Classification Methods

Andrew Nobel

October, 2021

Overview

Given: Data set $D_n = (x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \{0, 1\}$

Task: Produce a classification rule $\hat{\phi}_n(x) = \phi_n(x : D_n)$ from data D_n

Classification Procedures

1. Non-parametric: Histogram rules, Nearest Neighbor rules

2. Based on distributional assumptions

   - Naive Bayes: conditional independence of features given the response
   - LDA and QDA: multivariate normality of class conditional distributions
   - Logistic Regression: linearity of log-odds ratio

Assessing Performance

Task: Assess performance of rule $\hat{\phi}_n$ produced from data set D_n

Approach 1: Training error

- Examine error rate $n^{-1} \sum_{i=1}^{n} \mathbb{I}(\hat{\phi}_n(X_i) \neq Y_i)$ of rule on D_n
- Tends to be optimistic as $\hat{\phi}_n$ was trained on D_n

Approach 2: Test error

- Let $D_m = (\tilde{X}_1, \tilde{Y}_1), \ldots, (\tilde{X}_m, \tilde{Y}_m)$ be a test set independent of D_n
- Consider error rate $m^{-1} \sum_{j=1}^{m} \mathbb{I}(\hat{\phi}_n(\tilde{X}_j) \neq \tilde{Y}_j)$ of rule on test data
- More accurate than training error, requires additional observations

Histogram Rules

Histogram Rules

- Observations $D_n = (X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{0, 1\}$
- Partition $\pi = \{A_1, \ldots, A_K\}$ of $\mathcal{X}$ into disjoint sets called cells
- Let $\pi(x) =$ cell A_k of π containing x

Definition: The histogram classification rule for π is given by

$$\phi_n^\pi(x : D_n) \;=\; \hat{\phi}_n^\pi(x) \;=\; \text{maj-vote}\{Y_i : X_i \in \pi(x)\}$$

- Classifies x using "local" data in the same cell as x
- No assumptions about the distribution of (X, Y)
- Decision regions of rule determined by cells of π

Histogram Rules, Theory

Fact: When n is large, the histogram rule

$$\hat{\phi}_n^\pi(x) \;\approx\; \phi_\pi^*(x) \;:=\; \begin{cases} 1 & \text{if } \mathbb{P}(Y = 1 \,|\, X \in \pi(x)) \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Upshot: For large n the histogram rule mimics a “lumpy” version of the Bayes rule based on the partition π

Modifications and Extensions

- Let partition π depend on the *number* of observations
- Decision trees and random forests select π based on D_n

Nearest Neighbor Rules

Nearest Neighbor Rules

Idea: Classify $x \in \mathbb{R}^d$ based on the labels of the nearest feature vectors in the dataset: if it walks like a duck and quacks like a duck...

Observations: $D_n = (X_1, Y_1), \ldots, (X_n, Y_n) \in \mathbb{R}^d \times \{0, 1\}$

Defn: For $x \in \mathbb{R}^d$ let $X_{(1)}(x), \ldots, X_{(n)}(x)$ be reordering of $X_1, \ldots, X_n$ s.t.

$$||x - X_{(1)}(x)|| \leq ||x - X_{(2)}(x)|| \leq \cdots \leq ||x - X_{(n)}(x)||$$

and let $Y_{(j)}(x) =$ label of $X_{(j)}(x)$.

Terminology: $X_{(k)}(x)$ called k*th nearest neighbor of* x

Nearest Neighbor Rules

Definition: For $k \geq 1$ odd, the *k-nearest neighbor rule* takes a majority vote among the class labels of the k nearest neighbors of x, that is

$$\phi_n^{\text{k-NN}}(x : D_n) \;=\; \hat{\phi}_n^{\text{k-NN}}(x) \;=\; \text{majority-vote}\{Y_{(1)}(x), \ldots, Y_{(k)}(x)\}$$

Special case $k = 1$ yields *1-nearest neighbor rule* $\hat{\phi}_n^{\text{1-NN}}(x) \;=\; Y_{(1)}(x)$

- NN-rules rely on local information to classify feature vector x
- Choice of k determines how local estimates are
- No assumptions about distribution of (X, Y)
- Decision regions of NN rules are complicated

Asymptotic Performance of 1-NN Rule

Theorem (T. Cover): As the number of samples n tends to infinity,

$$\mathbb{E}R(\hat{\phi}_n^{\text{1-NN}}) \;\to\; 2\mathbb{E}[\eta(X)(1-\eta(X))] \;\leq\; 2R^*$$

In words, the asymptotic probability of error of the 1-NN rule is at most twice the Bayes risk (the best performance of any classification rule)!

Example: MNIST Database

MNIST database (LeCun, Cortes, Burges)

- Images of handwritten digits (0-9)
- Each image is 28×28 matrix of gray-scale pixel intensities
- Pixel intensity is an integer between 0 (white) and 255 (black)

Example: Handwritten Digits

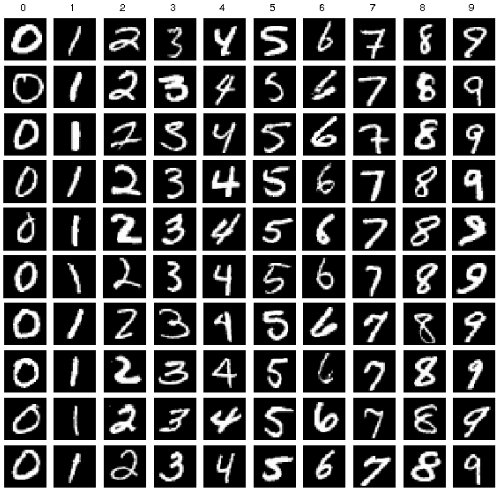
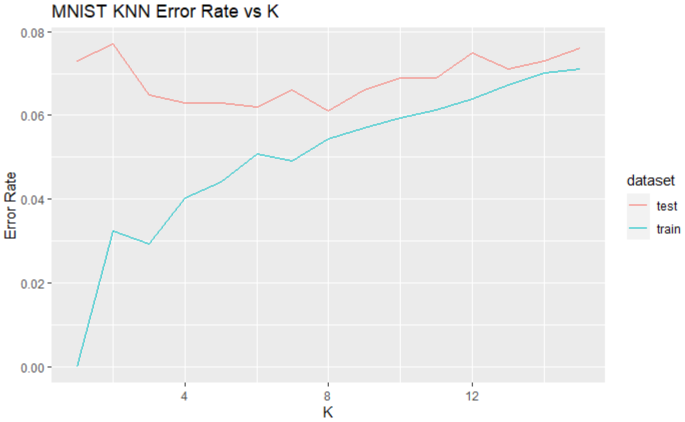


Figure: Examples of labeled digits (S.R. Young)

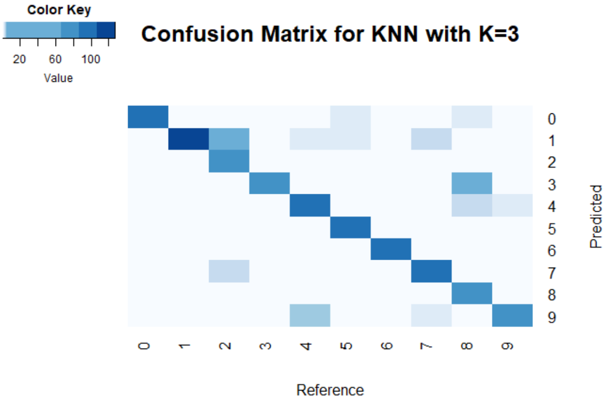
MNIST Training and Test Sets

Digit	Train	Test
0	476	105
1	617	130
2	508	98
3	488	75
4	460	108
5	447	99
6	489	91
7	523	111
8	478	89
9	514	94

Performance of kNN on MNIST



Confusion Matrix of kNN with $k = 3$



Overview: Classification Methods from Stochastic Assumptions

Begin with assumptions about class-conditional distributions f_0, f_1 or conditional probability η resulting in simplified statistical model

$$\Downarrow$$

Use training data D_n to fit statistical model via MLE or gradient descent, and to estimate π_0, π_1 if needed

$$\Downarrow$$

Produce estimate $\hat{\eta}$ of η using fitted model

$$\Downarrow$$

Classify new samples following Bayes rule, using $\hat{\eta}$ instead of η,

$$\hat{\phi}_n(x) = \begin{cases} 1 & \text{if } \hat{\eta}(x) \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Naive Bayes

Naive Bayes

Setting: Observe (X, Y) where $X = (X_1, \ldots, X_d)^t$ has d components

Assumption: Given label Y components $X_1, \ldots, X_d$ of X are independent

Equivalently, class-conditional distributions factor as a product of univariate distributions. For $k = 0, 1$

$$f(x_1, \ldots, x_d \,|\, Y = k) \;=\; f_1(x_1 \,|\, Y = k) \cdots f_d(x_d \,|\, Y = k)$$

Approach

- Estimate marginal distributions $f_j(x_j \,|\, Y = k)$ one at a time
- Estimate $f(x \,|\, Y = k)$ by a product of the marginal estimates
- Combine with estimates of π_0, π_1 to approximate Bayes rule

Estimating Marginal (Univariate) Distributions

Parametric: Assume marginal distribution comes from a parametric family

- Estimate parameters using MLE or method of moments
- Plug in parameters to get estimate of distribution (pmf or pdf)

Non-Parametric: No assumptions about univariate distribution

- Discrete case: Estimate mass function using relative frequencies
- Continuous case: Use histogram or kernel methods to estimate density

Outline of Naive Bayes

Observations: $(X_1, Y_1), \ldots, (X_n, Y_n)$ where $X_i = (X_{i1}, \ldots, X_{id})^t$

Step 1: Estimate prior of class k by $\hat{\pi}_k = n^{-1} \sum_{i=1}^n \mathbb{I}(Y_i = k)$

Step 2: For $1 \leq j \leq d$ and $k = 0, 1$ use univariate data $\{X_{ij} : Y_i = k\}$ to form estimate $\hat{f}_j(x_j \,|\, Y = k)$. Estimate class conditional by product

$$\hat{f}(x \,|\, Y = k) = \prod_{j=1}^{d} \hat{f}_j(x_j \,|\, Y = k)$$

Step 3: Define $\hat{\phi}_n^{\mathrm{NB}}(x) = \mathrm{argmax}_{k=0,1} \hat{\mathbb{P}}(Y = k \,|\, X = x)$ where

$$\hat{\mathbb{P}}(Y = k \,|\, X = x) \;=\; \frac{\hat{\pi}_k \hat{f}(x \,|\, Y = k)}{\hat{\pi}_0 \hat{f}(x \,|\, Y = 0) + \hat{\pi}_1 \hat{f}(x \,|\, Y = 1)}$$

Naive Bayes, Smoking Cessation

Example: Predict who will benefit from smoking cessation program

Observation: Response $Y \in \{0, 1\}$, feature vector X with components

- Usage $U \in \{1, \ldots, 10\} \times 10$ cigarettes/day, model with general pmf
- Number $A \in \{0, 1, \ldots\}$ of previous attempts to quit, model as Poisson
- Time $T \in (0, \infty)$ in days since last attempt to quit, model as Exponential

Naive Bayes: Assume that class conditionals factor as

$$\mathbb{P}(X = (u, a, t)^t \,|\, Y = k) = p_k(u)\, q_k(a)\, f_k(t)$$

Naive Bayes, Estimating Marginal Distributions

1. Estimate pmf of usage based on relative frequencies

$$\hat{p}_k(u) \;=\; \sum_{i=1}^{n} \mathbb{I}(u_i = u \text{ and } y_i = k) / \sum_{i=1}^{n} \mathbb{I}(y_i = k)$$

2. Estimate pmf of quitting attempts by $\hat{q}_k = \text{Poiss}(\hat{\lambda}_k)$ where

$$\hat{\lambda}_k \;=\; \sum_{i=1}^{n} a_i \, \mathbb{I}(y_i = k) / \sum_{i=1}^{n} \mathbb{I}(y_i = k)$$

3. Estimate density of time since last attempt by $\hat{f}_k(t) = \text{Exp}(\hat{\gamma}_k)$ where

$$\hat{\gamma}_k \;=\; \left(\sum_{i=1}^{n} t_i \, \mathbb{I}(y_i = k) / \sum_{i=1}^{n} \mathbb{I}(y_i = k) \right)^{-1}$$

Naive Bayes, Pluses and Minuses

Minuses: Naive Bayes is based on strong assumption of conditional independence of features, which does not hold in most settings

Pluses

- Conditional independence may hold approximately in some cases
- NB classifier is fast/easy to compute
- Easily handles mix of discrete, categorical, continuous features
- Does not require intimate domain knowledge
- Not affected by features that are independent of class label

Linear and Quadratic Discriminant Analysis

Hyperplanes and Half-Spaces

Definition: Given vector $\mathbf{u} \in \mathbb{R}^d$ with $||u|| = 1$ and $b \in \mathbb{R}$ let

- Hyperplane $H = \{x : \langle x, u \rangle = b\}$
- Half-space $H_+ = \{x : \langle x, u \rangle > b\}$ contains points "above" H
- Half-space $H_- = \{x : \langle x, u \rangle < b\}$ contains points "below" H

Note

- $\mathbf{u}$ called *normal vector*, b called *offset*
- H is translation of $(n-1)$-dimensional subspace $\{x : \langle x, u \rangle = 0\}$
- Signed distance from x to H is equal to $\langle x, u \rangle - b$

Another Look at the Bayes Rule

Fact: Bayes rule ϕ^* for pair (X, Y) is 1 if and only if

$$0 \ \leq \ \log \frac{\eta(x)}{1 - \eta(x)} \ = \ \log \frac{\mathbb{P}(Y = 1 | X = x)}{\mathbb{P}(Y = 0 | X = x)} \ = \ \log \frac{\pi_1 f_1(x)}{\pi_0 f_0(x)}$$

Thus the Bayes rule can be written as

$$\phi^*(x) \ = \ \mathbb{I}(\delta_1(x) \geq \delta_0(x)) \ = \ \operatorname*{argmax}_{k=0,1} \delta_k(x)$$

where $\delta_k(x) = \log(\pi_k \, f_k(x))$ is the *discriminant function* for class k. Decision boundary of Bayes rule given by

$$B = \{x : \delta_1(x) = \delta_0(x)\}$$

Overview: Linear and Quadratic Discriminant Analysis

Idea: Assume class-conditional densities are multivariate normal

$$f_k = \mathcal{N}_d(\mu_k, \Sigma_k) \text{ for } k = 0, 1$$

In this case the discriminant function $\delta_k(x) = \log(\pi_k \, f_k(x))$ has the form

$$\delta_k(x) = -\frac{1}{2} x^t \Sigma_k^{-1} x + \langle x, \Sigma_k^{-1} \mu_k \rangle - \frac{1}{2} \left\{ \log[(2\pi)^d \pi_k^{-2} \text{det}(\Sigma_k)] + \mu_k^t \Sigma_k^{-1} \mu_k \right\}$$

1. LDA: Assume that covariance matrices are equal, i.e., $\Sigma_0 = \Sigma_1$

2. QDA: Allow covariance matrices Σ_0 and Σ_1 to be different

Models for Linear and Quadratic Discriminant Analysis

Recall: Bayes rule $\phi^*(x) = \mathrm{argmax}_k \, \delta_k(x)$

LDA: Assume $\Sigma_0 = \Sigma_1 = \Sigma$. Then decision boundary of ϕ^* is a hyperplane

$$B \; = \; \{x : \delta_1(x) = \delta_0(x)\} \; = \; \{x : x^t \, \Sigma^{-1}(\mu_1 - \mu_0) + (c_0 - c_1) = 0\}$$

where c_0, c_1 are constants. [Quadratic terms in $\delta_0(x), \delta_1(x)$ cancel]

QDA: Allow $\Sigma_0 \neq \Sigma_1$. Decision boundary of ϕ^* is a quadratic surface

$$B = \left\{ x : -\frac{1}{2} x^t (\Sigma_1^{-1} - \Sigma_0^{-1}) x + x^t (\Sigma_1^{-1} \mu_1 - \Sigma_0^{-1} \mu_0) + (c_0 - c_1) = 0 \right\}$$

In practice: Estimate unknown quantities π_k, μ_k, and Σ_k via MLE

Using Data: Maximum Likelihood Estimates of Parameters

1. Prior probabilities: $\hat{\pi}_k = n^{-1} \sum_{i=1}^n \mathbb{I}(Y_i = k)$

2. Mean vectors: $\hat{\mu}_k = \sum_{i=1}^n X_i \, \mathbb{I}(Y_i = k) / \sum_{j=1}^n \mathbb{I}(Y_j = k)$

3. Variance matrix: Individual/pooled estimates

$$\hat{\Sigma}_k = \frac{\sum_{i=1}^n (X_i - \hat{\mu}_k)(X_i - \hat{\mu}_k)^t \, \mathbb{I}(Y_i = k)}{\sum_{j=1}^n \mathbb{I}(Y_j = k)}$$

$$\hat{\Sigma} = (n-2)^{-1} \sum_{k=0,1} \sum_{i=1}^n (X_i - \hat{\mu}_k)(X_i - \hat{\mu}_k)^t \mathbb{I}(Y_i = k)$$

Important: Covariance estimates $\hat{\Sigma}_k$, $\hat{\Sigma}$ are *not* invertible if $p > n$

Linear Discriminant Analysis in Practice

- Use $(x_1, y_1), \ldots, (x_n, y_n)$ to estimate parameters π_k, μ_k, Σ
- Form empirical discriminant functions $\hat{\delta}_k$ by replacing π_k, μ_k, Σ with maximum likelihood estimates $\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}$

Upshot: LDA rule $\hat{\phi}_n^{\text{LDA}}(x) = \operatorname{argmax}_k \hat{\delta}_k(x)$ can be written in the linear form

$$\hat{\phi}_n^{\text{LDA}}(x) = \begin{cases} 1 & \text{if } \langle \hat{\Sigma}^{-1} x, (\hat{\mu}_1 - \hat{\mu}_0) \rangle \geq \hat{\tau} \\ 0 & \text{otherwise} \end{cases}$$

In particular, the decision boundary is a hyperplane

Limitation: Vanilla LDA rule *not defined* if $p > n$

Quadratic Discriminant Analysis (QDA)

QDA Prediction Rule

- Use data $(x_1, y_1), \ldots, (x_n, y_n)$ to estimate π_k, μ_k, Σ_k
- Form empirical discrimination functions $\hat{\delta}_k$ from estimates $\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k$
- QDA rule is $\hat{\phi}_n^{\text{QDA}}(x) = \text{argmax}_k \, \hat{\delta}_k(x)$

QDA rule is non linear, with quadratic decision boundary

Limitation: Vanilla QDA rule *not defined* if $p > n$

Cousin of LDA

Recall: LDA rule can be written in the form

$$\hat{\phi}_n(x) \;=\; \begin{cases} 1 & \text{if } \langle\, \hat{\Sigma}^{-1}x, (\hat{\mu}_1 - \hat{\mu}_0)\,\rangle \geq \hat{\tau} \\ 0 & \text{otherwise} \end{cases}$$

If Gaussian assumption does *not* hold, one can still use the LDA-type rule

$$\hat{\phi}_n^{\text{LDA}}(x) \;=\; \begin{cases} 1 & \text{if } \langle\, \hat{\Sigma}^{-1}x, (\hat{\mu}_1 - \hat{\mu}_0)\,\rangle \geq \tilde{\tau} \\ 0 & \text{otherwise} \end{cases}$$

where the threshold $\tilde{\tau}$ selected to minimize number of missclassifications

Logistic Regression

Conditional Odds Ratio

Recall: Bayes rule for pair (X, Y) has form $\phi^*(x) = \mathbb{I}(\log O(x) \geq 0)$ where

$$O(x) \;=\; \frac{\mathbb{P}(Y=1|X=x)}{\mathbb{P}(Y=0|X=x)} \;=\; \frac{\eta(x)}{1-\eta(x)} \;\in\; [0, \infty]$$

is the *conditional odds ratio* of $Y = 1$ given $X = x$.

Basic Idea: Model $\log O(x)$ as a linear function of x.

Preliminary: Augment predictors by adding zeroth coordinate equal to 1

$$x = (1, x_1, \ldots, x_d)^t \in \mathbb{R}^{d+1}$$

Logistic Regression Model

LogReg Model: For some coefficient vector $\beta \in \mathbb{R}^{d+1}$ we have

$$\log \frac{\eta(x)}{1 - \eta(x)} \;=\; \beta_0 + \sum_{i=1}^{d} \beta_i \, x_i \;=\; \langle \beta, x \rangle$$

Note: The model can be written in the equivalent form

$$\eta(x : \beta) \;=\; \frac{e^{\langle \beta, x \rangle}}{1 + e^{\langle \beta, x \rangle}}$$

where $\eta(x : \beta)$ indicates that $\eta(x)$ depends on the vector β

Logistic Regression Model

Recall: The logistic regression model has the form

$$\log \frac{\eta(x:\beta)}{1-\eta(x:\beta)} \;=\; \beta_0 + \sum_{i=1}^{d} \beta_i\, x_i \;=\; \langle \beta, x\rangle$$

Interpretation of coefficient vector

- $\beta_0 =$ offset, baseline bias for $Y=1$ vs $Y=0$
- $\beta_i =$ effect on log odds ratio resulting from unit change in x_i
- $\beta_i = 0$: odds ratio does not depend on x_i
- $\beta_i > 0$: increasing x_i makes $Y=1$ more likely
- $\beta_i < 0$: increasing x_i makes $Y=1$ less likely

Logistic Regression in Practice

1. Data $D_n = (x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^{d+1} \times \{0, 1\}$

2. Estimate coefficient vector β by maximizing the *conditional log-likelihood*

$$\ell(\beta) \;=\; \log \mathbb{P}_\beta(Y = y_1 \,|\, X = x_1) \times \cdots \times \mathbb{P}_\beta(Y = y_n \,|\, X = x_n)$$

where

$$\mathbb{P}_\beta(Y = y \,|\, X = x) \;=\; \begin{cases} e^{\langle \beta, x \rangle}/(1 + e^{\langle \beta, x \rangle}) & \text{if } y = 1 \\ 1/(1 + e^{\langle \beta, x \rangle}) & \text{if } y = 0 \end{cases}$$

3. Given estimate $\hat{\beta}$ of β the logistic regression prediction rule is

$$\hat{\phi}_n^{\mathrm{LR}}(x) \;=\; \begin{cases} 1 & \text{if } e^{\langle \hat{\beta}, x \rangle}/(1 + e^{\langle \hat{\beta}, x \rangle}) \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Maximizing the Conditional Log-Likelihood

Fact: Note that $\ell : \mathbb{R}^{d+1} \to \mathbb{R}$ depends on D_n. For each $\beta \in \mathbb{R}^{d+1}$

- $\nabla \ell(\beta) = \sum_{i=1}^{n} x_i (\mathbb{I}(y_i = 1) - \eta(x_i : \beta))$
- $\nabla^2 \ell(\beta) < 0$ so $\nabla^2 \ell(\beta)$ invertible and $\ell(\cdot)$ is concave

Approach: Find $\hat{\beta}_n = \mathrm{argmax}_\beta \, \ell(\beta)$ by solving equation $\nabla \ell(\beta) = 0$

- Equation can't be solved in closed form, but we can find an approximate solution using Newton's method
- Use fitted $\eta(x : \hat{\beta})$ to classify unlabeled examples
- Test and interpret components of coefficient vector $\hat{\beta}$: features for which $\beta_i = 0$, features that increase or decrease the log odds ratio

Working Adults Data

Overview: Data on $n = 32,561$ working adults in the US from 1994 Census

- $X_i =$ demographic info (age, race, education, etc.) about adult i
- $Y_i = 1$ if adult i makes $\geq \$50k$ a year, $Y_i = 0$ otherwise

```
> summary(adult)
      age                workclass       education_num       marital_status          occupation
 Min.   :17.00   Government   : 4351   Min.   : 1.00   Divorced : 4443   Blue-Collar   :10062
 1st Qu.:28.00   Other/Unknown: 1857   1st Qu.: 9.00   Married  :15417   Other/Unknown: 1852
 Median :37.00   Private      :22696   Median :10.00   Separated: 1025   Professional : 4140
 Mean   :38.58   Self-Employed: 3657   Mean   :10.08   Single   :10683   Sales        : 3650
 3rd Qu.:48.00                         3rd Qu.:12.00   Widowed  :  993   Service      : 5021
 Max.   :90.00                         Max.   :16.00                     White-Collar : 7836
                 race            sex         hours_per_week      income
 Amer-Indian-Eskimo:  311   Female:10771   Min.   : 1.00   <=50K:24720
 Asian-Pac-Islander: 1039   Male  :21790   1st Qu.:40.00   >50K : 7841
 Black             : 3124                  Median :40.00
 Other             :  271                  Mean   :40.44
 White             :27816                  3rd Qu.:45.00
                                           Max.   :99.00
```

Working Adults Data

```
> m1 <- glm(income ~ ., data = adult, family = binomial('logit'))
> summary(m1)

Call:
glm(formula = income ~ ., family = binomial("logit"), data = adult)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7268  -0.5846  -0.2562  -0.0692   3.5080

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              -9.467139   0.250563 -37.783  < 2e-16 ***
age                       0.029430   0.001470  20.024  < 2e-16 ***
workclassOther/Unknown   -1.587717   0.720358  -2.204  0.02752 *
workclassPrivate          0.054364   0.047837   1.136  0.25577
workclassSelf-Employed   -0.175373   0.061803  -2.838  0.00455 **
education_num             0.318807   0.008392  37.990  < 2e-16 ***
marital_statusMarried     1.987371   0.059766  33.252  < 2e-16 ***
marital_statusSeparated  -0.135370   0.144532  -0.937  0.34896
marital_statusSingle     -0.513678   0.074089  -6.933 4.11e-12 ***
marital_statusWidowed    -0.029609   0.134118  -0.221  0.82527
occupationOther/Unknown   1.228633   0.720030   1.706  0.08794 .
occupationProfessional    0.753587   0.060190  12.520  < 2e-16 ***
occupationSales           0.515410   0.056694   9.091  < 2e-16 ***
occupationService         0.172611   0.060073   2.873  0.00406 **
occupationWhite-Collar    0.803544   0.046961  17.111  < 2e-16 ***
raceAsian-Pac-Islander    0.290622   0.222968   1.303  0.19243
raceBlack                 0.388039   0.213560   1.817  0.06922 .
raceOther                -0.228417   0.320930  -0.712  0.47663
raceWhite                 0.589683   0.204381   2.885  0.00391 **
sexMale                   0.391584   0.046322   8.453  < 2e-16 ***
hours_per_week            0.031120   0.001454  21.397  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Logistic Regression vs. LDA

Model

- Both methods assume log of $\eta(x)/(1-\eta(x))$ is a linear function of x
- Given π_0 and π_1, LDA specifies overall distribution of (X, Y)
- LogReg only specifies the conditional distribution of Y given X

Fitting

- LDA: maximize full likelihood via MLEs of unknown parameters
- LogReg: maximize conditional likelihood using Newton's method