Classification

Wei Li

Syracuse University

Spring 2021

Classification

Sensitivity and Specificity

Binary classification

Unequal losses

LDA and QDA

Logistic regression

Sensitivity and Specificity

Sensitivity and Specificity

Imagine a scenario where people are tested for a disease:

- The test outcome can be positive (sick) or negative (healthy)
- The actual health status of the persons can be sick or healthy

There are four possible scenarios:

- True positive (TP): sick people correctly diagnosed as sick
- False positive (FP) : healthy people incorrectly identified as sick
- True negative (TN): healthy people correctly identified as healthy
- False negative (FN): sick people incorrectly identified as healthy.

	Test Outcome		
True outcome	Positive	Negative	Total
Positive	True Pos. (TP)	False Neg. (FN)	P
Negative	False Pos. (FP)	True Neg. (TN)	N
	P^*	N^*	

	Test Outcome		
True outcome	Positive	Negative	Total
Positive	True Pos. (TP)	False Neg. (FN)	P
Negative	False Pos. (FP)	True Neg. (TN)	N
	P^*	N^*	

Sensitivity: the proportions of positives that are correctly identified (true positive rate)

$$\text{Sensitivity} = \text{TP}/\text{P} = \text{TP}/(\text{TP} + \text{FN})$$

Specificity: the proportions of negative that are correctly identified (true negative rate)

$$\text{Specificity} = \text{TN}/\text{N} = \text{TN}/(\text{FP} + \text{TN})$$

- Type I error: false alarm rate, Type I error $= 1-$ Specificity.
- Type II error: false negative rate, $= 1-$ Sensitivity, Power $=$ Sensitivity.
- False Discovery Rate (FDR): the proportion of predicted positives that are in fact false positives

$$FDR = FP/P^*$$

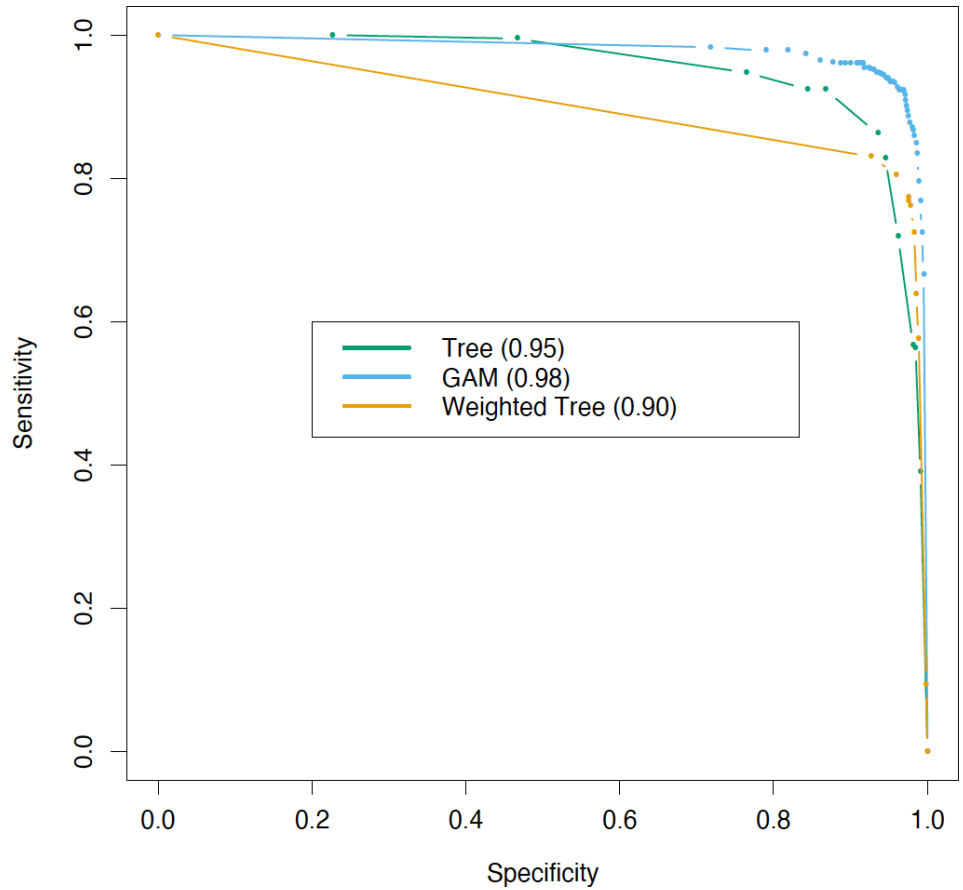
Example:

	Predicted	
True	email	spam
email	57.3%	4.0%
spam	5.3%	33.4%

spam =presence of disease, email=absence of disease.

$$\begin{aligned} \text{specificity} &= 100 \times \frac{57.3}{57.3+4.0} = 93.4\% \\ \text{sensitivity} &= 100 \times \frac{33.4}{33.4+5.3} = 86.3\% \end{aligned}$$

The **ROC curve** is a plot of the sensitivity versus specificity. Area under curve (AUC) is a commonly used quantitative summary. An ideal ROC curve will hug the top right corner, so the larger the AUC the better the classifier.



ESL Figure 9.6

Binary classification

Binary classification

- input vector $X \in \mathcal{X} \subset R^d$
- output $Y \in \{0, 1\}$
- the goal is to construct a function $h : \mathcal{X} \longrightarrow \{0, 1\}$

A classification rule is characterized as

$$h(X) = I(b(X) > 0)$$

where b is the boundary function (or discriminant function) that gives the decision boundary $\{x : b(x) = 0\}$.

- If $b(X)$ is a linear in X, then the classifier has a linear boundary.

The classification error rate, of h is defined as

$$R(h) = E_{X,Y}(I(Y \neq h(X))) = P(Y \neq h(X))$$

and the empirical classification error or training error is

$$\widehat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} I\left(h\left(X_i\right) \neq Y_i\right)$$

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 $m(x) = P(Y = 1 \mid X = x) = E(Y \mid X = x)$.

- This optimal rule is called the **Bayes rule** (under equal costs).
- The risk $R(h^*)$ is called the **Bayes risk**.
- The set $\{x : m(x) - \frac{1}{2} = 0\}$ is called the **Bayes decision boundary**.

Alternatively, the Bayes rule is h^*, is given by

$$h^*(x) = \begin{cases} 1 & \text{if} \quad P(Y=1 \mid X=x) > P(Y=0 \mid X=x) \\ 0 & \text{if} \quad P(Y=1 \mid X=x) < P(Y=0 \mid X=x) \end{cases}$$

The classification boundary of the Bayes rule is

$$\begin{aligned} &\{x : P(Y=1 \mid X=x) = P(Y=0 \mid X=x)\} \\ =&\{x : P(Y=1 \mid X=x) - 0.5 = 0\} \end{aligned}$$

Examples of linear boundary

Linear logit models assume: the **logit function** is linear in x, i.e.,

$$\log \frac{\Pr(Y = 1 \mid X = x)}{\Pr(Y = 0 \mid X = x)} = \beta_0 + \beta_1^T x$$

Thus the classification boundary is given by $\{x : \beta_0 + \beta_1^T x = 0\}$

Examples: LDA, Logistic regression

Note that the posterior class probabilities $P(Y = j|X = x)$ provide updated class probabilities after observing x. From Bayes' theorem

$$p(Y = 1 \mid X = x) = \frac{p(x \mid Y = 1)p(Y = 1)}{p(x \mid Y = 1)p(Y = 1) + p(x \mid Y = 0)p(Y = 0)}$$
$$= \frac{\pi_1 p_1(x)}{\pi_1 p_1(x) + (1 - \pi_1)\, p_0(x)}$$

where $\pi_1 = p(Y = 1), \pi_0 = p(Y = 0)$ are the marginal distribution of Y (prior class probabilities); and $p_j(x) = p(x \mid Y = j)$, denote the conditional density of X given that $Y = j$.

Thus the Bayes rule becomes

$$h^*(x) = \begin{cases} 1 & \text{if } \frac{p_1(x)}{p_0(x)} > \frac{\pi_0}{\pi_1} \\ 0 & \text{otherwise.} \end{cases}$$

Unequal losses

Unequal losses

For any decision function, there are two possible errors:

- misclassifying a sample in class 0 to 1 (false positive)
- misclassifying a sample in class 1 to 0 (false negative)

Each type of error is associated with a cost (the price to pay for the consequence):

- $L(1,0)$ is the cost of misclassifying a sample in class 1 to 0
- $L(0,1)$ is the cost of misclassifying a sample in class 0 to 1.

We assume $L(j,j) = 0$ for $j = 0, 1$; but it may not be $L(0,1) = L(1,0)$.

The loss becomes

$$L(Y, h(X)) = L(1,0)I(Y = 1, h(X) = 0) + L(0,1)I(Y = 0, h(X) = 1)$$

For fixed x, the Bayes rule is given as

$$h^*(x) = \begin{cases} 1 & \text{if } L(1,0)P(Y=1 \mid X=x) > L(0,1)P(Y=0 \mid X=x) \\ 0 & \text{if } L(1,0)P(Y=1 \mid X=x) < L(0,1)P(Y=0 \mid X=x) \end{cases}$$

Equivalently,

$$h^*(x) = \begin{cases} 1 & \text{if } \quad \frac{P(Y=1|X=x)}{P(Y=0|X=x)} > \frac{L(0,1)}{L(1,0)} \\ 0 & \text{if } \quad \frac{P(Y=1|X=x)}{P(Y=0|X=x)} < \frac{L(0,1)}{L(1,0)} \end{cases}$$

the Bayes rule

$$h^*(x) = 1\left\{x : P(Y=1 \mid X=x) > \frac{L(0,1)}{L(0,1)+L(1,0)}\right\}.$$

Using the Bayes' theorem,

$$h^*(x) = \begin{cases} 1 & \text{if} \quad \frac{p_1(x)}{p_0(x)} > \frac{\pi_0 L(0,1)}{\pi_1 L(1,0)} \\ 0 & \text{if} \quad \frac{p_1(x)}{p_0(x)} < \frac{\pi_0 L(0,1)}{\pi_1 L(1,0)} \end{cases}$$

Any classifier aims to minimize the total losses incurred by its predictions. In building a binary classifier, by changing the weights for L_{01} and L_{10}, we can effectively change the classification threshold.

- L_{01} = the loss of predicting a "non-disease" sample to "disease"
- L_{10} = the loss of predicting a "disease" sample to "non-disease"

Changing the weights for L_{01} and L_{10} alone for a particular classifier, the ROC curve can also be changed.

Multi-class classification

- Class label $Y \in \{1, \ldots, K\}, K \geq 3$.
- The classifier $h : R^d \longrightarrow \{1, \ldots, K\}$.

The loss function $L(Y, h(X)) = \sum_{k=1}^{L} \sum_{l=1}^{K} C(l, k) I(Y = l, h(X) = k)$ where $C(l, k) =$ cost of classifying a sample in class l to class k.

The classification risk, or error rate, of h is defined as

$$R(h) = E_{X,Y}(L(Y, h(X)))$$

Using the 0-1 loss, $C(k, k) = 0$ for any $k = 1, \cdots, K$, but equal to 1 otherwise, the rule h that minimizes $R(h)$ is

$$h^*(x) = \arg \max_{k=1,\ldots,K} P(Y = k \mid x)$$

i.e., assign x to the most probable class using $P(Y \mid x)$.

For the general loss function, the Bayes rule can be derived as

$$h^*(x) = k^* \quad \text{if} \quad k^* = \arg \min_{k=1,\ldots,K} \sum_{l=1}^{K} C(l,k) P(Y = l \mid x).$$

We generally need to estimate multiple **discriminant functions** $\delta_k(x), k = 1, \cdots, K$

- Each $\delta_k(x)$ is associated with class k.
- $\delta_k(x)$ represents the evidence strength of a sample (x, y) belonging to class k.

The decision rule constructed using δ_k 's is

$$\hat{h}(x) = k^*, \quad \text{where} \quad k^* = \arg \max_{k=1,\ldots,K} \delta_k(x)$$

The decision boundary of the classification rule $\hat{h}$ between class k and class l is defined as

$$\{x : \delta_k(x) = \delta_l(x)\}$$

LDA and QDA

Consider the binary classification

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 $r_i = \sqrt{(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}$ for $i = 0, 1$ is the **Mahalanobis distance** between x and μ_i.

QDA

$$\log \frac{\Pr(Y=1 \mid X=x)}{\Pr(Y=0 \mid X=x)} = \log \frac{\pi_1 \phi\left(x; \boldsymbol{\mu}_1, \Sigma_1\right)}{\pi_0 \phi\left(x; \boldsymbol{\mu}_0, \Sigma_0\right)} = \delta_1(x) - \delta_0(x).$$

This rule can also be written as

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

where

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

is called the **Gaussian discriminant function**.

- The decision boundary: $\{x \in \mathcal{X} : \delta_1(x) = \delta_0(x)\}$
- **quadratic discriminant analysis** (QDA).

To estimate $\pi_0, \pi_1, \mu_0, \mu_1, \Sigma_0, \Sigma_1$:

$$\widehat{\pi}_0 = \frac{1}{n}\sum_{i=1}^{n}(1-Y_i), \quad \widehat{\pi}_1 = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

$$\widehat{\mu}_0 = \frac{1}{n_0}\sum_{i:Y_i=0} X_i, \quad \widehat{\mu}_1 = \frac{1}{n_1}\sum_{i:Y_i=1} X_i$$

$$\widehat{\Sigma}_0 = \frac{1}{n_0-1}\sum_{i:Y_i=0}(X_i-\widehat{\mu}_0)(X_i-\widehat{\mu}_0)^T$$

$$\widehat{\Sigma}_1 = \frac{1}{n_1-1}\sum_{i:Y_i=1}(X_i-\widehat{\mu}_1)(X_i-\widehat{\mu}_1)^T$$

LDA

LDA assumes both classes are from Gaussian and they have the same covariance matrix

$$\Sigma_k = \Sigma, \quad k = 0, 1$$

Note that

$$\log \Pr(Y = k \mid X = x) = -\frac{1}{2} (x - \boldsymbol{\mu}_k)^T \Sigma^{-1} (x - \boldsymbol{\mu}_k) + \log \pi_k + \text{ const.}$$

Alternatively,

$$h^*(x) = \text{argmax}_k \, \delta_k(x)$$

discriminant function is simplified

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

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 classification rule is

$$h^*(x) = \begin{cases} 1 & \text{if } \delta_1(x) > \delta_0(x) \\ 0 & \text{otherwise} \end{cases}$$

multi-class classification

QDA assume that $X \mid Y = k \sim N(\mu_k, \Sigma_k)$. The Bayes rule for the multiclass QDA can be written as

$$h^*(x) = \text{argmax}_k \, \delta_k(x)$$

where

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

If all Gaussians assumed to have equal variance Σ,

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

The corresponding estimates are given by

$$\widehat{\pi}_k = \frac{1}{n} \sum_{i=1}^n I(y_i = k), \quad \widehat{\mu}_k = \frac{1}{n_k} \sum_{i:Y_i=k} X_i$$
$$\widehat{\Sigma}_k = \frac{1}{n_k - 1} \sum_{i:Y_i=k} (X_i - \widehat{\mu}_k)(X_i - \widehat{\mu}_k)^T$$
$$\widehat{\Sigma} = \frac{\sum_{k=0}^{K-1} (n_k - 1) \widehat{\Sigma}_k}{n - K}.$$

Logistic regression

Binary case

The logistic regression assumes that

$$p_1(x;\beta_0,\boldsymbol{\beta}_1) := P(Y=1 \mid X=x) = \frac{\exp\left(\beta_0 + x^T\boldsymbol{\beta}_1\right)}{1+\exp\left(\beta_0 + x^T\boldsymbol{\beta}_1\right)}$$

The model can be written as

$$\text{logit}(x) := \text{logit}(\Pr(Y=1 \mid X=x)) = \log\frac{\Pr(Y=1 \mid X=x)}{\Pr(Y=0 \mid X=x)} = \beta_0 + \boldsymbol{\beta}_1^T x$$

- logit function: $\text{logit}(a) = \log(a/(1-a)) : (0,1) \mapsto \mathbb{R}$
- The inverse of logit function is called "logistic function" (or sigmoid function), which is given by

$$\sigma(a) = \exp(a)/(1+\exp(a)) : \mathbb{R} \mapsto (0,1)$$

So the model can be written as
$P(Y=1 \mid X=x) = Ber(Y=1|\sigma(\beta_0 + x^T\boldsymbol{\beta}_1))$

Interpretation of β_j

$$e^{\beta_j} = \frac{\text{odds}\,(\ldots, X_j = x+1, \ldots)}{\text{odds}\,(\ldots, X_j = x, \ldots)} = \text{oddsratio}$$

If $X_j = 0$ or 1, then odds for group with $X_j = 1$ are e^{β_j} higher than for group with $X_j = 0$, with other values of X_{-j} fixed.

With rare incidents $P(Y = 1) \approx 0$, odds $\approx Pr(Y = 1|\cdots)$,

$$e^{\beta_j} \approx \frac{\Pr\,(\ldots, X_j = x+1, \ldots)}{\Pr\,(\ldots, X_j = x, \ldots)} = \text{relativerisk}$$

MLE for logistic models

Notations: assuming x_i contains the constant term 1 (thus a $p+1$ vector).

$$\begin{aligned}
\boldsymbol{\beta} &:= \{\beta_0, \boldsymbol{\beta}_1^T\}^T \\
\mathbf{y} &:= [y_1, \cdots, y_n]^T \\
\mathbf{p} &:= \mathbf{p}(\boldsymbol{\beta}) = [p(x_1; \boldsymbol{\beta}), \cdots, p(x_n; \boldsymbol{\beta})]^T \\
\mathbf{W} &:= \mathbf{W}(\boldsymbol{\beta}) = \operatorname{diag}\{p(x_i; \boldsymbol{\beta})(1 - p(x_i; \boldsymbol{\beta}))\} : n \times n
\end{aligned}$$

The log (conditional) likelihood function is

$$\begin{aligned}
\ell(\boldsymbol{\beta}) &= \sum_{i=1}^{n} \{y_i \log p(x_i; \boldsymbol{\beta}) + (1 - y_i) \log [1 - p(x_i, \boldsymbol{\beta})]\} \\
&= \sum_{i=1}^{n} \left\{ y_i \boldsymbol{\beta}^T x_i - \log \left[1 + \exp\left(\boldsymbol{\beta}^T x_i \right) \right] \right\}.
\end{aligned}$$

The score and Hessian are given by

$$\frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} x_i \left[y_i - p\left(x_i; \boldsymbol{\beta}\right) \right] = \mathbf{X}^T (\mathbf{y} - \mathbf{p})$$

$$\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = -\sum_{i=1}^{n} x_i x_i^T p\left(x_i; \boldsymbol{\beta}\right) \left[1 - p\left(x_i; \boldsymbol{\beta}\right) \right] = -\mathbf{X}^T \mathbf{W} \mathbf{X}$$

Newton-Raphson step: In the k-th sep,

$$\begin{aligned}
\boldsymbol{\beta}^{(\mathrm{k}+1)} &= \boldsymbol{\beta}^{(\mathrm{k})} + \left(\mathbf{X}^T\mathbf{W}^{(k)}\mathbf{X}\right)^{-1}\mathbf{X}^T(\mathbf{y}-\mathbf{p}^{(k)}) \\
&= \left(\mathbf{X}^T\mathbf{W}^{(k)}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{W}^{(k)}\left(\mathbf{X}\beta^{(\mathrm{k})} + {\mathbf{W}^{(k)}}^{-1}(\mathbf{y}-\mathbf{p}^{(k)})\right) \\
&= \left(\mathbf{X}^T\mathbf{W}^{(k)}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{W}^{(k)}\mathbf{z}^{(k)}
\end{aligned}$$

where we defined the adjusted response

$$\mathbf{z}^{(k)} = \mathbf{X}\boldsymbol{\beta}^{(\mathrm{k})} + {\mathbf{W}^{(k)}}^{-1}(\mathbf{y}-\mathbf{p}^{(k)})$$

The update is equivalent to solving the weighted LS till convergence (**Iteratively Reweighted Least Squares Algorithm**):

$$\boldsymbol{\beta}^{(\mathrm{k}+1)} = \arg\min_{\boldsymbol{\beta}} (\mathbf{z}^{(k)} - \mathbf{X}\boldsymbol{\beta})^T \mathbf{W}^{(k)} (\mathbf{z}^{(k)} - \mathbf{X}\boldsymbol{\beta}).$$

Using central limit theorem,

$$\widehat{\boldsymbol{\beta}} \to N\left(\boldsymbol{\beta}, \left(\mathbf{X}^T \mathbf{W}(\beta^*)\mathbf{X}\right)^{-1}\right)$$

.

Multi-class case

Suppose there are K groups. Let the K-th group be the base group. One may model $Pr(Y = k|x; \boldsymbol{\beta}_0, \boldsymbol{\beta})$ as

$$Pr(Y = k|x; \boldsymbol{\beta}_0, \boldsymbol{\beta}) = \frac{\exp\left(x^T \boldsymbol{\beta}_k + \beta_{k0}\right)}{\sum_{k'} \exp\left(x^T \boldsymbol{\beta}_{k'} + \beta_{k'0}\right)}$$

- $\boldsymbol{\eta} = \boldsymbol{\beta} x + \boldsymbol{\beta}_0$ the vector of logits
- $\boldsymbol{\beta} := (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)^T$, a K by p matrix.

The **S** is the **softmax function** $\mathbb{R}^K \mapsto \mathbb{R}^K$, defined as

$$\mathbf{S}(\boldsymbol{\eta})_k = \frac{e^{\eta_k}}{\sum_{k'}^{K} e^{\eta_{k'}}}, \quad k = 1, \ldots, K; \quad \boldsymbol{\eta} = (\eta_1, \ldots, \eta_K)^T$$

- We set $\beta_{K0} = 0, \boldsymbol{\beta}_K = 0$ to avoid overparametrization.

The multi-class logistic regression (or multinomial logistic regression) models $K-1$ logits:

$$\log \frac{\Pr(Y=1 \mid X=x)}{\Pr(Y=K \mid X=x)} = \beta_{10} + \boldsymbol{\beta}_1^T x$$
$$\log \frac{\Pr(Y=2 \mid X=x)}{\Pr(Y=K \mid X=x)} = \beta_{20} + \boldsymbol{\beta}_2^T x$$
$$\log \frac{\Pr(Y=K-1 \mid X=x)}{\Pr(Y=K \mid X=x)} = \beta_{(K-1)0} + \boldsymbol{\beta}_{K-1}^T x$$

Equivalently,

$$p_k(x) \equiv \Pr(Y = k \mid x) = \frac{\exp\left(\beta_{k0} + \beta_k^T x\right)}{1 + \sum_{l=1}^{K-1} \exp\left(\beta_{l0} + \beta_I^T x\right)} \text{ for } \quad k = 1, \ldots, K-1$$

$$p_K(x) \equiv \Pr(Y = K \mid x) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp\left(\beta_{I0} + \boldsymbol{\beta}_l^T x\right)}$$

Clearly $\sum_{k=1}^{K} p_k(x) = 1$. The parameter vector

$$\boldsymbol{\theta} = \left\{\beta_{10}, \boldsymbol{\beta}_1^T, \ldots, \beta_{(K-1)0}, \boldsymbol{\beta}_{K-1}^T\right\}^T$$

Let $p_{k,i} := Pr(Y_i = k|X = x_i, \boldsymbol{\theta})$ and
$p_{y_i}(x_i; \boldsymbol{\theta}) = Pr(Y_i = y_i|X = x_i, \boldsymbol{\theta})$.

$$\begin{aligned}
\ell(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \log p_{y_i}(x_i; \boldsymbol{\theta}) \\
&= \log \left(\prod_{i=1}^{n} \prod_{k=1}^{K} p_{k,i}^{1(y_i=k)} \right) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} 1(y_i = k) \log(p_{k,i})
\end{aligned}$$

Let $\beta_{K0} := 0$ and $\boldsymbol{\beta}_K := 0$, The local log-likelihood:

$$\sum_{i=1}^{n} \left\{ \beta_{y_i 0} + \beta_{y_i}^T x_i - \log \left[1 + \sum_{k=1}^{K-1} \exp\left(\beta_{k0} + \beta_k^T x_i\right) \right] \right\}$$

Compare logistic regression with LDA

Logistic regression:

- Maximizing the conditional likelihood, the multinomial likelihood with probabilities $\Pr(Y = k \mid \mathbf{X})$
- The marginal density $\Pr(X)$ is totally ignored (fully nonparametric using the empirical distribution function which places $1/n$ at each observation)

LDA:

- Maximizing the full log-likelihood based on the joint density

$$\Pr(X, Y = k) = \phi\left(X; \boldsymbol{\mu}_k, \Sigma\right) \pi_k$$

- Standard MLE theory leads to estimators $\hat{\boldsymbol{\mu}}_k, \hat{\Sigma}, \hat{\pi}_k$
- Marginal density does play a role $\Pr(\mathbf{X}) = \sum_k \pi_k \phi\left(X; \boldsymbol{\mu}_k, \Sigma\right)$

Regularized logistic regression

The idea is to minimize the penalized negative likelihood function (binary response):

$$\min_{\beta_0,\boldsymbol{\beta}_1} \sum_{i=1}^{n} \left(-y_i \left(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}_1 \right) + \log \left(1 + e^{\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}_1} \right) \right) + \lambda J(\boldsymbol{\beta}_1)$$

The update is equivalent to solving the weighted LS till convergence (Iteratively Reweighted Least Squares Algorithm):

$$(\beta_0^{(k+1)}, \boldsymbol{\beta_1}^{(\mathrm{k+1})}) = \arg\min_{\boldsymbol{\beta}} \left\{ (\mathbf{z}^{(k)} - \mathbf{X}\boldsymbol{\beta})^T \mathbf{W}^{(k)} (\mathbf{z}^{(k)} - \mathbf{X}\boldsymbol{\beta}) + \lambda J(\boldsymbol{\beta}_1^{(1)}) \right\}.$$

For the Lasso penalty, it can be solved using coodinate descent.