Linear regression

Reminders

- Assignment should be submitted on eclass
  - due Thursday
- You should try to talk to TAs during the lab session and office hours about assignment questions
- My office hours are more for clarifying concepts
- I have permanently moved my office hours to Thursday, from 2-4
- Updates notes with a few typo fixes

Solution approach and Prediction approach

- You learn a model to make predictions, e.g., p(x l lambda)
- Regardless of how you learn the model parameter lambda, the model is your approximation of the true p(x l lambda)
  - The way you use the model is the same
  - e.g., we talked about using the most likely value as a prediction
- You can use MAP or MLE to learn the parameters
  - The quality of the model will be different, based on the choice

Summary of optimal models

- Expected cost introduced to formalize our objective
- For classification (with uniform cost)

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

- For regression (with squared-error cost)

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

- For both prediction problems, useful to obtain p(y | x) or some statistics on p(y | x) (i.e., E[Y | x])

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 l x] or p(y lx)

Optimal versus Estimated models

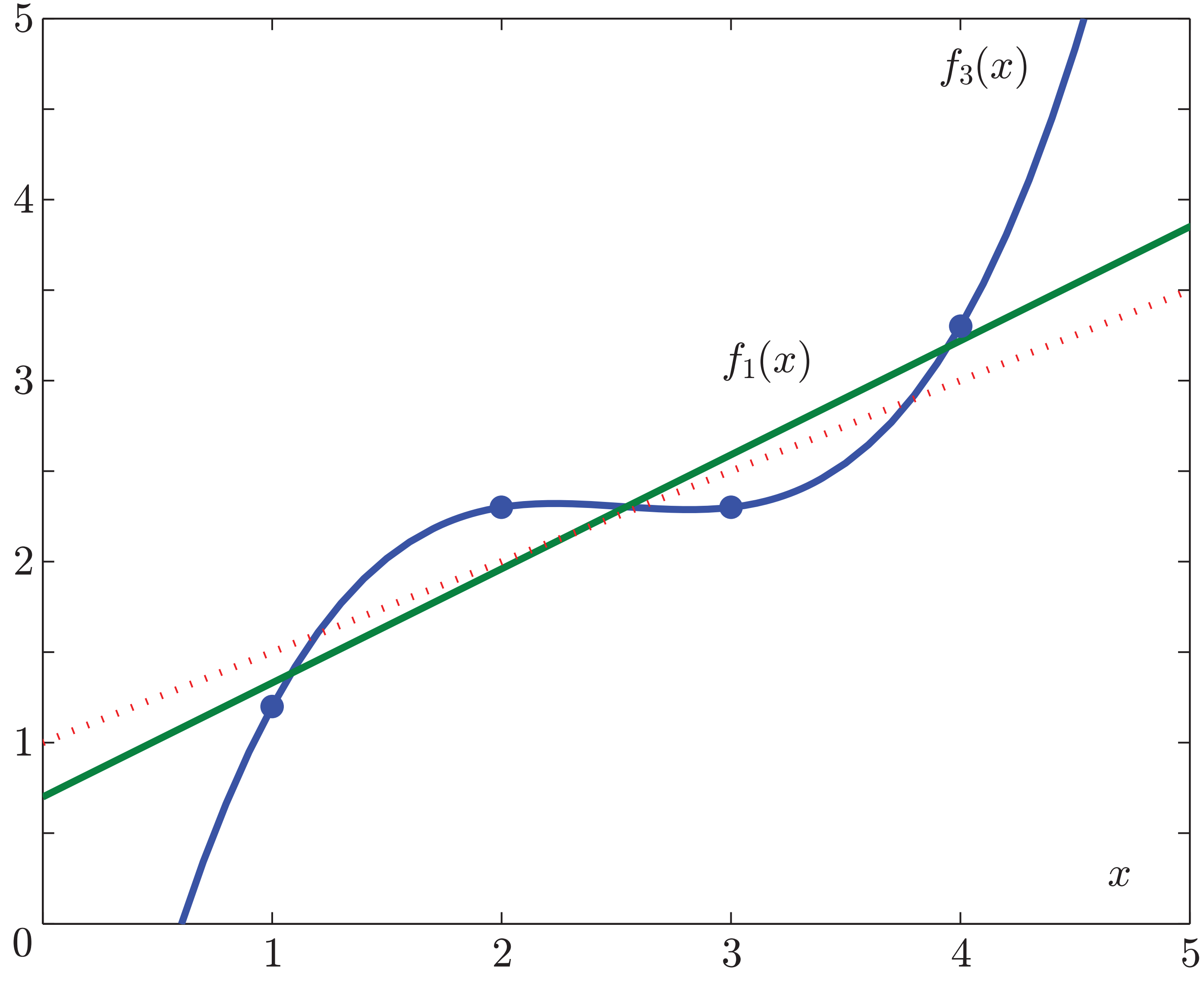
- The discussion about optimal models does not tell us how to obtain f^* (just what it is)
- What prevents us from immediately specifying f^*?
- 1: f^* could be a complicated function of x, whereas we might be restricted to simpler functions (e.g., linear functions)
- 2: Even if f^* is in the function class we consider, we only have a limited amount of data and so will have estimation error
- Both 1 and 2 contribute to **reducible error**, where 1 contributes to the **bias** and 2 contributes to the **variance**

(we will talk more about bias and variance later)

Exercise: Reducible error (bias)

- Can $f(\mathbf{x}) = \sum_{j=1}^{d} w_j x_j$ always represent E[Y I 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
- Simplistic functional form means we cannot predict y

Linear versus polynomial function



Exercise: Reducible error (variance)

Imagine again that $y = w^* x_1 x_2$ for some w^*

Imagine this time that $f(x)$ has the right functional form :

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

$\mathcal{F} = \{f : \mathbb{R}^2 \to \mathbb{R} \mid f(x) = w x_1 x_2 \text{ for } w \in \mathbb{R}\}$

Imagine you estimate w from a batch of n samples

Does $w = w^*$ (i.e., zero reducible error)?

Is w biased?

Let’s start with linear functions

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

Linear Regression

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

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

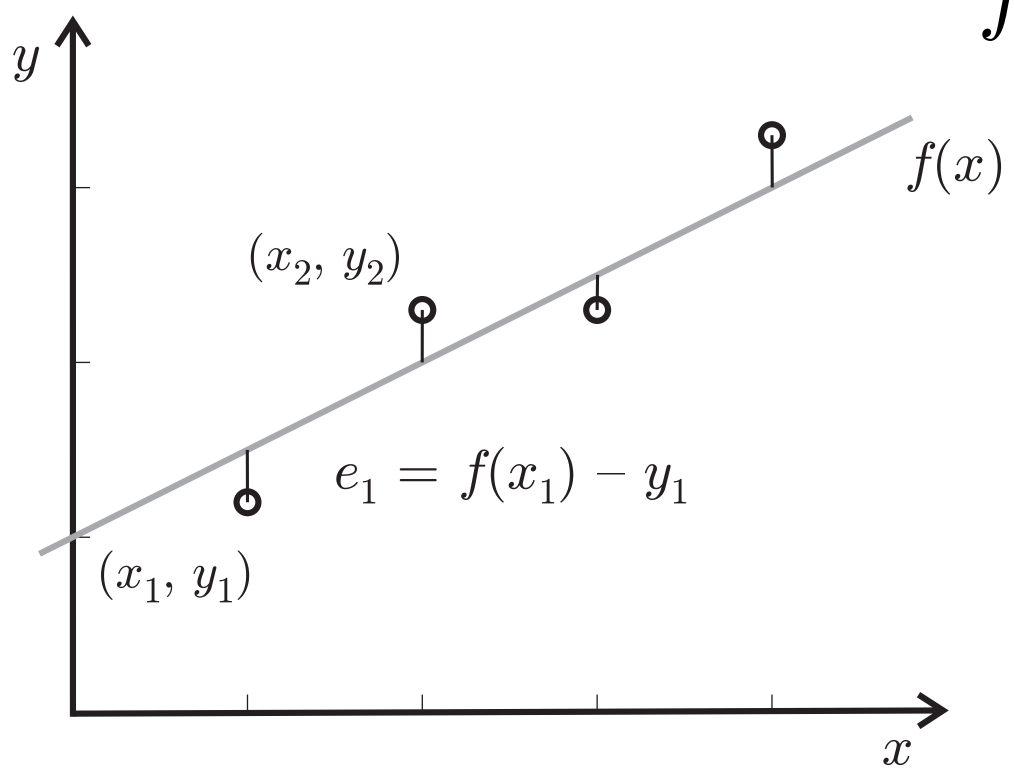
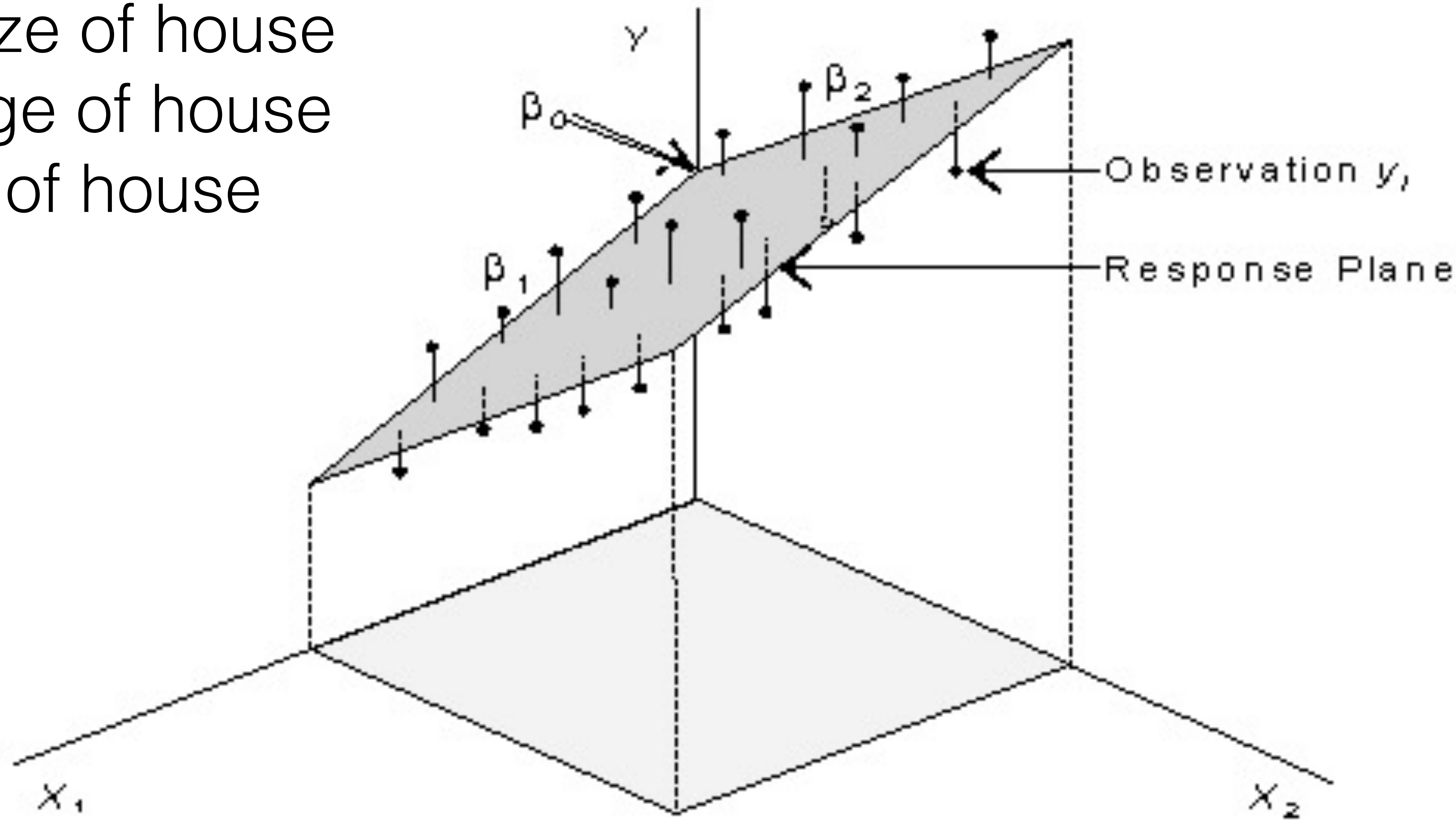


Figure 4.1: An example of a linear regression fitting on data set $\mathcal{D} = \{(1, 1.2), (2, 2.3), (3, 2.3), (4, 3.3)\}$. *The task of the optimization process is to find the best linear function* $f(x) = w_0 + w_1 x$ *so that the sum of squared errors* $e_1^2 + e_2^2 + e_3^2 + e_4^2$ *is minimized.*

(Multiple) Linear Regression

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

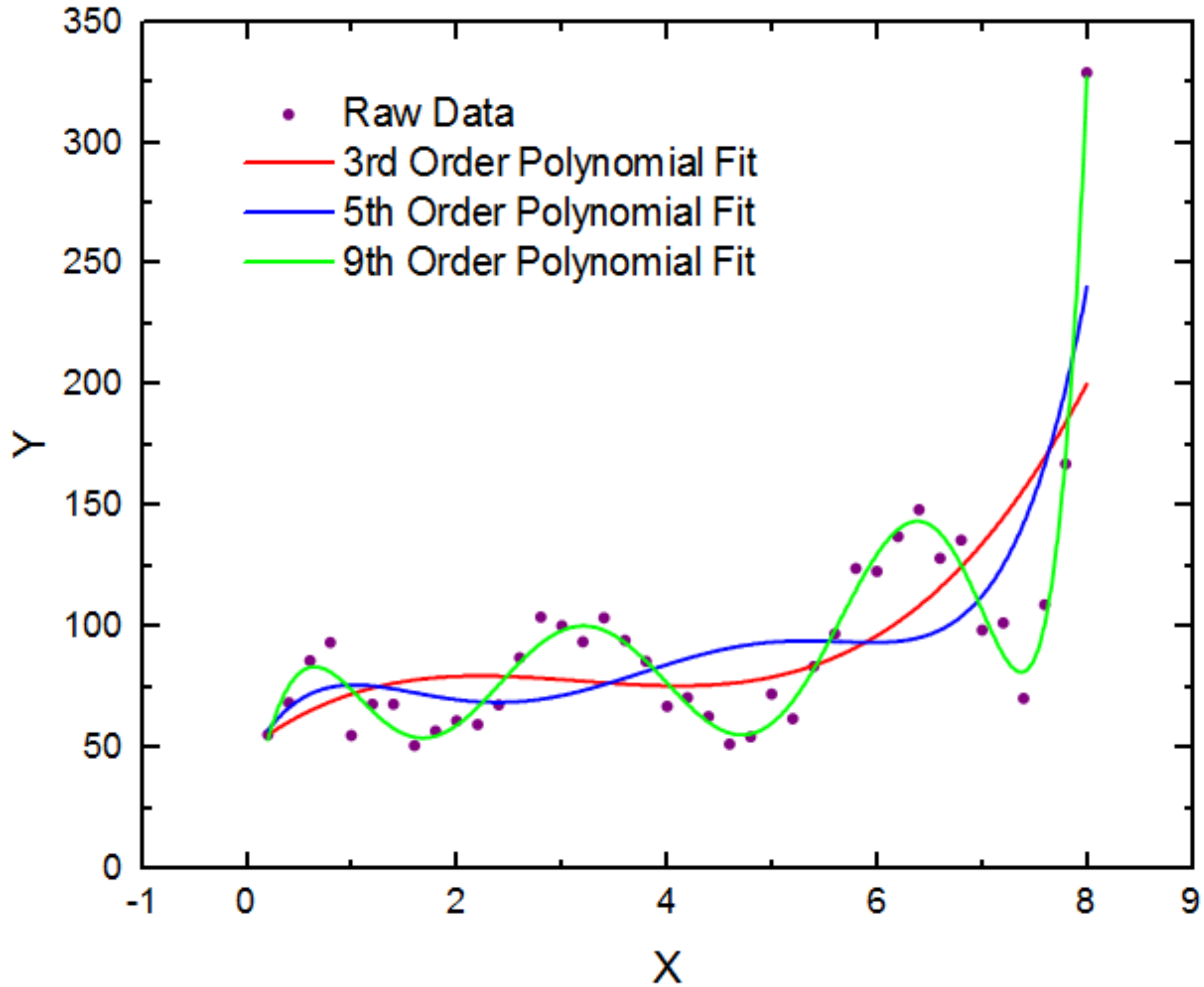


Linear regression importance

- 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

Polynomial representations

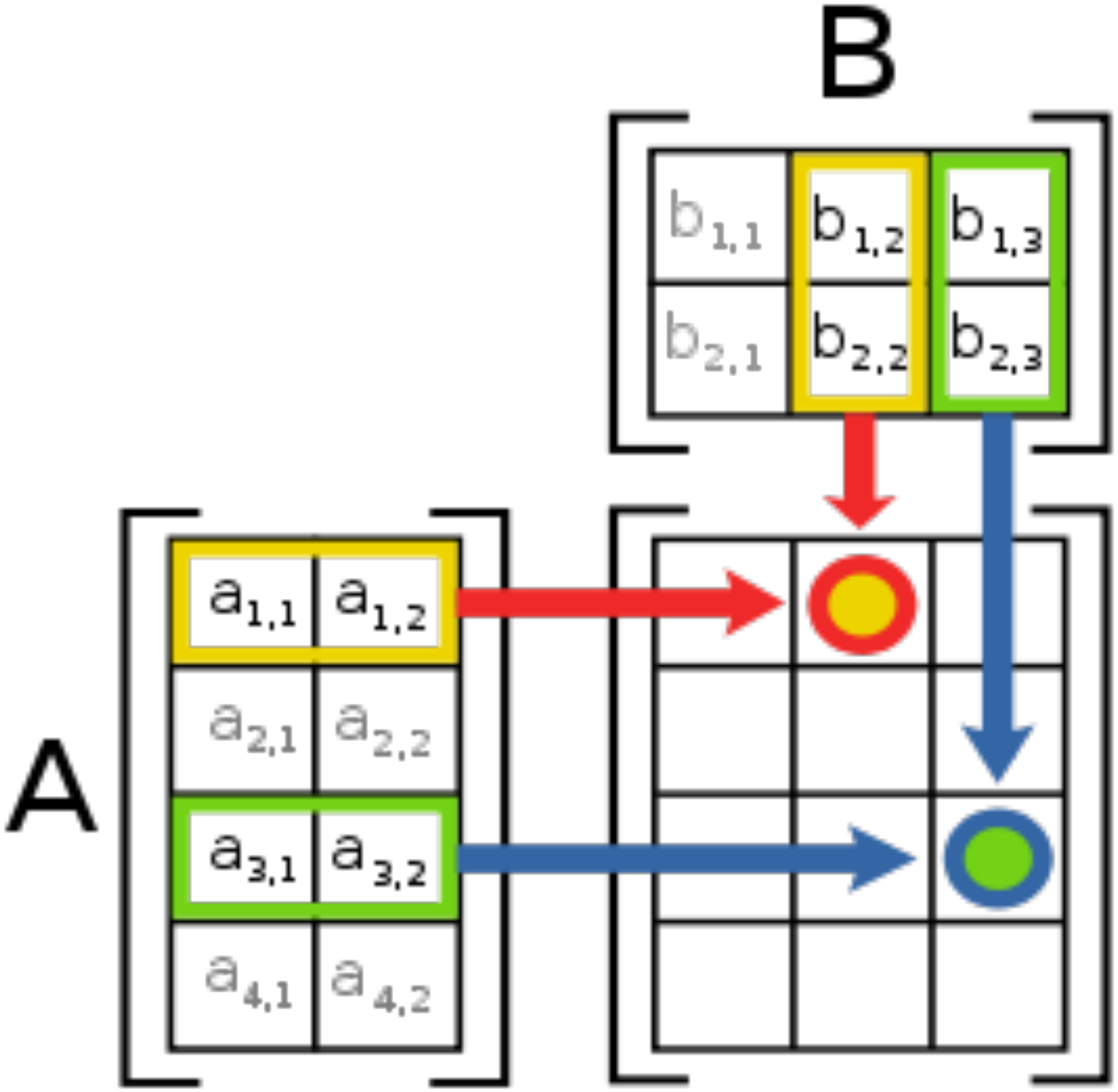
$$w_0 + w_1 x^1 + w_2 x^2 + \ldots + w_9 x^9$$



$$\text{For } \boldsymbol{\phi}(x) = [1, x, x^2, x^3, \ldots, x^9]$$

$$f(\boldsymbol{\phi}(x)) = \boldsymbol{\phi}(x)^\top \mathbf{w}$$

Reminder: Matrix multiplication



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

Reminder: SVD

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

Every matrix is a linear operator that can be decomposed into a rotation (V), scaling (Sigma), and rotation (U) operation

Whiteboard

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

September 26, 2019

- Assignment 1 due today
- Your thought questions will be marked soon
- I sometimes go beyond the notes in lecture, to give you extra info and insights, but I will only expect you to know the topics provided in the notes
  - e.g., I will not require you to know what an SVD is for an exam
  - But understanding sensitivity due to small singular values helps in understanding solution quality, and bias-variance
  - Questions about bias-variance will be on an exam

Clarification: Adding a column of ones

- We have mostly ignored estimating the intercept coefficient w0
- This is because we can always add a feature that is 1 (e.g., x1 = 1 for all instances)
- The weight for this feature gives the intercept term
- We estimate the vector w, assuming some has added a bias unit (aka intercept unit)
  - in the notes we index j from zero, and assume we have d+1 dimensional vector x
- What if we don't estimate the bias unit?

Last time we talked about:

- Formulating regression as a maximum likelihood problem
  - by assuming Y was Gaussian with mean < x, w >
- How to solve that maximum likelihood problem
  - by taking partial derivatives to find the stationary point
  - this resulted in a system of equations, for which we can use system solvers A w = b
- Starting to understand the properties of that solution
  - Sensitivity/conditioning of that linear system
  - Today: Unbiasedness of the solution
  - Today: Variance of the solution and relationship to singular values of X

Why do small singular values of X matter?

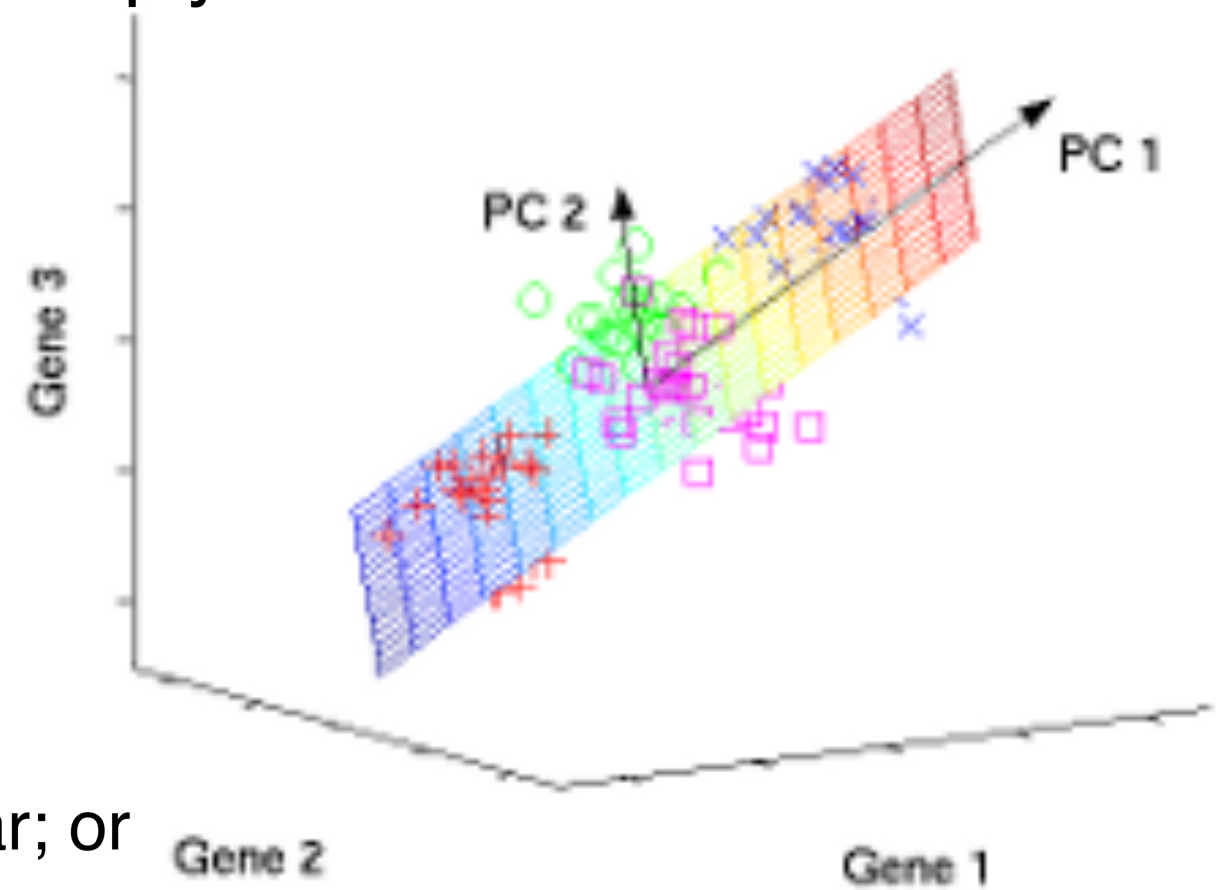
$$\mathrm{Var}(\mathbf{w}(\mathcal{D})_k) = \sigma^2 \mathbb{E}\left[\sum_{j=1}^{d} \frac{\mathbf{v}_{jk}^2}{\sigma_j^2}\right]$$

← We will do this today

- Indicates components in the weight vector can vary more across different dataset
- Small changes in v are magnified by division by tiny singular values
- By would singular values be small across datasets? What does this all really mean?

When might X have very small singular values?

- Singular values being small imply that the data lies in a lower-dimensional space



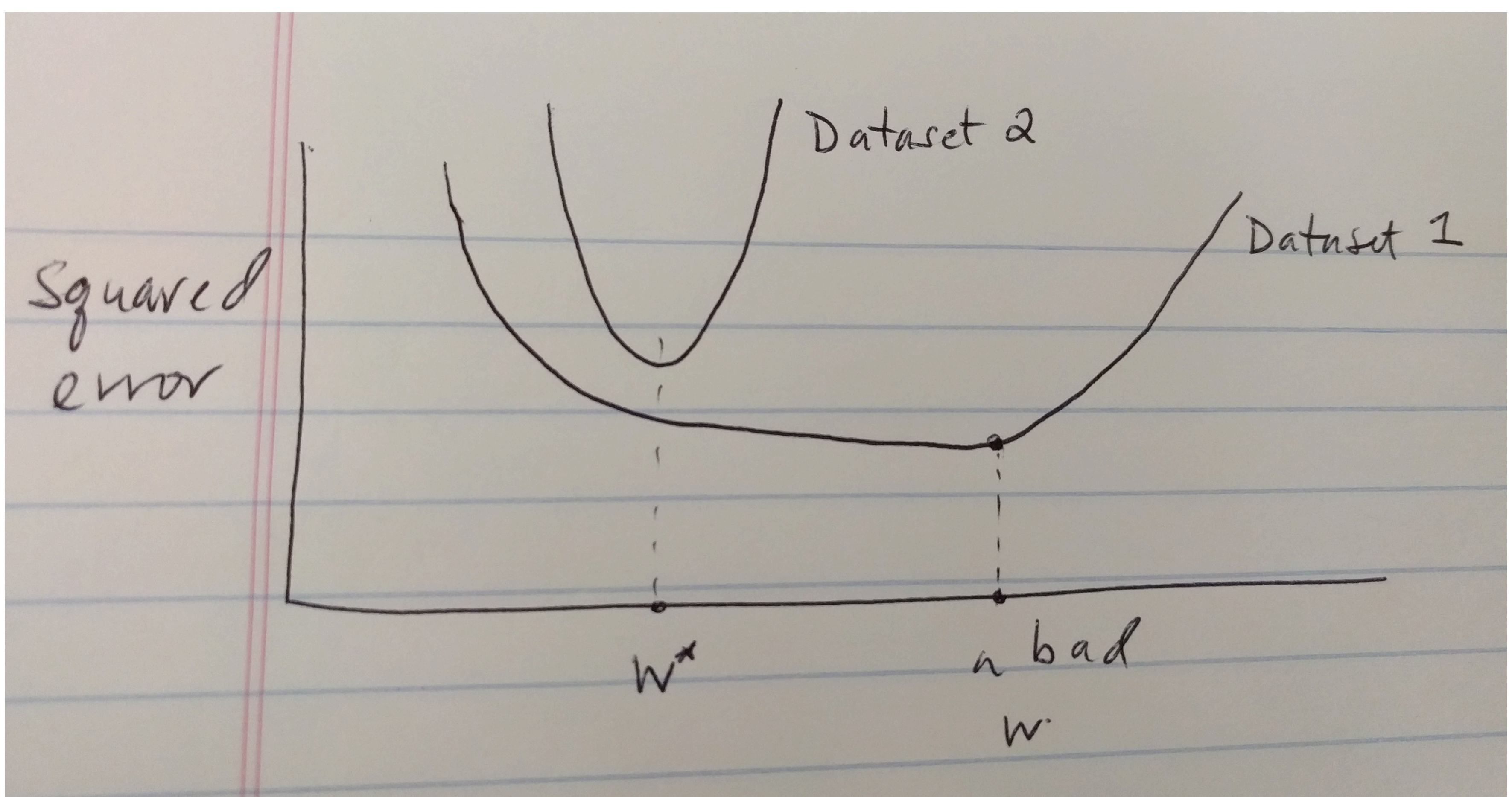
- This might happen if
  - variables are nearly co-linear; or
  - there is not enough data, so it looks like the data lies in a lower-dimensional space —> if you got more samples, it would start filling out the space more

Another interpretation

- If X has small singular values, then $A = \langle X, X \rangle$ in the linear system has small eigenvalues
  - eigenvalues for A are squared singular values of X
- Consider if X has one zero singular value, then A has one zero eigenvalue —> This means that there are infinitely many solutions to the linear system
  - $A w = b$ has infinitely many feasible w. Which one is closest to w^*?
- Similarly, for very small singular values, many solutions that are almost equally good. Which one is best?

Another interpretation (cont…)

- The flexibility in picking w (because there is not enough data constraining the system) allows the least-squares solution to fit to the noise
- If more data had been observed, the system would not have that w as a reasonable solution



But would the data ever really lie in a low-dimensional space?

- It is a bit less likely for low-dimensional input observations to lie in a lower-dimensional
- But a common strategy is to generate an expansion, projecting the input data up into a higher-dimensional space
- It becomes more likely that it lies in a lower-dimensional within that higher-dimensional space

Linear regression for non-linear problems

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

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

$\mathbf{X}$ $\rightarrow$ $\mathbf{\Phi}$

	$\mathbf{X}$
1	x_1
	x_2
	...
n	x_n

$\rightarrow$

$\phi_0(x_1)$	...	$\phi_p(x_1)$
...	...	...
...	...	...
$\phi_0(x_n)$	...	$\phi_p(x_n)$

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, $j = 0, 1, \ldots, p$.

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

Overfitting

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

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

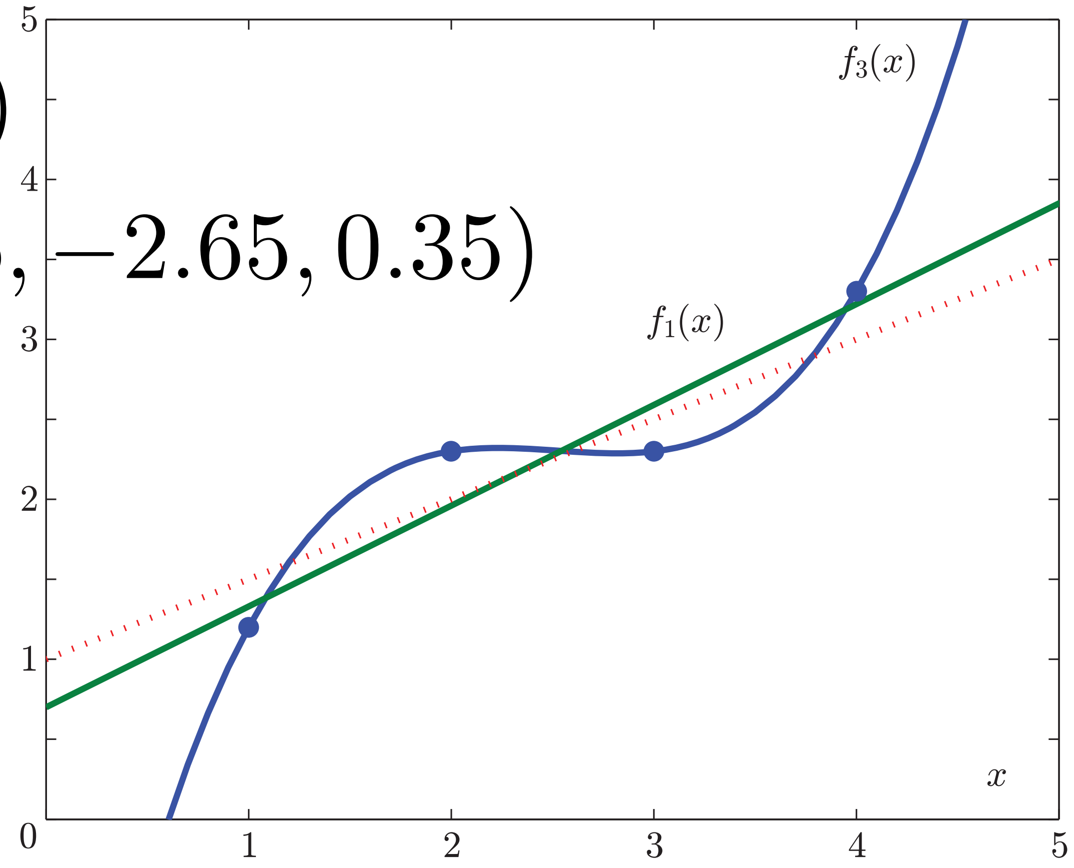


Figure 4.4: Example of a linear vs. polynomial fit on a data set shown in Figure 4.1. The linear fit, $f_1(x)$, is shown as a solid green line, whereas the cubic polynomial fit, $f_3(x)$, is shown as a solid blue line. The dotted red line indicates the target linear concept.

In the higher-dimensional space with (1, x, x^2, x^3), a linear plane can perfectly fit the four points, but not for (1, x)

Whiteboard

- Couple of clarifications on notation
  - singular values are non-negative
  - dimensions of variables
- Adding a prior that prefers simpler w (l2 regularizer)
- Bias and variance of linear regression solution with an l2 regularizer
  - and exercise where we truncate the singular values

October 1, 2019

- Thought Questions due next Thursday (October 10)
- Assignment 2 is due October 24
- Projects can be done in pairs or threes
- We will release a document soon on how grad students can get bonus marks, by
  - volunteering to review projects
  - doing a more complete project, as the chapter of a thesis or as a complete paper that could be submitted to a workshop/conference
- Any questions?

Why regularize?

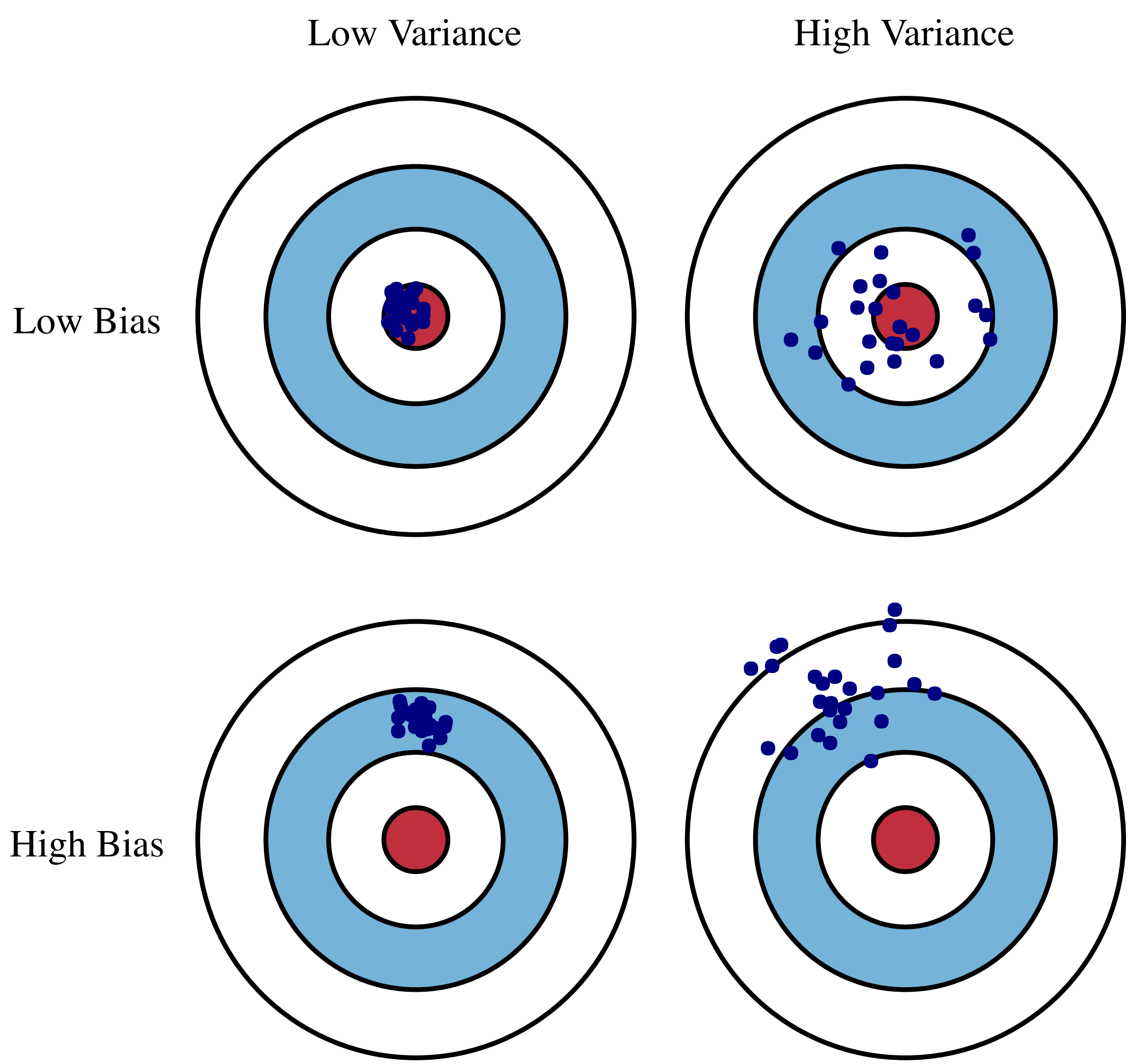
$$\|\mathbf{Xw} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

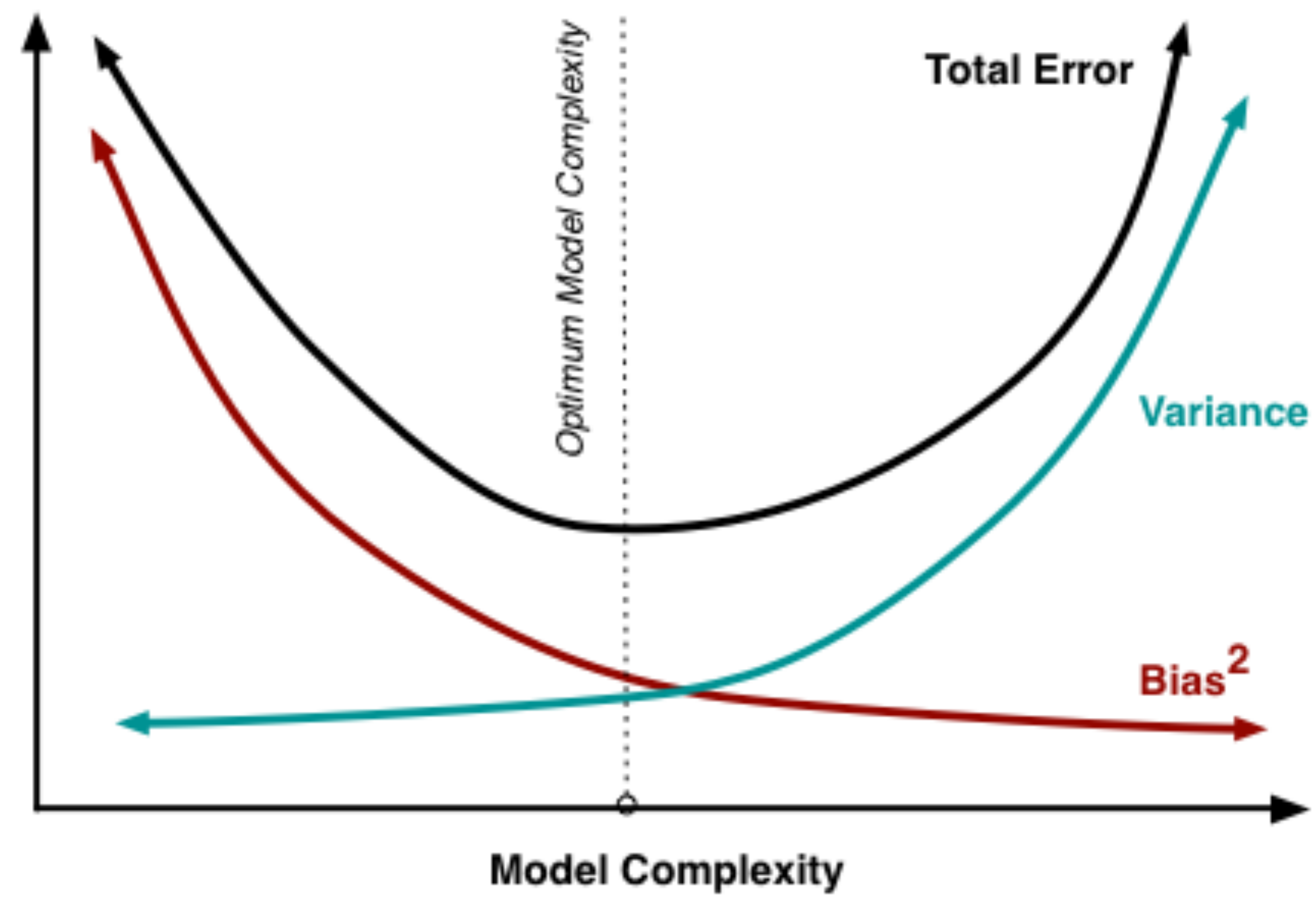
- Why would we a priori believe our weights should be close to zero? What if one of our coefficients needs to be big?
- What happens if one magnitude of the features is really big and another is small?
  - e.g., x1 = house price (100000), x2 = number of rooms (3)
- What is the disadvantage to regularizing? What does it do to the weights?
- How can we fix this problem?

Whiteboard

- Bias-variance trade-off

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? $\mathbf{w} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$

What if don’t want the regularization to disappear?

$$\mathbf{w} = \left(\frac{1}{n}\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\left(\frac{1}{n}\mathbf{X}^\top\mathbf{y}\right)$$

- Implicitly, the regularization weight is lambda x t
- It is more common to pick a fixed regularization (as above)
- Why?
  - Still often picking over-parameterized models, compared to the amount of available data
  - Can improve trainability, which is desired even if there is lots of data (e.g., l2 regularizer is strongly convex)

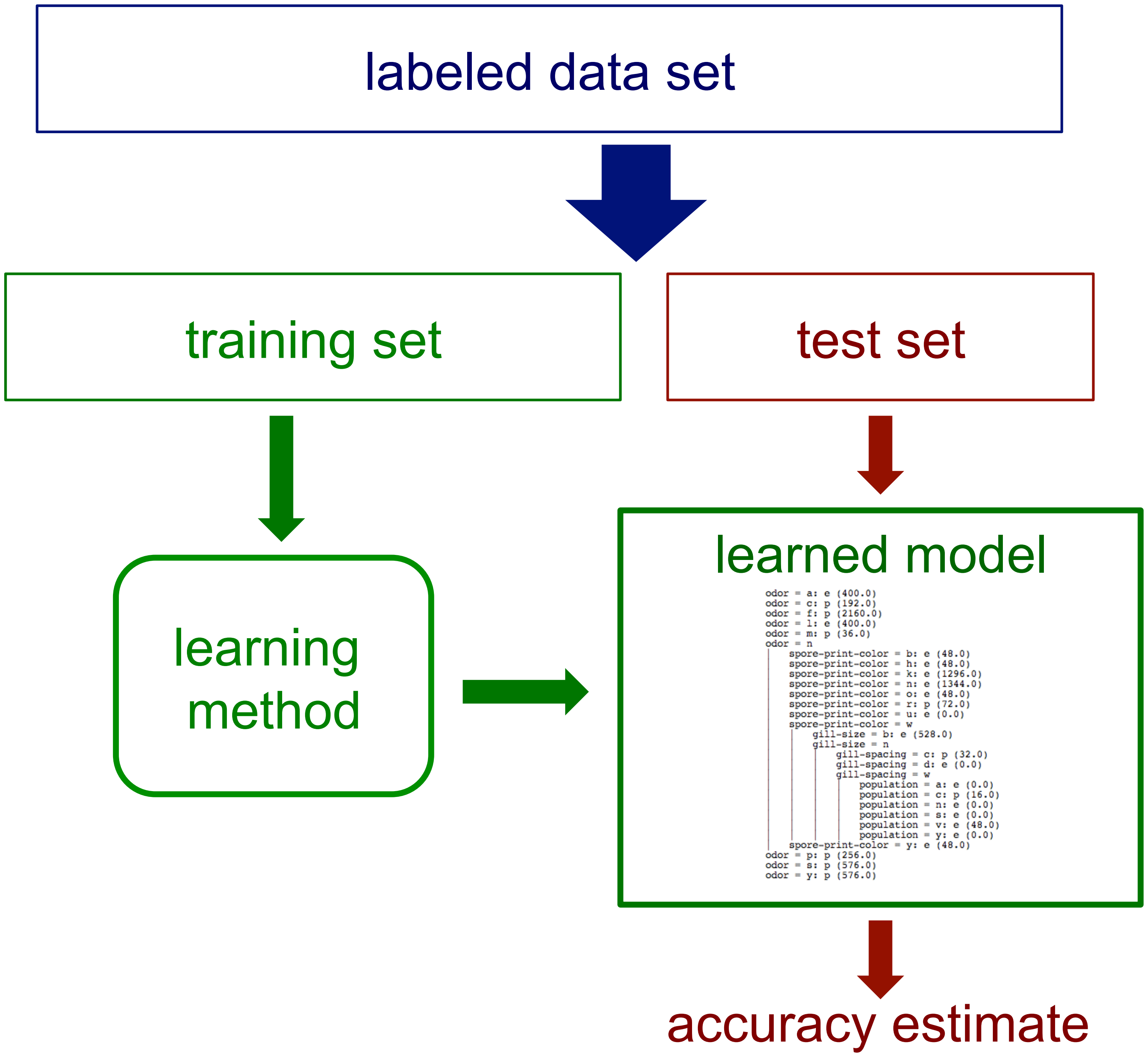
But how do we pick lambda?

- 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

- Imagine you are comparing two models and get two test accuracies with this approach (split into training and test)
- Imagine model 1 has lower error than model 2. Can you be confident that model 1 has better generalization error?
- What if we split the data 90% to 10%?
- What if we have a small test set?
- What if we test 100 different lambda values?
- Another strategy is cross-validation, with multiple training-test splits

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

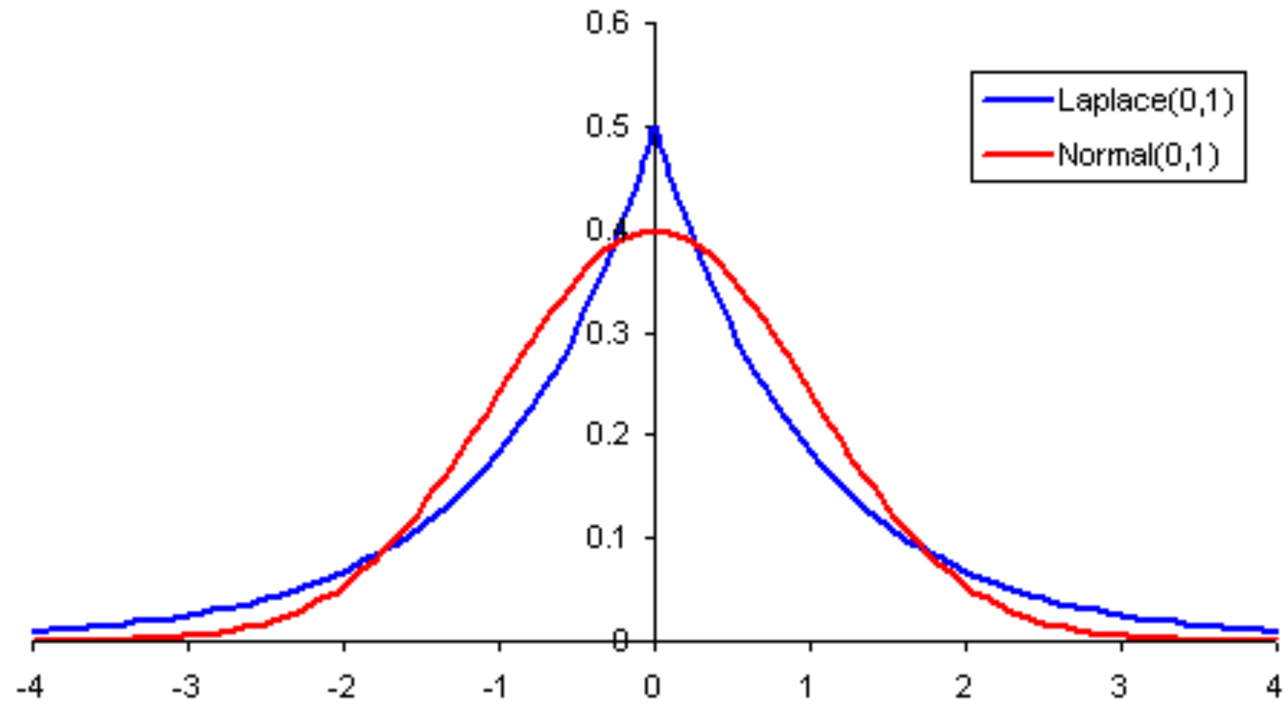
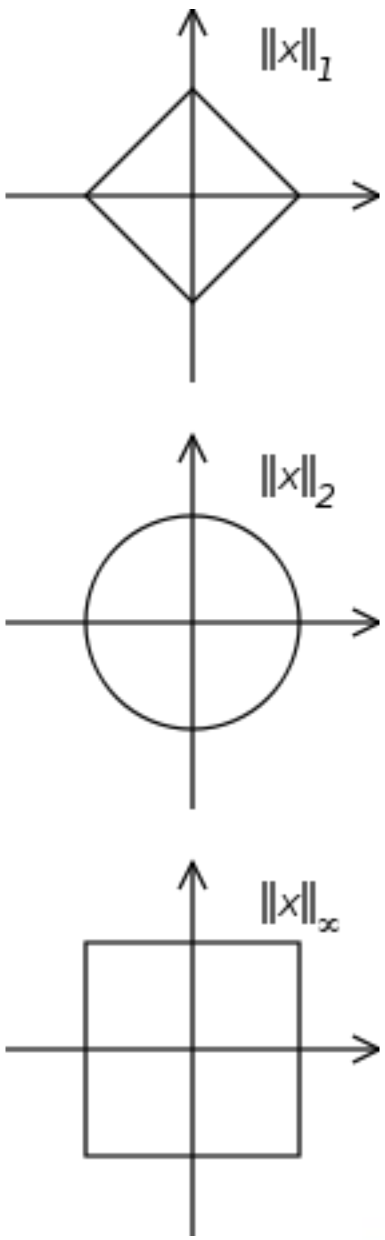


Figure 4.5: A comparison between Gaussian and Laplace priors. The Gaussian prior prefers the values to be near zero, whereas the Laplace prior more strongly prefers the values to equal zero.

Regularization intuition



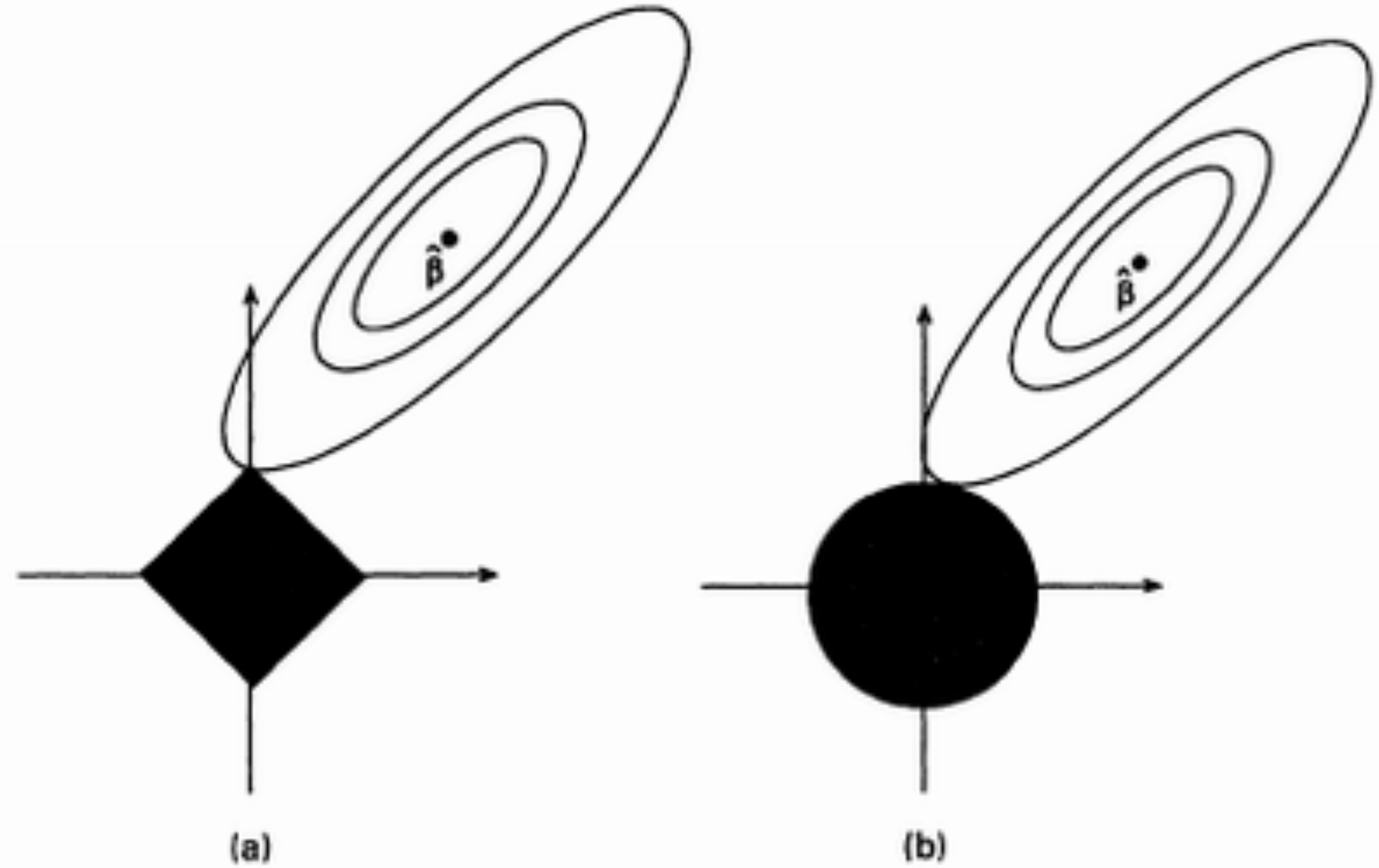
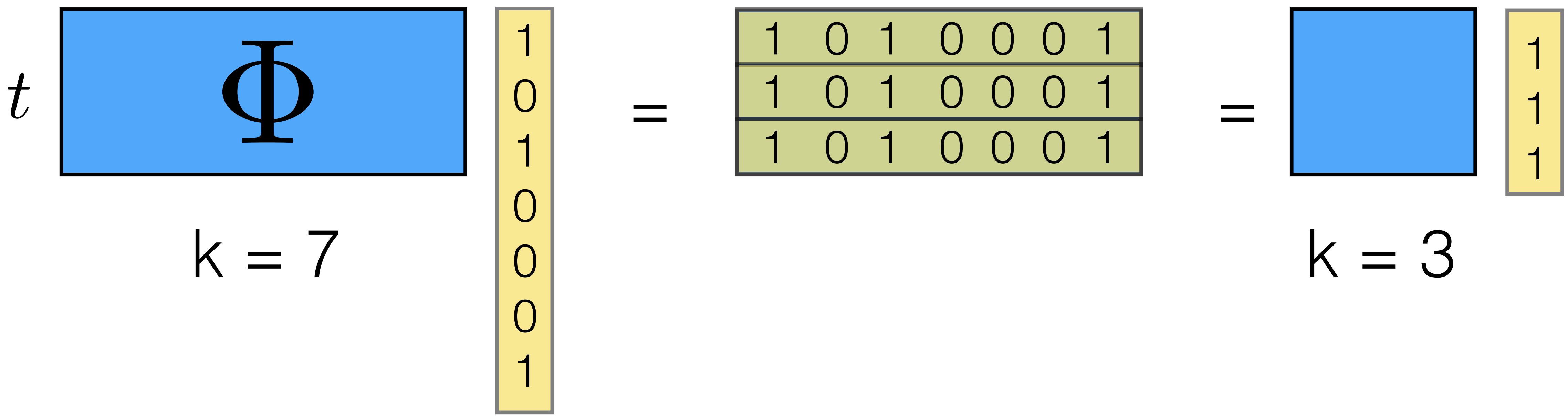


Fig. 2. Estimation picture for (a) the lasso and (b) ridge regression

$p = \infty$ $p = 2$ $p = 1$ $0 < p < 1$ $p = 0$

l1 regularization

- Feature selection, as well as preventing large weights



-

Exercise: l1 and l2 regularization

- Imagine there are exactly two features x1 = x2
  - i.e., only one feature for prediction, with an added redundant feature
- Want to learn best linear function w0 + w1 x1 + w2 x2
  - i.e., w0 + (w1 + w2) x2
- What would least-squares plus l2 regularization provide?
- What would least-squares plus l1 regularization provide?
- What if a bit of noise is added to x2?

Why would we do feature selection?

- Why not use all the features? It is more information?
- What settings might you care to do feature selection?
- Are there any settings where using l1 for feature selection might be problematic?
- What is an alternative approach for feature selection?

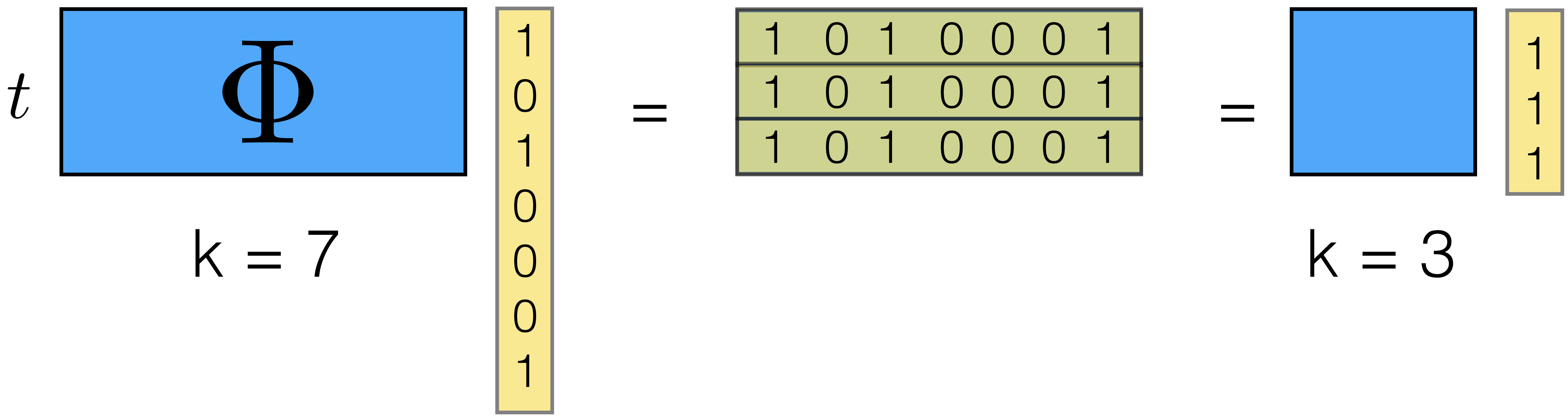
Matching pursuit: Greedy approach where add one feature at a time

Feature selection versus dimensionality reduction

- Another option is to do dimensionality reduction
    - e.g., project features x into a lower-dimensional space P x
- Exercise: what are the pros and cons?
- We'll talk about this more later

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 optimizations

- 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?