Lecture 1 – Part 2
Some Basics

(January 9, 2015)

Mu Zhu
University of Waterloo
Regression

- **training data** \( \{ (x_i, y_i); i = 1, 2, ..., n \} \)

- \( x_i \in \mathbb{R}^d \) and \( y_i \in \mathbb{R} \)

- want \( f(\cdot) \) so that we can **predict** \( y \) from \( x \) with \( f(x) \)

- **mean squared error**, \( \text{MSE}(f) = \mathbb{E}[y - f(x)]^2 \), a common criterion for how good \( f \) is
Let $g(x) = \mathbb{E}(y|x)$, and $h(x)$ be any other function of $x$. Then,

$$\mathbb{E}[y - h(x)]^2$$

$$= \mathbb{E}[y - g(x) + g(x) - h(x)]^2$$

$$= \mathbb{E}[y - g(x)]^2 + \mathbb{E}[g(x) - h(x)]^2 + 2 \mathbb{E}[(y - g(x))(g(x) - h(x))]$$

$$\geq 0$$

$$= 0$$

$$\geq \mathbb{E}[y - g(x)]^2.$$ 

\[\downarrow\]

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

**Exercise** Show that $\mathbb{E}[(y - g(x))(g(x) - h(x))] = 0$. (**Hint:** Use $\mathbb{E}(\cdot) = \mathbb{E}[\mathbb{E}(\cdot|\mathbf{x})]$.)
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 $\alpha$ and $\beta$ from training data

- for $x \in \mathbb{R}^d$, simply
  \[ f(x) = \alpha + \beta^T x \]
Ex II: Nearest-Neighbor Regression

• may feel uncomfortable with assuming $\mathbb{E}(y|x)$ to be linear

• can choose to estimate $\mathbb{E}(y|x)$ by

$$\hat{\mathbb{E}}(y|x) = \text{average} \{ y_i : x_i \in \mathcal{N}(x) \},$$

where $\mathcal{N}(x)$ denotes a “neighborhood” around $x$

• meaning of $\mathbb{E}(y|x)$: the average of $y$ given a particular $x$

• almost literal interpretation of $\mathbb{E}(y|x)$

• relaxes “given a particular $x$” to “within a neighborhood of $x$”
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

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

- need to specify $h$ a priori ... called a tuning parameter
Ex III: Kernel Regression

- can further generalize to

\[ \hat{E}(y|x) = \sum_{i=1}^{n} \frac{1}{nh} K \left( \frac{x_i - x}{h} \right) y_i \]

\[ \sum_{i=1}^{n} \frac{1}{nh} K \left( \frac{x_i - x}{h} \right) \]

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

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

- simple average vs weighted average
(a) $K(u) = \frac{1}{2} I(|u| < 1); \quad$ (b) $K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$. 
Bias-Variance Analysis

• typical model assumption:

\[ y_i = f(x_i) + \varepsilon_i \]

\[ \mathbb{E}(\varepsilon_i) = 0, \quad \text{Var}(\varepsilon_i) = \sigma^2, \quad \text{Cov}(\varepsilon_i, \varepsilon_j) = 0 \text{ for } i \neq j \]

• let

\[ w_i = \frac{1}{nh} K \left( \frac{x_i - x}{h} \right) \]

\[ \sum_{j=1}^{n} \frac{1}{nh} K \left( \frac{x_j - x}{h} \right) \]

so that

\[ \hat{f}(x) = \sum_{i=1}^{n} w_i y_i \]

• further simplifying assumption: \( x_i \sim \text{Unif}(0, 1) \)
Bias

\[ \hat{f}(x) = \sum w_i y_i \implies \mathbb{E} \left[ \hat{f}(x) \right] = \sum w_i \mathbb{E}(y_i) = \sum w_i f(x_i) \]

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

\[ \downarrow \]

\[ \mathbb{E} \left[ \hat{f}(x) \right] \approx f(x) \sum w_i + f'(x) \sum w_i (x_i - x) + \frac{1}{2} f''(x) \sum w_i (x_i - x)^2 \approx h^2 \left[ \int u^2 K(u) du \right] \]

\[ \therefore \text{Bias} \left[ \hat{f}(x) \right] \approx h^2 B_0 \quad \text{where} \quad B_0 = \frac{1}{2} f''(x) \left[ \int u^2 K(u) du \right] \]
\[ \hat{f}(x) = \sum w_i y_i \quad \Rightarrow \quad \text{Var} \left[ \hat{f}(x) \right] = \sum w_i^2 \text{Var}(y_i) = \sigma^2 \left[ \sum w_i^2 \right] \approx \frac{1}{nh} \left[ \int K^2(u) du \right] \]

\[ \therefore \quad \text{Var} \left[ \hat{f}(x) \right] \approx \frac{1}{nh} V_0 \quad \text{where} \quad V_0 = \sigma^2 \left[ \int K^2(u) du \right] \]
Discussion

\[ h \uparrow \Rightarrow \text{bias} \uparrow \text{ and variance} \downarrow \]

\[ h \downarrow \Rightarrow \text{bias} \downarrow \text{ and variance} \uparrow \]

Are these intuitively “obvious”?
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

\( \text{denominator} \approx 1 \quad \text{(previous slide)} \)

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

\[ \text{MSE}(\hat{f}) \equiv E(\hat{f} - f)^2 \]
\[ = E[\hat{f} - E(\hat{f}) + E(\hat{f}) - f]^2 \]
\[ = E[\hat{f} - E(\hat{f})]^2 + [E(\hat{f}) - f]^2 + 2 E[(\hat{f} - E(\hat{f}))(E(\hat{f}) - f)] \]
\[ \text{Var}(\hat{f}) \quad \text{Bias}^2(\hat{f}) \quad \text{= 0} \]

Exercise  Show that \( E[(\hat{f} - E(\hat{f}))(E(\hat{f}) - f)] = 0. \)
Bias-Variance Trade-Off

• for kernel regression,

\[ \text{MSE} = \text{Var} + \text{Bias}^2 \approx h^4 B_0^2 + \frac{V_0}{nh} \]

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

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

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

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

- training data $\{(\mathbf{x}_i, y_i); i = 1, 2, \ldots, n\}$

- $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, 2, \ldots, K\}$

- want $f(\cdot)$ so that we can classify $y$ from $\mathbf{x}$ with $f(\mathbf{x})$

- mean 0-1 error,

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

a common criterion for how good $f$ is
Exercise  Show that the function that minimizes the mean 0-1 error is

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

\[ \Downarrow \]

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

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

- “go after” \( P(y|x) \) directly

- use Bayes theorem,

\[
P(y = k|x) = \frac{\pi_k p_k(x)}{\pi_1 p_1(x) + \ldots + \pi_K p_K(x)},
\]

and “go after” \( P(y|x) \) indirectly by first “going after”

- \( p_k(x) \), the conditional distribution of \( x|y = k \), and
- \( \pi_k \), the prior probability of class \( k \),

for each \( k = 1, \ldots, K \)
Ex IV: Logistic Regression

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

\[
\mathbb{P}(y = 1|x) = \frac{\exp\{\alpha + \beta^T x\}}{1 + \exp\{\alpha + \beta^T x\}}
\]

or, equivalently,

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

- just have to estimate $\alpha$ and $\beta$ from training data
Ex V: Linear Discriminant Analysis

- alternatively, can model

\[ p_k(x) \sim \mathcal{N}(\mu_k, \Sigma), \quad k = 0, 1 \]

- recall multivariate normal density function

\[
p_k(x) = \frac{1}{\sqrt{\left(2\pi\right)^d |\Sigma|}} \exp \left[ -\frac{1}{2} (x - \mu_k)^T \Sigma^{-1} (x - \mu_k) \right]
\]
Ex V: Linear Discriminant Analysis

- then,

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

where

$$\log \frac{p_1(x)}{p_0(x)} = -\frac{1}{2}[(x - \mu_1)^T \Sigma^{-1} (x - \mu_1)$$

$$- (x - \mu_0)^T \Sigma^{-1} (x - \mu_0)]$$

- just have to estimate $\pi_1, \pi_0, \mu_1, \mu_0$ and $\Sigma$ (actually, $\Sigma^{-1}$)
Comparison

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

\[
\log \frac{P(y = 1|x)}{P(y = 0|x)} = \log \frac{\pi_1}{\pi_0} - \frac{1}{2} \left( \mu_1^T \Sigma^{-1} \mu_1 - \mu_0^T \Sigma^{-1} \mu_0 \right) \\
\quad + (\mu_1 - \mu_0)^T \Sigma^{-1} x
\]

• linear discriminant analysis and logistic regression: two different ways of “going after” the same, linear decision boundary
Which Is Better?


- $n \to \infty$; 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 $\Sigma$ is $d \times d$
- very hard for relatively large $d$
- $\Sigma^{-1}$ can be estimated by the graphical LASSO (Friedman, Hastie & Tibshirani, 2008; *Biostatistics*)
- Fan, Feng & Wu (2009; *Ann. Appl. Stat.*) applied gLASSO-estimated $\Sigma^{-1}$ to perform linear discriminant analysis
- Cai & Liu (2012; *JASA*) proposed to estimate $\beta \equiv \Sigma^{-1}(\mu_1 - \mu_0)$ 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 $p_k(x) \sim N(\mu_k, \Sigma)$
- data scientists have long used (and still like) the model,

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

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

\[
\log \frac{\mathbb{P}(y = 1 | \mathbf{x})}{\mathbb{P}(y = 0 | \mathbf{x})} = \log \frac{\pi_1}{\pi_0} + \sum_{j=1}^{d} \log \frac{f_{1,j}(x_j)}{f_{0,j}(x_j)}
\]

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

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 $\ell = 1, 2, \ldots, L - 1$

$$z^{(\ell)}_t = \sigma \left( \alpha^{(\ell)}_t + \sum_b w^{(\ell)}_{t,b} z^{(\ell-1)}_b \right)$$

- top layer $L$

$$z^{(L)}_t = \mathbb{P}(y = t | \ldots)$$

- bottom layer 0

$$z^{(0)}_b = x_b, \quad b = 1, 2, \ldots, d$$
Ex VIII: Nearest-Neighbor Classifier

- can also estimate $P(y|x)$ by

$$\hat{P}(y = k|x) = \text{fraction} \{ y_i = k : x_i \in N(x) \},$$

where $N(x)$ denotes a “neighborhood” around $x$. 
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 $P(y|x)$ (or $p_k(x)$, $\pi_k$, ...) perfectly, we might still have considerable misclassification error — these errors are called the **Bayes error**.
Bayes Error

\[ x \in \mathbb{R} \]
\[ \pi_1 = \pi_0 = 1/2 \]
\[ p_1(x) \sim N(\mu_1, \sigma^2) \]
\[ p_0(x) \sim N(\mu_0, \sigma^2) \]

**Exercise**  How does the Bayes error change with \( \Delta \equiv |\mu_1 - \mu_0| \), and with \( \sigma^2 \)?

**Question**  Can we reduce the Bayes error?
Summary

• key ideas:
  – regression; mean squared error; \( \mathbb{E}(y|x) \)
  – bias-variance trade-off; curse of dimensionality
  – classification; mean 0-1 error; \( \mathbb{P}(y|x) \); 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