Bagging

Un problema muy común es intentar predecir el comportamiento de una variable dependiente Y a partir de un conjunto de p variables independientes $X \in R^p$

A partir de una muestra de observaciones $(\vec{x_i}, y_i),\ i = 1, \ldots, n$ se busca una regla de clasificación, RC mediante regresión logística, árboles de clasificación, etc.

Bagging

Si se conociera la ditribución conjunta de $(\vec{X}, Y)$, es decir,

$$P[(\vec{X}, Y) = (\vec{x}, y)]$$

el procedimiento más eficiente sería utilizar como regla de predicción la probabilidad a posteriori

$$RC(\vec{x}) = argmax_y P(Y = y|\vec{x}) = argmax_y P[(\vec{X}, Y) = (\vec{x}, y)]$$

El problema es que la distribución conjunta suele ser desconocida y es necesario estimarla.

Bagging. Bootstrap

El método Bootstrap se basa en sustituir la Función de distribución teórica $F_{(X,Y)}$ por la función de distribución empírica $\hat{F}(X,Y)$ definida por

$$\hat{F}(X,Y) = \begin{cases} \frac{1}{n} & si(\vec{x},y) = (\vec{x_i},y_i) \text{ para algún i} \\ 0 & \text{en otro caso} \end{cases}$$

(Teorma de Glivenko Catelli)

Una muestra bootstrap de tamaño n $(\vec{x_i}^*, y_i^*)$, se obtiene eligiendo al azar y con reemplazamiento n observaciones de la muestra original $(\vec{x_1}, y_1), \ldots, (\vec{x_n}, y_n)$

Así se tiene una idea más clara de lo que ocurriría si se pudiera repetir varias veces el proceso de muestreo inicial

Bagging

A partir de una muestra $(\vec{x}_i, y_i),\ i = 1, \ldots, n$ el método bagging utiliza el siguiente algorítmo

- Se eligen b muestras bootstrap $(\vec{x}_i^{*b}, y_i^{*b},\ b = 1, \ldots, B)$
- Se calcula una regla de clasificación RC^b basada en la correspondiente muestra bootstrap b-ésima.
- Se aplica la regla de clasificación para una de esas B muestras, $RC^b(\vec{x}_i^{*b})$

La clasificación final de cada individuo $\vec{x}$ se hace como promedio de la B predicciones

$$RC_{bag}(\vec{x}) = argmax_k \sum_{b=1}^{B} I(RC^b(\vec{x}) = k)$$

es decir, se asigna a la categoría más elegida.

Bagging

La estimación de la probabilidad de pertenecia a un categoría de k, se debe basar en las probabilidades de clasificación obtenida en cada una de las B réplicas realizadas

$$\hat{P}_{bag}(\vec{x} \in k) = \frac{1}{B} \sum_{b=1}^{B} \hat{P}^b(\vec{x} \in k)$$

Esta estimación sugiere asignar las observaciones a clases empleando el siguiente criterio:

$$RC_{bag}(\vec{x}) = argmax_k \{\hat{P}_{bag}(\vec{x} \in k)\}$$

Bagging

Estimación del error Out-of-bag (OOB) error

- Usar validación cruzada para estimar el error en la predicción.
- En cada Bootstrap se utiliza aprsximadamente el 63 % de las muestras.
- Usar el 37 % restante para estimar el error. $e_i = y_i - \hat{y}_i^{oob}$
- Calcular el Error Cuadrático Medio de tosa las observaciones

Bagging

DESVENTAJAS

- La tasa de error tiende a reducirse al promediar muchos criterios de clasificación.
- Permite estimar la importancia de las variables
  1. Se estima la reducción del Error Cuadrático Medio cada vez que interviene una variable.
  2. Se calcula el promedio de las reducciones en todas las muestra bottstrap.

Bagging

DESVENTAJAS

- No se debe emplear cuando los clasificadores individuales funcionan mal
- La interpretación del criterio de clasificación suele ser mucho más difícil.
- Requiere muchos más cálculos

CÓDIGO DE R

```
## Ejemplo

library(ipred)

library(rpart)

library(mlbench)

BC<- data(BreastCancer)

n<- length(BC[,1])

entrena <- sample(1:n,2*n/3)
```

CÓDIGO DE R

```
BC.bagg<- bagging(Class ~.,data=BC[entrena,-1],
                  nbagg=10,control=rpart.control(maxdepth=3))

BC.bagg.pred<- predict(BC.bagg, newdata= BC[-entrena,-1],
                    type="prob")

BC.bagg.pred <- predict(BC.bagg, newdata=BC[-entrena,-1],
                     type="class") ## Por defecto

names(BC.bagg)
BC.bagg$mtrees
BC.bagg$OOB
BC.bagg$comb
BC.bagg$err
```

CÓDIGO DE R

```
## Bagging y predict dependen de las rutinas de rpart

## En cada bootstrap rpart se controla con los
      parámetros habituales.

## Por defecto rpart.control tiene minsize=2 y cp=0.
       Conviene poner xval=0

## Se puede utilizar la opción prune para podar
      los árboles.
```