Models for classification

Linear and quadratic discriminant analysis

A quick recap

Last lecture, we did...

- **Models for classification**: Linear discriminant analysis
  Bayes' theorem for classification
  ⇝ We want to minimize the total probability of misclassification, which depends on class probabilities and multivariate distributions of the predictors.
  ⇝ We use the Bayes' theorem:

$$P(Y=k|X) = \frac{P(X|Y=k)P(Y=k)}{\sum_{l=1}^{K} P(X|Y=k)P(Y=l)}$$

  - $\pi_k = P(Y=k)$: **prior probability** of the membership in class k.
    These values are either known (determined by the proportions of samples in each class), or are unknown (all values of the priors are set to be equal).

A quick recap

Last lecture, we did...

- **Models for classification**: Linear discriminant analysis
  Bayes' theorem for classification
  ⇝ We want to minimize the total probability of misclassification, which depends on class probabilities and multivariate distributions of the predictors.
  ⇝ We use the Bayes' theorem:

$$P(Y=k|X) = \frac{P(X|Y=k)P(Y=k)}{\sum_{l=1}^{K} P(X|Y=k)P(Y=l)}$$

  - $f_k(x) = P(X|Y=k)$: **conditional probability** of the observing predictors X.
    Here we assume that the data are generated from a probability distribution (**multivariate normal distribution**), which then defines this quantity's mathematical form.

A quick recap

Last lecture, we did...

- **Models for classification**: Linear discriminant analysis
  Bayes' theorem for classification
  ⇝ We want to minimize the total probability of misclassification, which depends on class probabilities and multivariate distributions of the predictors.
  ⇝ We use the Bayes' theorem:

$$P(Y=k|X) = \frac{P(X|Y=k)P(Y=k)}{\sum_{l=1}^{K} P(X|Y=k)P(Y=l)}$$

  - $p_k(X) = P(Y=k|X)$: **posterior probability** that the sample X is a member of the class k.

A quick recap

Last lecture, we did...

- **Models for classification**: Linear discriminant analysis

  Bayes' theorem for classification

  ⇝ For a two-group classification problem, the rule that minimizes the total probability of misclassification would be to classify X into group k_1 if

  $$P(Y = k_1|X) > P(Y = k_2|X)$$

  and into class k_2 if the inequality is reversed.

  ⇝ Using Bayes, this rule directly translated to classifying X into k_1 if:

  $$P(Y = k_1)P(X|Y = k_1) > P(Y = k_2)P(X|Y = k_2)$$

  ⇝ This rule can be extended to $K > 2$: We classify X into group k_l if $P(Y = k_l)P[X|Y = k_l]$ has the largest value across all of the K classes.

A quick recap

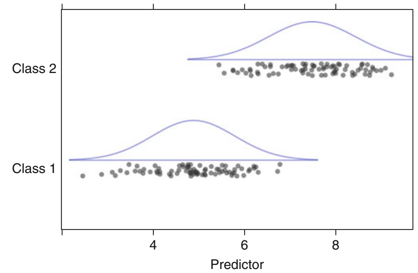
Last lecture, we did...

- **Models for classification**: Linear discriminant analysis

  Linear discriminant analysis for one predictor, $p = 1$

  ⇝ A single predictor is used to classify samples into two groups.

  The blue figures above each group represent the probability density function for a normal distribution determined by the class-specific means and variances.



A quick recap

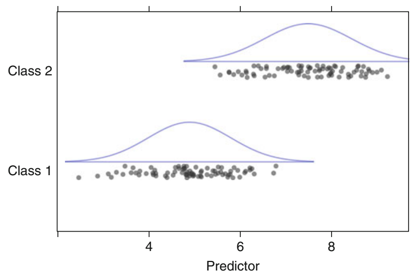
Last lecture, we did...

- **Models for classification**: Linear discriminant analysis

  Linear discriminant analysis for one predictor, $p = 1$

  ⇝ A single predictor is used to classify samples into two groups.

  A new sample is classified by finding its value on the x-axis, then determining the value for each of the PDFs for each class.



A quick recap

Last lecture, we did...

- **Models for classification**: Linear discriminant analysis

  Linear discriminant analysis for one predictor, $p > 1$

  ⇝ For classification, the number of predictors is almost always greater than one and can be extremely large.

  ⇝ We assume that the distribution of the predictors is multivariate normal:

  $\boldsymbol{\mu}_k$: the multidimensional mean vector
  $\boldsymbol{\Sigma}_k$: covariance matrix.

  ⇝ We further assume that the means of the groups are unique (i.e., a different $\boldsymbol{\mu}_k$ for each class) but the covariance matrices are identical across groups.

A quick recap

Last lecture, we did...

- **Models for classification**: Linear discriminant analysis

  Linear discriminant analysis for one predictor, $p > 1$

  ⇝ We can solve the classification problem the more general multi-class problem to find the linear discriminant function of the lth class:

$$\delta_k(x) = x^T \mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}^{-1}\mu_k + \log(\pi_k) \quad \text{is largest}$$

The theoretical means μ_k are estimated using the class specific means:

$$\hat{\mu}_k = \frac{1}{n_k}\sum_{i:y_i=k} x_i$$

A quick recap

Last lecture, we did...

- **Models for classification**: Linear discriminant analysis

  Linear discriminant analysis for one predictor, $p > 1$

  ⇝ We can solve the classification problem the more general multi-class problem to find the linear discriminant function of the lth class:

$$\delta_k(x) = x^T \mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}^{-1}\mu_k + \log(\pi_k) \quad \text{is largest}$$

The theoretical covariance matrix, $\mathbf{\Sigma}$, is estimated by:

$$\hat{\mathbf{\Sigma}} = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k}(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

A quick recap

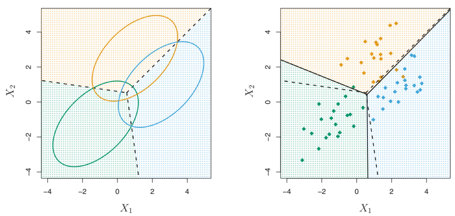
Last lecture, we did...

- **Models for classification**: Linear discriminant analysis

  Linear discriminant analysis for one predictor, $p > 1$

  ⇝ We can solve the classification problem the more general multi-class problem to find the linear discriminant function of the lth class:

$$\delta_k(x) = x^T \mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}^{-1}\mu_k + \log(\pi_k) \quad \text{is largest}$$



Today's goal

Today, we going to do...

- **Models for classification**: Discriminant analysis

  ⇝ Example on LDA in R

  ⇝ Quadratic discriminant analysis

Reading list

- Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*, Springer (2014)
- Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*, Springer (2017)

Linear discriminant analysis in R

We consider again the `Smarket` data (cfr. Lecture 15).

- We fit the model using only the observations before 2005.

```
> library(MASS)
> lda.fit=lda(Direction~Lag1+Lag2,data=Smarket,subset=train)
> lda.fit
Call:
lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)

Prior probabilities of groups:
 Down    Up
0.492 0.508
```

- The LDA output indicates $\hat{\pi}_1 = 0.492$ and $\hat{\pi}_2 = 0.508$.
  - ⤳ 49.2% of the training observations correspond to days during which the market went down.

Linear discriminant analysis in R

We consider again the `Smarket` data (cfr. Lecture 15).

- We fit the model using only the observations before 2005.

```
> library(MASS)
> lda.fit=lda(Direction~Lag1+Lag2,data=Smarket,subset=train)
> lda.fit
Call:
lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)

Group means:
        Lag1     Lag2
Down  0.0428  0.0339
Up   -0.0395 -0.0313
```

- The LDA output also provides the group means:
  - ⤳ These are the average of each predictor within each class, and are used by LDA as estimates of $\hat{\mu}_k$.
  - ⤳ These suggest that there is a tendency for the previous 2 days' returns to be negative on days when the market increases, and a tendency for the previous days' returns to be positive on days when the market declines.

Linear discriminant analysis in R

We consider again the `Smarket` data (cfr. Lecture 15).

- We fit the model using only the observations before 2005.

```
> library(MASS)
> lda.fit=lda(Direction~Lag1+Lag2,data=Smarket,subset=train)
> lda.fit
Call:
lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)

Coefficients of linear discriminants:
        LD1
Lag1 -0.642
Lag2 -0.514
```

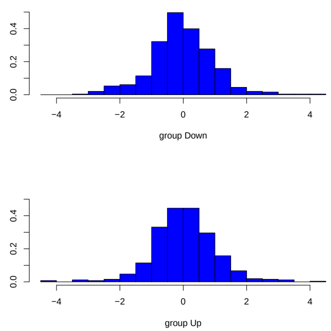
- The LDA output also provides the group means:
  - ⤳ The coefficients of linear discriminants output provides the linear combination of `Lag1` and `Lag2` that are used to form the LDA decision rule.
  - ⤳ In other words, these are the multipliers of the elements of $X = x$ in $\delta_k(x)$.
  - ⤳ If $-0.642\times$ `Lag1` $-0.514\times$ `Lag2` is large, then the LDA classifier will predict a market increase, and if it is small, then the LDA classifier will predict a market decline.

Linear discriminant analysis in R

We consider again the `Smarket` data (cfr. Lecture 15).

- We fit the model using only the observations before 2005.

```
> plot(lda.fit,col='blue')
```



- The `plot()` function produces plots of the linear discriminants.
- It is obtained by computing

  $-0.642\times$ `Lag1` $-0.514\times$ `Lag2`

  for each of the training observations.
- The two histograms are very similar!

Linear discriminant analysis in R

We consider again the `Smarket` data (cfr. Lecture 15).

- We fit the model using only the observations before 2005.

```
> # LDA prediction
> lda.pred=predict(lda.fit, Smarket.2005)
> names(lda.pred)
[1] "class"     "posterior" "x"
```

The LDA predictions give three outputs:

- `class` contains LDA's predictions about the movement of the market.
- `posterior`, is a matrix whose kth column contains the posterior probability that the corresponding observation belongs to the kth class computed using the Bayes theorem.
- `x` contains the linear discriminants.

Linear discriminant analysis in R

We consider again the `Smarket` data (cfr. Lecture 15).

- We fit the model using only the observations before 2005.

```
> # LDA prediction
> lda.class=lda.pred$class
> table(lda.class,Direction.2005)
         Direction.2005
lda.class Down  Up
     Down   35  35
     Up     76 106
> mean(lda.class==Direction.2005)
[1] 0.56
> sum(lda.pred$posterior[,1]>=.5)
[1] 70
> sum(lda.pred$posterior[,1]<.5)
[1] 182
```

The LDA and logistic regression predictions are almost identical.

- Applying a 50% threshold to the posterior probabilities allows us to recreate the predictions contained in `lda.pred$class`.

Quadratic discriminant analysis

LDA assumes that the observations within each class are drawn from a multivariate Gaussian distribution with a class-specific mean vector and a covariance matrix that is common to all K classes.

Quadratic discriminant analysis (QDA) provides an alternative approach.

- Like LDA, the QDA classifier results from assuming that the observations from each class are drawn from a Gaussian distribution, and plugging estimates for the parameters into Bayes' theorem in order to perform prediction.
- Unlike LDA, QDA assumes that **each class has its own covariance matrix**.
  - It assumes that an observation from the kth class is of the form $X \sim N(\mu_k, \mathbf{\Sigma}_k)$ $\mathbf{\Sigma}_k$ is a covariance matrix for the kth class.
  - Under this assumption, we cannot simplify in the Bayes' formula.

Quadratic discriminant analysis

LDA assumes that the observations within each class are drawn from a multivariate Gaussian distribution with a class-specific mean vector and a covariance matrix that is common to all K classes.

Quadratic discriminant analysis (QDA) provides an alternative approach.

- The Bayes' classifier assigns an observation to $X = x$ to the class for which

$$\delta_k(x) = -\frac{1}{2}(x-\mu_k)^T \mathbf{\Sigma}_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\mathbf{\Sigma}_k| + \log(\pi_k) =$$
$$= -\frac{1}{2}x^T\mathbf{\Sigma}_k^{-1}x + x^T\mathbf{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\log|\mathbf{\Sigma}_k| + \log(\pi_k)$$

is the largest

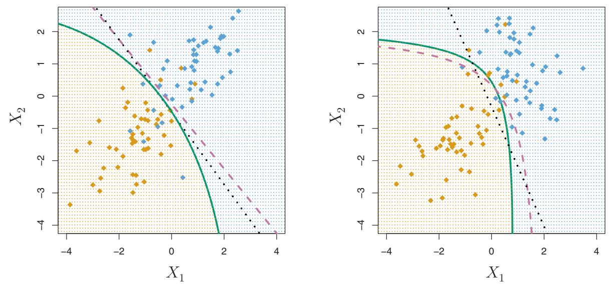
- QDA involves finding estimates for $\mathbf{\Sigma}_k, \mu_k$ and π_k and the assigning an observation to $X = x$ to the class for which this quantity is largest.
- Unlike LDA, the quantity x appears as a quadratic function. This is where QDA gets its name.

Quadratic discriminant analysis

Why does it matter whether or not we assume that the K classes share a common covariance matrix?

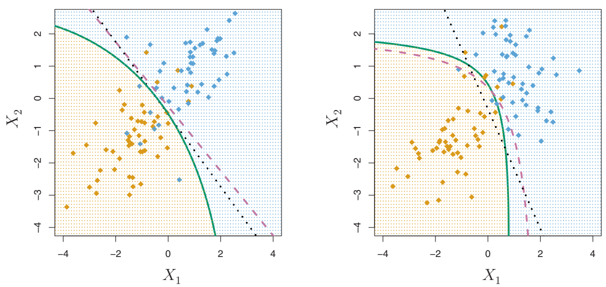
- When there are p predictors, then estimating a covariance matrix requires estimating $p(p+1)/2$ parameters.
- QDA estimates a separate covariance matrix for each class, for a total of $Kp(p+1)/2$ parameters.
- By assuming that the K classes share a common covariance matrix, the LDA model becomes linear in x, which means there are Kp linear coefficients to estimate.
- **LDA is a much less flexible classifier than QDA**.
    - ⇝ LDA tends to be a better bet than QDA if there are relatively few training observations and so reducing variance is crucial.
    - ⇝ QDA is recommended if the training set is very large, so that the variance of the classifier is not a major concern, or if the assumption of a common covariance matrix for the K classes is clearly untenable.

Quadratic discriminant analysis



- The Bayes (purple dashed), LDA (black dotted), and QDA (green solid) decision boundaries for a two-class problem.
- **Left**: The two Gaussian classes have $\Sigma_1 = \Sigma_2 = 0.7$.
    - ⇝ The Bayes decision boundaries is linear and accurately approximated by the LDA decision boundary.
    - ⇝ The QDA decision boundary is inferior, because it suffers from higher variance without a corresponding decrease in bias.

Quadratic discriminant analysis



- The Bayes (purple dashed), LDA (black dotted), and QDA (green solid) decision boundaries for a two-class problem.
- **Right**: The orange class has a correlation of 0.7 between the variables and the blue class has a correlation of -0.7.
    - ⇝ The Bayes decision boundaries is quadratic.
    - ⇝ The QDA decision boundary approximated the Bayes classifier more accurately.

Quadratic discriminant analysis in R

We continue with the `Smarket` data. QDA is implemented in R using the `qda()` function

```
> qda.fit=qda(Direction ~ Lag1+Lag2,data=Smarket ,subset=train)
> qda.fit
Call:
qda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)

Prior probabilities of groups:
 Down    Up
0.492 0.508

Group means:
        Lag1    Lag2
Down  0.0428  0.0339
Up   -0.0395 -0.0313
```

- The output contains output contains the group means.
- But it does not contain the coefficients of the linear discriminants, because the QDA classifier involves a quadratic, rather than a linear, function of the predictors.

Quadratic discriminant analysis in R

We continue with the `Smarket` data. QDA is implemented in R using the `qda()` function

```
> qda.class=predict(qda.fit,Smarket.2005)
> table(qda.class$class,Direction.2005)
      Direction.2005
       Down  Up
  Down   30  20
  Up     81 121
> mean(qda.class$class==Direction.2005)
[1] 0.599
```

- The QDA predictions are accurate almost 60% of the time, even though the 2005 data was not used to fit the model.
- This level of accuracy is quite impressive for stock market data, which is known to be quite hard to model accurately.
- This suggests that the quadratic form assumed by QDA may capture the true relationship more accurately than the linear forms assumed by LDA and logistic regression.