Instituto Cántabro de ESTADÍSTICA

Estadística con R

Clasificadores

Instituto Cántabro de ESTADÍSTICA

- Análisis discriminante lineal (estadístico)
- Árbol de decisión (aprendizaje automático)
- Máquina soporte vector (aprendizaje automático)

Análisis discriminante lineal (AD)

- El AD trata de establecer una relación entre una variable dependiente no métrica (dicotómica o multidicotómica) y un conjunto de variables independientes métricas:

$$Y_1 = X_{1i} + X_{2i} + ... + X_p$$

- El propósito del AD consiste en aprovechar la información contenida en las variables independientes para crear una función Z combinación lineal de las p explicativas, capaz de diferenciar lo más posible a los k grupos:

$$Z_{jk} = \beta_0 + \beta_1 X_{1k} + \beta_2 X_{2k} + ... + \beta_p X_{pk}$$

- es la puntuación Z_{ik} discriminante j para el objeto k
- término constante β_o
- β_i ponderación discriminante para la variable independiente i
- X_i variable independiente i para el objeto k.

Instituto Cántabro de ESTADÍSTICA

Dos predictores y dos grupos

- Sustituyendo en la función discriminante el valor de las medias del grupo 1 en las variables X_1 y X_2 ,obtenemos el centroide del grupo 1:

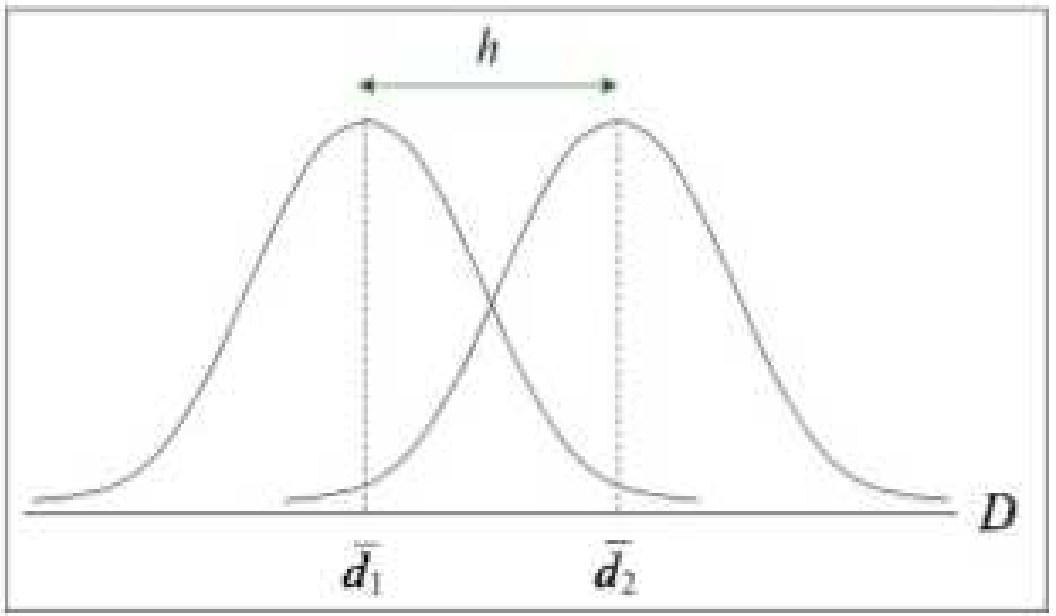
$$\overline{Z}_1 = \beta_0 + \beta_1 \overline{X}_{11} + \beta_2 \overline{X}_{21}$$

- De igual modo, sustituyendo las medias del grupo 2, obtenemos el centroide del grupo 2:

$$\overline{Z}_2 = \beta_0 + \beta_1 \overline{X}_{12} + \beta_2 \overline{X}_{22}$$

- La función Z debe ser tal que la distancia entre los dos centroides sea máxima :

$$h = \overline{Z}_1 - \overline{Z}_2$$



Una manera de clasificar los casos consiste en calcular la distancia existente entre los centroides de ambos grupos y situar un punto de corte :

$$z_0 = \frac{\bar{Z}_1 + \bar{Z}_2}{2}$$

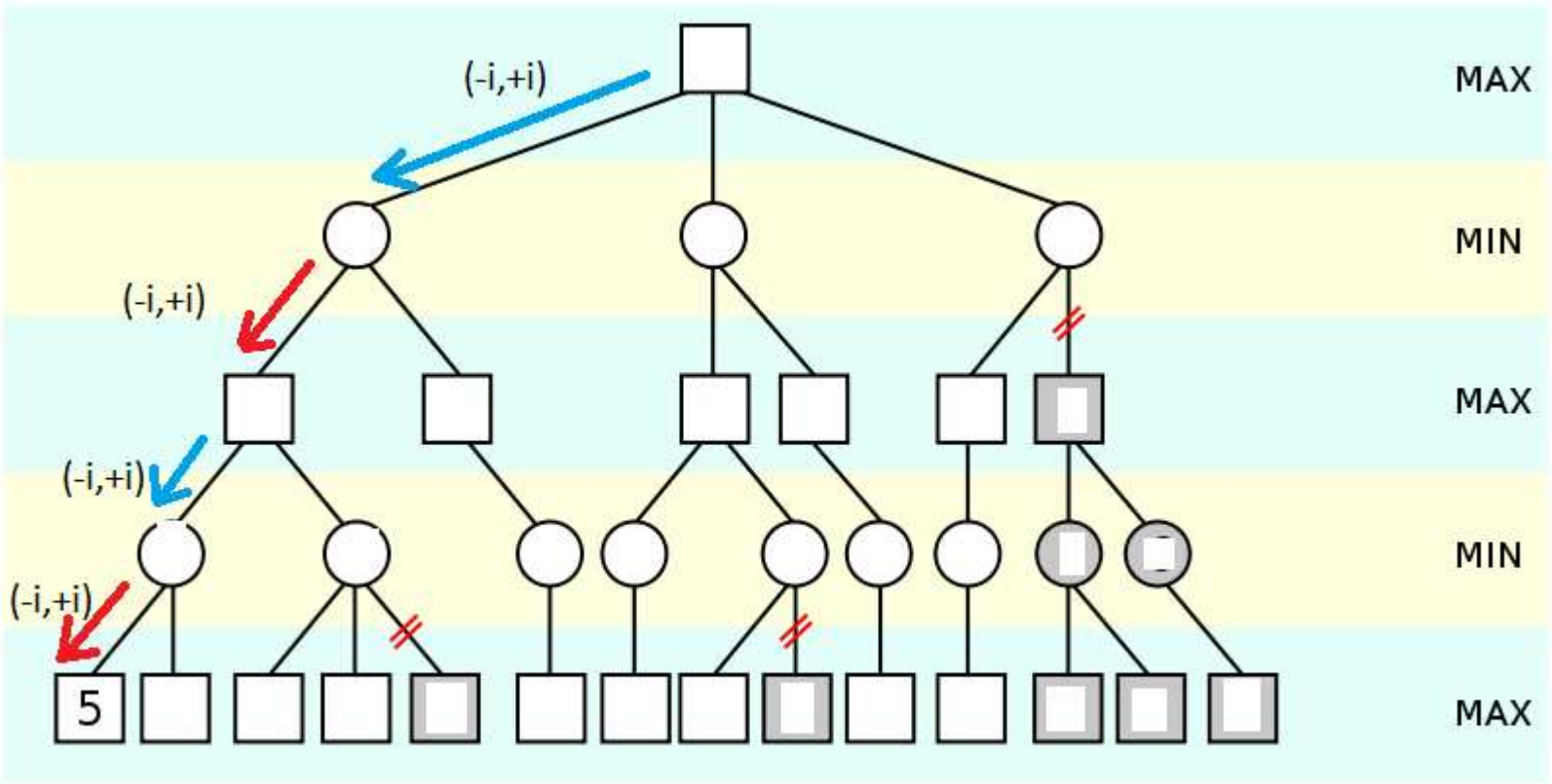
Clasificación AD en R

- library(MASS)
- data(iris)
- attach(iris)
- # división de la muestra en entrenamiento y validación
- train=sample(seq(length(iris$Species)),length(iris$Species)*0.70,replace=FALSE)
- # Lineal Discriminat Analisys
- lda.tr=lda(Species[train]~.,data=iris[train,])
- #predicción
- probs=predict(lda.tr,newdata=iris[-train,],type="prob")
- data.frame(probs)[1:5,]
- table(probs$class,iris$Species[-train])
- mean(probs$class==iris$Species[-train]) #porcentaje de bien clasificados
- data(cpus, package="MASS")

Árbol de decisión

- Los árboles de decisión o de clasificación son un modelo surgido en el ámbito del aprendizaje automático (Machine Learning) y de la Inteligencia Artificial que partiendo de una base de datos, crea diagramas de construcciones lógicas que nos ayudan a resolver problemas (segmentación jerárquica).
- Utiliza un proceso de división secuencial, iterativo y descendente que partiendo de una variable dependiente, forma grupos homogéneos definidos específicamente mediante combinaciones de variables independientes en las que se incluyen la totalidad de los casos recogidos en la muestra.

Arbol de decisión y ejemplo de poda



Poda de un árbol de decisión

- Pureza de nodo. Si el nodo solo contiene ejemplos o registros de una única clase se decide que la construcción del árbol ya ha finalizado.
- Cota de profundidad. Previamente a la construcción se fija una cota que nos marque la profundidad del árbol, cuando se alcanza se detiene el proceso.
- Umbral de soporte. Se especifica un número de ejemplos mínimo para los nodos, y cuando se encuentre un nodo con ejemplos por debajo del mínimo se para el proceso, ya que no consideramos fiable una clasificación abalada con menos de ese número mínimo de ejemplos.

Instituto Cántabro de ESTADÍSTICA

Arbol de decisión con R

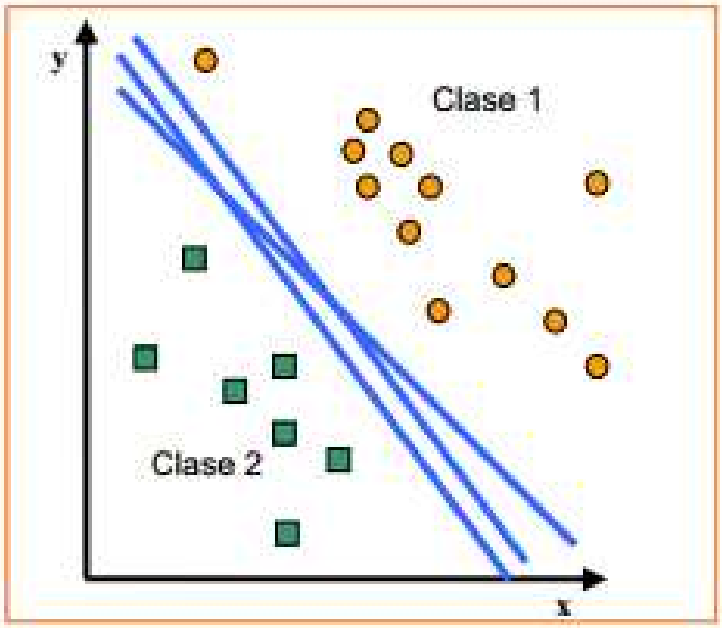
- library(tree)
- data(iris)
- attach(iris)
- # división de la muestra en entrenamiento y validación
- train=sample(seq(length(Species)),length(Species)*0.70,replace=FALSE)
- # Árgbol de decisión
- iris.tree = tree(Species~.,iris,subset=train)
- summary(iris.tree)
- plot(iris.tree) ;text(iris.tree,pretty=0)
- #Predicción
- tree.pred=predict(iris.tree,iris[-train,],type="class")
- summary(tree.pred)
- with(iris[-train,],table(tree.pred,Species))

Poda Arbol de decisión

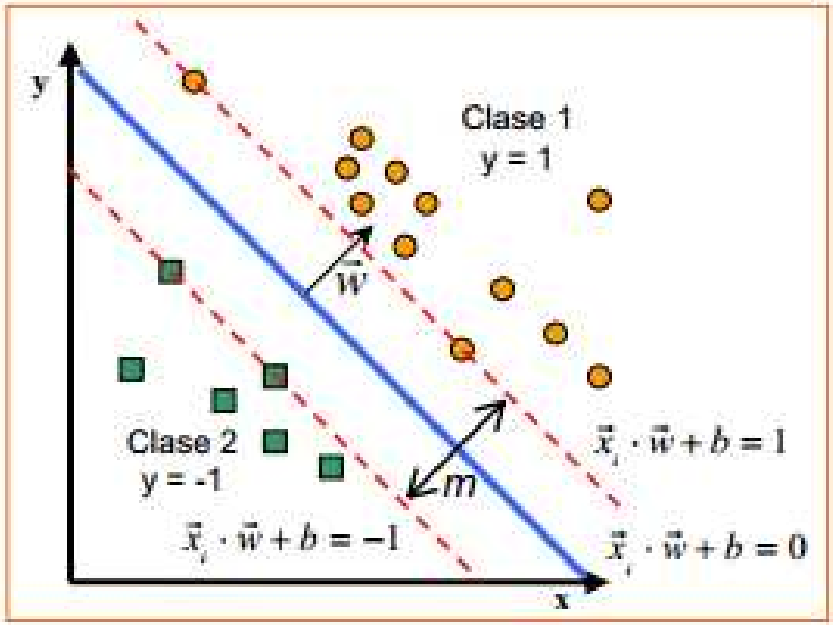
- cv.iris=cv.tree(iris.tree,FUN=prune.misclass)
- cv.iris
- plot(cv.iris)
- prune.iris=prune.misclass(iris.tree,best=3)
- plot(prune.iris);text(iris.tree,pretty=0)

Máquina Soporte Vector (SVM)

- Partiendo de un conjunto de datos etiquetados en diferentes clases, representar estos en puntos en el espacio para tratar de separar las diferentes clases mediante un espacio lo más amplio posible (separación optima)
- Algoritmo SVM buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. (clasificadores de margen máximo).
- Los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado.



Posibles hiperplanos de separación



Hiperplano de separación óptimo

Maquina soporte vector con R

- library(e1071)
- data(iris)
- attach(iris)
- # división de la muestra en entrenamiento y validación
- train=sample(seq(length(Species)),length(Species)*0.70,replace=FALSE)
- # Máquina Soporte Vector
- svmfit=svm(Species~.,data=iris,kernel="linear",scale=FALSE,subset=train)
- print(svmfit)
- table(iris$Species[train],svmfit$fitted)
- # Predicción para la muestra test
- svm.pred=predict(svmfit,iris[-train,])
- summary(svm.pred)
- with(iris[-train,],table(svm.pred,Species))

Multiclasificadores

- Los métodos multiclasificadores más conocidos son el Bagging (Breiman, 1966) y Boosting (Freund y Schapire, 1996).
- La operativa del Bagging es la siguiente:
  - Se generan muestras aleatorias que serán los conjuntos de entrenamiento. Las muestras se generan a través de un muestreo aleatorio con reemplazamiento.
  - Cada subconjunto de entrenamiento aprende un modelo.
  - Para clasificar un ejemplo se predice la clase de ese ejemplo para cada clasificador y se clasifica en la clase con mayor voto.
- El Boosting se usa para el entrenamiento de una forma ponderada de un conjunto de datos en el cual el coeficiente de puntos asociado a cada peso depende del desempeño del clasificador anterior, dando mayor pesos a los elementos mal clasificados. El procedimiento iterativo se inicia dado a cada elemento el mismo peso (1/n).
- El package R: “ipred” opera multiclasificadores por los métodos bagging y boosting.