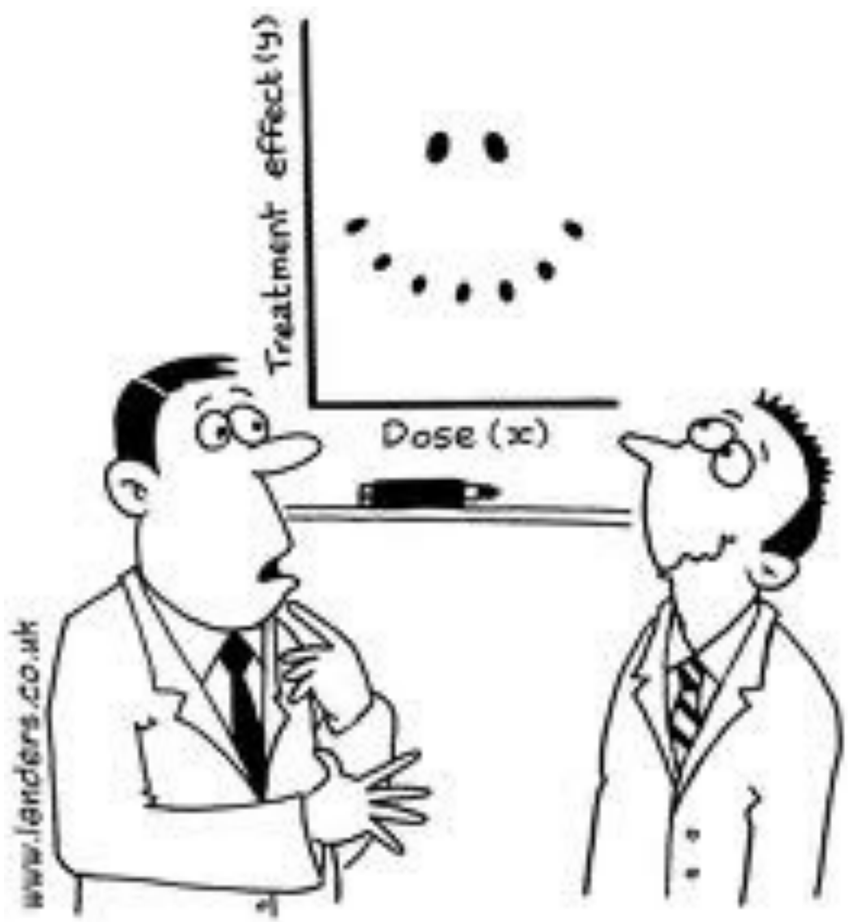
Linear regression



"It's a non-linear pattern with outliers.....but for some reason I'm very happy with the data."

Reminders

- Thought questions should be submitted on eclass
- Please list the section related to the thought question
  - If it is a more general, open-ended question not exactly related to a section, label the question with a topic (e.g., Picking models)

Properties of distributions

- Mean is the expected value ($E[X]$)
- Mode is the most likely value (i.e., x with largest $p(x)$)
- Median m is the value such that X is equally likely to fall above or below m: $P(X \leq m) = P(X \geq m)$
  - When we use a squared-error cost, obtain $f(x) = E[Y \mid x]$
  - If we use an absolute-error cost, obtain $f(x) = \text{median} (p (y \mid x))$

Summary of optimal models

- Expected cost introduced to formalize our objective

- Bayes risk function indicates best we could do
  - f(x) specified for each x, rather than having some simpler (continuous) function class
  - can think of it as a table of values

- For classification (with uniform cost)

$$f^*(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} \{p(y|\mathbf{x})\}.$$

- For regression (with squared-error cost)

$$f^*(\mathbf{x}) = \int_{\mathcal{Y}} y p(y|\mathbf{x}) dy$$

Learning functions

- Hypothesize a functional form, e.g.

$$f(\mathbf{x}) = \sum_{j=1}^{d} w_j x_j$$

$$f(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2$$

$$f(x_1, x_2) = w x_1 x_2$$

$$\vdots$$

- Then need to find the “best” parameters for this function; we will find the parameters that best approximate E[y I x]

Exercise: Reducible error

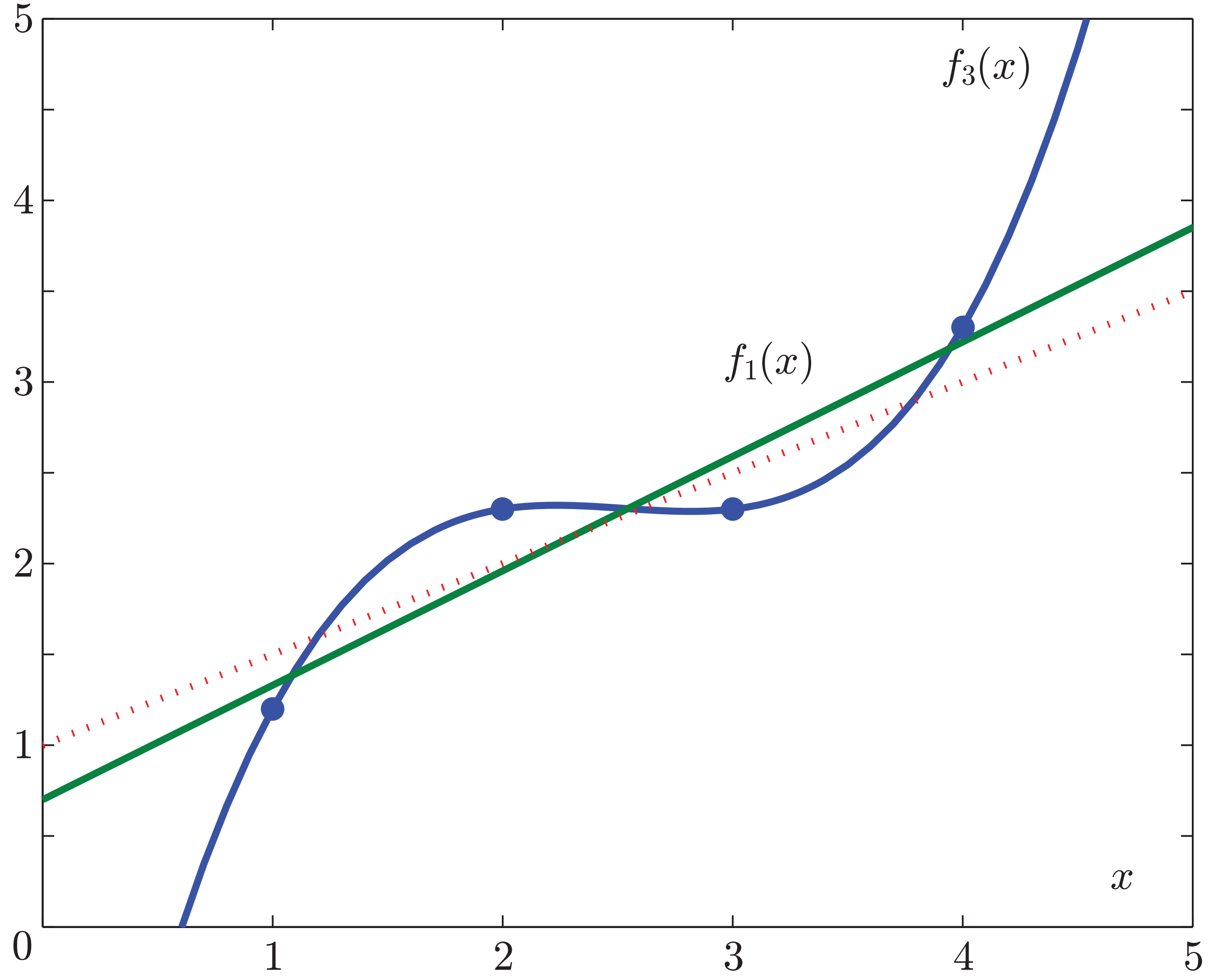
- Can $f(\mathbf{x}) = \sum_{j=1}^{d} w_j x_j$ always represent E[Y | x]?

- No. Imagine y $= w x_1 x_2$

- This is deterministic, so there is enough information in x to predict y
  - i.e., the stochasticity is not the problem, have zero irreducible error

- Rather simplistic functional form means we cannot predict y

Linear versus polynomial function



Linear Regression

$$f(x) = w_0 + w_1 x$$

e.g.,
x_i = size of house
y_i = cost of house

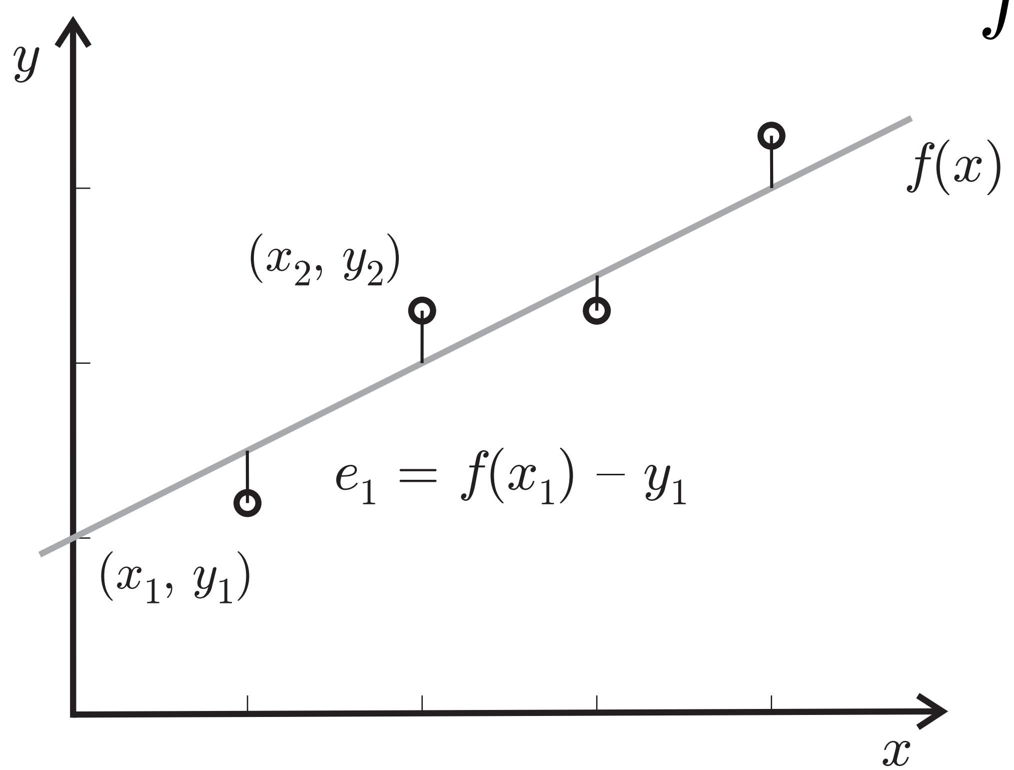
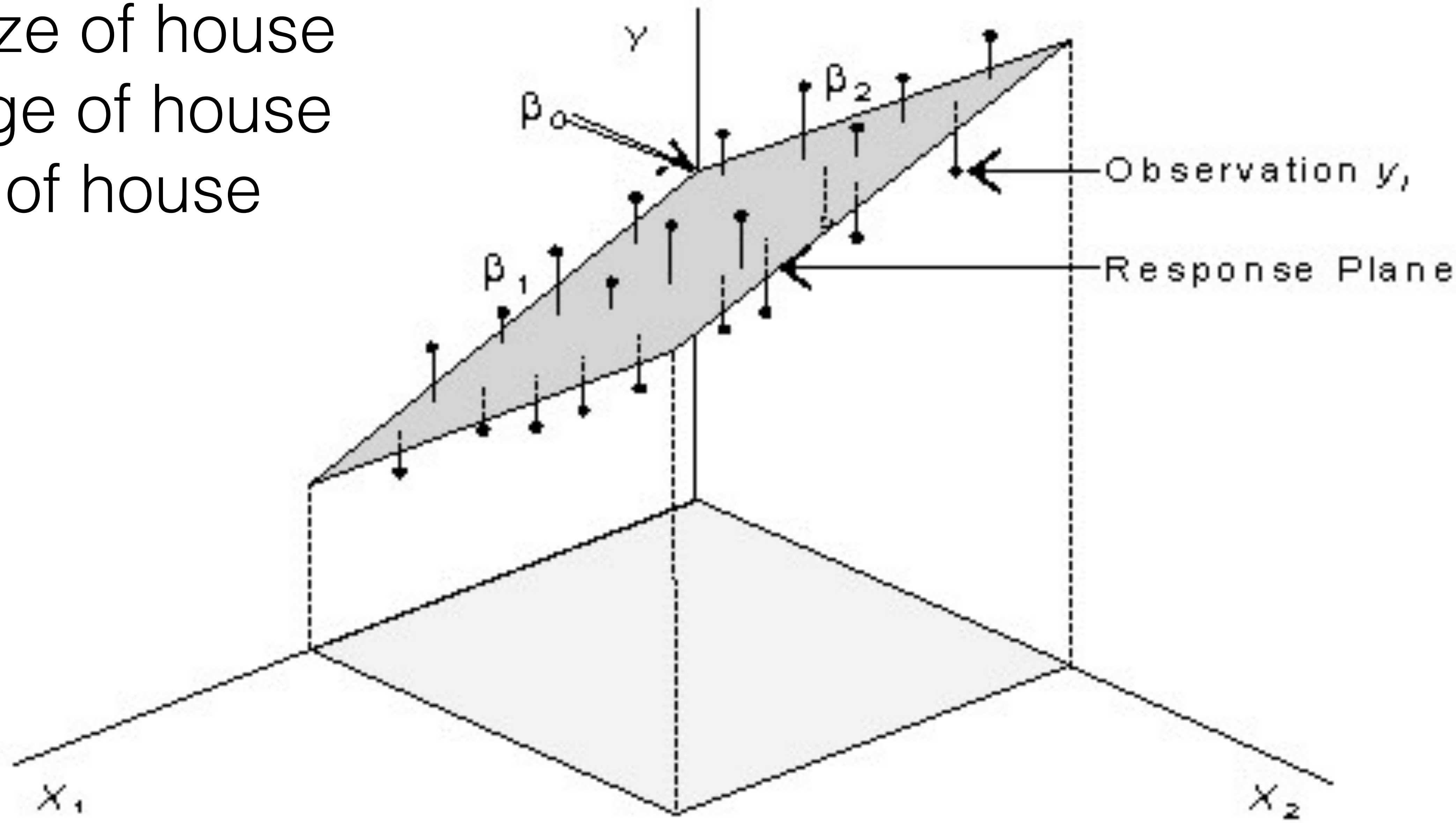


Figure 4.1: An example of a linear regression fitting on data set $\mathcal{D} = \{(1, 1.2), (2, 2.3), (3, 2.3), (4, 3.3)\}$. *The task of the optimization process is to find the best linear function* $f(x) = w_0 + w_1 x$ *so that the sum of squared errors* $e_1^2 + e_2^2 + e_3^2 + e_4^2$ *is minimized.*

(Multiple) Linear Regression

e.g.,
x_{i1} = size of house
x_{i2} = age of house
y_i = cost of house



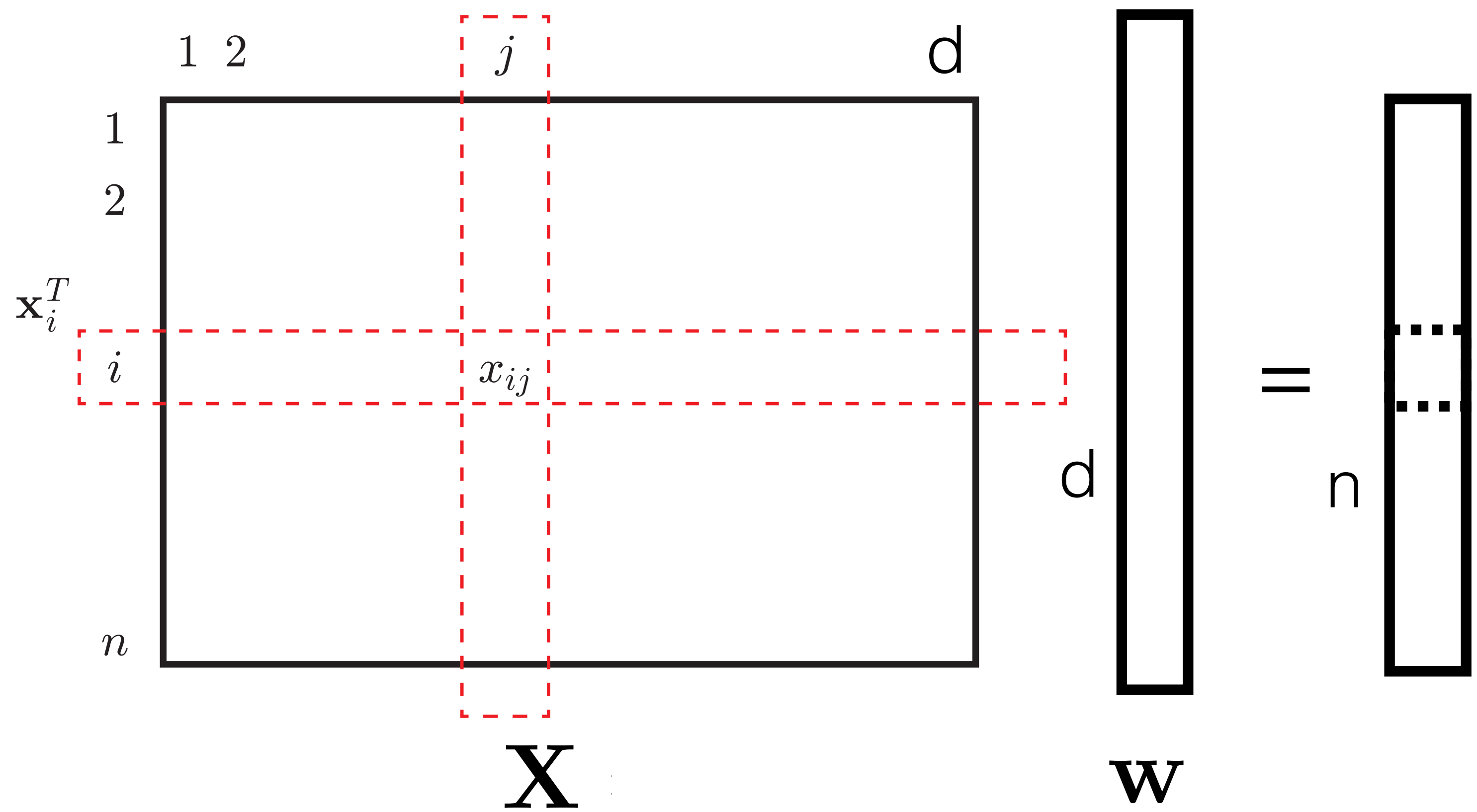
Linear regression importance

- Many other techniques will use linear weighting of features
  - including neural networks
- Often, we will add non-linearity using
  - non-linear transformations of linear weighting
  - non-linear transformations of features
- Becoming comfortable will linear weightings, for multiple inputs and outputs, is important

Example: regression

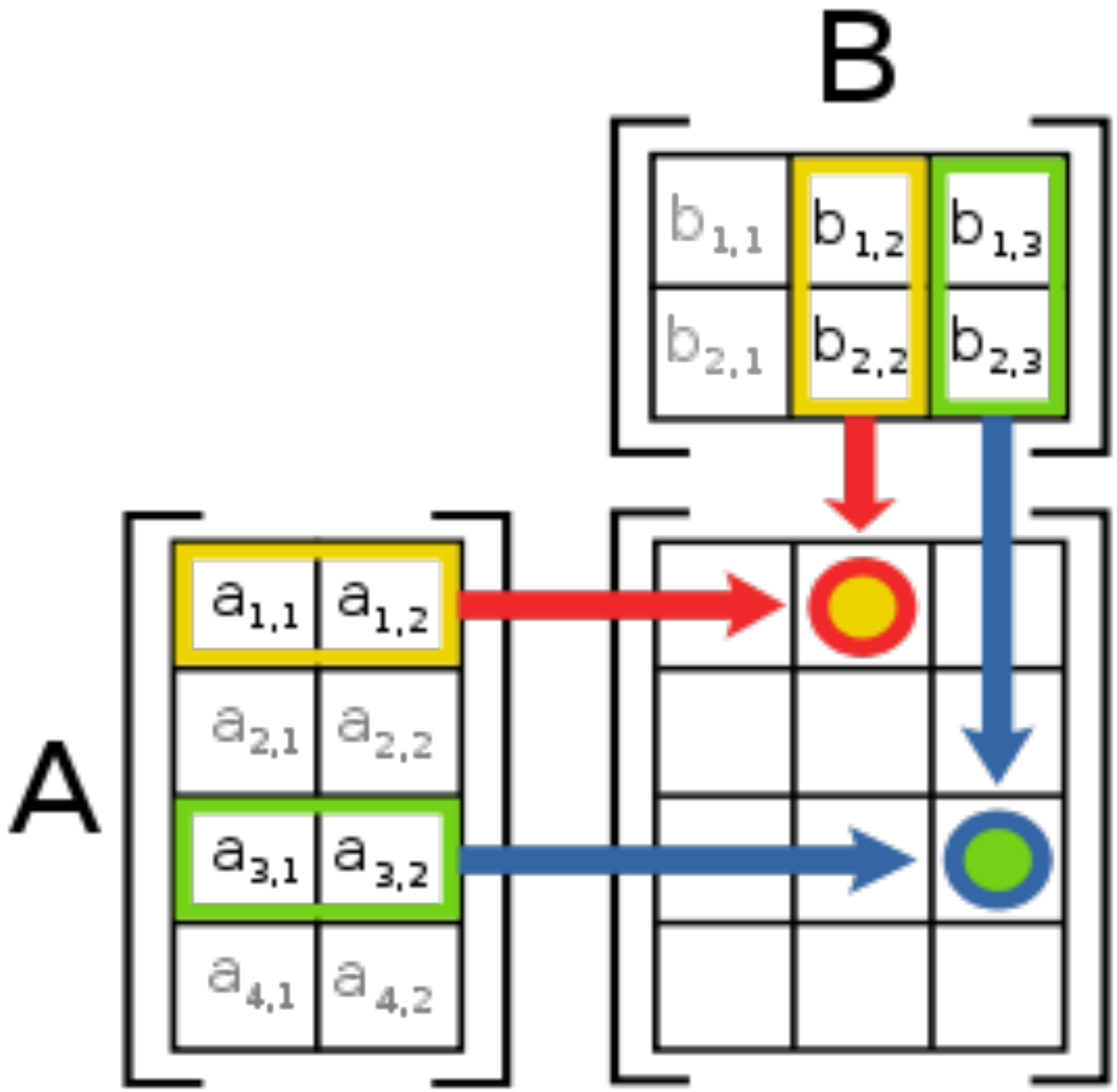
Example 11: Consider again data set $\mathcal{D} = \{(1, 1.2), (2, 2.3), (3, 2.3), (4, 3.3)\}$

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 1.2 \\ 2.3 \\ 2.3 \\ 3.3 \end{bmatrix},$$



$$\hat{y}_i = \mathbf{x}_i^\top \mathbf{w}$$

Matrix multiplication



Whiteboard

- Maximum likelihood formulation (and assumptions)
- Solving the optimization
- Weighted error functions, if certain data points “matter” more than others
- Predicting multiple outputs (multivariate y)

Comments (Sep 26, 2017)

- Assignment 1 due on Thursday
- More review of linear algebra today

If she loves you more each and every day,
by linear regression she hated you before you met.

$$\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$$

SVD

$$\mathbf{M}\mathbf{x} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\mathbf{x} = \mathbf{U}\boldsymbol{\Sigma}(\mathbf{V}^\top\mathbf{x})$$

What we’ve done so far

- Discussed linear regression
  - goal: obtain weights w such that < x, w> approximates E[Y | x]
- Discussed maximum likelihood formulation
- Solution: $\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ $\quad\quad\quad \hat{\mathbf{y}} = \mathbf{X}\mathbf{w}^*$
- Starting discussing the properties of the solution
  - When is it stable?
  - Today: What does this mean for accuracy in predicting on new data?

Example: OLS

Example 11: Consider again data set $\mathcal{D} = \{(1, 1.2), (2, 2.3), (3, 2.3), (4, 3.3)\}$

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \; \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \; \mathbf{y} = \begin{bmatrix} 1.2 \\ 2.3 \\ 2.3 \\ 3.3 \end{bmatrix},$$

In Matlab, can compute

1. $\mathbf{X}^\top \mathbf{X}$
2. $(\mathbf{X}^\top \mathbf{X})^{-1}$
3. $(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$

$$\mathbf{X}^\top \mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}$$

What if we did not add the column of 1s?

Whiteboard

- More about inverses of matrices
- Refresh about stability of the solution
- Using regularization to fix the problem
- Properties of solution:
  - Bias (and underfitting)
  - Variance (and overfitting)

Overfitting

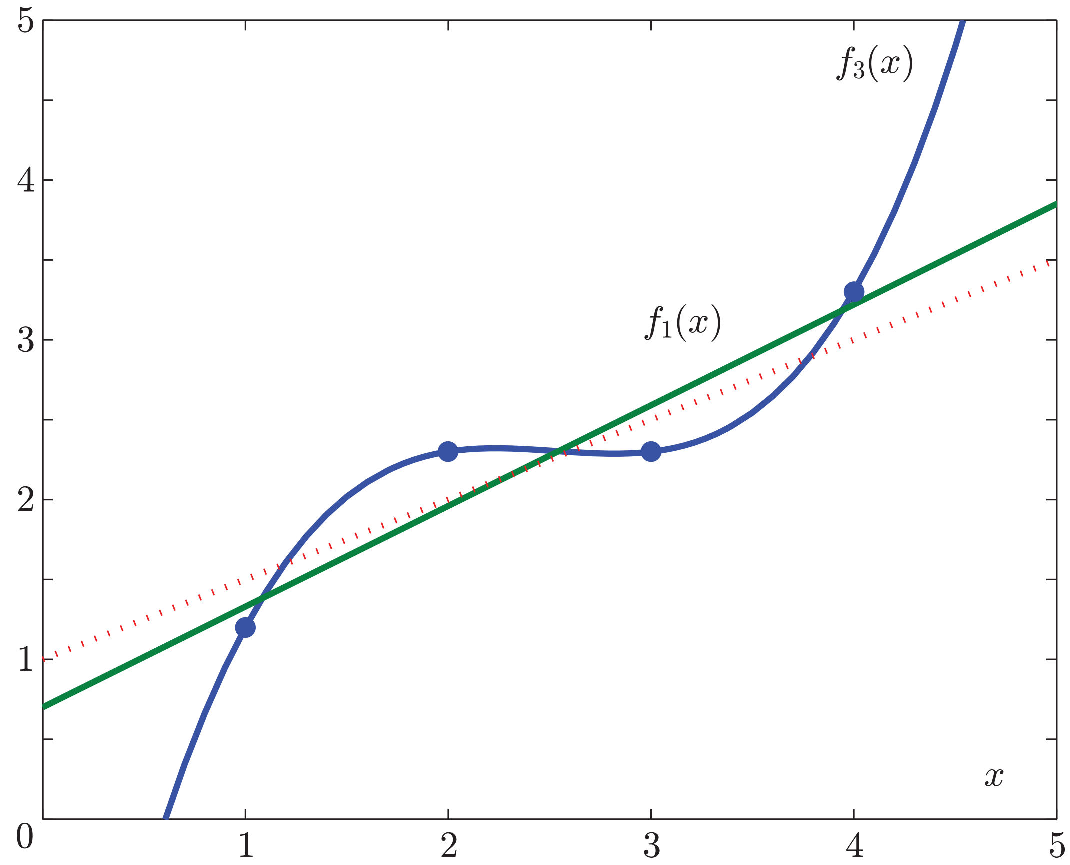


Figure 4.4: Example of a linear vs. polynomial fit on a data set shown in Figure 4.1. The linear fit, $f_1(x)$, is shown as a solid green line, whereas the cubic polynomial fit, $f_3(x)$, is shown as a solid blue line. The dotted red line indicates the target linear concept.

$$\mathbf{w}_1^* = (0.7, 0.63)$$

$$\mathbf{w}_3^* = (-3.1, 6.6, -2.65, 0.35)$$

Comments (Sep 28, 2017)

- Assignment 1 due today
- Need matplotlib for simulate.py
- Today: finish-off bias-variance

Terminology clarification

- What is a parameter? Any coefficients (i.e., scalars, vectors or matrices) that define the function you care about

- e.g., a and b are both parameters for function f

$$f(x) = \begin{cases} a & \text{if } x < 0 \\ b & \text{if } x \geq 0 \end{cases}$$

- Maximum likelihood solution: parameters for the pmf or pdf that make the data the most likely
  - The following function is not the maximum likelihood solution

$$f(x) = \arg\max_{y \in \mathcal{Y}} p(y|x)$$

Linear regression for non-linear problems

e.g. $\quad f(x) = w_0 + w_1 x, \longrightarrow f(x) = \sum_{j=0}^{p} w_j x^j,$

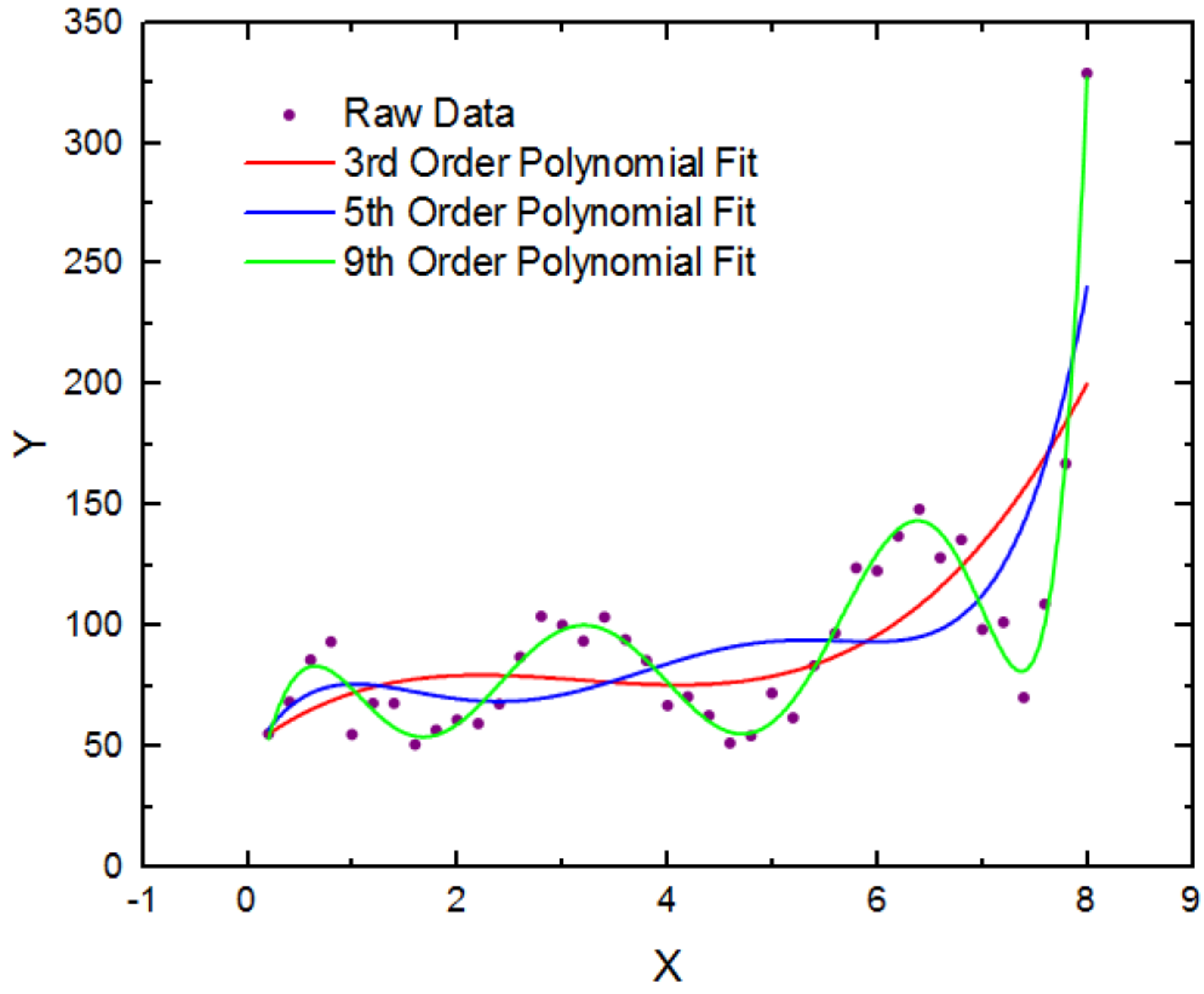
e.g. $f(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2$

$$
\begin{array}{cc}
 & \mathbf{X} \\
1 & \begin{array}{|c|}\hline x_1 \\ \hline x_2 \\ \hline \ldots \\ \hline x_n \\ \hline \end{array} \\
n &
\end{array}
\quad \rightarrow \quad
\begin{array}{c}
\mathbf{\Phi} \\
\begin{array}{|c|c|c|}\hline \phi_0(x_1) & \ldots & \phi_p(x_1) \\ \hline \ldots & \ldots & \ldots \\ \hline \ldots & \ldots & \ldots \\ \hline \phi_0(x_n) & \ldots & \phi_p(x_n) \\ \hline \end{array}
\end{array}
$$

Figure 4.3: Transformation of an $n \times 1$ data matrix $\mathbf{X}$ into an $n \times (p+1)$ matrix $\mathbf{\Phi}$ using a set of basis functions ϕ_j, $j = 0, 1, \ldots, p$.

$$\mathbf{w}^* = \left(\mathbf{\Phi}^\top \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^\top \mathbf{y}.$$

Polynomial representations

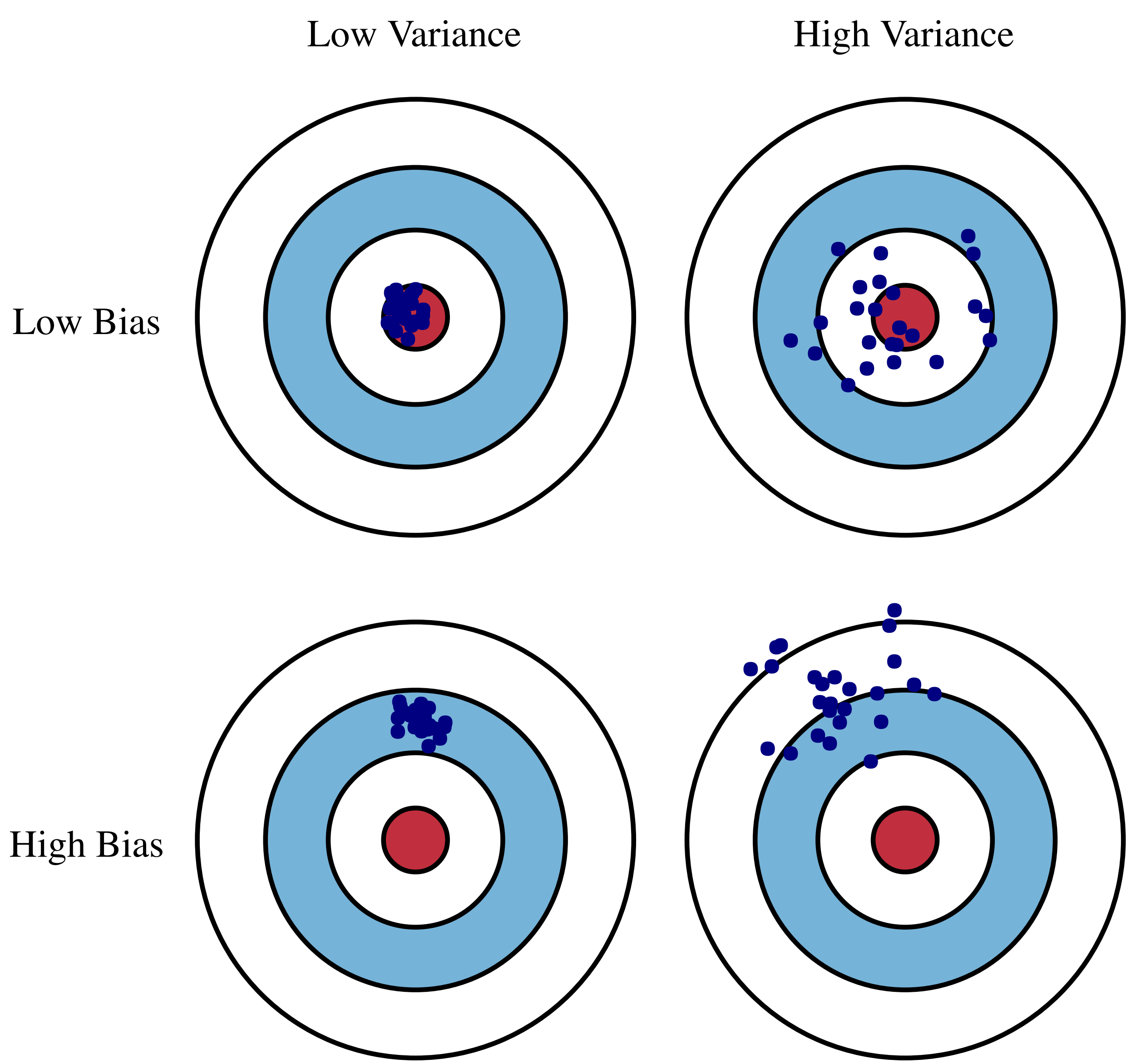


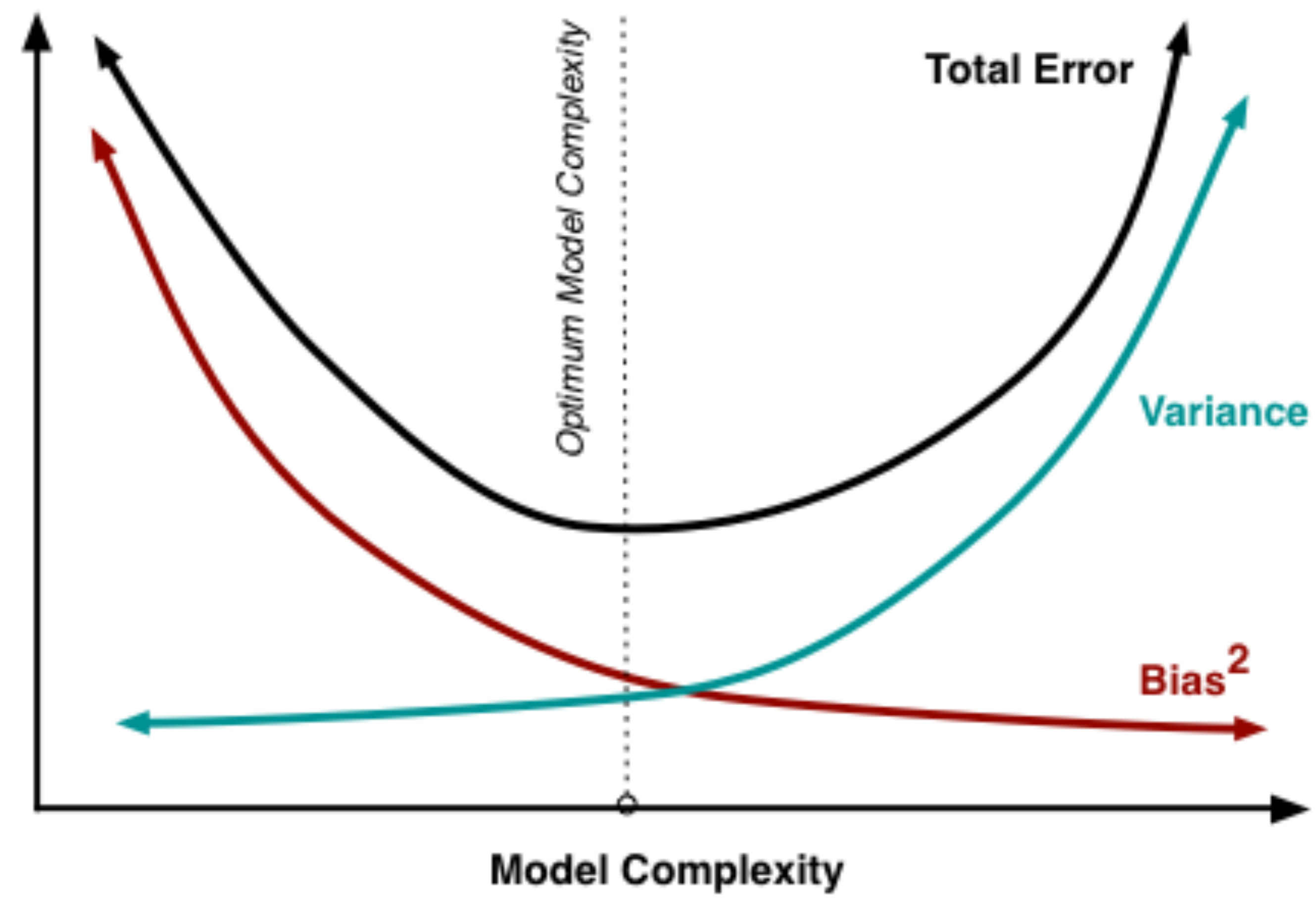
$$w_0 + w_1 x^1 + w_2 x^2 + \ldots + w_9 x^9$$

Whiteboard

- Bias and variance of linear regression solutions

Bias-variance trade-off





*Nice images from: http://scott.fortmann-roe.com/docs/BiasVariance.html

Example: regularization and bias

- Picked a Gaussian prior and obtained l2 regularization
- We discussed the bias of this regularization
  - no regularization was unbiased E[w] = true w
  - with regularization meant E[w] was not equal to the true w
- Previously, however, mentioned that MAP and ML converge to the same estimate
- Does that happen here? $\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$

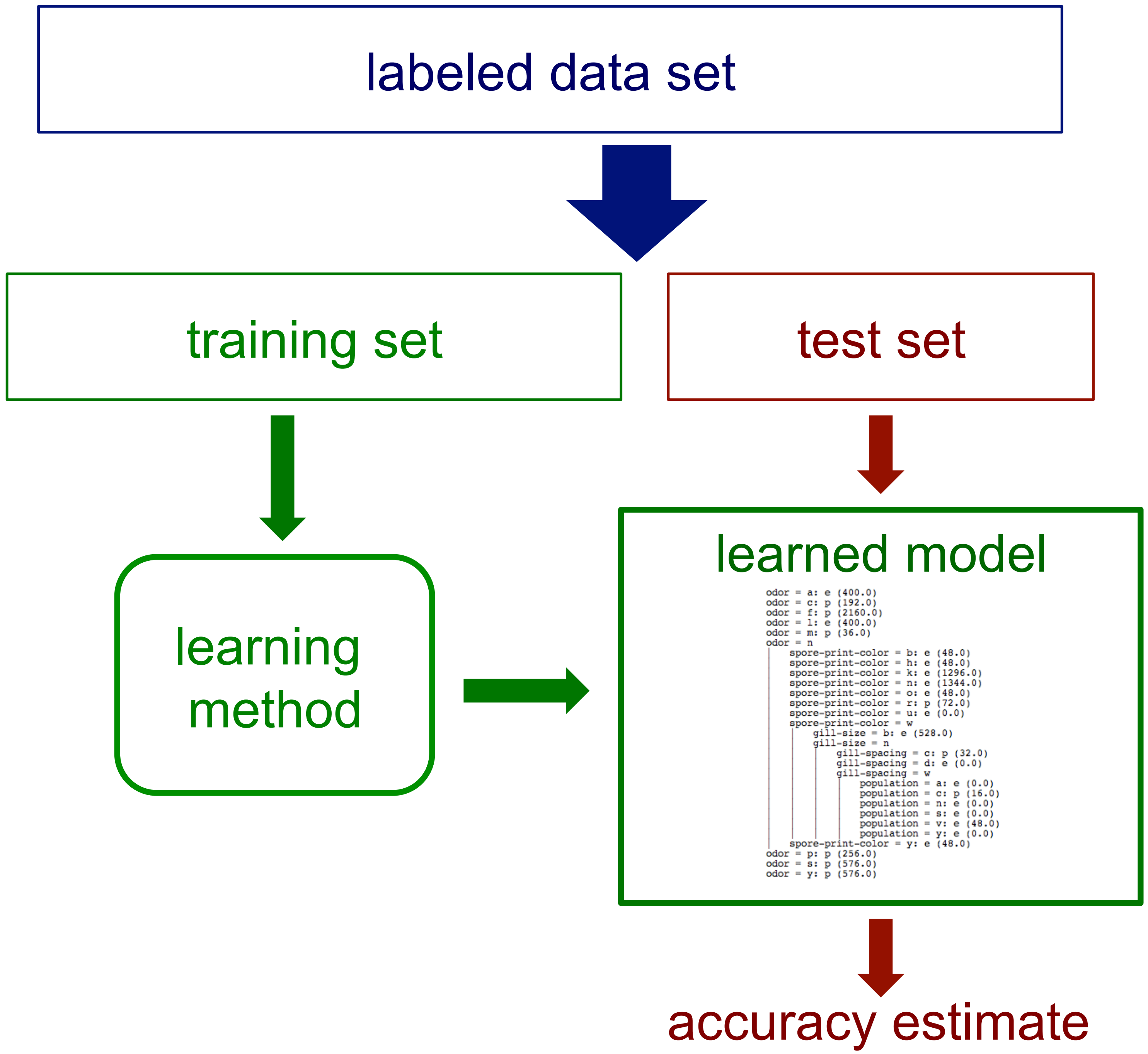
How do we pick lambda?

- Discussed goal to minimize bias-variance trade-off
  - i.e., minimizing MSE
- But, this involves knowing the true w!
- Recall our actual goal: learn w to get good prediction accuracy on new data
  - Called generalization error
- Alternative to directly minimize MSE: use data to determine which choice of lambda provides good prediction accuracy

How can we tell if its a good model?

- What if you train many different models on a batch of data, check their accuracy on that data, and pick the best one?
  - Imagine your are predicting how much energy your appliances will use today
  - You train your models on all previous data for energy use in your home
  - How well will this perform in the real world?
- What if the models you are testing are only different in terms of the regularization parameter lambda that they use? What will you find?

Simulating generalization error



Simulating generalization error

- Now we have one model, trained similarly to how it will be trained, and a measure of accuracy on new data (but distributed identically to trained data)

- What if we pick the model with the best test accuracy? Any issues?

Picking other priors

- Picked Gaussian prior on weights
  - Encodes that we want the weights to stay near zero, varying with at most 1/lambda
- What if we had picked a different prior?
  - e.g., the Laplace prior?

$$\frac{1}{2b}\exp(-|x-\mu|/b)$$

Regularization intuition

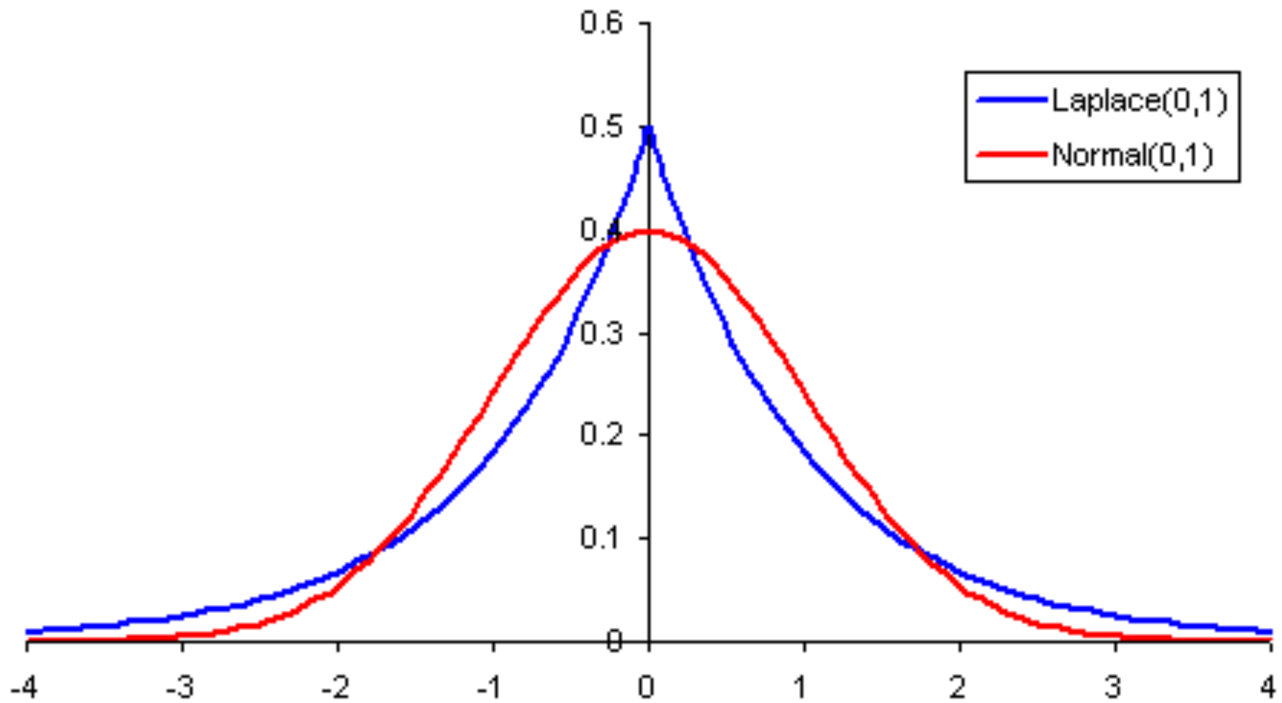
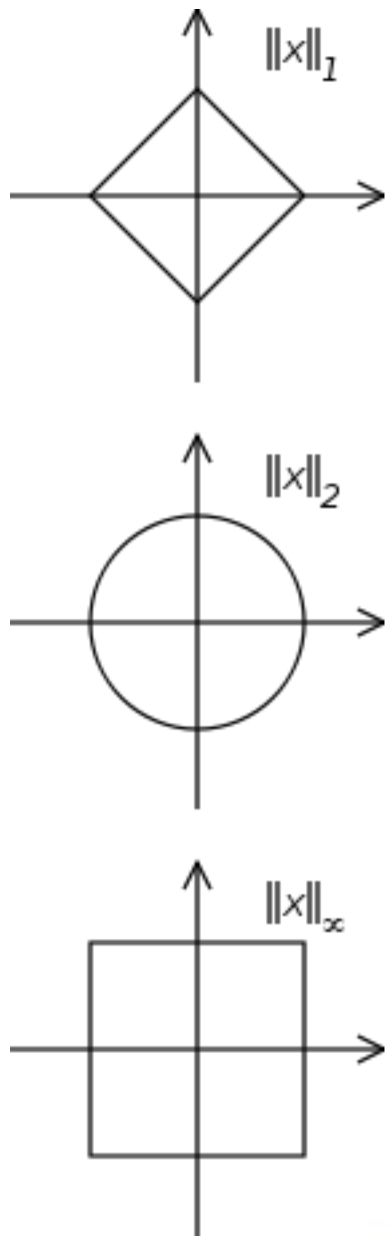


Figure 4.5: A comparison between Gaussian and Laplace priors. The Gaussian prior prefers the values to be near zero, whereas the Laplace prior more strongly prefers the values to equal zero.

Regularization intuition



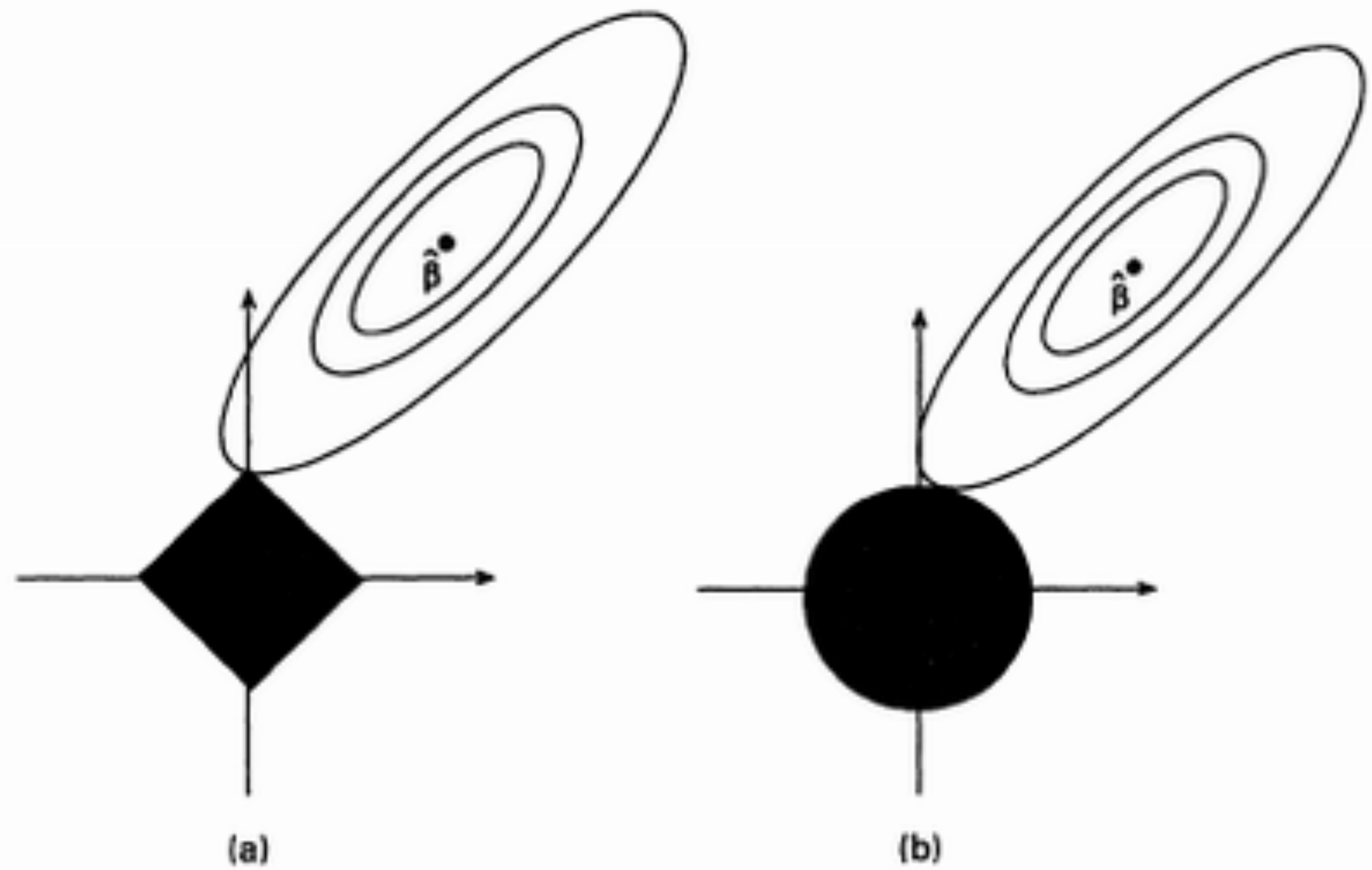
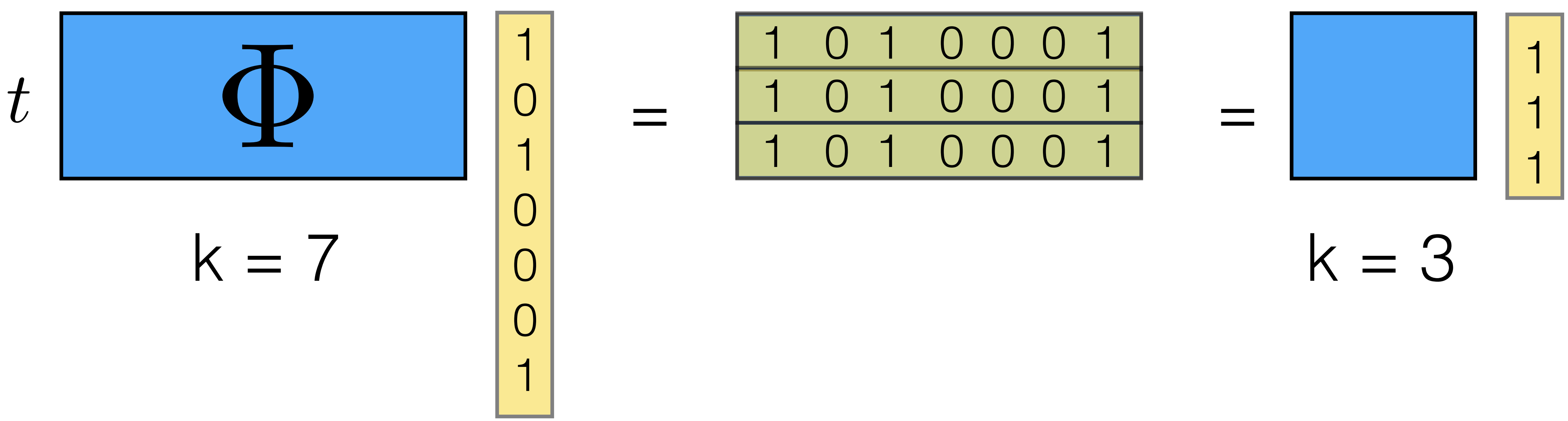


Fig. 2. Estimation picture for (a) the lasso and (b) ridge regression

l1 regularization

- Feature selection, as well as preventing large weight



- How do we solve this optimization?

$$\min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_1$$

How do we solve with l1 regularizer?

$$\min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_1$$

- Is there a closed form solution?
- What approaches can we take?

Practically solving optimization

- In general, what are the advantages and disadvantages of the closed form linear regression solution?

  + Simple approach: no need to add additional requirements, like stopping rules

  - Is not usually possible

  - Must compute an expensive inverse

  - With a large number of features, inverting large matrix

  ? What about a large number of samples?