# Machine learning algorithms

**Ridge regression 2**

2020–10-09

CSCI 471 / 571, Fall 2020

Kameron Decker Harris

# Homework questions?

$f$ : vectors $\to$ scalar

$$\frac{\partial}{\partial x_k} M_{ij} x_i x_j$$ ← product rule

$i=j \quad i \neq j$

(just a derivative)

$$\nabla_{\vec{v}} f := \begin{bmatrix} \frac{\partial f}{\partial v_1} \\ \frac{\partial f}{\partial v_2} \\ \vdots \end{bmatrix}$$

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$$

$$\frac{\partial \vec{x}}{\partial x_2} = \begin{bmatrix} \frac{\partial x_1}{\partial x_2} \\ \frac{\partial x_2}{\partial x_2} \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

$$\frac{\partial}{\partial v_1} \sum (\dots) = \sum \frac{\partial (\dots)}{\partial v_1}$$

$$f(\vec{u}, \vec{v}) = \sum \sum (\dots) + (\dots)$$

$A, B, c$ ← constants

class notes on polynomials 9/29, all in-class notebooks

$$1 \cdot \beta_0 + \beta_1 x + \beta_2 x^2 = f(x) \quad (d=1) \qquad \boxed{\min_{\vec{\beta}} \|F\vec{\beta} - \vec{y}\|^2}$$

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}_{n \times 1} \longrightarrow F = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$F \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} 1 \cdot \beta_0 + x_1 \beta_1 + x_1^2 \beta_2 \\ \vdots \end{bmatrix}$$

ISLR book

James, etc. 7.1

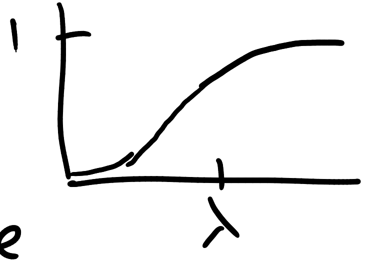
function $X$ : $n \times 2$ array

function

def test_error(true_fun, prediction_fun, n_test):

X_test = generate_grid(n_test)

# Ridge regression 2




- Reading: ISLR 6.2.1

$$\sum_{i=1}^{\mathrm{rank}(X)} \vec{v}_i \left( \frac{\vec{u}^T \vec{y}}{\sigma_i + \lambda} \right) = \vec{\beta}_{ridge}$$

huge $\vec{\beta}$ in polynomial regression

- Last time:
  - Ridge regression shrinks effects of small singular values
  - Helps deal with variance due to collinearity ⟷ small singular values

bias - var tradeoff

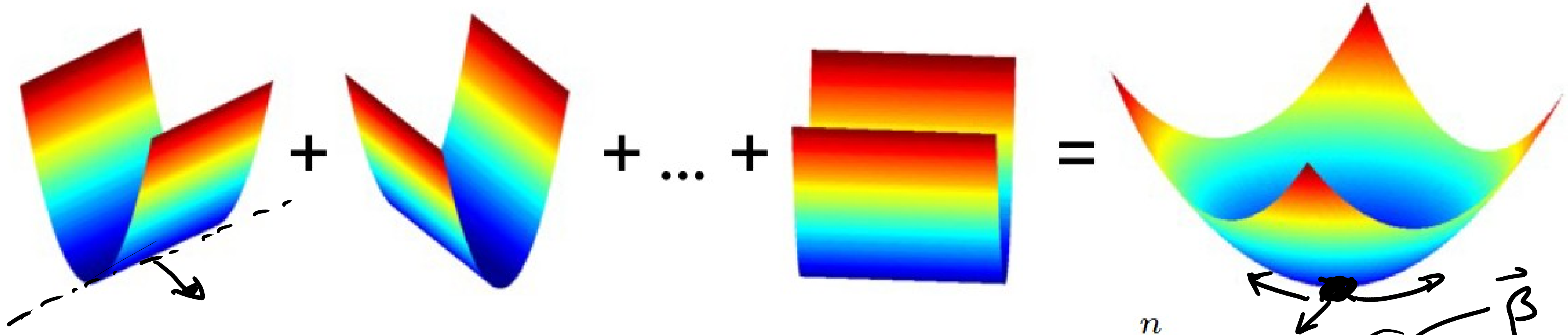
correlated features (columns of X)

- Today:
  - Geometric view
  - Probabilistic interpretation

hyperplane: $\{\vec{x} : \vec{w}^T \vec{x} + b = 0\}$

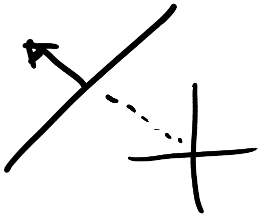
## Situation for n >> d

"bowl" shape



$\vec{\beta}$

$$\underbrace{(y_1 - x_1^T w)}_{\vec{x}^T \vec{\beta} = y_1}{}^2 + (y_2 - x_2^T w)^2 + \cdots + (y_n - x_n^T w)^2 = \sum_{i=1}^{n} (y_i - \vec{x}_i^T \vec{w})^2$$

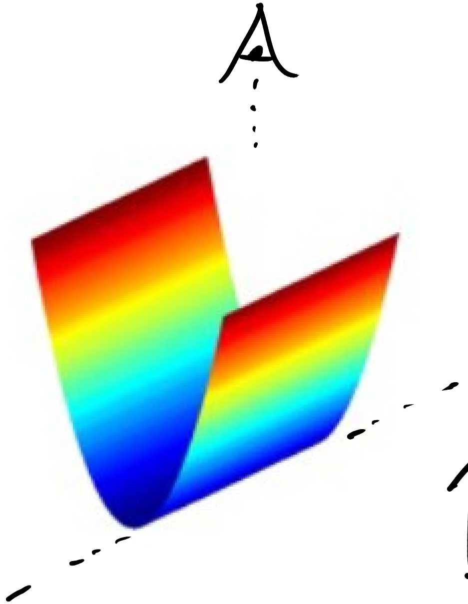


least-squares loss

Image credit: Kevin Jamieson, Jamie Morgenstern

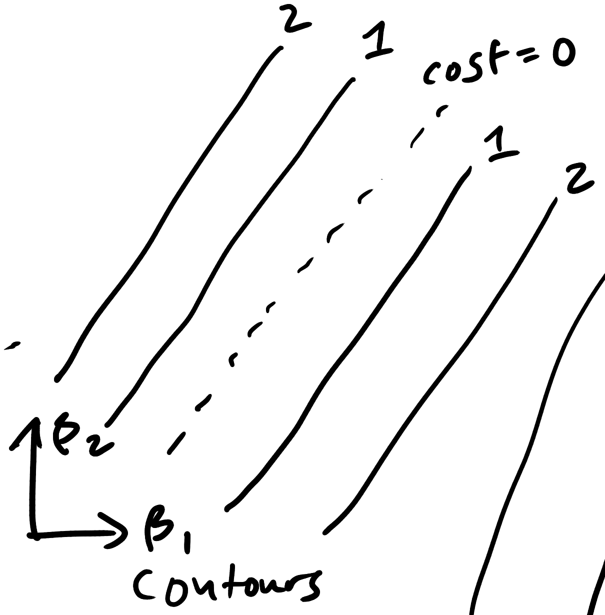
# Situation for n < d

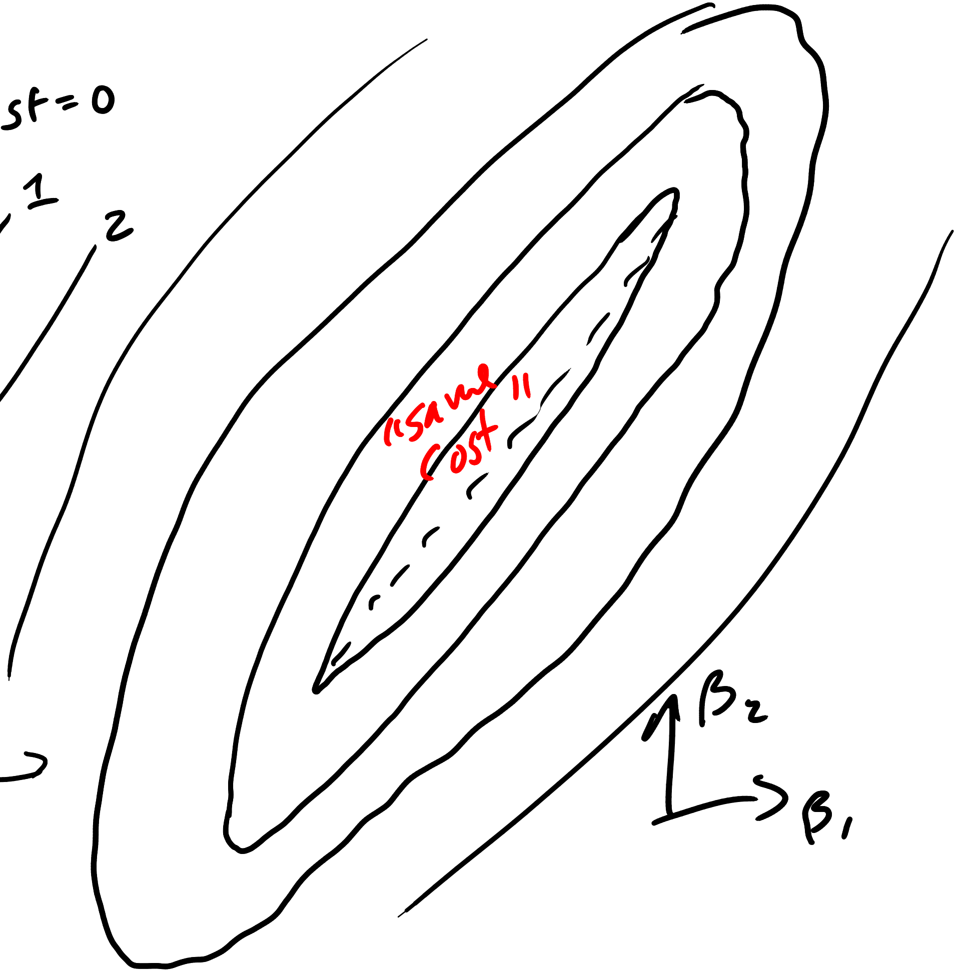
some columns correlated



"flat" directions of least squares loss

( X not full rank )

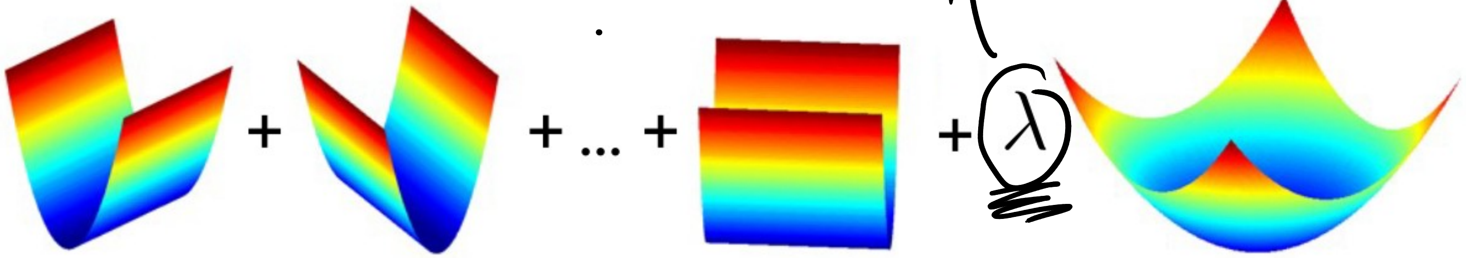



Least squares cost

Image credit: Kevin Jamieson, Jamie Morgenstern

# Solution: constrain solutions




$$\vec{\beta}_{ridge} = \arg\min_{\vec{\beta}} \boxed{\|X\vec{\beta} - \vec{y}\|^2} + \lambda \|\vec{\beta}\|^2$$

Problem 7
$D = I$
$\vec{\beta}' = 0$

Loss function : $\|X\vec{\beta} - \vec{y}\|^2$ goodness of fit

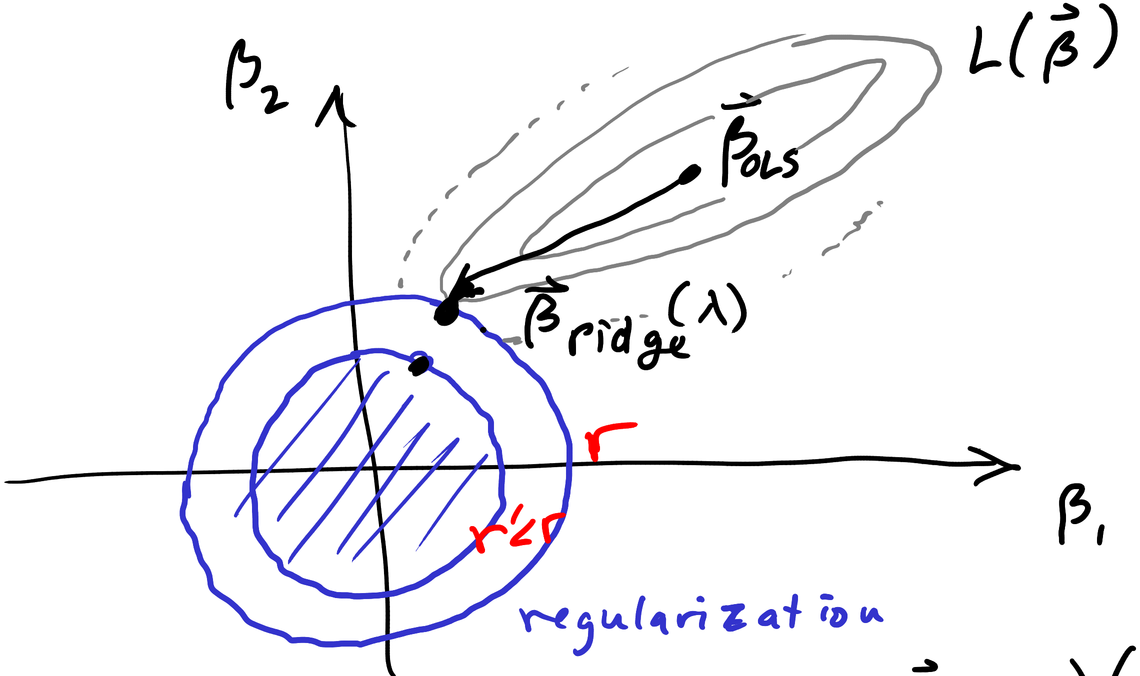
Regularization : $\underbrace{\lambda \|\vec{\beta}\|^2}_{R(\vec{\beta})} = \lambda \sum_{i=1}^{d} \beta_i^2$ penalty term
Keep $\vec{\beta}$ close to 0

Image credit: Kevin Jamieson, Jamie Morgenstern

# Picture of loss & regularizer

$$(X^TX + \lambda I)^{-1}(X^T\vec{y}) = \vec{\beta}_{ridge}$$

$$L(\vec{\beta}) + \underbrace{\lambda \|\vec{\beta}\|^2}_{R(\vec{\beta})}$$




$\lambda$ huge $\Rightarrow$ small $r$

$\lambda \to 0$, $r \to \infty$

Same as picking some radius

$r = r(\lambda)$

force

$\|\vec{\beta}_{ridge}\| \leq r$

$$R(\vec{\beta}) = \lambda(\beta_1^2 + \beta_2^2) = c$$

# Bayes’ rule

# MAP estimator

# Practical considerations: Ridge

- Best if features X are standardized

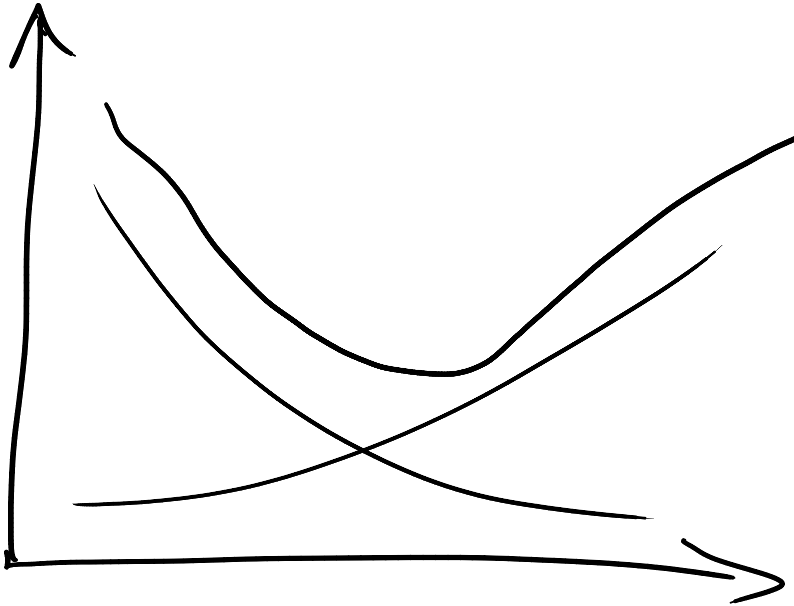
- Don’t penalize intercept

```
def prediction_fun (X, beta)
    ...

def test_error( ... )


generates training data
fit training data ⟶ beta

err = test_error(true_fun, prediction_fun)
```

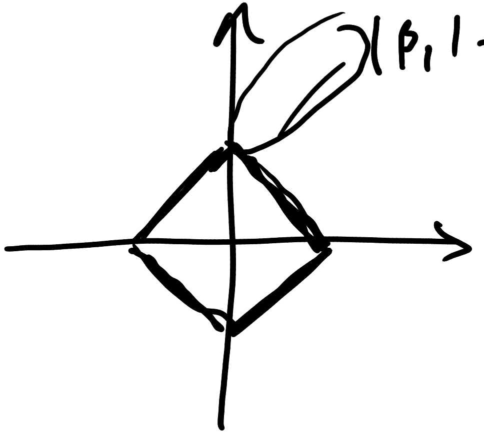


Complexity

$\|\beta\|$

or $1/\lambda$

$\ell_1$ norm



$|\beta_1| + |\beta_2|$

ISLR has picture