Árboles de Decisión y Reglas

García Prado, Sergio
sergio@garciparedes.me

27 de marzo de 2017

Resumen

En este documento se realizan distintos experimentos basados en Holdout *para analizar el comportamiento de algoritmos de aprendizaje basados en reglas y compararlos con los basados en generación de árboles de decisión. Dicha tarea se lleva a cabo sobre conjuntos de datos de distinta naturaleza.*

1. Introducción

En este documento se exponen los resultados obtenidos de realizar un conjunto de experimentos sobre varios conjuntos de datos de distintas características sobre algoritmos de aprendizaje automático basados en aprendizaje supervisado. Para ello se ha utilizado la suite de aprendizaje automático **Weka** [too], la cual ha sido es desarrollada por la *Universidad de Waikato*, Nueva Zelanda.

A partir de dichos experimentos se realiza una comparación entre estrategias de aprendizaje basadas en *árboles de decisión* frente a basadas en *generación de reglas*. En las subsecciones 1.1 y 1.2 se describen los *Algoritmos de Aprendizaje* y los *Conjuntos de Datos*. Seguidamente, en la sección 2 se exponen los resultados de realizar un experimento de *Holdout* sobre 4 de los conjuntos de datos, y en la sección 3 se realiza el mismo experimento, pero esta vez sobre un conjunto de datos cuya partición de instancias de entrenamiento y test viene dada a priori tal y como se explicará. Por último, en la sección 4 se realiza un breve comentario acerca de los resultados obtenidos.

1.1. Algoritmos de Aprendizaje

Tal y como se ha dicho anteriormente, los algoritmos de aprendizaje utilizados se corresponden con aprendizaje basado en árboles de decisión y basados en reglas. En primer lugar se describe el algoritmo *J48*:

- **J48**: Es la implementación en Java de *C4.5*, un método de generación de árboles de decisión basado en la *Teoría de la Información*. En cada iteración trata de maximizar la ganancia de información producida tras cada partición con respecto de la clase de destino. Además, proporciona otras mejoras como *poda de ramas* para evitar el sobreajuste, el uso de *valores continuos* o el tratamiento de *valores desconocidos*.

Una vez descrito el algoritmo utilizado para representar los algoritmos basasados en generación de árboles de decisión, se describe el caso de los basados en reglas. En este caso son *1R*, *PRISM*, *JRIP* y *PART*:

- **1R**: Es uno de los métodos más simples de generación de reglas. Se basa en la generación de un conjunto de reglas a partir de un único atributo. Por tanto, genera un árbol de profundidad 1. Es uno de los métodos más simples y de menor coste computacional, lo cual presenta una gran diferencia en rendimiento con respecto a alternativas más complejas en los casos en que el conjunto de datos posee una estructura muy simple.

- **PRISM**: Es un algoritmo de *aprendizaje basado en reglas* básico cuya intuición se basa en el recubrimiento secuencial del espacio de búsqueda. Es equivalente a *ID3* en el caso de los árboles de decisión puesto que su uso está restringido a conjuntos de datos con atributos discretos y sin valores desconocidos. Una extensión mejorada del mismo que si permite su uso en dichos casos es *RIPPER*

- **JRIP**: Es la implementación en Java de *RIPPER*, un método de aprendizaje supervisado basado en reglas cuyas siglas significan "*Repeated Incremental Pruning to Produce Error Reduction*", lo que puede entenderse como la eliminación de reglas que se cumplen con pocas instancias para reducir el sobreajuste producido en la fase de aprendizaje, que genera todo el conjunto de reglas posibles a partir de una determinada heurística.

- **PART**: Es un algoritmo similar a *RIPPER*, solo que en este caso genera el conjunto de reglas a partir de árboles podados previamente, lo cual evita la realización de una poda global. Utiliza conjuntamente técnicas de "separa y vencerás" junto con "divide y vencerás".

1.2. Conjuntos de Datos

El siguiente paso es describir el los conjuntos de datos que se han utilizado para la realización de los distintos experimentos. Esto se llevará a cabo en dos partes. En primer lugar se describen los conjuntos de datos predefinidos en la suite de aprendizaje automático *Weka*:

- **Iris**[data]: Está formado por *150 instancias* formadas por *4 atributos*, todos ellos de carácter real. La clase de destino puede tomar *3 valores* distintos. El conjunto de datos se corresponde con instancias referidas a atributos de la especie de plantas *Iris* y la clase de destino representa una subcategoría de la misma.

- **Labor**[datb]: Está formado por *57 instancias* formadas por *16 atributos* de los cuales, 8 de ellos son de tipo numérico mientras que el resto son de carácter nominal. La clase de destino puede tomar *2 valores* distintos. El conjunto de datos se corresponde con resultados de negociaciones industriales en Canadá.

- **Soybean**[datc]: Está formado por *683 instancias* formadas por *35 atributos*, todos ellos de carácter nominal. La clase de destino puede tomar *19 valores* distintos. El conjunto de datos se corresponde con instancias referidas a atributos de plantas y la clase de destino representa el tipo de planta.

- **Weather**[datd]: Está formado por *13 instancias* formadas por *4 atributos*, todos ellos de carácter nominal. La clase de destino puede tomar *2 valores* distintos. El conjunto de datos se corresponde con un conjunto de instancias referidas a características climatológicas que sirve para predecir si es posible jugar al tennis en dichas condiciones.

El conjunto de datos restante posee una característica diferenciadora del resto. En este caso se suministra divido en dos ficheros, de los cuales uno representa las instancias que se deben utilizar para entrenamiento mientras que el segundo se corresponde con los casos de test:

- **Image Segmentation**[UCI]: Está formado por *210 instancias* de entramiento y *2100 instancias* destinadas a test. Están formadas por *19 atributos*, todos ellos de carácter real. La clase de destino puede tomar *7 valores* distintos. El conjunto de datos se corresponde con un conjunto de instancias referidas a características de imágenes que pretenden determinar el contenido de las mismas.

2. Experimento *Holdout* $\frac{2}{3}/\frac{1}{3}$ sobre *Iris*, *Labor*, *Soybean* y *Weather*

El método de *Holdout* consiste en el particionamiento del conjunto global de datos en 2 subconjuntos. Dicho método de experimentación requiere como entrada el porcentaje de datos que se utilizará para la tarea de entrenamiento, del cual se deriva el que se utilizará para test. En este caso se ha decidido utilizar $\frac{2}{3}$ del conjunto de datos para entrenamiento y $\frac{1}{3}$ para test. El método de selección que utiliza *Holdout* para seleccionar las instancias que formarán cada conjunto es la *selección aleatoria sin reemplazamiento*.

Los resultados obtenidos tras realizar el experimento descrito en el párrafo anterior se muestran en la tabla 1. Debido a la similitud en cuanto a los resultados obtenidos en este experimento y los obtenidos en la sección 3 se ha decidido realizar un comentario acerca de los mismos en la secciónn 4 destinada a la conclusión.

Holdout $2/3, 1/3$					
Datos	Tasa de Error				
	J48	*1R*	*PRISM*	*JRIP*	*PART*
Iris	3,9216 %	3,9216 %	—	7,8431 %	3,9216 %
Labor	10,5263 %	15,7895 %	—	10,5263 %	21,0526 %
Soybean	9,4828 %	60,7759 %	—	8,6207 %	9,9138 %
Weather	60,0 %	60,0 %	40,0 %	60,0 %	60,0 %

Tabla 1: *Tasas de error obtenidas mediante la metodología experimental* Holdout $2/3, 1/3$

3. Experimento sobre *Image Segmentation*

En este caso, el experimento es similar a la realización de un *Holdout estratificado*. La razón por la cual en este caso se denomina estratificado es consecuencia de la descripción que acompaña al conjunto de datos. En la misma se especifica que tanto en el conjunto de datos de entrenamiento como de prueba se presenta la misma proporción de instancias pertenecientes a cada clase. El *Holdout* se ha realizado siguiendo una partición $210/2310, 2100/2310$. Nótese que el conjunto de datos de prueba es aproximadamente del orden de 10 veces más grande que el de entrenamiento. Esto es algo poco común en tareas experimentales, donde se suele seguir la regla $2/3, 1/3$, pero a partir de la intuición se puede comprobar que dicha estrategia es más semejante a un caso real, en el cual es más difícil poseer más instancias de entrenamiento que las que las que se utilizarán en la realidad.

Los resultados obtenidos tras realizar el experimento descrito en el párrafo anterior se muestran en la tabla 2. Debido a la similitud en cuanto a los resultados obtenidos en este experimento y los obtenidos en la sección 2 se ha decidido realizar un comentario acerca de los mismos en la secciónn 4 destinada a la conclusión.

Holdout Estratificado					
Datos	Tasa de Error				
	J48	*1R*	*PRISM*	*JRIP*	*PART*
Image Segmentation	9,0 %	42,5714 %	—	15,7143 %	10,4286 %

Tabla 2: *Tasas de error obtenidas sobre el conjunto de datos* Image Segmentation

4. Conclusiones

Tras la realización de los distintos experimentos y el análisis de los resultados contenidos en las tablas 1 y 2 se pueden apreciar las siguientes peculiaridades: *a*) el algoritmo *J48* es quien mejores resultados obtiene en promedio, siendo gravemente penalizado en el caso de *Weather* debido al escaso número de instancias de entrenamiento y la poda de ramas, *b*) el algoritmo *1R* presenta los peores resultados en promedio con respecto al resto de alternativas, sin embargo destaca sobre el conjunto de datos *Iris* obteniendo la misma tasa de error que *J48*, *c*) en el caso de *PRISM* y debido a su simplicidad, que no permite la entrada de atributos continuos, tan solo es posible su utilización en el caso del conjunto de datos *Weather*, en el cual presenta los mejores resultados, *d*) el algoritmo *JRIP* obtiene tasas de error similares a su homónimo basado en árboles de decisión (*J48*) pero con tasas de error algo peores en promedio y *e*) el algoritmo *PART* otiene resultados aceptables exceptuando el caso del conjunto de datos *Labor*, para el cual se presenta como la peor alternativa, y *Weather* por razones similares al resto de estrategias de aprendizaje (conjunto de entrenamiento demasiado pequeño junto con poda).

Tras el análisis de los resultados obtenidos, conviene remarcar que la elección entre las distintas estrategias de aprendizaje no es una tarea arbitraria, sino que depende de muchos factores como la estructura del conjunto de datos, la cantidad de instancias de entrenamieno que se posean, las limitaciones computacionales donde se pretenda instaurar el sistema o la tasa de error admisible en la clasificación de resultados.

Referencias

[CCAG17] Teodoro Calonge Cano and Carlos Javier Alonso GonzáLez. Técnicas de Aprendizaje Autómatico, 2016/17.

[data] Iris Data Set. http://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/iris.arff.

[datb] Labor Data Set. http://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/labor.arff.

[datc] Soybean Data Set. http://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/soybean.arff.

[datd] Weather Nominal Data Set. http://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/weather.nominal.arff.

[GP17] Sergio García Prado. Árboles de decisión y reglas. https://github.com/garciparedes/machine-learning-decision-trees-and-rules, 2017.

[too] Weka. http://www.cs.waikato.ac.nz/ml/weka/.

[UCI] UCI Machine Learning Repository. Image Segmentation Data Data Set. http://archive.ics.uci.edu/ml/datasets/Image+Segmentation.