Lecture 1 – Part 2
Some Basics

(January 9, 2015)

Mu Zhu
University of Waterloo

Regression

- training data $\{(\boldsymbol{x}_i, y_i); i = 1, 2, ..., n\}$
- $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$
- want $f(\cdot)$ so that we can predict y from $\boldsymbol{x}$ with $f(\boldsymbol{x})$
- mean squared error,

$$\mathrm{MSE}(f) = \mathbb{E}[y - f(\boldsymbol{x})]^2,$$

  a common criterion for how good f is

$\mathbb{E}(y|\boldsymbol{x})$

Let $g(\boldsymbol{x}) = \mathbb{E}(y|\boldsymbol{x})$, and $h(\boldsymbol{x})$ be any other function of $\boldsymbol{x}$. Then,

$$
\begin{aligned}
& \mathbb{E}[y - h(\boldsymbol{x})]^2 \\
= \ & \mathbb{E}[y - g(\boldsymbol{x}) + g(\boldsymbol{x}) - h(\boldsymbol{x})]^2 \\
= \ & \mathbb{E}[y - g(\boldsymbol{x})]^2 + \underbrace{\mathbb{E}[g(\boldsymbol{x}) - h(\boldsymbol{x})]^2}_{\geq 0} + 2\,\underbrace{\mathbb{E}[(y - g(\boldsymbol{x}))(g(\boldsymbol{x}) - h(\boldsymbol{x}))]}_{=0} \\
\geq \ & \mathbb{E}[y - g(\boldsymbol{x})]^2.
\end{aligned}
$$

$$\Downarrow$$

main task for regression: "go after" the function, $\mathbb{E}(y|\boldsymbol{x})$

Exercise Show that $\mathbb{E}[(y - g(\boldsymbol{x}))(g(\boldsymbol{x}) - h(\boldsymbol{x}))] = 0$. (Hint: Use $\mathbb{E}(\cdot) = \mathbb{E}[\mathbb{E}(\cdot|\boldsymbol{x})]$.)

Ex I: Linear Regression

- for $x \in \mathbb{R}$, can start by modelling $\mathbb{E}(y|x)$ as

$$f(x) = \alpha + \beta x$$

- further justification: $\mathbb{E}(y|x)$ linear in x if (x, y) have a joint normal distribution

- just have to estimate α and β from training data

- for $\boldsymbol{x} \in \mathbb{R}^d$, simply

$$f(\boldsymbol{x}) = \alpha + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}$$

Ex II: Nearest-Neighbor Regression

- may feel uncomfortable with assuming $\mathbb{E}(y|\boldsymbol{x})$ to be linear
- can choose to estimate $\mathbb{E}(y|\boldsymbol{x})$ by

$$\widehat{\mathbb{E}}(y|\boldsymbol{x}) \quad = \quad \text{average}\,\{y_i : \boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{x})\}\,,$$

  where $\mathcal{N}(\boldsymbol{x})$ denotes a "neighborhood" around $\boldsymbol{x}$

- meaning of $\mathbb{E}(y|\boldsymbol{x})$: the average of y given a particular $\boldsymbol{x}$
- almost literal interpretation of $\mathbb{E}(y|\boldsymbol{x})$
- relaxes "given a particular $\boldsymbol{x}$" to "within a neighborhood of $\boldsymbol{x}$"

Ex II: Nearest-Neighbor Regression

- for $x \in \mathbb{R}$, suppose

$$\mathcal{N}(x) = \left\{ x_i : \frac{|x_i - x|}{h} < 1 \right\}$$

- can express estimate as

$$\widehat{\mathbb{E}}(y|x) = \frac{\sum_{i=1}^{n} I\left(\frac{|x_i - x|}{h} < 1\right) y_i}{\sum_{i=1}^{n} I\left(\frac{|x_i - x|}{h} < 1\right)}$$

- need to specify h a priori ... called a tuning parameter

Ex III: Kernel Regression

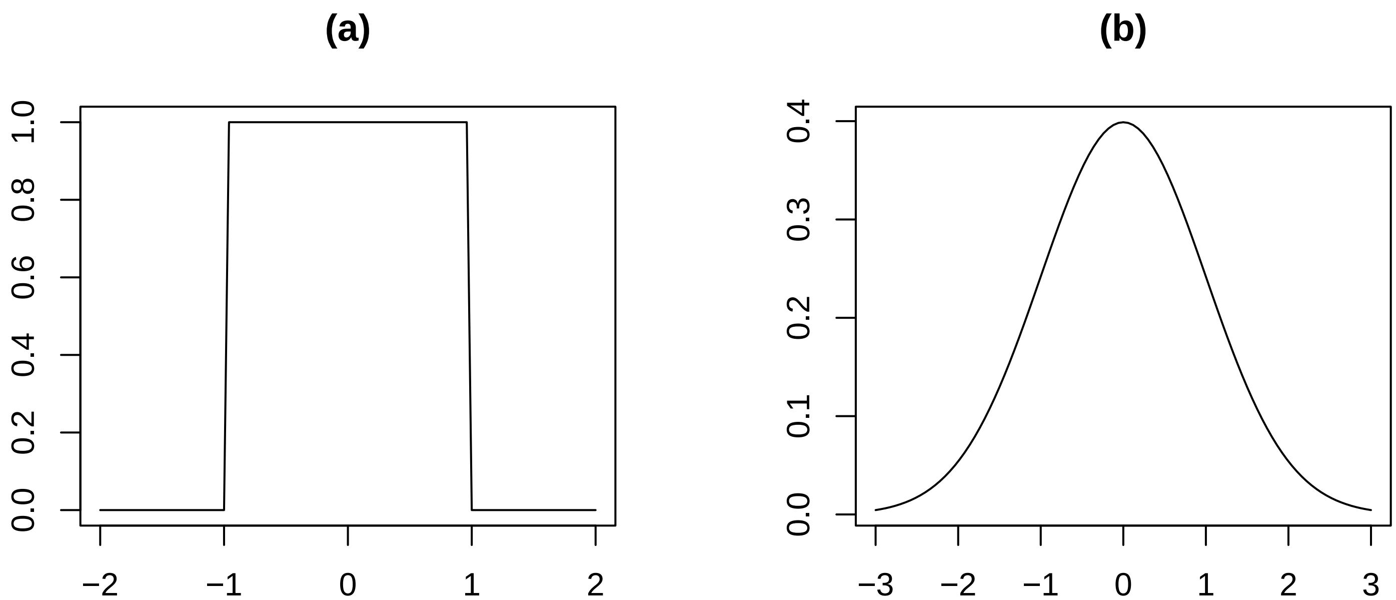
- can further generalize to

$$\widehat{\mathbb{E}}(y|x) = \frac{\displaystyle\sum_{i=1}^{n} \frac{1}{nh} K\left(\frac{x_i - x}{h}\right) y_i}{\displaystyle\sum_{i=1}^{n} \frac{1}{nh} K\left(\frac{x_i - x}{h}\right)},$$

where $K(u)$ is a kernel function such that

$$\int K(u)du = 1, \quad \int uK(u)du = 0, \quad \int u^2 K(u)du < \infty$$

- simple average vs weighted average



(a) $K(u) = \frac{1}{2} I(|u| < 1)$; (b) $K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$.

Bias-Variance Analysis

- typical model assumption:

$$y_i = f(x_i) + \varepsilon_i$$

$$\mathbb{E}(\varepsilon_i) = 0, \quad \mathbb{V}\mathrm{ar}(\varepsilon_i) = \sigma^2, \quad \mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0 \text{ for } i \neq j$$

- let

$$w_i = \frac{\frac{1}{nh} K\left(\frac{x_i - x}{h}\right)}{\sum_{j=1}^{n} \frac{1}{nh} K\left(\frac{x_j - x}{h}\right)} \quad \text{so that} \quad \widehat{f}(x) = \sum_{i=1}^{n} w_i y_i$$

- further simplifying assumption: $x_i \sim \mathrm{Unif}(0,1)$

Bias

$$\widehat{f}(x) = \sum w_i y_i \quad \Rightarrow \quad \mathbb{E}\left[\widehat{f}(x)\right] = \sum w_i \mathbb{E}(y_i) = \sum w_i f(x_i)$$

$$f(x_i) \approx f(x) + (x_i - x) f^{'}(x) + \frac{1}{2}(x_i - x)^2 f^{''}(x)$$

$$\Downarrow$$

$$\mathbb{E}\left[\widehat{f}(x)\right] \approx$$

$$f(x) \underbrace{\sum w_i}_{=1} + f^{'}(x) \underbrace{\sum w_i (x_i - x)}_{\approx 0} + \frac{1}{2} f^{''}(x) \underbrace{\sum w_i (x_i - x)^2}_{\approx h^2 [\int u^2 K(u) du]}$$

$$\therefore \quad \text{Bias}\left[\widehat{f}(x)\right] \approx h^2 B_0 \quad \text{where} \quad B_0 = \frac{1}{2} f^{''}(x) \left[\int u^2 K(u) du\right]$$

Variance

$$\widehat{f}(x) = \sum w_i y_i \quad \Rightarrow$$

$$\mathbb{V}\text{ar}\left[\widehat{f}(x)\right] = \sum w_i^2 \mathbb{V}\text{ar}(y_i) = \sigma^2 \underbrace{\left[\sum w_i^2\right]}_{\approx \frac{1}{nh}\left[\int K^2(u)du\right]}$$

$$\therefore \quad \mathbb{V}\text{ar}\left[\widehat{f}(x)\right] \approx \frac{1}{nh} V_0 \quad \text{where} \quad V_0 = \sigma^2 \left[\int K^2(u)du\right]$$

Discussion

$$h \uparrow \quad \Rightarrow \quad \text{bias} \uparrow \text{ and variance } \downarrow$$

$$h \downarrow \quad \Rightarrow \quad \text{bias} \downarrow \text{ and variance } \uparrow$$

Are these intuitively "obvious"?

Some Details

First,

$$\sum w_i(x_i - x) = \frac{\sum \frac{1}{nh} K\left(\frac{x_i - x}{h}\right)(x_i - x)}{\sum \frac{1}{nh} K\left(\frac{x_i - x}{h}\right)},$$

where

$$\text{numerator} \approx \int \left(\frac{v-x}{h}\right) K\left(\frac{v-x}{h}\right) dv \overset{(*)}{=} h \int uK(u)du = 0,$$

and

$$\text{denominator} \approx \int \frac{1}{h} K\left(\frac{v-x}{h}\right) dv \overset{(*)}{=} \int K(u)du = 1.$$

(*) $u = (v-x)/h$, $du = (1/h)dv$

Some Details

Likewise,

$$\sum w_i (x_i - x)^2 = \frac{\sum \frac{1}{nh} K\left(\frac{x_i - x}{h}\right)(x_i - x)^2}{\sum \frac{1}{nh} K\left(\frac{x_i - x}{h}\right)},$$

where

$$\text{denominator} \approx 1 \qquad \text{(previous slide)}$$

and

$$\text{numerator} \approx \int h \left(\frac{v - x}{h}\right)^2 K\left(\frac{v - x}{h}\right) dv = h^2 \int u^2 K(u) du.$$

Some Details

Exercise Use the same argument to "show" that

$$\sum w_i^2 \approx \frac{1}{nh}\left[\int K^2(u)du\right].$$

Bias-Variance Trade-Off

$$\begin{aligned}
\text{MSE}(\widehat{f}) &\equiv \mathbb{E}(\widehat{f} - f)^2 \\
&= \mathbb{E}[\widehat{f} - \mathbb{E}(\widehat{f}) + \mathbb{E}(\widehat{f}) - f]^2 \\
&= \underbrace{\mathbb{E}[\widehat{f} - \mathbb{E}(\widehat{f})]^2}_{\mathbb{V}\text{ar}(\widehat{f})} + \underbrace{[\mathbb{E}(\widehat{f}) - f]^2}_{\text{Bias}^2(\widehat{f})} + 2\underbrace{\mathbb{E}[(\widehat{f} - \mathbb{E}(\widehat{f}))(\mathbb{E}(\widehat{f}) - f)]}_{=0}
\end{aligned}$$

Exercise Show that $\mathbb{E}[(\widehat{f} - \mathbb{E}(\widehat{f}))(\mathbb{E}(\widehat{f}) - f)] = 0$.

Bias-Variance Trade-Off

- for kernel regression,

$$\text{MSE} = \mathbb{V}\text{ar} + \text{Bias}^2 \approx h^4 B_0^2 + \frac{V_0}{nh}$$

- can find the "optimal" h (in terms of the MSE):

$$\frac{d}{dh}\text{MSE} \approx 4B_0^2 h^3 - \frac{V_0}{nh^2} = 0 \quad \Rightarrow \quad h^* \sim O(n^{-1/5})$$

- general phenomenon, not just for kernel regression

Curse of Dimensionality

For $\boldsymbol{x} \in \mathbb{R}^d$, neighborhood-based methods such as kernel regression still apply ("just" use $K(\boldsymbol{u})$ for $\boldsymbol{u} \in \mathbb{R}^d$) but they become increasingly difficult.

Example Suppose data are uniformly distributed inside the unit ball, $\{\boldsymbol{x} : \|\boldsymbol{x}\| \leq 1\}$. Consider a neighborhood around $\mathbf{0}$ with radius $h < 1$. What fraction of data does the neighborhood contain?

$$(\text{fraction of data}) = \frac{\text{vol(neighborhood)}}{\text{vol(unit ball)}} = \frac{\dfrac{\pi^{d/2}}{\Gamma(d/2+1)} h^d}{\dfrac{\pi^{d/2}}{\Gamma(d/2+1)} 1^d} = h^d.$$

Thus, in $d = 100$ dimensions, even a neighborhood with radius $h = 0.95$ contains $< 0.6\%$ of the data.

Classification

- training data $\{(\boldsymbol{x}_i, y_i); i = 1, 2, ..., n\}$
- $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, 2, ..., K\}$
- want $f(\cdot)$ so that we can classify y from $\boldsymbol{x}$ with $f(\boldsymbol{x})$
- mean 0-1 error,

$$\text{error}(f) = \mathbb{E}[I(y \neq f(\boldsymbol{x}))],$$

a common criterion for how good f is

$\mathbb{P}(y|\boldsymbol{x})$

Exercise Show that the function that minimizes the mean 0-1 error is

$$f(\boldsymbol{x}) = \arg\max_{k=1,\ldots,K} \mathbb{P}(y = k|\boldsymbol{x}).$$

$$\Downarrow$$

main task for classification: "go after" the function, $\mathbb{P}(y|\boldsymbol{x})$

Remark For binary $y \in \{0, 1\}$, also have $\mathbb{E}(y|\boldsymbol{x}) = \mathbb{P}(y = 1|\boldsymbol{x})$.

Two Strategies

- "go after" $\mathbb{P}(y|\boldsymbol{x})$ directly

- use Bayes theorem,

$$\mathbb{P}(y = k|\boldsymbol{x}) = \frac{\pi_k p_k(\boldsymbol{x})}{\pi_1 p_1(\boldsymbol{x}) + ... + \pi_K p_K(\boldsymbol{x})},$$

  and "go after" $\mathbb{P}(y|\boldsymbol{x})$ indirectly by first "going after"

  – $p_k(\boldsymbol{x})$, the conditional distribution of $\boldsymbol{x}|y = k$, and

  – π_k, the prior probability of class k,

  for each $k = 1, ..., K$

Ex IV: Logistic Regression

- for binary $y \in \{0, 1\}$, can model

$$\mathbb{P}(y = 1|\boldsymbol{x}) = \frac{\exp\{\alpha + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}\}}{1 + \exp\{\alpha + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}\}},$$

  or, equivalently,

$$\log \frac{\mathbb{P}(y = 1|\boldsymbol{x})}{\mathbb{P}(y = 0|\boldsymbol{x})} = \alpha + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}$$

- just have to estimate α and $\boldsymbol{\beta}$ from training data

Ex V: Linear Discriminant Analysis

- alternatively, can model

$$p_k(\boldsymbol{x}) \quad \sim \quad \mathrm{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}), \quad k = 0, 1$$

- recall multivariate normal density function

$$p_k(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k)\right]$$

Ex V: Linear Discriminant Analysis

- then,

$$\log \frac{\mathbb{P}(y=1|\boldsymbol{x})}{\mathbb{P}(y=0|\boldsymbol{x})} \quad = \quad = \log \frac{\pi_1}{\pi_0} + \log \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})}$$

where

$$\log \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} = -\frac{1}{2}[(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) - (\boldsymbol{x}-\boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_0)]$$

- just have to estimate $\pi_1, \pi_0, \boldsymbol{\mu}_1, \boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}$ (actually, $\boldsymbol{\Sigma}^{-1}$)

Comparison

- slight rearrangement of the last equation from the previous slide gives

$$\log \frac{\mathbb{P}(y=1|\boldsymbol{x})}{\mathbb{P}(y=0|\boldsymbol{x})} = \underbrace{\log \frac{\pi_1}{\pi_0} - \frac{1}{2}\left(\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0\right)}_{\alpha} + \underbrace{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x}}_{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}}$$

- linear discriminant analysis and logistic regression: two different ways of "going after" the same, linear decision boundary

Which Is Better?

B. Efron (1975), "The efficiency of logistic regression compared to normal discriminant analysis", *JASA* **70**, pp. 892–898.

- $n \to \infty$; fixed d
- if p_k normal, logistic regression less efficient
- loss of efficiency between $1/3$ to $1/2$

Enter "Big Data"

- if all this sounds easy, don't forget $\mathbf{\Sigma}$ is $d \times d$
- very hard for relatively large d
- $\mathbf{\Sigma}^{-1}$ can be estimated by the graphical LASSO (Friedman, Hastie & Tibshirani, 2008; *Biostatistics*)
- Fan, Feng & Wu (2009; *Ann. Appl. Stat.*) applied gLASSO-estimated $\mathbf{\Sigma}^{-1}$ to perform linear discriminant analysis
- Cai & Liu (2012; *JASA*) proposed to estimate $\boldsymbol{\beta} \equiv \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$ directly with sparsity constraints

Research Perform an analysis like that of Efron (1975) when $d \to \infty$ as well.

Ex VI: Naïve Bayes

- may feel uncomfortable with assuming $p_k(\boldsymbol{x}) \sim \mathrm{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$

- data scientists have long used (and still like) the model,

$$p_k(\boldsymbol{x}) = \prod_{j=1}^{d} f_{k,j}(x_j),$$

  where each $f_{k,j}(\cdot)$ can be estimated separately

- especially helpful if the predictors are of mixed types (e.g., some continuous, some categorical)

Ex VI: Naïve Bayes

- may feel uncomfortable with assuming independence
- but

$$\log \frac{\mathbb{P}(y=1|\boldsymbol{x})}{\mathbb{P}(y=0|\boldsymbol{x})} = \underbrace{\log \frac{\pi_1}{\pi_0}}_{\alpha} + \sum_{j=1}^{d} \underbrace{\log \frac{f_{1,j}(x_j)}{f_{0,j}(x_j)}}_{g_j(x_j)}$$

$$\equiv \alpha + \sum_{j=1}^{d} g_j(x_j),$$

and most people comfortable with generalizing linear logistic regression to additive logistic regression

Ex VII: Neural Networks

- sigmoid function

$$\sigma(u) = \frac{e^u}{1+e^u}$$

- hidden layer $\ell = 1, 2, ..., L-1$

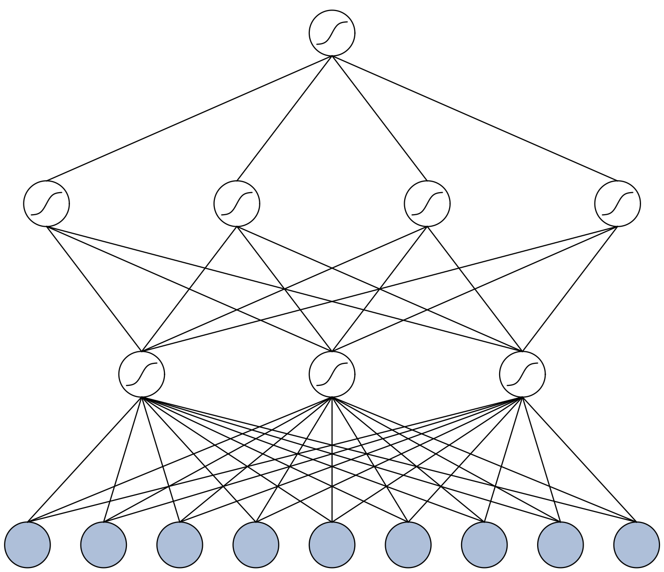
$$z_t^{(\ell)} = \sigma\left(\alpha_t^{(\ell)} + \sum_b w_{t,b}^{(\ell)} z_b^{(\ell-1)}\right)$$

- top layer L

$$z_t^{(L)} = \mathbb{P}(y = t|...)$$

- bottom layer 0

$$z_b^{(0)} = x_b, \quad b = 1, 2, ..., d$$



Ex VIII: Nearest-Neighbor Classifier

- can also estimate $\mathbb{P}(y|\boldsymbol{x})$ by

$$\widehat{\mathbb{P}}(y = k|\boldsymbol{x}) \quad = \quad \text{fraction}\,\{y_i = k : \boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{x})\}\,,$$

  where $\mathcal{N}(\boldsymbol{x})$ denotes a "neighborhood" around $\boldsymbol{x}$

Bayes Error

Myth If my misclassification error is not very close to zero, my classifier must not be very good.

Myth If I know the true model, my misclassification error must be zero.

Truth Even if we knew $\mathbb{P}(y|\boldsymbol{x})$ (or $p_k(\boldsymbol{x})$, π_k, ...) perfectly, we might still have considerable misclassification error — these errors are called the Bayes error.

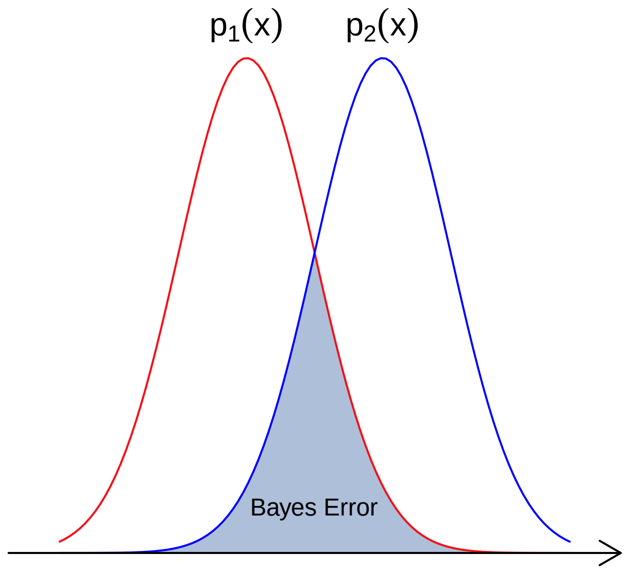
Bayes Error

$$x \in \mathbb{R}$$

$$\pi_1 = \pi_0 = 1/2$$

$$p_1(x) \sim \mathrm{N}(\mu_1, \sigma^2)$$

$$p_0(x) \sim \mathrm{N}(\mu_0, \sigma^2)$$



Exercise How does the Bayes error change with $\Delta \equiv |\mu_1 - \mu_0|$, and with σ^2?

Question Can we reduce the Bayes error?

Summary

- key ideas:
  - regression; mean squared error; $\mathbb{E}(y|\boldsymbol{x})$
  - bias-variance trade-off; curse of dimensionality
  - classification; mean 0-1 error; $\mathbb{P}(y|\boldsymbol{x})$; Bayes error
- specific methods:
  - linear regression; nearest-neighbors; kernel regression
  - logistic regression; linear discriminant analysis
  - naïve Bayes; neural network
  - graphical LASSO
- didn't discuss:
  - actual estimation procedures

Next ...

- course administration, logistics, etc