Árboles de Decisión y Reglas

García Prado, Sergio
sergio@garciparedes.me

27 de marzo de 2017

Resumen

[TODO]

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.

El motivo de dichos experimentos es la comparación entre estrategias de aprendizaje basadas en *árboles de decisión* contra estrategias basadas en *generación de reglas*. En las subsecciones 1.1 y 1.2 se describen respectivamente los *Algoritmos de Aprendizaje* y los *Conjuntos de Datos* utilizados. Seguidamente, en la sección 2 se exponen los resultados de realizar un experimento de *Holdout* sobre 4 de los conjuntos de datos. Seguidamente, 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**:
- **PRISM**:

- **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**:

1.2. Conjuntos de Datos

[TODO]

- **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.

[TODO]

- **Image Segmentation**[UCI]:

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.

Holdout 2/3, 1/3 Repetido					
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 mediante la metodología experimental* Holdout 2/3, 1/3

3. Experimento sobre *Image Segmentation*

[TODO]

Holdout 2/3, 1/3 Repetido					
Datos	Tasa de Error				
	J48	*1R*	*PRISM*	*JRIP*	*PART*
Image Segmentation	9,0 %	42,5714 %	—	15,7143 %	10,4286 %

Tabla 2: *Tasas de Error [TODO]*

4. Conclusiones

[TODO]

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. Técnicas de aprendize automático: 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.