Universidad Simón Bolívar
Decanato de Estudios Profesionales
Coordinación de Ingeniería de la Computación

Título

Por:
Antonio Álvarez

Realizado con la asesoría de:
Emely Arráiz B.

PROYECTO DE GRADO
Presentado ante la Ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero de Computación

Sartenejas, septiembre de 2018

Resumen

Hola mundo

Índice general

Resumen I

Índice de Figuras III

Lista de Tablas IV

Índice de algoritmos V

Acrónimos y Símbolos VI

Introducción 1

1. Marco teórico 2
1.1. Descubrimiento de Conocimiento y preprocesamiento de datos 2
1.2. Selección de Instancias 6

2. Marco metodológico 8
2.1. Descripción general 8

3. Resultados 9

Conclusiones y Recomendaciones 10

Bibliografía 11

A. Apéndice A 14

Índice de figuras

Índice de Tablas

Índice de algoritmos

Acrónimos y Símbolos

KDD	Knowledge Discovery in Databases
DM	Data Mining
IS	Instance Selection
PS	Prototype Selection
NN	Nearest Neighbor
NE	Nearest Enemy
CNN	Condensed Nearest Neighbor
ENN	Edited Nearest Neighbor
RSS	Relaxed Selective Subset
GGA	Generational Genetic Algorithm
SGA	Steady-State Genetic Algorithm
CHC	CHC Adaptive Search Algorithm
MEM	Memetic Algorithm

$\in$	Relación de pertenencia, «*es un elemento de*»
$\subset$	Subconjunto propio
$\setminus$	Diferencia de conjuntos

Introducción

Introdución

Capítulo 1

Marco teórico

1.1. Descubrimiento de Conocimiento y preprocesamiento de datos

Hoy en día, existe una creciente necesidad de procesar grandes volúmenes de datos, estos datos son producto de la recolección de información de procesos y actividades de distintas índoles y se vuelven un material valioso para extraer información sobre posibles tendencias que puedan existir en dichos procesos. Es aquí donde entra el descubrimiento de conocimiento en bases de datos (KDD por su ssiglas en inglés) como disciplina encargada del procesamiento de datos para la extracción de información.

KDD es definida por Smyth, P. et al. [FSS96] como "el proceso no trivial de identificar patrones en los datos que sean válidos, novedosos, potencialmente útiles y finalmente entendibles". Para este fin, KDD se subdivide en distintas etapas a llevar a cabo para lograr el fin último de identificar patrones, éstas son [GLH16]: especificación del problema, entendimiento del problema, preprocesamiento de los datos, minería de datos, evaluación de los resultados y explotación de los resultados. En este trabajo es de especial interés la etapa de preprocesamiento de datos.

El preprocesamiento de datos consiste en el conjunto actividades destinadas a preparar los datos para ser usado por un algoritmo de minería de datos. Las actividades realizadas en el preprocesamiento pueden ser clasificadas como actividades para la preparación de los datos y la reducción de los mismos [GLH16].

La preparación de datos es un paso obligatorio en el preprocesamiento ya que transforma los datos, que inicialmente son inservibles para el algoritmo de DM por asuntos como la presencia de atributos faltantes en instancias, datos erróneos y atributos con formatos no aceptables para el algoritmo a utilizar [GLH16]. Dependiendo del enfoque dado, estas actividades pueden clasificarse en:

- **Limpieza de datos [GLH16, KCH+03]** Incluye el tratamiento de los atributos faltantes y los datos erróneos, que si se dejan sin tratar resulta en un modelo de minería de datos poco confiable. Un atributo faltante en una instancia resulta de no haberlo introducido al momento del registro o por la pérdida en el proceso de almacenamiento. Los datos con atributos faltantes pueden tratarse de 3 maneras [FKP07]: la eliminación de las instancias que presenten el problema, utilizar métodos de estimación de máxima verosimilitud para calcular promedios y variancias con lo cual llenar los atributos faltantes y utilizar algoritmos del repertorio de machine learning como k-nn, k-means o Suport Vector Machine para estimar el valor de los atributos faltantes.

  Por su parte, los datos erróneos (también conocidos como datos ruidosos) pueden venir de dos formas [CAB11]: ruido de clase cuando la instancia está mal clasificada y ruido de atributo cuando uno o más valores de atributos en una instancia están distorisionados y no representan la realidad. Para tratar los datos ruidosos se puede usar 3 métodos: construir algoritmos de minería de datos que no se vean afectados en cierta medida ante el ruido (sean robustos), pulir los datos [Ten99] de tal manera que se corrijan los errores y por último se puede identificar los datos ruidosos para eliminarlos del conjunto y así quedarse sólo con datos correctos [BF99]. Cada uno de estos métodos tiene sus ventajas y desventajas; si sólo se cuenta con lo robusto del algoritmo de clasificación o regresión se tiene que tendrá un nivel de tolerancia del cual al pasarse los resultados serán inservibles, pulir los datos sólo es aplicable a conjuntos de tamño pequeño y mediano debido al alto costo computacional que tienden a tener los algoritmos que hacen el trabajo y si se decide filtrar todos los datos ruidosos se puede disminuir considerablemente el conjunto hasta un punto que no sea utilizable por los algoritmos de clasificación y regresión; por lo tanto, lo que se

estila es usar una combinación en lo que sea posible de estos 3 métodos para obtener los mejores resultados

- **Transformación de datos [GLH16]** Se centra en aplicar fórmulas matemáticas a los valores de los atributos para así obtener valores sintéticos que pueden proporcionar más información respecto a la instancia y al conjunto que pertenencen. Las transformaciones más comunes son la lineal y la cuadrática, la primera se usa principalmente para combinar distintos atributos y así crear uno sintético para ser usado por el algoritmo de minería de datos, la transformación cuadrática por su parte, es usada cuando una transformación lineal no es suficiente para derivar información útil de los atributos. En este sentido, existen otros tipos de transformaciones como la polinomial que engloba a la lineal y a la cuadrática y la no polinomial que trata con transformaciones más complejas.

- **Integración de los datos [GLH16, BLN86]** Consiste en la unión de las conjuntos de datos provenientes de distintas fuentes en un único conjunto. La integración tiene que tomar en cuenta algunos aspectos que se pueden presentar durante el proceso, entre ellos están la redundacia de atributos, la cual sucede cuando 2 atributos están fuertemente correlacionados y por lo tanto, con tener uno de ellos se puede derivar el otro. La redundancia de atributos puede traer consigo un sobre ajuste (overfitting) de los modelos predictivos, además de aumentar el tiempo de cómputo de los mismos, es por eso que se debe eliminar esta redundancia y para ello se usa una prueba de correlación χ^2 con el fin de identificar los atributos redundantes y así decidir con cual quedarse.

  Continuando, con los problemas que se pueden presentar al momento de la integración, se tiene también la duplicación de instancias. Problema que normalmente trae consigo la inconsistencia en los valores de los atributos debido a las diferencias con las que se registran los valores. Para solucionar este problema primero se tiene que identificar las instancias duplicadas usando técnicas de identificicación de similitud como la propuesta de *Fellegi y Sunter* [FS69] que lo modela como un problema de inferencia bayesiana o como en [CKLS01] donde se usan árboles de clasificación y regresión (CART por sus siglas en inglés) para cumplir este trabajo.

- **Normalización de datos [GLH16]** Busca cambiar la distribución de los datos originales de tal manera que se acoplen a las necesidades de los algoritmos predictivos. Dos de los tipos de normalización más usadas son la normalización min-max en la cual se aplica la fórmula en la ecuación 1.1, donde max_A es el valor máximo del atributo sobre los valores en el conjunto, min_A es el valor mínimo existente, $nuevo\_max_A$ y $nuevo\_min_A$ son los nuevos rangos para el atributo:

$$v' = \frac{v - min_A}{max_A - min_A}(nuevo\_max_A - nuevo\_min_A) + nuevo\_min_A \tag{1.1}$$

  El otro tipo de normalización es la puntuación Z (Z-score) en donde se llevan los datos a promedio 0 y desviación estándar 1 aplicando la fórmula de la ecuación 1.2:

$$v' = \frac{v - \mu_A}{\sigma_A} \tag{1.2}$$

  .

Pasando a la reducción de los datos, se tiene que engloba todas las técnicas que reducen el conjunto de datos original para obtener uno representativo con el cual trabajar en los modelos predictivos. La reducción de datos cobra especial importancia cuando se tienen conjuntos muy grandes que retardarían en gran medida el tiempo de cómputo de los algorimtos que los van a usar. Las técnicas de reducción de datos son [GLH16]:

- **Discretización de datos [GLH16, GLS$^+$13]** Es el proceso de transformar datos numéricos en datos categóricos, definiendo un número finito de intervalos que representan rangos entre distintos valores consecutivos con el fin de poder tratarlos como valores nominales. Es de especial importancia el conseguir el número correcto de intervalos que mantengan la ínformación original de los datos, ya que muy pocos intervalos puede llegar a ocultar la relación existente entre un rango en específico y una clase dada y muchos intervalos puede llevar a un sobre ajuste [CPSK07]. El principal atractivo de la discretización es que

permite utilizar un algoritmo de minería de datos que trabaje principalmente con datos nominales como Naïve Bayes [YW09] a partir de datos numéricos. Para un estudio más completo de la discretización se referencia a [GLS+13].

- **Selección de características [GLH16, LM12]** Busca eliminar atributos que sean redundantes o irrelevantes de tal manera que el subconjunto de características restantes mantenga la distribución original de las clases. El proceso de selección de características tiene ventajas como mantener e incluso mejorar la precisión de los modelos predictivos, reducir los tiempos de cómputo y reducir la complejidad de los modelos resultantes. La búsqueda de un subconjunto de atributos puede realizarse de 3 maneras: búsqueda exhaustiva, búsqueda heurística y métodos no determinísticos. La búsqueda exhaustiva cubre todo el espacio de soluciones, normalmente van probando todas las combinaciones posibles de atributos para conseguir el que mejor se acople a la métrica a optimizar, entre los métodos exhaustivos están Focus [AD91], Automatic Branch & Bound [LMD98], Best First Search [XYC88], entre otros. Por su parte, la busqueda heurística busca una solución aproximada a la óptima en poco tiempo, entre sus métodos están los propuestos en [DL97, KS96, Bat94]. Por último, están los métodos no determinísticos, de entre los que destacan los algoritmos géneticos, recocido simulado y Las Vegas Filter [LS+96].
- **Selección de instancias [GLH16]** Consiste en elegir un subconjunto de las instancias totales manteniendo las características del conjunto original. Es el problema a tratar en este trabajo y se elabora más sobre el mismo en la siguiente sección.

1.2. Selección de Instancias

La selección de instancias (IS por sus siglas en inglés) consiste en reducir el conjunto de datos dado a un conjunto reducido que va a ser utilizado con un algoritmo clasificador, manteniendo el desempeño del algorimto como si se usara el conjunto original.

Definición 1. Dado un conjunto de datos **X**, se tiene que una instancia $X_i = (X_i^1, X_i^2, \ldots, X_i^p)$ donde X_i^j es el atributo j para la instancia X_i con $X_i \in X$ y siendo p el número de atributos. La instancia X_i es de clase Y_j donde $Y_i \in Y$, siendo Y el conjunto de todas las clases definidas con $j \in (1 \ldots q)$ donde q es el número de clases totales. Se divide el conjunto **X** en un conjunto **TR** de entrenamiento y un conjunto **TS** de prueba. El problema de **Selección de Instancias** consiste en conseguir un conjunto **S** $\subset$ **TR** con el cual al usarse con el clasificador **T** se obtengan los mismos valores de precisión o mejores que al usar **T** con **TR** [GLH16].

La respuesta óptima de un método de selección de instancias es un conjunto *consistente* y de cardinalidad mínima.

Definición 2. "Un conjunto R es **consistente** con T, *si y solo si* toda instancia $t \in T$ es clasificada correctamente mediante el uso de un clasificador M y las instancias en R como conjunto de entrenamiento." [Ale14]

Sin embargo, conseguir la respuesta óptima es un problema NP-Duro (NP-Hard) como lo demuestra *Zukhba* en [Zuk10]. Por lo tanto, la mayoría de los métodos propuestos hasta la fecha se enfocan en obtener una solución aproximada.

Capítulo 2

Marco metodológico

2.1. Descripción general

Marco metodológico

Capítulo 3

Resultados

Resultados

Conclusiones y Recomendaciones

Conclusiones

Bibliografía

[AD91] Hussein Almuallim and Thomas G Dietterich. Learning with many irrelevant features. In *AAAI*, volume 91, pages 547–552, 1991.

[Ale14] Flores Alejandro. *Metaheurísticas Bio-Inspiradas para Selección de Instancias*. PhD thesis, Undergraduate thesis,Departamento de Ciencias de la Computación, Universidad Simón Bolívar, Venezuela, 2014.

[Bat94] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4):537–550, 1994.

[BF99] Carla E Brodley and Mark A Friedl. Identifying mislabeled training data. *Journal of artificial intelligence research*, 11:131–167, 1999.

[BLN86] Carlo Batini, Maurizio Lenzerini, and Shamkant B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM computing surveys (CSUR)*, 18(4):323–364, 1986.

[CAB11] Cagatay Catal, Oral Alan, and Kerime Balkan. Class noise detection based on software metrics and roc curves. *Information Sciences*, 181(21):4867–4877, 2011.

[CKLS01] Munir Cochinwala, Verghese Kurien, Gail Lalk, and Dennis Shasha. Efficient data reconciliation. *Information Sciences*, 137(1-4):1–15, 2001.

[CPSK07] Krzysztof J Cios, Witold Pedrycz, Roman W Swiniarski, and Lukasz Andrzej Kurgan. *Data mining: a knowledge discovery approach*. Springer Science & Business Media, 2007.

[DL97] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[FKP07] Alireza Farhangfar, Lukasz A Kurgan, and Witold Pedrycz. A novel framework for imputation of missing values in databases. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(5):692–709, 2007.

[FS69] Ivan P Fellegi and Alan B Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[FSS96] Usama M Fayyd, Gregory P Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: An overview. 1996.

[GLH16] Salvador García, Julián Luengo, and Francisco Herrera. *Data preprocessing in data mining*. Springer, 2016.

[GLS+13] Salvador Garcia, Julian Luengo, José Antonio Sáez, Victoria Lopez, and Francisco Herrera. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):734–750, 2013.

[KCH+03] Won Kim, Byoung-Ju Choi, Eui-Kyeong Hong, Soo-Kyung Kim, and Doheon Lee. A taxonomy of dirty data. *Data mining and knowledge discovery*, 7(1):81–99, 2003.

[KS96] Daphne Koller and Mehran Sahami. Toward optimal feature selection. Technical report, Stanford InfoLab, 1996.

[LM12] Huan Liu and Hiroshi Motoda. *Feature selection for knowledge discovery and data mining*, volume 454. Springer Science & Business Media, 2012.

[LMD98] Huan Liul, Hiroshi Motoda, and Manoranjan Dash. A monotonic measure for optimal feature selection. In *European conference on machine learning*, pages 101–106. Springer, 1998.

[LS+96] Huan Liu, Rudy Setiono, et al. A probabilistic approach to feature selection-a filter solution. In *ICML*, volume 96, pages 319–327. Citeseer, 1996.

[Ten99] Choh-Man Teng. Correcting noisy data. In *ICML*, pages 239–248. Citeseer, 1999.

[XYC88] Lei Xu, Pingfan Yan, and Tong Chang. Best first strategy for feature selection. In *Pattern Recognition, 1988., 9th International Conference on*, pages 706–708. IEEE, 1988.

[YW09] Ying Yang and Geoffrey I Webb. Discretization for naive-bayes learning: managing discretization bias and variance. *Machine learning*, 74(1):39–74, 2009.

[Zuk10] AV Zukhba. Np-completeness of the problem of prototype selection in the nearest neighbor method. *Pattern Recognition and Image Analysis*, 20(4):484–494, 2010.

Apéndice A

Apéndice A

Apéndice A