MICROARRAY DE EXPRESIÓN GENÉTICA

Práctica 1

Resumen ejecutivo de conjunto de datos

Descripción del conjunto de datos

Conjunto de datos: *ALL-AML.arff*

Enlace: https://hastie.su.domains/CASI_files/DATA/leukemia.html

El conjunto de datos está formado por **72 instancias** y **7129 atributos,** todos ellos de tipo numérico. La clase es binaria y de tipo nominal y puede tomar **2 posibles valores**: ALL y AML. En cuanto a la distribución de cada valor: ALL (65,28 %) y AML (34.72 %).

Transformación de la entrada

Realizar las operaciones de limpieza y transformación que considere necesarias.

En cuanto a operaciones de limpieza, no se encontraron valores ausentes en los atributos vistos, pero como el número total de estos es muy elevado, consideramos que ningún atributo presenta valores ausentes. En cuanto a la presencia de outliers, tampoco se han encontrado en los atributos observados, pero por el mismo motivo anterior, no vamos a eliminar ningún outlier. Por lo tanto, el conjunto de datos sigue siendo el mismo por ahora.

Pasamos a normalizar el conjunto de datos escalando en el intervalo [0,1] para darles a todos los atributos el mismo peso. Llamamos al conjunto resultante ***ALL-AML-transformado.arff,*** que será con el que trabajaremos a partir de ahora.

Sin seleccionar atributos

Estimar, mediante validación cruzada de 10 particiones, el error de las hipótesis generadas por los siguientes clasificadores:

- **J48, NB, IBK1, Regresión Logística, MLP (H10), SVM(lineal).**

Procedemos a calcular las tasas de error utilizando los clasificadores pedidos sobre el conjunto de datos normalizado.

Validación Cruzada de 10 particiones						
Clasificador	J48	NB	IBK1	Regresión Logística	MLP (H10)	SVM(lineal)
Tasa de error	0.208	0	0.153	0.125	0.028	0.014

Tabla 1

Selección de atributos: filtro

Seleccionar sucesivamente 4, 8, 16 y 32 atributos mediante los siguientes métodos de filtro:

- **incertidumbre simétrica, ReliefF, eliminación recursiva con SVM y CFsubsetEval (correlación atributos-atributos/clase)**

Procedemos a realizar una selección de atributos mediante métodos de filtro, los cuales son independientes del algoritmo de aprendizaje y, además, se basan en características generales de los datos.

Antes de mostrar los atributos, se comentan cuáles son los métodos a utilizar (evaluadores), siendo los 3 primeros métodos de **selección de atributos individuales**, mientras que el último es un método de **selección de conjunto de atributos**.

1. Incertidumbre simétrica - **weka.attributeSelection.SymmetricalUncertAttributeEval**: método basado en la correlación.

2. ReliefF - **weka.attributeSelection.ReliefFAttributeEval:** método de aprendizaje en instancias para la selección de atributos.

3. SVM - **weka.attributeSelection.SVMAttributeEval**: modelo lineal para la selección de atributos. Se basan en asumir que, en un modelo lineal, los atributos con coeficientes menores son menos relevantes para predecir la clase.

4. CFsubsetEval - **weka.attributeSelection.CfsSubsetEval**: método basado en la correlación. Selección de un conjunto de atributos que se correlacionan bien con la clase y poco entre ellos.

Como método de búsqueda utilizamos **Ranker** en el caso de los métodos de selección de atributos individuales, mientras que para los métodos de selección de conjunto de atributos utilizamos la búsqueda primero el mejor (**BestFirst**).

Incertidumbre simétrica

4 atributos: 1834,4847,1882,3252.

8 atributos: 1834,4847,1882,3252,2288,760,6041,6855.

16 atributos:

1834,4847,1882,3252,2288,760,6041,6855,1685,6376,2354,4373,4377,4366,2402,758.

32 atributos:

1834,4847,1882,3252,2288,760,6041,6855,1685,6376,2354,4373,4377,4366,2402,758,4328, 1144,3320,2642,2335,1829,2128,6281,4229,2020,1779,2121, 4196,1902,1926,1400.

ReliefF

4 atributos: 3252,4196,1779,4847.

8 atributos: 3252,4196,1779,4847,2402,4951,1834,1829.

16 atributos:

3252,4196,1779,4847,2402,4951,1834,1829,6041,2288,1882,6201,1745,3320, 6919,2363.

32 atributos:

3252,4196,1779,4847,2402,4951,1834,1829,6041,2288,1882,6201,1745,3320,6919,2363,2111, 4052,2642,2121,1674,6225,461,1249,4366,2354,2020,6539, 1291,2546,1260,235.

SVM. Debido a que el coste de este tipo de método es elevado, lo que hacemos es reducir n coeficientes por iteración. Para este conjunto de atributos se ha seleccionado **n=100**.

4 atributos: 4196,4951,1779,3847.

8 atributos: 4196,4951,1779,3847,5107,3714,1882,1834.

16 atributos:

4196,4951,1779,3847,5107,3714,1882,1834,5002,1928,4725,5348,3017,1685,6271,1933.

32 atributos:

4196,4951,1779,3847,5107,3714,1882,1834,5002,1928,4725,5348,3017,1685,6271,1933,4922, 2134,1962,4142,3104,5950,1207,1796,1465,5121,538,4381,1941,2410,400,4054.

CFsubsetEval. Como método de búsqueda se ha utilizado una **búsqueda voraz** (GreedyStepwise) hacia delante.

4 atributos: 538,620,683,699.

8 atributos: 538,620,683,699,758,774,885,1087.

16 atributos:

538,620,683,699,758,774,885,1087,1106,1120,1239,1497,1630,1674,1685,1723.

32 atributos:

538,620,683,699,758,774,885,1087,1106,1120,1239,1497,1630,1674,1685,1723,1779,1800, 1829,1834,1882,1904,1926,2020,2111,2128,2141,2223,2288,2354,2441,2458.

Examinar los atributos seleccionados por cada método.

Para el caso de 4 y 8 atributos, se puede observar cómo se comparten una serie de atributos entre los 3 primero métodos utilizados, es decir, los de selección individual. Mientras que con el cuarto método no hay ninguna combinación.

Para el caso de 16 atributos, sigue habiendo combinaciones de atributos entre los 3 primeros métodos, pero ahora también nos encontramos con combinaciones con el cuarto método, aunque son reducidas.

Por último, para los 32 atributos, nos encontramos con un aumento en las combinaciones de atributos entre los 3 primeros con el cuarto.

Estimar, mediante validación cruzada de 10 particiones, el error de las hipótesis generadas por los siguientes clasificadores:

- **J48, NB, IBK1, Regresión Logística, MLP (H10), SVM(lineal)**

  - Incertidumbre simétrica

Número de atributos	Validación Cruzada de 10 particiones					
	J48	**NB**	**IBK1**	**Regresión Logística**	**MLP (H10)**	**SVM(lineal)**
4	0,097	0,055	0,083	0,069	0,069	0,069
8	0,153	0,055	0,069	0,055	0,042	0,069
16	0,153	0,042	0,042	0,042	0,014	0,055
32	0,139	0,042	0,042	0,042	0,028	0,028

Tabla 2

  - ReliefF

Número de atributos	Validación Cruzada de 10 particiones					
	J48	**NB**	**IBK1**	**Regresión Logística**	**MLP (H10)**	**SVM(lineal)**
4	0.083	0.083	0.111	0.055	0.055	0.055
8	0.139	0.028	0.055	0.097	0.069	0.055
16	0.153	0.055	0.069	0.069	0.069	0.028
32	0.167	0.042	0.069	0.042	0.028	0.28

Tabla 3

  - SVM

Número de atributos	Validación Cruzada de 10 particiones					
	J48	**NB**	**IBK1**	**Regresión Logística**	**MLP (H10)**	**SVM(lineal)**
4	0.181	0.083	0.042	0	0	0.083
8	0.111	0.014	0.028	0	0.014	0.028
16	0.111	0	0	0	0	0
32	0.111	0	0	0	0	0

Tabla 4

  - CFsubsetEval

Número de atributos	Validación Cruzada de 10 particiones					
	J48	**NB**	**IBK1**	**Regresión Logística**	**MLP (H10)**	**SVM(lineal)**
4	0.139	0.194	0.222	0.153	0.194	0.250
8	0.069	0.069	0.083	0.097	0.042	0.097
16	0.055	0.028	0.055	0.083	0.042	0.028
32	0.111	0	0.014	0	0.014	0.014

Tabla 5

Discusión de resultados

Discutir resultados, primero para cada tabla y luego en conjunto. Sea breve y no incluya suposiciones en la discusión (limitarse a un breve resumen de los resultados).

- Para la tabla 2 se puede observar con claridad que conforme aumenta el número de atributos, la tasa de error disminuye para todos los métodos. Para el método de Incertidumbre Simétrica la mejor tasa de error se obtiene con 16 atributos con el algoritmo: MLP(H10).

- Para la tabla 3 en cambio, no se da el caso anterior para todos los algoritmos ya que se puede ver que a medida que aumentan los atributos también lo hacen las tasas de error. Para el método de ReliefF la mejor tasa de error se obtiene con 32 atributos con los algoritmos: MPL(H10) y SVM.

- Para la tabla 4 volvemos a observar el mismo comportamiento que el de la tabla 2. Para el método SVM la mejor tasa de error se consiguen con 16 y 32 atributos (mismas tasas).

- Para la tabla 5 se observa el mismo comportamiento que el de las tablas 2 y 4, salvo para el caso del atributo J48, en el que a partir de 32 atributos vuelve a subir la tasa de error. Para el método CFsubsetEval la mejor tasa de error se obtiene con 32 atributos con los algoritmos: NB y Regresión Logística.

En general, se nota un descenso de las tasas de error a medida que el número de atributos aumenta, salvo para el caso del método ReliefF como se ha comentado, y que el algoritmo con el que mejor tasa de error se obtienen es MLP(H10).

Selección de atributos: envolvente

Utilizar un método de envolvente y selección hacia adelante para seleccionar los atributos con los siguientes métodos (si el coste computacional se lo permite):

- **J48, NB, IBK1, Regresión Logística, MLP (H10), SVM(lineal)**

Procedemos a realizar una selección de atributos mediante métodos envolventes - **weka.attributeSelection.WrapperSubsetEval** en los cuales el propio algoritmo de aprendizaje proporciona criterio. Estos métodos realizan una búsqueda en el espacio de atributos. Debido al tamaño del espacio de atributos, se suele utilizar un método de búsqueda voraz: **Selección hacia delante** (GreedyStepwise).

Algoritmos	Atributo/s seleccionado
J48	attribute4847
NB	attribute6, attribute461, attribute760 attribute6615
IBK1	attribute28, attribute1834, attribute3258, attribute3549
Regresión Logística	attribute43, attribute1882, attribute6049
MLP(H10)	attribute1795, attribute1834, attribute2288
SVM(lineal)	attribute162, attribute1796, attribute2111 y attribute3252

Examinar la selección de atributos para cada algoritmo.

Se observa que se realiza una mayor selección de atributos para los algoritmos: NB, IBK1 y SVM(lineal) con 4 atributos cada uno. Después tenemos los algoritmos: Regresión Logística y MLP(H10) con 3 atributos. Finalmente, tenemos el algoritmo J48 con un único atributo seleccionado.

Estimar, mediante validación cruzada de 10 particiones, el error de las hipótesis generadas por los siguientes clasificadores:

- **J48, NB, IBK1, Regresión Logística, MLP (H10), SVM(lineal)**

Validación Cruzada de 10 particiones						
	J48	**NB**	**IBK1**	**Regresión Logística**	**MLP (H10)**	**SVM(lineal)**
J48	0.055					
NB		0.014				
IBK1			0			
Regresión Logística				0.014		
MLP(H10)					0.055	
SVM(lineal)						0.028

Comparar también los resultados con los métodos de filtro, tanto las tasas de error obtenidas como los atributos seleccionados. Sea breve y no incluya suposiciones en la discusión (limitarse a un breve resumen de los resultados).

En cuanto a las tasas de error obtenidas a partir de la selección de atributos envolventes, podemos observar que el algoritmo que mejor se comporta es el IBK1 seguido de NB y de la Regresión Logística. En cambio, los que peor se comportan son el J48 y MLP(H10).

Comparación de resultados con los métodos de filtro

En cuanto a los atributos seleccionados, todos los algoritmos presentan al menos 1 atributo que ya fue seleccionado por los métodos de filtro anteriores como, por ejemplo: attribute4847, attribute461, attribute1834, attribute1882, attribute2288 o attribute2111.

En cuanto a las tasas de error obtenidas:

- Para el caso de J48, se obtiene una mejor tasa de error que con los métodos de filtro utilizados anteriormente.
- Para el resto de los algoritmos, las tasas de error obtenidas con métodos envolventes son mejores que las obtenidas con métodos de filtro como: Incertidumbre Simétrica, ReliefF o CFsubsetEval.
- Sin embargo, las mejores tasas de error que se han obtenido han sido con el método de filtro de eliminación recursiva SVM.

PCA

El análisis de componentes es un método clásico para detectar las direcciones principales de los datos. Se utiliza para reducir la dimensionalidad y realizar visualizaciones.

Sea M1 el método que obtiene una menor tasa de error entre los métodos anteriores (en caso de empate, el que selección menos atributos)

Como se ha comentado en el apartado anterior, el método con el que obtenemos mejores tasas de error es el método de filtro SVM con eliminación recursiva. **M1 = SVM**

Denominar n al número de atributos seleccionado por el método M1.

Para este método tenemos 2 conjuntos de datos con las mismas tasas de error, que son con 16 y 32 atributos. Como nos dicen que escojamos el conjunto con menos atributos, seleccionamos el conjunto con 16 atributos. Por lo tanto, **n=16**.

Intentar realizar un análisis de componentes principales del conjunto de datos original.

Volvemos a utilizar como conjunto de datos ALL-AML.arff. Como evaluador seleccionamos **PrincipalComponents** y como método de búsqueda **Ranker** ya que se trata de una selección de atributos individual. Una vez realizado, guardamos el resultado en un nuevo archivo .arff.

Entrenar el algoritmo utilizado por M1 con los n primeros componentes principales. Estimar la tasa de error mediante validación cruzada de 10 particiones.

Seleccionamos los 16 primeros componentes principales que hemos obtenido y aplicamos sobre ese conjunto de datos el algoritmo SVM, ya que es el algoritmo utilizado por M1.

Tasa de error obtenida: **0.125**

Entrenar el algoritmo utilizado por M1 con los 2n primeros componentes principales. Estimar la tasa de error mediante validación cruzada de 10 particiones.

Ahora seleccionamos los 32 primeros componentes principales.

Tasa de error obtenida: **0.153**

Discutir los resultados.

Como se puede observar, la tasa de error obtenida para los n primeros componentes principales es menor que la obtenida para los 2n primeros componentes principales. Esto se puede deber a que los primeros componentes tienen mayores desviaciones estándar, lo que hace que se realicen mejores predicciones de clase.

Conclusiones

Resumir, en menos de 100 palabras, las conclusiones obtenidas.

Por lo general, se han obtenido mejores resultados en los métodos que evalúan subconjuntos de atributos, aunque es cierto que el coste computacional es notoriamente superior. Estos métodos envolventes han proporcionado muy buenas tasas de error, pero el problema es que son propensos al sobreajuste al utilizar el método de aprendizaje como evaluador.

Por su parte, los métodos de filtro proporcionan una ejecución más rápida y son aplicables a una familia de clasificadores mayor. Sin embargo, seleccionan muchos atributos pues las funciones objetivo tienden a ser monótonas.

Por último, tenemos el PCA, el cual se ha comprobado que efectivamente tiene un coste bastante elevado y en el cual, los primeros componentes son los que tienen mayor variaciones por lo que proporcionan mejores aproximaciones a las clases.