# CS578 Statistical Machine Learning Lecture 7

Jean Honorio
Purdue University

*(based on slides by Tommi Jaakkola, MIT CSAIL)*

# Today's topics

- Preface: regression
  - linear regression, kernel regression
- Feature selection
  - information ranking, regularization, subset selection

# Linear regression

- We seek to learn a mapping from inputs to continuous valued outputs (e.g., price, temperature)
- The mapping is assumed to be linear in the feature space so that the predicted output is given by

$$\hat{y}(\underline{x}) = \underline{\theta} \cdot \phi(\underline{x})$$

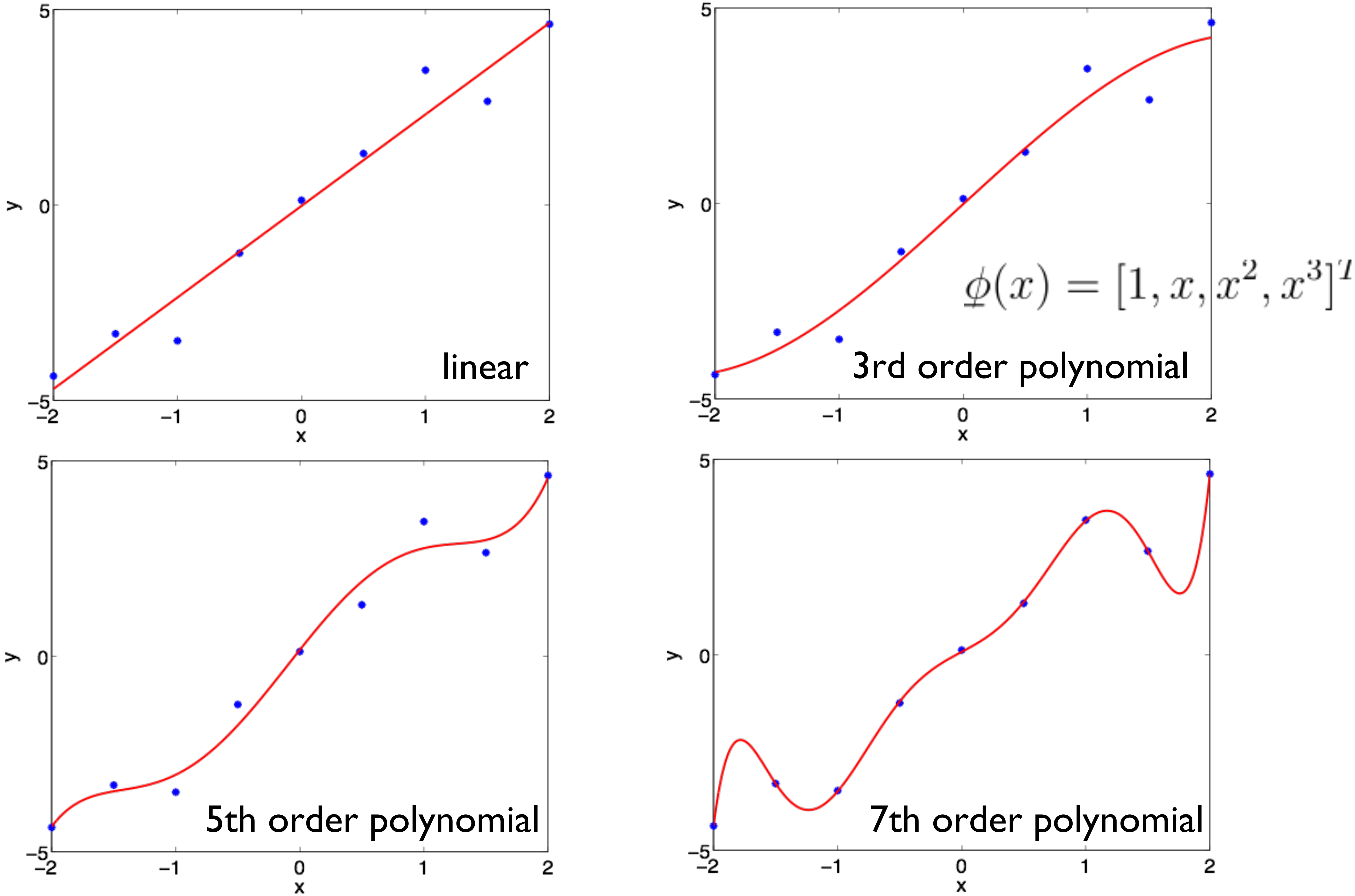
- Assuming that the noise in the observed outputs is additive zero mean Gaussian we can obtain the parameters from training samples by minimizing

$$J(\underline{\theta}) = \frac{1}{2} \underbrace{\sum_{t=1}^{n}}_{\text{sum over the training examples}} \underbrace{\left(y_t - \underline{\theta} \cdot \phi(\underline{x}_t)\right)^2}_{\text{squared prediction loss on the example}} + \underbrace{\frac{\lambda}{2} \|\underline{\theta}\|^2}_{\text{regularization term}}$$

- The regularization term guarantees that any unused parameter dimensions are set to zero

# Linear regression

- We can easily obtain non-linear regression functions by considering different feature mappings



# Linear regression solution

$$J(\underline{\theta}) = \frac{1}{2}\sum_{t=1}^{n}\left(y_t - \underline{\theta}\cdot\underline{\phi}(\underline{x}_t)\right)^2 + \frac{\lambda}{2}\|\underline{\theta}\|^2$$

by computing primal and dual

$$\frac{d}{d\underline{\theta}}J(\underline{\theta}) = \sum_{t=1}^{n} - \overbrace{\boxed{\left(y_t - \underline{\theta}\cdot\underline{\phi}(\underline{x}_t)\right)}}^{\alpha_t}\underline{\phi}(\underline{x}_t) + \lambda\underline{\theta} = 0$$

$$\Rightarrow \underline{\theta}(\alpha) = \frac{1}{\lambda}\sum_{t=1}^{n}\alpha_t\underline{\phi}(\underline{x}_t)$$

scalar: positive, negative or zero

- The solution lies in the span of the feature vectors (this is due to the regularization term).

# Dual linear regression

- The dual parameters are obtained as the solution to a linear equation

$$\begin{aligned}\alpha_t &= y_t - \underline{\theta}(\alpha) \cdot \underline{\phi}(\underline{x}_t) \\ &= y_t - \frac{1}{\lambda} \sum_{i=1}^{n} \alpha_i \underbrace{[\underline{\phi}(\underline{x}_i) \cdot \underline{\phi}(\underline{x}_t)]}_{\text{kernel } K(\underline{x}_i, \underline{x}_t)}\end{aligned}$$

$$\Rightarrow \underset{n \times 1}{\alpha^*} = (I + \frac{1}{\lambda} \underset{n \times n \text{ Gram matrix}}{K})^{-1} \underset{n \times 1}{y}$$

- Predicted output for a new input is given by

$$\hat{y}(\underline{x}) = \underline{\theta}(\alpha^*) \cdot \underline{\phi}(\underline{x}) = \frac{1}{\lambda} \sum_{i=1}^{n} \alpha_i^* \underbrace{[\underline{\phi}(\underline{x}_i) \cdot \underline{\phi}(\underline{x})]}_{\text{kernel } K(\underline{x}_i, \underline{x})}$$

# Today's topics

- Preface: regression
  - linear regression, kernel regression
- Feature selection
  - information ranking, regularization, subset selection

# Coordinate selection

- Linear models

classification

$$y = \text{sign}\big(\,\underline{\theta} \cdot \underline{\phi}(\underline{x})\,\big) \in \{-1, 1\}$$

regression

$$y = \underline{\theta} \cdot \underline{\phi}(\underline{x}) \in \mathcal{R}$$

etc.

feature coordinate

$$\underline{\phi}(\underline{x}) = \begin{bmatrix} \phi_1(\underline{x}) \\ \cdots \\ \phi_d(\underline{x}) \end{bmatrix}$$

- We seek to identify a few feature coordinates that the class label or regression output primarily depends on
- This is often advantageous in order to improve generalization (as feature selection exerts additional complexity control) or to gain interpretability

# Simple coordinate selection

- There are a number of different approaches to coordinate selection
- Information analysis
  - rank individual features according to their mutual information with the class label
  - limited to discrete variables
- 1-norm regularization
  - replaces the two-norm regularizer with 1-norm that encourages some of the coordinates to be set exactly to zero
- Iterative subset selection
  - iteratively add (or prune) coordinates based on their impact on the (training) error

# Information analysis

- Suppose the feature vector is just the input vector x whose coordinates take values in {1,...,k}
- Given a training set of size n, we can evaluate an empirical estimate of the mutual information between the coordinate values and the binary label

$$\hat{I}(Y, X_i) = \sum_{y \in \{-1,1\}} \sum_{x_i=1}^{k} \hat{P}(y, x_i) \log \frac{\hat{P}(y, x_i)}{\hat{P}(y)\hat{P}(x_i)}$$

$$\hat{P}(y, x_i) = \frac{1}{n} \sum_{t=1}^{n} \delta(y, y_t)\underline{\delta(x_i, x_{it})}$$

$$\begin{cases} 1, & \text{if } x_i = x_{it} \\ 0, & \text{otherwise} \end{cases}$$

empirical estimates

$$\hat{P}(y) = \frac{1}{n} \sum_{t=1}^{n} \delta(y, y_t)$$

$$\hat{P}(x_i) = \frac{1}{n} \sum_{t=1}^{n} \delta(x_i, x_{it})$$

# Information analysis

- Suppose the feature vector is just the input vector x whose coordinates take values in {1,...,k}
- Given a training set of size n, we can evaluate an empirical estimate of the mutual information between the coordinate values and the binary label

$$\hat{I}(Y, X_i) = \sum_{y \in \{-1,1\}} \sum_{x_i=1}^{k} \hat{P}(y, x_i) \log \frac{\hat{P}(y, x_i)}{\hat{P}(y)\hat{P}(x_i)}$$

  - provides a ranking of the features to include
  - weights redundant features equally (would include neither or both)
  - not tied to the linear classifier (may select features that the linear classifier cannot use, or omit combinations of features particularly useful in a linear classifier)

# Information analysis

| Y | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| -1 | 1 | 1 | 1 |
| -1 | 1 | 2 | 1 |
| -1 | 2 | 2 | 1 |
| -1 | 2 | 3 | 2 |
| +1 | 1 | 1 | 2 |
| +1 | 1 | 1 | 2 |
| +1 | 1 | 3 | 2 |
| +1 | 2 | 3 | 2 |

| Y | $\hat{P}(Y)$ |
|---|---|
| -1 | 4/8 |
| +1 | 4/8 |

| $X_2$ | $\hat{P}(X_2)$ |
|---|---|
| 1 | 3/8 |
| 2 | 2/8 |
| 3 | 3/8 |

| Y | $X_2$ | $\hat{P}(Y, X_2)$ |
|---|---|---|
| -1 | 1 | 1/8 |
| -1 | 2 | 2/8 |
| -1 | 3 | 1/8 |
| +1 | 1 | 2/8 |
| +1 | 2 | 0 |
| +1 | 3 | 2/8 |

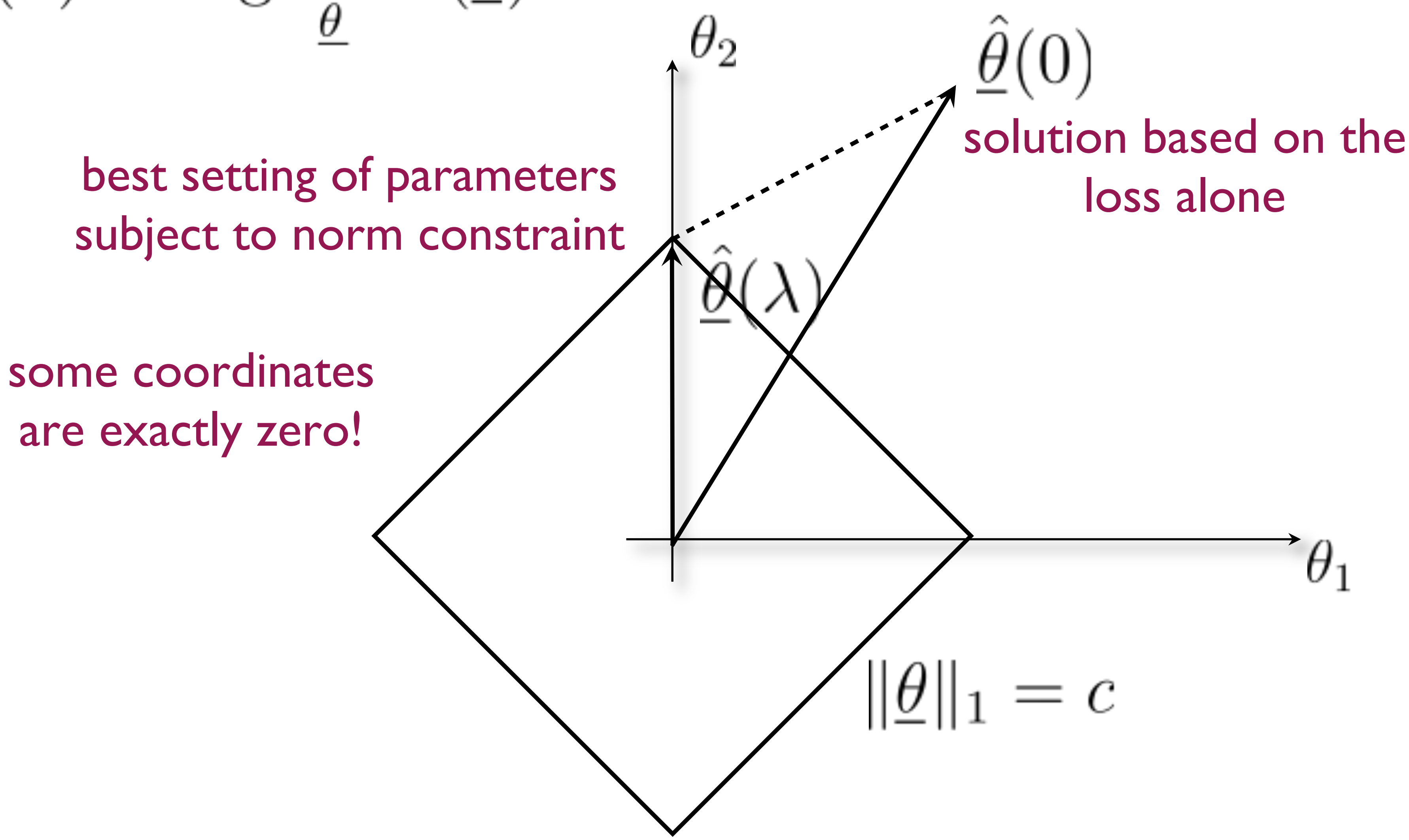
$$\begin{aligned}\hat{I}(Y, X_2) = {} & (1/8) \log [(1/8) / \{(4/8)(3/8)\}] + (2/8) \log [(2/8) / \{(4/8)(2/8)\}] \\ & + (1/8) \log [(1/8) / \{(4/8)(3/8)\}] + (2/8) \log [(2/8) / \{(4/8)(3/8)\}] \\ & + (2/8) \log [(2/8) / \{(4/8)(3/8)\}]\end{aligned}$$

# Regularization approach (Lasso)

- By using a 1-norm regularizer we will cause some of parameters to be set exactly to zero

$$J(\underline{\theta}) = \frac{1}{2}\sum_{i=1}^{n}\left(y_t - \underline{\theta}\cdot\phi(\underline{x}_i)\right)^2 + \lambda\|\underline{\theta}\|_1 \qquad \text{where} \qquad \|\underline{\theta}\|_1 = \sum_{i=1}^{d}|\theta_i|$$

$$\underline{\hat{\theta}}(\lambda) = \underset{\underline{\theta}}{\operatorname{argmin}}\, J(\underline{\theta})$$



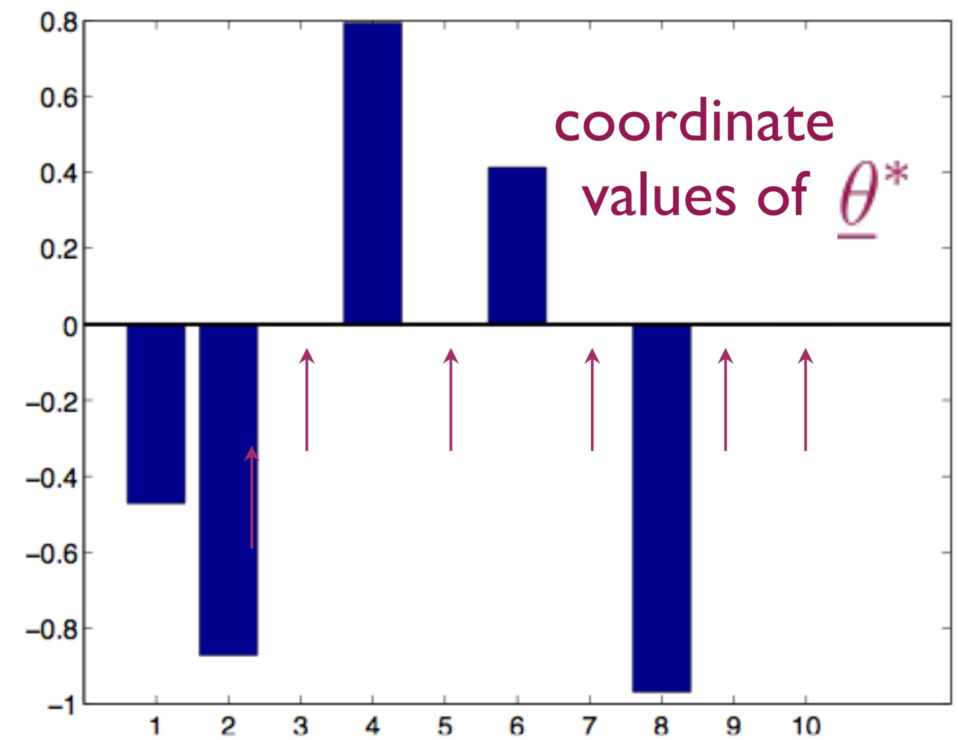
# Regularization example

- n=100 samples, d=10, training outputs generated from

$$y_t \sim N\left(\underline{\theta}^* \cdot \underline{x}_t,\, \sigma^2\right),\ t = 1, \ldots, n$$

- If we increase the regularization penalty, we get fewer non-zero parameters in the solution to

$$J(\underline{\theta}) = \frac{1}{2}\sum_{i=1}^{n}\left(y_t - \underline{\theta}\cdot\underline{\phi}(\underline{x}_i)\right)^2 + \lambda\|\underline{\theta}\|_1$$



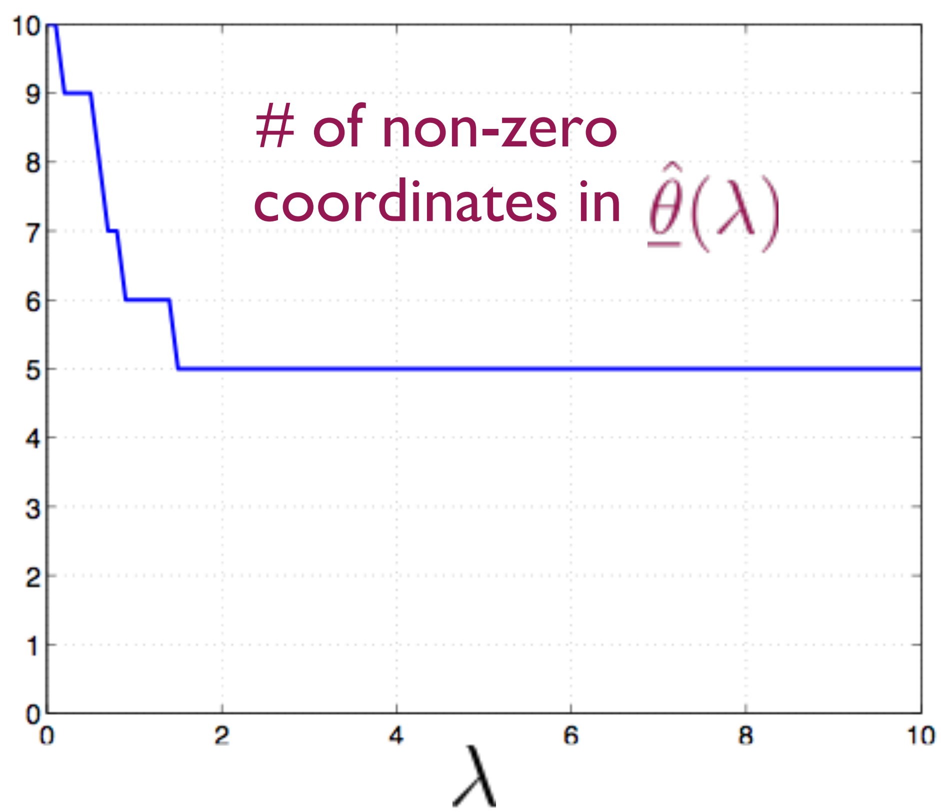
# Regularization example

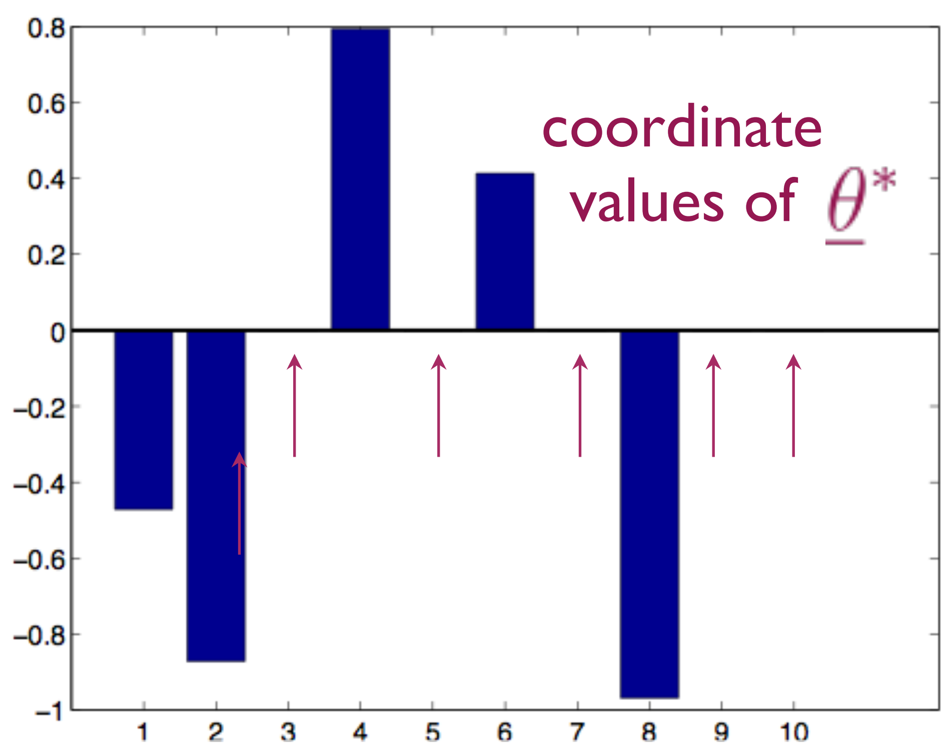
- n=100 samples, d=10, training outputs generated from

$$y_t \sim N\left(\underline{\theta}^* \cdot \underline{x}_t,\, \sigma^2\right),\; t = 1, \ldots, n$$

- If we increase the regularization penalty, we get fewer non-zero parameters in the solution to

$$J(\underline{\theta}) = \frac{1}{2}\sum_{i=1}^{n}\left(y_t - \underline{\theta}\cdot\phi(\underline{x}_i)\right)^2 + \lambda\|\underline{\theta}\|_1$$

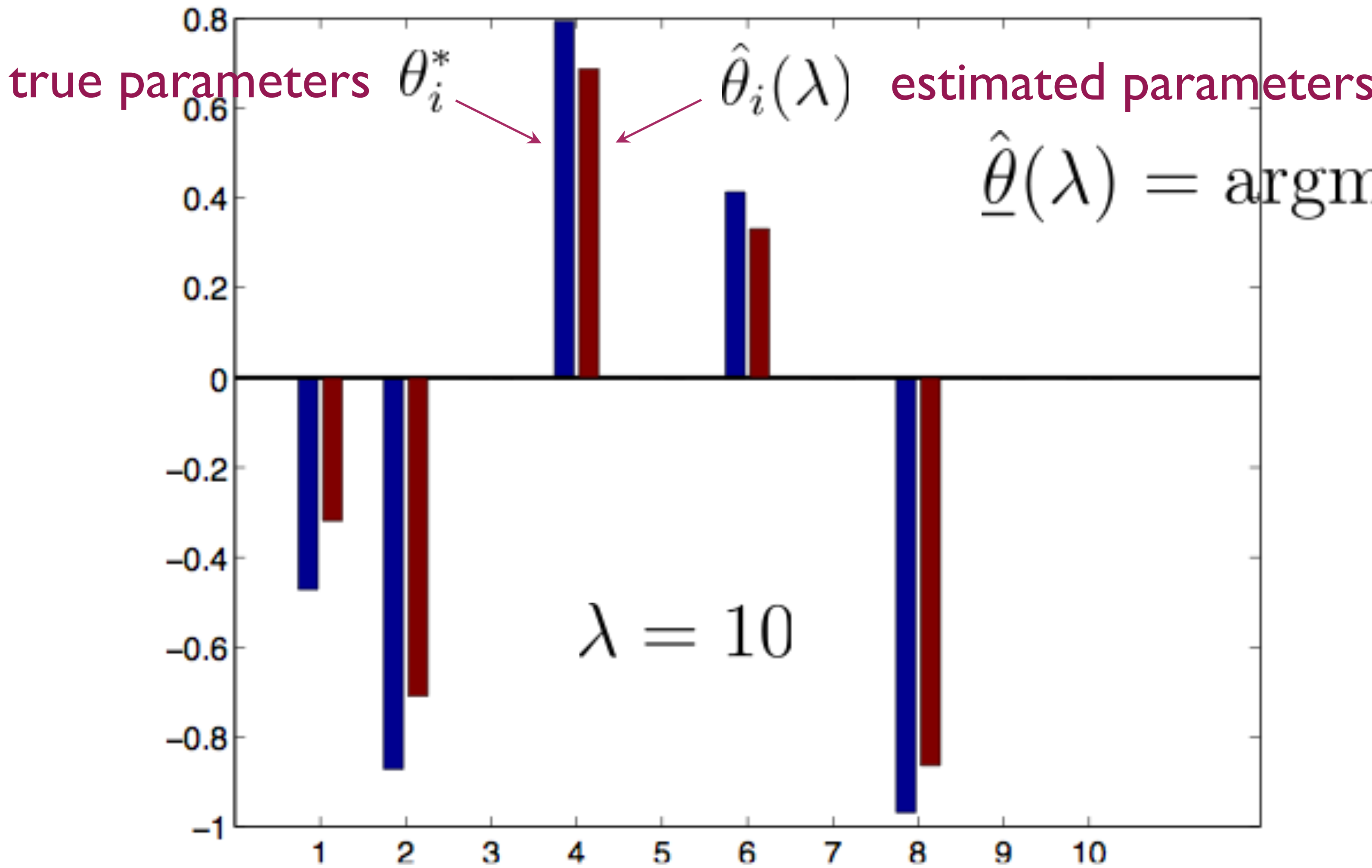


# Regularization example

- The 1-norm penalty term controls the number of non-zero coordinates but also reduces the magnitude of the coordinates

$$J(\underline{\theta}) = \frac{1}{2}\sum_{i=1}^{n}\left(y_t - \underline{\theta}\cdot\phi(\underline{x}_i)\right)^2 + \lambda\|\underline{\theta}\|_1$$




$$\hat{\underline{\theta}}(\lambda) = \underset{\underline{\theta}}{\operatorname{argmin}}\, J(\underline{\theta})$$

# Subset selection

- A greedy algorithm for finding a good subset of feature coordinates (feature functions) to rely on

$$\phi_1(\underline{x}), \ldots, \phi_d(\underline{x}) \qquad \underline{\phi}_S(\underline{x}) = \{\phi_j(\underline{x})\}_{j \in S}$$

- For instance:

$$S = \{7, 10, 29\} \qquad \underline{\phi}_S(\underline{x}) = \begin{bmatrix} \phi_7(\underline{x}) \\ \phi_{10}(\underline{x}) \\ \phi_{29}(\underline{x}) \end{bmatrix} \in \mathbb{R}^3$$

$$\underline{\theta}_S = \begin{bmatrix} \theta_7 \\ \theta_{10} \\ \theta_{29} \end{bmatrix} \in \mathbb{R}^3$$

# Subset selection

- A greedy algorithm for finding a good subset of feature coordinates (feature functions) to rely on

$$\phi_1(\underline{x}), \ldots, \phi_d(\underline{x}) \qquad \underline{\phi}_S(\underline{x}) = \{\phi_j(\underline{x})\}_{j \in S}$$

$$\text{for each subset } S,\ |S| = k, \text{ evaluate}$$

$$J(S) = \min_{\underline{\theta}_S} \frac{1}{2} \sum_{t=1}^{n} \left(y_t - \underline{\theta}_S \cdot \underline{\phi}_S(\underline{x}_t)\right)^2$$

each feature subset is assessed on the basis of the resulting training error

$$\hat{S} = \operatorname*{argmin}_S J(S)$$

find the best subset

- k is used for (statistical) complexity control
- computationally hard (exponential in k)

# Greedy subset selection

- A greedy algorithm for finding a good subset of feature coordinates (feature functions) to rely on

$$\phi_1(\underline{x}), \ldots, \phi_d(\underline{x}) \qquad \underline{\phi}_S(\underline{x}) = \{\phi_j(\underline{x})\}_{j \in S}$$

$$S = \emptyset$$

repeat until $|S| = k$

for each $j \notin S$ evaluate

try each new coordinate

$$J(S \cup j) = \min_{\underline{\theta}_{S\cup j}} \frac{1}{2} \sum_{t=1}^{n} \left( y_t - \underline{\theta}_{S\cup j} \cdot \underline{\phi}_{S\cup j}(\underline{x}_t) \right)^2$$

re-estimate all the parameters in the context of the new coordinate

$$\hat{j} = \operatorname*{argmin}_{j \notin S} J(S \cup j)$$

find the best coordinate to include

$$S \leftarrow S \cup \{\hat{j}\}$$

- each new feature is assessed in the context of those already included
- the method is not guaranteed to find the optimal subset

# Forward-fitting

- We can also choose feature coordinates without re-estimating the parameters associated with already included coordinates

$$\phi_1(\underline{x}), \ldots, \phi_d(\underline{x}) \qquad \underline{\phi}_S(\underline{x}) = \{\phi_j(\underline{x})\}_{j \in S}$$

$$S = \emptyset, \quad \underline{\hat{\theta}}_{\emptyset} = 0$$

repeat until $|S| = k$

for each $j$ evaluate

fixed at this stage

$$J(\underline{\hat{\theta}}_S, j) = \min_{\theta_j} \frac{1}{2} \sum_{t=1}^{n} \Big( y_t - \underline{\hat{\theta}}_S \cdot \underline{\phi}_S(\underline{x}_t) - \underline{\theta_j \phi_j(\underline{x}_t)} \Big)^2$$

re-estimate only the parameter associated with the new coordinate

$$\hat{j} = \arg\min_{j} J(\underline{\hat{\theta}}_S, j)$$

select the best new feature

$$\underline{\hat{\theta}}_{S \cup j} = \{\underline{\hat{\theta}}_S, \hat{\theta}_{\hat{j}}\}, \quad S \leftarrow S \cup \{\hat{j}\},$$

- same feature may be included more than once

# Myopic forward fitting

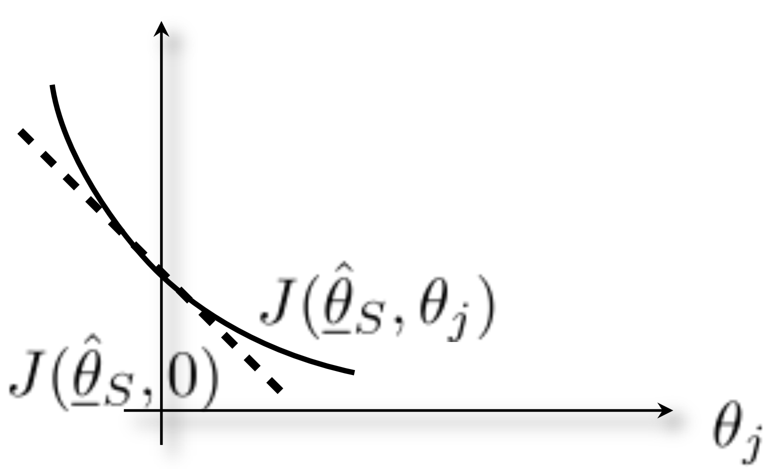
- We can also identify new features in a more limited way by focusing on how much “potential” each feature has for reducing the training error

$$\phi_1(\underline{x}), \ldots, \phi_d(\underline{x}) \qquad \underline{\phi}_S(\underline{x}) = \{\phi_j(\underline{x})\}_{j \in S}$$

$$J(\hat{\underline{\theta}}_S, \theta_j) = \frac{1}{2} \sum_{t=1}^{n} \left( y_t - \hat{\underline{\theta}}_S \cdot \underline{\phi}_S(\underline{x}_t) - \theta_j \phi_j(\underline{x}_t) \right)^2$$

$$DJ(\hat{\underline{\theta}}_S, j) = \frac{\partial J(\hat{\underline{\theta}}_S, \theta_j)}{\partial \theta_j}\Big|_{\theta_j = 0} = - \sum_{t=1}^{n} \left( y_t - \hat{\underline{\theta}}_S \cdot \underline{\phi}_S(\underline{x}_t) \right) \phi_j(\underline{x}_t)$$

|derivative| is large if the corresponding feature would have a large effect on the error



# Myopic forward fitting

- We can identify new features with minimal fitting

$$\phi_1(\underline{x}), \ldots, \phi_d(\underline{x}) \qquad \underline{\phi}_S(\underline{x}) = \{\phi_j(\underline{x})\}_{j \in S}$$

$$S = \emptyset, \quad \underline{\hat{\theta}}_\emptyset = 0$$

repeat until $|S| = k$

for each $j$ evaluate

fixed at this stage

$$DJ(\underline{\hat{\theta}}_S, j) = \frac{\partial J(\underline{\hat{\theta}}_S, \theta_j)}{\partial \theta_j}\Big|_{\theta_j = 0}$$

the criterion does not involve any parameter fitting

$$\hat{j} = \underset{j}{\operatorname{argmax}} \, |DJ(\underline{\hat{\theta}}_S, j)|$$

selection of best coordinate

$$\hat{\theta}_{\hat{j}} = \underset{\theta_{\hat{j}}}{\operatorname{argmin}} \, J(\underline{\hat{\theta}}_S, \theta_{\hat{j}})$$

estimate only the parameter associated with the selected feature

$$\underline{\hat{\theta}}_{S \cup \hat{j}} = \{\underline{\hat{\theta}}_S, \hat{\theta}_{\hat{j}}\}, \quad S \leftarrow S \cup \{\hat{j}\},$$

# Forward-fitting example

- 1 dimensional polynomial regression

$$\underline{\phi}(x) = [1, x, x^2, x^3, x^4]^T \qquad \underline{\phi}_S(\underline{x}) = \{\phi_j(\underline{x})\}_{j \in S}$$

$$J(\underline{\hat{\theta}}_S, \theta_j) = \frac{1}{2} \sum_{t=1}^{n} \left( y_t - \underline{\hat{\theta}}_S \cdot \underline{\phi}_S(\underline{x}_t) - \theta_j \phi_j(\underline{x}_t) \right)^2$$

| iter | deg | $\hat{\theta}_{\hat{j}}$ | $J(\underline{\hat{\theta}})$ |
|---|---|---|---|
| 1 | 0 | $+1.089$ | 0.874 |
| 2 | 1 | $-0.553$ | 0.163 |
| 3 | 2 | $-0.288$ | 0.085 |
| 4 | 1 | $-0.101$ | 0.062 |
| 5 | 0 | $-0.033$ | 0.051 |
| 6 | 3 | $+0.053$ | 0.049 |
| 7 | 1 | $-0.043$ | 0.045 |
| 8 | 3 | $+0.088$ | 0.041 |
| 9 | 1 | $-0.035$ | 0.039 |
| 10 | 3 | $+0.072$ | 0.036 |

forward-fitting

| iter | deg | $\hat{\theta}_{\hat{j}}$ | $J(\underline{\hat{\theta}})$ |
|---|---|---|---|
| 1 | 0 | $+1.089$ | 0.874 |
| 2 | 1 | $-0.553$ | 0.163 |
| 3 | 0 | $-0.085$ | 0.091 |
| 4 | 1 | $-0.056$ | 0.084 |
| 5 | 2 | $-0.127$ | 0.069 |
| 6 | 0 | $+0.021$ | 0.065 |
| 7 | 1 | $-0.031$ | 0.063 |
| 8 | 2 | $-0.078$ | 0.057 |
| 9 | 0 | $+0.013$ | 0.055 |
| 10 | 1 | $-0.018$ | 0.054 |

myopic forward-fitting

# Forward-fitting example

- 1 dimensional polynomial regression

$$\phi(x) = [1, x, x^2, x^3, x^4]^T$$