Chapter 6

Classification

6.1 Introduction

Often encountered in applications is the situation where the response variable Y takes values in a finite set of labels. For example, the response Y could encode the information whether a patient has disease type A, B or C; or it could describe whether a customer responds positively about a marketing campaign.
We always encode such information about classes or labels by the numbers $0, 1, \ldots, J-1$. Thus, $Y \in \{0, \ldots, J-1\}$, without any ordering among these numbers $0, 1, \ldots, J-1$. In other words, our sample space consists of J different *groups* ("sub-populations") and our goal is to *classify* the observations using the (p-dimensional) explanatory variables.
Given data which are realizations from

$$(X_1, Y_1), \ldots, (X_n, Y_n) \text{ i.i.d. },$$

the goal is often to assign the probabilities

$$\pi_j(x) = \mathbb{P}[Y = j \mid X = x] \quad (j = 0, 1, \ldots, J-1),$$

which is similar to the regression function $m(x) = \mathbb{E}[Y \mid X = x]$ in regression. The multivariate function $\pi_j(\cdot)$ then also allows to predict the class Y_{new} at a new observed predictor X_{new}.

6.2 The Bayes classifier

A classifier $\mathcal{C} : \mathbb{R}^p \to \{0, \ldots, J-1\}$ is a function which assigns to a predictor $X \in \mathbb{R}^p$ a class or label which is a prediction for the corresponding Y. The quality of a classifier is often measured by the expected zero-one test set error:

$$\mathbb{P}[\mathcal{C}(X_{new}) \neq Y_{new}].$$

In case where $\mathcal{C} = \widehat{\mathcal{C}}$ is estimated from training data, we consider the generalization error

$$\mathbb{P}_{\text{train},(X_{new},Y_{new})}[\widehat{\mathcal{C}}(X_{new}) \neq Y_{new}].$$

The optimal classifier with respect to the zero-one error is the **Bayes classifier**, defined "pointwise", i.e., for each x individually, as

$$\mathcal{C}_{Bayes}(x) = \underset{0 \leq j \leq J-1}{\arg\max}\, \pi_j(x). \tag{6.1}$$

Its corresponding expected zero-one test set error is called the **Bayes risk**

$$\mathbb{P}[\mathcal{C}_{Bayes}(X_{new}) \neq Y_{new}].$$

In practice, we do not know $\pi_j(\cdot)$ (and hence the Bayes classifier or risk are as unknown as MSE or bias in regression). Various methods and models are then used to come up with multivariate function estimates, either parametric or nonparametric, to obtain $\hat{\pi}_j(\cdot)$. With this, we can then estimate a classifier by plugging into the Bayes classifier

$$\widehat{\mathcal{C}}(x) = \operatorname*{arg\,max}_{0 \le j \le J-1} \hat{\pi}_j(x). \tag{6.2}$$

Such estimated classifiers are widely used, by using various models for $\pi_j(\cdot)$. However, there are also direct ways to come up with estimated classifiers without trying to estimate the conditional probability function $\pi_j(\cdot)$: an important class of examples are the support vector machines (which we will not discuss in this course). Another is the discriminant analysis view:

6.3 The view of discriminant analysis

6.3.1 Linear discriminant analysis

For the so-called linear discriminant analysis, we assume the following model:

$$\begin{aligned} (X \mid Y = j) &\sim \mathcal{N}_p(\mu_j, \boldsymbol{\Sigma}), \\ \mathrm{P}[Y = j] &= p_j, \quad \sum_{j=0}^{J-1} p_j = 1; \qquad j = 0, 1, \ldots, J-1. \end{aligned} \tag{6.3}$$

The conditional distribution of $Y \mid X$ can then be computed by the Bayes formula

$$\mathbb{P}[Y = j \mid X = x] = \pi_j(x) = \frac{f_{X|Y=j}(x) p_j}{\sum_{k=0}^{J-1} f_{X|Y=k}(x) p_k}, \tag{6.4}$$

where $f_{X|Y=j}(\cdot)$ denotes the density of the p-dimensional Gaussian distribution $\mathcal{N}_p(\mu_j, \boldsymbol{\Sigma})$. We can interpret this conditional distribution as the a-posteriori distribution of Y given X by using the a-priori distribution p_j for Y.

The unknown parameters in (6.4) are μ_j and $\boldsymbol{\Sigma}$ (or $\boldsymbol{\Sigma}_j$ if the covariances may differ per group, see 6.3.2) which can be estimated by standard moment estimators:

$$\begin{aligned} \hat{\mu}_j &= \sum_{i=1}^n X_i \mathbf{1}_{[Y_i=j]} / \sum_{i=1}^n \mathbf{1}_{[Y_i=j]} = \frac{1}{n_j} \sum_{i; Y_i = j} X_i, \qquad \text{where} \quad n_j = \#\{i; Y_i = j\}, \\ \hat{\boldsymbol{\Sigma}} &= \frac{1}{n - J} \sum_{j=0}^{J-1} \sum_{i=1}^n (X_i - \hat{\mu}_j)(X_i - \hat{\mu}_j)^\intercal \mathbf{1}_{[Y_i=j]}, \quad \text{and} \\ \hat{\boldsymbol{\Sigma}}_j &= \frac{1}{n_j - 1} \sum_{i=1}^n (X_i - \hat{\mu}_j)(X_i - \hat{\mu}_j)^\intercal \mathbf{1}_{[Y_i=j]} \end{aligned} \tag{6.5}$$

Moreover, we need to specify the (a-priori) distribution for Y: quite often, one takes $\hat{p}_j = n^{-1} \sum_{i=1}^n \mathbf{1}_{[Y_i=j]} = n_j / n$. Using these parameter estimates, we obtain a classifier via

formula (6.4) and (6.2):

$$\begin{aligned}\widehat{\mathcal{C}}_{lin.discr.}(x) &= \arg\max_{0\le j\le J-1} \hat{\delta}_j(x),\\ \hat{\delta}_j(x) &= x^\intercal \hat{\boldsymbol{\Sigma}}^{-1}\hat{\mu}_j - \hat{\mu}_j{}^\intercal \hat{\boldsymbol{\Sigma}}^{-1}\hat{\mu}_j/2 + \log(\hat{p}_j) =\\ &= (x-\hat{\mu}_j/2)^\intercal \hat{\boldsymbol{\Sigma}}^{-1}\hat{\mu}_j + \log(\hat{p}_j).\end{aligned}$$

This classifier is called "linear discriminant classifier" because the estimated decision functions $\hat{\delta}_j(\cdot)$ are **linear** in the predictor variables x and since the regions are determined by $\hat{\delta}_j(x) - \hat{\delta}_{j'}(x) \ge 0$, the decision boundaries are hyperplanes (i.e., lines for $p = 2$), $x^\intercal \hat{\boldsymbol{\Sigma}}^{-1}(\hat{\mu}_j - \hat{\mu}_{j'}) + c_{j,j'} = 0$.

6.3.2 Quadratic discriminant analysis

The model underlying linear discriminant analysis assumes equal covariances for all the groups, see formula (6.3). More generally, we sometimes assume

$$\begin{aligned}(X \mid Y = j) &\sim \mathcal{N}_p(\mu_j, \boldsymbol{\Sigma}_j),\\ \mathrm{P}[Y = j] &= p_j, \quad \sum_{j=0}^{J-1} p_j = 1; \qquad j = 0, 1, \ldots, J-1.\end{aligned}$$

with non-equal covariances for all the groups $j \in \{0, 1, \ldots, J-1\}$. Analogously to linear discriminant analysis, we then obtain discriminant functions which are **quadratic** in the predictor variables x,

$$\hat{\delta}_j(x) = -\log(\det(\hat{\boldsymbol{\Sigma}}_j))/2 - (x-\hat{\mu}_j)^\intercal \hat{\boldsymbol{\Sigma}}_j^{-1}(x-\hat{\mu}_j)/2 + \log(\hat{p}_j).$$

Such quadratic discriminant classifiers are more flexible and general than their linear "cousins": but the price to pay for this flexibility are $J \cdot p(p+1)/2$ parameters for all covariance matrices $\boldsymbol{\Sigma}_j$ $(j = 0, 1, \ldots, J-1)$ instead of $p(p+1)/2$ for one $\boldsymbol{\Sigma}$ in linear discriminant analysis. Particularly when p is large, quadratic discriminant analysis typically overfits (too large variability).

6.4 The view of logistic regression

As we have seen in (6.1), all we need to know for a good classifier is a good estimator for the conditional probabilities $\pi_j(\cdot)$.

6.4.1 Binary classification

For the case with binary response $Y \in \{0, 1\}$, the conditional probability function

$$\pi(x) = \mathbb{P}[Y = 1 \mid X = x]$$

provides the full information about the conditional distribution of Y given X (since $\mathbb{P}[Y = 0 \mid X = x] = 1 - \pi(x)$). The logistic model for $\pi(\cdot)$ in general is

$$\begin{aligned}&\log\left(\frac{\pi(x)}{1-\pi(x)}\right) = g(x),\\ &g : \mathbb{R}^p \to \mathbb{R}.\end{aligned} \tag{6.6}$$

Note that the so-called logistic transform $\pi \mapsto \log(\pi/(1-\pi))$ maps the interval $(0, 1)$ to the real line $\mathbb{R}$. Models for real-valued functions can thus be used for $g(\cdot)$.

Linear logistic regression

In analogy to linear regression in chapter 1, a popular and simple model for $g(\cdot)$ is

$$g(x) = \sum_{j=1}^{p} \beta_j x_j. \tag{6.7}$$

The model in (6.6) with $g(\cdot)$ from (6.7) is called linear logistic regression.

Fitting of the parameters is usually done by maximum-likelihood. For fixed predictors x_i, the probability structure of the response variables is

$$Y_1, \ldots, Y_n \text{ independent}, \; Y_i \sim \text{ Bernoulli}(\pi(x_i)).$$

The likelihood is thus

$$L(\beta; (x_1, Y_1), \ldots, (x_n, Y_n)) = \prod_{i=1}^{n} \pi(x_i)^{Y_i} (1 - \pi(x_i))^{1-Y_i},$$

and the negative log-likelihood becomes

$$\begin{aligned} -\ell(\beta; (x_1, Y_1), \ldots, (x_n, Y_n)) &= -\sum_{i=1}^{n} \left(Y_i \log(\pi(x_i)) + (1 - Y_i) \log(1 - \pi(x_i)) \right) \\ &= -\sum_{i=1}^{n} \left(Y_i \sum_{j=1}^{p} \beta_j x_{ij} - \log(\exp(\sum_{j=1}^{p} \beta_j x_{ij}) + 1) \right). \end{aligned}$$

Minimization of the negative log-likelihood is a **nonlinear** problem. It is usually solved numerically by versions of Newton's gradient descent method which then yield the maximal likelihood estimate $\hat{\beta}_{p \times 1}$.

Asymptotic inference and an example

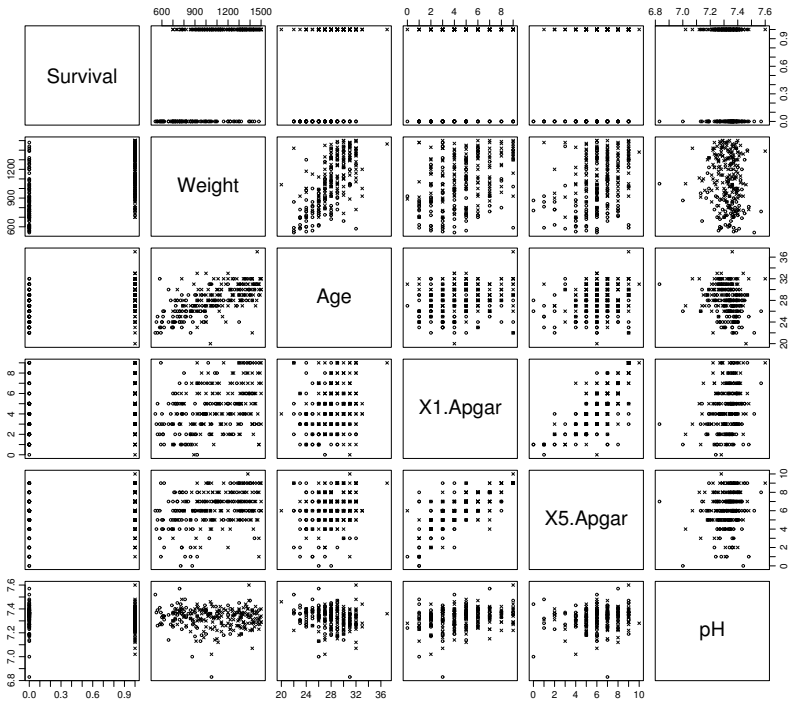

Figure 6.1: *Survival of 247 prenatal babies as a function of age (in weeks), weight (in grams) and 3 other clinical variables.*

Based on classical theory for maximum likelihood estimation, the distribution of the estimated coefficients $\hat{\beta}$ can be derived from an asymptotic point of view where the sample size $n \to \infty$. The output in R then yields a summary of the estimated coefficients, of their estimated standard errors $\widehat{s.e.}(\hat{\beta}_j)$, of the individual t-test statistics $\hat{\beta}_j / \widehat{s.e.}(\hat{\beta}_j)$ (which are asymptotically $\mathcal{N}(0,1)$ distributed under the null-hypothesis $H_{0,j} : \beta_j = 0$) and the P-values for the individual t-tests for the null-hypotheses $H_{0,j} = 0$ $(j = 1, \ldots, p)$.

As an example, we consider a dataset about survival of prenatal babies (represented by the response variable $Y \in \{0, 1\}$) as a function of age (in weeks), weight (in grams) and 3 other clinical variables (see Fig. 6.1). Fitting a linear logistic regression model can be done in R using the function `glm`:

```
> d.baby ← read.table("http://stat.ethz.ch/Teaching/Datasets/baby.dat",header=T)

> fit ← glm(Survival ~ ., data = d.baby, family= "binomial")
> summary(fit)

Call:
glm(formula = Survival ~ ., family = "binomial", data = d.baby)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3994  -0.7393   0.4220   0.7833   1.9445

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.0933685 14.3053767  -0.216   0.8288
Weight       0.0037341  0.0008468   4.410 1.03e-05 ***
Age          0.1588001  0.0761061   2.087   0.0369 *
X1.Apgar     0.1159864  0.1108339   1.046   0.2953
X5.Apgar     0.0611499  0.1202222   0.509   0.6110
pH          -0.7380214  1.8964578  -0.389   0.6972
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 319.28  on 246  degrees of freedom
Residual deviance: 236.14  on 241  degrees of freedom
AIC: 248.14

Number of Fisher Scoring iterations: 4
```

As seen from the P-values for the individual hypotheses $H_{0,j} : \beta_j = 0$, the predictor variables weight and age, which are strongly correlated, turn out to be significant for describing whether a prenatal baby will survive.
For classification, we can extract the probabilities with the function `predict`:

```
predict(fit, type="response")
```

which yields the estimated probabilities $\hat{\pi}(x_i)$, $i = 1, \ldots, n$. Thus, the average in-sample classification accuracy $\frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{[Y_i = \widehat{\mathcal{C}}(x_i)]}$ is given by

```
mean((predict(fit, type = "response") > 0.5) == d.baby$Survival) ,
```

which turns out to be `0.789`. Remember that such an in-sample estimate is over-optimistic for measuring the true generalization performance.

Linear logistic regression or LDA?

In linear logistic regression, the model for the log-odds (the logit-transform)

$$\log\left(\frac{\pi(x)}{1-\pi(x)}\right) = \sum_{j=1}^{p} \beta_j x_j$$

is *linear* in the predictors. But also the log-odds from LDA in model (6.4) yield a linear model in the predictor variables which is a consequence of the Gaussian assumption. Thus, the two methods are quite similar. In our example, the LDAcoefficients of the five predictors are 0.00309, 0.124, 0.0685, 0.067, -0.763, and the in-sample accuracy is slightly smaller, at `0.785`.

The linear logistic regression does not make any assumptions on the predictor variables such a multivariate Gaussianity; but the restriction comes in by requiring a linear log-odds function. Logistic regression can also be used when having factor variables (e.g. $x \in \{0, 1, \ldots, K\}$) as predictors.

While LDA does make a Gaussian assumption for the predictors, it can also be used as a "linear technique" in case of non-Gaussian predictors (even with factors). Empirically, LDA and linear logistic regression yield similar answers, even for non-Gaussian predictors, with respect to classification accuracy.

6.4.2 Multiclass case, $J > 2$

Logistic regression cannot be directly applied to the multiclass case with $Y \in \{0, 1, \ldots J-1\}$ and $J > 2$. But we can always encode a multiclass problem with J classes as J binary class problems by using

$$Y_i^{(j)} = \begin{cases} 1 & \text{if } Y_i = j, \\ 0 & \text{otherwise.} \end{cases}$$

This means that we consider class j against all remaining classes.

We can then run (linear or "general") logistic regressions

$$\log\left(\frac{\pi_j(x)}{1-\pi_j(x)}\right) = g_j(x) = \sum_{r=1}^{p} \beta_r^{(j)} x_r$$

yielding estimates

$$\hat{\pi}_j(x) = \frac{\exp(\sum_{r=1}^{p} \hat{\beta}_r^{(j)})}{1 + \exp(\sum_{r=1}^{p} \hat{\beta}_r^{(j)})}. \tag{6.8}$$

The estimates $\hat{\pi}_j(\cdot)$ will not sum up to one: but a normalization will do the job,

$$\tilde{\pi}_j(x) = \frac{\hat{\pi}_j(x)}{\sum_{j=0}^{J-1} \hat{\pi}_j(x)}.$$

Note that for this (parametric, linear) case, the problem can be formulated more nicely, but slightly differently, using the *multinomial* distribution, and solved by maximum likelihood, very similarly to the linear logistic ($J = 2$) case[1]. This is implemented in R 's `multinom()` function (standard package `nnet`).

[1]The Likelihood is $L = \pi_0^{n_0}\pi_1^{n_1} \cdot \pi_{J-1}^{n_{J-1}}$, the log-likelihood therefore $l = \sum_{j=0}^{J-1} n_j \log \pi_j$, where the constraint $\sum_j \pi_j \equiv 1$ has to be fulfilled.

A related approach works with a *reference class*, say, $j = 0$, and instead of "one against all" models "*everyone against the reference*",

$$\log\left(\pi_j(x)/\pi_0(x)\right) = g_j(x), \quad \text{for } j = 1, \ldots, J-1.$$

Other ways are also possible to reduce a J class problem into several binary class problems. Instead of modelling class j against the rest, we can also model class j against another class k for all pairs (j, k) with $j \neq k$. This will require fitting $\binom{J}{2}$ logistic models (involving $\binom{J}{2} \cdot p$ estimated parameters in the linear case), instead of J models in the one against the rest approach. We should keep in mind that the models are different: in the one against the rest approach, the coefficients in (6.8) describe the effect of the predictors for distinguishing class j from all others, and in a pairwise approach, we would model the distinction between two classes.

Note that for the particular situation of **ordered classes**, one can use a more simple "proportional odds" model

$$\text{logit}(P[Y \leq k \mid x]) = \alpha_k + g(x), \qquad k = 0, 1, \ldots, J-1, \text{ with } \alpha_0 \leq \alpha_1 \leq \ldots \leq \alpha_{J-1}. \quad (6.9)$$