Outline

Supervised Learning: Parametric Methods

- Decision Theory
- Linear Discriminant Analysis
- Quadratic Discriminant Analysis
- Naíve Bayes
- Logistic Regression
- Evaluation Methodology

Given training data with K classes, assume a parametric form for $f_k(x)$, where for each class

$$X|Y=k \quad \sim \quad (\mu_k, \Sigma_k),$$

i.e. instead of assuming that every class has a different mean μ_k with the *same* covariance matrix Σ, we now allow each class to have its own covariance matrix.

Considering $-2\log P(Y=k|X=x)$ as before,

$$\begin{aligned}
-2\log P(Y=k|X=x) &\propto (x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) - 2\log(\pi_k) + \text{const}_k \\
&= \mu_k^T\Sigma_k^{-1}\mu_k - 2\mu_k^T\Sigma_k^{-1}x + x^T\Sigma_k^{-1}x \\
&\qquad -2\log(\pi_k) + \text{const}_k \\
&= a_k + b_k^T x + x^T c_k x
\end{aligned}$$

i.e. we find a *quadratic* function instead (the function const_k includes the term $\log(|\Sigma_k|)$

Again, by considering when we choose class k over k',

$$\begin{aligned} 0 &> a_k + b_k^T x + x^T c_k x - (a_{k'} + b_{k'}^T x + x^T c_{k'} x) \\ &= a_\star + b_\star^T x + x^T c_\star x \end{aligned}$$

we see that the Bayes Classifier partitions $\{x : \hat{Y}(x) = k\}$ are using quadratic surfaces.

The Bayes Classifer under these assumptions is more commonly known as the *Quadratic Discriminant Analysis* Classifier.

The exact form of the QDA classifier is given by

$$\hat{Y}_{qda}(x) = \operatorname*{arg\,min}_{k=1,\dots,K} \Big\{ (x-\hat{\mu}_k)^T \hat{\Sigma}_k{}^{-1}(x-\hat{\mu}_k) - 2\log(\hat{\pi}_k) + \log(|\hat{\Sigma}_k|) \Big\}$$

for each point $x \in \mathcal{X}$ where the plug-in estimate $\hat{\mu}_k$ is as before and $\hat{\Sigma}_k$ is (in contrast to LDA) estimated for each class $k = 1, \dots, K$ separately:

$$\hat{\Sigma}_k = \frac{1}{n_k} \sum_{j:Y_j=k} (X_j - \hat{\mu}_k)(X_j - \hat{\mu}_k)^T.$$

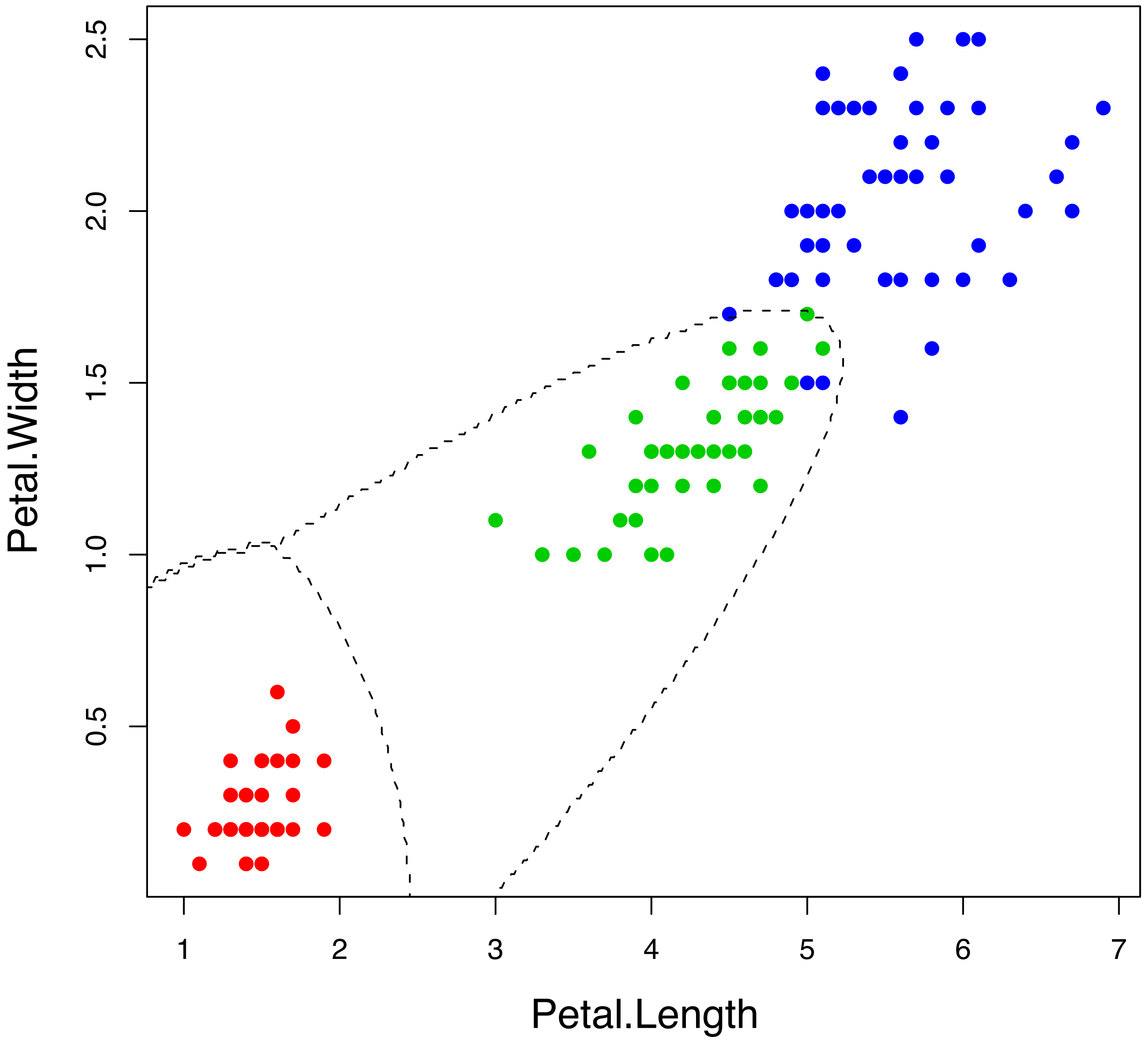
Computing and plotting the QDA (and LDA) boundaries.

```
##fit LDA
iris.lda <- lda(x=iris.data,grouping=ct)
iris.qda <- qda(x=iris.data,grouping=ct)

##create a grid for our plotting surface
x <- seq(-6,6,0.02)
y <- seq(-4,4,0.02)
z <- as.matrix(expand.grid(x,y),0)
m <- length(x)
n <- length(y)


iris.qdp <- predict(iris.qda,z)$class
contour(x,y,matrix(iris.qdp,m,n),
        levels=c(1.5,2.5), add=TRUE, d=FALSE, lty=2)
```

Iris example: QDA boundaries



LDA or QDA?

Having seen both LDA and QDA in action, it is natural to ask which is the “better” classifier.

It is obvious that if the covariances of different classes are very distinct, QDA will probably have an advantage over LDA.

As parametric models are only ever approximations to the real world, allowing more flexible decision boundaries (QDA) may seem like a good idea.

However, there is a price to pay in terms of increased variance.

Regularized Discriminant Analysis

In the case where data is scarce , to fit

- LDA, need to estimate $K \times p + p \times p$ parameters
- QDA, need to estimate $K \times p + K \times p \times p$ parameters.

Using LDA allows us to better estimate the covariance matrix Σ. Though QDA allows more flexible decision boundaries, the estimates of the K covariance matrices Σ_k are more variable.
RDA combines the strengths of both classifiers by regularizing each covariance matrix Σ_k in QDA to the single one Σ in LDA

$$\Sigma_k(\alpha) = \alpha\Sigma_k + (1 - \alpha)\Sigma \text{ for some } \alpha \in [0, 1].$$

This introduces a new parameter α and allows for a continuum of models between LDA and QDA to be used. Can be selected by Cross-Validation for example.