

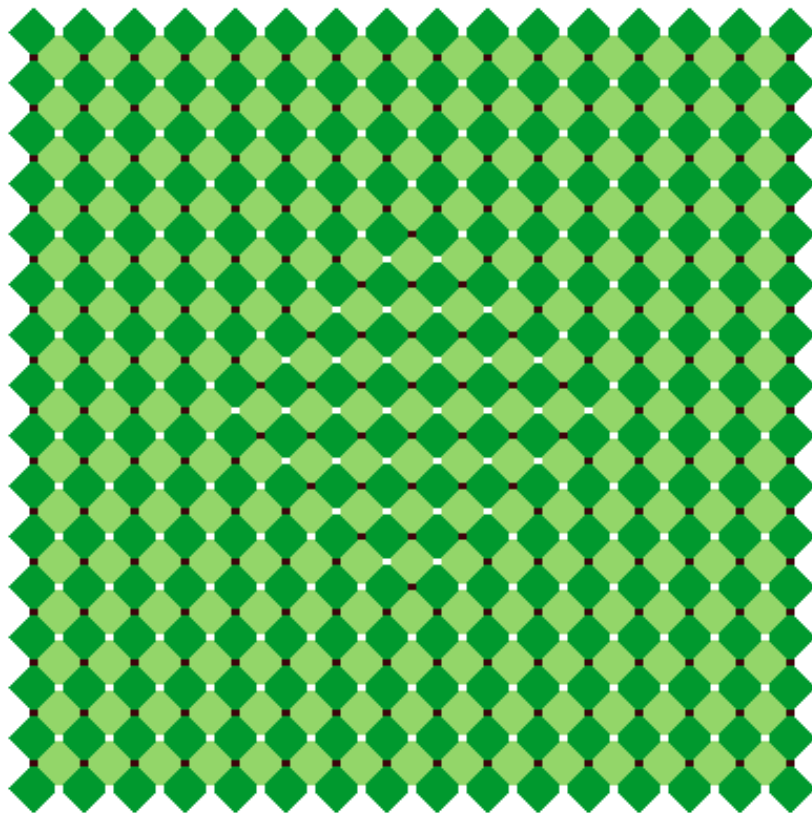
PROGRAMA DE DOCTORADO INTERUNIVERSITARIO

APRENDIZAJE AUTOMÁTICO Y DATA MINING

Práctica 4: **Comparación de métodos de aprendizaje.**

Objetivos:

- Utilización de diversos métodos de aprendizaje automático desde WEKA: (vecino más cercano, árbol de decisión, lista de reglas, naive Bayes y perceptrón multicapa).
 - Comparación de resultados mediante validación cruzada.
-



1. LANZAMIENTO DE DISTINTOS ALGORITMOS DE APRENDIZAJE DESDE WEKA.

Hasta ahora todas las pruebas sobre WEKA se han hecho utilizando siempre árboles de decisión como método de aprendizaje automático.

En este apartado mostraremos la forma de lanzar distintos métodos de aprendizaje y la forma de indicar sus parámetros, utilizando en todos los casos la misma base de datos. La base de datos se encuentra en el directorio de WEKA:

```
c:\iarp\weka_java\data\weather.nominal.arff
```

Se trata del problema que indica si una cierta persona practicará deporte en función de las condiciones atmosféricas. En este caso se utiliza la versión de la base de datos en la que todos los atributos son discretos (nominales) y se utiliza precisamente esta versión para que se puedan lanzar todos los algoritmos de aprendizaje.

Los métodos que probaremos serán los siguientes:

A. Árbol de decisión

Ya se conoce de anteriores prácticas. Se lanza desde MS-DOS (o desde Matlab anteponiendo '!') con el siguiente comando:

```
java weka.classifiers.trees.J48 -t data/weather.nominal.arff -C 0.30
```

...y el principal parámetro a ajustar es el umbral de confianza para la poda, que en este caso se ha ajustado a 0.30:

```
-C 0.30
```

El resultado se muestra de la forma siguiente (se elimina parte común a otros algoritmos):

```
J48 pruned tree
-----
outlook = sunny
|  humidity = high: no (3.0)
|  humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
|  windy = TRUE: no (2.0)
|  windy = FALSE: yes (3.0)

Number of Leaves :    5
Size of the tree :    8
```

B. Vecino más cercano

El comando necesario para utilizar el método del vecino más cercano es el siguiente:

```
java weka.classifiers.lazy.IBk -t data/weather.nominal.arff -K 3
```

Donde el parámetro indica el número de vecinos considerado, que se ha fijado a 3:

```
-K 3
```

Y dado que este método no genera modelo propiamente dicho, el resultado no ofrece apenas información (salvo la común al resto de algoritmos):

```
IB1 instance-based classifier  
using 3 nearest neighbour(s) for classification
```

C. Listas de reglas

Existen diversos algoritmos de generación de listas de reglas. Utilizaremos el método Prism, que no admite ningún parámetro:

```
java weka.classifiers.rules.Prism -t data/weather.nominal.arff
```

El resultado es el siguiente:

```
Prism rules  
-----  
If outlook = overcast then yes  
If humidity = normal  
  and windy = FALSE then yes  
If temperature = mild  
  and humidity = normal then yes  
If outlook = rainy  
  and windy = FALSE then yes  
If outlook = sunny  
  and humidity = high then no  
If outlook = rainy  
  and windy = TRUE then no
```

D. Naive Bayes

El algoritmo Naïve Bayes se lanza de la forma siguiente:

```
java weka.classifiers.bayes.NaiveBayes -t data/weather.nominal.arff  
java weka.classifiers.bayes.NaiveBayes -t data/weather.nominal.arff -K
```

Donde el parámetro -K (si está presente) indica que se estiman funciones de distribución de probabilidad para los atributos mediante el método kernel (suma de gaussianas) en lugar de considerar que cada atributo sigue una distribución gaussiana.

El resultado mostrado en pantalla es el siguiente:

```
Naive Bayes Classifier

Class yes: Prior probability = 0.63

outlook: Discrete Estimator. Counts = 3 5 4 (Total = 12)
temperature: Discrete Estimator. Counts = 3 5 4 (Total = 12)
humidity: Discrete Estimator. Counts = 4 7 (Total = 11)
windy: Discrete Estimator. Counts = 4 7 (Total = 11)

Class no: Prior probability = 0.38

outlook: Discrete Estimator. Counts = 4 1 3 (Total = 8)
temperature: Discrete Estimator. Counts = 3 3 2 (Total = 8)
humidity: Discrete Estimator. Counts = 5 2 (Total = 7)
windy: Discrete Estimator. Counts = 4 3 (Total = 7)
```

Se muestran las probabilidades a priori de las clases y también las estadísticas para cada atributo. Con atributos discretos (como es el caso en este ejemplo, la opción -K no tiene ningún efecto).

E. Perceptrón multicapa

En WEKA las redes neuronales (perceptrón multicapa) se utilizan como otro tipo de clasificador cualquiera. Como ejemplo, se generará desde Matlab una red neuronal para uno de los ficheros de ejemplo presentes en WEKA:

```
>> !java weka.classifiers.functions.MultilayerPerceptron -H 2 -t
data/weather.nominal.arff
```

El parámetro -H indica el número de neuronas en la capa oculta (se utiliza una única capa oculta con 2 neuronas): -H 2

El resultado debe mostrar un aspecto similar al siguiente:

```
Sigmoid Node 0
  Inputs  Weights
Threshold -3.925551968222202
Node 2    4.627849386939848
Node 3    4.588578266461385
Sigmoid Node 1
  Inputs  Weights
Threshold 3.9257068267317576
Node 2    -4.57920864729093
Node 3    -4.637545979236874
```

```

Sigmoid Node 2
  Inputs      Weights
  Threshold   0.14255030544892583
  Attrib outlook=sunny    -2.0506778243062564
  Attrib outlook=overcast  2.461148720592726
  Attrib outlook=rainy    -0.6479794593243129
  Attrib temperature=hot  -0.5222180497188509
  Attrib temperature=mild  1.6260358823527061
  Attrib temperature=cool -1.3484554795851704
  Attrib humidity    4.385557582632426
  Attrib windy      2.9025436555132296
Sigmoid Node 3
  Inputs      Weights
  Threshold   0.1660359095228123
  Attrib outlook=sunny    -2.0601099108985923
  Attrib outlook=overcast  2.4585455551638904
  Attrib outlook=rainy    -0.653642662283134
  Attrib temperature=hot  -0.49337549460881014
  Attrib temperature=mild  1.6697630527458742
  Attrib temperature=cool -1.311797657300884
  Attrib humidity    4.395649912484702
  Attrib windy      2.9093819614516385
Class yes
  Input
  Node 0
Class no
  Input
  Node 1

```

Entre los resultados de WEKA se muestran los pesos asignados durante el entrenamiento a cada una de las conexiones entre neuronas. Se trata de un proceso más lento y más costoso computacionalmente que el resto de algoritmos probados hasta el momento.

2. COMPARACIÓN BÁSICA DE ALGORITMOS.

La forma más utilizada para comparar el funcionamiento de diversos algoritmos de aprendizaje es evaluar el porcentaje de clasificaciones correctas en una prueba de validación cruzada. Este dato lo devuelve siempre WEKA entre la información que genera al lanzar cada algoritmo:

```

=== Stratified cross-validation ===

Correctly Classified Instances      8    57.1429 %
Incorrectly Classified Instances    6    42.8571 %

```

... y es posible leer el dato manualmente o automáticamente utilizando el programa lee_weka.m de la práctica 2.

A ENTREGAR: EJERCICIO NÚMERO 1

Se trabajará sobre un sistema de cálculo de puntos de contacto para el agarre de objetos mediante un robot. y una pinza de dos dedos Cada posible par de puntos de contacto se clasifica en función de una serie de medidas de distancias y ángulos. Los valores que pueden tomar los atributos y la clase son los siguientes:

ATRIBUTOS		CLASE
Distancia al centro de gravedad	Ángulos respecto de la normal (diez medidas)	Agarre válido
MUY_CERCA	NULO	SI NO
CERCA	AGUDO	
LEJOS	RECTO	
MUY_LEJOS	OBTUSO	
	PLANO	

El fichero con los datos de entrenamiento está disponible en la página web de la asignatura:

<http://isa.umh.es/isa/es/asignaturas/aprendizaje/robot.arff>

Se deberán comparar los resultados de clasificación de los siguientes métodos:

- Árbol de decisión con cinco valores distintos de confianza para la poda (parámetro C): 0.005, 0.01, 0.1, 0.2, 0.3.
- Vecino más cercano con cinco valores distintos para el número de vecinos (parámetro K): de 1 vecino a 5 vecinos.
- Naive Bayes con estimación de función de densidad gaussiana o suma de gaussianas (parámetro K).
- Lista de reglas.
- Perceptrón multicapa con distinto número de neuronas en la capa oculta: 2, 4, 6, 8, 10.

El resultado debe ser un gráfico como el que se muestra en la figura de la página siguiente, donde Xxxxx Yyyyyy Zzzzzz corresponde al alumno. Aparecen:

- Los cinco resultados obtenidos para el método del árbol
- Los cinco resultados obtenidos para el método del vecino más cercano.
- Los dos resultados obtenidos para el método Naive Bayes.
- El único resultado obtenido para el método de la lista de reglas.
- Los cinco resultados obtenidos con el perceptrón multicapa.

La leyenda sirve para identificar los distintos resultados obtenidos con cada método.

Además del gráfico, se deberán comentar los resultados obtenidos: cuáles son los mejores métodos y qué efecto tienen sobre ellos los parámetros.

Xxxxx Yyyyyy Zzzzzz

