Statistical Machine Learning

Lecture 3: Linear Methods for Classification

W.Q.Cui Research Group

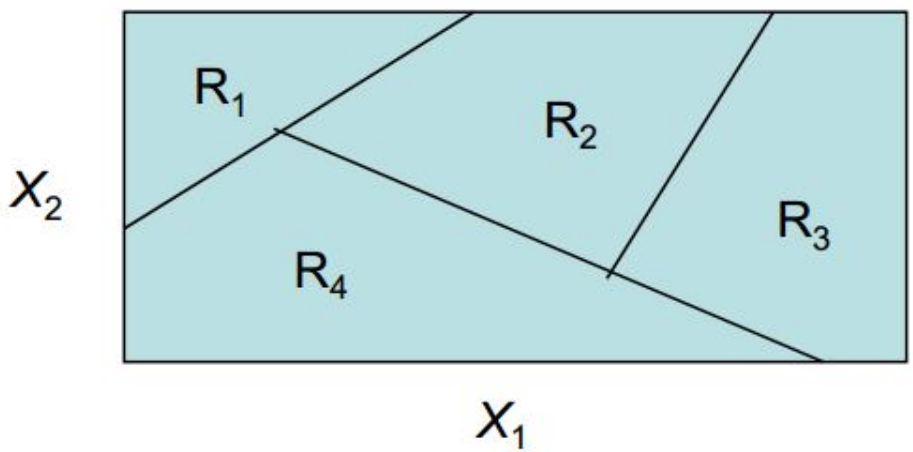
Department of Statistics and Finance
University of Science and Technology of China

2018 Autumn

Contents

Linear Classification

- What is meant by linear classification?
  — The decision boundaries in the in the feature (input) space is linear
- Should the regions be contiguous?



Piecewise linear decision boundaries in 2D input space

Linear Classification

- There is a discriminant function $\delta_k(x)$ for each class k
- Classification rule: $R_k = \{x : k = argmax\ \delta_j(x)\}$
- In higher dimensional space the decision boundaries are piecewise hyperplane
- Remember that 0-1 loss function led to the classification rule: $R_k = \{x : k = \text{argmax}\ Pr(G = j|X = x)\}$
- So, $Pr(G = k|X)$ can serve as $\delta_k(x)$

Linear Classification

- All we require here is the class boundaries $x : \delta_k(x) = \delta_j(x)$ be linear for every (k, j) pair
- One can achieve this if $\delta_k(x)$ themselves are linear or any monotone transform of $\delta_k(x)$ is linear
  – An example:

$$P(G = 1|X = x) = \frac{\exp(\beta_0 + \beta^T x)}{1 + \exp(\beta_0 + \beta^T x)}$$

$$P(G = 2|X = x) = \frac{1}{1 + \exp(\beta_0 + \beta^T x)}$$

So that $\ln \left[\frac{P(G=1|X=x)}{P(G=2|X=x)} \right] = \beta_0 + \beta^T x$ is linear.

Linear Classification as a Linear Regression

2D Input space: $X = (X_1, X_2)$
Number of classes/categories $K = 3$,so output $Y = (Y_1, Y_2, Y_3)$
Training sample, size N $= 5$,

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ 1 & x_{31} & x_{32} \\ 1 & x_{41} & x_{42} \\ 1 & x_{51} & x_{52} \end{pmatrix}, Y = \begin{pmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \\ y_{41} & y_{42} & y_{43} \\ y_{51} & y_{52} & y_{53} \end{pmatrix}$$

Each row of Y has exactly one 1 indicating the category/class.

Linear Classification as a Linear Regression

Regression output:

$$\hat{Y}((x_1, x_2)) = (1\ x_1\ x_2)(X^T X)^{-1} X^T Y = (x^T \beta_1\ x^T \beta_2\ x^T \beta_3)$$

Or,

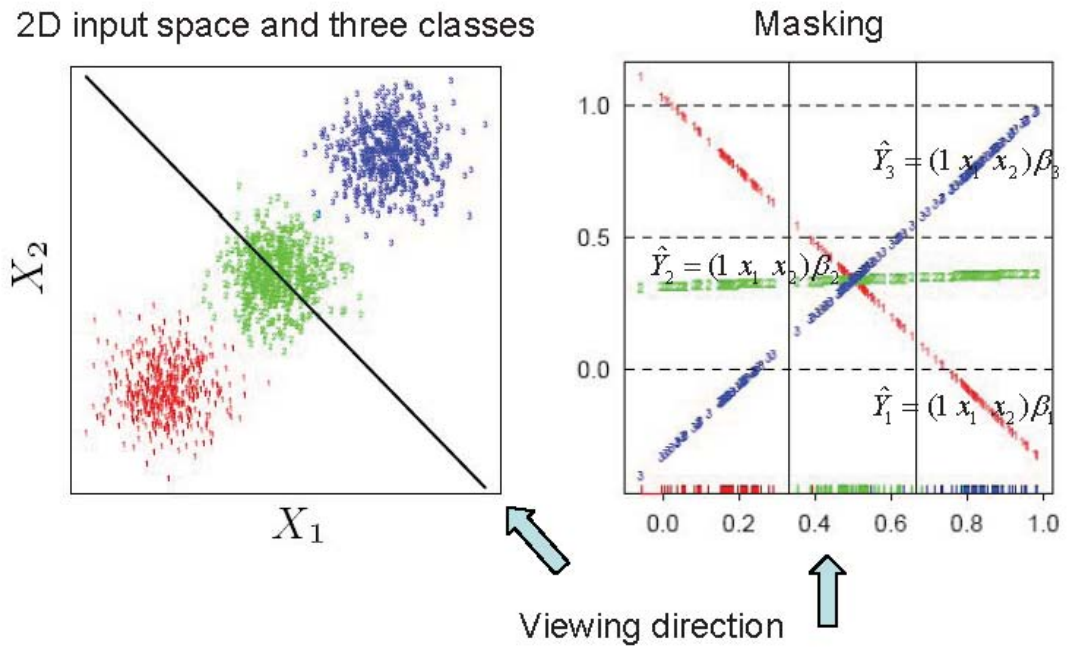
$$\hat{Y}_1((x_1, x_2)) = (1\ x_1\ x_2)\beta_1$$
$$\hat{Y}_2((x_1, x_2)) = (1\ x_1\ x_2)\beta_2$$
$$\hat{Y}_3((x_1, x_2)) = (1\ x_1\ x_2)\beta_3$$

Classification rule:

$$\hat{G}((x_1, x_2)) = \text{argmax}_k \hat{Y}_k((x_1, x_2))$$

The Masking

Linear regression of the indicator matrix can lead to masking



LDA can avoid this maskings

Linear Discriminant Analysis

Essentially minimum error Bayes' classifier

Assumes that the conditional class densities are (multivariate) Gaussian

Assumes equal covariance for every class

Posterior probability

$$Pr(G = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{\ell=1}^{K} f_\ell(x)\pi_\ell}$$

where π_k is the prior probability for class k, $f_k(x)$ is class conditional density or likelihood density

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k))$$

LDA

$$\ln \frac{Pr(G=k|X=x)}{Pr(G=\ell|X=x)}$$

$$= \ln \frac{\pi_k}{\pi_\ell} + \ln \frac{f_k}{f_\ell}$$

$$= (\ln \pi_k + x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k) - (\ln \pi_\ell + x^T\Sigma^{-1}\mu_\ell - \frac{1}{2}\mu_\ell^T\Sigma^{-1}\mu_\ell)$$

where

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

Classification rule:

$$\hat{G}(x) = \text{argmax}_k \delta_k(x)$$

is equivalent to

$$\hat{G}(x) = \text{argmax}_k Pr(G=k|X=x)$$

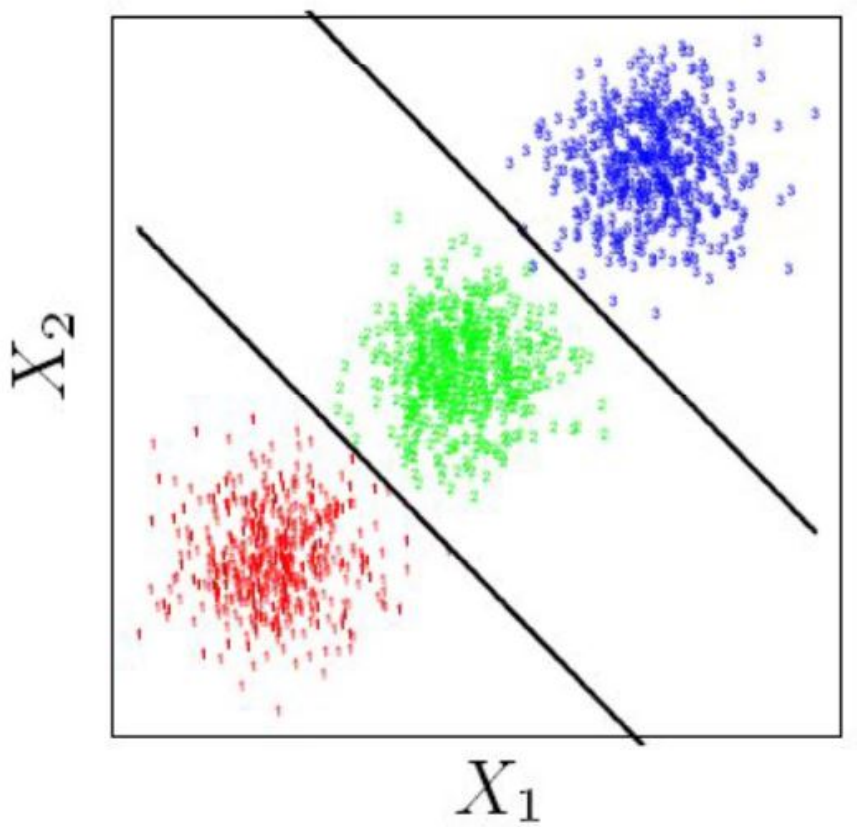
The good old Bayes classifier!

LDA

When are we going to use the training data?
Total N input-output pairs: $(g_i, x_i), i = 1, \cdots, N$
number of pairs in class k: N_k
Total number of classes: K
Training data utilized to estimate

- Prior Probabilities: $\hat{\pi}_k = N_k/N$
- Means: $\hat{\mu}_k = \sum_{g_i=k} x_i/N_k$
- Covariance Matrix: $\hat{\Sigma} = \sum_{k=1}^{K} \sum_{g_i} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T/(N - K)$

LDA Example



LDA was able to avoid masking here

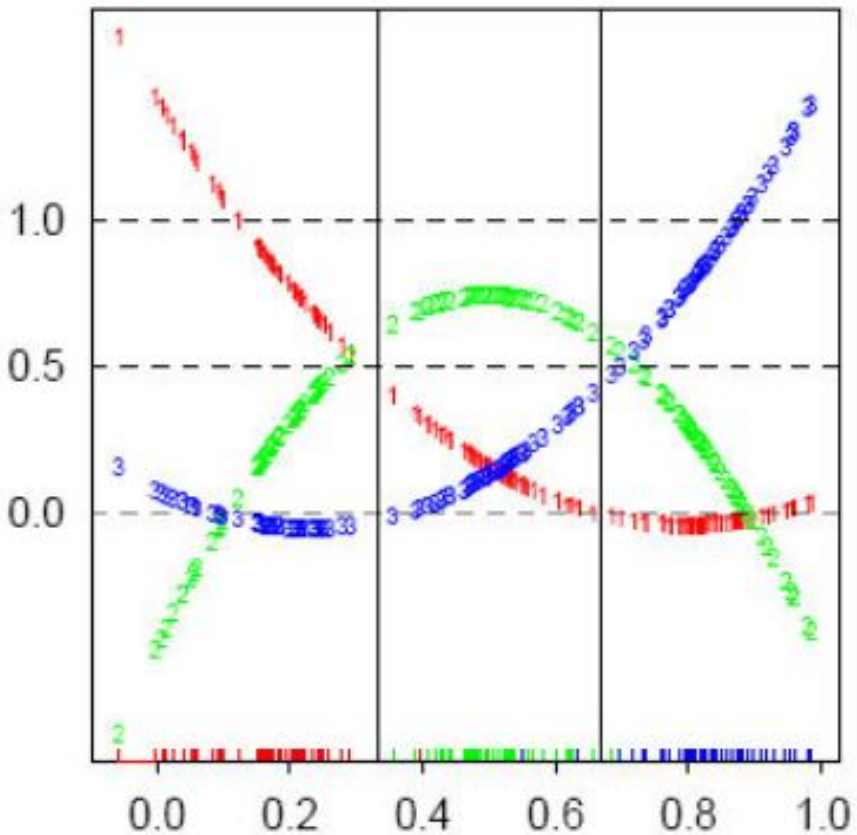
Quadratic Discriminant Analysis

- Relaxes the same covariance assumption - class conditional probability densities (still multivariate Gaussians) are allowed to have **different** covariant matrices
- The class decision boundaries are not linear rather **quadratic**

$$
\begin{aligned}
&\ln \frac{Pr(G=k|X=x)}{Pr(G=\ell|X=x)} \\
&= \ln \frac{\pi_k}{\pi_\ell} + \ln \frac{f_k}{f_\ell} \\
&= (\ln \pi_k - \frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k) - \frac{1}{2}\ln|\Sigma_k|) \\
&\quad - (\ln \pi_\ell - \frac{1}{2}(x-\mu_\ell)^T \Sigma_\ell^{-1}(x-\mu_\ell) - \frac{1}{2}\ln|\Sigma_\ell|)
\end{aligned}
$$

QDA and Masking

Better than Linear Regression in terms of handling masking:



Usually computationally more expensive than LDA

Fisher's Linear Discriminant [DHS]

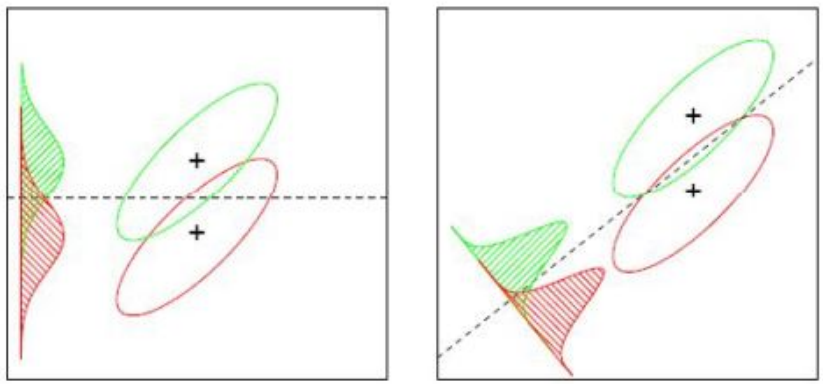
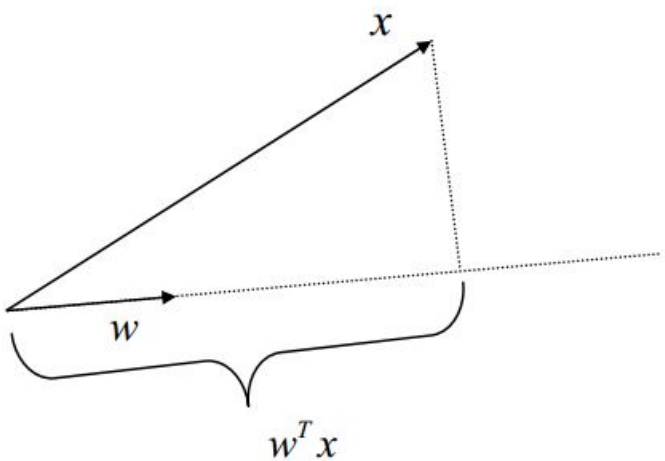


Figure 4.9: *Although the line joining the centroids defines the direction of greatest centroid spread, the projected data overlap because of the covariance (left panel). The discriminant direction minimizes this overlap for Gaussian data (right panel).*

From training set we want to find out a direction where the separation between the class means is **high** and overlap between the classes is **small**

Fisher's LD

Projection of a vector x on a unit vector w: $w^T x$
Geometric interpretation:



From training set we want to find out a direction w where the separation between the **projections** of class means is **high** and the **projections** of the class overlap is **small**

Fisher's LD

Class means: $m_1 = \frac{1}{N_1} \sum_{x_1 \in R_1} x_i$, $m_2 = \frac{2}{N_2} \sum_{x_2 \in R_2} x_i$
Projected class means:
$\tilde{m}_1 = \frac{1}{N_1} \sum_{x_i \in R_1} w^T x_i = w^T m_1$, $\tilde{m}_2 = \frac{1}{N_2} \sum_{x_i \in R_2} w^T x_i = w^T m_2$
Difference between projected class means: $\tilde{m}_2 - \tilde{m}_1 = w^T (m_2 - m_1)$
Scatter of projected data (this will indicate overlap between the classes):

$$\tilde{s}_1^2 = \sum_{y_i : x_i \in R_1} (y_i - \tilde{m}_1)^2 = \sum_{x_i \in R_1} (w^T x_i - w^T m_1)^2$$

$$= w^T \left(\sum_{x_i \in R_1} (x_i - m_1)(x_i - m_1)^T \right) w = w^T S_1 w$$

$$\tilde{s}_2^2 = \sum_{y_i : x_i \in R_2} (y_i - \tilde{m}_2)^2 = \sum_{x_i \in R_2} (w^T x_i - w^T m_2)^2$$

$$= w^T \left(\sum_{x_i \in R_2} (x_i - m_2)(x_i - m_2)^T \right) w = w^T S_2 w$$

Fisher's LD

Ratio of difference of projected means over total scatter:

$$Rayleigh\ Quotient: \ r(w) = \frac{(\tilde{m}_2 - \tilde{m}_1)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{w^T S_B w}{w^T S_w w}$$

where

$$S_w = S_1 + S_2$$
$$S_B = (m_2 - m_1)(m_2 - m_1)^T$$

We want to maximize $r(w)$, the solution is

$$w = S_w^{-1}(m_2 - m_1)$$

Fisher's LD: Classifier

So far so good. However, how do we get the classifier?

All we know at this point is that the direction $w = S_w^{-1}(m_2 - m_1)$ separates the projected data very well

Since we know that the projected class means are well separated, we can choose average of the two projected means as a threshold for classification

Classification rule: x in R_2 if $y(x) > 0$, else x in R_1, where

$$y(x) = w^T x - \frac{1}{2}(\tilde{m}_1 + \tilde{m}_2) = w^T x - \frac{1}{2} w^T$$
$$= S_w^{-1}(m_2 - m_1)(x - \frac{1}{2}(m_1 + m_2))$$

Fisher's LD and LDA

They become same when

- Prior probabilities are same
- Common covariance matrix for the class conditional densities
- Both class conditional densities are multivariate Gaussian

Ex. Show that Fisher's LD classifier and LDA produce the same rule of classification g p iven the above assumptions

Note:

(1) Fisher's LD does not assume Gaussian densities

(2) Fisher's LD can be used in dimension reduction for a multiple class scenario

Logistic Regression

- The output of regression is the posterior probability i.e., Pr(output — input)
- Always ensures that the sum of output variables is 1 and each output is non-negative
- A linear classification method
- We need to know about two concepts to understand logistic regression
  — Newton-Raphson method
  — Maximum likelihood estimation

Newton-Raphson Method

A technique for solving non-linear equation $f(x) = 0$

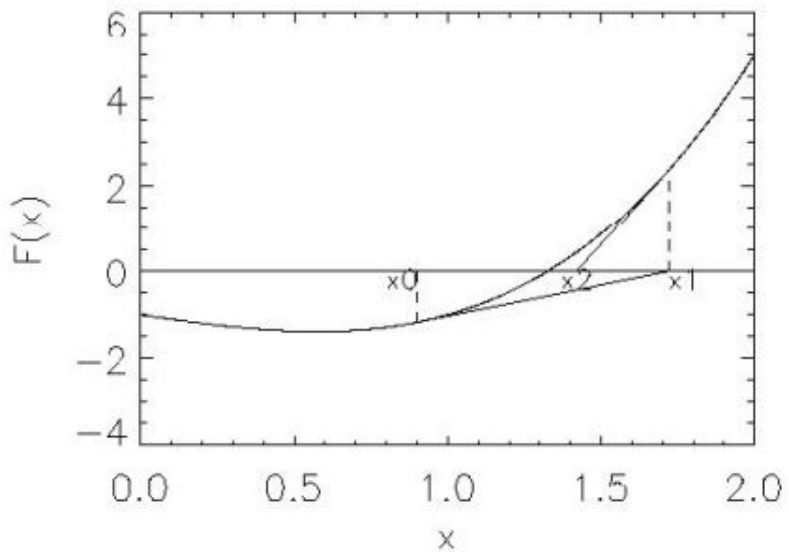
Taylor series: $f(x_{n+1}) = f(x_n) + (x_{n+1} - x_n)f^{'}(x_n)$

After rearrangement: $x_{n+1} = x_n + \frac{f(x_{n+1} - f(x_n))}{f^{'}(x_n)}$

If x_{n+1} is a root or very close to the root, then $f(x_{n+1}) \approx 0$

So the rule for iteration(Need an initial guess x_0):

$$x_{n+1} = x_n - \frac{f(x_n)}{f^{'}(x_n)}.$$



Newton-Raphson in Multi-dimensions

We want to solve the equations:

$$\begin{aligned} f_1(x_1, x_2, \cdots, x_N) &= 0 \\ f_2(x_1, x_2, \cdots, x_N) &= 0 \\ &\vdots \\ f_N(x_1, x_2, \cdots, x_N) &= 0 \end{aligned}$$

Taylor series: $f_j(x + \Delta x) = f_j(x) + \sum_{k=1}^{N} \frac{\partial f_i}{\partial x_k} \Delta x_k, j = 1, \cdots, N$

After some rearrangement etc. the rule for iteration(Need an initial guess):

$$\begin{bmatrix} x_1^{n+1} \\ x_2^{n+1} \\ \vdots \\ x_N^{n+1} \end{bmatrix} = \begin{bmatrix} x_1^{n} \\ x_2^{n} \\ \vdots \\ x_N^{n} \end{bmatrix} \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_N} \end{bmatrix}^{-1} \begin{bmatrix} f_1(x_1^n, x_2^n, \cdots, x_N^n) \\ f_2(x_1^n, x_2^n, \cdots, x_N^n) \\ \vdots \\ f_N(x_1^n, x_2^n, \cdots, x_N^n) \end{bmatrix}$$

Newton-Raphson : Example

Solve:

$$f_1(x_1, x_2) = x_1^2 - cos(x_2) = 0$$
$$f_2(x_1, x_2) = sin(x_1) + x_1^2 + x_2^3 = 0$$

$$\begin{bmatrix} x_1^{n+1} \\ x_2^{n+1} \end{bmatrix} = \begin{bmatrix} x_1^n \\ x_2^n \end{bmatrix} - \begin{bmatrix} 2x_1^n & sin(x_2^n) \\ cos(x_1^n) + 2x_1^n & 3(x_2^n)^2 \end{bmatrix}^{-1} \begin{bmatrix} (x_1^n)^2 - cos(x_2^n) \\ sin(x_1^n) + (x_1^n)^2 + (x_2^n)^3 \end{bmatrix}$$

Also Need initial guess.

Maximum Likelihood Parameter Estimation

Let's start with an example. We want to find out the unknown parameters mean and standard deviation of a Gaussian pdf, given N independent samples from it.

$$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

Samples: $x_1, \cdots, x_N$
Form the likelihood function:

$$L(\mu,\sigma) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(x_i-\mu)^2}{2\sigma^2})$$

Estimate the parameters that maximize the likelihood function

$$(\hat{\mu},\hat{\sigma}) = argmax_{\mu,\sigma} L(\mu,\sigma)$$

Let's find out $(\hat{\mu},\hat{\sigma})$

Logistic Regression Model

The method directly models the posterior probabilities as the output of regression

$$Pr(G = k|X = x) = \frac{exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{\ell=1}^{K-1} exp(\beta_{\ell 0} + \beta_\ell^T x)}$$

$$Pr(G = K|X = x) = \frac{1}{1 + \sum_{\ell=1}^{K-1} exp(\beta_{\ell 0} + \beta_\ell^T x)}$$

x is p-dimensional input vector, β_k is p-dimensional vector for each k, Total number of parameters is $(K-1)(p+1)$

Note that the class boundaries are **linear**
How can we show this linear nature?
What is the **discriminant function** for every class in this model?

Logistic Regression Computation

Let's fit the logistic regression model for $K = 2$, i.e., number of classes is 2.

Training set: $(x_i, g_i), i = 1, \cdots, N$.

log-likelihood:

$$\begin{aligned}
\ell(\beta) &= \sum_{i=1}^{N} logPr(G = y_i|X = x_i) \\
&= \sum_{i=1}^{N} y_i log(Pr(G = 1|x = x_i)) + (1 - y_i)log(Pr(G = 0|X = x_i)) \\
&= \sum_{i=1}^{N} \left(y_i \beta^T x_i + (1 - y_i) log \frac{1}{1 + exp(\beta^T x_i)} \right) \\
&= \sum_{i=1}^{N} (y_i \beta^T x_i - (1 - y_i) log(1 + exp(\beta^T x_i)))
\end{aligned}$$

where x_i are (p+1)-dimensional input vector with leading entry 1,β is a (p+1)-dimensional vector, $y_i = 1$ if $g_i = 1$;$y_i = 0$ if $g_i = 2$.

Newton-Raphson for LR

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{N} \left(y_i - \frac{exp(\beta^T x)}{1 + exp(\beta^T x)} \right) x_i = 0$$

(p + 1) Non-linear equations to solve for (p + 1) unknows

Solve by Newton-Raphson method:

$$\beta \leftarrow \beta - [Jacobian(\frac{\partial \ell(\beta)}{\partial \beta})]^{-1} \frac{\partial \ell(\beta)}{\partial \beta}$$

where,

$$Jacobian(\frac{\partial \ell(\beta)}{\partial \beta}) = -\sum_{i=1}^{N} x_i x_i^T \left(\frac{exp(\beta^T x_i)}{1 + exp(\beta^T x_i)} \right) \left(\frac{1}{1 + exp(\beta^T x_i)} \right)$$

Newton-Raphson for LR

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{N} \left(y_i - \frac{exp(\beta^T x)}{1 + exp(\beta^T x)} \right) x_i = X^T (y - p)$$

$$Jacobian(\frac{\partial \ell(\beta)}{\partial \beta}) = -X^T W X$$

So, NR rule becomes: $\beta \leftarrow \beta + (X^T W X)^{-1} X^T (y - p)$

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ . \\ . \\ . \\ x_N^T \end{bmatrix}_{N-by-(p+1)}, y = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_N \end{bmatrix}_{N-by-1}, p = \begin{bmatrix} exp(\beta^T x_1)/(1 + exp(\beta^T x_1)) \\ exp(\beta^T x_2)/(1 + exp(\beta^T x_2)) \\ . \\ . \\ . \\ exp(\beta^T x_N)/(1 + exp(\beta^T x_N s)) \end{bmatrix}_{N-by-1}$$

W is a N-by-N diagonal matrix with ith diagonal entry

$$\left(\frac{exp(\beta^T x_i)}{1 + exp(\beta^T x_i)} \right) \left(\frac{1}{1 + exp(\beta^T x_i)} \right)$$

Newton-Raphson for LR

— Newton-Raphson

$$\begin{aligned}\beta^{new} &= \beta^{old} + (X^T W X)^{-1} X^T (y - p) \\ &= (X^T W X)^{-1} X^T W (X\beta^{old} + W^{-1}(y - p)) \\ &= (X^T W X)^{-1} X^T W z\end{aligned}$$

— Adjusted response

$$z = X\beta^{old} + W^{-1}(y - p)$$

— Iteratively reweighted least squares (IRLS)

$$\begin{aligned}\beta^{new} &\leftarrow argmin_\beta (z - X\beta^T)^T W (z - X\beta^T) \\ &\leftarrow argmin_\beta (y - p)^T W^{-1} (y - p)\end{aligned}$$

Example: South African Heart Disease

After data fitting in the logistic regression model:

$$\Pr(MI = yes \mid x) = \frac{\exp(-4.130 + 0.006x_{sbp} + 0.08x_{tobaco} + 0.185x_{ldl} + 0.939x_{famhist} - 0.035x_{obesity} + 0.001x_{alcohol} + 0.043x_{age})}{1 + \exp(-4.130 + 0.006x_{sbp} + 0.08x_{tobaco} + 0.185x_{ldl} + 0.939x_{famhist} - 0.035x_{obesity} + 0.001x_{alcohol} + 0.043x_{age})}$$

	Coefficient	Std. Error	Z Score
(Intercept)	-4.130	0.964	-4.285
sbp	0.006	0.006	1.023
tabacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	-1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

Example: South African Heart Disease

After ignoring negligible coefficients:

$$Pr(MI = yes|x) = \frac{exp(-4.204 + 0.081x_{tobacco} + 0.168x_{ldll} + 0.924x_{famhist} + 0.044x_{age})}{1 + exp(-4.204 + 0.081x_{tobacco} + 0.168x_{ldll} + 0.924x_{famhist} + 0.044x_{age})}$$

What happened to systolic blood pressure? Obesity?

Multi-Class Logistic Regression

NR update: $\tilde{\beta} \leftarrow \tilde{\beta} + (\tilde{X}^T \tilde{W} \tilde{X})^{-1} \tilde{X}^T (\tilde{y} - \tilde{p})$

$$\tilde{\beta} = \begin{bmatrix} \beta_{10} \\ \vdots \\ \beta_{1p} \\ \beta_{20} \\ \vdots \\ \beta_{2p} \\ \vdots \\ \beta_{(K-1)0} \\ \vdots \\ \beta_{(K-1)p} \end{bmatrix}, \tilde{X} = \begin{bmatrix} X & & & \\ & X & & \\ & & X & \\ & & & X \end{bmatrix}_{N(K-1)-by-(K-1)(p+1)}$$

where,

$$X = \begin{bmatrix} 1 & x_1^T \\ 1 & x_2^T \\ \vdots & \vdots \\ 1 & x_N^T \end{bmatrix}_{N-by-(p+1)}$$

Multi-Class Logistic Regression

$\tilde{y}$ is a $N(K-1)$ dimension vector:

$$\tilde{y} = (y_1, y_2, \cdots, y_{K-1})^T$$

where $y_k = (\delta(g_1 - k), \delta(g_2 - k), \cdots, \delta(g_N - k)), 1 \leq k \leq K-1.$

$\delta(z)$ is an indicator function:

$$\delta(z) = \begin{cases} 1, & if\ z = 0 \\ 0, & otherwize \end{cases}$$

$\tilde{p}$ is a $N(K-1)$ dimension vector:

$$\tilde{p} = \begin{bmatrix} p_1 \\ p_2 \\ . \\ . \\ . \\ p_{K-1} \end{bmatrix}, where\ p_k = \begin{bmatrix} exp(\beta_{k0} + \beta_k^T x_1)/(1 + \sum_{\ell=1}^{K-1} exp(\beta_{\ell 0} + \beta_\ell^T x_1)) \\ exp(\beta_{k0} + \beta_k^T x_2)/(1 + \sum_{\ell=1}^{K-1} exp(\beta_{\ell 0} + \beta_\ell^T x_2)) \\ . \\ . \\ . \\ exp(\beta_{k0} + \beta_k^T x_N)/(1 + \sum_{\ell=1}^{K-1} exp(\beta_{\ell 0} + \beta_\ell^T x_N)) \end{bmatrix}, 1 \leq k \leq K-1.$$

Multi-Class Logistic Regression

$$W = \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1(K-1)} \\ W_{21} & W_{22} & \cdots & W_{2(K-1)} \\ \vdots & \vdots & \vdots & \vdots \\ W_{(K-1)1} & W_{(K-1)2} & \cdots & W_{(K-1)(K-1)} \end{bmatrix}$$

where $W_{km}(1 \leq k, m \leq K-1)$ is an N-by-N diagonal matrix,

if $k = m$, then the ith diagonal entry is

$$\left(\frac{exp(\beta_{k0} + \beta_k^T x_i)}{1 + \sum_{\ell=1}^{K-1} exp(\beta_{\ell 0} + \beta_\ell^T x_i)} \right) \left(1 - \frac{exp(\beta_{k0} + \beta_k^T x_i)}{1 + \sum_{\ell=1}^{K-1} exp(\beta_{\ell 0} + \beta_\ell^T x_i)} \right)$$

if $k \neq m$, then the ith diagonal entry is

$$- \left(\frac{exp(\beta_{k0} + \beta_k^T x_i)}{1 + \sum_{\ell=1}^{K-1} exp(\beta_{\ell 0} + \beta_\ell^T x_i)} \right) \left(1 - \frac{exp(\beta_{k0} + \beta_k^T x_i)}{1 + \sum_{\ell=1}^{K-1} exp(\beta_{\ell 0} + \beta_\ell^T x_i)} \right)$$

LDA vs Logistic Regression

- LDA(Generative model)
  1. Assumes Gaussian class-conditional densities and a common covariance
  2. Model parameters are estimated by maximizing the full log likelihood, parameters for each class are estimated independently of other classes, $Kp + p(p+1)/2 + (K-1)$ parameters
  3. Makes use of marginal density information $Pr(X)$
  4. Easier to train, low variance, more efficient if model is correct
  5. Higher asymptotic error, but converges faster
- Logistic Regression(Discriminative model)
  1. Assumes class-conditional densities are members of the (same) exponential family distribution
  2. Model parameters are estimated by maximizing the conditional log likelihood, simultaneous consideration of all other classes, $(K-1)(p+1)$ parameters
  3. Ignores marginal density information $Pr(X)$
  4. Harder to train, robust to uncertainty about the data generation process
  5. Lower asymptotic error, but converges more slowly

Generative vs Discriminative Learning

	Generative	Discriminative
Example	Linear Discriminant Analysis	Logistic Regression
Objective Functions	Full log likelihood: $\sum_i log p_\theta(x_i, y_i)$	Conditional log likelihood: $\sum_i log p_\theta(y_i \| x_i)$
Model Assumptions	Class densities: $p(x\|y = k)$ e.g. Gaussian in LDA	Discriminant functions $\lambda_k(x)$
Parameter Estimation	"Easy"-One single sweep	"Hard"-iterative opimization
Advantages	if model correct, More efficient borrows strength from $p(x)$	More flexible, robust because fewer assumptions
Disadvantages	is incorrect Bias if model	May also be biased. Ignores information in $p(x)$