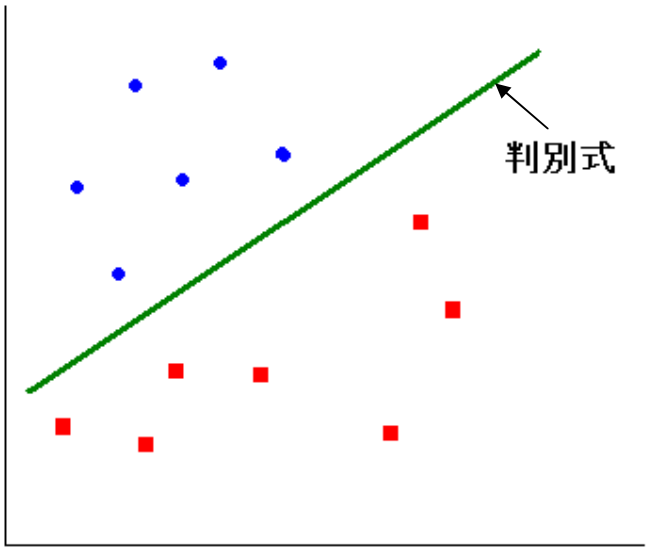
村上　明
1/29/09

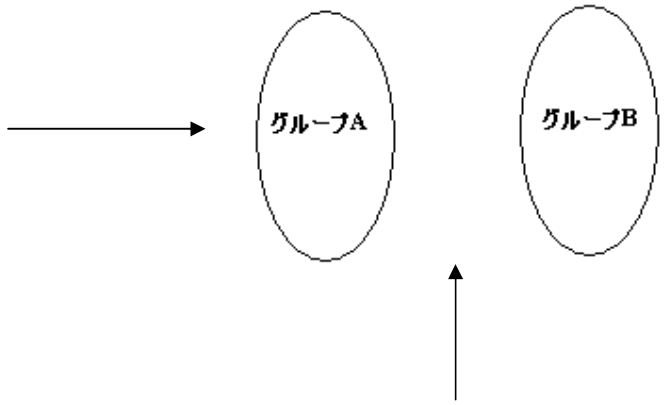
5.2.2 Discriminant analysis

判別分析とは
- 教師あり
- 目的はいくつかの変数に基づいて，各データがどの群に所属するかを判定すること
- PCA と同様、データを少ない次元で表す。通常散布図によって描かれる二次元。
- 何らかの数学的な基準に基づいて、大量のデータを複数のグループに分類する手法
- 目的変数（どんなグループに分類するか）と、説明変数（何を手がかりに分類するか）を明確にする
- 散布図に散らばった青と赤の点をできるだけ正確にわけるためにはどういう分割線を引けば良いか
  - その分割線が判別式



- データの母分散が等分散なら判別式は線形で、等分散じゃないなら非線形
  - 等分散か否かを調べるにはボックスの M 検定を用いる
- 線形は「線形判別関数による分析」と「マハラノビスの距離による判別分析」に分けられる。二つの良いとこ取りをしたのが SPSS で用いられている正準判別分析
- SVM は非線形判別分析

- 母語話者と非母語話者の文章の判別をしたい
  →複数の変数に基づく新たな合成変数を作成して、その視点からデータを分析
- よく分離して見えるようになる視点を探す



・ 最も良い判別のことを相関比と呼び、（グループ間の分散/全分散）を最大化するような値を求める
  - つまり全体の分散におけるグループ内の分散を小さくしたい
  - 群間分散を最大化、郡内分散を最小化

以下、著者推定の例で説明する
・ 3 人×各 5 作品＝15 作品がある（spanishMeta）

```
spanishMeta = spanishMeta[order(spanishMeta$TextName),]
```

・ それぞれのテキストから約 3000 語を抜粋し、タグ付けし、タグの 3-gram の相対頻度を算出したデータが spanish に格納されている

```
dim(spanish)
```
※dim は行数と列数を返す

```
spanish[1:5, 1:5]
```

・ 縦と横を入れ替える

```
spanish.t = t(spanish)
```

・ まずは PCA で分類してみる

```
spanish.pca = prcomp(spanish.t, center = T, scale = T)
spanish.x = data.frame(spanish.pca$x)
spanish.x = spanish.x[order(rownames(spanish.x)),]
library(lattice)
super.sym = trellis.par.get("superpose.symbol")
splom(~ spanish.x[, 1:3], groups = spanishMeta$Author,
        panel = panel.superpose,
        key = list(
                title = "",
                text = list(levels(spanishMeta$FullName)),
                points = list(pch = super.sym$pch[1:3],
                        col = super.sym$col[1:3])
        )
)
```

→PC1 と PC2 で成る平面を見ると、Cela と Mendoza は分かれている。しかし VargasLLosa は残りの二名からは区別できない

→そこで教師なしのクラスタリングを、教師ありの classification に変えてみる

- まずは spanish.t を spanishMeta のデータと合わせるため、行の順番を変える

spanish.t = spanish.t[order(rownames(spanish.t)),]

- そして判別分析を行う lda( )関数を用いるために、MASS パッケージを読み込む

library(MASS)

- lda( )関数は第一引数として数値予測変数のマトリックスを、第二引数として分類名のベクトルを指定する

spanish.lda = lda(spanish.t, spanishMeta$Author)

- 共線性の warning が出る。これは spanish.t の列間の相関が lda 関数に入れるには高すぎるということ

→よって先ほど行った PCA の主因子の上から 8 個で行う

- 80%を説明していることが以下で確かめられる

summary(spanish.pca)

- 主因子は相関がないように抽出するので、warning は消える

spanish.pca.lda = lda(spanish.x[, 1:8], spanishMeta$Author)
plot(spanish.pca.lda)

→きれいに著者ごとに分かれた

- predict( )関数の第一引数にモデル、第二引数にデータを入れることによりモデルを検証する
  - posterior 変数に確率が格納されている

round(predict(spanish.pca.lda, spanish.x[, 1:8])$posterior, 4)

→それぞれのテキストの著者が非常に高い確率で正確に判別されていることがわかる

- 実はこのモデルは過剰適合している。与えられたデータを分類しているだけで、新しいデータを分類できるという保障はない。
- それが各グループの平均を見てみるとわかる

spanish.pca.lda

→平均値の差がそれほど大きくない。統計的に有意なほど差があるのか？
→多変量分散分析（MANOVA）で検証する

- 数値ベクトル群を従属変数と、因子を予測変数と捉える
- 従属変数の平均間に差があるのか

spanish.manova = manova(cbind(PC1, PC2, PC3, PC4, PC5, PC6, PC7, PC8) ~ spanishMeta$Author, data = spanish.x)

・ 結果を出力。Pillai-Bartlett 統計量（F 分布と似ている）を用いている

summary(spanish.manova)

→p 値が小さい
→統計的に有意な差
しかし違いはそれほど強いものではない

・ どの程度一般化できる結果であるかを cross-validation で試す
  - X グループと Y グループを判別する式を出して、それを X と Y に当てはめてもトートロジー
  - 本来であれば新しいデータでテストしたいが、全データ数が少ないとそれもできない。そこで以下のような方法を用いる
  - 15 テキスト中、14 テキストを用いて判別式を立て、残りの 1 つを予想する。それを全てのテキストについて行う（＝15 回）
  - その正答率で、新しいテキストが与えられた時にどの程度の確率で判別に成功するのかを探る
・ lda( )関数にもそのオプション（CV=TRUE）があるが、spanish.x のデータが全ての著者の全ての作品を含んでいるため使えない

```
spanish.t = spanish.t[order(rownames(spanish.t)),]
n= 8
```

・ 何も入っていないベクトルを 15 個準備する

```
predictedClasses = rep("", 15)
```

・ 実際に判別分析を 15 回繰り返す

```
for (i in 1:15) {
        training = spanish.t[-i,]
        trainingAuthor = spanishMeta[-i,]$Author
        training.pca = prcomp(training, center = T, scale = T)
        training.x = data.frame(training.pca$x)
        training.x = training.x[order(rownames(training.x)),]
        training.pca.lda = lda(training[, 1:n], trainingAuthor)
        predictedClasses[i] =
                as.character(predict(training.pca.lda, spanish.t[, 1:n])$class[i])
}
```

・ 予想と実際のデータを比較

```
data.frame(obs = as.character(spanishMeta$Author), pred = predictedClasses)
```

・ 正しく予想できている数を出す

```
sum(predictedClasses == as.character(spanishMeta$Author))
```

・ 二項検定によると有意

```
sum(dbinom(9:15, 15, 1/3))
```