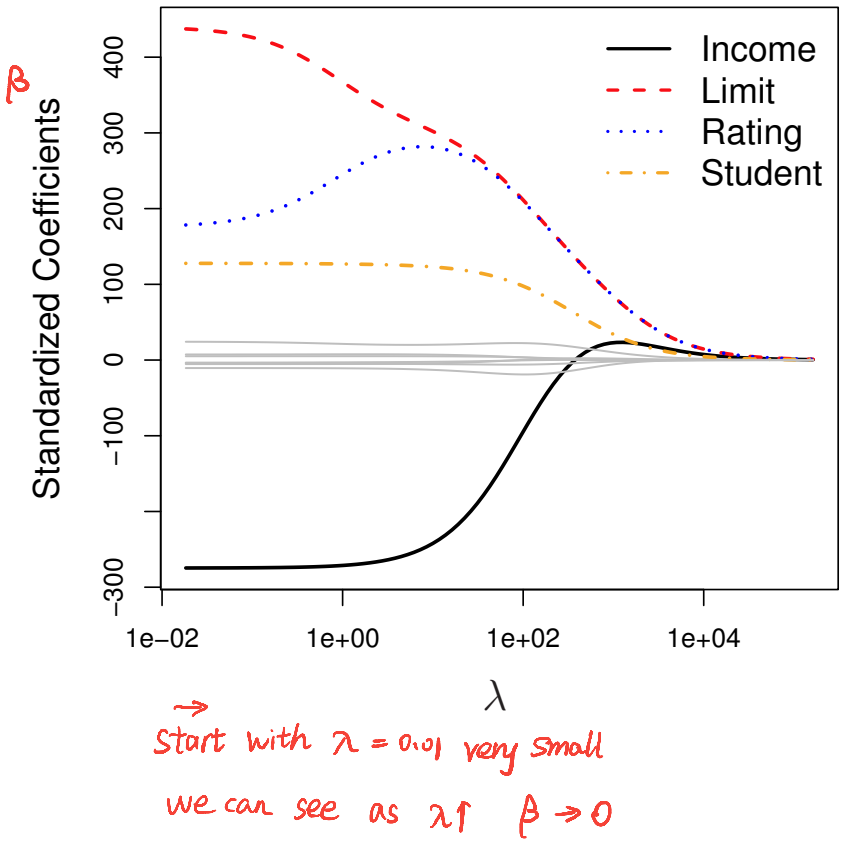
信用卡



Credit example

predict How much credit a client should get

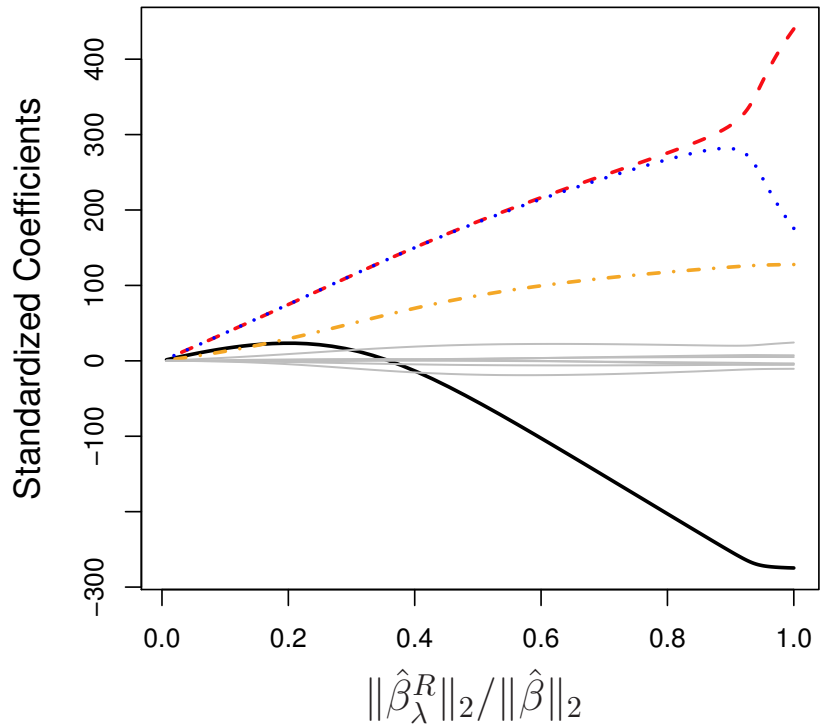
Ridge Regression solution



β

→ Start with $\lambda = 0.01$ very small

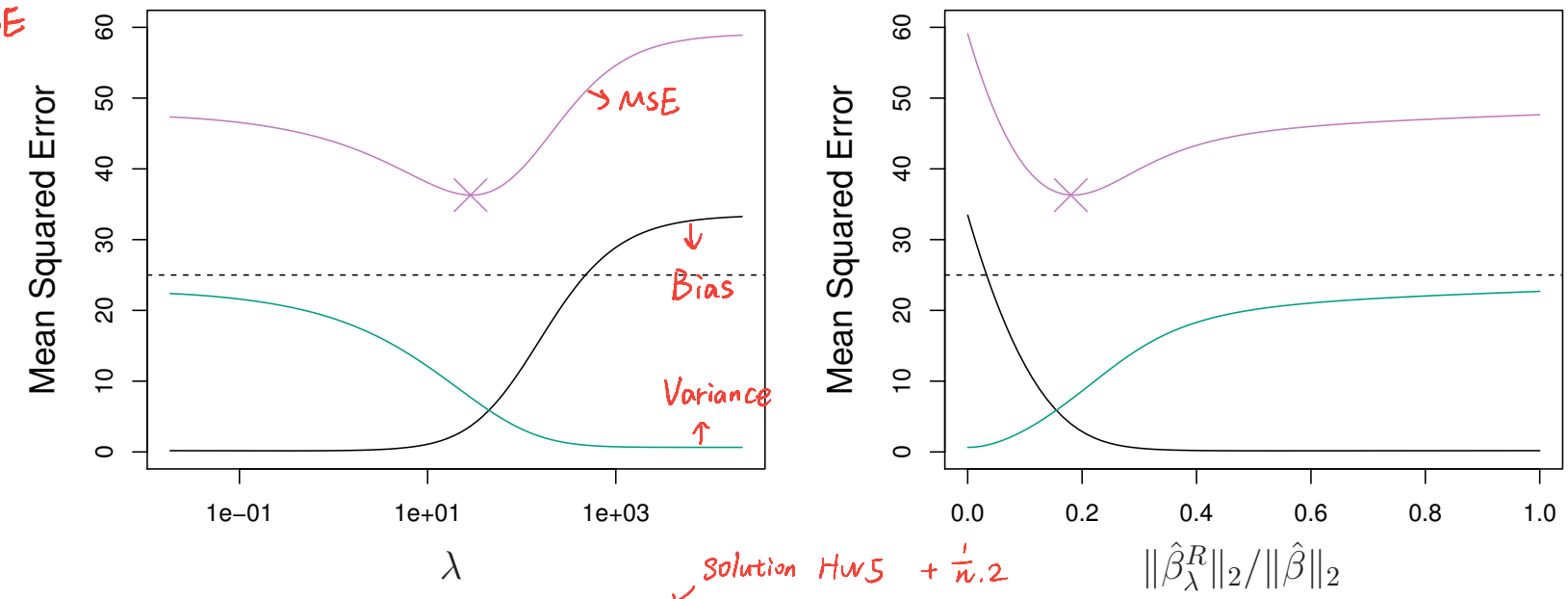
we can see as $\lambda \uparrow$ $\beta \to 0$



How does ridge regression work?

Consider model $y = X\beta + \epsilon$ where $\epsilon_{(i)}$ zero-mean/variance σ^2.

MSE vs λ → Bias-Variance Tradeoff



- $E(\widehat{\beta}^{(\lambda)}) = \beta - \lambda(X^TX + \lambda I)^{-1}\beta$. Not unbiased! (Solution Hw5 + $\frac{1}{n}$.2) if $\lambda=0$ $E=\beta$ unbiased
- $Var(\widehat{\beta}^{(\lambda)}) = \sigma^2(X^TX + \lambda I_p)^{-1}X^TX(X^TX + \lambda I_p)^{-1} \leq Var(\widehat{\beta})$. $\lambda\uparrow$ Bias $\uparrow$ / $\lambda\uparrow$ Var $\downarrow$

Scale-equivariant

For Linear Regression. Standardization doesn't matter but help numerically for large X calculation

Least squares coefficients are *scale-equivariant*.

Multiplying values of a predictor by a constant c simply results in multiplying its associated $\widehat{\beta}_j$ by $1/c$.

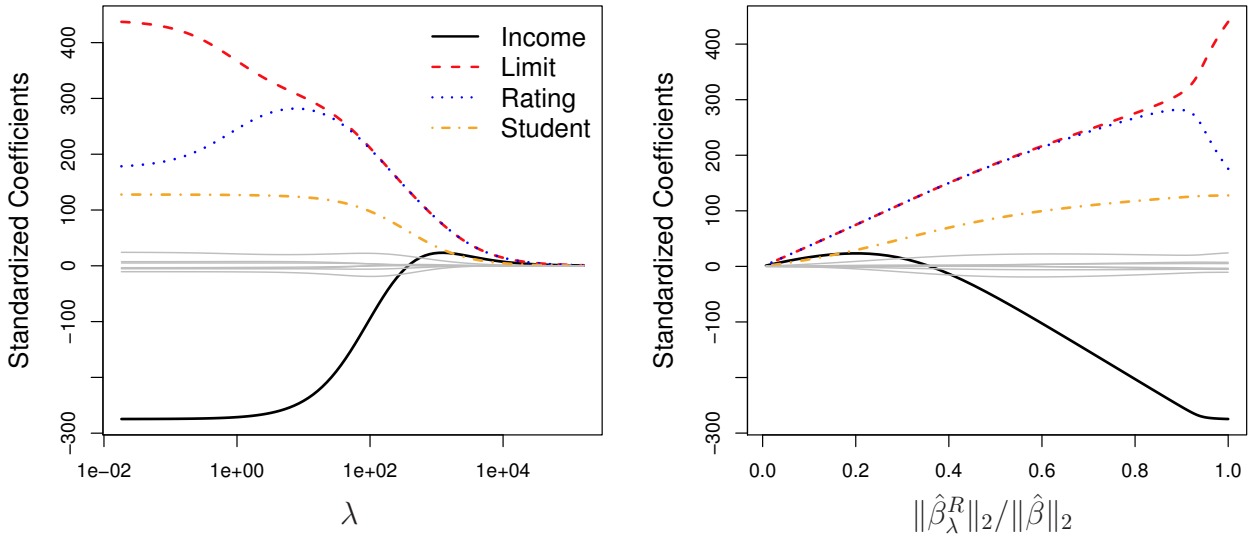
What about for ridge regression? Not the case, Standarization is important eg. for KNN

$$(y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta.$$

Because the shrinkage penalty depends on the magnitude of the β_j, scales now matter! So it's important we standardize the predictors.

continuous: to have mean=0, sd=1
categorical x = -1, +1

Credit example again



Note that for any choice of λ, $|\beta_j^{(\lambda)}| > 0$ for $j = 1, \ldots, p$. This is fine for prediction, but not so good for interpretation.

causality : A set of features can have accurate prediction of y don't mean they couse y

Example

$$Y = 3X_1 + \cdots + 3X_5 + 0X_6 + \cdots + 0X_{1000} + \epsilon$$

$n = 100,\ p = 1{,}000.$

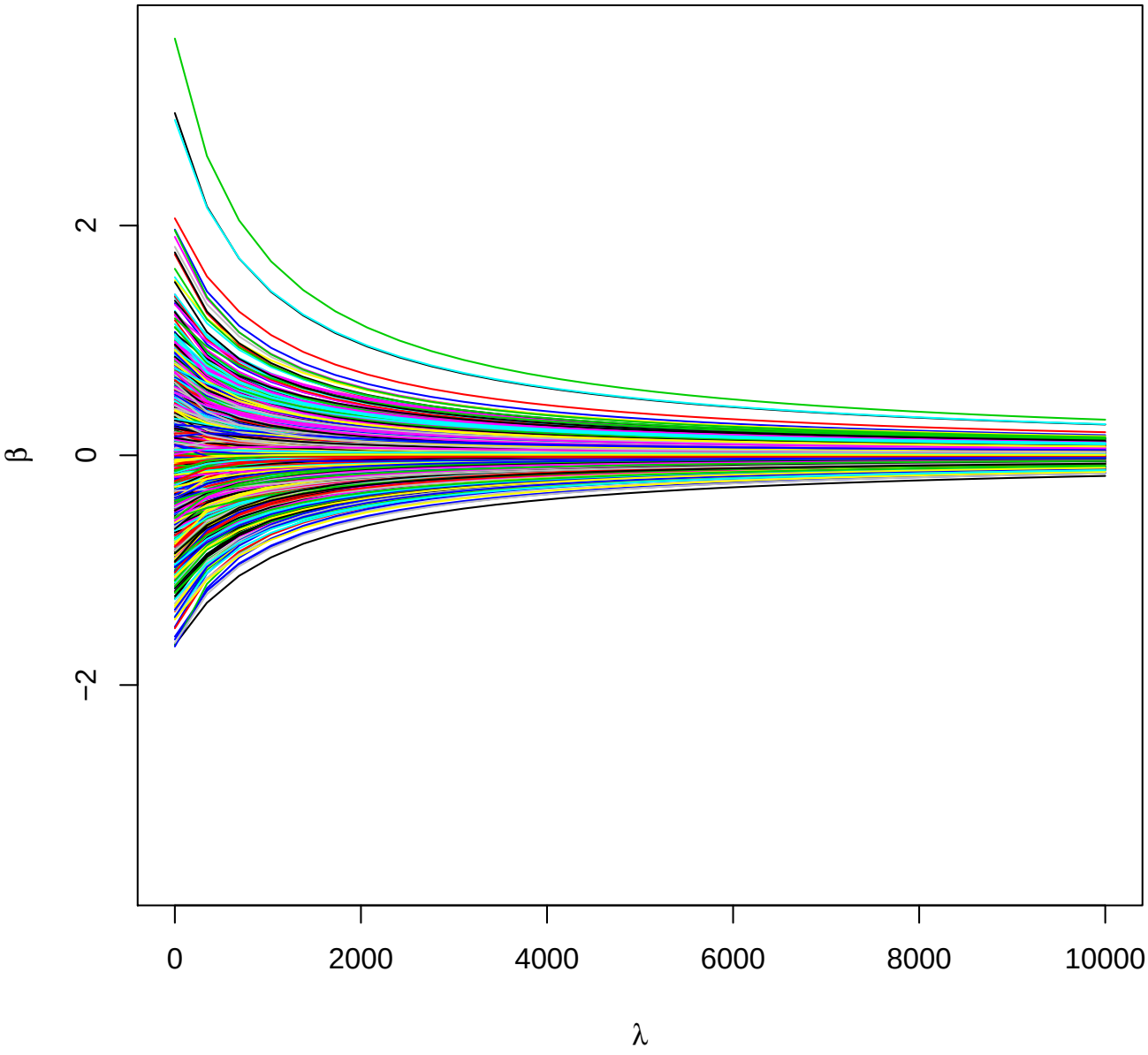
So there are 1000 covariates but only 5 are relevant.

What does ridge regression do in this case?

A bad Predictor

instead we use Lasso

Ridge Regularization Paths



Lasso

Lasso (Least absolute shrinkage and selection operator) uses ℓ_1 penalty:

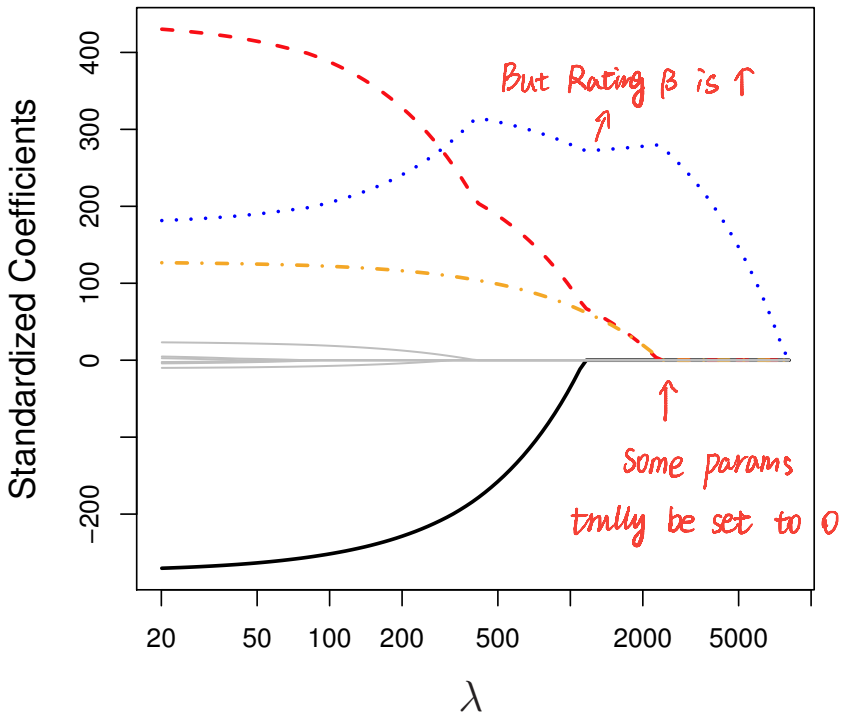
$$RSS(\beta) + \underbrace{\lambda \sum_{j=1}^{p} |\beta_j|}_{\text{shrinkage penalty}},$$

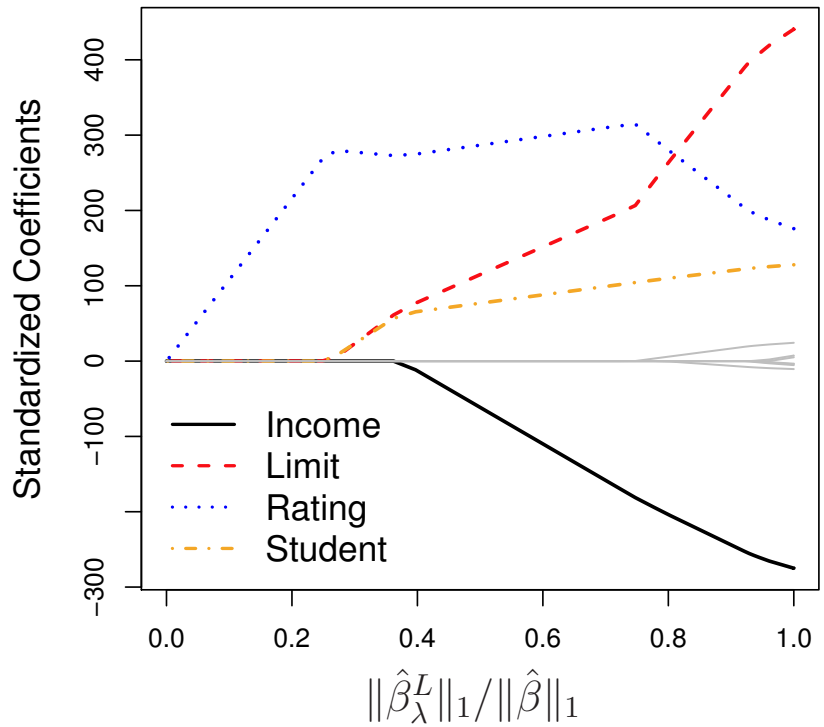
No closed form solution for $\widehat{\beta}^L$, but numerical optimization is "easy"

eg. ① Gradient Descent
② threshold Gradient Descent

This modified penalized RSS has the effect of forcing some $\widehat{\beta}_j^L$ to be exactly 0 for large λ.

Credit example (with lasso)





An alternate view of lasso and ridge

Help understand geometric difference between ridge and Lasso why Lasso encourage solutions to be sparse

An equivalent formulation of the ridge regression:

constrained optimization

Residual Sum of Squares

$$RSS(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

constrained version

subject to the constraint $\sum_{j=1}^{p} \beta_j^2 \leq s$ for some $s > 0$.

penalized version $\sum_{j=1}^{p} |\beta_j|$

Likewise, an equivalent formulation of lasso is to minimize $RSS(\beta)$ s.t. $\sum_{j=1}^{p} |\beta_j| \leq s$ for some $s > 0$.

such that

In each case, we can show that there's a 1-1 correspondence between s and λ.

Geometry of two methods

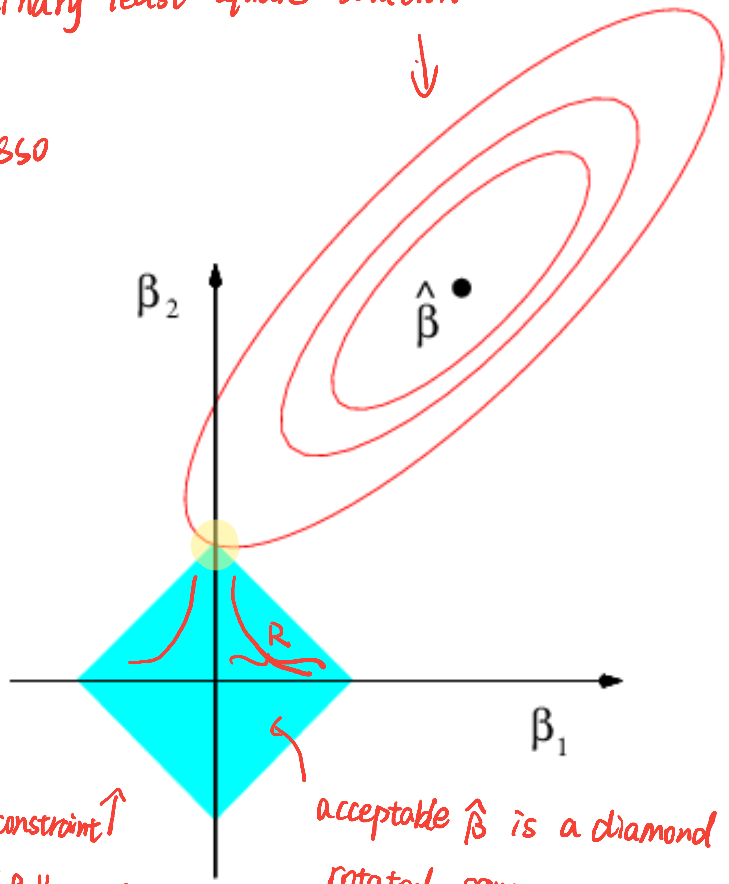
③ l_p norm $p \to \infty$
encourage $|\beta_1| = |\beta_2|$

assume $\beta \in R^2$

○ ellipse: least square objects

$\hat{\beta}$: ordinary least square solution with no constraint

Lasso



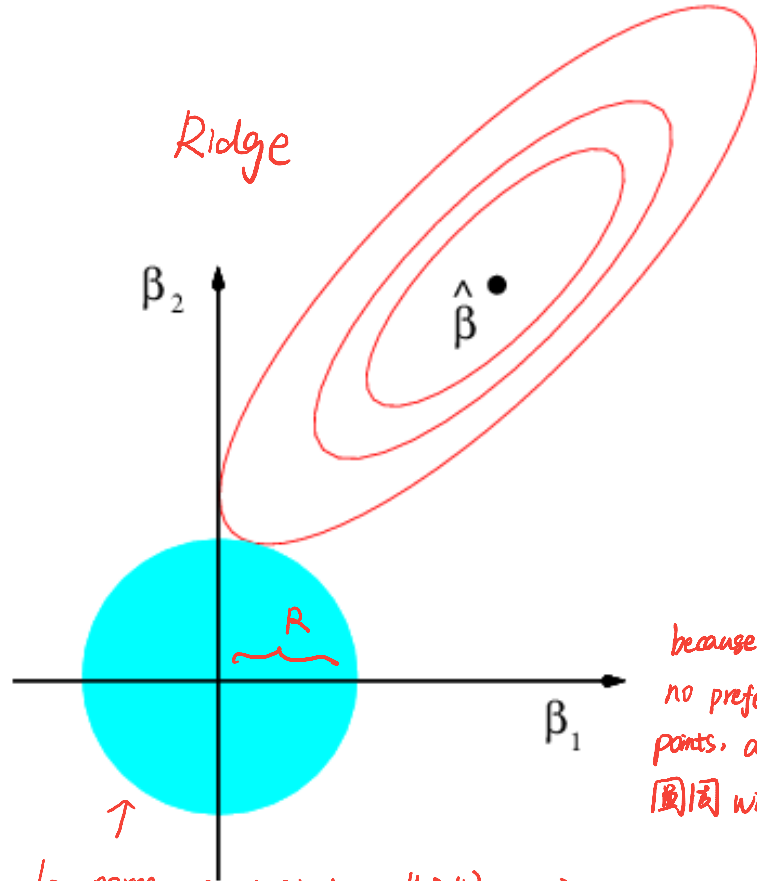
l_1-norm constraint ↑

$\|\beta\|_1 \leq R$

acceptable $\hat{\beta}$ is a diamond

rotated square as R ↑

Ridge



↑

l_2-norm constraint $\|\beta\|_2^2 \leq R^2$

$\beta_1^2 + \beta_2^2 \leq R^2$

because all $\beta \to 0$

no preference for touch points, any point along 圆|球 will touch

Find β in the blue Set which has smallest ellipse

Because ◇ is pointy, the first place ◇ and ○ will touch (some β_j to 0, some $\beta_j \to 0$) maybe one of points

not the flat region, thus solution of Lasso is sparse

Transforming predictors

Be careful when

How does transformation effact Lasso or Ridge ?

- Important to scale for both
- But lasso is not orthogonally invariant ($(X, \beta) \equiv (X\Gamma, \Gamma^T\beta)$)
- Destroys the sparsity pattern

orthogonal transformation

$\Gamma^T\beta$ for a general Γ (gamma)

正交不变

orthogonal transform row of matrix X

But for Ridge, no effact because l2-norm of a vector is invariant for orthogonal transformation

Summary of shrinkage methods

- General ℓ_q regularization criterion:

$$\widehat{\beta} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\}$$

  What would ℓ_0 be doing?

- Idea can apply to many methods, including classification. For example, logistic regression, modify negative log-likelihood problem to:

$$\frac{1}{n} \sum_{i=1}^{n} \left[\log(1 + \exp(-x_i^T \beta)) - y_i x_i^T \beta \right] + \frac{\lambda}{2} \sum_{j=1}^{p} \beta_j^2$$

- Compared to ℓ_1 (lasso), ℓ_2 (ridge) produces models that are less interpretable because there will be a large number of coefficients.

Comparing lasso and ridge

An equivalent formulation of the ridge regression optimization problem is that we want to minimize:

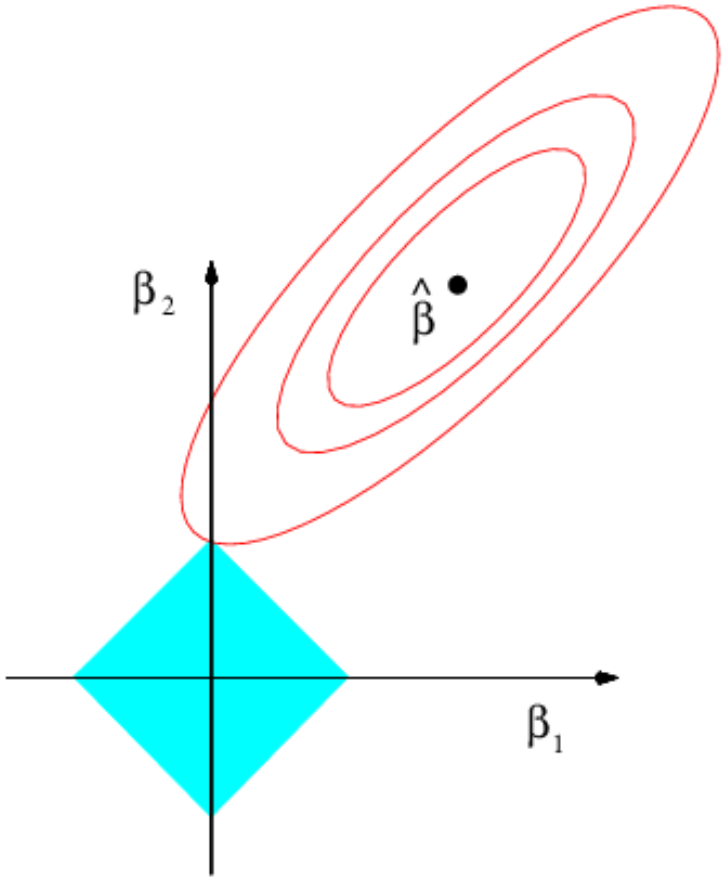
$$RSS(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

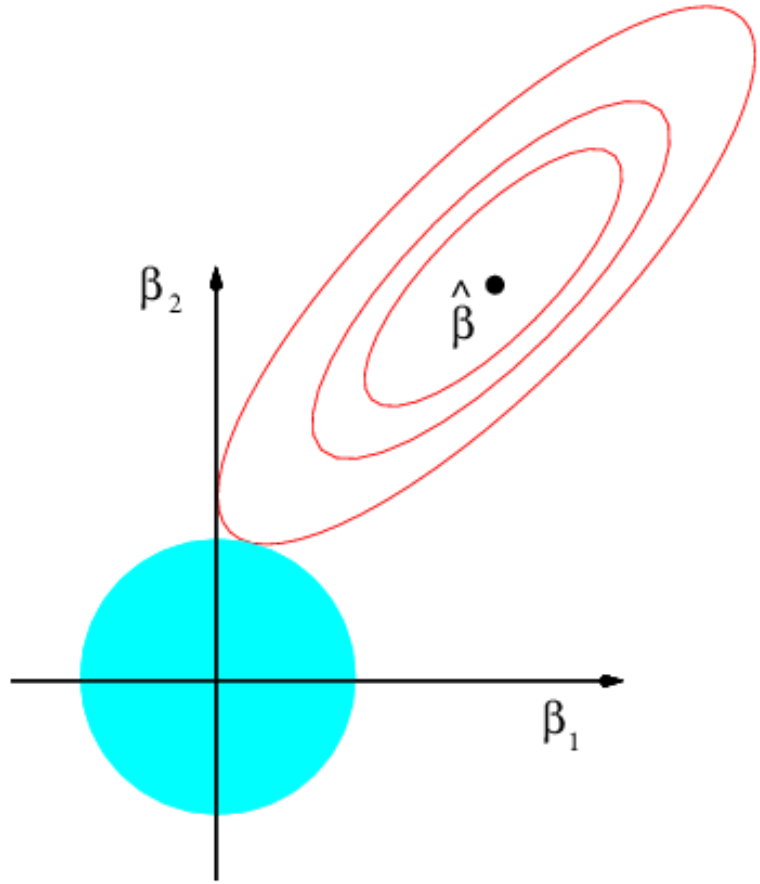
subject to the constraint $\sum_{j=1}^{p} \beta_j^2 \leq s$ for some $s > 0$.

Likewise, an equivalent formulation of lasso is to minimize $RSS(\beta)$ s.t. $\sum_{j=1}^{p} |\beta_j| \leq s$ for some $s > 0$.

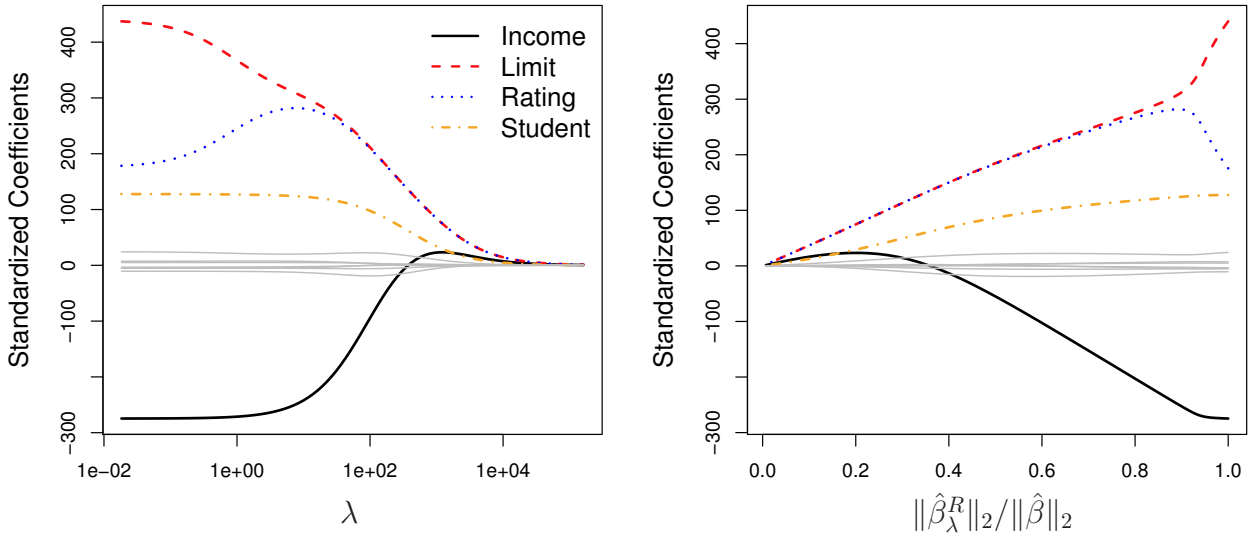
In each case, we can show that there's a 1-1 correspondence between s and λ.

Comparing lasso and ridge

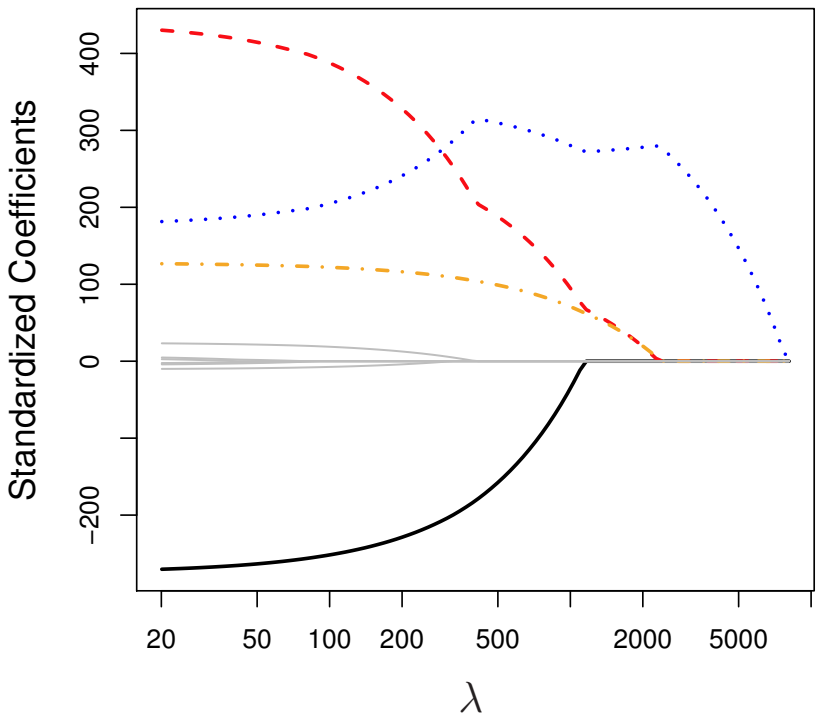


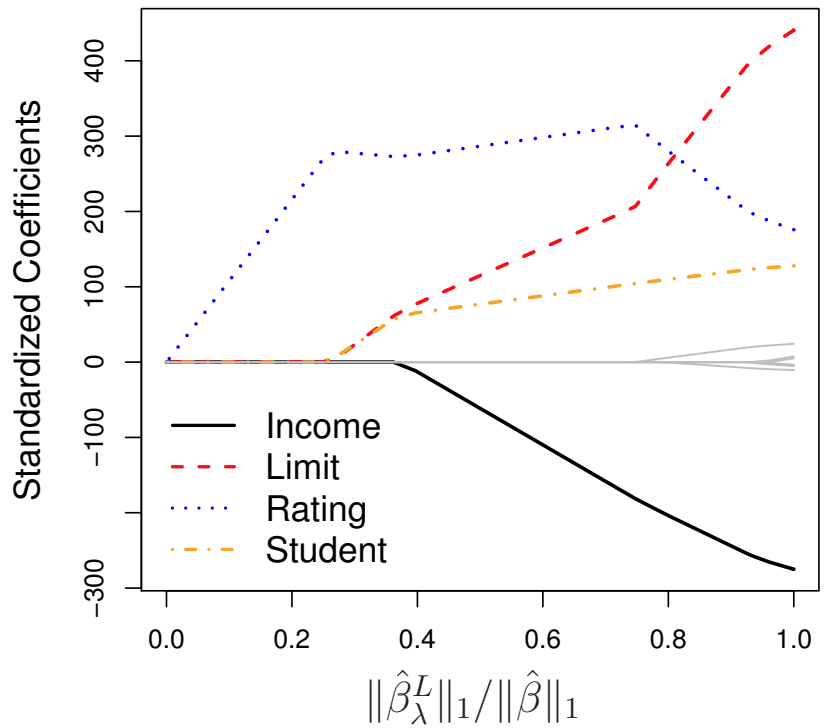


Credit example again



Credit example (with lasso)





How to choose λ

- Cross-validation is straight-forward:
  - Set up a grid of λ values.
  - Estimate test error using 10-fold cross-validation for each λ.
  - Choose λ with lowest cross-validation error.

Hw5 Boosting test error 5%

Comparing lasso and ridge

Solution to HW5

Consider the special case where we have orthonormal predictors (and no intercept). That is to say, that $X^T X = I$. Then, the lasso coefficients are (Additive Shrinkage) ↓ identity ↙ minus

where $\widehat{\beta}_j$ is least square solution

$$\widehat{\beta}_j^L = \begin{cases} \widehat{\beta}_j - \lambda & \text{if } \widehat{\beta}_j > \lambda \\ 0 & \text{if } |\widehat{\beta}_j| \leq \lambda \\ \widehat{\beta}_j + \lambda & \text{if } \widehat{\beta}_j < -\lambda \end{cases}.$$

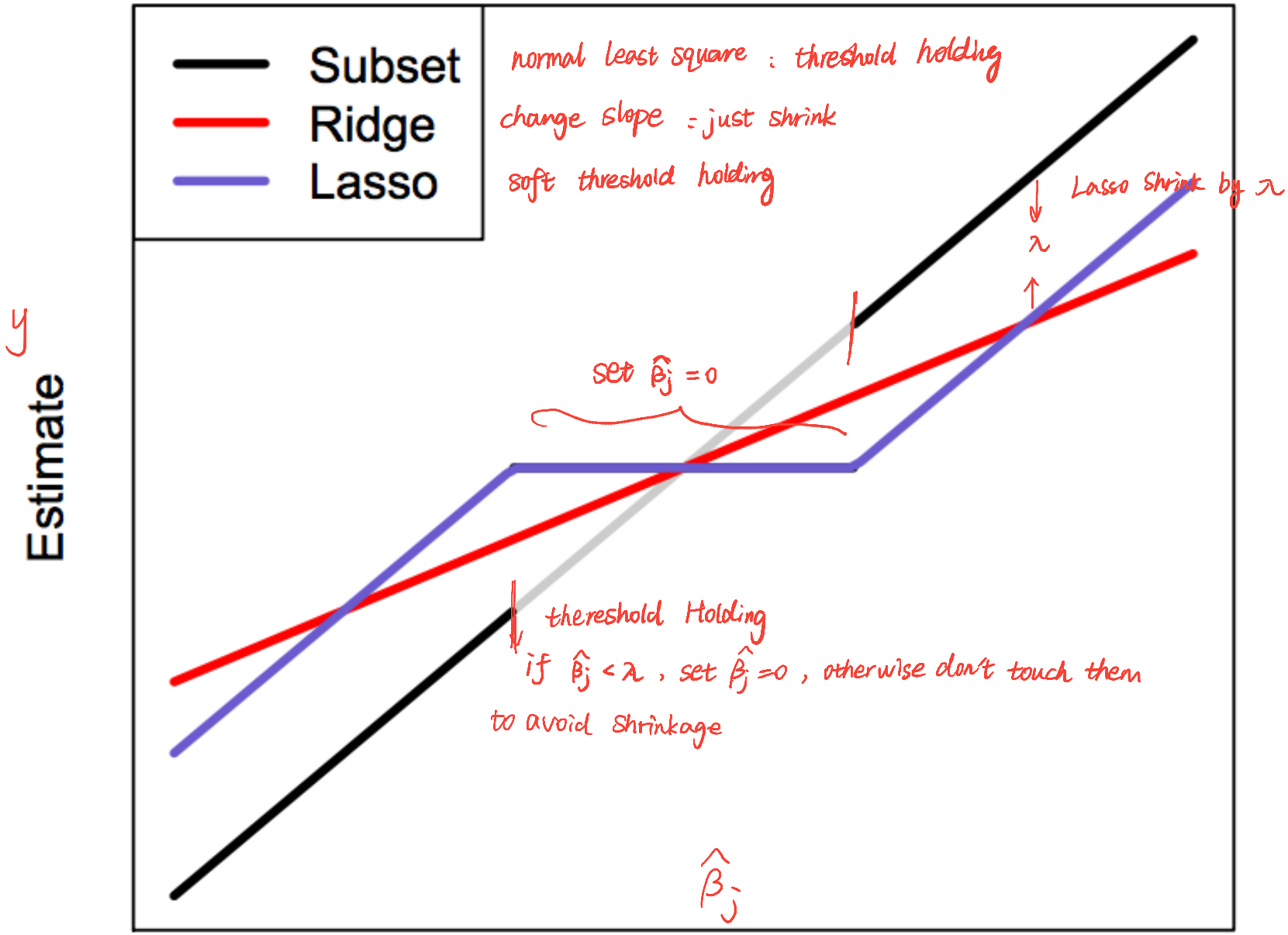
↑ Add

And the ridge coefficients are: (multiplicative Shrinkage)

$$\widehat{\beta}_j^{(\lambda)} = \frac{\widehat{\beta}_j}{1+\lambda}.$$

Problem with Lasso: if true β_j is very large, $\widehat{\beta}_j^L = \widehat{\beta}_j - \lambda$ might penalty too much by λ. eg. $\widehat{\beta}_j = 10 \to \widehat{\beta}_j^L = 1$ if you know $\widehat{\beta}_j = 10$, you should be very confident that $\widehat{\beta}_j = 10$. So we also use Bayes method to do variable selection to take this phenomenun into account: if you are very confident with $\widehat{\beta}_j$, you won't shrink it by λ, similar to Ridge

Comparing lasso and ridge

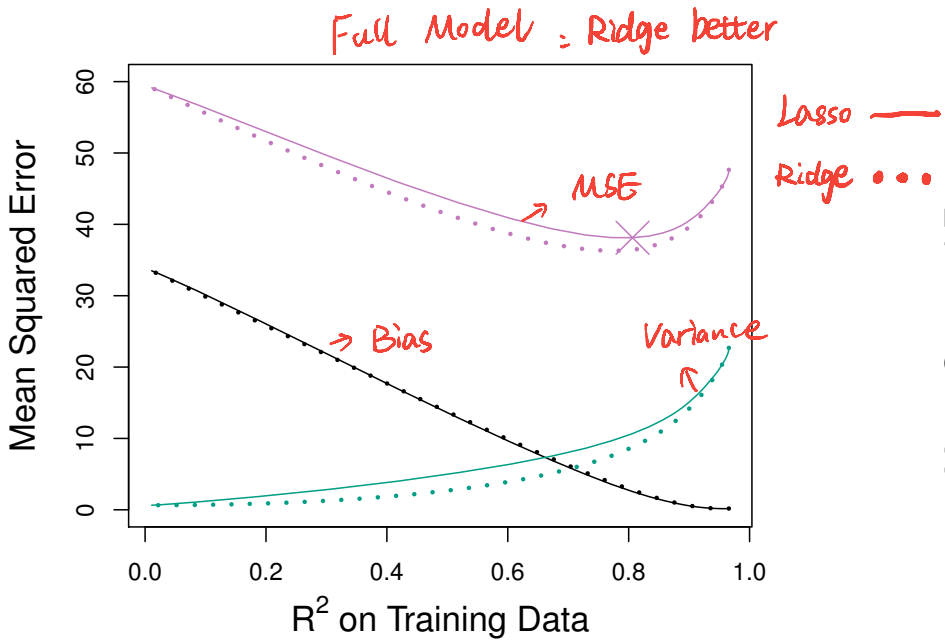


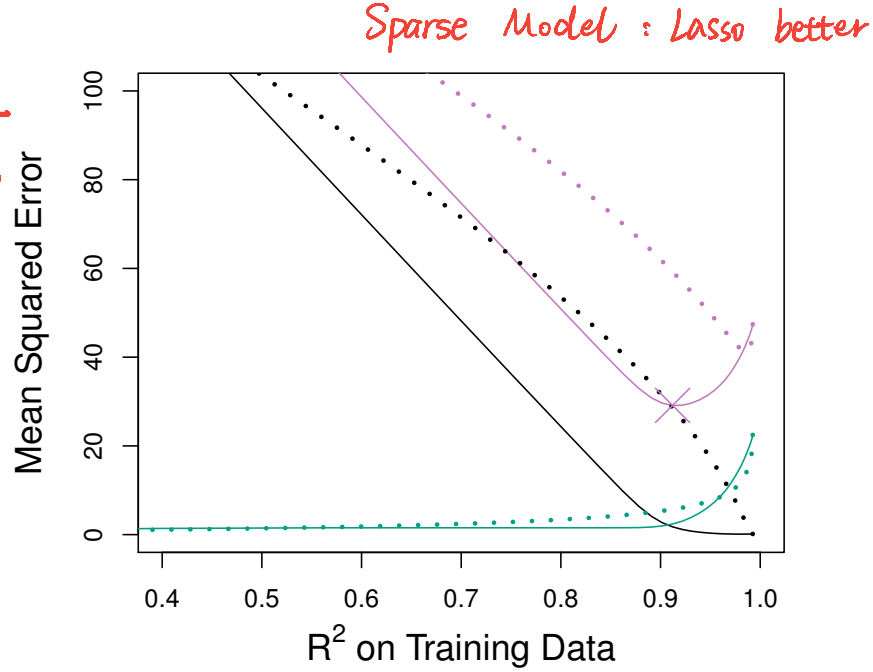
Comparing lasso and ridge

- Lasso yields simpler (sparser) models.
- Specifically, lasso can select at most n features.
- Lasso and ridge deal with *multicollinearity* differently.
- Actual performance (MSE) depends on underlying truth.

Comparing lasso and ridge

Simulated data: 45 predictors.





- Lasso (solid) vs. ridge (dotted) lines
- Bias^2 (black), variance (green), test MSE (purple)
- Generating model:
  - left: $y = \beta_0 + \beta_1 x_1 + \cdots + \beta_{45} x_{45} + \epsilon$ (full model)
  - right: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$ (sparse model (reduced)) only use 2 predictors

Often using an ℓ_2 penalty encourages better prediction performance.
For that reason ℓ_1 and ℓ_2 penalties are combined to form **elastic net**

Lasso Ridge 岭回归 弹性网络

2 hyper params λ, μ

$$\arg\min_{\beta} \mathcal{L}(\beta) + \lambda\|\beta\|_1 + \mu\|\beta\|_2^2$$

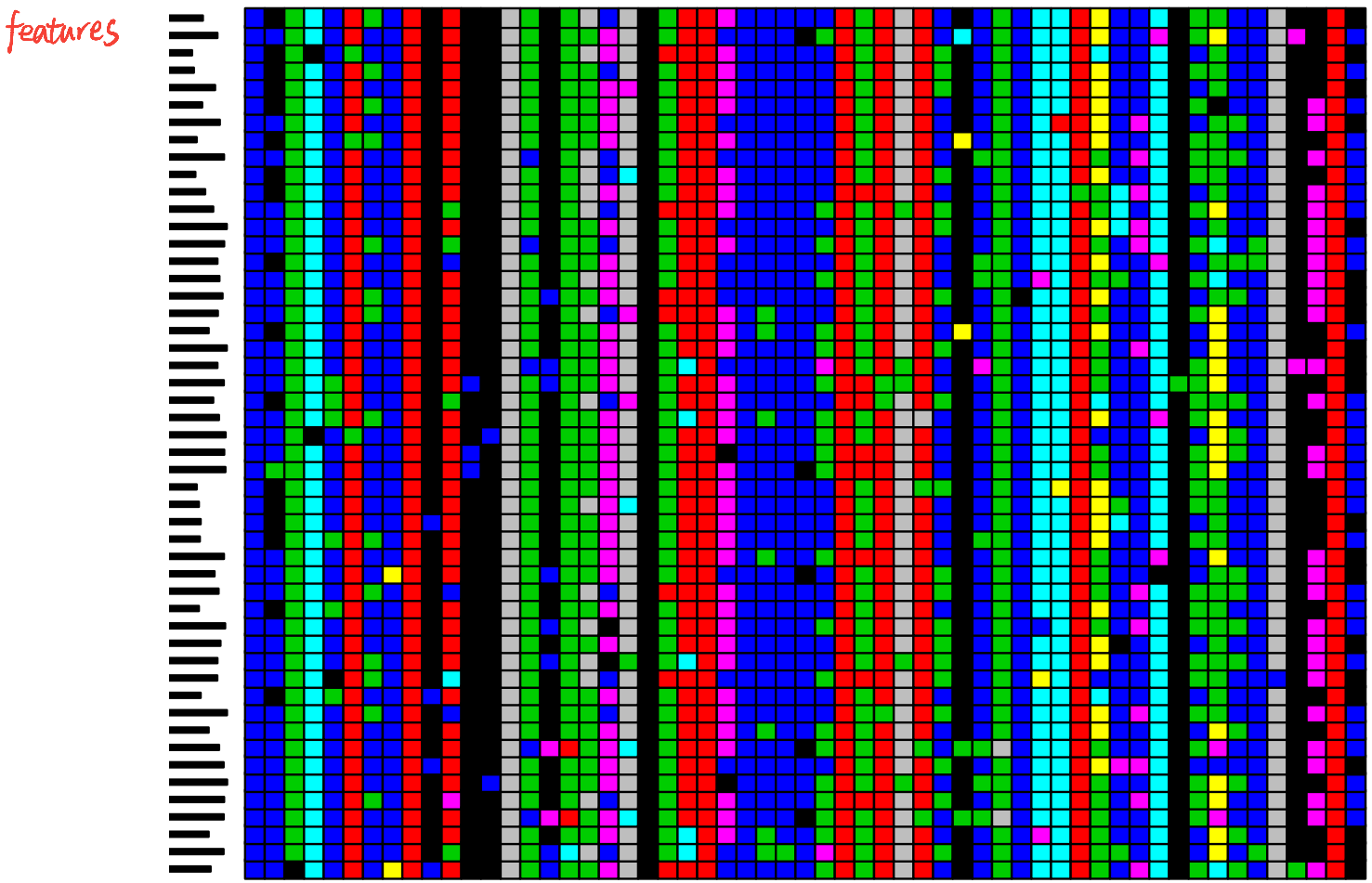
λ: l_1-norm; μ: l_2-norm

{ hyper params: params setted before training
{ params: β params obtained after training

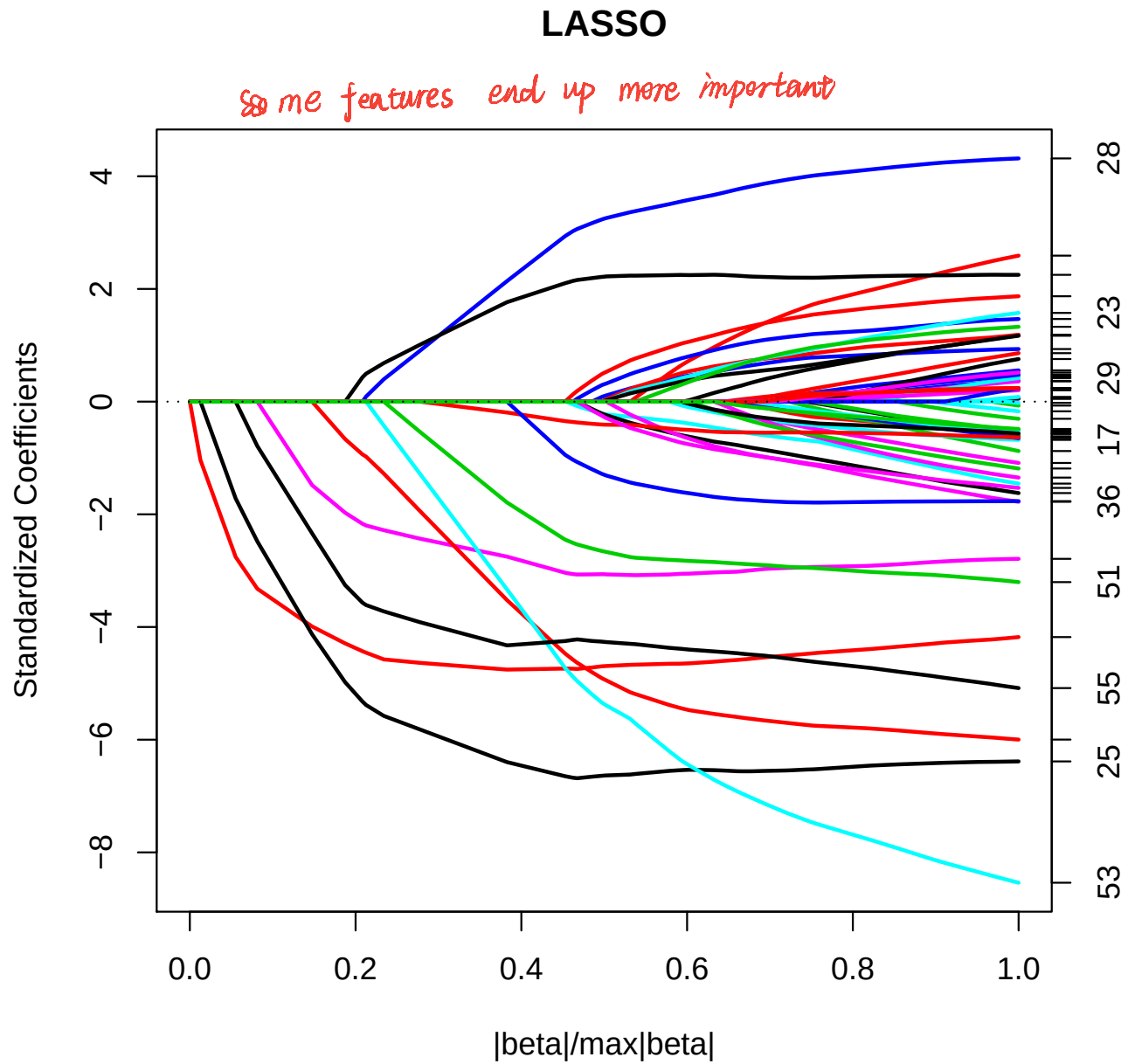
pick optimal λ and μ using cross-validation by testing combination of λ and μ

HIV example

- Y is resistance to HIV drug.
- X_j = amino acid in position j of the virus.
- $p = 99$, $n \approx 100$.



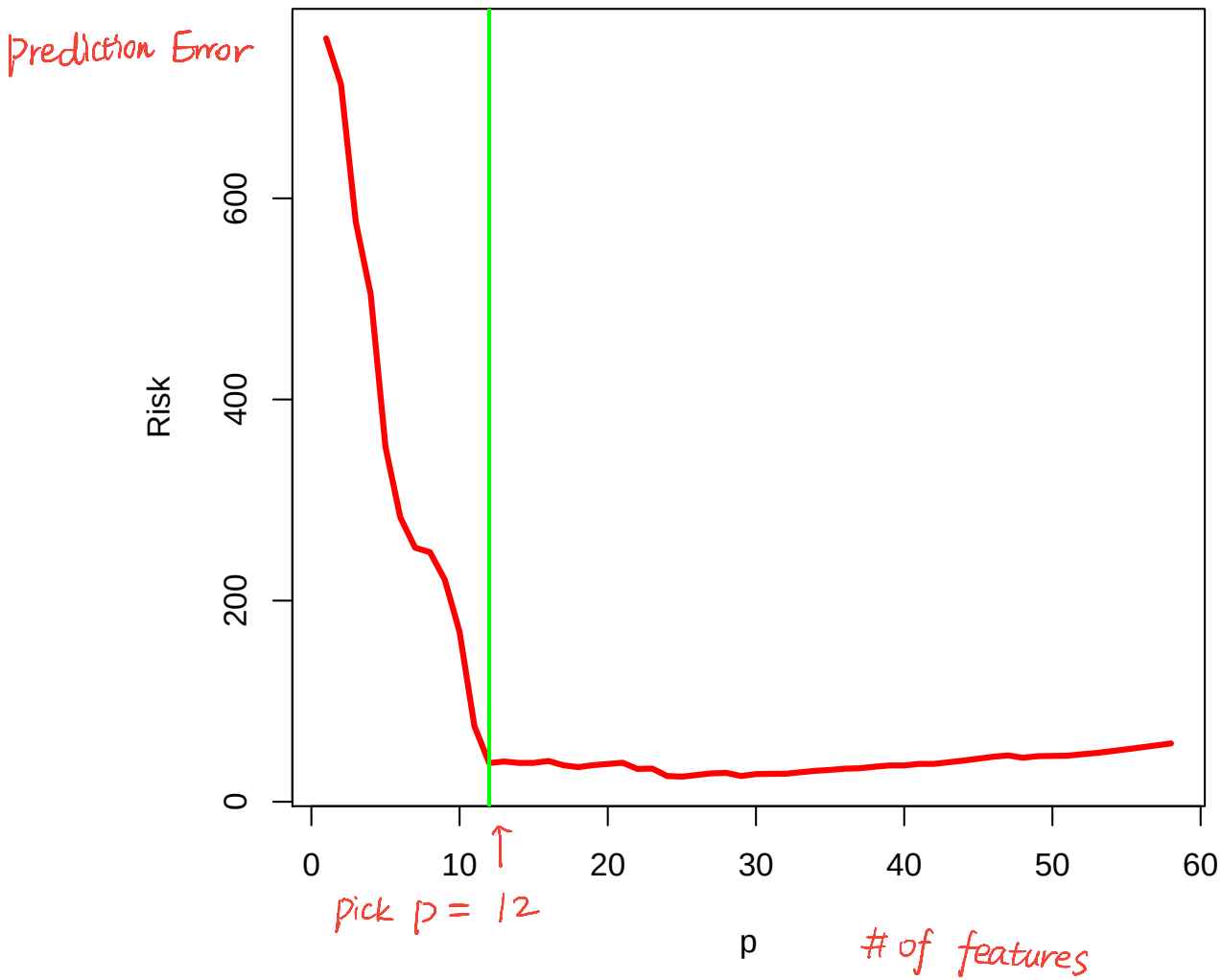
HIV example


LASSO
some features end up more important
Standardized Coefficients
4
2
0
−2
−4
−6
−8
28
23
29
17
36
51
55
25
53
0.0
0.2
0.4
0.6
0.8
1.0
|beta|/max|beta|

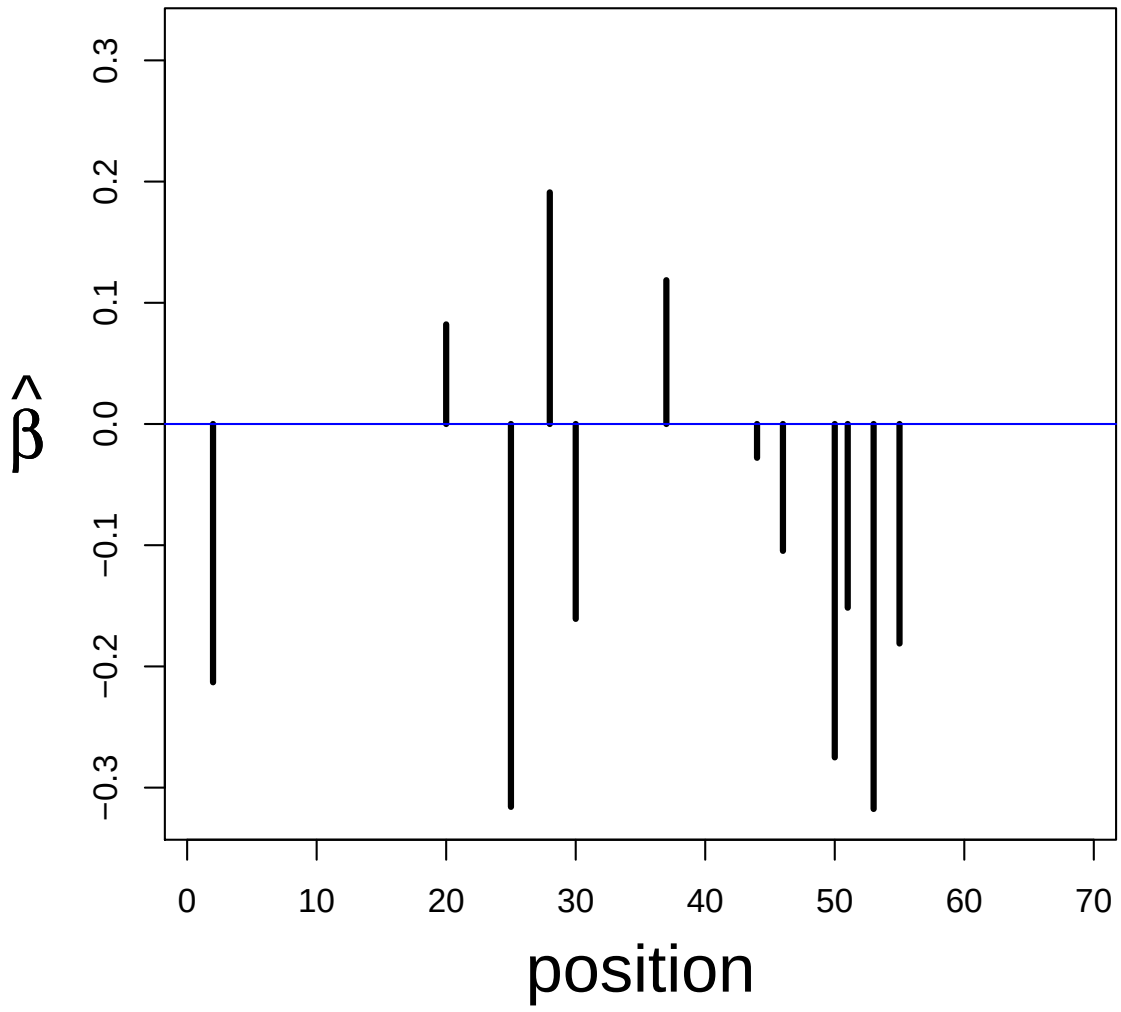
Selecting λ

We choose the sparsity level by estimating prediction error.

(# of features)



HIV example



Conclusion

- scale matters Standardization is important
- Ridge regression has closed form
- Penalizing by ℓ_1 norm leads to sparsity (and shrinkage)
- If the true model only involves a few predictors, lasso may be better than ridge

feature 变少 β变小

Q: When to use ridge is better than lasso ?

↓ l2 ↓ l1

- Lasso: help robustness
- Ridge: help robustness, help protect from noise, better prediction accuracy
- elastic net: combine both strengths