Artículo **Revista de Tecnología Informática**
Septiembre 2017 Vol.1 No.2 9-18

Estudio del comportamiento de algoritmos de clasificación según la naturaleza de los datos

CRUZ-GUERRERO, René*†, ALONSO-LAVERNIA, Ma. de los Ángeles', FRANCO-ARCEGA, Anilú', SIMÓN-MARMOLEJO, Isaías''

Universidad Autónoma del Estado de Hidalgo, Carretera Pachuca-Tulancingo, Km 4.5, CP. 42186, Mineral de la Reforma, Hidalgo, México.
'Instituto Tecnológico Superior del Oriente del Estado de Hidalgo, Carretera Apan-Tepeapulco Km 3.5, Colonia Las Peñitas, C.P. 43900, Apan Hidalgo.
''ESSAH, Universidad Autónoma del Estado de Hidalgo, Carretera Cd. Sahagún-Otumba s/n. Zona Industrial, CP. 43990, Tepeapulco, Hidalgo, México.

Recibido Julio 3, 2017; Aceptado Septiembre 14, 2017

Resumen

La clasificación es una técnica de Minería de Datos que se utiliza para averiguar con qué grupo una instancia de datos está relacionada dentro de un determinado conjunto de datos. Actualmente, existe una gran diversidad de algoritmos que ejecutan esta tarea, teniendo cada uno una base teórica distinta. El dilema con el que se enfrenta un usuario al realizar la tarea de clasificación es la de seleccionar el algoritmo que responda con mayor eficacia. Este trabajo presenta un estudio de la aplicación de los algoritmos Naive Bayes, C4.5, Perceptrón multicapa y K-vecinos a 38 conjuntos de datos con diferentes características, de lo cual resultaron algunas reglas que describen patrones de comportamiento en correspondencia con la población tratada. Los resultados de este trabajo proporcionaron un conjunto de criterios, los cuáles son un recurso útil que permite reducir el tiempo dedicado a la selección del clasificador, sobre todo para aquellos usuarios que no tienen dominio sobre cómo trabajan los diferentes algoritmos.

Minería de Datos, clasificación supervisada, eficacia de algoritmos de clasificación

Abstract

Classification is a Data Mining technique that is used to find out with which group an instance of data is related within a given set of data. Currently, there is a great diversity of algorithms that execute this task, each having a different theoretical base. The dilemma faced by a user when performing the classification task is to select the algorithm that responds most effectively. This paper presents a study of the application of the algorithms Naive Bayes, C4.5, Perceptron multi-layer and K-neighbors to 38 data sets with different characteristics, resulting in some rules that describe behavior patterns in correspondence with the treated population. The results of this work provided a set of criteria, which are a useful resource that allows reducing the time devoted to the selection of the classifier, especially for those users who do not have control over how the different algorithms work.

Data Mining, Sspervised classification, efficiency of classification algorithms

Citación: CRUZ-GUERRERO, René, ALONSO-LAVERNIA, Ma. de los Ángeles, FRANCO-ARCEGA, Anilú, SIMÓN-MARMOLEJO, Isaías. Estudio del comportamiento de algoritmos de clasificación según la naturaleza de los datos. Revista de Tecnología Informática 2017, 1-2: 9-18

* Correspondencia al Autor (Correo Electrónico: rencrug@gmail.com)
† Investigador contribuyendo como primer autor.

Introducción

Dentro de las técnicas predictivas, la clasificación es una tarea muy socorrida en cualquier área del conocimiento por su capacidad de identificar automáticamente para un objeto, la clase a la cual está asociado, utilizando para ello el conocimiento que relaciona las características de las instancias con sus respectivas clases (Hernández, Ramírez, & Ferri, 2008).

Para llevar a cabo la tarea de clasificación se han desarrollado diversos algoritmos, los cuales ofrecen soluciones bajo diferentes enfoques como son: árboles de decisión, basados en vecindad, probabilísticos, redes neuronales, entre otros. Sin embargo, la diversidad de clasificadores y la versatilidad de las poblaciones que se estudian en el ámbito real, hace compleja la tarea de seleccionar un algoritmo en específico que se ejecute con mayor eficacia.

En investigaciones realizadas, con la intención de identificar relaciones entre clasificadores y conjuntos de datos, se ha encontrado que algunos algoritmos funcionan mejor que otros para ciertos conjuntos de datos. El presente trabajo tiene el objetivo de estudiar diversos algoritmos y poblaciones de datos para analizar su comportamiento y encontrar relaciones entre ambos aspectos, con el propósito de poder proponer algunas reglas que aunque no sean absolutas permitan realizar una selección del clasificador lo más acertada posible en función de las características de la base de datos que se analiza. Para este estudio se utilizan los algoritmos de clasificación más utilizados en la literatura, mismos que fueron probados con diversas bases de datos de características diferentes, considerando la cantidad de instancias, número de clases, número de atributos, tipo de datos, si es o no balanceada, entre otras.

El resto del documento presenta en la Sección 2 algunos trabajos relacionados a estudios de algoritmos de clasificación, en la Sección 3 se describen los algoritmos utilizados en la experimentación, en la Sección 4 se muestran los resultados obtenidos y en la Sección 5, la discusión de estos. Finalmente, se presentan las conclusiones y trabajo futuro.

Antecedentes

En los últimos años, se han desarrollado algunos trabajos dirigidos al análisis comparativo de algoritmos de clasificación, considerando aspectos como velocidad de ejecución, precisión y tipos de datos tratados.

Akinola y Oyabugbe (2015) realizaron un estudio con los algoritmos Árboles de Decisión (AD), Naive Bayes y Perceptrón multicapa respecto a su velocidad de ejecución, utilizando un solo conjunto de datos. En este estudio se pudo observar que el algoritmo Naive Bayes consumía menos tiempo en ejecutar el proceso de clasificación, sin embargo el haber considerado una sola población, no permitió identificar alguna dependencia entre la velocidad y algunas otras características de la población como por ejemplo la cantidad de instancias o los tipos de atributos. Otro trabajo similar lo realizaron Ashari et al. (2013) con los algoritmos AD, Naive Bayes y k-Vecinos más cercanos (KNN), probandolos con cinco bases de datos. Después de realizar su trabajo experimental, mostraron en sus resultados que el algoritmo AD fue el más rápido, sin embargo estos resultados requieren constatarse con una mayor cantidad de conjuntos de datos, porque con mayores cantidades de instancias o atributos los resultados pueden cambiar.

ISSN-2531-2197

Respecto a estudios realizados para comparar la precisión de los clasificadores dependiendo del conjunto de datos tratados, Entezari et al. (2009) compararon los algoritmos KNN, LogR, Naive Bayes, AD, C4.5, Máquinas de Vectores de Soporte (SVM por sus siglas en inglés Support Vector Machine) y Linear Classifier (LC), tomando en cuenta características como: cantidad de instancias, tipo de atributos y número de atributos discretos o continuos. Para realizar este estudio, los autores generaron 29 conjuntos de datos sintéticos, mismos que se organizaron en cuatro grupos de acuerdo al número de atributos que contenían (3, 5, 7 y 10, respectivamente). Con el fin de tener una diversidad de poblaciones, de cada grupo de datos se consideraron para la creación de los conjuntos distintas cantidades de atributos numéricos y discretos y distintas cantidades de instancias (200, 500, 1000, 3000 y 5000). Los algoritmos que mejor desempeño mostraron fueron KNN, SVM, AD y C4.5, no importando si el número de instancias o de atributos aumentaba. Un comportamiento particular se observó con SVM, quien obtuvo mejor precisión que KNN cuando la cantidad de atributos numéricos es mayor y en caso contrario, KNN trabaja mejor con los discretos. Sólo se analizan dos características (número de instancias y tipo de atributo), omitiendo otras que pueden incidir en los resultados que proporciona el clasificador sobre determinada población.

Otro estudio lo desarrollaron Moran et al. (2009) en donde se procesaron 39 bases de datos, las cuales se agruparon en un total de 12 conjuntos mediante la combinación de tres de sus características: número de instancias, total de atributos y porcentaje de atributos categóricos. Tomando la cantidad promedio del número de instancias de las poblaciones tratadas (286 instancias), los conjuntos se separaron en dos grupos, los que tenían más de esta cantidad de instancias de los que tienen menos o igual.

Cada uno de estos grupos se subdividió en dos grupos de acuerdo al número de atributos, en los que tenían más de 16 y los que tenían menos o igual que 16. Por último, se subdividió cada subgrupo anterior en tres, ahora de acuerdo al número de atributos nominales, con el 100% de atributos de este tipo, con más del 50% y con menos o igual que el 50%. En este estudio sólo se utilizaron clasificadores que se derivan o son variantes del algoritmo Naive Bayes, siendo estos: Averaged One Dependence Estimator (AODE), Tree Augmented Naive Bayes (TAN), BN K2, Genetic Search (BN-GS) y Simulated Annealing (BN-SA). El trabajo se divide en dos etapas, primero se probaron los algoritmos con los distintos conjuntos de datos para verificar cuál funciona mejor y posteriormente, considerando los resultados y con el uso del algoritmo J48 se generó un conjunto de reglas de clasificación que ayudan a seleccionar el mejor clasificador para un conjunto de datos particular. En los resultados de la primera etapa detectaron que el clasificador que brindó mejores porcentajes de precisión fue Tree Augmented Naive Bayes (TAN) para bases de datos con atributos numéricos o combinados y el algoritmo ODE para datos 100% nominales. Respecto a las reglas obtenidas en la segunda etapa, se comprobó con los conjuntos de datos iniciales obteniéndose un 78% de efectividad en su aplicación sobre dichos conjuntos. A pesar de los resultados alcanzados, no se efectúa una validación con nuevos conjuntos de datos y se utilizan solo clasificadores de enfoque probabilístico.

Algoritmos de clasificación

El objetivo de la técnica de clasificación es obtener un valor particular de un atributo a partir de los valores de otros atributos. El atributo a obtener es comúnmente llamado Clase o variable dependiente, mientras que los atributos usados para hacer la predicción se llaman Variables Independientes. Para llevar a cabo esta tarea existen diversos algoritmos de clasificación, a continuación se describen los utilizados en el desarrollo de este trabajo.

ISSN-2531-2197

C4.5

El método C4.5 está basado en árboles de decisión, el cual utiliza la ganancia de información por medio del cálculo de la entropía para medir qué tan bien un atributo separa el conjunto de instancias de acuerdo a sus clases (Quinlan, 1986). A continuación, se explica el algoritmo de dicho método.

La construcción del árbol inicia con un nodo raíz vacío. Después de haberse creado la raíz, se comprueba si las diferentes instancias tienen el mismo valor para el atributo clase, de ser así se obtiene solo un nodo, posteriormente para todos los atributos no clase se verifica mediante el cálculo de entropías cuál es el que proporciona mayor ganancia para ubicarlo en el nodo del nivel más alto del árbol, asignando al nodo el nombre del atributo ganador y a los arcos los valores de dicho atributo. Para el resto de los atributos no clase, esto se realiza de forma iterativa hasta llegar a los nodos hoja.

Cuando se quiere clasificar una nueva instancia, se utilizan sus diferentes atributos (incluyendo sus valores) para recorrer el árbol creado (modelo de clasificación) iniciando por el nodo raíz. El recorrido de los nodos y arcos se efectúan de forma descendente hasta encontrar la hoja buscada (clase).

KNN

El método KNN (por sus siglas en inglés, K Nearest Neighbors) se basa en efectuar aprendizaje por analogía, consiste en realizar una serie de comparaciones para que a una nueva instancia que contiene n atributos se le asigne la clase mayoritaria de sus k vecinos más cercanos (Coomans & Massart, 1982).

Las comparaciones entre las instancias se realizan mediante algún criterio de vecindad definida en términos de algún tipo de métrica, cuando todas las variables son numéricas se aplican métricas como la distancia Euclidiana, de Manhattan, de Chebyshev, entre otras, por otra parte, cuando se tienen tanto variables numéricas como categóricas, se puede aplicar la métrica de Gower (Deza & Deza, 2009).

El proceso inicia especificando los datos de la nueva instancia, posteriormente es importante indicar el número de vecinos a evaluar, así como seleccionar la métrica con la que se calculará la distancia entre las instancias. El siguiente paso consiste en calcular la distancia que existe entre la nueva instancia y las demás, posteriormente se ordenan los resultados de manera ascendente y de sus vecinos más cercanos se verifica cuál es la clase más frecuente para asignarla a la nueva instancia.

Perceptrón multicapa

El Perceptrón Multicapa (MLP por sus siglas en inglés MultiLayer Perceptron) se basa en un algoritmo de propagación hacia atrás para clasificar nuevas instancias con el uso de redes neuronales (Lu & Setiono, 1997). La red neuronal de retro propagación es esencialmente una interconexión de elementos simples de procesamiento que trabajan juntos para producir una salida. El entrenamiento de la red consiste en calcular de manera iterativa un conjunto de pesos para la predicción de la etiqueta de la clase de las instancias analizadas.

Una red neuronal de tipo MLP está formada de una capa de entrada, una o más capas ocultas y una capa de salida. Cada capa de una red neuronal está compuesta por un conjunto de unidades (neuronas). Las entradas a la red corresponden a valores de los atributos de cada instancia usada para entrenar.

Las entradas son alimentadas simultáneamente en las neuronas que forman la capa de entrada y luego se ponderan y alimentan a una segunda capa llamada oculta. Las salidas de las neuronas de la capa oculta se pueden introducir a otra capa oculta, la capa de salida corresponde al resultado esperado.

Naive Bayes

El método Naive Bayes permite usar el conocimiento apriori para predecir una suposición mediante el cálculo de probabilidades. El uso del teorema de Bayes en la tarea de clasificación se debe a que permite calcular las probabilidades a posteriori de cualquier hipótesis consistente con el conjunto de datos de entrenamiento para así escoger la hipótesis más probable (Miquelez, Bengoetxea, & Larranaga, 2004).

El método consiste en calcular tanto la probabilidad de cada clase como la de los diferentes valores de cada variable independiente. Debido a que este clasificador asume que las características son condicionalmente independientes, evita comparaciones de ganancia de información entre combinaciones de variables o atributos.

Máquinas de Soporte Vectorial

El método de Máquinas de Soporte Vectorial (SVM por sus siglas en inglés, Support Vector Machines) se centra en lo que se conoce como Teoría del Aprendizaje Estadístico donde se tiene el objetivo de establecer un margen máximo de separación entre clases (Taylor, 2004).

El método consiste en crear un modelo que permita separar instancias de una clase de otra clase. Cuando las instancias se pueden representar a través de vectores, lo más fácil para separar dos clases es utilizar una línea. Para saber la clase de una nueva instancia dependerá saber de qué lado queda de la línea.

Las SVM tienen como meta encontrar la línea que maximiza la distancia entre dos instancias de cada clase, las instancias usadas para definir la línea se conocen como vectores de soporte. Si el problema se puede separar usando líneas se dice que es linealmente separable, de lo contrario, para resolver el problema MSV lo traslada a una dimensión mayor en la que sí es separable, para lo cual se utiliza una función llamada Núcleo o Kernel. En el algoritmo cinco se muestran de forma resumida los pasos del método.

Trabajo experimental

Con la finalidad de encontrar las particularidades que influyen en la eficacia de los algoritmos de clasificación, se llevó a cabo la ejecución de cuatro algoritmos de este tipo sobre un conjunto de 38 bases de datos y se realizó un análisis de los resultados obtenidos, identificándose comportamientos particulares de desempeño relacionados con las poblaciones en estudio mediante obtención de reglas de clasificación. Posteriormente, se validaron los patrones encontrados con 8 bases de datos distintas.

Aspectos a considerar en el estudio

Para elegir los conjuntos de datos a utilizar en este estudio se consideraron características básicas de las bases de datos como: cantidad de instancias, número de atributos, número de clases y tipos de datos. Adicionalmente, se consideró que existían otras características de las bases de datos que también podrían influir en la elección de un algoritmo de clasificación. Estas características son:

Estructura.- Propiedad de la base de datos que permite especificar si tiene una forma vertical, horizontal o mixta. Vertical se refiere a que tiene muchas instancias y pocos atributos, Horizontal indica lo contrario. Por otra parte, cuando tiene pocos atributos y pocas instancias o muchos atributos y muchas instancias se le denomina Mixta.

ISSN-2531-2197

Multivaluada.- Permite especificar si la mayoría de los atributos nominales tiene más de dos valores.

Completa.- Permite especificar si una base de datos tiene o no datos faltantes.
Balanceada.- Especifica si el conjunto de datos contiene el mismo número de instancias en todas las clases.

Cantidad de atributos numéricos o nominales.- Se detalla la cantidad de variables numéricas o nominales.

Bases de datos y clasificadores utilizados

Se utilizaron 30 bases de datos para la etapa de experimentación y 8 para la de validación, obtenidas de los repositorios de UCI Machine Learning (Lichman, 2013.), portal de WEKA (Hall M. , Frank, Holmes, & Pfahringer, 2009) y portal de Promise (Zwirk & Wigury, 2013). Los algoritmos de clasificación elegidos para este estudio son los descritos en la sección anterior, Árboles de Decisión (C4.5), Naive Bayes, Perceptrón Multicapa y KNN, por ser ellos de los más utilizados en la literatura.

La Tabla 1 muestra los conjuntos de datos utilizados en la experimentación para analizar el comportamiento de los algoritmos elegidos. El valor NumNom, de la característica Tipo de datos, significa que el conjunto de datos tiene atributos de ambos tipos (Numérico y Nominal).

Base de datos	Tipo de datos	Atributos	Numéricos	Nominales	Instancias	Clases	Balanceada	Incompleta	Estructura
Weather_Nom	Nominal	4	0	4	14	2	No	No	Mixta
Car	Nominal	6	0	6	1728	4	No	No	Vertical
Nursery	Nominal	8	0	8	12960	5	No	No	Vertical
Primary_tumor	Nominal	17	0	17	339	22	No	No	Horizontal
Led7	Nominal	7	0	7	200	10	No	No	Vertical
lymphography	Nominal	18	0	18	148	4	No	No	Mixta
Splice	Nominal	60	0	60	3190	3	No	No	Mixta
Breast_Cancer	Nominal	9	0	9	286	2	No	No	Vertical
Spect	Nominal	23	0	23	80	2	No	No	Horizontal
Balance_Scale	Numérico	4	4	0	625	3	No	No	Vertical
Diabetes	Numérico	8	8	0	768	2	No	No	Vertical
Glass	Numérico	9	9	0	214	7	No	No	Vertical
Breast_w	Numérico	9	9	0	699	2	No	No	Vertical
Messidor_features	Numérico	19	19	0	1151	2	Si	No	Mixta
Sonar	Numérico	60	60	0	208	2	Si	No	Horizontal
Iris	Numérico	4	4	0	150	3	Si	No	Vertical
Vehicle	Numérico	18	18	0	846	4	Si	No	Mixta
Waveform_5000	Numérico	40	40	0	5000	3	Si	No	Mixta
Column3	Numérico	7	7	0	310	3	No	No	Horizontal
Ecoli	Numérico	7	7	0	336	6	No	No	Horizontal
Weather_Num	NumNom	4	2	2	14	2	No	No	Mixta
Vowel	NumNom	13	10	3	990	11	Si	No	Mixta
Zoo	NumNom	17	1	16	121	7	No	No	Horizontal
Tae	NumNom	5	3	2	151	3	Si	No	Mixta
Zoo	NumNom	17	1	16	121	7	No	No	Horizontal
Credit-g	NumNom	20	7	13	1000	2	No	No	Horizontal
Bands	NumNom	39	22	17	539	2	No	Si	Horizontal
Credit_Aproval	NumNom	15	6	9	690	2	No	Si	Vertical
Autos	NumNom	25	15	10	205	7	No	Si	Horizontal
Colic	NumNom	22	7	15	368	2	No	Si	Horizontal

Tabla 1 Bases de datos para pruebas

Base de datos	Tipo datos	Atributos	Numéricos	Nominales	Instancias	Clases	Balanceada	Incompleta	Estructura
Vote	Nominal	16	0	16	435	2	No	Si	Horizontal
Monk	Nominal	6	0	6	124	2	Si	No	Horizontal
Letter	Numérico	16	16	0	20000	26	Si	No	Mixta
Sick	NumNom	29	7	22	3772	2	No	Si	Horizontal
Ozone	Numérico	72	72	0	2536	2	No	Si	Horizontal
Segment	Numérico	19	19	0	2310	7	Si	No	Horizontal
Page_Blocks	Numérico	10	10	0	5473	5	No	No	Vertical
Squash_Stored	NumNom	24	21	3	52	3	No	No	Horizontal

Tabla 2 Bases de datos para validación

Etapa de experimentación

Para ejecutar los algoritmos de clasificación se utilizó la herramienta Weka versión 3.6, en un procesador Core i7 de 8 GB de memoria RAM con Windows 7.

Resultados obtenidos

La evaluación de los resultados se hizo usando validación cruzada con 10 particiones, obteniéndose los resultados que se muestran en la Tabla 3, en donde se puede observar de manera resaltada quien de los cuatro algoritmos obtuvo un mejor desempeño para cada conjunto de datos. Resultando AD mejor en 3 conjuntos, Naive Bayes en 6, KNN en 10 y MLP en 11.

ISSN-2531-2197

Base de datos	C4.5 (J48)	Naive Bayes	KNN	MLP
Weather_Nom	50%	57.10%	57.10%	71.42%
Car	92.30%	85.70%	93.50%	99.53%
Nursery	97%	90.30%	98.37%	99.72%
Primary_tumor	39.80%	50.10%	39.20%	38.30%
Led7	71.50%	74%	70%	69.50%
lymphography	77.02%	83.10%	82.40%	84.40%
Splice	94.35%	95.36%	74.67%	95.55%
Breast_Cancer	75.52%	76.67%	72.37%	64.68%
Spect	71.05%	71.25%	53.75%	63.75%
Balance_Scale	76.64%	90.40%	86.56%	90.72%
Diabetes	73.80%	76.30%	70.18%	75.39%
Glass	66.82%	48.59%	71.96%	67.75%
Breast_w	94.56%	95.99%	96.85%	95.27%
Messidor_features	64.37%	77.68%	81.15%	72.02%
Sonar	71.15%	67.78%	86.53%	82.21%
Iris	96%	96%	95.30%	97.33%
Vehicle	72.45%	44.79%	69.85%	81.67%
Waveform_5000	75.08%	80%	73.62%	83.56%
Column3	81.61%	83.22%	78.38%	85.48%
Ecoli	84.22%	85.41%	80.35%	86.01%
Weather_Num	64.28%	64.28%	78.57%	78.37%
Vowel	81.51%	63.73%	99.29%	92.82%
Zoo	92.07%	95.04%	96.03%	95.90%
Tae	59.60%	54.30%	62.25%	54.30%
Zoo	92.07%	95.04%	96.03%	95.90%
Credit-g	70.50%	75.40%	72.00%	71.50%
Bands	64.74%	73.28%	78.29%	77%
Credit_Aproval	86.08%	77.60%	81.10%	83.18%
Autos	81.95%	56.09%	76.09%	80%
Colic	85.32 %	77.98 %	81.25 %	80.43 %

Tabla 2 Porcentajes de precisión obtenidos por método

Recomendación de uso de algoritmos de clasificación

Una vez obtenidos los resultados de los algoritmos, se creó una base de datos con la información que se muestra en la Tabla 1, eliminando la columna del nombre de la base de datos y agregando en una última columna como atributo clase o variable dependiente el nombre del clasificador que obtuvo el mejor porcentaje de precisión, obtenido de los resultados que se mostraron en la Tabla 3.

Este conjunto de datos se creó con el objetivo de aplicarle algoritmos de reglas de clasificación para generar un conjunto de patrones o reglas que permitan saber qué clasificador utilizar bajo ciertas características de una población de datos. Se aplicaron los algoritmos Apriori, BFTree, JRIP rules, J48 y Ridor, obteniéndose los siguientes resultados:

Algoritmo A priori

```
 1. Tipo_Datos=Nominal Num_Clases=3_7 4 ==> Clase=MLP 4 conf:(1)
 2. Tipo_Datos=Nominal Num_Clases=3_7 Completa=No 4 ==> Clase=MLP 4  conf:(1)
 3. Tipo_Datos=Numérico Num_Clases=3_7 7 ==> Clase=MLP 6 conf:(0.86)
 4. Tipo_Datos=Numérico Num_Clases=3_7 Completa=No 7 ==> Clase=MLP 6  conf:(0.86)
 5. Completa=Si 5 ==> Clase=J48 4  conf:(0.8)
 6. Tipo_Datos=NumNom Completa=No 5 ==> Clase=KNN 4 conf:(0.8)
 7. Tipo_Datos=NumNom Completa=Si 5 ==> Clase=J48 4  conf:(0.8)
 8. Balanceada=No Completa=Si 5 ==> Clase=J48 4  conf:(0.8)
 9. Tipo_Datos=NumNom Balanceada=No Completa=Si 5 ==> Clase=J48 4  conf:(0.8)
Algoritmo BFTree
Tipo_Datos = (NumNom)
| Completa = (No): KNN(4.0/1.0)
| Completa != (No): J48(3.0/1.0)
Tipo_Datos != (NumNom)
| Num_Clases < 2.5: NB(3.0/4.0)
| Num_Clases >= 2.5
| | Num_Clases < 6.5: MLP(10.0/0.0)
| | Num_Clases >= 6.5: NB(2.0/1.0)
Algoritmo JRIP rules
(Completa = Si) => Clase=J48 (4.0/1.0)
(Tipo_Datos = NumNom) => Clase=KNN (5.0/1.0)
 => Clase=MLP (20.0/9.0)
```

Algoritmo J48

```
Completa = Si: J48 (4.0/1.0)
Completa = No
|  Tipo_Datos = Nominal
|  |  Num_Clases <= 7
|  |  |  Num_Clases <= 2: NB (3.0/1.0)
|  |  |  Num_Clases > 2: MLP (4.0)
|  |  Num_Clases > 7: NB (2.0)
|  Tipo_Datos = Numérico
|  |  Num_Clases <= 2: KNN (4.0/1.0)
|  |  Num_Clases > 2: MLP (7.0/1.0)
|  Tipo_Datos = NumNom: KNN (5.0/1.0)
Algoritmo Ridor
Clase = J48  (29.0/26.0)
     Except (Completa = No) => Clase = NB  (17.0/0.0) [8.0/0.0]
          Except (Num_Clases > 2.5) and (Num_Clases <= 8.5) => Clase = KNN
                    (10.0/0.0) [3.0/0.0]
               Except (Num_Clases <= 6.5) => Clase = MLP  (7.0/0.0) [4.0/1.0]
               Except (Balanceada = Si) => Clase = KNN  (2.0/0.0) [1.0/0.0]
               Except (Tipo_Datos = Numérico) => Clase = MLP (3.0/1.0) [2.0/1.0]
```

A partir de estos resultados, se verificó qué reglas coinciden, obteniendo en común las que se muestran en la Tabla 4.
A partir de estos resultados, se verificó qué reglas coinciden, obteniendo en común las o criterios que se muestran en la Tabla 4.

No.	Regla	Algoritmo	Porcentaje
1	BD_Incompleta = Si	C4.5	75%
2	((Num_Clases >= 3 and Num_Clases < 7) and (Tipo_Dato = Numérico)) or ((Num_Clases >= 3 and Num_Clases < 7) and (Tipo_Dato=Nominal))	MLP	90%
3	((Num_Clases < 3 or Num_Clases >= 7) and Tipo_Datos = Numérico)) OR (Si BD=Mixta)	KNN	81%
4	(Tipo_Datos = Nominal) and (Num_Clases < 3 or Num_Clases >= 7)	Naive Bayes	100%

Tabla 3 Reglas obtenidas por método de clasificación

Como puede observarse en la Tabla 4, las características que más influyeron en la creación de estas reglas fueron: número de clases, tipos de atributos y si la base de datos tiene valores faltantes o no. La misma tabla presenta a qué algoritmo corresponde cada una y con qué precisión son creadas. Esta precisión corresponde a los 30 conjuntos de datos usados en la experimentación.

Para probar el funcionamiento de las reglas o criterios obtenidos anteriormente, fue necesario desarrollar un framework en Java, donde se solicitan las características de la base de datos y el sistema automáticamente sugiera el clasificador a utilizar, permtiendole al usuario cambiarlo si asi lo desea. En la Figura 1 se muestra la interface principal.

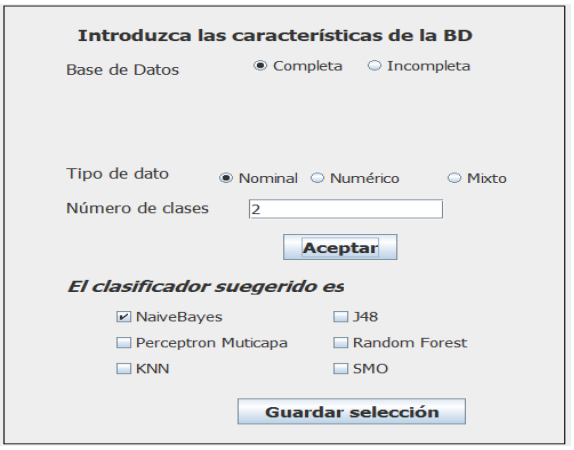

Figura 1 Sistema para seleccionar el clasificador

Otros comportamientos que se detectaron en la etapa de experimentación con los diferentes algoritmos fueron los siguientes:

- El algoritmo que mostró en promedio ser más rápido fue Naive Bayes y el que se mostró más lento en generar el modelo de clasificación fue Perceptrón Multicapa.

- El método que mostró mejor desempeño ante datos con ruido fue J48.

Etapa de validación

Con la finalidad de validar las reglas obtenidas en la sección anterior, se utilizaron las bases de datos mostradas en la Tabla 2. En esta etapa, se aplicaron los mismos métodos que en la etapa de experimentación. En la Tabla 5 se muestran los porcentajes de precisión obtenidos, indicando en la última columna si el resultado coincide con la recomendación emitida por la regla correspondiente.

ISSN-2531-2197

Base de datos	C4.5 J48	Naive Bayes	KNN	MLP
Vote	96.30%	90.10%	92.40%	94.71%
Letter	87.98%	64.11%	96.03%	82.08%
Monk	82.25%	77.41%	76.61%	96.36%
Sick	98.80%	92.60%	96.18%	97.24%
Ozone	96.33%	70.78%	95.26%	95.67%
Page_Blocks	96.87%	90.84%	96.01%	96.23%
Segment	96.92%	80.21%	97.14%	96.14%
Squash_Stored	65.38%	61.53%	73.07%	63.46%

Tabla 4 Porcentajes de precisión en la validación

Discusión

Los resultados generados en la etapa de experimentación con el uso de 30 bases de datos arrojaron en promedio un 86% de efectividad en las reglas obtenidas, mientras que en la etapa de validación se obtuvo un 87%. En particular, donde no se cumplió (conjunto de datos Page_Blocks de la Tabla 5), el algoritmo que debió ser seleccionado según la regla (Perceptrón Multicapa) quedó en segundo lugar y con una diferencia no significativa, lo cual indica que se podría usar este último sin perder precisión en su ejecución sobre los datos tratados. Los porcentajes de efectividad obtenidos en ambas etapas muestran un comportamiento bastante estable en la recomendación brindada por la regla, ya que la variación mínima en la etapa de validación mostró consistencia en el uso de las reglas.

Para cada algoritmo se detectaron características que influyen en los resultados: el método AD mostró una ventaja sobre el resto de los algoritmos cuando trata con bases de datos con valores faltantes, con Perceptrón Multicapa el factor que más influye es el número de clases tratadas y por último, con KNN y Naive Bayes la característica que más influyó fue el tipo de datos. Naive Bayes se comportó mejor con atributos nominales y KNN con numéricos y combinados.

Otro resultado no menos importante en este estudio es que algunas de las características consideradas no influyeron en los resultados alcanzados, estas son: la estructura de la base de datos, si es o no balanceada, número de atributos y número de instancias.

Conclusiones y trabajo futuro

El presente trabajo permitió comprobar que las características del conjunto de datos bajo estudio pueden influir en los resultados que se obtienen al aplicar un determinado algoritmo de clasificación, obteniéndose en la aplicación de las reglas generadas un buen porcentaje de efectividad.

Los resultados de este trabajo proporcionan una alternativa para decidir qué clasificadores son los mejores para ser utilizados para un conjunto de datos con unas características particulares. Por lo tanto, las reglas propuestas en este trabajo son un recurso útil que permite reducir el tiempo dedicado a la selección del clasificador, sobre todo para aquellos usuarios que no tienen dominio sobre cómo trabajan los diferentes algoritmos y/o cómo influye la naturaleza de los datos en esta tarea. Como trabajo futuro se propone extender el estudio realizando experimentos con más bases de datos y proporcionando criterios más comprensibles para los usuarios.

Referencias

Akinola S. y Oyabugbe O. (2015). Accuracies and Training Times of Data Mining Classification Algorithms: An Empirical Comparative Study, *Indian Journal of Science and Technology*, Vol 8, No. 15, 440-447.

Ashari A., Paryudi I., Tjoa, A. (2013). Performance Comparison between Naive Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool, *International Journal of Advanced Computer Science and Applications* (IJACSA), Vol. 4, No. 11, 33-39.

Coomans D., Massart D. (1982). Alternative k-nearest neighbour rules in supervised pattern recognition: k-Nearest neighbour classification by using alternative voting rules, *Analytica Chimica Acta*, Vol. 136, 15-27.

Deza E., Deza, M. (2009). Encyclopedia of Distances, *Springer*, 94-105.

Entezari, R., Rezaei, A., Minaei, B. (2009). Comparison of Classification Methods Based on the Type of Attributes and Sample Size, *Journal of Convergence Information Technology*, Vol. 4, No. 3, 94-102.

Miquelez, T., Bengoetxea E., Larranaga P. (2004). Evolutionary Computation based on Bayesian Classifier, *International Journal Application Mathematics Computation*. Vol. No. 3, pp. 335 – 349.

Moran, S., He Y., Liu, K. (2009). Choosing the Best Bayesian Classifier: An Empirical Study, *International Journal of Computer Science* (IJCS), Vol. 36, No. 4, 25-34.

Hernández, J. Ramírez, M., y Ferri, C. (2006): Introducción a la Minería de Datos, *Ed. Pearson* Prentice Hall, 25-28.

Lichman N., (2013). UCI Machine Learning Repository, [http://archive.ics.uci. edu/ml], Irvine, CA: University of California, *School of Information and Computer Sci-ence*.

Lu, H., Setiono, R. (1997). Effective Data Mining Using Neural Networks, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 1, 957-961.

Quinlan, J. (1986). Induction of decision trees. Machine Learning, *Academic Publishers Boston* Vol. 1, 81-106.

Taylor, N. y Shawe J., (2004), An Introduction to Support Vector Machines and other kernel based learning methods, *Cambridge University Press*, 15-24.

ISSN-2531-2197