COMPUTER SCIENCE
INDIANA UNIVERSITY
School of Informatics and Computing
Bloomington

Practical linear regression

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

Reminders

- Thought questions 2 due on Wednesday
  - We won’t cover generalized linear models in class, but they set up logistic regression well
- Any general questions?

Some review

- Discussed linear regression
  - goal: obtain weights w such that < x, w> approximates E[Y | x]
- Discussed maximum likelihood formulation
- Discussed gradient descent algorithm for linear regression
  - avoid computing inverse of (X^T X)
- Discussed some linear algebra concepts

ML for regression

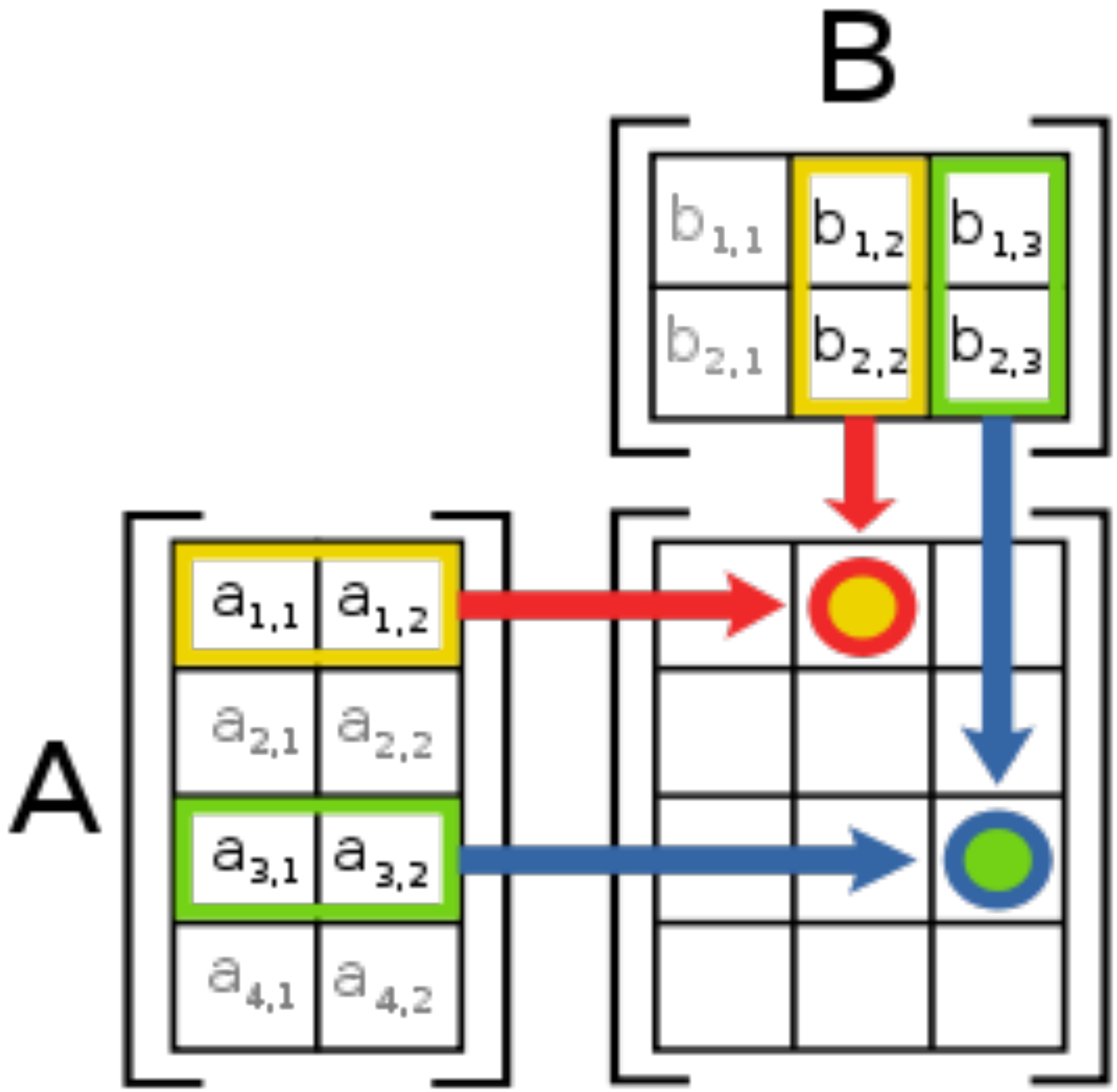
$$Y = \sum_{j=0}^{d} \omega_j X_j + \varepsilon,$$

conditional density is $p(y|\mathbf{x}, \boldsymbol{\omega}) = \mathcal{N}(\boldsymbol{\omega}^\top \mathbf{x}, \sigma^2)$.

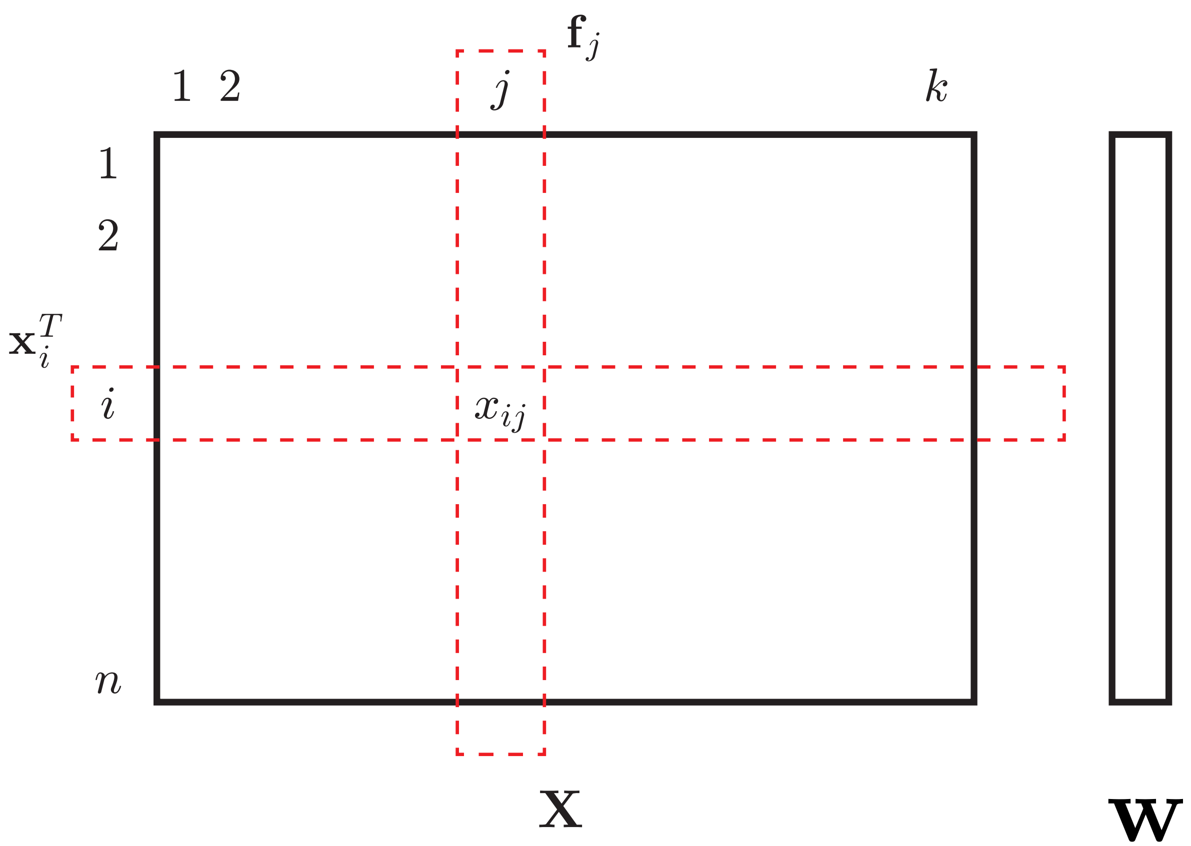
$$\begin{aligned} p(D|\mathbf{w}) &= p(\mathbf{X}, \mathbf{y}|\mathbf{w}) \\ &= p(\mathbf{y}|\mathbf{X}, \mathbf{w}) p(\mathbf{X}) \end{aligned}$$

$p(\mathbf{X})$ could be anything

Matrix multiplication



Xw

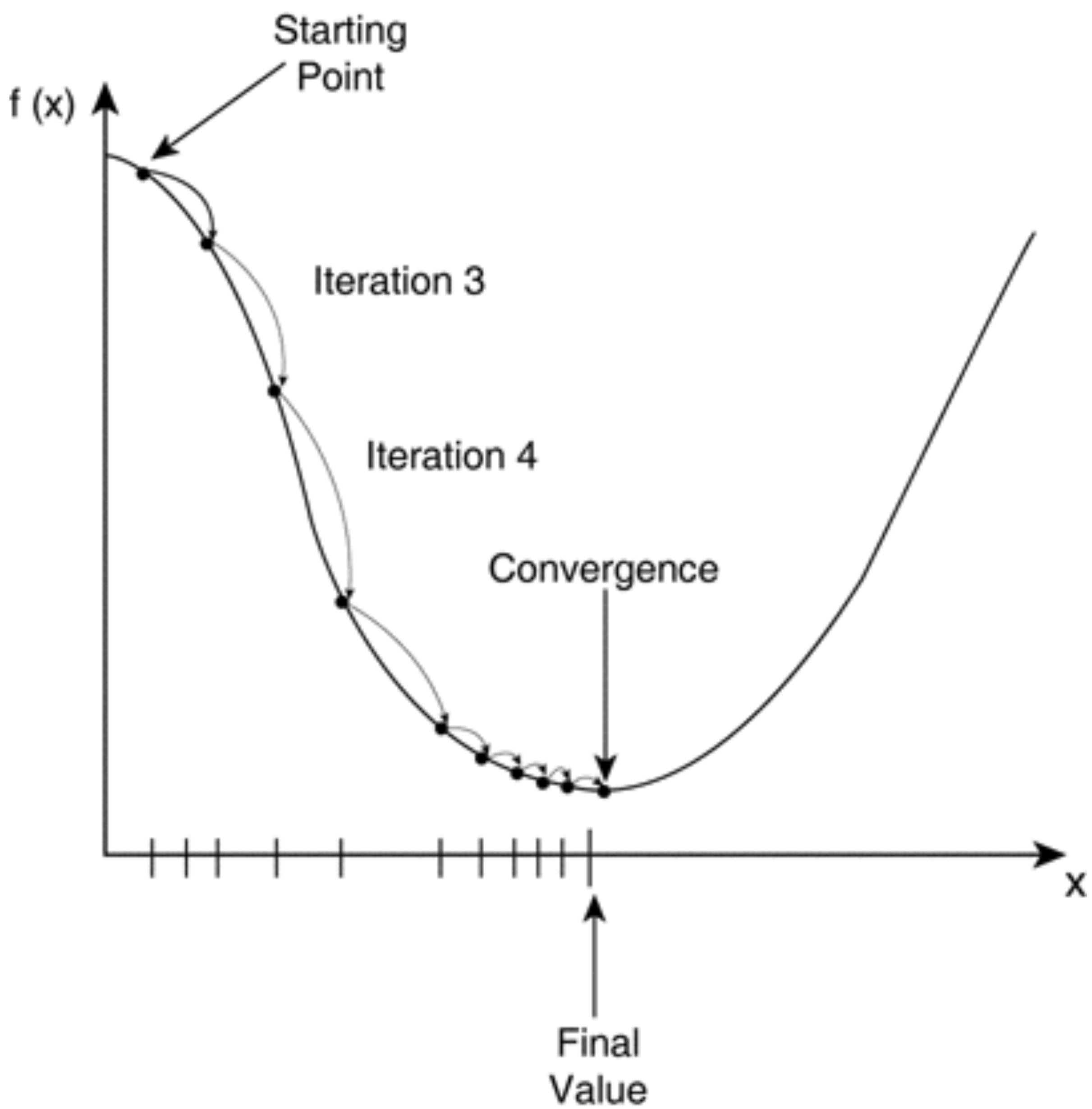


More intuition on solution

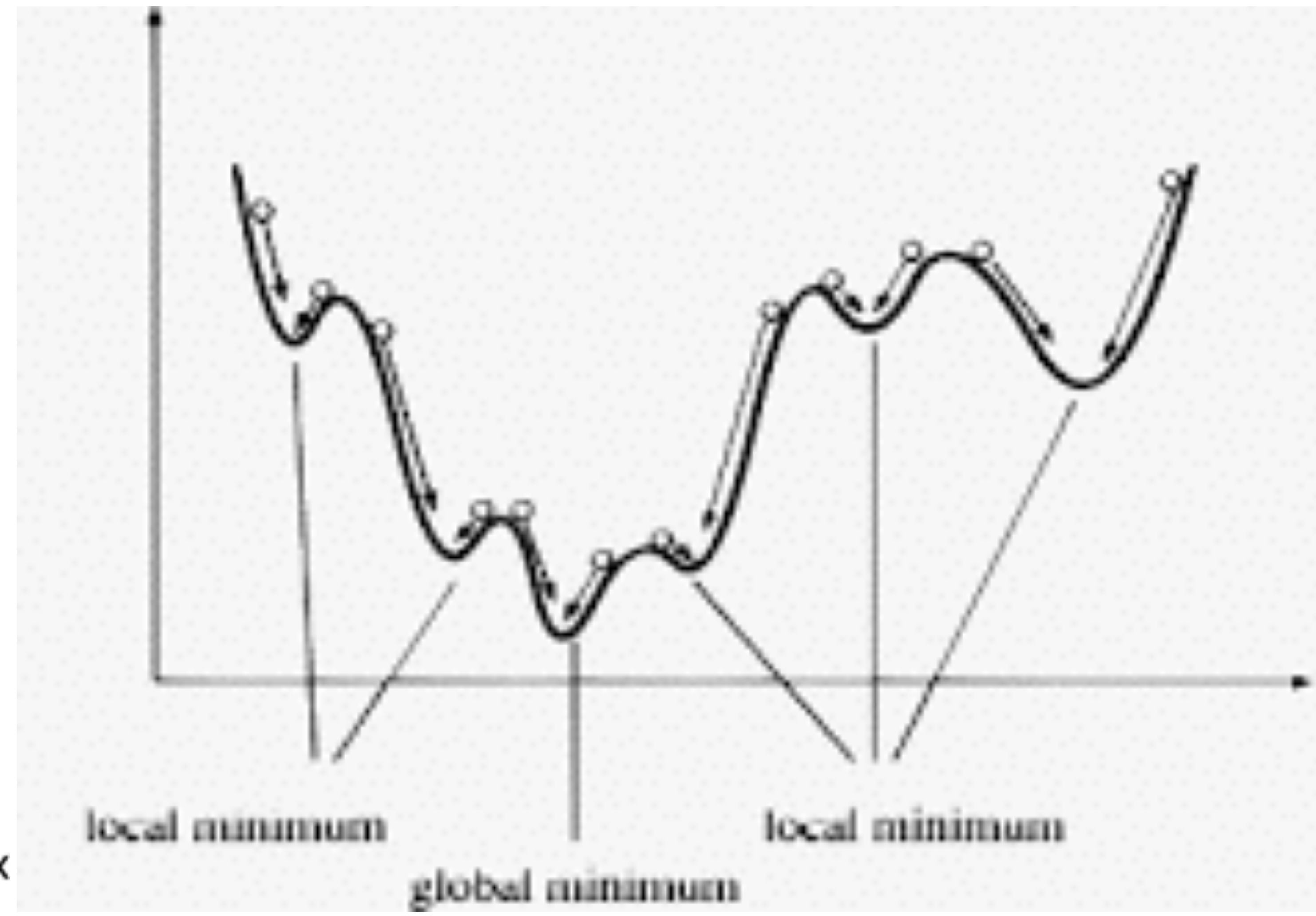
$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \qquad \hat{\mathbf{y}} = \mathbf{X}\mathbf{w}^*$$

- Gradient being zero gives a stationary point
  - only in a few cases can we solve the equation gradient E(w) = 0
  - e.g., we will not be able to do so for logistic regression
  - for other cases we will step in the direction of the gradient until we reach such a stationary point

Gradient descent intuition



Convex function



Non-convex function

$$w_{t+1} = w_t - \alpha_t \nabla f(w_t)$$

Exercise question

- What does it mean for there to be a closed-form solution?
- How can we tell if there is a closed form solution?
- Why do we use gradient descent?
- How do we
    - pick step-sizes for gradient descent?
    - pick the initial point w? Does this matter?

Practical additions

- How do we generically include an intercept (bias unit)?
- What if some samples are more important than others?
  - e.g., rare cases, or expensive cases
- What if we have more than one output?
- What are the properties of the solution?
- How do we learn more complex functions?
- How can we say we learned well on a small number of features, i.e., prevent overfitting?

Example: OLS

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

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

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?

Linear regression for non-linear problems

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

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

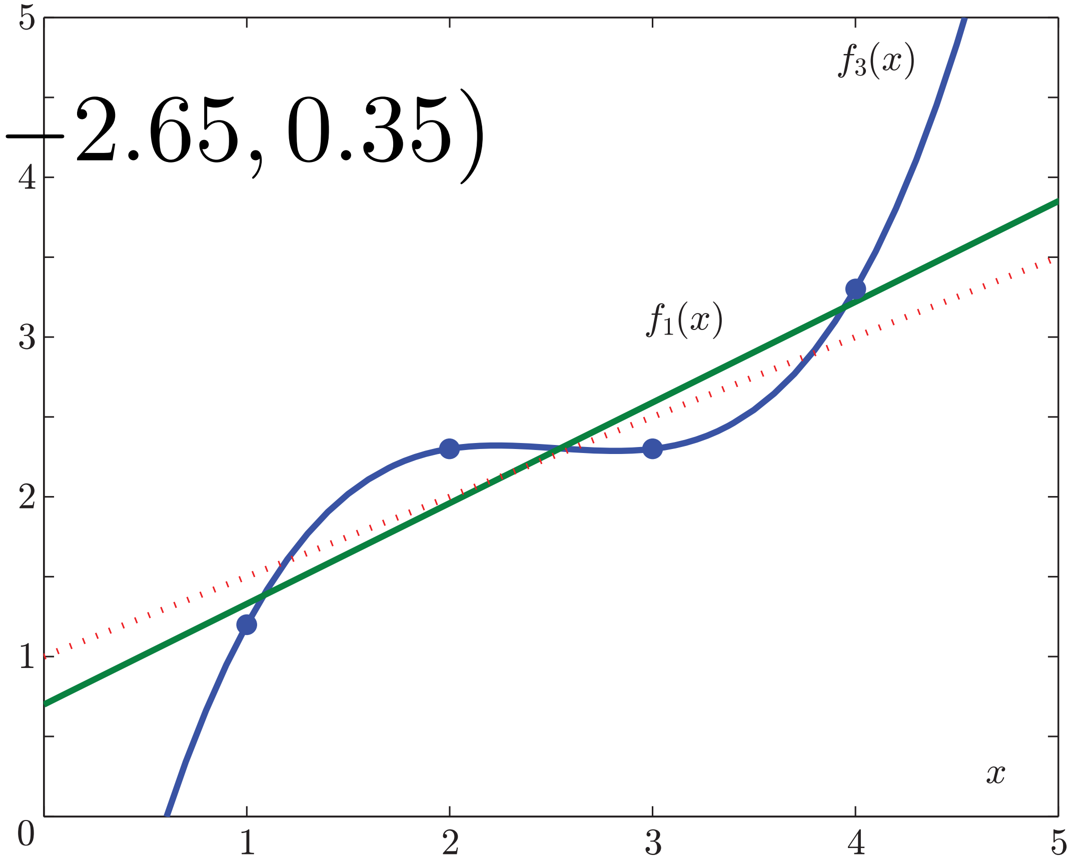
$$\begin{array}{c|c|}
 & \mathbf{X} \\
\cline{2-2}
1 & x_1 \\
\cline{2-2}
 & x_2 \\
\cline{2-2}
 & \dots \\
\cline{2-2}
n & x_n \\
\cline{2-2}
\end{array}
\quad \rightarrow \quad
\begin{array}{|c|c|c|}
\multicolumn{3}{c}{\mathbf{\Phi}} \\
\hline
\phi_0(x_1) & \dots & \phi_p(x_1) \\
\hline
\dots & \dots & \dots \\
\hline
\dots & \dots & \dots \\
\hline
\phi_0(x_n) & \dots & \phi_p(x_n) \\
\hline
\end{array}$$

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

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

Overfitting

$$\mathbf{w}_1^* = (0.7, 0.63)$$
$$\mathbf{w}_3^* = (-3.1, 6.6, -2.65, 0.35)$$



$$\sum_{i=1}^{4}(f_1(x_i) - y_i)^2 > \sum_{i=1}^{4}(f_3(x_i) - y_i)^2$$

$$f_1(x) = w_{1,0}^* + w_{1,1}^* x$$
$$f_3(x) = w_{3,0}^* + w_{3,1}^* x + w_{3,2}^* x^2 + w_{3,3}^* x^3$$

Regularization intuition

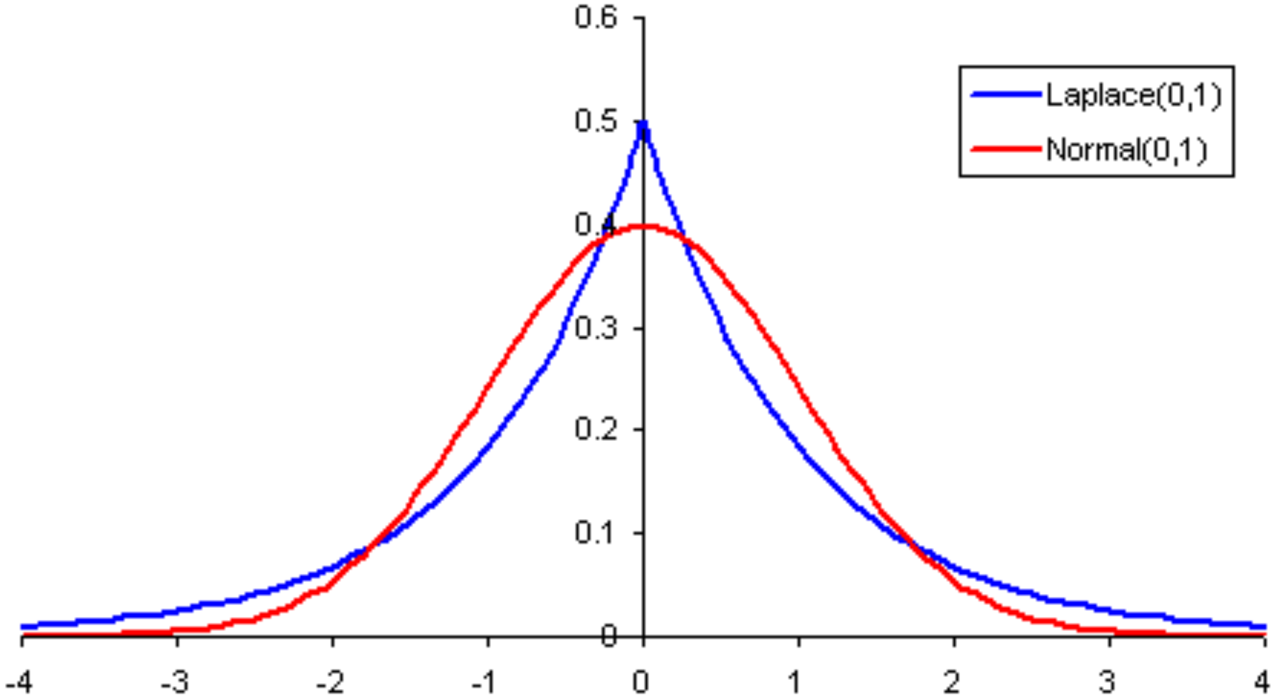


Figure 4.5: A comparison between Gaussian and Laplace priors. The Gaussian prior prefers the values to be near zero, whereas the Laplace prior more strongly prefers the values to equal zero.

Whiteboard

- Stability analysis with SVD
  - including bias-variance of solution
- Regularization
- Stochastic optimization for big datasets