RANDOM FOREST

Miguel Cárdenas-Montes

Los árboles de decisión son estructuras lógicas con amplia utilización en la toma de decisión, la predicción y la minería de datos.

Objetivos:

- Entender como funcionan los algoritmos basados en *random forest*.
- Conocer las diferencias entre *random forest* y árboles de decisión.
- Conocer las capacidades que proporciona el algoritmo *random forest*: importancia de las variables, y proximidad.

Este documento puede contener imprecisiones o errores. Por favor no lo utilice para citarlo como una fuente fiable.

1 *Introducción*

Random Forests (Breiman, 2001) [1] es un algoritmo para clasificación y regresión de amplio uso en la comunidad que tiene un rendimiento especialmente bueno para datos de alta dimensionalidad.

Random Forest puede considerarse como la integración de las siguientes técnicas: *Decision Trees*, *Bagging*, y *Random Subspace*.

[1] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 10.1023/A:1010933404324. URL http://dx.doi.org/10.1023/A:1010933404324

2 *Random Forest*

El esquema del algoritmo *random forest* es:

Este algoritmo pertenece a los métodos denominados *Ensemble Methods*. Otros *Ensemble Methods* son bagging y Boosting.

1. Aleatoramente se crea (seleccionando con reemplazado) el conjunto de entrenamiento de igual tamaño que el conjunto original. Al seleccionarse aleatoriamente con reemplazo no todos los datos de conjunto general estarán en el conjunto de entrenamiento. La probabilidad de que un dato particular esté en el conjunto de entrenamiento es aproximadamente 66 %.
2. Los datos que no forman parte del conjunto de entrenamiento forman el conjunto de validación o out of bag data (OOB data)
3. En cada punto de división del árbol o nodo, la búsqueda de la mejor variable para dividir los datos no se realiza sobre todas las variables sino sobre un subconjunto, m, de las mismas. La elección del subconunto de variables se realiza de forma aleatoria.
4. Se busca la mejor división de los datos de entrenamiento teniendo en cuenta solo al m variables seleccionadas. Para esta tarea se debe implementar una función objetivo. Habitualemtne ésta es la entropía o el índice de Gini.

5. Los anteriores procesos son repetidos varias veces, de forma que se tienen un conjunto de árboles de decisión entrenados sobre diferentes conjuntos de datos y de atributos.

6. Una vez el algoritmo entrenado, la evaluación de cada nueva entrada es realizado con el conjunto de árboles. La categoría final de la clase (clasificación) es realizado por el voto mayoritario del conjunto de árboles, y en caso de regresión por el valor promedio de los resultados.

La técnica denominada *Random subspace* o *attribute bagging* consiste en la selección de un subconjunto $m << M$ aleatorio de atributos de una muestra bootstrap $D_i >> D$. De foma que al final se tiene un subconjunto $D_i >> D$ de objetos tomando solamente un subconjunto de atributos $m << M$.

Los datos OOB se usan para determinar la impureza en los nodos terminales. La suma de estas impurezas determina la impureza del árbol.

Cada registro del conjunto global de datos estará *in bag* para algunos árboles del *random forest*, y *out of the bag* para otros árboles. Es probable que cualquier par de registros no compartan un patrón idéntico sobre en qué árboles están *in bag* y en cuales están *out of the bag*.

En el lado izquierdo de la figura 1 se muestran ejemplos de la aplicación de algoritmo *decision tree* para un conjunto de datos. Para el mismo conjunto de datos, los ejemplos equivalentes para *random forest* son mostrados en el lado derecho de la figura 1. Como puede apreciar el algoritmo *random forest* ofrece más matices en el mapa de predicción que el algoritmo *decision tree*. Los resultados del algoritmo *decision tree* son categóricos en las áreas donde no hay datos.

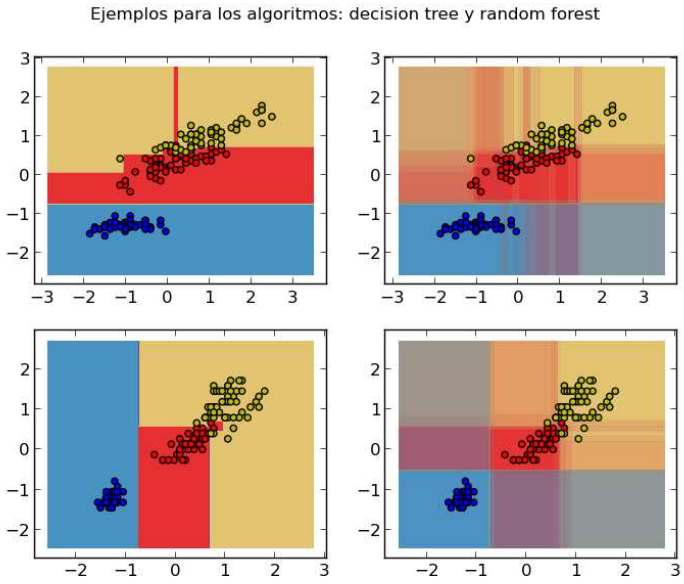

Figura 1: Ejemplo de *random forest*.

Para medir el error de *random forest* se suele utilizar la técnica denominada *out-the-bag error*. Para cada árbol se utiliza el conjunto de objetos no seleccionados por su muestra *bootstrap* de entrenamiento para ser clasificados con dicho árbol. Promediando sobre el conjunto de árboles de *random forest* se puede estimar el error del algoritmo.

Por otro lado, *random forest* puede ser paralelizado eficazmente puesto que cada árbol puede construirse de manera independiente a los otros árboles. Esta característica lo hace deseable para entornos paralelos.

3 *Importancia de las Variables*

El mecanismo de construcción de *random forest* permite establer un baremo de la importancia de cada variable en la predicción final. Para ello se calcula el error de la muestra OOB. A continuación para cada variable de la muestra OOB, se permuta un par de elementos y se vuelve a calcular el error de la muestra OOB permutada. El resultado debería ser peor que para la muestra OOB original.

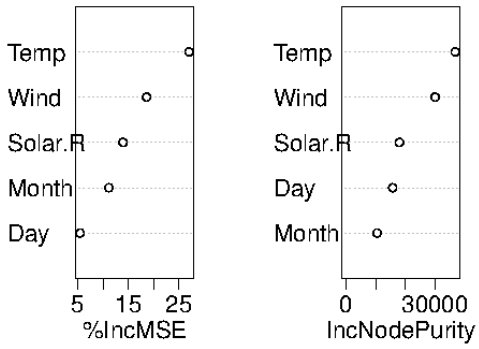

Figura 2: Ejemplo de la importancia de las variables del conjunto de datos denominado ozono. Las variables de la parte superior del gráfico son más importantes que las de la parte inferior.

El procedimiento anterior se realiza para todos los valores de cada variable y se calcula el promedio. Esto se hace para todas la variables. Así las variables menos importantes deberían alterar menos la diferencia entre el error de la muestra OOB y el error de la muestra OOB permutada, que las variables importantes.

4 *Comportamiento en Función del Número de Árboles*

En la figura 3 se ilustra el comportamiento generalizado del algoritmo *random forest* en función del número de árboles que incorpora el algoritmo entrenado. Inicialmente, un incremento del número de árboles permite una mayor diversidad de los mismos, y por lo tanto reduce el error OOB. Sin embargo, la mejora se estanca a partir de un determinado número de árboles.

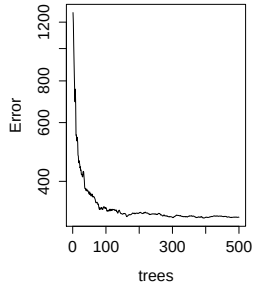

Figura 3: Ejemplo del comportamiento del algoritmo *random forest* en función del número de árboles.

5 Proximidad

El algoritmo *random forest* puede ser utilizado para medir la proximidad de los datos. Esto significa que el algoritmo puede ser utilizado para hacer aprendizaje no supervisado.

Dados dos elementos (i,j), la proximidad entre ambos puede ser medida como la fracción de árboles en los cuales los elementos i y j están en el mismo nodo terminal. Si se hace esta operación para todos los pares de elementos, se formará una matriz de proximidad entre los elementos. Esta matrix de proximidad puede ser utilizada tanto para aprendizaje no supervisado como para explorar la extructura de los datos.

Referencias

[1] Leo Breiman. Random forests. *Machine Learning*, 45 (1):5–32, 2001. 10.1023/A:1010933404324. URL http://dx.doi.org/10.1023/A:1010933404324.