Árboles de decisión

Rafael Jordá Muñoz

En este PDF se resumen las tres actividades que se van a realizar para comprender el algoritmo de ML basado en vecinos más cercanos.

1. Ejercicio 1

Dado el siguiente conjunto de datos de clasificación con 6 observaciones, 3 variables de entrada y una variable de salida:

Observación	**X1**	**X2**	**X3**	**Y**
1	4	3	-1	1
2	-3	-1	-1	0
3	3	-2	0	0
4	1	4	0	1
5	-2	3	1	0
6	-3	5	5	0

Tabla 1: Datos de entrenamiento

Construye el árbol de clasificación (sin podar) mediante CART y utilizando como criterio la entropía. La condición de parada debe ser que los nodos hoja sean puros (todos los ejemplos son de la misma clase). En cada nodo del árbol se debe indicar:

- La variable y su valor umbral.
- La entropía correspondiente.
- En los nodos hoja, la clase del nodo y los ejemplos que pertenecen al mismo.

Este ejercicio debe hacerse SIN usar scikit-learn, para aplicar directamente los conceptos teóricos del método.

2. Ejercicio 2

Dado el problema de clasificación Blood Transfusion Service Center:

a) La clase que implementa el algoritmo CART en problemas de clasificación en scikit-learn es sklearn.tree.DecisionTreeClassifier. Revisa los parámetros y métodos que tiene. En esta versión de scikit-learn aún no está implementado el método de podado del árbol. Dado que en determinadas operaciones (división del conjunto de datos en entrenamiento y test, validación cruzada, entrenamiento) el resultado obtenido depende de números aleatorios, y se quiere que el resultado sea repetible en otras máquinas, es necesario que utilicéis la función np.random.seed(1) al comienzo del fichero .ipynb.

b) Divide los datos en entrenamiento (80 %) y test (20 %).

c) Realiza la experimentación con DecisionTreeClassifier usando los valores por defecto de los parámetros, excepto para criterion que debe tomar el valor 'entropy'. Además, utiliza como hiper-parámetro la variable min_samples_split (permitirá modificar el tamaño del árbol).

   - Muestra la gráfica del error de entrenamiento con validación cruzada (5-CV) frente al valor del hiper-parámetro. ¿Cuál es el menor error de validación cruzada, su desviación estándar y el valor del hiper-parámetro para el que se consigue? ¿Cuál es el valor del hiper-parámetro si se aplicase la regla de una desviación estándar?
   - Muestra la gráfica del error de test frente al valor del hiper-parámetro, y valora si la gráfica del error de entrenamiento con validación cruzada ha hecho una buena estimación del error de test. ¿Cuál es el menor error de test y el valor del hiper-parámetro para el que se consigue? ¿Cuál es el error de test para el valor del hiper-parámetro seleccionado por la validación cruzada? ¿Cuál es el error de test para el valor del hiper-parámetro seleccionado por la validación cruzada mediante la regla de una desviación estándar?

3. Ejercicio 3

Repite el ejercicio 2 pero para el problema de regresión Energy Efficiency con la variable de salida cooling load. La clase que implementa el algoritmo CART en problemas de regresión en scikit-learn es tree.DecisionTreeRegressor.