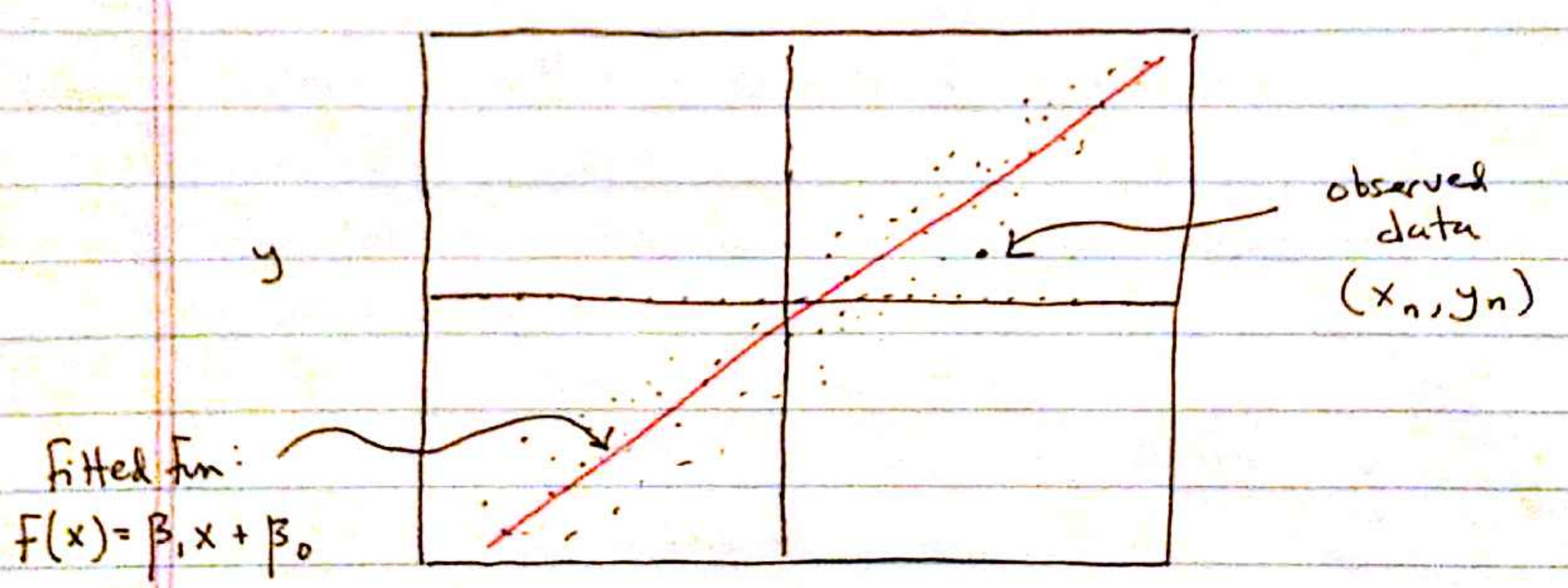
Linear regression is ~~the~~ a solution to the problem of predicting a real valued variable, called the <u>response</u>, from a set of inputs, called <u>covariates</u>.

In some communities, like ORFE, this is taught in depth. In others, like CS, you are meant to pick it up if needed.

The goal is to predict y from x with a linear fun:



Example:

- Given the stock price today, what is the price tomorrow?
- Given today's precipitation, what is the precipitation in a week?
- Given my mother's height, what is my shoe size?
- Others? Where have you seen linear regression?

Usually, we have multiple covariates, each representing a different <u>feature</u> of the data that might be predictive of the response.

$$X_n = \langle X_{n,1}, X_{n,2}, X_{n,3}, \dots, X_{n,p} \rangle$$

We will fit a linear function of these inputs

$$f(x) = \beta_0 + \sum_{i=1}^{p} \beta_i X_i$$

In general, $\beta^T x = 0$ is a hyperplane.

This set-up is less limiting than you might imagine:
Inputs can be

- any feature of the data
- transformations of original features, e.g.,

$$X_2 = \log X_1 \quad \text{or}$$

$$X_2 = \sqrt{X_1}$$

- a basis expansion, e.g., ← fits a polynomial rather than a line.

$$X_2 = X_1^2 \qquad X_3 = X_1^3$$

- indicator functions of qualitative inputs, e.g.,

$$1[\text{is the subject brown-haired}]$$

- <u>interactions</u> b/w inputs, e.g.,

$$X_3 = X_1 \cdot X_2$$

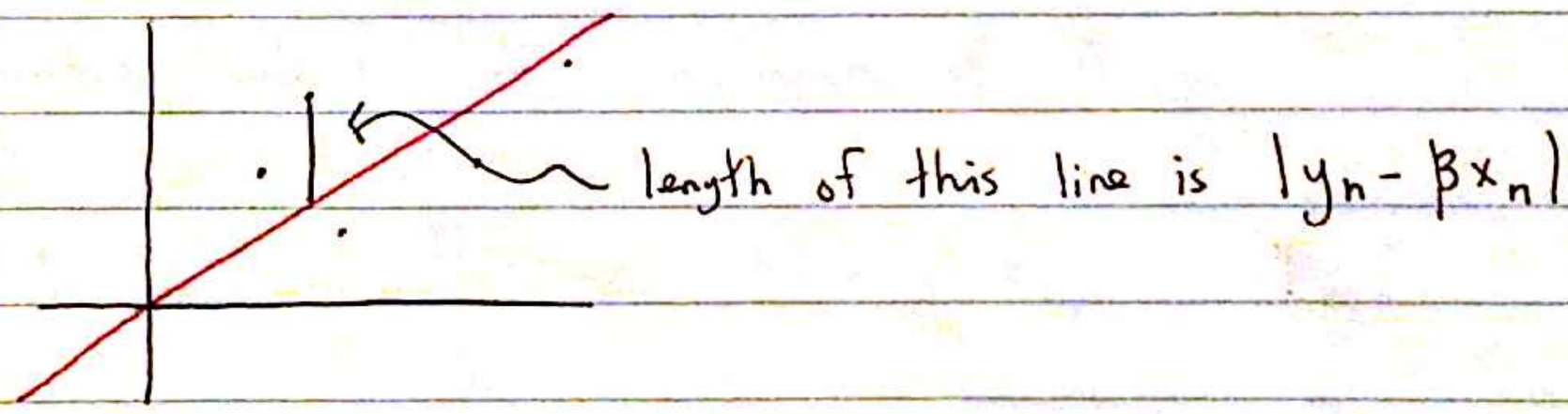
Its simplicity and flexibility make linear regression one of the most important and widely used statistical prediction methods.

Fitting a regression

Given data $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$, find the coefficient β that can predict y_{new} from x_{new}

Simplification: $\emptyset$ intercept, i.e., $\beta_0 = 0$
one input ($p = 1$)

Reasonable approach: minimize the sum of the squared Euclidean distance between each fitted response βx_n and true response y_n.



RSS: "residual sum of squares"

$$RSS(\beta) = \frac{1}{2}\sum_{n=1}^{N}(y_n - \beta x_n)^2 \quad \leftarrow \text{Objective as a function of } \beta.$$

$$\frac{d\,RSS(\beta)}{d\beta} = -\sum_{n=1}^{N}(y_n - \beta x_n)x_n \quad \leftarrow \text{Derivative}$$

$$\hat{\beta} = \frac{\sum_{n=1}^{N} y_n x_n}{\sum_{n=1}^{N} x_n^2} \quad \leftarrow \text{Optimal value}$$

$$\hat{y}_{new} = \hat{\beta} x_{new} \quad \leftarrow \text{Prediction. Note: } x_{new} \text{ is always assumed given.}$$

In general we have an intercept and multiple inputs:

$$y = \beta_0 + \sum_{i=1}^{p} \beta_i x$$

To simplify, set $\beta_{p+1} = \beta_0$ and $x_{p+1} \triangleq 1$.

$$y = \beta^T x$$

and

$$RSS(\beta) = \frac{1}{2} \sum_{n=1}^{N} (y_n - \beta^T x_n)^2$$

can go right to here

The derivative w/r/t one component of β is

$$\frac{\partial RSS(\beta)}{\partial \beta_i} = -\sum_{n=1}^{N} (y_n - \beta_i^T x_{n,i}) x_{n,i}$$

The gradient is

$$\nabla_\beta RSS(\beta) = -\sum_{n=1}^{N} (y_n - \beta^T x_n) x_n$$

[Here you might stop and plug this into a blackbox optimizer.]

β

But, in this case we can solve exactly...

The <u>design matrix</u> is defined:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & & & \\ x_{N1} & x_{N2} & \cdots & x_{Np} \end{bmatrix}$$

Each row is an input vector from our data.

The <u>response vector</u> is

$$y = \langle y_1, y_2, \cdots, y_N \rangle$$

Each component is a response from our data.

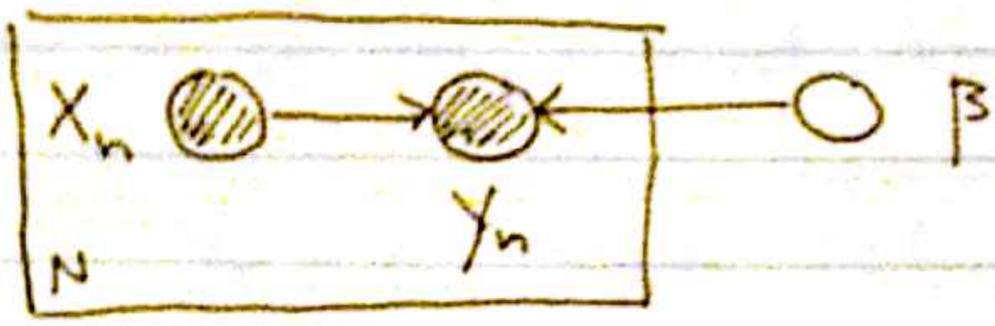
Now, $\nabla_\beta RSS(\beta) = -\underset{(p\times N)}{X^T}\underset{(N\times 1)}{(y - X\beta)} \rightarrow p\times 1$

Setting to the 0-vector and solving gives

$$X^T y - X^T X \hat{\beta} = 0$$
$$X^T X \hat{\beta} = X^T y$$
$$\hat{\beta} = (X^T X)^{-1} X^T y \longleftarrow \text{normal equations.}$$

Note: $X^T X$ must be invertible,
i.e., X must be full rank.

Probabilistic interpretation



$$y_n \sim N(\beta^T x_n, \sigma^2)$$

Like putting a Gaussian "bump" around the mean, which is a linear function of the inputs.

Note: <u>Conditional model</u> – inputs are not modeled.

Conditional ML:

$$\ell(\beta; x_{1:N}, y_{1:N}) = \log \prod \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ \frac{-(y_n - \beta^T x_n)^2}{2\sigma^2} \right\}$$

$$= \sum_{n=1}^{N} -\frac{1}{2} \log 2\pi\sigma^2 - \frac{1}{2} (y_n - \beta^T x_n)^2 / \sigma^2$$

Optimize w/r/t β gives

$$\hat{\beta} = \arg\max -\frac{1}{2} \sum_{n=1}^{N} (y_n - \beta^T x_n)^2 / \sigma^2$$

This is equivalent to <u>minimizing</u> the RSS.

Prediction comes from expectation:

$$\mathbb{E}[Y_{new} \mid x_{new}] = \hat{\beta}^T x_{new}$$

note: σ^2 does not play a role.

Important aside: Bias - Variance trade off

Consider a random data set drawn from a LR model

$$Y_n \mid x_n, \beta \sim N(\beta^* x_n, \sigma^2)$$

We can contemplate the MLE $\hat{\beta}$ as a random variable governed by the distribution of this data set.

Consider a new input x. How good is the predicted response mean compared to the true one?

$$MSE(\hat{\beta}x) = \mathbb{E}_{\mathcal{D}}[(\hat{\beta}x - \beta x)^2]$$

↑ random ↑ not random

$$= \mathbb{E}[(\hat{\beta}x)^2] - 2\mathbb{E}[\hat{\beta}x]\beta x + (\beta x)^2$$

$$= \mathbb{E}[(\hat{\beta}x)^2] - 2\mathbb{E}[\hat{\beta}x]\beta x + (\beta x)^2 + \mathbb{E}[\hat{\beta}x]^2 - \mathbb{E}[\hat{\beta}x]^2$$

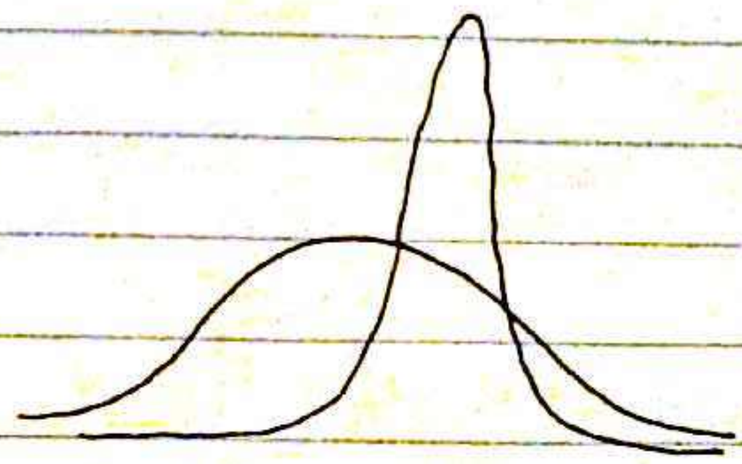
$$= \underbrace{(\mathbb{E}[(\hat{\beta}x)^2] - \mathbb{E}[\hat{\beta}x]^2)}_{\text{variance}} + \underbrace{(\mathbb{E}[\hat{\beta}x] - \beta x)^2}_{\text{squared bias}}$$

variance

reflects how sensitive the estimator is to the randomness inherent in the data.

squared bias

$\mathbb{E}[\hat{\beta}x] - \beta x = 0$ means $\hat{\beta}$ is unbiased.



- Classical stats focuses on unbiased estimators
- Modern stats explores the trade off.

Gauss - Markov theorem

The MLE / least squares estimate is the unbiased estimate with the smallest variance.

In regression, we trade-off bias for variance through regularization — placing constraints on β.

If the true MLE $\hat{\beta}$ lives outside the space then our estimator must be biased.

Constraining → lower variance
G.M. → biasedness.

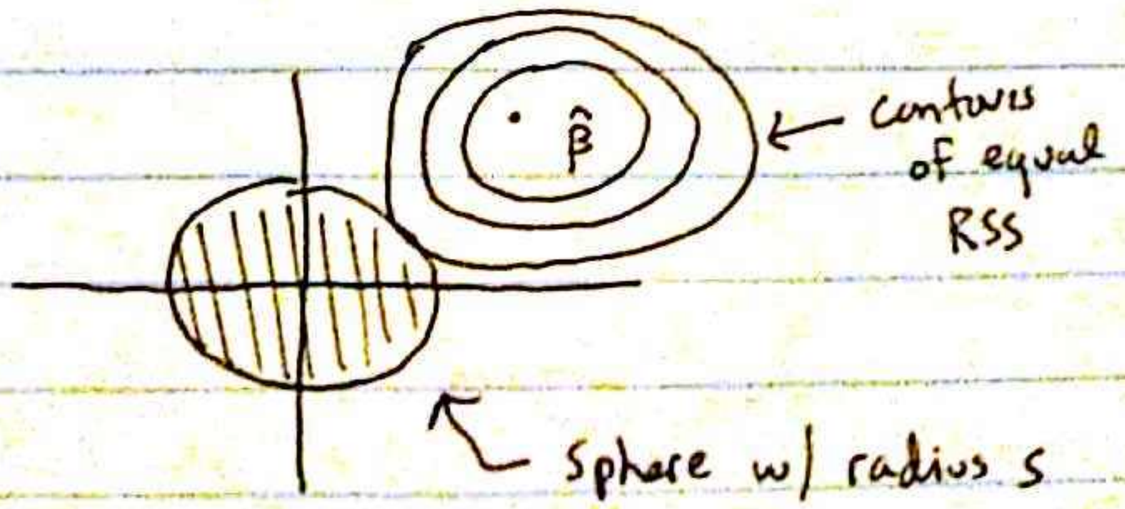
- Regularization encourages smaller and simpler models.
- Intuitively, simpler models are more robust to overfitting, generalizing poorly b/c of close match to training data
- Simpler models are also more interpretable, often a goal of regression.

Ridge regression

Optimize the RSS subject to a constraint on the sum of squares of the coefficients. This constrains β to be in a sphere.

$$\min \sum_{n=1}^{N} (y_n - \beta^T x_n)^2$$

$$s/t \quad \sum_{i=1}^{p} \beta_i^2 \leq s$$



As s increases, bias ↓, variance ↑

With simple calculus, can express the ridge optimization problem as:

$$\hat{\beta}^{ridge} = \arg\min_{\beta} \sum_{n=1}^{N} \frac{1}{2}(y_n - \beta x_n)^2 + \lambda \sum_{i=1}^{p} \beta_i^2$$

Nice: - for fixed λ, this is convex.
- 1-1 mapping b/w λ and s
- λ is the ~~complexity parameter~~

Choice of λ has a large effect on our fit.

Q What happens if we use training error to fit λ?

A No regularization!

In practice: use X-validation.

- Divide the data into k folds; grid λ into candidates
- For each fold, and candidate λ:
  - Estimate $\hat{\beta}_{k,\lambda}^{ridge}$ on the out-of-fold samples
  - Compute error within fold

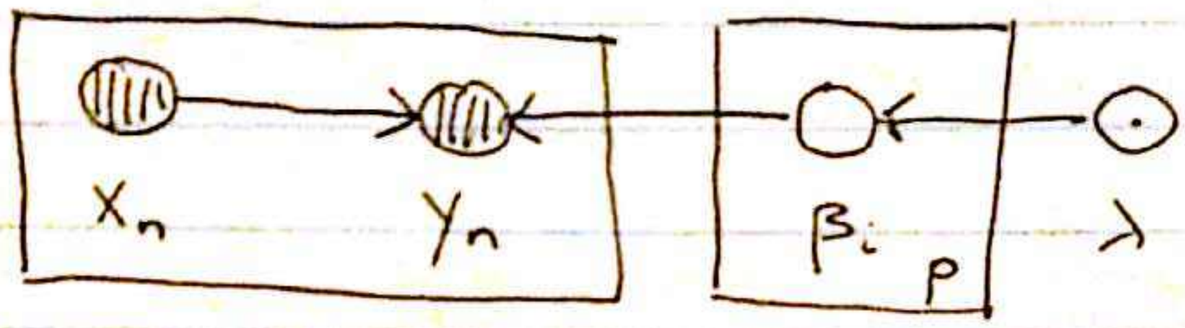
$$\epsilon_{n,\lambda} = [redacted] (y_n - \hat{\beta}_{k,\lambda}^{ridge} x_n)^2 \quad \text{for } n \in \text{fold } k.$$

- Now, we have an estimate of error for each point, computed from a parameter estimate that did not include that point.

- Finally, choose $\lambda = \arg\min_{\lambda} \frac{1}{N} \sum_{n=1}^{N} \epsilon_{n,\lambda}$

Bayesian linear regression

Ridge regression has a close connection to a Bayesian model:



$$\beta_i \sim N(0, 1/\lambda)$$
$$y_n \mid x_n, \beta \sim N(\beta^T x_n, \sigma^2)$$

Consider MAP estimation of β:

$$\hat{\beta}^{MAP} = \arg\max_{\beta} \log p(\beta \mid x_{1:N}, y_{1:N}, \lambda)$$

$$\arg\max_{\beta} \log \; p(y_{1:N} \mid x_{1:N}, \beta) \; \prod_{i=1}^{p} p(\beta_i \mid \lambda)$$

$$\arg\max_{\beta} \underbrace{\log p(y_{1:N} \mid x_{1:N}, \beta)}_{\text{Recall: this } \equiv RSS(\beta)} + \sum_{i=1}^{p} \log p(\beta_i \mid \lambda)$$

Note: $p(\beta_i \mid \lambda) = \left(1 \Big/ \sqrt{2\pi \tfrac{1}{\lambda}}\right) \exp\{\lambda \beta_i^2\}$

Same as ridge regression! →

$$\arg\max_{\beta} \; RSS(\beta) + \sum_{i=1}^{p} \lambda \beta_i^2$$

As λ increases, we move further from MLE.
As λ decreases, we move closer.
Both data and prior influence the Bayes estimate.

Note: a true blue Bayesian would never set λ by X-validation. This uses the data to set the prior!

Summary of ridge

- constrain $\hat{\beta}$ to be inside a sphere
- equiv. to minimize RSS + regularization
- also called shrinkage, b/c coefficients are pushed towards 0.
- Ridge trades off bias for decrease in var.
- Equivalent to MAP in Bayesian model.

Briefly: Lasso

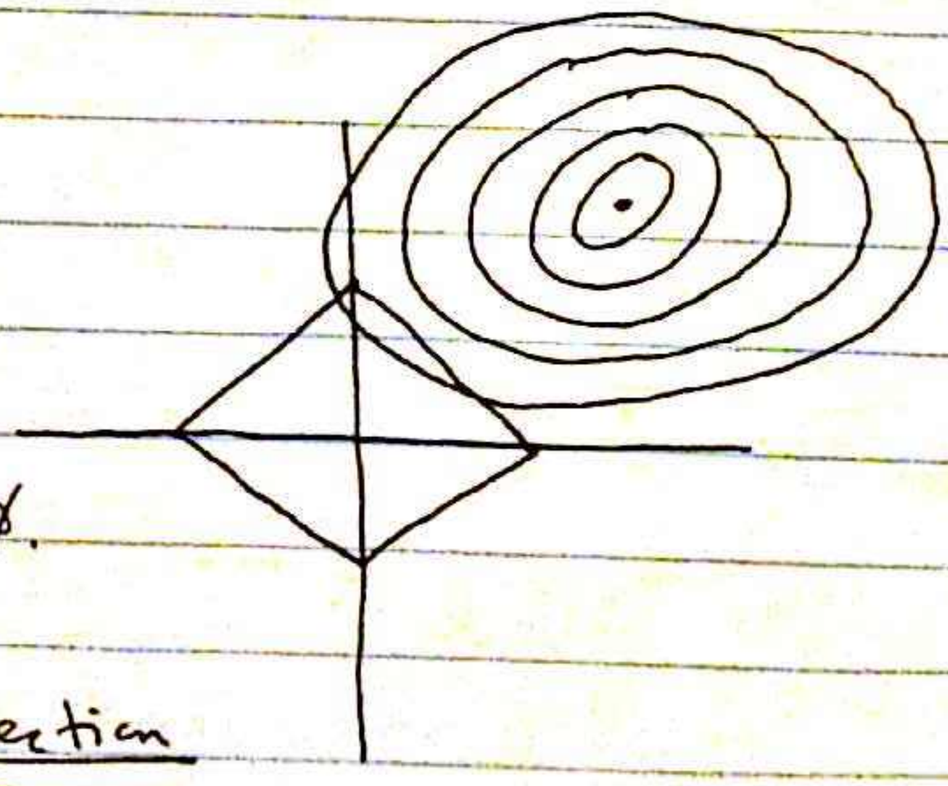
minimize $RSS(\beta)$ s/t $\sum_{i=1}^{p} |\beta_i| < s$

(replace L_2 with L_1)

$\Longleftrightarrow$ $\min_{\beta} RSS(\beta) + \lambda \sum_{i=1}^{p} |\beta_i|$

Nice: Still convex.

The contour will first hit the region at a corner. This finds sparse solution w/ coefficients equal to 0.



Efficient form of feature selection