Algoritmo de votación incremental INC-ALVOT para clasificación supervisada

The incremental voting algorithm INC-ALVOT for supervised classification

Uriel Escobar Franco[1] *Guillermo Sánchez Díaz*[2*]

[1]Universidad Politécnica de Tulancingo, Ingenierías N.° 100, Col. Huapalcalco, Tulancingo, Hgo., C.P. 43629, México

[2]Universidad de Guadalajara, Departamento de Ciencias Computacionales e Ingenierías, CUValles Carr. Guadalajara-Ameca, Km. 45.5, Ameca, Jal., C.P. 46600

(Recibido el 10 de noviembre de 2008. Aceptado el 24 de agosto de 2009)

Resumen

En este trabajo, se presenta un algoritmo incremental para clasificación supervisada llamado INC-ALVOT (algoritmo de votación incremental). Este algoritmo permite manejar conjuntos de datos mezclados, los cuales no se almacenan en la memoria principal. Además, el algoritmo permite incorporar nuevos objetos en el conjunto de datos inicial, realizando un número mínimo de operaciones para la clasificación de nuevos objetos con el conjunto de datos expandido. Se presentan los resultados obtenidos al aplicar el algoritmo propuesto en diversos conjuntos de datos reales comparado con el algoritmo clásico de votación ALVOT.

----- *Palabras clave:* Clasificación supervisada, algoritmos de votación, algoritmos incrementales

Abstract

In this paper, an incremental supervised classifier called INC-ALVOT (incremental voting algorithm) is presented. This algorithm allows handle mixed data sets which do not keep in main memory. Besides, it allows that when the classification of a goal object was realize, new objects are incorporated in original database, carrying out a minimal operations for the classification of goal object with the expanded data set. Result obtained with the proposed algorithm and classical ALVOT algorithm on different real data sets is presented.

----- *Keywords:* Supervised classification, voting algorithms, incremental algorithms

* Autor de correspondencia: teléfono: + 52 + 375 + 758 05 00 ext. 7291, fax: + 52 + 375 + 758 01 48 ext. 7291, correo electrónico: guillermo.sanchez@profesores.valles.udg.mx (G. Sánchez).

Introducción

El modelo de algoritmos de clasificación supervisada denominado ALVOT [1,2], ha sido desarrollado en el enfoque lógico combinatorio de patrones [1]. Este modelo se basa en el concepto de precedencia parcial, que radica en que la comparación entre dos objetos se puede realizar parte a parte (parcialmente), y no necesariamente entre toda la descripción completa del objeto. Para la aplicación de ALVOT es necesario determinar algunos parámetros, incluyendo el conjunto de sistemas de apoyo, el cual indica que partes de los objetos serán relevantes para compararse.

Se han propuesto diferentes mejoras al modelo de algoritmos ALVOT, basadas principalmente en optimizar la estrategia de búsqueda para computar el sistema de conjuntos de apoyo [3], y en la edición de objetos y de los conjuntos de apoyo manejados por el algoritmo [4]. Sin embargo, solamente han sido reportados en la literatura modelos de algoritmos ALVOT no incrementales, y estos algoritmos presentan la necesidad de mantener el conjunto de datos completo en la memoria principal. Pero, si el tamaño del conjunto de datos es grande, entonces la aplicación del algoritmo puede no ser factible de aplicarse. Otro inconveniente es la re-clasificación de un objeto cuando la muestra de aprendizaje es incrementada con más objetos (i.e. un conjunto de datos al cual le han sido añadidos nuevos objetos), lo cual implica procesar nuevamente todos los objetos de la muestra de aprendizaje.

Por otro lado, fue desarrollado un algoritmo CR+ paralelo de clasificación supervisada basado en precedencias parciales, el cual tiene un comportamiento análogo a ALVOT [5]. Sin embargo, este algoritmo en el paso de la generación de candidatos de conjuntos de apoyo, realiza una cantidad significativa de repeticiones en diferentes procesadores, lo cual repercute en el tiempo de ejecución del algoritmo, obteniendo tiempos similares al algoritmo secuencial.

Una alternativa para mejorar algunas de las deficiencias mencionadas anteriormente, es el desarrollo de algoritmos incrementales [6], específicamente para clasificación supervisada. En el ámbito del enfoque lógico combinatorio, se han desarrollado diversos algoritmos incrementales [7, 8, 9]. Estas técnicas han reportado mejores tiempos que algunos modelos de algoritmos no incrementales.

En este trabajo, se propone un algoritmo ALVOT incremental, denominado INC-ALVOT, el cual mejora este modelo de algoritmos, ya que no mantiene el conjunto completo de datos en memoria (solamente trabaja con el objeto en estudio y con el cálculo parcial de todos los objetos procesados previamente). Además, si el conjunto de datos inicial es incrementado, el algoritmo propuesto solamente procesará los nuevos objetos añadidos en el conjunto de datos expandido.

Conceptos básicos

Sea $U = \{O_1, O_2, ..., O_s, ...\}$ un universo de objetos, $MA = \{O_1, O_2, ..., O_m\}$ un subconjunto de U (denominado también como muestra de aprendizaje para la clasificación); $R = \{X_1, X_2, ..., X_n\}$ el conjunto de atributos que describen a los objetos de U. Cada $X_i \in M_i$, donde M_i se denomina conjunto de valores admisibles de la variable X_i. MA es la unión finita de c conjuntos disjuntos $K_1, K_2, ..., K_c$ llamados clases. Cada objeto $O_i \in MA$ tiene asociado un c-tuplo de pertenencia, el cual describe la correspondencia del objeto O_i a las clases $K_1, K_2, ..., K_c$ Esta c-tupla de pertenencia se denota por $P(O_i)$, donde $P(O_i) = \{P_1(O_i), ..., P_c(O_i)\}$, donde $P_t(O_i) = 1$ significa que $O_i \in K_t$ y $P_t(O_i) = 0$ significa que O_i no pertenece a la clase K_t [10].

Las bases que describen los modelos de algoritmos de votación (ALVOT) fueron tomadas de [4].

Definición 1.- Un sistema de conjuntos de apoyo denotado por $\{W\}$ es un conjunto de subconjuntos de atributos de R (i.e. $\{W\} = \{W_1, W_2, \ldots, W_v\}$). Este sistema indica qué partes (i.e. que subconjuntos de atributos se considerarán) de los objetos serán comparados. Cada $W_j \in \{W\}$ es llamado conjunto de apoyo.

Definición 2.- Sea $W \subseteq R$ un conjunto de atributos. La sub-descripción de un objeto *O* usando solamente los atributos de *W,* se denomina la *W*-parte del objeto *O*, y se denotará como *WO*.

El modelo de algoritmos de votación está basado en las siguientes ideas:

Analogía. Se usa una función de semejanza entre objetos, la cual refleja la analogía existente en el problema real.

Precedencia parcial. Las comparaciones no son efectuadas entre las descripciones completas de los objetos, sino entre sub-descripciones previamente seleccionadas (i.e. conforme el sistema de conjuntos de apoyo definido).

Frecuencia. En los algoritmos de votación, un objeto corresponderá a una clase, si este es más similar a los objetos de esa clase.

El modelo de algoritmo de votación es determinado por seis parámetros. Cada uno de ellos, puede cambiarse de acuerdo al problema a resolver. Este hecho caracteriza a esta familia de algoritmos. Los parámetros que definen un algoritmo de votación son los siguientes:

Sistema de conjuntos de apoyo $\{W\}$. El sistema de conjuntos de apoyo determina que partes de los objetos, serán comparados al aplicarse el algoritmo. Cualquier subconjunto del conjunto potencia de los atributos puede ser usado como un sistema de conjunto de apoyo. Por ejemplo, pueden tomarse en consideración aquellos subconjuntos con un cardinal fijo de atributos, el conjunto de los testores típicos [11], entre otros.

Función de semejanza: ($FS: M_{i_1} \times \square \times M_{i_q}{}^2 \square\square$). Esta función determina como deben ser comparadas las sub-descripciones de los objetos.

Función de evaluación entre objetos para un conjunto de apoyo fijo ($F_{W_j}: M_{i_1}$ x □ x $M_{i_q}{}^2$ □ □). Esta función determina el valor de la semejanza entre el objeto a clasificar *O* y cada uno de los objetos del conjunto de datos pertenecientes a *MA*, para cada conjunto de apoyo fijo W_j. El resultado de esta función es el voto generado por cada objeto de *MA* con respecto al objeto a clasificar *O*, tomando solamente los atributos del conjunto de apoyo considerado W_j.

Esta función puede considerar el peso asignado a cada objeto de *MA*, además de los pesos asignados a los atributos del conjunto de apoyo considerado. Las funciones (1) y (2) son ejemplos de estas funciones. Donde $P_O(O_i)$ es el peso del objeto O_i y $P_X(X_i)$ es el peso del atributo X_i

$$F_{W_j}(W_j) = FS(W_j O_i, W_j O) \quad (1)$$

$$F_{W_j}(W_j) = P_O(O_i) * P_X(X_i) * FS(W_j O_i, W_j O) \quad (2)$$

Función de evaluación por clase para un conjunto de apoyo fijo ($FC_{W_j}: \square\square\square$). Esta función contabiliza todas las evaluaciones realizadas entre los objetos de *MA* y el nuevo objeto a clasificar *O* con respecto a cada clase K_t, para un conjunto de apoyo fijo W_j. El resultado de esta función es el voto generado por cada clase para el objeto a clasificar *O*, con respecto al conjunto de apoyo fijo W_j. Algunos ejemplos de estas funciones son dados en (3) y (4), donde $[v_t]$ es el número de objetos que contiene la clase K_t.

$$FC_{W_j}(t) = \frac{1}{[v_t]} * \sum F_{W_i}(W_j) \quad (3)$$

$$FC_{W_j}(t) = \{1 \text{ si } \sum F_{W_i}(W_j) >= d, d > 0; 0 \text{ en otro caso}\} \quad (4)$$

Función de evaluación por clase para el sistema de conjuntos de apoyo ($F_{\{W\}}: \square\ \square\ \square$). Sumariza todas las evaluaciones por clase efectuadas para el objeto a clasificar *O*, para el sistema de conjuntos de apoyo completo. Al resultado que genera esta función se le llama el voto dado por cada clase hacia el objeto *O* a clasificar, para todos los conjuntos de apoyo procesados. Algunos ejemplos de estas funciones son mostrados en (5) y (6), donde $[\{W\}]$ es el número de conjuntos de apoyo que contiene $\{W\}$.

$$F_{\{W\}}(t) = \frac{1}{[\{W\}]} * \sum FC_{W_i}(t) \quad (5)$$

$$F_{\{W\}}(t)=\left\{1 \text{ si} \sum FC_{W_i}(t) >= d, d > 0; 0 \text{ en otro caso}\right\} \quad (6)$$

Regla de solución S (*S: B^c □□; P_i: □□ B;* donde $B = \{0,1\}$). Sumariza todas las evaluaciones globales obtenidas por cada clase. Esta función determina a que clase(s) corresponde el objeto a clasificar. La regla de solución tiene la forma $S = \left(P_1\left(F_{\{w\}}(i)\right), P_2\left(F_{\{w\}}(i)\right), \ldots, P_c\left(F_{\{w\}}(i)\right)\right)$ donde $P_i\left(F_{\{w\}}(i)\right) = 1$ si el objeto a clasificar es asignado a la clase i, y $P_i\left(F_{\{w\}}(i)\right) = 0$ en otro caso. Un ejemplo de esta función es mostrada en (7).

$$P_i\left(F_{\{W\}}(i)\right)=\left\{1 \text{ si} F_{\{W\}}(i) \geq F_{\{W\}}(j), \forall i \neq j; 0 \text{ en otro caso}\right\} \quad (7)$$

Con estos parámetros definidos, los algoritmos de votación tienen las siguientes etapas:

a) Determinación de los parámetros del modelo

b) Aplicación de la función de evaluación entre objetos, para el objeto a clasificar con cada conjunto de apoyo.

c) Ejecución de la función de evaluación por clase, para cada conjunto de apoyo.

d) Aplicación de la función de evaluación por clase, para todo el sistema de conjuntos de apoyo.

e) Aplicación de la regla de solución que determinará a que clase(s) corresponderá el objeto a clasificar.

El algoritmo INC-ALVOT propuesto

INC-ALVOT es un algoritmo incremental, el cual procesa objeto por objeto del conjunto de datos que se vaya procesando. El algoritmo propuesto no almacena el conjunto de datos en la memoria principal, solamente guarda y maneja algunas estructuras simples las cuales conservan las operaciones parciales entre las ecuaciones para comparar objetos con las funciones de evaluación de clases para los conjuntos de apoyo, evaluación por clases para todo el sistema de conjunto de apoyo, así como la regla de solución.

Por cada objeto del conjunto de datos que INC-ALVOT procesa, se genera una clasificación parcial del objeto a clasificar y al procesarse todos los objetos del conjunto de datos el algoritmo generará la misma clasificación que el algoritmo ALVOT clásico, con la diferencia de que INC-ALVOT, podrá continuar anexando nuevos objetos en el conjunto de datos, con la misma filosofía de procesamiento que con los objetos iniciales. Este hecho garantiza que la precisión de la clasificación no decrece.

El procedimiento que realiza el algoritmo propuesto, le permite manejar y procesar conjuntos de datos que rebasen la capacidad de la memoria principal.

La principal diferencia entre el algoritmo clásico ALVOT y el propuesto, radica en que ALVOT basa sus comparaciones entre el objeto a clasificar y los del conjunto de datos. De manera diferente, INC-ALVOT basa sus comparaciones entre cada objeto del conjunto de datos y el objeto a clasificar, permitiéndole utilizar los resultados de los cálculos efectuados con los objetos ya procesados, con los objetos restantes del conjunto de datos.

Este funcionamiento le permite al algoritmo propuesto manejar nuevos objetos añadidos en el conjunto de datos, como cualquier otro objeto ya incluido en el conjunto mencionado. A diferencia de ALVOT, el cual debe volver a procesar nuevamente todos los objetos con los cambios realizados en el conjunto de datos.

Las variables utilizadas por el algoritmo propuesto, además de su tipo, dimensiones y valor inicial se muestran en la tabla 1. Los símbolos usados en esta tabla, así como su significado son los siguientes: E – Estática; A – Arreglo; M – Matriz; n – número de atributos; t_j – cardinal del conjunto de apoyo j; C_w - cardinal del conjunto {W} ; c – número de clases.

Tabla 1 Características de las variables utilizadas en el algoritmo

Nombre del elemento; estructura usada	*Tipo*	*Dimensión*	*Valor inicial*
O (objeto a clasificar; A)	E	n	$[X_1,\cdots,X_n]$
O_j (objeto tomado de MA; A)	E	n	$[X_1,\cdots,X_n]$
W_j (conjunto de apoyo W_j ; A)	E	t_j n	$\left[X_{j_1},\cdots,X_{j_t}\right]$
FC_{W_j} (evaluación por clase para cada W_j; M)	E	t^x C_w	[0,0,...,0,...,0,0,...,0]
$F_{\{W\}}$ (evaluación por clase para todos los W_j;A)	E	c	[0,0,...,0]
S (salida o valor de la clasificación; A)	E	c	[0,0,...,0]

Nota: La descripción del arreglo $[X_1,\cdots,X_n]$ indica que éste toma el valor definido para el objeto respectivo, es decir O (objeto a clasificar) u O_j (objeto tomado de MA). De igual manera, la descripción del arreglo $\left[X_{j_1},\square,X_{j_{tj}}\right]$ indica que éste toma valores de un subconjunto de variables, correspondientes a la definición de cada conjunto de apoyo W_j

A continuación, se describe el algoritmo INC-ALVOT propuesto.

Entrada: O (objeto a clasificar); O_i (objeto tomado de MA)

Salida: $S=(P_1(O),P_2(O),...,P_c(O))$ *(clasificación generada hasta el objeto O_i parcialmente)*

Para el objeto $O_i \in K_t \subseteq MA$

Calcular $FC_{W_j}(t,O,W_j)$ donde $O_i \in K_t$, para cada conjunto de apoyo W_j, incrementando solamente los valores de la clase K_t

Aplicar $F_{\{W\}}(t\,O)$ incrementando solamente los valores de la clase K_t

Modificar $S=(P_1(O),P_2(O),...,P_c(O))$, tomando exclusivamente el valor de la clase K_t

El algoritmo incremental presentado, va generando una clasificación parcial conforme va procesando cada objeto O_i de la muestra de aprendizaje, incrementando solamente aquellos valores en los arreglos o matrices que representan a las funciones FC_{W_j} y $F_{\{W\}}$, en el lugar correspondiente a la clase que pertenece el objeto O_i. De esta manera, se va generando de manera incremental la clasificación del objeto O, guardando en los arreglos previamente descritos, el valor que aporta cada objeto O_i de la muestra de aprendizaje, para cada conjunto de apoyo W_j definido. Siguiendo este proceso, se va generando una clasificación parcial del objeto O, tomando en cuenta hasta el último objeto O_i procesado. El pseudocódigo del algoritmo se expone en el apéndice A.

Al procesarse el último objeto de *MA*, la clasificación generada dada en *S*, será la misma que la obtenida por el algoritmo ALVOT clásico. Este hecho se basa en la proposición 1.

Discusión del algoritmo. El algoritmo propuesto, por su naturaleza incremental, es capaz de procesar una cantidad considerable de funciones para realizar la evaluación por clase tanto para un conjunto de apoyo fijo (FC_{W_j}), como para el sistema de conjuntos de apoyo ($F_{\{W\}}$). Esta familia de funciones debe ser expresada por ejemplo, en términos de sumatorias y productos que contemplen la evaluación del objeto tomado de la muestra de aprendizaje, y al mismo tiempo, conserven los resultados previamente generados con los objetos ya procesados. De esta manera, si se usara una función similar a la expresada en (3), se guardaría el resultado de la suma que aporte el objeto de la muestra de aprendizaje, con la suma previamente generada con los objetos ya proce-

sados. Entonces, cada vez que se procese un objeto de la muestra de aprendizaje, se actualiza y guarda la suma parcial de estos elementos, y el factor multiplicativo se generará solamente como el número de objetos ya procesados correspondientes a la clase en cuestión (este valor se puede manejar por medio de un contador). El valor de la función retornado dependerá de la suma parcial generada y del factor multiplicativo parcialmente generado.

De igual manera, existen funciones como el caso de la mediana, las cuales no puedan ser representadas de la manera anteriormente explicada, y entonces no se podrían transformar en una función equivalente que pueda calcularse de manera incremental. El algoritmo propuesto no está concebido para manejar este tipo de funciones.

Proposición 1. Sea dado un conjunto de datos con m-objetos. El resultado al aplicar INC-ALVOT m-veces a este conjunto de datos (sin repetir objetos) generará el mismo resultado de clasificación que al aplicar el algoritmo ALVOT al mismo conjunto de datos.

Demostración. Se realizará por inducción.

Para k=1 objeto. Al aplicar ALVOT, tomando solamente un objeto del conjunto de datos, se calcula FC_{W_j} y luego $F_{\{W\}}$, modificando exclusivamente el valor de la clase K_t, debido a que el objeto procesado pertenece a esta clase. Cuando se aplica INC-ALVOT a este mismo objeto, se calcula FC_{W_j}, para cada conjunto de apoyo W_j, para posteriormente calcular $F_{\{W\}}$, lo cual incrementará solamente los valores de la clase K_t a la cual pertenece el objeto procesado. Los restantes valores permanecen en cero. Como se usan las mismas funciones de evaluación en ambos algoritmos, se genera el mismo resultado de clasificación para ambos.

Suponiendo que se cumple para k=m-1 objetos, se demuestra que se cumple para k=m objetos.

Al aplicar ALVOT a todo el conjunto de datos el algoritmo realiza los mismos cálculos para los conjuntos de apoyo fijo, de FC_{W_j} y de $F_{\{W\}}$ que al procesar los m-1 objetos, más los incrementos en estas funciones sobre la clase K_t correspondientes al último objeto procesado del conjunto de datos. En este paso, se ha aplicado INC-ALVOT m-1 veces sobre el conjunto de datos, faltando por procesar un último objeto del conjunto de datos (el objeto m). Al aplicar nuevamente INC-ALVOT a este último objeto, se calcula FC_{W_j} y $F_{\{W\}}$ incrementando solamente el valor de la clase K_t, a la cual pertenece el último objeto procesado del conjunto de datos, generando entonces el mismo resultado de clasificación que ALVOT aplicado en todo el conjunto de datos con m-objetos. Por lo tanto, el resultado de clasificación aplicando INC-ALVOT m-veces al conjunto de datos, genera el mismo resultado de clasificación cuando se aplica ALVOT al mismo conjunto de datos.

Experimentación

En esta sección, se muestran dos ejemplos utilizando el algoritmo ALVOT clásico, y el algoritmo incremental INC-ALVOT propuesto. Los conjuntos de datos utilizados fueron tomados de [12]. El primer ejemplo, consta de un subconjunto de diez objetos del conjunto de datos *car*, el cual se muestra en la tabla 2.

El objeto a clasificarse es: *O={low, low, 5more, more, big, high}*. Los resultados obtenidos por INC-ALVOT de manera parcial y al procesar todos los objetos del conjunto de datos, así como el resultado generado por el algoritmo clásico ALVOT son mostrados en la tabla 3. En esta tabla, se muestra y remarca el mayor valor obtenido al procesar cada nuevo objeto del conjunto de datos (en este caso, desde O_1 hasta O_{10}), obteniendo al final del último objeto procesado la clasificación final del objeto, la cual es la misma que la generada por el algoritmo ALVOT clásico. En la tabla, se puede visualizar que cada vez que un nuevo objeto del conjunto de datos es procesado, el valor de la clasificación es incrementada. INC-ALVOT, después de procesar todos los objetos del conjunto de datos de prueba, clasifica al objeto *O* en la clase vgood, coincidiendo con el resultado obtenido por ALVOT clásico.

Tabla 2 Subconjunto de diez objetos tomados de car

Objeto	X_1	X_2	X_3	X_4	X_5	X_6	*Clase*
1	low	low	5more	4	med	low	unacc
2	low	low	5more	4	med	med	good
3	low	low	5more	4	med	high	vgood
4	low	low	5more	4	big	low	unacc
5	low	low	5more	4	big	med	good
6	low	low	5more	4	big	high	vgood
7	low	low	5more	more	small	low	unacc
8	low	low	5more	more	small	med	acc
9	low	low	5more	more	small	high	good
10	low	low	5more	more	med	low	unacc

Para el segundo ejemplo, se utilizaron 4 conjuntos de datos reales, tomados de [12]: Zoo, Cars, Nursery y Mushroom. Se consideraron 96 objetos para Zoo, 1.723 para Cars, 12.954 y finalmente 8.119 objetos de Mushroom. Además, para verificar la eficiencia del algoritmo propuesto cuando nuevos objetos son anexados al conjunto de datos original, se anexaron 4 objetos para Zoo, Cars y Mushroom, y 5 para Nursery, obteniendo así nuevos conjuntos de datos actualizados.

Los conjuntos de datos no estáticos (i.e. actualizados) donde nuevos objetos son añadidos, eliminados o modificados, son de especial interés en áreas como Minería de Datos, donde uno de los requerimientos altamente deseables que deben cumplir los algoritmos es que sean eficientes ante cambios realizados en los conjuntos de datos [13], no debiendo procesar nuevamente todo el conjunto de datos actualizado. El algoritmo propuesto, fue concebido para ser eficiente ante la adición de nuevos objetos en el conjunto de datos. En este trabajo, no se muestra la aplicación del algoritmo propuesto a ningún problema real de clasificación en particular. Sin embargo, existen problemas reales de clasificación, para los cuales es adecuado el uso del algoritmo propuesto, debido a que van incrementando el número de instancias de la muestra de aprendizaje. O donde el número de elementos de la muestra de aprendizaje es más grande que el tamaño de la memoria principal de la computadora. Algunos problemas de este tipo son: a) la detección y clasificación de casos en epidemias en la población (como la influenza H1N1), ya que puede aumentar la muestra original de pacientes contagiados con diferentes variantes del virus; b) la detección de fraudes realizados por pagos con tarjetas bancarias, donde se van incrementando e identificando las maneras de efectuar los fraudes mencionados.

En todos los experimentos, el sistema de conjuntos de apoyo se formó por combinaciones de atributos de longitud 2. También, fueron usadas las ecuaciones (1), (3), (5) y (7) en las diferentes etapas del algoritmo INC-ALVOT.

Los algoritmos fueron implementados en Java, en una PC con procesador Pentium IV, con 1 Gigabyte de Memoria RAM, y bajo el sistema operativo SUSE LINUX 9.2.

En la tabla 4, es mostrado el tiempo de ejecución de los algoritmos ALVOT e INC-ALVOT. En todos los experimentos realizados, INC-ALVOT mejoró los tiempos obtenidos por ALVOT clásico. En algunos casos, INC-ALVOT fue 39 veces más rápido que ALVOT (con Mushroom).

Tabla 3 Resultados parciales y final obtenidos por INC-ALVOT y ALVOT clásico

0_1	0_2	0_3	0_4	0_5	0_6	0_7	0_8	0_9	0_{10}	*ALVOT*	*Clase*
7,5	**7,5**	7,5	8,75	8,75	8,75	9,166	9,166	9,166	9,375	9,375	unacc
0	**7,5**	7,5	7,5	8,75	8,75	8,75	8,75	9,999	9,999	10,0	good
0	0	**10,0**	**10,0**	**10,0**	**11,25**	**11,25**	**11,25**	**11,25**	**11,25**	11,25	vgood
0	0	0	0	0	0	0	10,0	10,0	10,0	10,0	acc

Tabla 4 Tiempos de ejecución en segundos de los algoritmos INC-ALVOT y ALVOT clásico, para diferentes conjuntos de datos reales

Conjunto de datos	*Número objetos*	*Número atributos*	*Tiempo INC-ALVOT*	*Tiempo ALVOT*
Zoo	96	18	0,204	0,214
Cars	1.723	6	0,232	0,910
Nursery	12.954	8	2,294	85,923
Mushroom	8.119	22	43,171	1.713,718

En la tabla 5, son mostrados los tiempos necesitados para realizar la clasificación de un objeto, cuando fueron agregados nuevos objetos a las muestras de aprendizaje iniciales. De esta manera, INC-ALVOT solamente procesa los nuevos objetos añadidos a la muestra de aprendizaje, al contrario que ALVOT, el cual debe contemplar nuevamente todos los datos del conjunto inicial más los añadidos. La diferencia de tiempos obtenida en estos experimentos es muy notoria, por ejemplo, cuando Nursey y Mushroom son procesados.

Tabla 5 Tiempos de ejecución en segundos de los algoritmos INC-ALVOT y ALVOT clásico, al añadir varios objetos en los conjuntos de datos reales iniciales

Conjunto de datos	*Número objetos añadidos*	*Número atributos*	*Tiempo INC-ALVOT*	*Tiempo ALVOT*
Zoo	4	18	0,006	0,099
Cars	4	6	0,001	0,889
Nursery	5	8	0,001	79,566
Mushroom	4	22	0,006	1.644,169

Finalmente, en la figura 1 se muestra el tiempo de ejecución de los algoritmos ALVOT e INC-ALVOT, al procesar el conjunto de datos Mushroom, de 1.000 en 1.000 objetos, hasta completar los 8.119 que conforman la muestra de aprendizaje en estudio. Esto significa que a la muestra de aprendizaje inicial, se le adicionaron 1.000 objetos y se vuelve a procesar. Posteriormente, a la nueva muestra de aprendizaje generada, se le vuelven a adicionar otros 1.000 objetos y se procesa nuevamente. Y así sucesivamente hasta completar el número de objetos que componen la muestra de aprendizaje.

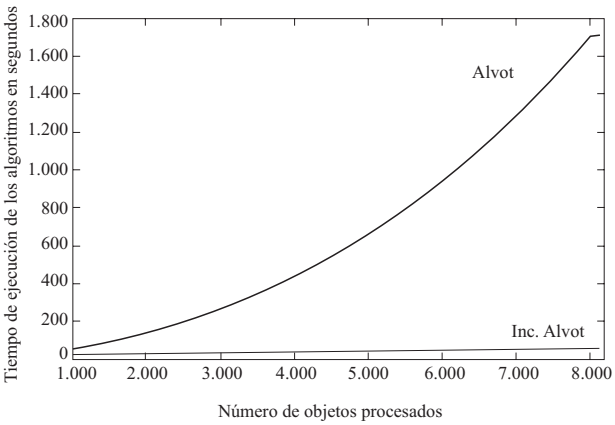

Figura 1 Tiempo de ejecución de los algoritmos, cuando se procesa Mushroom, tomando de 1.000 en 1.000 objetos

En general, estos experimentos muestran que INC-ALVOT tiene un mejor desempeño que ALVOT clásico, lo cual es reflejado en el tiempo de ejecución utilizado por ambos algoritmos

Conclusiones

En este artículo se presentó un algoritmo incremental de clasificación supervisada, denominado INC-ALVOT. El algoritmo propuesto permite manejar conjuntos de datos sin mantenerlos en la memoria principal, además de permitir anexiones de nuevos objetos en los conjuntos de datos iniciales, requiriendo solamente un mínimo tiempo de procesamiento para reclasificar un objeto en cuestión, debido al funcionamiento incremental del algoritmo propuesto. Este hecho es posible, debido a que INC-ALVOT mantiene todos los cálculos de los objetos previamente procesados, al contrario del algoritmo clásico ALVOT, el cual realiza los cálculos de todos los objetos sin guardar información anteriormente procesada, incluyendo los nuevos objetos añadidos en los conjuntos de datos.

De la experimentación realizada, se puede concluir que el algoritmo incremental propuesto tiene un mejor desempeño que el algoritmo ALVOT clásico. Pudiéndose notar el crecimiento cuadrático del tiempo para ALVOT, contra el crecimiento mostrado por el algoritmo INC-ALVOT propuesto.

El trabajo futuro incluye la paralelización del algoritmo, además del desarrollo de una fase decremental, cuando se eliminen objetos en vez de añadirlos en los conjuntos de datos.

Referencias

1. J. F. Martínez Trinidad, A. Guzman Arenas. "The logical combinatorial pattern recognition an overview through selected works". *Pattern Recognition*. Vol. 34. 2001. pp. 741-751.
2. J. Ruiz Shulcloper, M. Lazo Cortés. "Mathematical algorithms for the supervised classification based on fuzzy partial precedence". *Mathematical and Computer Modeling*. Vol. 29. 1999. pp. 111-119.
3. E. López Espinoza, J. Carrasco Ochoa, J. F. Martinez Trinidad. "Two floating search strategies to compute the suppot sets system for ALVOT". *Proc. CIARP 2004. LNCS.* Ed. Springer-Verlag. Puebla (México). 2004. pp. 677-684.
4. J. Carrasco Ochoa, J. F. Martínez Trinidad. "Editing and training for ALVOT, an evolutionary approach". *Proc. IDEAL 2003. LNCS.* Ed. Springer-Verlag. Hong Kong. 2003. pp. 452-456.
5. Y. Moyao Martinez. *Programa paralelo para el cálculo del sistema de conjuntos de apoyo del algoritmo de clasificación CR+*. Tesis de Maestría en Ciencias de la Computación. BUAP. Puebla. México. 1998. pp. 1-32.
6. D. Aha, D. Kibler, M. Albert. "Instance-based learning algorithms". *Machine learning*. Vol. 6. 1991. pp. 37-66.
7. J. Carrasco Ochoa. *Sensibilidad en el enfoque lógico combinatorio de patrones*. Tesis de Doctorado en Ciencias de la Computación. CIC-IPN. México. 2001. pp. 10-70.
8. A. Pons Porrata, G. Sánchez Díaz, M. Lazo Cortés, L. Alfonso Ramírez. "An incremental clustering algorithm base don compact set with radius α". *Proc. CIARP 2005. LNCS.* Ed. Springer-Verlag. La Habana (Cuba). 2005. pp. 518-527.
9. G. Sánchez Díaz, J. Ruíz Shulcloper. "A clustering method for very large mixed data sets". *Proc. IEEE ICDM 01*. San José (CA). 2001. pp. 643-644.
10. J. Ruíz Shulcloper, A. Guzmán Arenas, J. Martínez Trinidad. *Enfoque lógico combinatorio al reconocimiento de patrones*. *Serie avances en reconocimiento de patrones*. Edit. Instituto Politécnico Nacional. México. 1999. pp. 59-75.
11. .M. Lazo Cortés, J. Ruíz Shulcloper, E. Alba Cabrera. "An overview of the evolution of the concept of testor". *Pattern Recognition*. Vol. 34. 2001. pp. 753-762.
12. A. Asunción, D. J. Newman. "UCI repository of machine learning databases http://www.ics.uci.edu/~mlearn/MLRepository.html. Consultada el 11 de octubre de 2008. Irvine. CA: University of California. Department of information and computer science. 1998. pp. 1-1.
13. M. H. Dunham. *Data Mining. Introductory and Advances Topics*. Ed. Pearson Education Inc. New Jersey. 2003. pp. 3-20.

Apéndice A. Pseudocódigo del algoritmo INC-ALVOT

Entrada: O (objeto a clasificar); $O_i \in MA$ (i-ésimo objeto de la muestra de aprendizaje)

Salida: $S = [P_1(O_i), P_2(O_i), \ldots, P_c(O_i)]$ (clasificación generada hasta el objeto O_i)

Paso 1: Para cada K_t, donde $O_i \in K_t$ hacer

Para cada $W_j \in \{W\}$ hacer

D_1 = Calcula_Factor($Oi, Wj,$ opc1)

$$FC\_TEMP_{W_i}[t][j] = FC\_TEMP_{W_i}[t][j] + FS(W_j O_i, W_j O)$$

$$FC_{W_i}[t][j] = D * FC\_TEMP_{W_i}[t][j]$$

D_2 = Calcula_Factor($Oi, Wj,$ opc2))

$$F\_TEMP_{\{W\}}[t] = F\_TEMP_{\{W\}}[t] + FC_{W_j}[t][j]$$

$$F_{\{W\}}[t] = D_2 * F\_TEMP_{\{W\}}[t]$$

Paso 2: Asigna_Clase(S);

Asigna_Clase(S)

Para cada $i = 1,\ldots,c$ hacer

Si $F_{\{W\}}[i] \geq F_{\{W\}}[j]$, $i \neq j$ entonces

$S[i] = 1$; (es decir $P_i(O_i) = 1$,

el objeto pertenece a la clase i)

En otro caso

$S[i] = 0$; (es decir $P_i(O_i) = 0$,

el objeto no pertenece a la clase i)

La función $FS(W_j O_i, W_j O)$ retorna la semejanza entre los objetos O y O_i, considerando solamente los atributos definidos en el conjunto de apoyo W_j. Calcula_Factor(O_i, W_j, opc) retornará un valor que multiplique la sumatoria parcial realizada, como en la ecuación (3) y (5). Estos valores pueden depender de los O_i, y/o W_j en cuestión, y opc, le indicarán a la función que tipo de ecuación debe evaluarse. Si no debe anexarse un factor multiplicativo, entonces esta función puede retornar el valor 1, para no afectar los cálculos realizados. $FC\_TEMP_{W_i}[t][j]$ guardará los valores de la sumatoria que se hayan realizado hasta el objeto O_i, conservando intactos estos valores. $FC_{W_i}[t][j]$ Almacenará el valor de la sumatoria parcial, por el factor multiplicado que puede ser considerado en el cálculo. De igual manera, $F\_TEMP_{\{W\}}[t]$ conservará sin modificar los valores de la sumatoria de los $FC_{W_i}[t][j]$.