Lecture 14: Multiclass Classification (II)

Wenbin Lu

Department of Statistics
North Carolina State University

Fall 2019

Outline

1. Nonlinear Discriminant Analysis
   - Quadratic discriminant analysis (QDA)
   - Regularized discriminant analysis (RDA)

How to Fit a Quadratic Boundary

There are two popular ways:

1. Use LDA in the enlarged space containing quadratic polynomials
   - If $d = 2$, fit LDA in five-dimensional space spanned by
$$\{X_1, X_2, X_1X_2, X_1^2, X_2^2\}.$$
2. Use Quadratic Discriminant Analysis (QDA)

In practice, these two methods often give similar results.

Elements of Statistical Learning ©Hastie, Tibshirani & Friedman 2001 Chapter 4

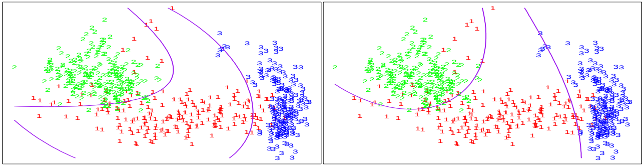


Figure 4.6: *Two methods for fitting quadratic boundaries. The left plot shows the quadratic decision boundaries for the data in Figure 4.1 (obtained using LDA in the five-dimensional space* $x_1, x_2, x_{12}, x_1^2, x_2^2$*). The right plot shows the quadratic decision boundaries found by QDA. The differences are small, as is usually the case.*

Quadratic Linear Discriminant Analysis

Model Setup: recall that

- $\pi_k = P(Y = k)$ for $k = 1, \cdots, K$
- $g_k(\mathbf{x})$ is the class-conditional densities of X in class k.

QDA Model Assumptions:

- Assume each class density is multivariate Gaussian, i.e.,

$$\mathbf{X}|Y = k \sim N(\boldsymbol{\mu}_k, \Sigma_k), \quad k = 1, \cdots, K$$

- Allow unequal covariances across classes.

Multivariate Normal Distribution

If $\mathbf{X} = (X_1, \cdots, X_d) \sim N_d(\boldsymbol{\mu}, \Sigma)$, then its density has the form

$$f(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = (2\pi)^{-d/2}|\Sigma|^{-1/2} \exp\{-(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})/2\},$$

where $\boldsymbol{\mu}$ is the mean, and Σ is the covariance matrix.

- Contour of constant density for $N_d(\boldsymbol{\mu}, \Sigma)$ are ellipsoids defined by $\mathbf{x}$ such that
$$(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}) = c^2.$$
- These ellipsoids are centered at $\boldsymbol{\mu}$ and have axes $\pm c\sqrt{\lambda_j}\mathbf{e}_j$, where
$$\Sigma\mathbf{e}_j = \lambda_j\mathbf{e}_j, \quad j = 1, \cdots, d.$$
Here $(\lambda_j, \mathbf{e}_j), j = 1, \cdots, p$ are the eigenvalue-eigenvector pairs of Σ.

Property of Multivariate Normal Distribution

Assume $\mathbf{X} \sim N_d(\boldsymbol{\mu}, \Sigma)$. Then

- Let χ_d^2 denote the chi-square distribution with d degrees of freedom.

$$(\mathbf{X} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{X} - \boldsymbol{\mu}) \sim \chi_d^2.$$

- The $N_d(\boldsymbol{\mu}, \Sigma)$ distribution assigns probability $(1 - \alpha)$ to the solid ellipsoid

$$\{\mathbf{x} : (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq \chi_d^2(\alpha)\},$$

  where $\chi_d^2(\alpha)$ denotes the upper $(100\alpha\%)$th percentile of the χ_d^2 distribution.

QDA Decision Rule

Under Gaussian assumption, the log-ratio of class k ad class l is:

$$\log \frac{\Pr(Y=k|\mathbf{X}=\mathbf{x})}{\Pr(Y=l|\mathbf{X}=\mathbf{x})} = \log \frac{\pi_k \phi(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k)}{\pi_j \phi(\mathbf{x}; \boldsymbol{\mu}_l, \Sigma_l)} = f_k(\mathbf{x}) - f_l(\mathbf{x}),$$

where the discriminant functions (score) f_k is given by

$$f_k(\mathbf{x}) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) + \log \pi_k, \quad k = 1, \cdots, K.$$

The decision boundary between each pair of classes k and l is

$$\{\mathbf{x} : f_k(\mathbf{x}) = f_l(\mathbf{x})\}.$$

The decision rule is

$$f(\mathbf{x}) = \text{argmax}_{k=1,\ldots,K} f_k(\mathbf{x}).$$

More About QDA Decision Boundary

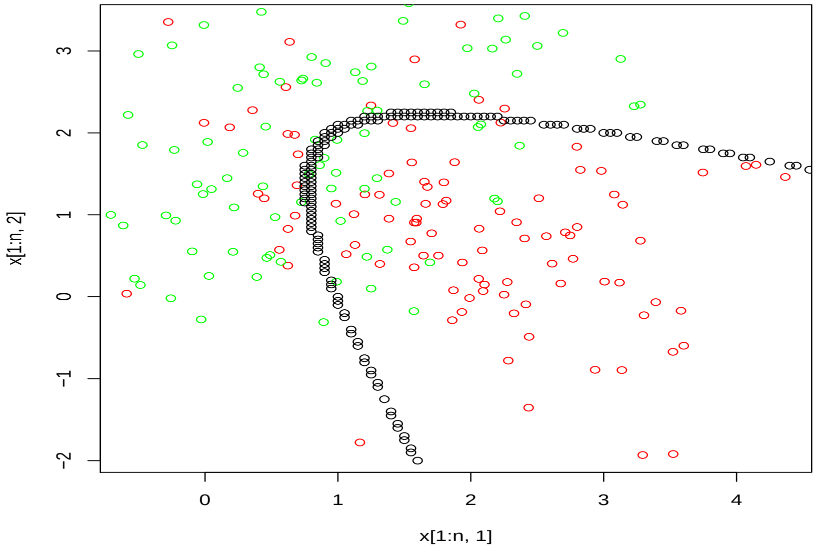
The discrimination function are quadratic in $\mathbf{x}$,

$$f_k(\mathbf{x}) = \mathbf{x}^T W_k \mathbf{x} + \boldsymbol{\beta}_{1k}^T \mathbf{x} + \beta_{0k}, \quad k = 1, \cdots, K.$$

The decision boundary between class k and class l is also quadratic

$$\{\mathbf{x} : \mathbf{x}^T (W_k - W_l)\mathbf{x} + (\boldsymbol{\beta}_{1k} - \boldsymbol{\beta}_{1l})^T \mathbf{x} + (\beta_{0k} - \beta_{0l}) = 0\}.$$

- QDA needs to estimate more parameters than LDA, and the difference is large when d is large.
    - Fitting LDA needs to estimate $(K-1) \times (d+1)$ parameters
    - Fitting QDA needs to estimate $(K-1) \times (d(d+3)/2+1)$ parameters


x[1:n, 2]
x[1:n, 1]
0
1
2
3
4
−2
−1
0
1
2
3

Interpretation of QDA

The discriminant function f_k depends on three factors:

- the generalized variance $|\Sigma_k|$
- the prior probability π_k
- the squared Mahalanobis distance from $\mathbf{x}$ to the population mean $\boldsymbol{\mu}_k$.

Here, a different distance function, with a different orientation and size of the constant-distance ellipsoid, is used for each class.

Parameter Estimation in QDA

In practice, we estimate the parameters from the training data

- $\hat{\pi}_k = n_k/n$, where n_k is the number of observations in class k for $k = 1, \cdots, K$.
- $\widehat{\boldsymbol{\mu}}_k = \sum_{Y_i=k} \mathbf{x}_i / n_k$ for $k = 1, \cdots, K$.
- The within-class sample covariance

$$\hat{\Sigma}_k = \frac{1}{n_k - 1} \sum_{Y_i=k} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)^T.$$

The implementation of QDA is computationally intensive, since

- we need to conduct matrix inversion multiple times: compute $\hat{\Sigma}_k$ for $k = 1, \cdots, K$.

Speed Up QDA Computation

Idea: Diagonalizing $\hat{\Sigma}_k$ (with eigen-decomposition)

$$\hat{\Sigma}_k = U_k D_k U_k^T,$$

where U_k orthonormal and D_k diagonal with positive eigenvalues.

- $$\begin{aligned} (\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) &= [U_k^T (\mathbf{x} - \boldsymbol{\mu}_k)]^T D_k^{-1} [U_k^T (\mathbf{x} - \boldsymbol{\mu}_k)] \\ &= [D_k^{-\frac{1}{2}} U_k^T (\mathbf{x} - \boldsymbol{\mu}_k)]^T [D_k^{-\frac{1}{2}} [U_k^T (\mathbf{x} - \boldsymbol{\mu}_k)] \end{aligned}$$
- $\log |\hat{\Sigma}_k| = \sum_{l=1}^K \log d_{kl}$

R code for QDA Fitting (I)

There are two ways to call the function "qda". The first way is to use a formula and an optional data frame.

```
library(MASS)
qda(formula, data, subset)
```

Arguments:

- *formula*: the form "groups $\sim x_1 + x_2 + \ldots$", where the response is the grouping factor and the right hand side specifies the (non-factor) discriminators.
- *data*: data frame from which variables specified
- *subset*: An index vector specifying the cases to be used in the training sample.

Output:

- an object of class "qda" with multiple components

R code for QDA Fitting (II)

The second way is to use a matrix and group factor as the first two arguments.

```
library(MASS)
qda(x, grouping, prior = proportions, CV = FALSE)
```

Arguments:

- *x*: a matrix or data frame or Matrix containing predictors.
- *grouping*: a factor specifying the class for each observation.
- *prior*: the prior probabilities of class membership. If unspecified, the class proportions for the training set are used.

Output:

- If CV = TRUE, the return value is a list with components "class" (the MAP classification, a factor) and "posterior" (posterior probabilities for the classes).

R code for QDA Prediction

We use the "predict" or "predict.qda" function to classify multivariate observations with qda

```
predict(object, newdata, ...)
```

Arguments:

- *object*: object of class "qda"
- *newdata*: data frame of cases to be classified or, if "object" has a formula, a data frame with columns of the same names as the variables used.

Output:

- a list with the components "class" (the MAP classification, a factor) and "posterior" (posterior probabilities for the classes)

Illustration 1

```
Iris <- data.frame(rbind(iris3[,,1], iris3[,,2],
   iris3[,,3]), Sp = rep(c("s","c","v"), rep(50,3)))
train <- sample(1:150, 75)
table(Iris$Sp[train])
z <- qda(Sp ~ ., Iris, prior = c(1,1,1)/3, subset = train)
ypred <- predict(z, Iris[-train, ])$class
ytest <- Iris$Sp[-train]
testerr <- mean(ypred!=ytest)
```

Illustration 2

```
tr <- sample(1:50, 25)
train <- rbind(iris3[tr,,1], iris3[tr,,2], iris3[tr,,3])
test <- rbind(iris3[-tr,,1], iris3[-tr,,2], iris3[-tr,,3])
cl <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
z <- qda(train, cl)
trainerr <- mean(predict(z,train)$class!=cl)
testerr <- mean(predict(z,test)$class!=cl)
```

Choice of LDA between QDA

Both LDA and QDA perform well on real classification problems.

- In STATLOG project (Michie et al. 1994), the LDA was among top 3 classifiers for 7 datasets; the QDA among top 3 for 4 datasets (totally 22 datasets)

Regularized Discriminant Analysis (RDA)

Friedman (1989) proposed a comprise between QDA and LDA:

- shrinking the separate covariances of QDA toward a common covariance in LDA.

The regularized covariance matrices are

$$\hat{\Sigma}_k(\alpha) = \alpha\hat{\Sigma}_k + (1-\alpha)\hat{\Sigma}, \quad \hat{\Sigma} \text{ pooled sample covariance matrix}$$

- $\alpha \in [0, 1]$, a continuum of models (compromise) between LDA and QDA. What if α is close to 1 (or 0)?
- In practice, choose α with validation data or CV.

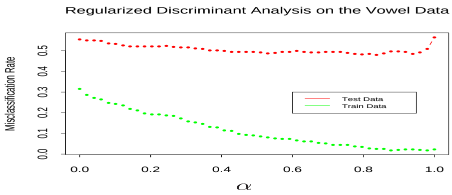


Figure 4.7: *Test and training errors for the vowel data, using regularized discriminant analysis with a series of values of* $\alpha \in [0,1]$. *The optimum for the test data occurs around* $\alpha = 0.9$, *close to quadratic discriminant analysis.*

More on Regularized DA

Other regularized methods:

- For LDA, we can shrink $\hat{\Sigma}$ toward the scalar covariance, $\gamma \in [0, 1]$

$$\hat{\Sigma}(\gamma) = \gamma\hat{\Sigma} + (1 - \gamma)\hat{\sigma}^2 I.$$

- A more general families of regularized QDA is indexed by (α, γ):

$$\hat{\Sigma}_k(\alpha, \gamma) = \alpha\hat{\Sigma}_k + (1 - \alpha)\gamma\hat{\Sigma} + (1 - \alpha)(1 - \gamma)\hat{\sigma}^2 I.$$