S&DS 355 / 365 / 565

Data Mining and Machine Learning

Classification

Thursday, September 5th

Yale

Outline

- Classification tasks
- Logistic Regression
- Generative vs. discriminative
- Algorithms for fitting the models

HW1

- Homework 1 will be posted later today.
- Due September 17, at 11:59pm
- 355: Jupyter Notebook, 365: R markdown

Classification tasks

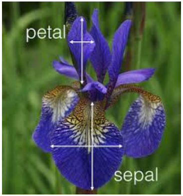
- The Iris Flower study. The data are 50 samples from each of three species of Iris flowers, *Iris setosa*, *Iris virginica* and *Iris versicolor* The length and width of the sepal and petal are measured for each specimen, and the task is to predict the species of a new Iris flower based on these features.



Iris setosa (Left), *Iris versicolor* (Middle), and *Iris virginica* (Right).

Fisher's iris classification

Iris setosa (Left), *Iris versicolor* (Middle), and *Iris virginica* (Right).



Classification tasks

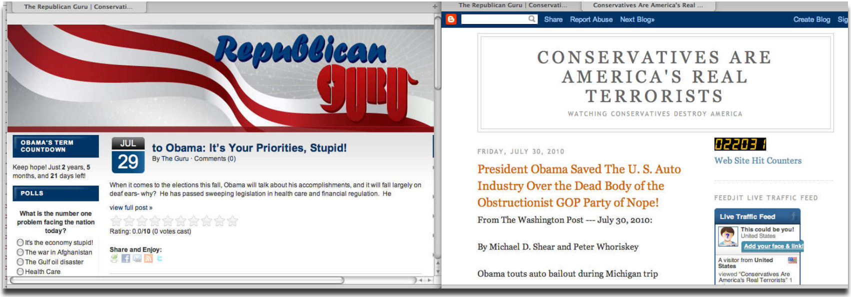
- The Coronary Risk-Factor Study (CORIS). Data: 462 males between ages of 15 and 64 from three rural areas in South Africa.

  Outcome Y is presence ($Y = 1$) or absence ($Y = 0$) of coronary heart disease

  9 covariates: systolic blood pressure, cumulative tobacco (kg), ldl (low density lipoprotein cholesterol), adiposity, famhist (family history of heart disease), typea (type-A behavior), obesity, alcohol (current alcohol consumption), and age.

Classification tasks

- Political Blog Classification. A collection of 403 political blogs were collected during two months before the 2004 presidential election. The goal is to predict whether a blog is *liberal* ($Y = 0$) or *conservative* ($Y = 1$) given the content of the blog.



New uses of text

Text as Data

Matthew Gentzkow
Stanford

Bryan T. Kelly
Chicago Booth

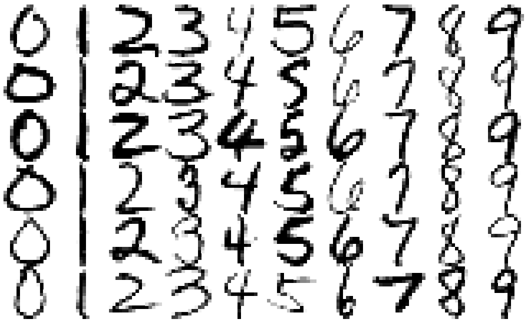
Matt Taddy
Chicago Booth and Microsoft Research

Abstract

An ever increasing share of human interaction, communication, and culture is recorded as digital text. We provide an introduction to the use of text as an input to economic research. We discuss the features that make text different from other forms of data, offer a practical overview of relevant statistical methods, and survey a variety of applications.

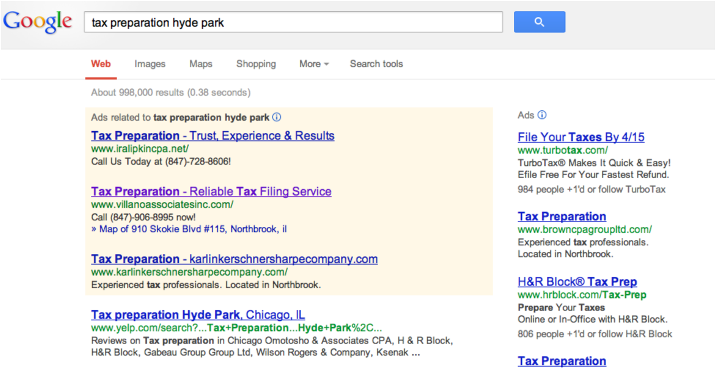
Classification tasks

- Handwriting Digit Recognition. Here each Y is one of the ten digits from 0 to 9. There are 256 covariates $X_1, \ldots, X_{256}$ corresponding to the intensity values of the pixels in a 16×16 image.

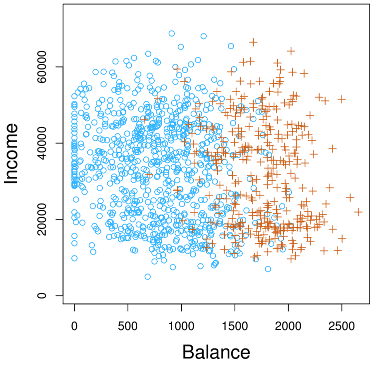


Classification tasks

- Ad click-through prediction. Predict whether or not a user will click on an ad presented. Used for ranking ads and setting prices.



Binary classifiers



Binary classifier h: function from $\mathcal{X}$ to $\{0, 1\}$

Y is 0 (blue, will not default) or 1 (brown, default)

Binary classifiers

binary classifier h: function from $\mathcal{X}$ to $\{0, 1\}$.

vector

β

x vector

Linear if exists a function $H(x) = \beta_0 + \beta^T x$ such that $h(x) = I(H(x) > 0)$.

indicator

$H(x)$ also called a *linear discriminant function*.

Decision boundary: set $\left\{x \in \mathbb{R}^d \ : \ H(x) = 0\right\}$

Bayes risk

Classification risk, or *error rate*, of h:

$$R(h) = \mathbb{P}\big(Y \neq h(X)\big)$$

and the *empirical classification error* or *training error* is

$$\widehat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} I\big(h(x_i) \neq y_i\big).$$

Optimal classification rule

Theorem. The rule h that minimizes $R(h)$ is

$$h^*(x) = \begin{cases} 1 & \text{if } m(x) > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

where $m(x) = \mathbb{E}(Y \mid X = x) = \mathbb{P}(Y = 1 \mid X = x)$ denotes the regression function.

The rule h^* is called the *Bayes rule*.

The risk $R^* = R(h^*)$ of the Bayes rule is called the *Bayes risk*.

The set $\{x \in \mathcal{X} : \ m(x) = 1/2\}$ is called the *Bayes decision boundary.*

Bayes classifier and k-nearest neighbours

The Bayes classifier cannot be utilized in practice. *Why?*

Bayes classifier and k-nearest neighbours

The Bayes classifier cannot be utilized in practice. *Why?*

k-NN provides a means to estimate $\mathbb{P}(Y = 1 \mid X = x)$.

Formally, define $\mathcal{N}_0(x)$ as the set of k observations that are closest to x in the feature space. Then we can approximate $\mathbb{P}(Y = 1 \mid X = x)$ using

$$\frac{1}{k} \sum_{i \in \mathcal{N}_0(x)} y_i$$

k-NN classification uses

$$h(x) = \begin{cases} 1 & \text{if } \frac{1}{k} \sum_{i \in \mathcal{N}_0(x)} y_i > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

(aka Majority voting)

The Bayes rule

From Bayes' theorem

$$\begin{aligned}\mathbb{P}(Y=1 \mid X=x) &= \frac{p(x \mid Y=1)\mathbb{P}(Y=1)}{p(x \mid Y=1)\mathbb{P}(Y=1)+p(x \mid Y=0)\mathbb{P}(Y=0)} \\ &= \frac{\pi_1 p_1(x)}{\pi_1 p_1(x)+(1-\pi_1)p_0(x)}.\end{aligned}$$

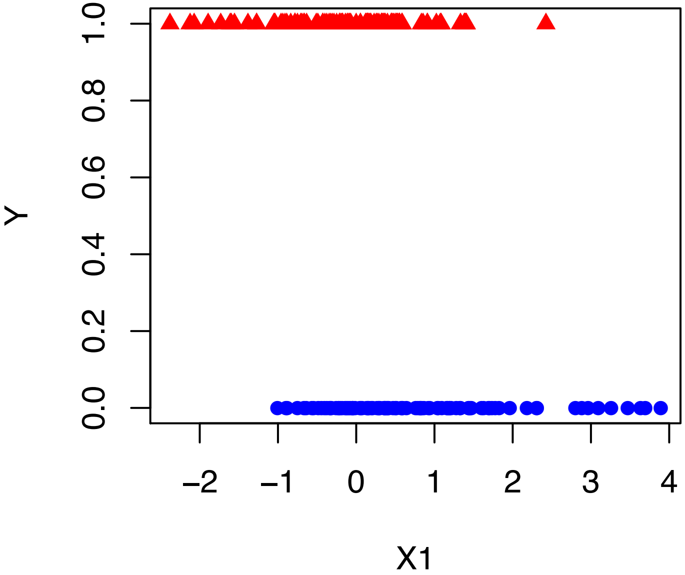
where $\pi_1 = \mathbb{P}(Y=1)$. So,

$$m(x) > \frac{1}{2} \quad \text{is equivalent to} \quad \frac{p_1(x)}{p_0(x)} > \frac{1-\pi_1}{\pi_1}.$$

Thus the Bayes rule can be rewritten as

$$h^*(x) = \begin{cases} 1 & \text{if } \dfrac{p_1(x)}{p_0(x)} > \dfrac{1-\pi_1}{\pi_1} \\ 0 & \text{otherwise.} \end{cases}$$

Simulated Data–one predictor



Logistic regression

Conditional probabilities of the class:

$$P(Y_i = 1|X = x_i) = p(x_i)$$

$$P(Y_i = 0|X = x_i) = 1 - p(x_i)$$

Logistic regression

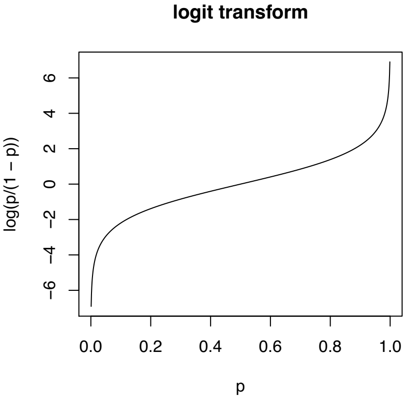
Conditional probabilities of the class:

$$P(Y_i = 1|X = x_i) = p(x_i)$$

$$P(Y_i = 0|X = x_i) = 1 - p(x_i)$$

We model the relationship between $p(x_i)$ and x_i.

Logistic regression



The *logit* transform:

$$logit(p) = \log\left(\frac{p}{1-p}\right)$$

The logit transform

- is monotone
- maps the interval $[0, 1]$ to $(-\infty, \infty)$

Logistic regression

Logistic regression is a linear regression model of the log odds:

$$logit(\widehat{p}) = X\widehat{\beta}$$

- p is a probability.
- $\frac{p}{1-p}$ is odds.
- $logit(p) = \log\left(\frac{p}{1-p}\right)$ is (natural) log odds.

Logistic regression

Logistic regression is a linear regression model of the log odds:

$$logit(\widehat{p}) = X\widehat{\beta}$$

- p is a probability.
- $\frac{p}{1-p}$ is odds.
- $logit(p) = \log\left(\frac{p}{1-p}\right)$ is (natural) log odds.

Equivalent formulation:

$$\widehat{p} = \frac{e^{X\widehat{\beta}}}{1 + e^{X\widehat{\beta}}} = logistic(X\widehat{\beta})$$

LR decision boundary is linear

- When $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$, $\frac{\widehat{p}}{1-\widehat{p}} = 1$, so $\widehat{p} = 0.5$.
- If our goal is to minimize the overall training error rate, then we use the rule:

$$\widehat{y} = \begin{cases} 1 & \widehat{p} \geq 0.5 \\ 0 & \widehat{p} < 0.5 \end{cases}$$

LR decision boundary is linear

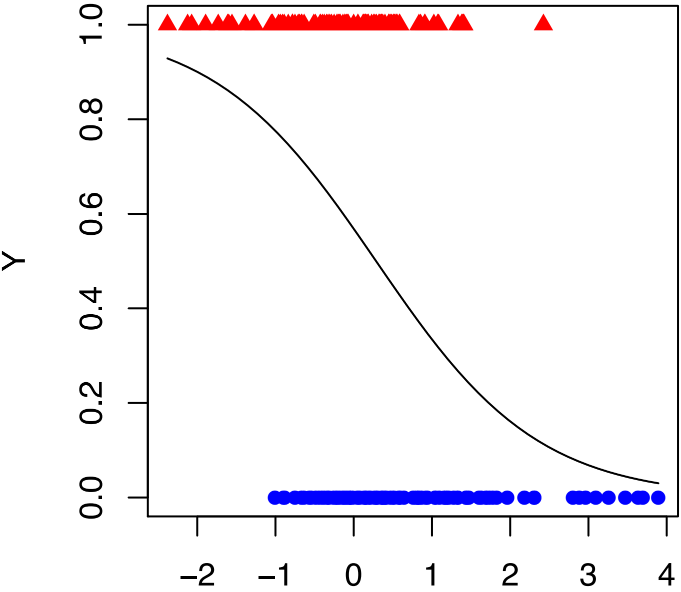
- When $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$, $\frac{\widehat{p}}{1-\widehat{p}} = 1$, so $\widehat{p} = 0.5$.
- If our goal is to minimize the overall training error rate, then we use the rule:

$$\widehat{y} = \begin{cases} 1 & \widehat{p} \geq 0.5 \\ 0 & \widehat{p} < 0.5 \end{cases}$$

- Hence, the decision boundary is given by $\{x : x^T \widehat{\beta} = 0\}$.

The decision boundary is linear in x!

Simulated data



Interpreting the coefficients

```
##
## Call:
## glm(formula = Y ~ X1, family = binomial, data = g.r.data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.73067  -1.09281   0.06873   1.04226   2.08824
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2754     0.1614   1.706    0.088 .
## X1           -0.9633     0.1907  -5.052 4.38e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 277.26  on 199  degrees of freedom
## Residual deviance: 241.17  on 198  degrees of freedom
## AIC: 245.17
##
## Number of Fisher Scoring iterations: 4
```

Interpreting the coefficients

For this example,

$$logit(\widehat{p}) = \widehat{\beta}_0 + \widehat{\beta}_1 x = 0.28 - 0.96x$$

A one unit increase in x_1 is associated with:

- a decrease of 0.96 in the log-odds of $y = 1$
- a decrease in odds of $y = 1$ by a factor of $e^{-0.96} = 0.38$.

Interpreting the coefficients

For this example,

$$logit(\widehat{p}) = \widehat{\beta}_0 + \widehat{\beta}_1 x = 0.28 - 0.96x$$

A one unit increase in x_1 is associated with:

- a decrease of 0.96 in the log-odds of $y = 1$
- a decrease in odds of $y = 1$ by a factor of $e^{-0.96} = 0.38$.

$$\widehat{p} = \frac{e^{0.28 - 0.96x_1}}{1 + e^{0.28 - 0.96x_1}}$$

Interpreting the coefficients

For this example,

$$logit(\widehat{p}) = \widehat{\beta}_0 + \widehat{\beta}_1 x = 0.28 - 0.96x$$

A one unit increase in x_1 is associated with:

- a decrease of 0.96 in the log-odds of $y = 1$
- a decrease in odds of $y = 1$ by a factor of $e^{-0.96} = 0.38$.

$$\widehat{p} = \frac{e^{0.28-0.96x_1}}{1 + e^{0.28-0.96x_1}}$$

The decision boundary is given by

$$x = \frac{0.28}{0.96} = 0.29$$

Fitting a logistic regression

Traditionally, use maximum likelihood estimation (MLE).

- Likelihood of a single observation (x_i, y_i):

$$L_i(\beta) = p_i^{y_i} \cdot (1 - p_i)^{1-y_i} = \left(\frac{1}{1 + e^{-x_i^T\beta}}\right)^{y_i} \cdot \left(1 - \frac{1}{1 + e^{-x_i^T\beta}}\right)^{1-y_i}$$

Fitting a logistic regression

Traditionally, use maximum likelihood estimation (MLE).

- Likelihood of a single observation (x_i, y_i):

$$L_i(\beta) = p_i^{y_i} \cdot (1 - p_i)^{1-y_i} = \left(\frac{1}{1 + e^{-x_i^T\beta}}\right)^{y_i} \cdot \left(1 - \frac{1}{1 + e^{-x_i^T\beta}}\right)^{1-y_i}$$

- Log-likelihood of a single observation:

$$\begin{aligned}\ell_i(\beta) &= -y_i \log(1 + e^{-x_i^T\beta}) + (1 - y_i) \log\left(\frac{e^{-x_i^T\beta}}{1 + e^{-x_i^T\beta}}\right) \\ &= -\log(1 + e^{-x_i^T\beta}) - y_i \log e^{-x_i^T\beta} \\ &= -\log(1 + e^{-x_i^T\beta}) + y_i x_i^T \beta\end{aligned}$$

This slide has an error

Fitting a logistic regression

Traditionally, use maximum likelihood estimation (MLE).

- Likelihood of a single observation (x_i, y_i):

$$L_i(\beta) = p_i^{y_i} \cdot (1 - p_i)^{1-y_i} = \left(\frac{1}{1 + e^{-x_i^T\beta}}\right)^{y_i} \cdot \left(1 - \frac{1}{1 + e^{-x_i^T\beta}}\right)^{1-y_i}$$

- Log-likelihood of a single observation:

$$\begin{aligned} \ell_i(\beta) &= -y_i \log(1 + e^{-x_i^T\beta}) + (1 - y_i) \log\left(\frac{e^{-x_i^T\beta}}{1 + e^{-x_i^T\beta}}\right) \\ &= -\log(1 + e^{-x_i^T\beta}) - y_i \log e^{-x_i^T\beta} \\ &= -\log(1 + e^{-x_i^T\beta}) + y_i x_i^T\beta \end{aligned}$$

This slide has an error

- Aggregate log-likelihood:

$$\ell(\beta) = \sum \left(y_i x_i^T\beta - \log(1 + e^{-x_i^T\beta})\right)$$

Extension to more than 2 classes

Multinomial logistic regression extends the logistic regression model to $K \geq 2$ classes.

$$\log\left(\frac{P(Y = k \mid X = x)}{P(Y = 0|X = x)}\right) = x^T\beta_k, \qquad k = 1, 2, \ldots, K - 1$$

Extension to more than 2 classes

Multinomial logistic regression extends the logistic regression model to $K \geq 2$ classes.

$$\log\left(\frac{P(Y = k \mid X = x)}{P(Y = 0 | X = x)}\right) = x^T \beta_k, \qquad k = 1, 2, \ldots, K - 1$$

$$P(Y = k \mid X = x) = \frac{\exp(x^T \beta_k)}{1 + \sum_{l=1}^{K-1} \exp(x^T \beta_l)}, \qquad k = 1, 2, \ldots, K - 1$$

Separable classes

Another point of consideration related to the likelihood...

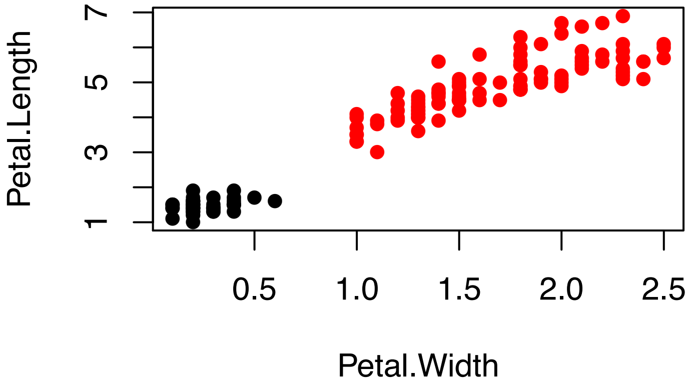
Recall that logistic regression fits a linear decision boundary:

$$\left\{x : x^T\widehat{\beta} = 0\right\}$$

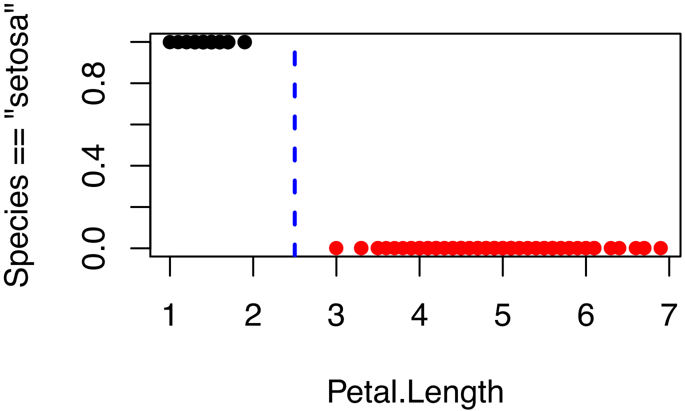
What happens when the two classes are *separable* (*i.e.*, a hyperplane can perfectly separate out the two classes)?

Separable classes

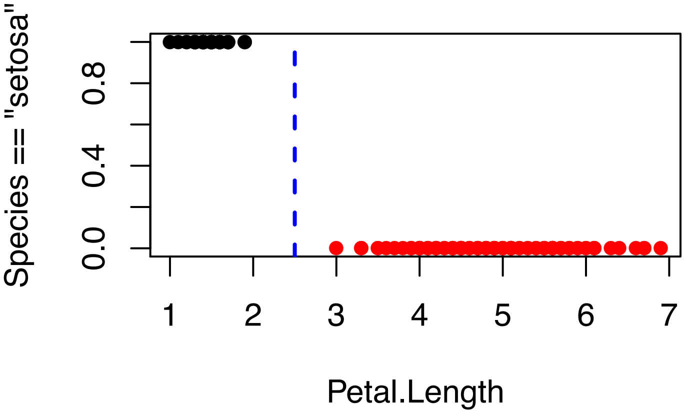
Pretend we only care for predicting setosas ($Y = 1$) vs. non-setosas ($Y = 0$):



Separable classes



Separable classes



Petal length of 2.5 can perfectly separate $Y = 1$ and $Y = 0$ groups.

Separable classes

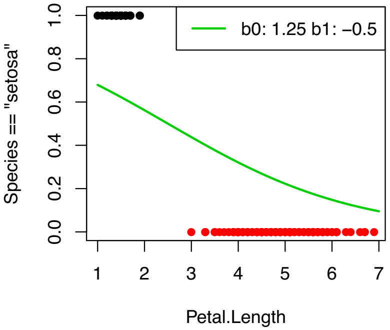
Decision boundary: $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$.

$\widehat{\beta}_1 = -\frac{\widehat{\beta}_0}{2.5}$ for $\widehat{\beta}_1 < 0$ will yield perfect fits.

Separable classes

Decision boundary: $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$.

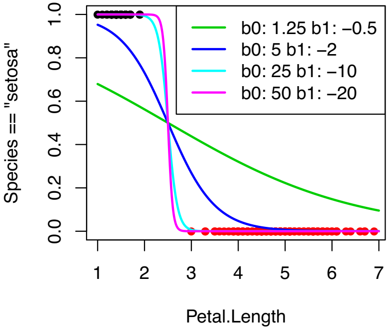
$\widehat{\beta}_1 = -\frac{\widehat{\beta}_0}{2.5}$ for $\widehat{\beta}_1 < 0$ will yield perfect fits.



Separable classes

Decision boundary: $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$.

$\widehat{\beta}_1 = -\frac{\widehat{\beta}_0}{2.5}$ for $\widehat{\beta}_1 < 0$ will yield perfect fits.

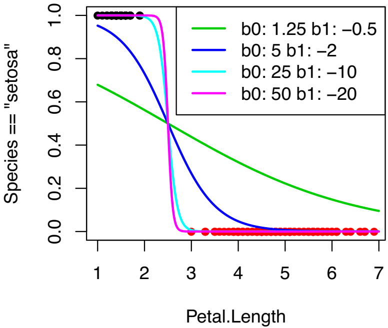


Int	Slope	Likelihood
1.25	-0.5	0.0000000
5.00	-2.0	0.0001696
25.00	-10.0	0.9846004
50.00	-20.0	0.9999415

Separable classes

Decision boundary: $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$.

$\widehat{\beta}_1 = -\frac{\widehat{\beta}_0}{2.5}$ for $\widehat{\beta}_1 < 0$ will yield perfect fits.



Int	Slope	Likelihood
1.25	-0.5	0.0000000
5.00	-2.0	0.0001696
25.00	-10.0	0.9846004
50.00	-20.0	0.9999415

As $\|\beta\|$ increases, likelihood approaches 1.

Separable classes

```
## Warning:  glm.fit:  algorithm did not converge

## Warning:  glm.fit:  fitted probabilities numerically 0 or 1
occurred

##
## Call:
## glm(formula = Species == "setosa" ~ Petal.Length, family = bin
##     data = iris)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -6.429e-05  -2.100e-08  -2.100e-08   2.100e-08   3.997e-05
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)     91.67   47334.35   0.002    0.998
## Petal.Length   -37.22   18357.58  -0.002    0.998
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

Separable classes

Problematic?

- Appears that predictor is not informative, but it is!
- Theoretically we obtained a perfect fit on the training data.

Separable classes

Problematic?

- Appears that predictor is not informative, but it is!
- Theoretically we obtained a perfect fit on the training data.
  - Overfitting is possible. Regularization can help.

Iris species

```
## Call:
## multinom(formula = Species ~ Petal.Length + Petal.Width, data
##     trace = FALSE)
##
## Coefficients:
##            (Intercept) Petal.Length Petal.Width
## versicolor   -22.79944      6.92122    7.878496
## virginica    -67.82521     12.64721   18.261016
##
## Std. Errors:
##            (Intercept) Petal.Length Petal.Width
## versicolor     44.3859     37.58715    81.00888
## virginica      46.3939     37.65702    81.09482
##
## Residual Deviance: 20.57901
## AIC: 32.57901
```

Two flavors of classifiers

Generative models model both the input X and the output Y.

Discriminative models model only the output Y given X.

Logistic regression is discriminative because we're only given X and figure out the Y.

Two flavors of classifiers

Generative models model both the input X and the output Y.

Discriminative models model only the output Y given X.

Which one is logistic regression? Which do you think is better?

Generative models

$$p(x_i, y_i) = p(x_i \mid y_i)p(y_i) = p(y_i|x_i)p(x_i).$$

In the generative case we typically estimate the joint distribution by maximizing the *joint likelihood*:

$$\prod_{i=1}^{n} p(x_i, y_i) = \underbrace{\prod_{i=1}^{n} p(x_i \mid y_i)}_{\text{parametric model}} \underbrace{\prod_{i=1}^{n} p(y_i)}_{\text{Bernoulli}}.$$

Bern (p) $p = \frac{1}{2}$

like coin flip

Discriminative models

$p(x_i, y_i) = p(x_i \mid y_i)p(y_i) = p(y_i|x_i)p(x_i)$.

discriminative

In the ~~generative~~ case we typically estimate the joint distribution by maximizing the *conditional likelihood*:

there is no probability distribution over x in discriminative models

$$\prod_{i=1}^{n} p(x_i, y_i) = \underbrace{\prod_{i=1}^{n} p(y_i \mid x_i)}_{\text{parametric model}} \underbrace{\prod_{i=1}^{n} p(x_i)}_{\text{ignored}}.$$

Generative Models

We parametrize conditional densities: first start with y and generate x.

- $p_{\theta_0,0}(x) = p_{\theta_0}(x \mid Y = 0)$
- $p_{\theta_1,1}(x) = p_{\theta_1}(x \mid Y = 1)$

In this case,

$$m_\theta(x) \equiv \mathbb{P}(Y = 1 \mid X = x) = \frac{\pi_1 p_{\theta_1,1}(x)}{(1-\pi_1)p_{\theta_0,0}(x) + \pi_1 p_{\theta_1,1}(x)}.$$

What's the purpose of generative model? Since we generate data ourselves

- more in Language processing

Generative Models

We parametrize conditional densities:

- $p_{\theta_0,0}(x) = p_{\theta_0}(x \mid Y = 0)$
- $p_{\theta_1,1}(x) = p_{\theta_1}(x \mid Y = 1)$

$\pi_1 \equiv \mathbb{P}(y=1)$

Bayes' Rule

In this case,

$$m_\theta(x) \equiv \mathbb{P}(Y = 1 \mid X = x) = \frac{\pi_1 p_{\theta_1,1}(x)}{(1 - \pi_1)p_{\theta_0,0}(x) + \pi_1 p_{\theta_1,1}(x)}.$$

Given an estimator $(\widehat{\theta}_n, \widehat{\pi}_1)$, define *plug-in estimator*

$$\widehat{h}(x) = I(m_{\widehat{\theta}_n}(x) > 1/2).$$

indicator function

$$I = \begin{cases} 1 & \text{if } x \text{ true} \\ 0 & \text{if } \neg x \end{cases}$$

Often, $\widehat{\theta}_n$ is the maximum likelihood estimator.

Gaussian discriminant analysis

355 skip

Suppose that $p_0(x) = p(x \mid Y = 0)$ and $p_1(x) = p(x \mid Y = 1)$ are multivariate Gaussian:

$$p_k(x) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)\right\}, \quad k = 0, 1.$$

where Σ_1 and Σ_2 are $d \times d$ covariance matrices:
$X \mid Y = 0 \sim N(\mu_0, \Sigma_0)$ and $X \mid Y = 1 \sim N(\mu_1, \Sigma_1)$.

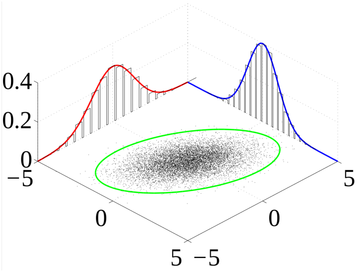
Gaussian discriminant analysis

Skip

Suppose that $p_0(x) = p(x \mid Y = 0)$ and $p_1(x) = p(x \mid Y = 1)$ are multivariate Gaussian:

$$p_k(x) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right\}, \quad k = 0, 1.$$

where Σ_1 and Σ_2 are $d \times d$ covariance matrices:
$X \mid Y = 0 \sim N(\mu_0, \Sigma_0)$ and $X \mid Y = 1 \sim N(\mu_1, \Sigma_1)$.



Gaussian discriminant analysis

Suppose that $p_0(x) = p(x \mid Y = 0)$ and $p_1(x) = p(x \mid Y = 1)$ are multivariate Gaussian:

$$p_k(x) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right\}, \quad k = 0, 1.$$

where Σ_1 and Σ_2 are $d \times d$ covariance matrices:
$X \mid Y = 0 \sim N(\mu_0, \Sigma_0)$ and $X \mid Y = 1 \sim N(\mu_1, \Sigma_1)$.

Calculation: If $X \mid Y = 0 \sim N(\mu_0, \Sigma_0)$ and $X \mid Y = 1 \sim N(\mu_1, \Sigma_1)$, then the Bayes rule is

$$h^*(x) = \begin{cases} 1 & \text{if } r_1^2 < r_0^2 + 2\log\left(\frac{\pi_1}{1-\pi_1}\right) + \log\left(\frac{|\Sigma_0|}{|\Sigma_1|}\right) \\ 0 & \text{otherwise} \end{cases}$$

where $r_i^2 = (x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)$ for $i = 1, 2$.

Quadratic discriminant analysis

An equivalent way of expressing the Bayes rule is

$$h^*(x) = \mathrm{argmax}_{k \in \{0,1\}} \delta_k(x)$$

where

$$\delta_k(x) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) + \log \pi_k$$

is called the *Gaussian discriminant function*.

The decision boundary is $\{x \in \mathcal{X} : \delta_1(x) = \delta_0(x)\}$, which is quadratic.

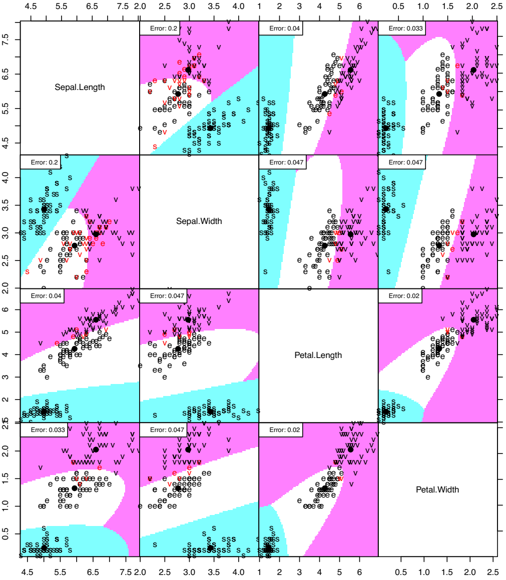
Quadratic discriminant analysis

To estimate this we use sample quantities of $\pi_0, \pi_1, \mu_1, \mu_2, \Sigma_0, \Sigma_1$

$$\begin{aligned}
\widehat{\pi}_0 &= \frac{1}{n}\sum_{i=1}^{n}(1-y_i), \;\; \widehat{\pi}_1 = \frac{1}{n}\sum_{i=1}^{n} y_i, \\
\widehat{\mu}_0 &= \frac{1}{n_0}\sum_{i:\ y_i=0} x_i, \;\; \widehat{\mu}_1 = \frac{1}{n_1}\sum_{i:\ y_i=1} x_i, \\
\widehat{\Sigma}_0 &= \frac{1}{n_0-1}\sum_{i:\ y_i=0}(x_i-\widehat{\mu}_0)(x_i-\widehat{\mu}_0)^T, \\
\widehat{\Sigma}_1 &= \frac{1}{n_1-1}\sum_{i:\ y_i=1}(x_i-\widehat{\mu}_1)(x_i-\widehat{\mu}_1)^T,
\end{aligned}$$

where $n_0 = \sum_i (1-y_i)$ and $n_1 = \sum_i y_i$.

Quadratic discriminant analysis: Iris data



Linear discriminant analysis

Suppose $\Sigma_0 = \Sigma_1 = \Sigma$. Bayes rule becomes $h^*(x) = \mathrm{argmax}_k \delta_k(x)$ with

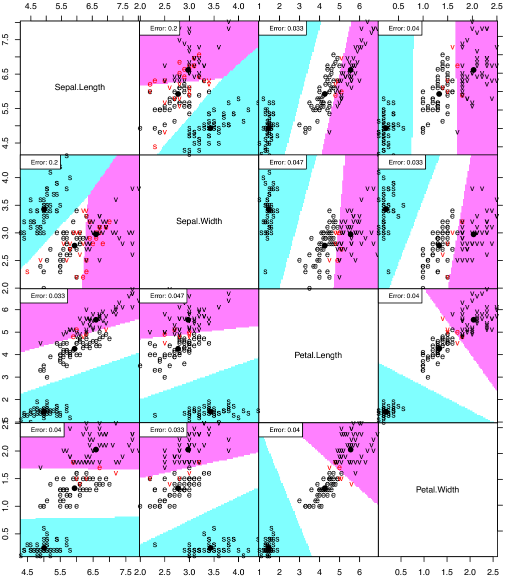
$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k.$$

Use a pooled estimate of the Σ:

$$\widehat{\Sigma} = \frac{(n_0 - 1)\widehat{\Sigma}_0 + (n_1 - 1)\widehat{\Sigma}_1}{n_0 + n_1 - 2}.$$

The decision boundary is now linear.

Linear discriminant analysis: Iris data



Logistic regression

If Y takes values 0 and 1, we say that Y has a Bernoulli distribution with parameter $\pi_1 = \mathbb{P}(Y = 1)$.

The probability mass function for Y is $p(y; \pi_1) = \pi_1^y (1 - \pi_1)^{1-y}$ for $y = 0, 1$.

The likelihood function for π_1 based on iid data $y_1, \ldots, y_n$ is

$$\mathcal{L}(\pi_1) = \prod_{i=1}^{n} p(y_i; \pi_1) = \prod_{i=1}^{n} \pi_1^{y_i} (1 - \pi_1)^{1-y_i}.$$

Logistic regression

In the logistic regression model,

$$m(x) = \mathbb{P}(Y = 1 \mid X = x) = \frac{\exp\big(\beta_0 + x^T\beta\big)}{1 + \exp\big(\beta_0 + x^T\beta\big)} \equiv \pi_1(x, \beta_0, \beta).$$

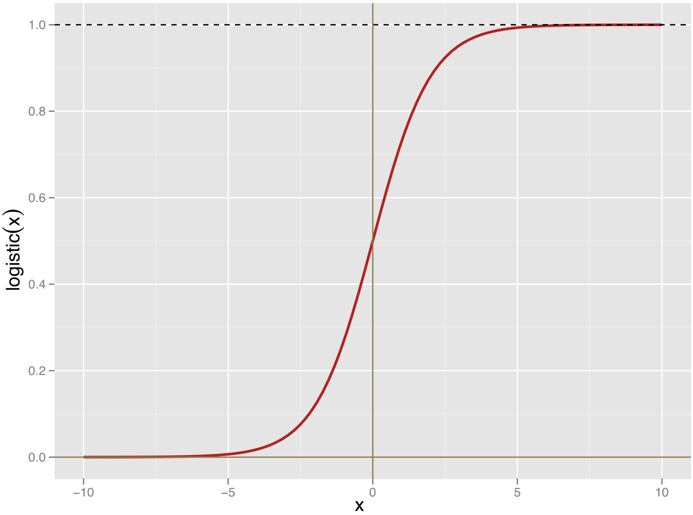
So, given $X = x$, Y is Bernoulli with mean $\pi_1(x, \beta_0, \beta)$. Can write as

$$\mathrm{logit}\big(\mathbb{P}(Y = 1 \mid X = x)\big) = \beta_0 + x^T\beta$$

where $\mathrm{logit}(a) = \log(a/(1-a))$.

The name “logistic regression” comes from the fact that $\exp(x)/(1 + \exp(x))$ is called the logistic function.

Logistic function



This is an example of a generative versus a discriminative model.

In Gaussian LDA we estimate the whole joint distribution by maximizing the full likelihood

$$\prod_{i=1}^{n} p(x_i, y_i) = \underbrace{\prod_{i=1}^{n} p(x_i \mid y_i)}_{\text{Gaussian}} \underbrace{\prod_{i=1}^{n} p(y_i)}_{\text{Bernoulli}}.$$

In logistic regression we maximize the conditional likelihood $\prod_{i=1}^{n} p(y_i \mid X_i)$ but ignore the second term $p(x_i)$:

$$\prod_{i=1}^{n} p(x_i, y_i) = \underbrace{\prod_{i=1}^{n} p(y_i \mid x_i)}_{\text{logistic}} \underbrace{\prod_{i=1}^{n} p(x_i)}_{\text{ignored}}.$$

Fitting a logistic regression model

- We maximize conditional likelihood. There is no closed form.
- Need to iterate.
- Standard approach is equivalent to Newton's algorithm
  - Make a quadratic approximation
  - Do a weighted least squares regression
  - Repeat

Newton's method

To find a zero of $f(x)$:

$$x \longleftarrow x - \frac{f(x)}{f'(x)}$$

To find a maximum of $f(x)$:

$$x \longleftarrow x - H(f,x)^{-1} \nabla f(x)$$

where ∇f is the (gradient) vector of first, derivatives, and H is the (Hessian) matrix of second derivatives

Iteratively reweighted least squares

Given the current estimate $\widehat{\beta}$, Newton's algorithm forms a quadratic approximation to the log-likelihood:

$$-\ell(\beta) = \frac{1}{2}(z - X\beta)^T W(z - X\beta) + \text{constant}.$$

where

$$z_i = \log\left(\frac{\pi_1(x_i)}{1 - \pi_1(x_i)}\right) + \frac{y_i - \pi_1(x_i)}{\pi_1(x_i)(1 - \pi_1(x_i))}.$$

is a "synthetic" response.

W is a diagonal weight matrix, with weight on the ith point given by

$$w_i = \pi_1(x_i)(1 - \pi_1(x_i))$$

This is a weighted least squares problem.

Big models

- Where else is logistic regression used?
- The Internet search players (Facebook, Microsoft, Google, ...) build ginormous logistic regressions.
- The models may have hundreds of thousands of features (covariates, d) and hundreds of millions of samples (data, n)
- We'll talk later about techniques for fitting such big models

What did we learn today?

- Classifiers come in two flavors: generative & discriminitive.
- Linear Gaussian discriminant analysis is a simple generative classifier.
- Logistic regression is the discriminative version. Default method.
- Can be fit with iterative, weighted least squared regression.

Readings

Classification is covered in Chapter 4 of our ISL book. In particular, Section 4.3 is on logistic regression, and Section 4.4 is on linear and quadratic discriminant analysis.