Revista *Márgenes*. Vol.5, No.1, Enero-Marzo, 2017

TÍTULO: ALGORITMO VFI: UN ESTUDIO EXPERIMENTAL EN WEKA

Autores: Ing. Jessie Guillemí Martín[1], MSc. Alain Pereira Toledo[2]

[1]Profesor Instructor. Departamento de Ingeniería Informática. Facultad de Ciencias Técnicas. Universidad de Sancti Spíritus "José Martí Pérez". Correo electrónico: jessie@uniss.edu.cu. Línea de investigación: aprendizaje automatizado. Ingeniera en Ciencias Informáticas

[2]Profesor Asistente. Departamento de Ingeniería Informática, Facultad de Ciencias Técnicas. Universidad de Sancti Spíritus "José Martí Pérez". Correo electrónico: alain@uniss.edu.cu. Línea de investigación: aprendizaje automatizado. Máster en Ciencia de la Computación. Licenciado en Ciencia de la Computación.

RESUMEN

Las grandes bases de datos son un reto hoy en día, ya que existe la necesidad de algoritmos de procesamiento más rápidos y confiables. Cuando se utilizan técnicas de aprendizaje automatizado, a menudo involucra un alto costo computacional asociado con el tiempo de entrenamiento; pero no es necesario un nuevo algoritmo si se selecciona el apropiado. Por esta razón, el presente artículo se propone como objetivo: realizar un estudio experimental para comparar un algoritmo conocido y simple llamado *Voting Feature Intervals* (VFI), con otros influyentes clasificadores, con base en la precisión de la clasificación. La experimentación se llevó a cabo mediante la herramienta WEKA, y se utilizó la metodología estadística de Demšar para evaluar los resultados. Finalmente, se mostró que su comportamiento, en cuanto a la correctitud de la clasificación, no es significativamente peor que otros algoritmos bien conocidos, mientras que su entrenamiento y tiempo de clasificación es lo suficientemente rápido en grandes bases de datos.

Palabras clave: algoritmo de clasificación; VFI; WEKA; grandes bases de datos

TITLE: VFI ALGORITHM: AN EXPERIMENTAL STUDY IN WEKA

ABSTRACT

Big databases are a challenge nowadays, so there is a need for faster and reliable data processing algorithms. When machine learning techniques are used, there is often a high computational cost associated with the training time. But there is no need for a new algorithm if we select the appropriate one. For this reason, this paper reports de results of an experimental study for comparing a known and simple algorithm, called *Voting Feature Intervals* (VFI), with other influential classifiers based on the classification accuracy. We used the WEKA tool to carry out the experiment, and Demšar's statistical methodology for evaluating the results. Finally, we showed that its behavior, in terms of classification accuracy, is not significantly worse than other well-known algorithms while its training and classification time is fast enough on large data sets.

Keywords: classification algorithm; VFI; WEKA; big databases

INTRODUCCIÓN

Los Métodos de aprendizaje automatizados (AA) están compuestos por un algoritmo de entrenamiento y un algoritmo de clasificación. El tiempo de entrenamiento puede ser un obstáculo cuando se intenta procesar grandes flujos de datos; es por esto que los investigadores continúan buscando nuevos algoritmos y plataformas para solucionar este problema (Tsai, Lai, Chao, & Vasilakos, 2015). Sin embargo, a veces olvidan mirar hacia atrás.

En la última década, muchos métodos de AA se han propuesto (Fernández-Delgado, Cernadas, & Barro, 2014; Liao, Chu, & Hsiao, 2012) y *Voting Feature Intervals* es uno de ellos. Sin duda es un método de clasificación muy conocido, que puede encontrarse implementado en uno de los paquetes de la herramienta WEKA. Fue creado por Demiröz y Güvenir (Demiröz & Güvenir, 1997; H.A. Güvenir, Acar, Demiröz, & Cekin, 1997) con el fin de obtener un algoritmo con un alto nivel de exactitud, manteniendo sus tiempos de entrenamiento y clasificación. Otros requisitos fueron tomados en cuenta en su diseño (H Altay Güvenir, Demiröz, & Ilter, 1998), tales como su comportamiento derivado del tratamiento de valores nulos y ejemplos ruidosos. Este clasificador también incorpora información sobre la relevancia de cada atributo con respecto a los valores de clase.

Por lo tanto, aun cuando se asume que cada atributo es independiente uno del otro, este método aprende el nivel de relevancia de cada atributo con respecto al dominio. Un algoritmo de AA también debe ser comprensible, tal que los expertos humanos puedan aprender de él.

Por otro lado, el análisis de grandes volúmenes de datos es otra cuestión importante que debe ser considerada tanto en las empresas (Bughin, 2016), como en los actuales proyectos de investigación de AA (Tsai et al., 2015). Exige mayor escalabilidad en las fases de entrenamiento y clasificación, pues cuando se analizan grandes volúmenes de datos, una mínima mejora en el consumo de tiempo puede hacer una gran diferencia.

El VFI podría ser un buen candidato a método de clasificación para el análisis de grandes volúmenes de datos, pues se dice que es un algoritmo rápido, preciso y robusto. Con el fin de mostrar sus características principales, se propone como **objetivo** en el presente artículo: realizar un estudio experimental y comparativo en cuanto a la exactitud de la clasificación sobre bases de datos ruidosas, desbalanceadas y con presencia de valores perdidos, con cuatro de los métodos de AA más conocidos, y con la ayuda de la herramienta WEKA.

Su bajo costo computacional se ha declarado ya en (Demiröz & Güvenir, 1997). Por lo tanto, se asume que el método VFI es una propuesta competitiva en términos de consumo de tiempo tanto en la fase de entrenamiento como en la de clasificación.

Las siguientes secciones proporcionan una breve descripción de los algoritmos de entrenamiento y clasificación del VFI, proporcionando una comprensión fundamental de sus características principales. Luego se describen la herramienta y el procedimiento usados para preparar y ejecutar el experimento de comparación. Hacia el final de este trabajo, se comparten y discuten algunos resultados estadísticos para evaluar si el método VFI es, efectivamente, comparable a otros métodos bien conocidos en cuanto a la exactitud de la clasificación.

DESARROLLO

CONCEPTOS FUNDAMENTALES SOBRE EL ALGORITMO VFI

Como resultado de la fase de entrenamiento de los algoritmos de votación (Kohavi, 1999), se obtienen los pesos informacionales de los atributos y, a menudo, de sus objetos. El método VFI, como cualquier otro método de votación, calcula estos pesos,

pero de una manera mixta (Figura 1). Divide el conjunto de valores de cada función en intervalos continuos y cada intervalo es o un rango de valores, o uno solo, en cuyo caso se conoce como *intervalo puntual*.

En contraste, la mayor parte de los métodos de votación estiman los pesos informacionales de sus objetos a través de un único objeto de la base de datos. Sin embargo, VFI interpreta un objeto como un grupo continuo de objetos simples proyectados sobre un atributo. Al final del proceso de entrenamiento, VFI construye el modelo, consistente en una matriz que contiene el número de objetos simples que comparten el mismo valor de clase en cada intervalo de un determinado atributo, y a esto es a lo que se le llama *voto*.

```
train(TrainingSet):
begin
    for each feature f
    if f is linear
        for each class c
            EndPoints[f] = EndPoints[f] U find_end_points(TrainingSet, f, c);
        sort( EndPoints[f]);

        for each end point p in EndPoints[f]
            form a point interval from end point p
            form a range interval between p and the next endpoint r p
        else /* f is nominal */
             form a point interval for each value of f

        for each interval i on feature f
            for each class c
                interval_class_count[f, i, c] = 0
         count_instances(f, TrainingSet);
         for each interval i on feature f
            for each class c
                interval_class_vote[f, i, c] =
                    interval_class_count[f, i, c]/class_count[c]
```

Figura 1. Algoritmo de entrenamiento del método VFI

Fuente: (H. Altay Güvenir, 1998).

La fase de clasificación para VFI también es muy simple (Figura 2). Cuando llega una nueva instancia, el procedimiento determina a cuál de los intervalos pertenece el valor de la función correspondiente. Entonces, se acumula cada voto dado por el intervalo para cada valor de clase. Por último, la clase con la mayor cantidad de votos es la clase *ganadora*.

```
classify(e):
/* e: example to be classified */
begin
    for each class c
    vote[c] = 0
    for each feature f
    for each class c
        feature_vote[f, c] = 0 /* vote of feature f for class e */
    if e_{f} value is known
        i = find_interval(f, e_{f})
        feature_vote[f,c] = interval_class_vote[f, i, c]
        for each class c
            vote[c] = vote[c] + feature_vote[f,c];
    return class c with highest vote[c];
end.
```

Figura 2. Algoritmo de clasificación del método VFI

Fuente: (H. Altay Güvenir, 1998).

Los algoritmos implementados en WEKA son similares a los mostrados anteriormente. Sin embargo, hay algunas diferencias respecto a cuándo se realiza la normalización y respecto a la opción *wheightByConfidence*.

Mientras que la normalización de los intervalos de atributo en el método VFI original ocurre durante la fase de entrenamiento, la versión implementada de WEKA pospone la normalización de los intervalos hasta que ya haya alcanzado la fase de clasificación.

Si a *wheightByConfidence* se le da valor verdadero, entonces cada voto de cada intervalo de atributo se pesa según la Ecuación 1.

$$w(A_i = I(A_i)^{bias} = (\frac{-(\sum_{i=0}^{nC} n_i \lg n_i) + n \lg n}{n \lg 2})^{bias}) \quad (1)$$

En la Ecuación 1, $I(A_i)$ es la entropía relacionada al atributo A_i, donde n es el número total de objetos, nC es el número de clases y n_i es el número de objetos de la clase i.

MATERIALES Y MÉTODOS

Se realizó un experimento para establecer si uno de los algoritmos se posiciona como el mejor en cuanto a clasificación. Su diseño y ejecución se realizó usando el *Experimenter* (Figura 3) de la herramienta WEKA (Bouckaert et al., 2015), versión 3.7.13. La herramienta WEKA es un conocido repositorio de algoritmos de AA. Todos los algoritmos en este repositorio se implementan para que puedan ser usados en estudios experimentales con bases de datos relacionales. Para ella se encuentran disponibles los cuatro métodos seleccionados para compararse con el VFI.

El primero es el Naïve Bayes (John & Langley, 1995), el cual es un algoritmo Bayesiano sencillo (Bielza & Larrañaga, 2014). El segundo es K-NN (Aha, Kibler, & Albert, 1991), renombrado por WEKA como *IBK*. El otro es una versión del método de la Máquina de Soporte Vectorial (*Support Vector Machine* o SVM) (Platt & others, 1998), con un *polikernel* para el entrenamiento. El último es el árbol de decisiones C4.5 de Ross Quinlan (Quinlan, 1993), también renombrado por WEKA como J48. Todos ellos son considerados como parte de los algoritmos más influyentes en el AA (Wu et al., 2008).

Las bases de datos que se utilizaron se clasifican según características bien establecidas:

- Tipos de valores mezclados
- Valores perdidos
- Valores ruidosos
- Atributos irrelevantes
- Grandes bases de datos
- Bases de datos desbalanceadas

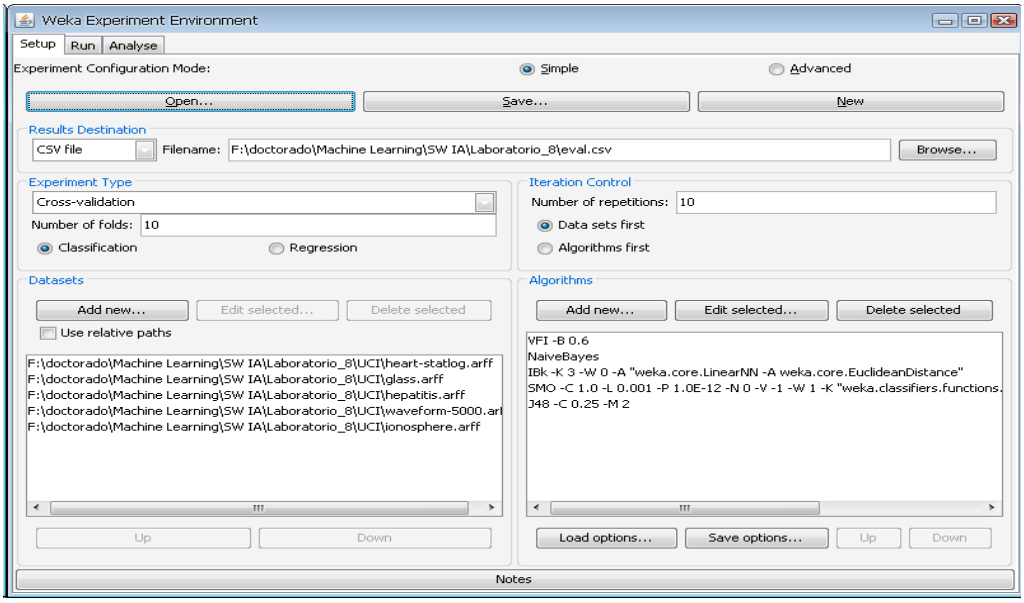

Figura 3. Interfaz del *Experimenter* en WEKA.

Fuente: Captura de pantalla.

Así que se eligió una lista de bases de datos basada en esas características. Hay cinco bases de datos, cada una de ellas fue descargada desde el repositorio de Aprendizaje Automatizado de la UCI (Lichman, 2013). Una clasificación según las características mencionadas se muestra en la Tabla 1.

Los métodos estadísticos y de AA no siempre se comportan como se espera de ellos cuando una de estas características está presente en la base de datos que se analiza. Se debe ser consciente de lo que dice la teoría *No Free Lunch*, y no intentar encontrar el mejor, sino el más adecuado para sus propósitos. Esto no significa que no pueda encontrarse un algoritmo con un mejor comportamiento en cuanto a, por ejemplo, exactitud de clasificación. No obstante, sólo será el mejor en un determinado dominio.

Tabla 1. Clasificación de las Bases de Datos usadas en el experimento

Base de datos	Mezclados	Perdidos	Ruidosos	Irrelevantes	Grande	Desbalance
Heart-s	X	-	-	-	-	-
Glass	X	-	-	-	-	X
Hepatitis	X	X	-	-	-	-
Waveform	-	-	X	X	X	-
Ionosphere	-	-	X	-	-	-

Fuente: (Lichman, 2013)

Como resultado, el procedimiento seguido en este estudio prescribe un orden en la serie de experimentos con la herramienta WEKA. En primer lugar, se probará cada método contra cada base de datos, con el objeto de llegar a conclusiones respecto a las características antes mencionadas. Luego, se debe probar cada clasificador contra la lista general de bases de datos.

La metodología para comparar *n* métodos de AA sobre *m* bases de datos se describe en (Demšar, 2006). Antes, algunas pruebas deben realizarse sobre bases de datos individuales, con el estadístico T-Test para comparar el algoritmo de referencia con el resto.

RESULTADOS Y DISCUSIÓN

Hay muchos tipos de estadísticos disponibles con el propósito de lograr exactitud en la clasificación. Aunque principalmente, hay dos medidas de clasificación de la calidad que compiten entre sí: *Precisión del Porcentaje de Clasificación* y la medida de precisión *Receiver Oparating Characteristics* (ROC). En (Provost, Fawcett, & Kohavi, 1998) y (Demšar, 2006) también se argumenta a favor de ROC; por lo tanto, esta es la medida que se utiliza para probar múltiples algoritmos. Los resultados del experimento según esta medida se muestran en la Tabla 2.

Tabla 2. Pruebas pareadas y corregidas T-Test usando el ROC promediado de cada algoritmo por cada base de datos

Base de datos	VFI	Naïve Bayes	K-NN	SVM	J48
Heart-s	0.87	**0.90**	0.84	0.83	0.79-
Glass	0.79	0.73	**0.87**+	0.78	0.79
Hepatitis	**0.88**	0.86	0.77-	0.77-	0.67-
Waveform	0.74	**0.94**+	0.89+	0.92+	0.81+
Ionosphere	**0.96**	0.94	0.90-	0.84-	0.89-
Promedio	0.85	**0.87**	0.85	0.83	0.79

+, - mejora o degradación estadísticamente significativa

Fuente: Resultados del experimento con *Experimenter* de WEKA.

Comparación de la precisión en la clasificación

La Tabla 2 muestra una comparación por pares con el método VFI como base de la prueba. En solo un caso, VFI se destaca como significativamente peor que los otros, exceptuando el método de Naïve Bayes. Esto induce a pensar que VFI no se comporta muy bien en el caso de atributos irrelevantes y ruidosos. La base de datos *waveform* se utiliza con frecuencia en pruebas basándose en estas características.

Sin embargo, no fue este el caso en la base de datos *ionosphere*, siendo esta particularmente ruidosa. En consecuencia, la única conclusión plausible es que VFI se degrada cuando está en presencia de atributos irrelevantes. Esto se explica por la base de su funcionamiento, pues VFI utiliza para clasificar nuevas instancias, el peso informacional de los intervalos construidos; por lo tanto, habrá una gran cantidad de atributos activos que aportan votos y son irrelevantes. Aun así, esto no es conclusivo, ya que se necesita un estudio del comportamiento de VFI que exceda las conclusiones alcanzables en un experimento.

Para la lista completa de las bases de datos seleccionadas, los resultados se lograron a través de la prueba no-paramétrica de Friedman. Primeramente, se construye una lista de clasificación de los algoritmos probados. Cuanto mayor sea el número, peor en cuanto a la exactitud de clasificación (Tabla 3).

Tabla 3. Rankings promedio de los algoritmos

Algoritmo	Ranking
VFI	2.3
Naïve Bayes	2.2
K-NN	2.7
SVM	3.7
J48	4.1

Fuente: Resultados del experimento con test de Fridman.

El J48 se clasificó como el peor método, mientras que Naïve Bayes se comportó como el mejor. Ahora es preciso comprobar si las diferencias en el ranking son significativas.

Las pruebas de Friedman o Iman-Davenport, parecen ser las estadísticamente adecuadas para esta tarea. El *valor de P* calculado por la Prueba de Friedman es 0.21142005422460597. Por lo tanto, no se puede rechazar la hipótesis nula a un nivel

de significación de α = 0.05, y esa es la razón por la qué se puede afirmar que no hay diferencias significativas con respecto a la exactitud de la clasificación. Esto es consistente con el *valor de P* calculado en la prueba de Iman-Davenport, que alcanza una probabilidad de 0.2106163603533215. Por lo tanto, no será necesario utilizar ninguno de los test post hoc.

CONCLUSIONES

Se ha presentado un estudio experimental del algoritmo VFI, con lo cual se pretende que sirva como punto de partida para futuras investigaciones.

Fue descrito el algoritmo VFI y luego comparado con cuatro de los principales y más conocidos métodos de AA. El estadístico T-Test ejecutado por el *Experimenter* de WEKA muestra que sólo aparecen diferencias significativas con respecto a la precisión de clasificación de VFI en el caso de atributos irrelevantes.

Por otro lado, los *valores P* calculados por las pruebas de Friedman e Iman-Davenport están por debajo del nivel de significación, por lo que no hay evidencia significativa para rechazar la hipótesis nula, y es por esto que todos los algoritmos mencionados se consideran similares en cuanto a la exactitud de la clasificación.

Ya que fue demostrado que la precisión de clasificación de VFI es similar a la presentada por los otros algoritmos, y que resulta rápido tanto en la fase de entrenamiento como en la de clasificación; puede afirmarse que el método VFI es un buen candidato para el procesamiento de grandes volúmenes de datos.

Las investigaciones futuras al respecto deben enfocarse principalmente en corroborar la hipótesis de que el método VFI no se comporta como se espera de él cuando están presentes atributos irrelevantes, y en descubrir una explicación más conclusiva para ello. También existe falta de evidencia respecto a su supuestamente muy bajo costo computacional, y esto es algo que debe tenerse en cuenta al procesar bases de datos muy grandes.

REFERENCIAS BIBLIOGRÁFICAS

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, *6*(1), 37–66.

Bielza, C., & Larrañaga, P. (2014). Discrete Bayesian Network Classifiers: A Survey. *ACM Computing Surveys (CSUR)*, *47*(1), 5:1–5:43. https://doi.org/10.1145/2576868

Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., & Scuse, D. (2015, octubre 9). WEKA Manual for Version 3-7-13. University of Waikato.

Bughin, J. (2016). Big data, Big bang? *Journal of Big Data*, *3*(2), 1-14. https://doi.org/10.1186/s40537-015-0014-3

Demiröz, G., & Güvenir, H. A. (1997). Classification by voting feature intervals. En *Machine Learning: ECML-97* (pp. 85–92). Springer.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Fernández-Delgado, M., Cernadas, E., & Barro, S. (2014). Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, *15*, 3133-3181.

Güvenir, H. A. (1998). A classification learning algorithm robust to irrelevant features. En *Artificial Intelligence: Methodology, Systems, and Applications* (pp. 281–290). Springer. Recuperado a partir de http://link.springer.com/10.1007%2FBFb0057452

Güvenir, H. A., Acar, S., Demiröz, G., & Cekin, A. (1997). A supervised machine learning algorithm for arrhythmia analysis. En *Computers in Cardiology 1997* (pp. 433-436). https://doi.org/10.1109/CIC.1997.647926

Güvenir, H. A., Demiröz, G., & Ilter, N. (1998). Learning differential diagnosis of erythemato-squamous diseases using voting feature intervals. *Artificial Intelligence in Medicine*, *13*(3), 147–165.

John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. En *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence* (pp. 338–345). Morgan Kaufmann Publishers Inc. Recuperado a partir de http://dl.acm.org/citation.cfm?id=2074196

Kohavi, E. B. R. (1999). An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, *36*(1-2). https://doi.org/10.1023/a:1007515423169

Liao, S.-H., Chu, P.-H., & Hsiao, P.-Y. (2012). Data mining techniques and applications – A decade review from 2000 to 2011. *Expert Systems with Applications*, *39*(12), 11303-11311. https://doi.org/10.1016/j.eswa.2012.02.063

Lichman, M. (2013). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. Recuperado a partir de http://archive.ics.uci.edu/ml

Platt, J., & others. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. Recuperado a partir de http://www.msr-waypoint.com/pubs/69644/tr-98-14.pdf

Provost, F. J., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. En *ICML* (Vol. 98, pp. 445–453).

Quinlan, R. (1993). *C4.5: programs for machine learning* (Vol. 1). Morgan Kaufmann Publishers.

Tsai, C., Lai, C., Chao, H., & Vasilakos, A. V. (2015). Big data analytics: a survey. *Journal of Big Data*, *2*(21), 1-32. https://doi.org/10.1186/s40537-015-0030-3

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., … Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, *14*, 1-37. https://doi.org/10.1007/s10115-007-0114-2

Recibido: 25/11/2016

Aceptado: 06/03/2017