Mario AI: Tutorial 2
Introducción a Weka

Grado en Ingeniería Informática

Aprendizaje Automático

Aitor Alonso Núñez NIA 100346169 Gr. 83
Daniel Gak Anagrov NIA 100318133 Gr. 83

Aitor Alonso Núñez NIA 100346169 Gr. 83
Daniel Gak Anagrov NIA 100318133 Gr. 83

uc3m | Universidad Carlos III de Madrid

Índice

Aitor Alonso Núñez NIA 100346169 Gr. 83
Daniel Gak Anagrov NIA 100318133 Gr. 83

Los ficheros de datos

¿Cuántos atributos de entrada tiene el fichero de datos? ¿De qué tipo son?

El fichero tiene dos atributos de tipo simbólico, pero solo el primero (*name*) es **atributo de entrada.**

¿Podría un algoritmo de aprendizaje automático identificar una función capaz de predecir dicha clase con los datos que hay en el fichero? ¿Por qué?

Sí, en el caso de que se usaran redes de neuronas, habría que adaptar la variable de entrada a una variable numérica (preferiblemente normalizada), y debido a que un perceprón multicapa es un aproximador universal, obtenedremos una función que predice la clase de una variable de entrada en todos los casos. No obstante, el error de la función que predice la clase dependerá de que los datos de entrenamiento son representativos o no, pues en caso de no serlo, signifcaría que la clase con la que se clasifica tiene poca relación con la variable de entrada, por tanto, se obtendrá una función que se adapte a estos datos de entrenamiento, pero que no será capaz de predecir datos nuevos.

Clasificando con ID3

En la pestaña Classify, seleccionar el clasificador trees/ID3 1 . En las Test options seleccionar Use training set y pulsar el botón de Start para que se genere el modelo. ¿Cómo de buenos son los resultados?

El clasificador ID3 es capaz de clasificar correctamente el 100 % de las instancias. Muy probablemente nos hayamos sobreajustado a los datos de ejemplo.

Incompatibilidades de clasificadores

Imagina 4 atributos que te parezca que podrían ser relevantes para este problema. Estos atributos inventados se deberían poder extraer tratando de alguna manera el único atributo de entrada del ejercicio anterior (name). Anótalos y describe en qué consiste cada uno.

A partir del nombre de los sujetos podríamos extraer los siguientes atributos:

- **N_palabras:** número de palabras del nombre del sujeto.
- **N_espacions:** número de espacios del nombre del sujeto.
- **N_vocales:** número de vocales del nombre del sujeto.
- **N_consonantes:** número de consonantes del nombre del sujeto.

Abre el fichero de datos badges1.arff con Weka. ¿Cuántos atributos de entrada tiene el fichero de datos? ¿De qué tipo son?

Tiene ocho atributos de entrada de un total de nueve atributos. En cuanto a los **atributos de entrada** tres son de tipo simbólico (*name, even_odd, first_char_vowel*) y los otros cinco (*length, consonants, spaces, dots, words*) son de tipo numérico.

¿Qué tipo de información estadística se muestra sobre los atributos?

En la esquina inferior derecha de la interfaz de weka observamos distintos diagramas de barras que clasifican de forma visual los atributos de entrada en conjuntos para aquellos que son simbólicos y en rangos para los atributos numéricos.

Pulsa el botón "Visualize All" ¿Qué se muestra?

Se abre una ventana emergente con los diagramas asociados a cada atributo, para una vista rápida y poder comparar a ojo las distribuciones.

Ejecutar el clasificador tree/ID3 sobre estos datos. ¿Qué es lo que ocurre? ¿Qué se podría hacer para evitar este problema con ID3?

No es posible ejecutar el clasificador ID3 sobre estos datos. El motivo es puesto que el algoritmo ID3 ramifica para cada posible valor del atributo de entrada, es imposible ramificar sobre valores continuos (atributos de tipo numérico) ya que los posibles valores que pueden tomar son infinitos. La única solución para evitar este problema pasa por dividir los valores numéricos en subconjuntos acotados.
Por ejemplo: $0 \leq X \leq 10$, $X \in$ [A, B, C].

Solucionando incompatibilidades de atributos

Seleccionar el filtro Filter/unsupervised/attribute/Discretize, fijar el numero de bins a 5 y aplicar al conjunto de datos. ¿Qué efecto tiene este filtro sobre los atributos?

Se han discretizado los valores numéricos. Es decir, tal y como indicamos en el apartado anterior, se han dividido en subconjuntos acotados y ahora sería posible aplicar ID3 sobre nuestros ejemplos.

En la pestaña Classifiers elige ID3 y marca Use training set en las Test options. Vuelve a generar el clasificador. ¿Qué número de instancias del conjunto de entrenamiento clasifica bien? ¿Qué porcentaje clasifica bien?

Clasfica bien 236 instancias, que se corresponde con un 80.27 % del total.

¿Qué significa cada uno de los valores que aparecen en la matriz de confusión?

En la matriz de confusión hemos obtenido los siguientes valores:

Tabla 1: Matriz de confusión ID3

a	b	<– classified as
93	51	a = -
7	143	b = +

Los datos que podemos obtener de esta tabla son los siguientes. Existen un total de $93+51 = 144$ instancias de clase *a*, que se corresponde con la clase definida como "-", de las cuales 93 han sido clasificadas correctamente y las 51 restantes de manera errónea. De la misma forma podemos observar que existen un total de $7+143 = 150$ instancias de clase *b*, definidas como "+", con 143 clasificadas correctamente y 7 de forma errónea.

La suma de las instancias clasificadas correctamente da un total de $93 + 143 = 236$ instancias, las mismas que indicamos en la pregunta anterior.

Pulsar el botón de More Options y seleccionar la opción Output Predictions (PlainText). Volver a clasificar y revisar los resultados. ¿Cuál es la primera instancia del conjunto de entrenamiento que se clasifica mal? ¿Qué atributos exactamente han causado que esta instancia se haya clasificado mal?

La primera instancia que se clasifica mal es la número 7, pues es de clase "-" y se ha predecido clase "+" para la misma. Esta instancia se corresponde con "Andrey Burago". El que el nombre tenga un solo espacio ha derivado en un subárbol en el que todos los nodos hojas indicaban clase "+" para el valor instancia.

¿Cómo se clasificaría la instancia "Carmen Machi"? ¿Cuáles son los atributos de este nombre? ¿Qué ocurre con los valores de esta instancia si utilizas el filtro usado anteriormente?

El nombre "Carmen Machi" tiene los siguientes valores:

- **length:** 12
- **even_odd:** 0
- **first_char_vowel:** 0
- **consonants:** 7
- **spaces:** 1
- **dots:** 0
- **words:** 2

Lo que produce la clasificación:
length = '(10.6-14.2]' $\rightarrow$ *consonants = '(5.6-8.2]'* $\rightarrow$ *dots = '(-inf-0.4]'* $\rightarrow$ *even_odd = 0* $\rightarrow$ *spaces = '(-inf-1.4]'* $\rightarrow$ *first_char_vowel = 0* $\rightarrow$ *"+"*.
Es decir, la instancia "Carmen Machi" quedaría clasificada como clase "+".

Comparando con ZeroR

¿Qué modelo genera el clasificador ZeroR?

Como su nombre indica, es un clasificador sin reglas (ZeroR = ZeroRules), por lo que clasifica todas las instancias en una misma clase, concretamente en la clase mayoritaria (la clase con más instancias en los ejemplos supervisados).

¿Cuál es el porcentaje de éxito de este modelo?

Este modelo clasifica con éxito 150 instancias que corresponden con un 51.0204 % del total.

¿Cómo se clasificaría la instancia "Carmen Machi"?

Lo clasificaría como "+", al igual que todas las instancias.

Clasificando con J48 (C4.5)

¿Cuántas hojas tiene el árbol generado con J48?

El árbol generado con J48 tiene 20 hojas.

¿Cuántas instancias del conjunto de entrenamiento clasifica bien?

Clasifica bien 287 instancias.

¿Qué porcentaje de instancias clasifica bien?

Estas 287 instancias se correponden con un 97.619 % del total.

¿Cuántas instancias de cada tipo se han clasificado mal?

En la siguiente matriz de confusión podemos observar que se han clasificado 4 instancias de *a* (clase "-") como *b* (clase "+"), y 3 instancias de *b* (clase "+") como *a* (clase "-"), sumando un total de 7 instancias mal clasificadas, que corresponde a un 2.381 % del total.

Tabla 2: Matriz de confusión J48

a	b	<– classified as
140	4	a = -
3	147	b = +

¿Cómo se clasificaría la instancia "Carmen Machi"?

El nombre "Carmen Machi" tiene los siguientes valores:

- **length:** 12
- **even_odd:** 0
- **first_char_vowel:** 0
- **consonants:** 7
- **spaces:** 1
- **dots:** 0
- **words:** 2

Lo que produce la clasificación:
length <= 13 $\rightarrow$ *consonants <= 7* $\rightarrow$ *length <= 12* $\rightarrow$ *consonants >6* $\rightarrow$ *"+"*.
Es decir, la instancia "Carmen Machi" quedaría clasificada como clase "+".

¿Elegirías este modelo o el generado por ID3? ¿Por qué?

Elegiríamos este modelo, pues J48 clasifica bien más instancias (287) que ID3 (236).

¿Hemos encontrado una función exacta para generar las etiquetas? ¿Por qué?

Esta función no es exacta, porque ni siquiera es capaz de clasificar el 100 % de los datos de entrenamiento correctamente.

Utilizando más atributos con J48 (C4.5)

Volver a la pestaña de preproceso y seleccionar el filtro Filter/unsupervised/attribute/AddExpression para generar un nuevo atributo que calcule el número de vocales. ¿Cuál es la función que has programado en el filtro?

Actualmente conocemos la longitud del nombre de cada instancia, así como el número de consonantes, espacios y puntos que contiene. Para calcular el número de vocales, nos valemos de estos atributos para generar una nueva exprisión. Esta expresión queda $vowels = length - consonants - spaces - dots$. La expresión tal cual la hemos introducido en el filtro es $a2 - a5 - a6 - a7$.

¿Podrías decir cuál es el rango de vocales más común en el fichero proporcionado?

La media obtenida es de 4,643 por lo que el número de vocales más común está en torno a las 4 o 5 vocales por nombre.

Anota el porcentaje de instancias bien clasificadas y la matriz de confusión.

El clasficador J48 ha clasificado correctamente el 100 % de las instancias, un total de 294. De esta forma, la matriz de confusión queda:

Tabla 3: Matriz de confusión J48

a	b	<– classified as
144	0	a = -
0	150	b = +

Haz click con el botón derecho del ratón en el modelo generado que aparece en Result list. Visualiza el árbol generado con Visualize Tree. ¿Qué indican los números que aparecen en las hojas?

Los números de cada hoja indica el número de instancias que han sido clasificadas al llegar a dicha hoja. En este caso, se han clasificado 150 instancias como clase “+” por tener cuatro vocales o menos, y se han clasificado 144 instancias como clase “-” por tener más de cuatro vocales.

Ir a la pestaña Visualize. Hacer click en la gráfica que relaciona el atributo creado con la clase y aumentar el valor de Jitter. ¿Qué efecto tiene?

Se muestran dos nubes de puntos diferenciadas en extremos opuestos de la gŕafica.

Tras todos estos resultados, ¿qué características o cualidades crees que deben tener los atributos para maximizar el éxito de los algoritmos de aprendizaje automático?

Los atributos de una instancia tienen que tener la máxima correlación posible con la clase en la que queremos clasificar misma. Asimismo, cuantos más atributos tenga una instancia más probabilidades hay de encontrar el atributo o atributos con mayor correlación que nos permitan generar el mejor clasificador posible.

Balanceado de datos, selección de características y otros filtros

¿Cuántos atributos de entrada tiene este fichero? ¿Cuántas instancias de entrenamiento?

El fichero posee 15 atributos, de los cuales la entrada se determinará según cual sea nuestro objetivo. Si suponemos que nuestra tarea es crear un modelo predictorio de salarios, entonces habrá 14 atributos de entrada. Este data set posee 32.561 instancias.

Ejecuta el clasificador J48. Selecciona en Test Options la opción “Cross-validation” ¿Qué resultados aparecen?

Weka nos devuelve como resultado:

- **Información de ejecución:** Parámetros utilizados en la ejecución

- **Modelo clasificador:** Modelo clasificador resultante tras la unión de resultados de la validación cruzada, con 564 hojas.
- **Resumen:** Datos estadísticos sobre la clasidicación. Donde se han clasificado correctamente 28.071 instancias, que corresponden a un 86,2105 % del total de instancias.
- **Matriz de confusión:** Datos sobre la clasificación entre las diferentes clases. Donde se han clasificado incorrectamente 4.490 instancias, un 13,7895 % del total de instancias.

Ahora vamos a evaluar el clasificador solamente con las instancias que figuren en el fichero adult-test.arff. Para ello selecciona en Test Options la opción "Supplied test set". ¿Qué resultados aparecen?

Weka nos devuelve como resultado:

- **Información de ejecución:** Parámetros utilizados en la ejecución
- **Modelo clasificador:** Modelo clasificador resultante tras la unión de resultados de la validación cruzada, con 564 hojas.
- **Resumen:** Datos estadísticos sobre la clasidicación. Donde se han clasificado correctamente 13.977 instancias, que corresponden a un 85,8485 % del total de instancias.
- **Matriz de confusión:** Datos sobre la clasificación entre las diferentes clases. Donde se han clasificado incorrectamente 2.304 instancias, un 14,1515 % del total de instancias.

¿Por qué los resultados difieren al emplear la opción de cross-validation o la de supplied test set, si usamos el mismo clasificador?

Usamos el mismo clasificador, pero no el mismo set de datos, pues cuando se realiza validación cruzada por 10 Folds, las instancias se parten en 10 grupos y se generan 10 modelos diferentes. De esta forma se generan datos estadísticos para cada modelo, obteniendo un modelo resultante como combinación de los anteriores.

Por otro lado, la opción de supplied test set lo único que hace es construir el modelo normalmente con el set entero de datos, pero el testeo lo realiza con los datos que propocionas, que al ser el mismo fichero que el de los datos de entrenamiento equivale a utilizar la opción use training set en este caso.

Vuelve a la pestaña Preprocess y haz click en el atributo de salida (la clase). ¿Qué proporción de datos hay de cada clase? ¿Crees que este porcentaje es apropiado para que un algoritmo de aprendizaje automático aprenda bien?

Existen 7.841 instancias cuyo salario es mayor a 50k (24,0809 %) y 24.720 instancias cuyo salario es menor o igual a 50k (75,9190 %). Esto no es apropiado para que un algoritmo de aprendizaje automático aprenda bien, pues cuanto más desequilibrados sea el número de instancias de una clase, mayor sesgo va a tener el modelo resultante del aprendizaje del algoritmo.

¿Qué ocurre con el atributo de salida? ¿Ha descendido el número de ejemplos de entrenamiento?

Se ha generado un subdataset aleatorio con ejemplos de las clases, reduciendo instancias de la clase que tenia más instancias, y duplicando instancias que tenía la clase con menos instancias. Como resultado, se ha igulado el número de instancias de cada clase en el atributo de salida, obteniendo 16.280 instancias de cada clase, que corresponde a un 50 % del total. El número de ejemplos de entrenamiento ha descendido en una unidad, de 32.561 a 32.560.

Este proceso en principio no debe de disminuir el número de instancias de los ejemplos de entrenamiento, pero ha tenido que eliminar una instancia (lo cual es admisible) para que el numero de estas sea par, y así poder hacer un equilibrio perfecto entre en número de las mismas repartidas entre las dos clases.

Tras aplicar este filtro, evalúa de nuevo con cross-validation y supplied test set el algoritmo J48. ¿Qué resultado ofrece ahora el algoritmo? ¿Ha mejorado o empeorado?

Weka nos devuelve como resultado para *supplied test set* un 80,5663 % de instancias clasificadas correctamente, y un 87,1898 % de instancias bien clasificadas con *cross-validation* con 10 Folds. Podemos observar que para este último los resultados han mejorado respecto a la clase no equilibrada, pues como se ha mencionado anteriormente, aunque eliminemos algunos ejemplos de aprendizaje, el modelo resultante tiene menos sesgo que si utilizáramos la clase desequilibrada.

Por último aplica el filtro de normalización unsupervised/instance/Normalize para los atributos numéricos. ¿Qué resultados se obtienen?

Se normalizan estos atributos, de esta forma en un mismo atributo todos sus valores rondarán entre 0 y 1, siendo 0 el valor mínimo y 1 el valor máximo del atributo.

Después del procesamiento de datos que has realizado en este apartado, ¿crees que esto ayuda al proceso de aprendizaje? ¿Por qué?

Sí, esto se debe a que algunos modelos (como las Redes de Neuronas) son modelos meramente matemáticos, por tanto, interesa que las entradas estén un mismo rango de valores. Es decir, no interesa tener dos atributos de los cuales uno tenga valores entre 500 y 1000, y otro de 1 a 5, pues puede complicar el proceso de aprendizaje. Al normalizar estos atributos, tienes como resultado valores que mantien una misma proporción, pero que están en un rango determinado.

¿Cuál es el mejor resultado obtenido? ¿Por qué?

Hemos ejecutado nuevamente el clasificador J48 con *supplied test set* y con *cross-validation*. Para el primero hemos conseguido clasificar bien el 89,3646 % de las instancias (frente al 80,5663 % anterior). En el caso del *cross-validation* hemos clasificado correctamente el 84,807 % de las instancias, empeorando nuestro resultado anterior del 87,1898 %.

El resultado ha mejorado para *supplied test set* fente a *cross-validation* porque este último, al realizar diez modelos para obtener el modelo definitivo, parece que en varias iteraciones ha cogido más de una vez las mismas instancias repetidas, lo que ha producido en alguna iteración un modelo érróneo que se ha propagado.