6.867 Machine Learning Fall 2017

Lecture 3. Regularization, Model Selection

Advertisement Campaign

- Planning Marketing Budget Across Channels: TV, Radio and NewsPaper

10 Markets

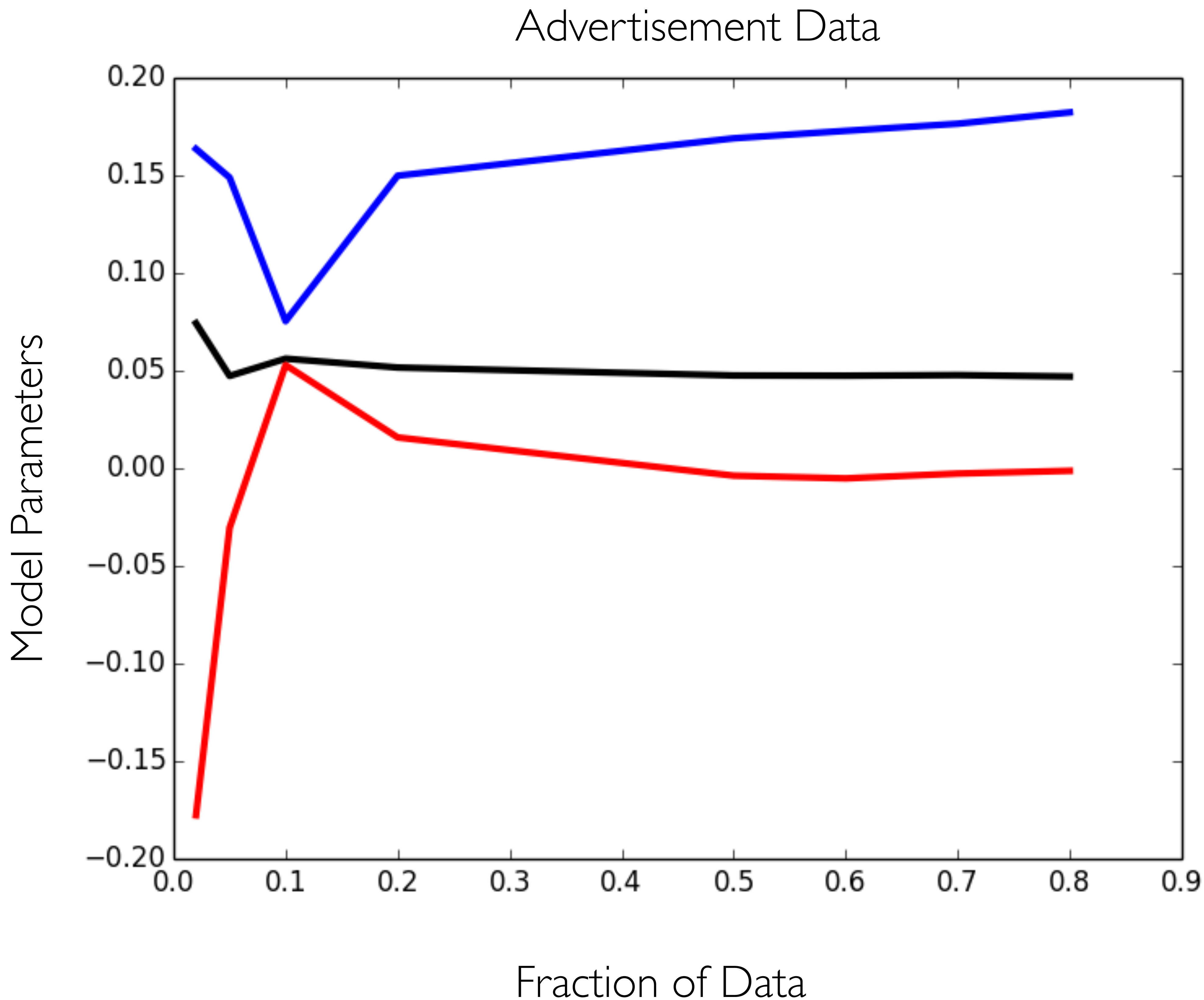
- Data across ~~200~~ Markets
  - Spending for TV, Radio, NewsPaper
  - Resulting Sales

	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9
6	8.7	48.9	75	7.2
7	57.5	32.8	23.5	11.8
8	120.2	19.6	11.6	13.2
9	8.6	2.1	1	4.8
10	199.8	2.6	21.2	10.6

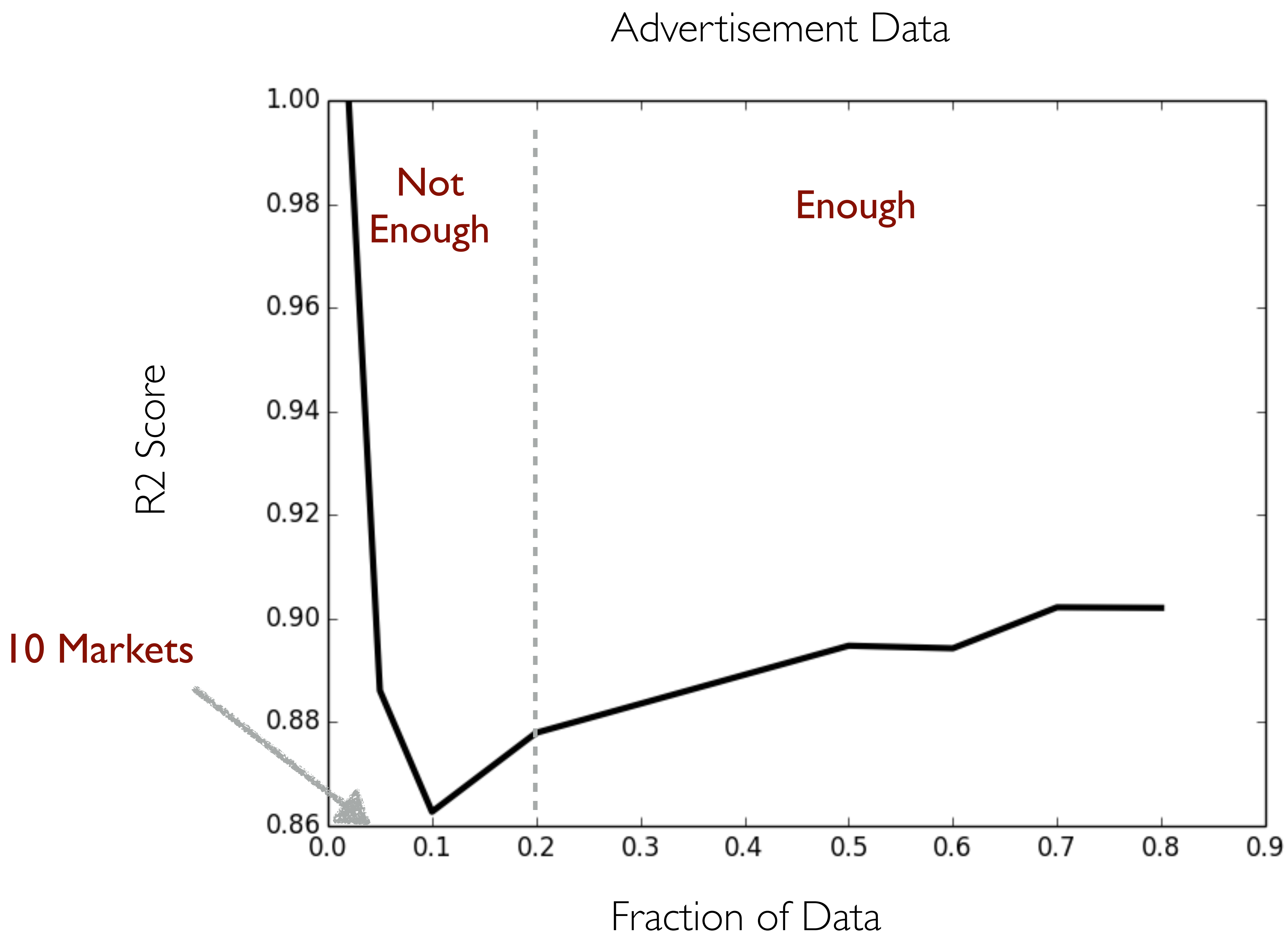
Sample Data

- Questions
  - Is there a relationship between Sales and Marketing Budget?
  - If yes, can we "predict" Sales given Marketing Budget across Channels?
  - And, how "important" are each of the channels? do they interact?

How Much Data is Enough?



How Much Data is Enough?



Recall: Linear Regression

- Ideal Solution:

$$f(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$$

- Linear approximation yields

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} = \sum_{i=0}^{p} w_i x_i, \text{ with } x_0 = 1$$

- Linear regression: find $\mathbf{w}$ that minimizes

$$\sum_{n=1}^{N} (y_n - \mathbf{w}^T\mathbf{x}_n)^2$$

- Questions:
  - Are we overfitting? Are there systematic ways to avoid overfitting?

Bias-Variance Tradeoff

- Consider any estimator $f(X)$
- Loss can be written as

$$\mathbb{E}[(Y - f(X))^2] =$$

$$\mathbb{E}[(Y - \mathbb{E}[Y|X])^2]$$ inherent loss

$$+\mathbb{E}[(\mathbb{E}[Y|X] - \mathbb{E}[f(X)])^2]$$ (bias)^2

$$+\mathbb{E}[(f(X) - \mathbb{E}[f(X)])^2]$$ variance

Bias-Variance Tradeoff with Linear Regression

- Increasing model complexity
  - Higher Variance, Low bias
- Decreasing model complexity
  - Lower Variance, Higher bias
- In Linear Regression
  - The model complexity is captured by "set" of feasible parameters
- Therefore, we can achieve bias-variance tradeoff
  - By changing restrictions on choice of allowed model parameter
    - This is precisely achieved via Regularization

Regularized Linear Regression

- Ridge regression: for $\lambda \geq 0$

$$\text{minimize} \;\; \sum_n (y_n - \mathbf{w}^T \mathbf{x}_n)^2 + \lambda \, \mathbf{w}^T \mathbf{w}$$

- It is Lagrangian formulation of

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_n (y_n - \mathbf{w}^T \mathbf{x}_n)^2 \\ \text{over} & \mathbf{w}^T \mathbf{w} \leq \eta \end{array}$$

Ridge Regression

- Goal:

$$\text{minimize } g(\mathbf{w}), \quad \text{where}$$

$$g(\mathbf{w}) = (\mathbf{Y} - X\mathbf{w})^T(\mathbf{Y} - X\mathbf{w}) + \lambda\,\mathbf{w}^T\mathbf{w}$$

- Now

$$\nabla g(\mathbf{w}) = -2X^T\mathbf{Y} + 2X^TX\mathbf{w} + 2\lambda\,\mathbf{w}$$

- Therefore, solution is

$$\mathbf{w} = (X^TX + \lambda\mathbf{I})^{-1}X^T\mathbf{Y}$$

Understanding Ridge Regression

- Consider a simple setting:
    - one dimensional feature and target; and $\sum_n x_n = 0$

$$X^T X = \begin{bmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_n \end{bmatrix} \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} = N \begin{bmatrix} 1 & \bar{x} \\ \bar{x} & \bar{x^2} \end{bmatrix}$$

$$X^T \mathbf{Y} = \begin{bmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = N \begin{bmatrix} \bar{y} \\ \bar{xy} \end{bmatrix}$$

Understanding Ridge Regression

- Therefore, solution of Ridge regression:

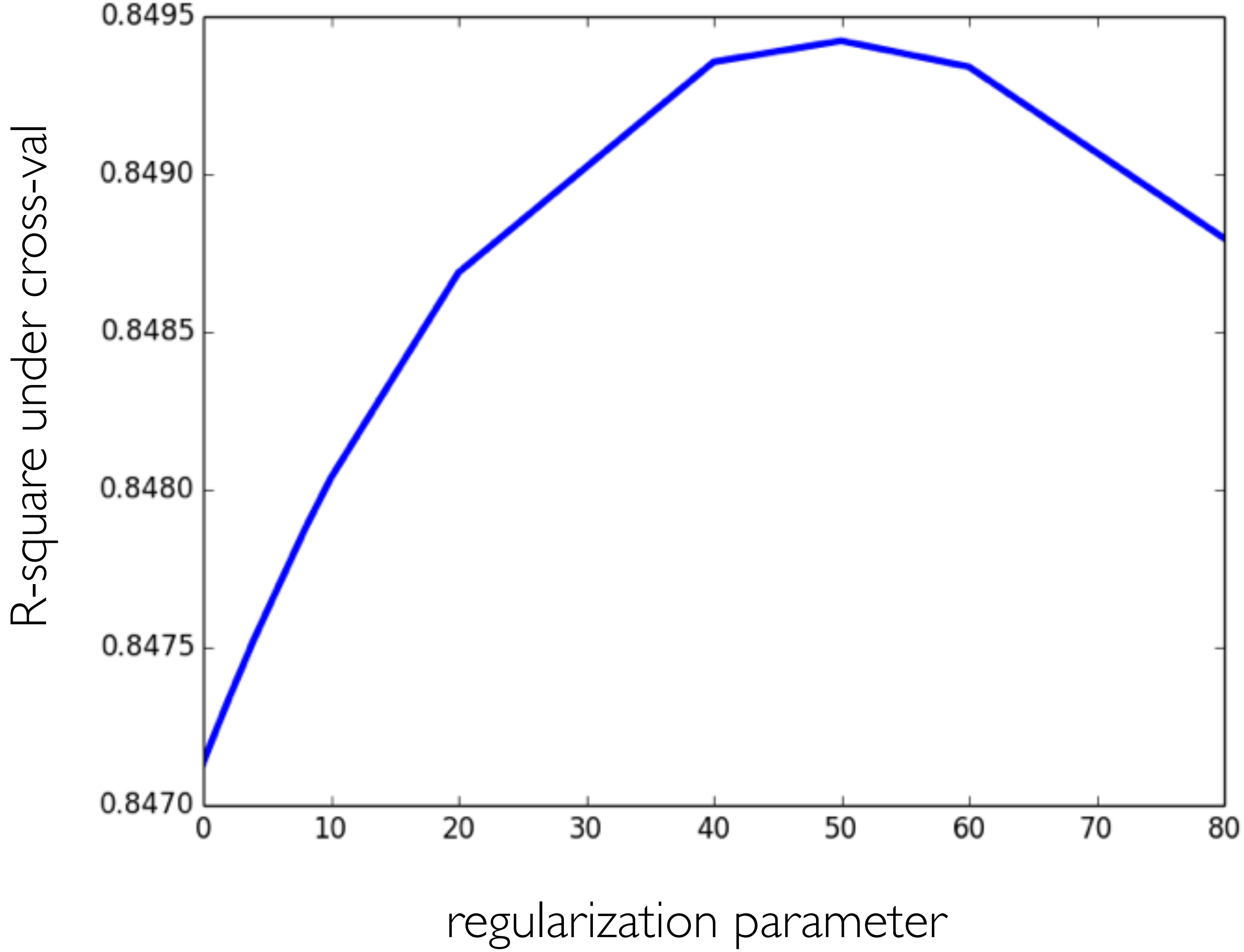
$$\mathbf{w} = \begin{bmatrix} (1 + \frac{\lambda}{N})^{-1}\bar{y} \\ (\bar{x^2} + \frac{\lambda}{N})^{-1}\bar{xy} \end{bmatrix}$$

- That is, by increasing λ , the model parameters simply *shrink!*

- That is why it is also called *Shrinkage*

Ridge Regression

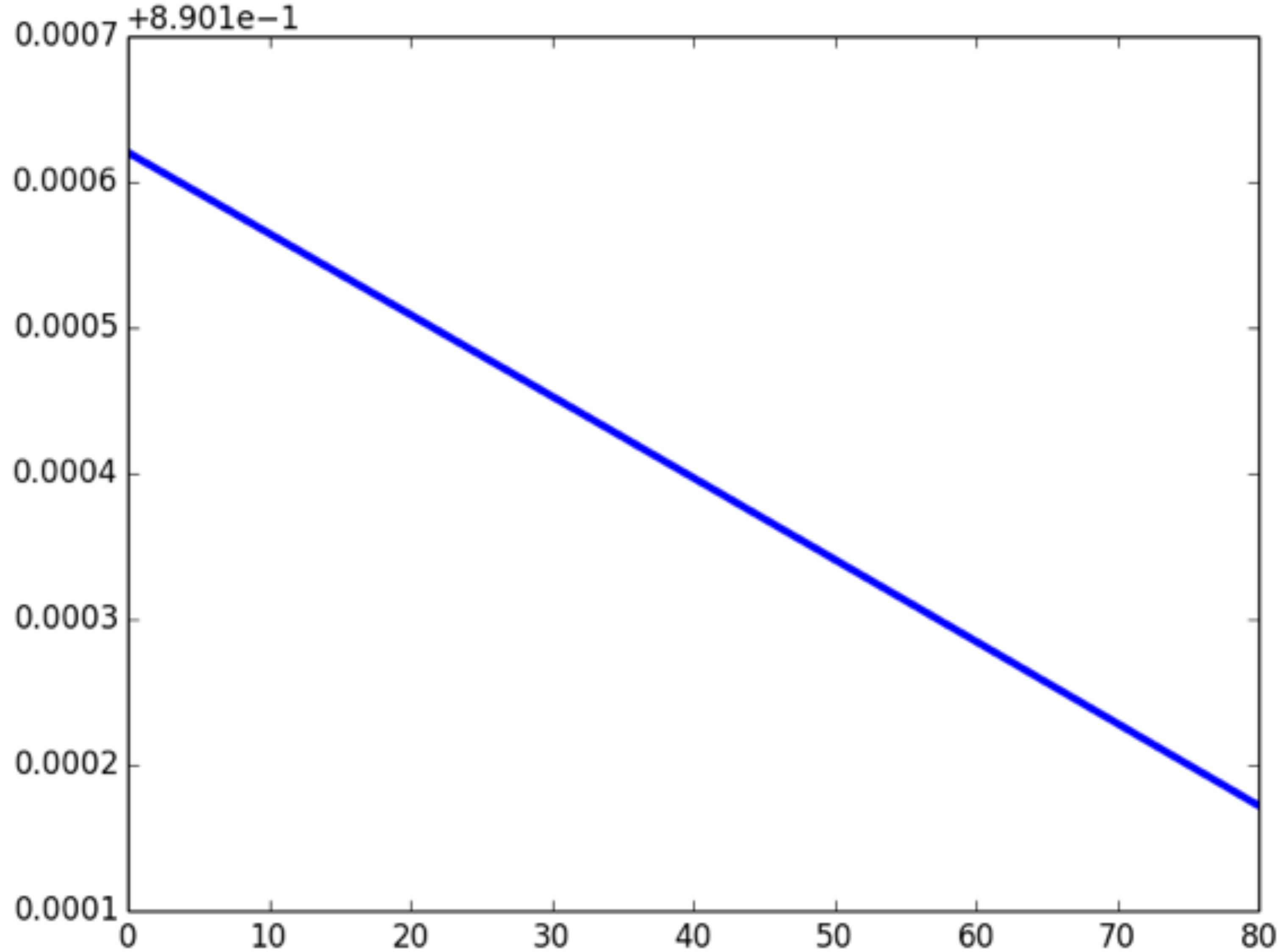
- Advertisement data with few different regularization parameter



4% data used

Ridge Regression

- Advertisement data with few different regularization parameter



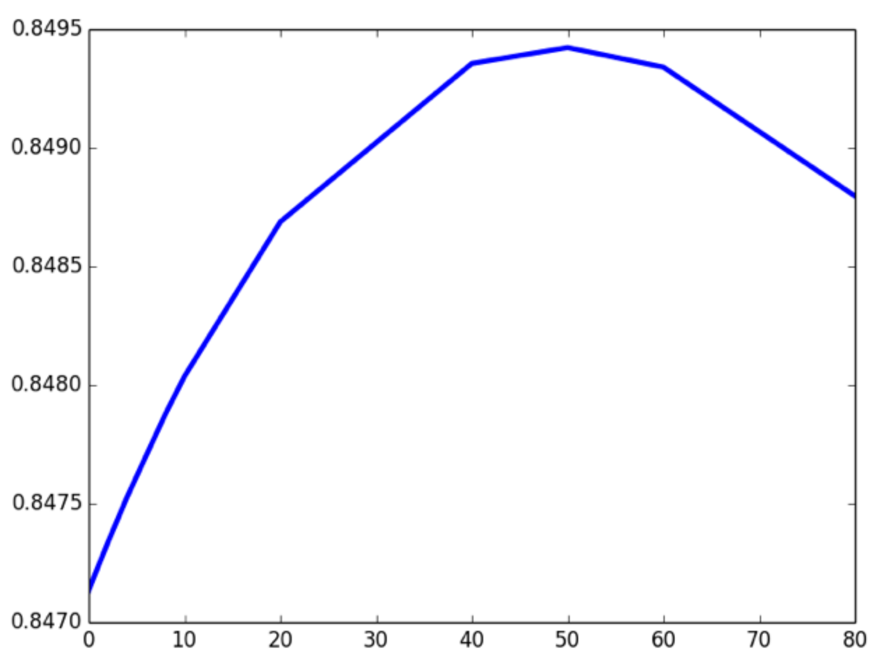
R-square under cross-val

regularization parameter

50% data used

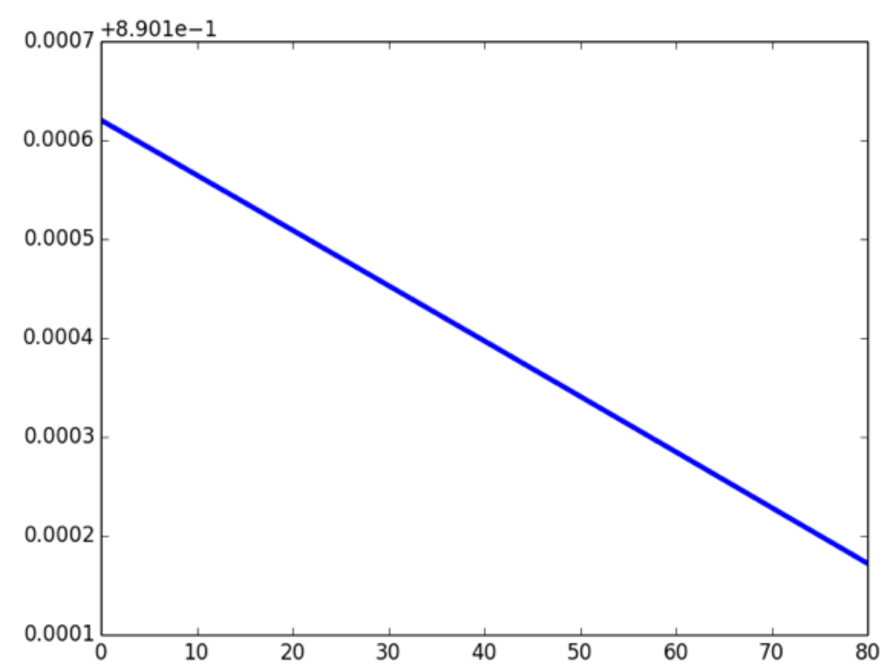
Ridge Regression

- Advertisement data with few different regularization parameter
  - Regularization is particularly useful when data is limited
  - In large data limit, "maximum likelihood" is reasonable



small

vs



large

Bias-Variance with Ridge Regression

- Recall simple example
  - As regularization parameter λ increases, model parameter *shrinks*
  - And bias increases

- Covariance of model parameter

$$
\begin{aligned}
\mathrm{Cov}[\mathbf{w}] &= \mathrm{Cov}[A\mathbf{Y}], \quad \text{where} \quad A = (X^T X + \lambda \mathbf{I})^{-1} X^T \\
&= A\mathrm{Cov}[\mathbf{Y}]A^T \\
&= \sigma^2 (X^T X + \lambda \mathbf{I})^{-1} (X^T X) (X^T X + \lambda \mathbf{I})^{-1}
\end{aligned}
$$

Bias-Variance with Ridge Regression

- For simple example with one dimensional feature (and $\sum_n x_n = 0$)

$$\mathrm{Cov}[\mathbf{w}] = \frac{\sigma^2}{N} \begin{bmatrix} \left(1 + \frac{\lambda}{N}\right)^{-2} & 0 \\ 0 & \bar{x^2}\left(\bar{x^2} + \frac{\lambda}{N}\right)^{-2} \end{bmatrix}$$

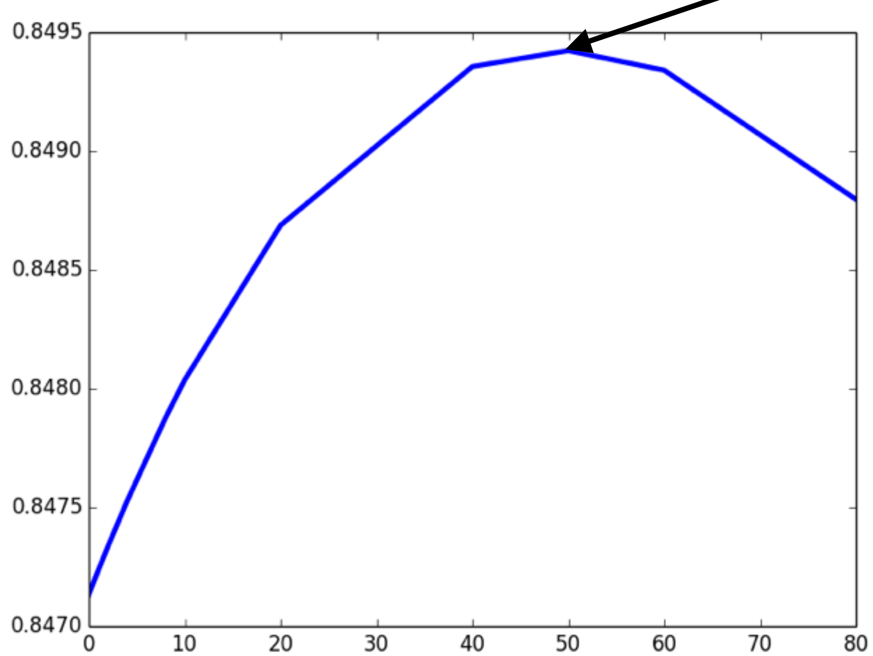
- That is
  - $\sum_i \mathrm{Var}(w_i)$ is decreasing with increase in λ

- In summary: as λ increases
  - bias increases and variance decreases giving us desired trade-off

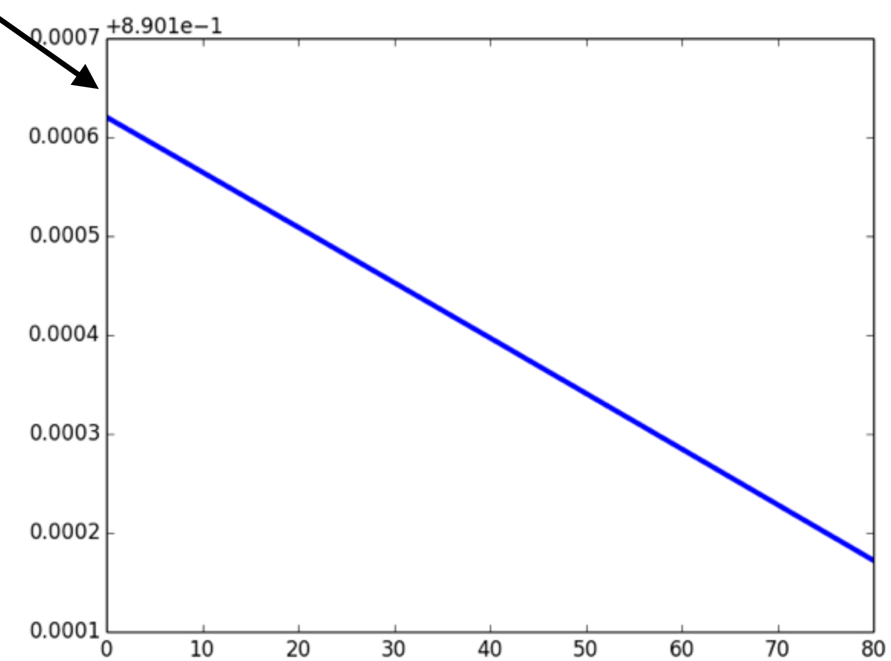
Model Selection

- Use cross-validation:

λ^*



vs



small

large

Other Forms of Regularization

- p-norm regularization

$$\text{minimize} \sum_n (y_n - \mathbf{w}^T \mathbf{x}_n)^2 + \lambda \|\mathbf{w}\|_p^p$$

$$\Updownarrow$$

$$\begin{aligned} &\text{minimize} && \sum_n (y_n - \mathbf{w}^T \mathbf{x}_n)^2 \\ &\text{over} && \|\mathbf{w}\|_p \leq \eta \end{aligned}$$

  - Ridge with p = 2
  - LASSO with p = 1

LASSO

- Brief history
  - LASSO = Least Absolute Shrinkage and Selection Operator
  - Automated selection of "relevant features"
  - A large number of features is useful to capture complex models, e.g.
    - variety of representations for capturing structure of image
    - or, higher order polynomials
  - But limited data does not allow meaningful selection
  - Regularization like Ridge Regression tends to select *everything*
  - LASSO, on the other hand, tries to choose *sparsest* model parameter

Ridge vs LASSO

- Ridge regression is a *Shrinkage* estimator

$$\text{minimize} \quad (y-w)^2 + \lambda w^2 \quad \Rightarrow \quad w = \frac{y}{1+\lambda}$$

- Lasso is *thresholding* estimator

$$\text{minimize} \quad (y-w)^2 + \lambda |w|$$

- Then $$w = \begin{cases} y - \frac{\lambda}{2} & \text{if } y > \frac{\lambda}{2} \\ y + \frac{\lambda}{2} & \text{if } y < -\frac{\lambda}{2} \\ 0 & \text{if } y \in [-\frac{\lambda}{2}, \frac{\lambda}{2}] \end{cases}$$

- That is, *small* values are forced to 0

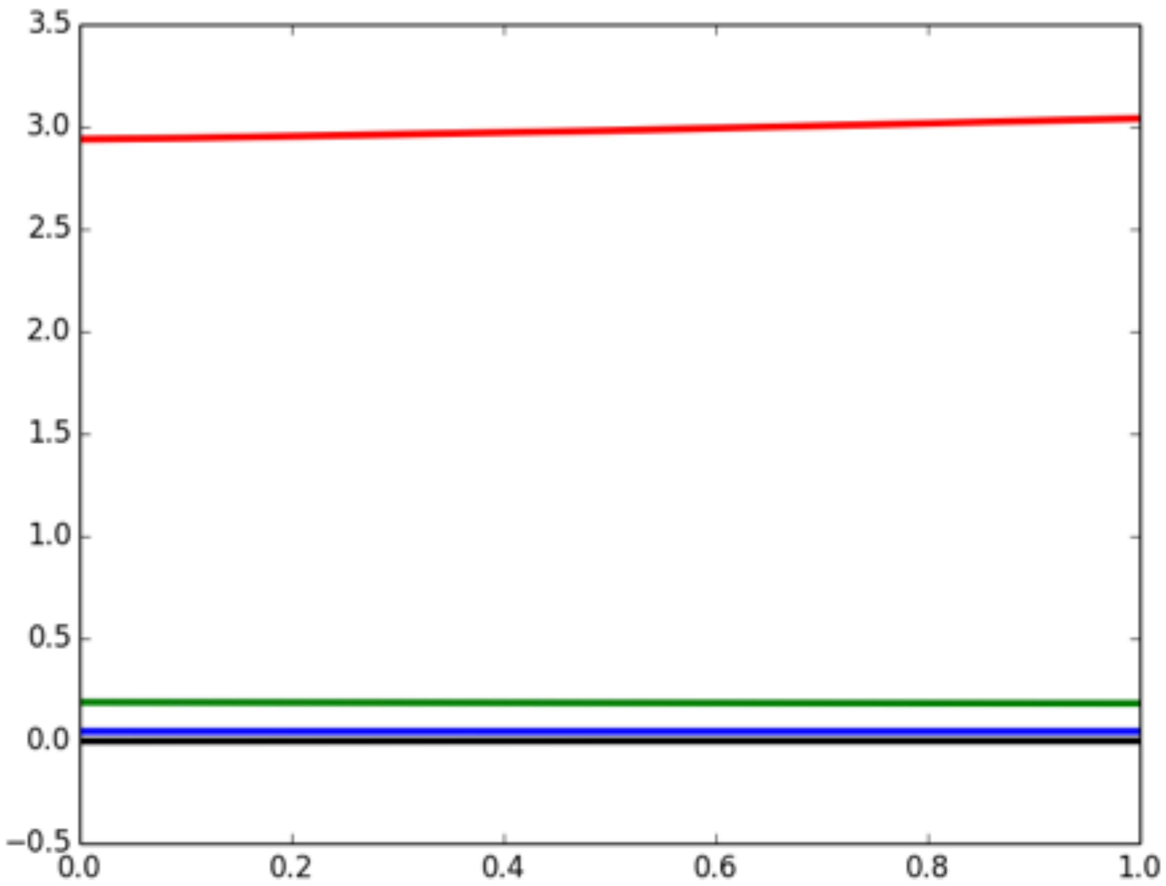
LASSO

Linear Regression Soln

$$w_0 = 2.939$$
$$w_{\mathrm{TV}} = 0.046$$
$$w_{\mathrm{Radio}} = 0.189$$
$$w_{\mathrm{NewsPaper}} = -0.001$$

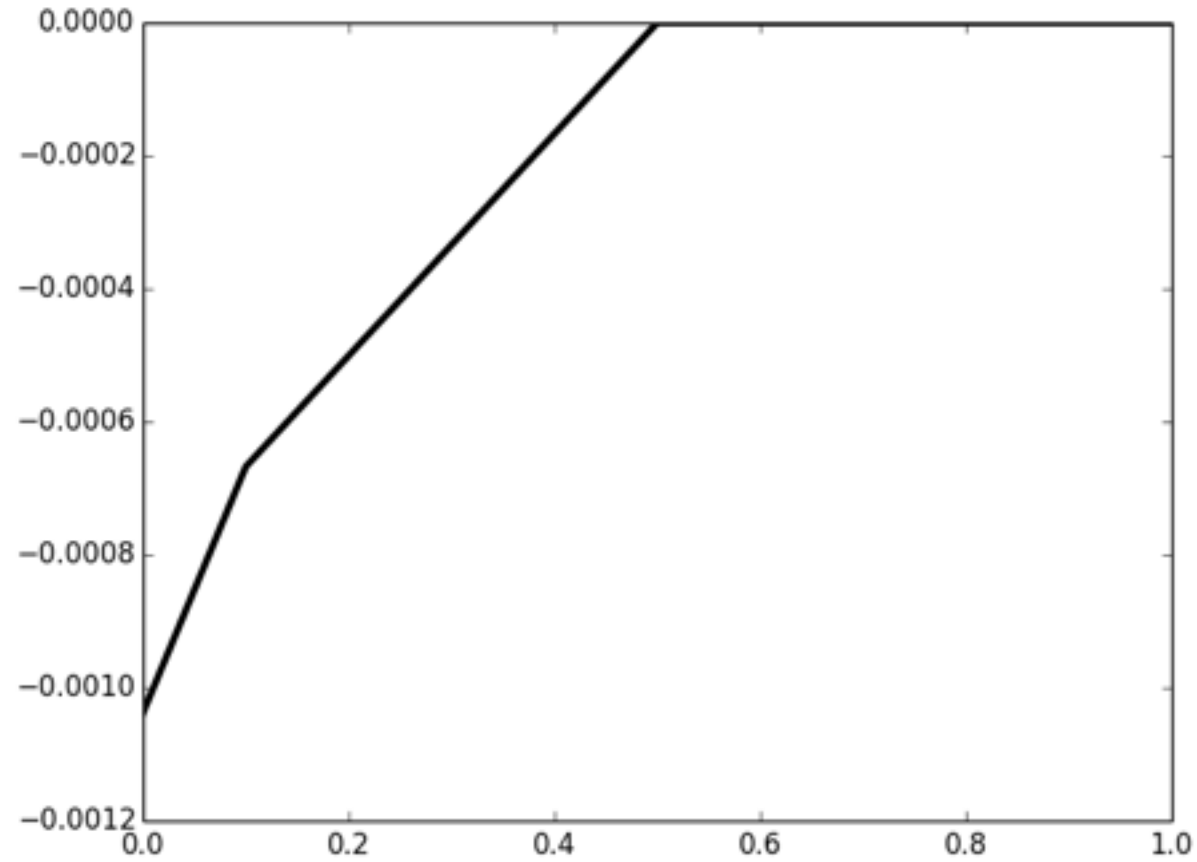
- Advertisement data with LASSO

all model parameters



regularization parameter

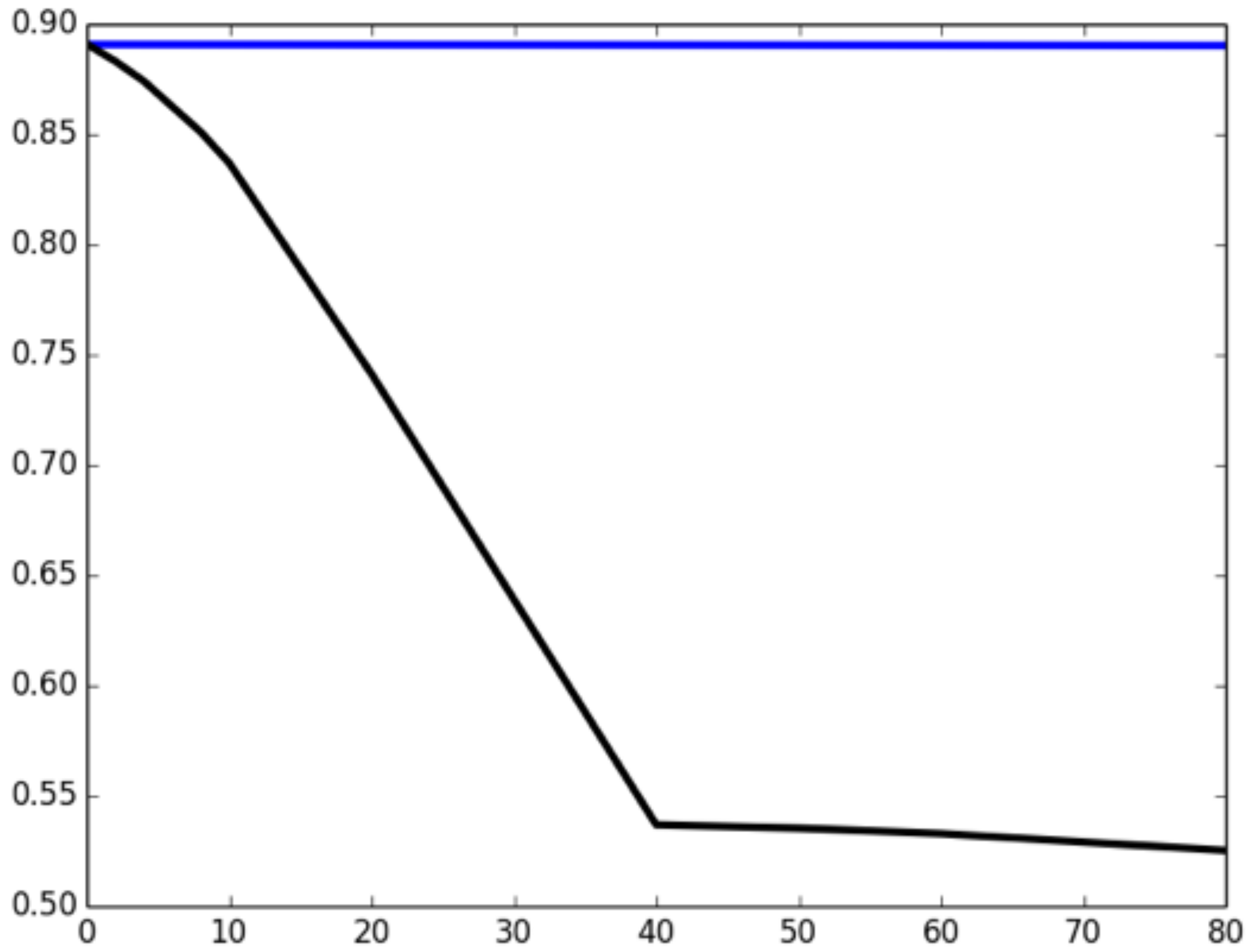
newspaper parameter



regularization parameter

LASSO vs Ridge

- Advertisement data with LASSO vs Ridge
    - R-square over various parameter values



Solving LASSO

- LASSO requires solving optimization problem

$$\text{minimize } g(\mathbf{w}) \equiv \sum_n (y_n - \mathbf{w}^T \mathbf{x}_n)^2 + \lambda \left\| \mathbf{w} \right\|_1$$

- There is no *closed* form solution
  - We need a generic algorithm
  - *Optimization to our rescue!*

Gradient Descent

- Optimization:

$$\text{minimize } g(\mathbf{w}) \text{ over } \mathbf{w} \in \mathbb{R}^d$$

- Iterative algorithm: in iteration t+1

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \alpha_t \nabla g(\mathbf{w}^t)$$

  - where

$$\alpha_t \geq 0, \quad \lim_{t\to\infty} \alpha_t = 0, \quad \sum_t \alpha_t = \infty$$

Solving LASSO

- LASSO requires solving optimization problem

$$\text{minimize } g(\mathbf{w}) \equiv \sum_n (y_n - \mathbf{w}^T \mathbf{x}_n)^2 + \lambda \|\mathbf{w}\|_1$$

- (Sub-)gradient algorithm: iterative algorithm
  - Initially, start with $\mathbf{w}^{(0)} = \mathbf{1}$
  - Iteratively, obtain $\mathbf{w}^{(t+1)} = [w_i^{(t+1)}]$ for $t \geq 0$ where

$$w_i^{(t+1)} = w_i^{(t)} - \alpha^{(t)} \frac{\partial g(\mathbf{w}^{(t)})}{\partial w_i}$$

$$\frac{\partial \mathbf{w}}{\partial w_i} = -2 \sum_n (y_n - \mathbf{w}^T x_n) x_{ni} + \lambda \, \text{sign}(w_i^t)$$

Projected Gradient Descent

- Optimization:

$$\text{minimize } g(\mathbf{w}) \text{ over } \mathbf{w} \in \mathcal{C}, \ \text{ where } \mathcal{C} \text{ is a convex set}$$

- Iterative algorithm: in iteration t+1

$$\mathbf{v}^{t+1} = \mathbf{w}^{t} - \alpha_t \nabla g(\mathbf{w}^{t})$$
$$\mathbf{w}^{t+1} = \mathrm{Proj}_{\mathcal{C}}(\mathbf{v}^{t+1})$$

$$\alpha_t \geq 0, \ \lim_{t \to \infty} \alpha_t = 0, \ \sum_t \alpha_t = \infty$$

Stochastic Gradient Descent

- Optimization for model learning:

$$\text{minimize } g(\mathbf{w}) \text{ over } \mathbf{w} \in \mathbb{R}^d$$

$$g(\mathbf{w}) = \sum_n (y_n - \mathbf{w}^T x_n) = \sum_n L(\mathbf{w}; x_n, y_n)$$

- Gradient has form

$$\nabla g(\mathbf{w}) = \sum_n \nabla L(\mathbf{w}; x_n, y_n)$$

- *Poor man*'s gradient descent

$$\mathbf{w}^{n+1} = \mathbf{w}^n - \alpha_n \nabla L(\mathbf{w}^n; x_n, y_n)$$

  - and potentially do this by passing over the dataset multiple times