ECE595 / STAT598: Machine Learning I

Lecture 13.1: Connecting Bayesian with Linear Regression - Linear Regression Review

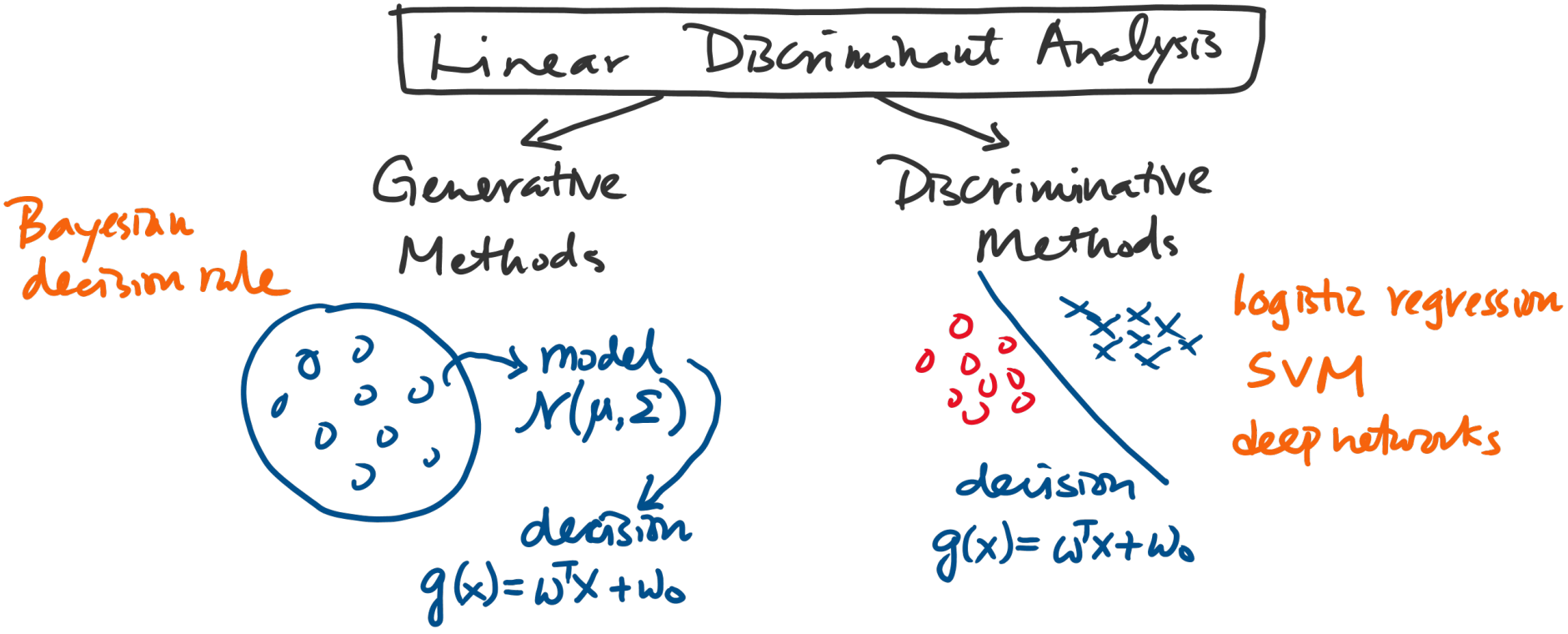
Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

PURDUE
UNIVERSITY

Overview



- In linear discriminant analysis (LDA), there are generally two types of approaches
- **Generative approach**: Estimate model, then define the classifier
- **Discriminative approach**: Directly define the classifier

Outline

Generative Approaches

- Lecture 9 Bayesian Decision Rules
- Lecture 10 Evaluating Performance
- Lecture 11 Parameter Estimation
- Lecture 12 Bayesian Prior
- Lecture 13 Connecting Bayesian and Linear Regression

Today's Lecture

- Linear Regression Review
  - Linear regression in the context of classification
  - Linking linear regression with MLE and MAP
- Connection between Linear Regression and Bayesian
  - Expected Loss
  - Main Result
  - Implications

Linear Regression Reviewed

- Linear regression is actually a **discriminative method**.
- Do not require a distributional model.
- Construct the hypothesis function directly:

$$h(\boldsymbol{x}) = \begin{cases} +1, & \text{if} \quad g(\boldsymbol{x}) > 0, \\ -1, & \text{if} \quad g(\boldsymbol{x}) < 0. \end{cases}$$

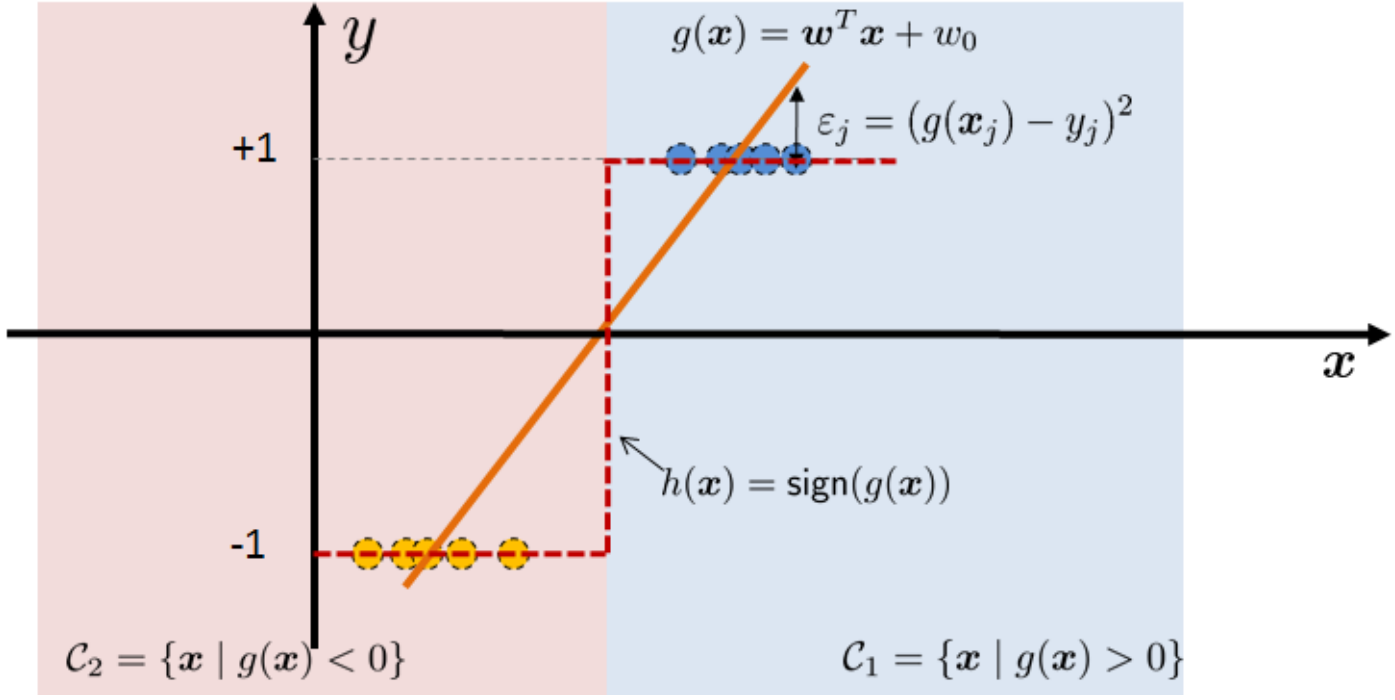
- Consider a binary classification problem with discriminant function:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0$$

- The goal is to determine the parameters $\boldsymbol{\theta} = \{\boldsymbol{w}, w_0\}$
- Training data: $(\boldsymbol{x}_n, y_n)_{n=1}^N$
  - $\boldsymbol{x}_n \in \mathbb{R}^d$ is the input vector
  - $y_n \in \{-1, +1\}$ is the corresponding label

Geometry of Linear Regression

- The discriminant function $g(\boldsymbol{x})$ is linear
- The hypothesis function $h(\boldsymbol{x}) = \text{sign}(g(\boldsymbol{x}))$ is a unit step



Loss Function

- All discriminant algorithms have a **Training Loss Function**

$$J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(g(\boldsymbol{x}_n), y_n).$$

- In linear regression,

$$\begin{aligned} J(\boldsymbol{\theta}) &= \frac{1}{N}\sum_{n=1}^{N}(g(\boldsymbol{x}_n) - y_n)^2 \\ &= \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{w}^T\boldsymbol{x}_n + w_0 - y_n)^2 \\ &= \frac{1}{N}\left\|\begin{bmatrix}\boldsymbol{x}_1^T & 1\\ \vdots & \vdots \\ \boldsymbol{x}_N^T & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{w}\\ w_0\end{bmatrix} - \begin{bmatrix}y_1\\ \vdots \\ y_N\end{bmatrix}\right\|^2 = \frac{1}{N}\|\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{y}\|^2. \end{aligned}$$

Solution of Linear Regression

Theorem (Linear Regression Solution)

The loss function of a linear regression model is given by

$$J(\boldsymbol{\theta}) = \|\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{y}\|^2,$$

of which the minimizer is

$$\boldsymbol{\theta}^* = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y}.$$

- Take derivative and setting to zero:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \left\{ \|\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{y}\|^2 \right\} \\ &= 2\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{y}) = \boldsymbol{0}.\end{aligned}$$

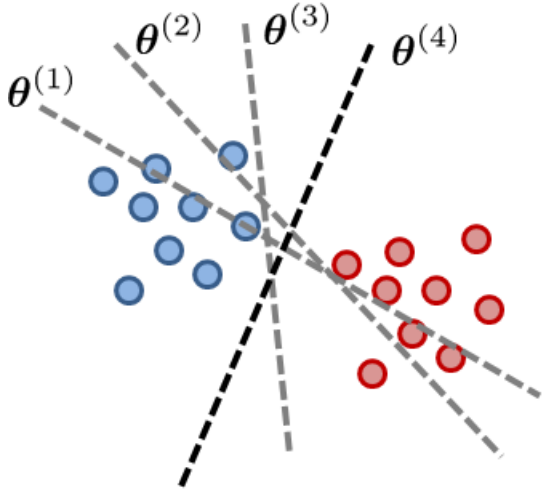
- So solution is $\boldsymbol{\theta}^* = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y}$, assuming $\boldsymbol{A}^T\boldsymbol{A}$ is invertible.

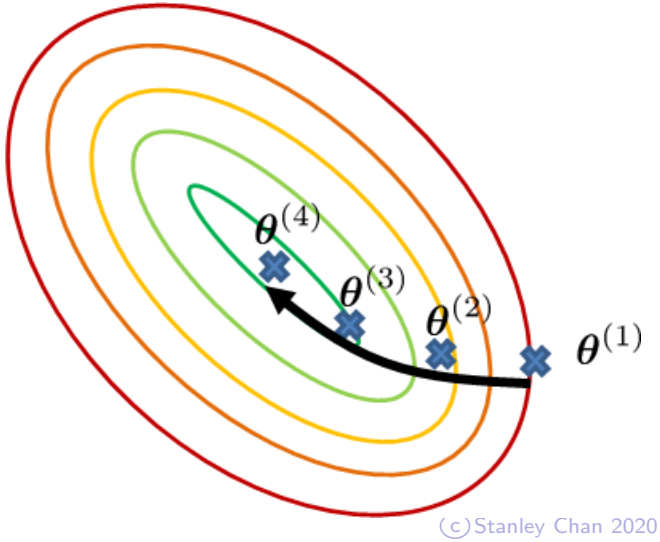
When $\boldsymbol{A}^T\boldsymbol{A}$ is large

- Computing $(\boldsymbol{A}^T\boldsymbol{A})^{-1}$ directly is infeasible for large-scale datasets with a large number of variables
- Consider using iterative algorithms such as gradient descent
- The gradient descent is given by the iteration:

$$\begin{aligned}\boldsymbol{\theta}^{(k+1)} &= \boldsymbol{\theta}^{(k)} - \eta\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{(k)}) \\ &= \boldsymbol{\theta}^{(k)} - \eta(2\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{\theta}^{(k)} - 2\boldsymbol{A}^T\boldsymbol{y})\end{aligned}$$

- A pictorial illustration of the gradient descent step:





Treating Linear Regression as Maximum-Likelihood

- Minimizing $J(\boldsymbol{\theta})$ is the same as solving a **maximum-likelihood**:

$$\begin{aligned}
\boldsymbol{\theta}^* &= \underset{\boldsymbol{\theta}}{\text{argmin}} \quad \|\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{y}\|^2 \\
&= \underset{\boldsymbol{\theta}}{\text{argmin}} \quad \sum_{n=1}^{N} (\boldsymbol{a}_n^T \boldsymbol{\theta} - y_n)^2 \\
&= \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \exp\left\{ -\sum_{n=1}^{N} (\boldsymbol{a}_n^T \boldsymbol{\theta} - y_n)^2 \right\} \\
&= \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(\boldsymbol{a}_n^T \boldsymbol{\theta} - y_n)^2}{2\sigma^2} \right\} \right\}
\end{aligned}$$

- Assume noise is i.i.d. Gaussian with variance σ^2.

Treating Linear Regression as Maximum-a-Posteriori

- We can modify the MLE by adding a prior

$$p_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) = \exp\left\{ -\frac{\rho(\boldsymbol{\theta})}{\beta} \right\}.$$

- Then, we have a MAP problem:

$$\begin{aligned}
\boldsymbol{\theta}^* &= \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(\boldsymbol{a}_n^T\boldsymbol{\theta} - y_n)^2}{2\sigma^2} \right\} \right\} \exp\left\{ -\frac{\rho(\boldsymbol{\theta})}{\beta} \right\} \\
&= \underset{\boldsymbol{\theta}}{\text{argmin}} \quad \frac{1}{2\sigma^2} \sum_{n=1}^{N} (\boldsymbol{a}_n^T\boldsymbol{\theta} - y_n)^2 + \frac{1}{\beta}\rho(\boldsymbol{\theta}) \\
&= \underset{\boldsymbol{\theta}}{\text{argmin}} \quad \|\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{y}\|^2 + \lambda\rho(\boldsymbol{\theta}), \qquad \text{where} \quad \lambda = 2\sigma^2/\beta.
\end{aligned}$$

- $\rho(\cdot)$ is called **regularization function**.
- Useful when $\boldsymbol{A}^T\boldsymbol{A}$ is not invertible.

Ridge Regression

- One option: Choose a Gaussian prior

$$\exp\left\{-\frac{\rho(\boldsymbol{\theta})}{\beta}\right\} = \exp\left\{-\frac{\|\boldsymbol{\theta}\|^2}{2\sigma_0^2}\right\}$$

- Then, the MAP becomes

$$\begin{aligned}
\boldsymbol{\theta}^* &= \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{n=1}^{N} \left\{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(\boldsymbol{a}_n^T\boldsymbol{\theta} - y_n)^2}{2\sigma^2}\right\}\right\} \exp\left\{-\frac{\|\boldsymbol{\theta}\|^2}{2\sigma_0^2}\right\} \\
&= \underset{\boldsymbol{\theta}}{\text{argmin}} \sum_{n=1}^{N} (\boldsymbol{a}_n^T\boldsymbol{\theta} - y_n)^2 + \underbrace{\frac{\sigma^2}{\sigma_0^2}}_{=\lambda} \|\boldsymbol{\theta}\|^2 \\
&= \underset{\boldsymbol{\theta}}{\text{argmin}} \|\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{y}\|^2 + \lambda\|\boldsymbol{\theta}\|^2
\end{aligned}$$

- This is called **Tikhonov regularization** or **Ridge regression**.