The likelihood function is given by

$$L = \prod_{i=1}^{n} \prod_{k=1}^{m} (p_{ik})^{z_{ik}},$$

where the inside product includes only those p_{ik} for which $z_{ik} = 1$. The log-likelihood function equals

$$\ln L = \sum_{i=1}^{n} \sum_{k=1}^{m} z_{ik} \ln p_{ik}.$$

The MLE $\widehat{\boldsymbol{\beta}}_k = (\widehat{\beta}_{0k}, \widehat{\beta}_{1k}, \ldots, \widehat{\beta}_{pk})'$ maximizes $\ln L$. The asymptotic variance-covariance matrix of $\widehat{\boldsymbol{\beta}}$ is given by the inverse of the information matrix, which can be used to make inferences on the parameters of the model.

Classification

To classify the outcome for any observation with given $\boldsymbol{x} = (1, x_1, \ldots, x_p)'$ vector, we can calculate the estimated probabilities of all m outcomes:

$$\widehat{p}_k(\boldsymbol{x}) = \exp(\boldsymbol{x}'\widehat{\boldsymbol{\beta}}_k)\widehat{p}_m(\boldsymbol{x}) \ (k = 1, \ldots, m-1),$$

where

$$\widehat{p}_m(\boldsymbol{x}) = \frac{1}{1 + \sum_{j=1}^{m-1} \exp(\boldsymbol{x}'\widehat{\boldsymbol{\beta}}_j)}.$$

Then we can predict the outcome k^* as that value of k, which maximizes $\widehat{p}_k(\boldsymbol{x})$. We refer to this as the **maximum probability classifier**.

Suppose we have available **prior probabilities**, $\pi_1, \ldots, \pi_m$, of the m possible responses, where $\sum_{k=1}^{m} \pi_k = 1$. Then their **posterior probabilities** are given by

$$\widehat{p}_k^*(\boldsymbol{x}) = \frac{\pi_k \widehat{p}_k(\boldsymbol{x})}{\sum_{j=1}^{m} \pi_j \widehat{p}_j(\boldsymbol{x})} \quad (1 \le k \le m).$$

The **Bayes classifier** (see Section 8.3) classifies $\boldsymbol{x}$ to that group k^*, which maximizes $\widehat{p}_k^*(\boldsymbol{x})$. If the prior probabilities are equal, $\pi_k = 1/m$ for all k then the Bayes classifier reduces to the maximum probability classifier. We will discuss the rationale behind the Bayes classifier in the next chapter.

EXAMPLE 7.11 (MBA Admissions: Nominal Logistic Regression Model)

We use the library `nnet` to perform nominal logistic regression, which we find more convenient to use than the library `mlogit`. The R script is shown below. Note that we specified the maximum number of iterations to be 1000 in the `multinom` function; the default is 100.

```
> library(nnet)
> MBA=read.csv("c:/data/MBA.csv")
> fit1=multinom(admit~GPA + GMAT, data = MBA,maxit=1000)
# weights:  12 (6 variable)

> summary(fit1)
```

The output is as follows.

```
Call:
multinom(formula = admit ~ GPA + GMAT, data = MBA, maxit = 1000)
```

```
Coefficients:
  (Intercept)         GPA        GMAT
2    155.2575   -28.86054 -0.1350340
3    424.2290  -101.90704 -0.2830152

Std. Errors:
  (Intercept)       GPA       GMAT
2   0.2360282 1.706583 0.01210057
3   0.6922946 2.588929 0.01785585

Residual Deviance: 11.16483
AIC: 23.16483
```

The `multinom` function took 390 iterations to converge because the MBA admissions data form nearly separated clusters as seen in Figure 7.1; as a result, fitting the nominal logistic regression model is more difficult. This may seem counterintuitive but can be seen from the following example. Consider fitting a simple logistic regression model to the data in which all x-values corresponding to $y = 0$ are less than those corresponding to $y = 1$, so the two clusters are completely separated. Then the MLE of the slope β_1 of the logistic response curve must approach ∞ and so its MLE does not converge. Exercise 7.2 gives a small data set to illustrate this phenomenon.

Note that `admit = 1` (admit) is used as the reference level by default which explains the negative coefficients on GPA and GMAT. Denoting the predicted probabilities of the three admission decisions by $\widehat{p}_1, \widehat{p}_2$ and $\widehat{p}_3$, the models are

$$\ln\left(\frac{\widehat{p}_2}{\widehat{p}_1}\right) = 155.2507 - 28.8605 \times \text{GPA} - 0.1350 \times \text{GMAT}$$

and

$$\ln\left(\frac{\widehat{p}_3}{\widehat{p}_1}\right) = 424.2290 - 101.9070 \times \text{GPA} - 0.2830 \times \text{GMAT}.$$

By computing the ratios of the estimated regression coefficients to their standard errors (the z-statistics) all the coefficients can be seen to be highly significant.

Next let us consider predictions using this model. First we compute predicted probabilities for an applicant with GPA = 3.2 and GMAT = 450.

```
> predicted=predict(fit1,type='probs',newdata=data.frame(GPA=3.20,GMAT=450))
> predicted
            1            2            3
1.054139e-01 8.945861e-01 2.129714e-14
```

Thus we get $\widehat{p}_1 \approx 0.1054, \widehat{p}_2 \approx 0.8946$ and $\widehat{p}_3 \approx 0$. Since $\widehat{p}_2$ is the largest, this applicant will be most likely wait-listed (`admit = 2`) with almost 90% probability.

We can check the values $\widehat{p}_,\widehat{p}_2$ and $\widehat{p}_3$ by hand calculation as follows. We have

$$\ln\left(\frac{\widehat{p}_2}{\widehat{p}_1}\right) = 155.2507 - 28.8605 \times 3.20 - 0.1350 \times 450 = 2.1539$$

and

$$\ln\left(\frac{\widehat{p}_3}{\widehat{p}_1}\right) = 424.2290 - 101.9070 \times 3.20 - 0.2830 \times 450 = -29.223.$$

Since $e^{-29.223} \approx 0$, we ignore it in the following calculation. Thus

$$\widehat{p}_1 = \frac{1}{1 + e^{2.1539}} = 0.1040, \widehat{p}_2 = \frac{e^{2.1539}}{1 + e^{2.1539}} = 0.8960 \quad \text{and} \quad \widehat{p}_3 \approx 0.$$

These hand-calculated probabilities match closely those calculated in the R output above.

Next we calculate the confusion matrix by applying the maximum probability rule for classification.

```
> Y.prob.1 = fitted(fit1, outcome= FALSE);
> Y.hat.1 = rep(0,n);
> for(i in 1:n){if(max(Y.prob.1[i,]) == Y.prob.1[i,1]){Y.hat.1[i]=1;}
> else if(max(Y.prob.1[i,]) == Y.prob.1[i,2]){Y.hat.1[i]=2;}
> else if(max(Y.prob.1[i,]) == Y.prob.1[i,3]){Y.hat.1[i]=3;}}
>
> ctable1 = table(MBA$admit, Y.hat.1);
> ctable1;
   Y.hat.1
     1  2  3
  1 30  1  0
  2  2 23  1
  3  0  0 28
> correct.rate1 = sum(diag(ctable1)[1:3])/n;
> correct.rate1
[1] 0.9529412
```

From the confusion matrix we see that there are four misclassifications so the correct classification rate (CCR) is $(30 + 23 + 28)/85 = 95.29\%$. ■

7.6.2 Logistic Regression for Ordinal Response

Now suppose the responses are ordered: $1 < 2 < \cdots < m$. Then it makes sense to define cumulative probabilities $P(y \leq k),\ k = 1, \ldots, m$. Note that $P(y \leq m) = 1$. Next define **cumulative logits**:

$$\ln\left[\frac{P(y \leq k)}{P(y > k)}\right], \quad k = 1, \ldots, m-1.$$

A linear model is postulated on these cumulative logits. The part of the model that depends on the predictor variables will be assumed to be common to all cumulative logits. Thus let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$ denote a common parameter vector and $\boldsymbol{x} = (x_1, \ldots, x_p)'$ denote the predictor variable vector. Note that these vectors don't include the intercept term β_0 as is the case with other models in this chapter. The final model is given by

$$\ln\left[\frac{P(y \leq k)}{P(y > k)}\right] = \beta_{0k} + \boldsymbol{x}'\boldsymbol{\beta},\ k = 1, \ldots, m-1, \tag{7.14}$$

where $\beta_{01} < \beta_{02} < \cdots < \beta_{0,m-1}$. Note that this model is equivalent to

$$P(y \leq k) = \frac{\exp(\beta_{0k} + \boldsymbol{x}'\boldsymbol{\beta})}{1 + \exp(\beta_{0k} + \boldsymbol{x}'\boldsymbol{\beta})},\ k = 1, \ldots, m-1. \tag{7.15}$$

The β_{0k} terms are constrained to be non-decreasing to ensure that the cumulative logits (and hence the cumulative probabilities) are non-decreasing.

Note from (7.14) that the difference in log-odds of two individuals i and j with covariate vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ for any category k is

$$\ln\left[\frac{P(y \leq k|\boldsymbol{x}_i)}{P(y > k|\boldsymbol{x}_i)}\right] - \ln\left[\frac{P(y \leq k|\boldsymbol{x}_j)}{P(y > k|\boldsymbol{x}_j)}\right] = (\boldsymbol{x}_i - \boldsymbol{x}_j)'\boldsymbol{\beta},$$