Model-based classification (LDA, QDA, etc.)

Matias Salibian

29 October, 2018

Model-based classification

Consider the following setting for a classification problem: there are p explanatory variables (features), collected in a vector $\mathbf{X} \in \mathbb{R}^p$, along with a categorical variable G that takes one of K possible values in the set $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$. Given a training set $(g_1, \mathbf{x}_1)$, $(g_2, \mathbf{x}_2)$, ..., $(g_n, \mathbf{x}_n)$ we are interested in constructing a classifier (i.e. a function $\hat{g} : \mathbf{R}^p \to \mathcal{C}$) with good prediction properties.

As we discussed in class, given a point $\mathbf{x}_0$, the optimal classification (with respect to the 0-1 loss) picks the class c_j with the highest conditional probability of occurring given $\mathbf{x}_0$. In symbols:

$$\hat{g}(\mathbf{x}_0) \, = \, \arg\max_{c_\ell \in \mathcal{C}} \, P\left(G = c_\ell | \, \mathbf{X} = \mathbf{x}_0\right) \, .$$

If we know (can model reasonably) and can estimate all the conditional probabilities $P(G = c_\ell | \mathbf{X} = \mathbf{x}_0)$ for all possible values of the features $\mathbf{x}_0 \in \mathbb{R}^p$ then we can estimate the optimal (with respect to the 0-1 loss) classifier. One way to model (and estimate) the above conditional probabilities is to do it indirectly by modeling the distribution of the vector of features $\mathbf{X}$ for each class, in symbols, the (conditional) distribution $\mathbf{X} | G = c_\ell$. If $f(\mathbf{x} | G = c_j)$ is the conditional density (or pmf) of the vector of explanatory variables given that the response is c_j, then we always have:

$$P\left(G = c_j | \, \mathbf{X} = \mathbf{x}_0\right) \, = \, \frac{f(\, \mathbf{x}_0 \, | \, G = c_j) P(G = c_j)}{\sum_{\ell=1}^{K} \, f(\, \mathbf{x}_0 | G = c_\ell) P(G = c_\ell)} \, .$$

Hence, to find the value c_i with the largest $P(G = c_i | \mathbf{X} = \mathbf{x}_0)$ we only need to compute the numerators in the right hand side above for each $c_j \in \mathcal{C}$ and select the class with the largest value. We now look at a simple example of the above general approach.

An example: LDA

One of the simplest models for $\mathbf{X} | G = c_\ell$ is a (multivariate) normal (Gaussian) distribution with mean $\mu_\ell \in \mathbb{R}^p$ and covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{p \times p}$ (the same for all $c_\ell \in \mathcal{C}$). In symbols:

$$\mathbf{X} | G = c_i \, \sim \, \mathcal{N}\left(\mu_\ell, \mathbf{\Sigma}\right) \, .$$

The density function for a $\mathcal{N}\left(\mu, \mathbf{\Sigma}\right)$ distribution is given by

$$f(\mathbf{x}) \, = \, \frac{1}{(2\pi)^p} \, \frac{1}{|\mathbf{\Sigma}|^{p/2}} \, \exp\left((\mathbf{x} - \mu)^\top \, \mathbf{\Sigma}^{-1} \, (\mathbf{x} - \mu) \right) \, .$$

In order to use all this machinery we need to estimate the density functions for all the $\mathcal{N}(\mu_j, \mathbf{\Sigma})$ distributions, and then, given a point $\mathbf{x}_0$, evaluate them and choose the class with the largest value of $f(\mathbf{x}_0 \,|\, G = c_j)P(G = c_j)$.

To estimate the above densities we need to estimate the following parameters and probabilities:

- $\mu_i, i = 1, \ldots, K$;
- $\mathbf{\Sigma}$; and
- $P(G = c_j), j = 1, \ldots, K$.

The MLE estimators for the mean vectors are the sample means of the features in each class in the training set, and the common covariance matrix (across classes) can be estimated by taking a weighted average of the sample covariance matrix in each class using the data in the training set. Finally, the probabilities of each class can be estimated with the observed frequency of each class in the training set. In symbols we have

- For class $c_i \in \mathcal{C}$ we have
$$\hat{\mu}_i = \frac{1}{n_i} \sum_{j \in \mathcal{N}_i} \mathbf{x}_j \,,$$
where $\mathcal{N}_i$ is the set of indices of observations in the training set with $g = c_i$, and n_i is the cardinal of that set.
- The common matrix $\mathbf{\Sigma}$ can be estimated with
$$\hat{\mathbf{\Sigma}} = \frac{1}{n - K} \sum_{\ell=1}^{K} (n_\ell - 1)\hat{\mathbf{\Sigma}}_\ell \,,$$
where
$$\hat{\mathbf{\Sigma}}_\ell = \frac{1}{n_\ell - 1} \sum_{j \in \mathcal{N}_\ell} (\mathbf{x}_j - \bar{\mathbf{x}}_\ell)(\mathbf{x}_j - \bar{\mathbf{x}}_\ell)^\top \,.$$
and
$$\bar{\mathbf{x}}_\ell = \frac{1}{n_\ell} \sum_{i \in \mathcal{N}_\ell} \mathbf{x}_i \,.$$

A numerical example

Consider the example used in class, with 2 classes and a 2-dimensional vector of explanatory variables. We first load the data

```
data(vaso, package='robustbase')
```

We will compute an approximation to the optimal predictor based on the Gaussian model behind LDA (assuming a Gaussian distribution for the vector of features in each class, with a common covariance matrix). The goal is to classify a new observation at $\mathbf{x}_0 = (\text{Volume}, \text{Rate})^\top = (1.3, 1.9)^\top$.

For this, we first estimate the parameters of the distributions of $\mathbf{X}|G = 0$ and $\mathbf{X}|G = 1$. These are the two vectors of means: $\mu_0 \in \mathbb{R}^2$, $\mu_1 \in \mathbb{R}^2$, and the common covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{2\times 2}$. Since we will have to perform a couple of operations on each subgroup (compute their sample means and covariance

matrices), we first create two new objects, one per class. This is not necessary, but it results in shorter and easier to read code. We call each subgrup d0 (for class "0") and d1 (for class "1"):

```
d0 <- subset(vaso, Y == 0, select = c('Volume', 'Rate'))
d1 <- subset(vaso, Y == 1, select = c('Volume', 'Rate'))
```

The vectors of means for each class are

```
( mu0 <- colMeans( d0 ) )
```

```
##   Volume     Rate
## 1.034211 1.397895
```

```
( mu1 <- colMeans( d1 ) )
```

```
##  Volume    Rate
## 1.67000 1.96425
```

To estimate Σ we first compute the covariance matrix of each group:

```
( sigma0 <- cov( d0 ) )
```

```
##             Volume       Rate
## Volume  0.2341813 -0.2158406
## Rate   -0.2158406  0.6228842
```

```
( sigma1 <- cov( d1 ) )
```

```
##             Volume       Rate
## Volume  0.9006316 -0.5839316
## Rate   -0.5839316  0.7962797
```

and then use their weighted average, using the number of observations in each group:

```
(sigma.hat <- ( (nrow(d0) - 1 ) * sigma0
              + (nrow(d1) - 1) * sigma1 ) / (nrow(vaso) - 2) )
```

```
##             Volume       Rate
## Volume  0.5764125 -0.4048603
## Rate   -0.4048603  0.7119251
```

We now estimate the probabilities $P(G = 0)$ and $P(G = 1)$:

```
( pi0 <- with(vaso, mean(Y == 0)) )
```

```
## [1] 0.4871795
```

```
( pi1 <- with(vaso, mean(Y == 1)) )
```

```
## [1] 0.5128205
```

We are now ready to compute estimates of $f(\mathbf{x}_0 \,|\, G = 0)P(G = 0)$ and $f(\mathbf{x}_0 \,|\, G = 1)P(G = 1)$. We do it first by hand, then using pre-coded multivariate Gaussian densities in R and finally compare the results

using the function `lda` in package `MASS`, as done in the lecture notes.

By hand

To do it by hand we use the formula of the density of a multivariate normal distribution, as above. We first compute the exponents in the exponential function `cp0` and `cp1` below, and then the value of the density function (without the $\sqrt{2\,\pi}$ that is common to both terms):

```
x0 <- c(1.9, 1.3)
p <- 2
cp0 <- crossprod( x0 - mu0, solve(sigma.hat, x0 - mu0))
dens0 <- det(sigma.hat)^(-p/2) * exp( - cp0 / 2 )
cp1 <- crossprod( x0 - mu1, solve(sigma.hat, x0 - mu1))
dens1 <- det(sigma.hat)^(-p/2) * exp( - cp1 / 2 )
```

We now compute $f(\mathbf{x}_0 \,|\, G = 0)P(G = 0)$:

```
( pp0 <- dens0 * pi0 )
```

```
##             [,1]
## [1,] 0.7609567
```

and $f(\mathbf{x}_0 \,|\, G = 1)P(G = 1)$:

```
( pp1 <- dens1 * pi1 )
```

```
##            [,1]
## [1,] 1.478977
```

We see that the (estimated) optimal classifier would classify a point at $\mathbf{x}_0 = (\text{Volume, Rate})^\top = (1.3, 1.9)^\top$ as belonging to class 1. The corresponding conditional probabilities for each class are:

```
c( posterior0 = pp0 / (pp0 + pp1), posterior1 = pp1 / (pp0 + pp1) )
```

```
## posterior0 posterior1
##  0.3397229  0.6602771
```

Using a built-in function to compute multivariate Gaussian densities

Using the estimated parameters $\mu_0 \in \mathbb{R}^2$, $\mu_1 \in \mathbb{R}^2$, and the common covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{2\times 2}$ as before, we can compute the corresponding values of Gaussian density functions using the function `dmvnorm` in package `mvtnorm` (see its help page for more details):

```
dens0 <- mvtnorm::dmvnorm(x0, mean=mu0, sigma=sigma.hat)
dens1 <- mvtnorm::dmvnorm(x0, mean=mu1, sigma=sigma.hat)
```

And now, the numerators of the conditional probabilities and the conditional probabilities are:

```
( pp0 <- dens0 * pi0 )
```

```
## [1] 0.06012361
```

```
( pp1 <- dens1 * pi1 )
```

```
## [1] 0.1168548
```

```
c( posterior0 = pp0 / (pp0 + pp1), posterior1 = pp1 / (pp0 + pp1) )
```

```
## posterior0 posterior1
##  0.3397229  0.6602771
```

Note: why do `pp0` and `pp1` take different values here, but the conditional probabilities are the same in the end?

Using `MASS::lda`

We now use the function `lda` in package `MASS` as in the lecture slides and notes:

```
library(MASS)
a.lda <- lda(Y ~ Volume + Rate, data=vaso)
names(x0) <- c('Volume', 'Rate')
predict(a.lda, newdata=data.frame(t(x0)))$posterior
```

```
##           0         1
## 1 0.3397229 0.6602771
```