Random Forest

Edgar Acuna

Departamento de Ciencias Matematicas

UPR-Mayaguez

Definicion

- **Random forest** es una combinacion de muchos arboles de decision y cuya clase predicha es obtenida por votacion (moda) de las clases predichas por los arboles indviduales.
- Fue introducido por L. Breiman en el 2001.
- El metodo combina "bagging" con la seleccion al azar de atributos predictores.

Algoritmo-1

1. Considerar que la muestra de entrenamiento tiene N instancias y que el numero de variables predictoras es M.

2. Construir n (usualmente 500) muestras de entrenamiento para el arbol, escogiendo muestras con reemplazo y del mismo tamano de la muestra orignal (muestras bootstrap). Las instancias que no son elegidas forman una muestra de prueba y son usadas para estimar la tasa de error del arbol mediante la prediccion de sus clases (estimacion out-of-bag del error de clasificacion).

3. Construir un arbol de decision para cada muestra bootstrap pero usando en cada particion solo m atributos elegidos al azar para hallar la mejor particion. Usualmente m=sqrt(M).

Algoritmo-2

4. A diferencia de un clasificador por arboles donde se hace poda, aqui se deja crecer el arbol hasta el final y no se hace poda.

Para predecir una nueva instancia se la hace recorrer el arbol y se le asigna la etiqueta del nodo terminal correspondiente. Este procedimiento es repetido sobre todos los arboles y al final se asigna la clase predicha por votacion.

Otros aspectos de RF

- Toma en cuenta el hecho que el conjunto de datos tiene clases desbalanceadas (Churn problem).
- Calcula proximidades entre pares de datos. Esta se define como la proporcion de las veces que dos instancias caen en el mismo nodo terminal en distintos arboles.
- Estas proximidades pueden ser usadas para hacer clustering, detectar outliers, para imputar valores perdidos (a lo knn-impute) o tambien para visualizar la data (se combina con MDS, multidimensional scaling).
- Tambien sirve para detectar experimentalmente si hay interacciones entre variables predictoras.

Estimando la importancia de cada predictora

1. Construir S random Forest. Para k=1…S, repetir pasos del 2 al 4.
2. Para cada arbol t del k-esimo RF, considerar el estimado OOB_t del error de mala clasificacion.
3. Para cada predictor Xj permutar al azar los valores de X_j para generar una nueva muestra y obtener un nuevo error $OOB_{t(j)}$ usando esta nueva muestra.
4. Una medida de imprtancia del predictor Xj es el promedio de las diferencias $OOB_{t(j)}$– OOBt sobre todos los arboles del k-esimo RF.
5. Una medida promedio de importancia del predictor Xj sera el promedio de las medidas del paso 4 en todos los S random Forests.

Librerias para Random Forest

- En R, esta disponible la libreria randomForest

```
 arbol.rf=randomForest(as.factor(V7)~.,data=bupa,importance=T)
> print(arbol.rf)
Call:
 randomForest(formula = as.factor(V7) ~ ., data = bupa, importance = T)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 26.38%
Confusion matrix:
   1      2     class.error
1 87  58       0.400
2 33 167       0.165
```

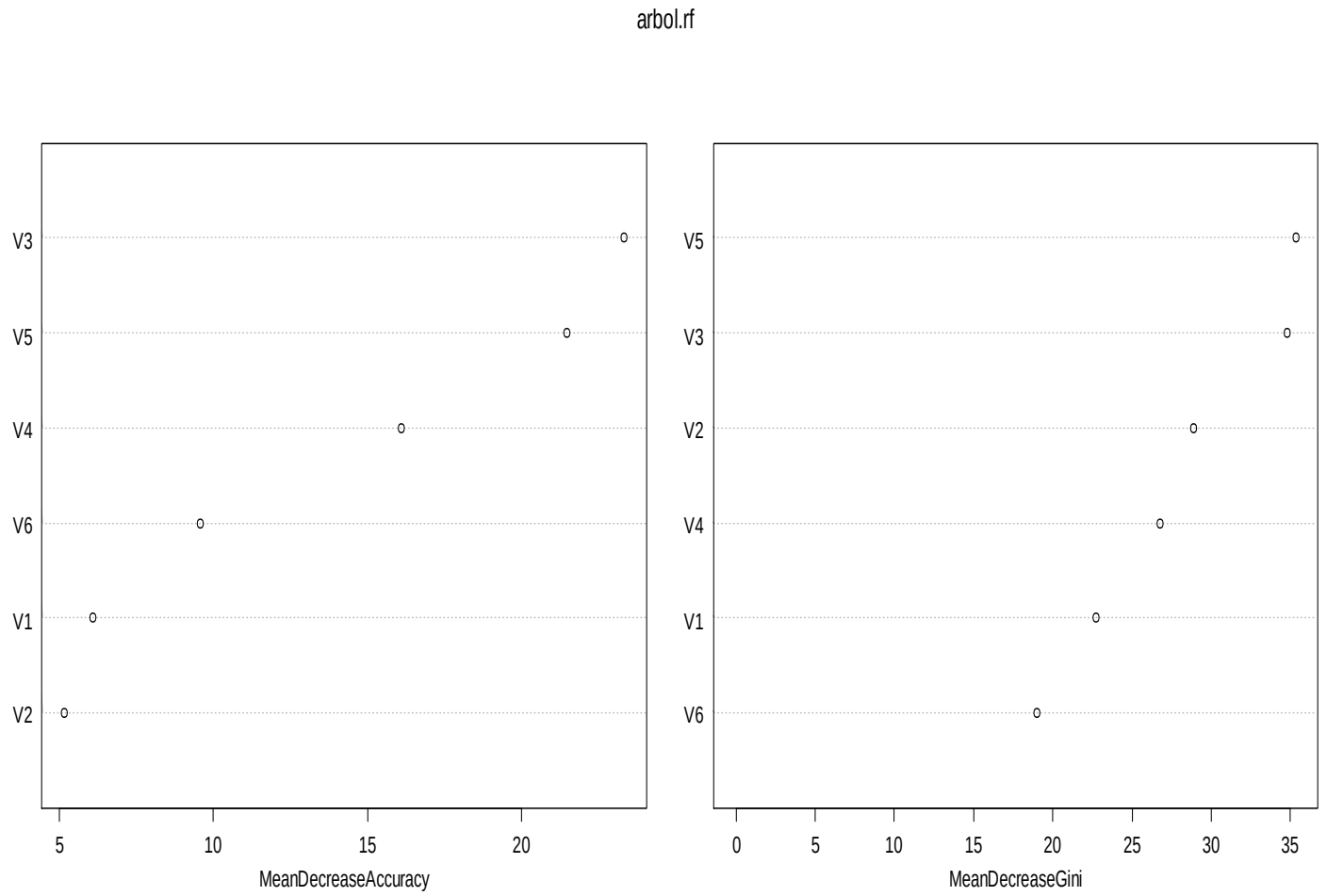
Librerias para Random Forest

La opcion importance=T calcula un indice de importancia de los atributos y se puede usar para seleccion de variables.

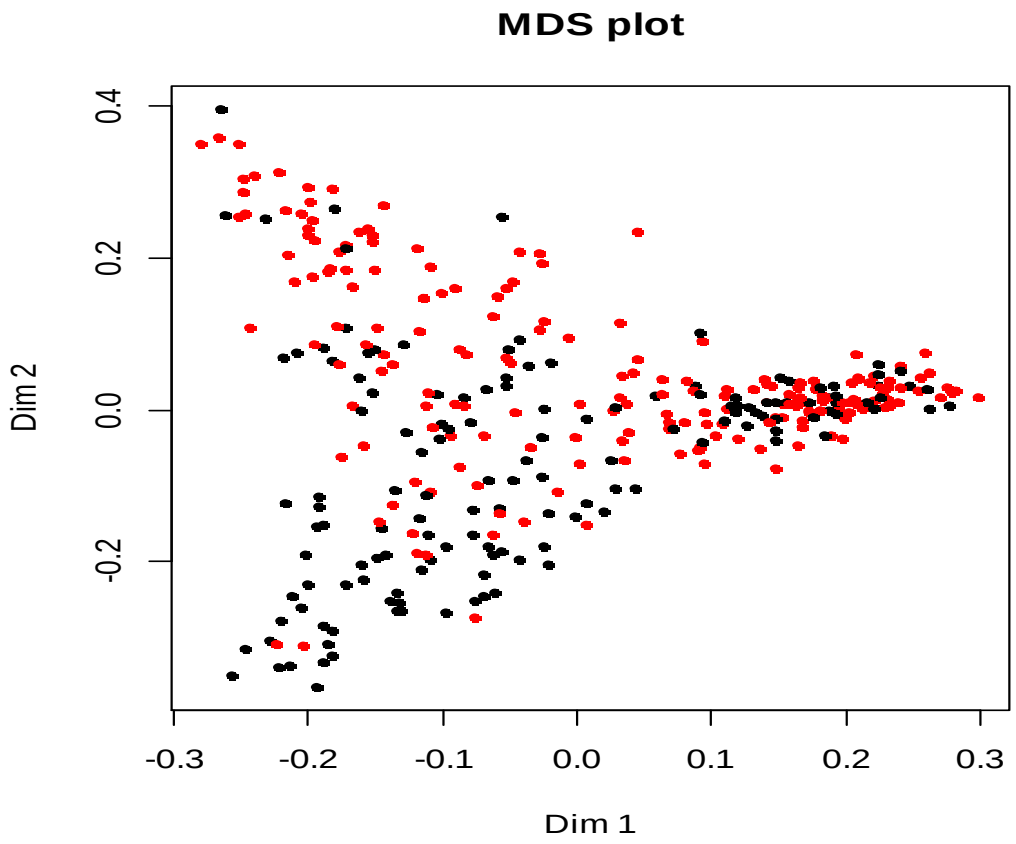
> importance(arbol.rf)

	1	2	MeanDecreaseAccuracy	MeanDecreaseGini
V1	10.014802	0.471505	6.726993	23.56741
V2	4.289097	4.334051	6.293027	28.34022
V3	10.596978	20.803267	22.838347	33.52794
V4	8.721250	13.370352	16.550863	27.54833
V5	16.147347	16.526996	23.186996	35.49875
V6	13.812461	-2.327833	8.138179	19.01831

De acuerdo a la primera medida V5 y V3 son las mas importantes y de acuerdo a la segunda medida V5, V3 V2 y V4



VarImpPlot(arbol.rf) indica que V3 y v5 son las variables mas importantes




```
arbol.rf=randomForest(as.factor(V7)~.,data=bupa,importance=T,proximity=T)
MDSplot(arbol.rf,fac=as.factor(bupa$V7), main="MDS plot",palette=1:2)
```

Ventajas de RF

- Es uno de los algoritmos de Machine Learning que produce un alto nivel de precision. Para muchos conjuntos de datos da el mas alto nivel de precision.
- Funciona eficientemente con grandes conjuntos de datos
- Puede manipular datos con miles de variables, sin hacer seleccion previa (Bioinfomatica). Fuerte competidor de SVM.
- Estima las variables que son mas importantes para la clasificacion.
- Calcula un estimador insesgado el error a medida que se generan mas arboles.
- Tiene un metodo eficiente de estimar los valores faltantes y mantiene la precision aun con un gran porcentaje de datos faltantes(usando medidad de proximidad).

Desventajas

- Se ha observado que Random forests sobreajusta algunos conjuntos que contiene variables ruidosas.
- Para datos que incluyen variables categoricas con diferente numero de niveles, random forests son sesgados en favor de aquellos atributos con mayor numero de niveles. Por lo tanto, los scores de importancia dados por el RF no son muy confiables para este tipo de datos.

Conclusiones y resumen:

- RF es bien rapido
  - RF es rapido para construir. Aun mas rapido para predecir!
  - En la practica, no requerir validacion cruzada, hace que se gane en rapidez de entrenamiento en un factor que puede superar a 100.
  - Es completamente paralelizable, lo cual lo puede hacer aun mas rapido!
- Incluye automaticamente seleccion de variables.
- RF es resistente a sobreajuste.
- RF tiene la capacidad de analizar datos sin la necesidad de preprocesamiento,
  - Los datos no necesitan ser reescalados, transformados o modificados.
  - Es resistente a outliers
  - El tratamiento de valores faltantes esta incluido en el algoritmo.
- Se puede usar para formar clusters usando la matriz de proximidad generada por RF.