VISIÓN ELECTRÓNICA

Preparación de Artículos revista VISIÓN ELECTRÓNICA: algo más que un estado sólido
Fecha de envío: 12 Febrero de 2016
Fecha de recepción: 13 Marzo 2016
Fecha de aceptación: 22 Abril2016

A Case-Study Vision

A RESEARCH COMPARATIVE AMONG ASSOCIATION RULES ALGORITHMS

UNA INVESTIGACIÓN COMPARATIVA ENTRE ALGORITMOS DE REGLAS DE ASOCIACIÓN

July Andrea Rocha Hernández[1] Diana Milena Rodríguez Herrera[2] Jorge Enrique Rodríguez Rodríguez[3]

Abstract: In this paper we propose three algorithms for the data association and presents a description of each with the purpose of showing how it performs the associations that leads to the association rules. Afterwards, a comparison among this is performed to determine its efficiency and effectiveness in the results. To do it, the algorithm complexity is to be found, and the quality of the results and the rules generated are evaluated. At the same time, different sets of data will be used to test each algorithm. Finally, the conclusions obtained from this research will be presented.

Keywords: Data association, a priori algorithm, frequent patterns tree, data mining, association rules, and depth search.

Resumen: En este artículo se proponen tres algoritmos para la asociación de datos y se presenta una descripción de cada uno de ellos con el propósito de mostrar cómo realiza las asociaciones que conducen a las reglas de asociación. Posteriormente, se realiza una

[1] BSc. In System Technology, Universidad Distrital Francisco José de Caldas, Colombia. E- mail: jarochah@correo.udistrital.edu.co

[2] BSc. In System Technology, Universidad Distrital Francisco José de Caldas, Colombia. E-mail: dimrodriguezh@correo.udistrital.edu.co

[3] MSc. In Systems Engineering, Universidad Nacional de Colombia, Colombia. Current position: professor Universidad Distrital Francisco José de Caldas, Colombia. E-mail: jerodriguezr@udistrital.edu.co

comparación entre éstos para determinar su eficiencia y efectividad en los resultados. Para lo anterior, se encuentra la complejidad del algoritmo y se evalúa la calidad de los resultados y las reglas generadas. Al mismo tiempo, se utilizan diferentes conjuntos de datos para probar cada algoritmo. Finalmente, se presentan las conclusiones obtenidas en la investigación.

Palabras clave: Asociación de datos, algoritmo apriori, árbol de patrones frecuentes, minería de datos, reglas de asociación, búsqueda de profundidad.

1 Introduction

With the inclusion of new technologies in databases and the management of information systems that the experts have been adopting , the data mining is introduced as a fundamental piece when it comes to use the knowledge that can be generated from a big set of transactions, generally used in the marketing area, what it’s important to observe is that it does not only have a commercial focus; but that the whole database can be analyzed and transformed if a proper KDD (Knowledge Discovery in Databases) [1] is performed, obtaining from this, useful knowledge to support the decision making process. According to the current problem in the organizations that are rich in data but poor in knowledge.

The data association task in the data mining phase, stores the algorithms for association that are in synthesis this paper objective; the A priori algorithm, known for being the pioneer of the algorithms that have been coming out across time for association [2-4], among them the Eclat and FP-Growth algorithms, which were selected as potential candidates considering its popularity and the good results generated by the association rules, with the purpose of determining which data association algorithm to implement when it comes to data mining

Fecha de envío: 12 Febrero de 2016
Fecha de recepción: 13 Marzo 2016
Fecha de aceptación: 22 Abril2016

2 Data Association

The association rules are used, because in the first place, is a relatively easy model to interpret for people (executives, department directors, managers) that make decisions and to implement for the software developers [5].

Secondly, because these offer reliable results due to the incorporation of algorithms, such as A priori, FP-growth and Eclat; and to the rate of veracity that this implies, because it performs an exhaustive research among the data to generate rules that satisfies the user needs.

The same way, it is used to relate data behaviors, keeping in mind the frequency in which a set of items involves the occurrence of another one in a transactional database, being this one a very used technique in the data mining.

2.1 Association rules

An association rule is an affirmation to the form $A \Rightarrow B$ where A, B are sets of items (evens attribute – value) of each transaction [6]; where the frequency is evaluated which{ B} its consequent to the{ A} background taking into consideration the confidence and support values that makes out these “strong” rules [7-8]. A transaction is the set of items, also called tuple database, in which a unique identifier called TID is assigned to it, the confidence of an association rule is the probability of a transaction that contains {A} also contains {B}. The support is the probability that a transaction contains {A} same as {B} {A, B}.

It’s been said that an association rule is interesting if its support and its confidence are higher or equal to the minimum support and confidence thresholds. [9]

2.1.1 Support

Is a measure criteria, that finds the occurrences of the background and consequent of a rule in the transactional set related to the total of data transactions and is given by the equation 1:

$$\text{support}(A \Rightarrow B) = Frequency(A \cup B) \qquad (1)$$

2.1.2 Confidence

This criteria determines the value of an association rule { $A \Rightarrow B$} and it’s given by the number of transactions that includes both {A and B} respectively, among the number of transactions that includes {A}, according to equation 2.

$$Confidence(A \Rightarrow B) = \frac{\text{support}(A \cup B)}{\text{support}(A)} \qquad (2)$$

2.1.3 Lift

This criteria determines the dependence degree of a rule’s terms { $A \Rightarrow B$}, it stablishes the occurrence of {A} in case {B} occurs and vice versa and it’s given by equation 3.

$$Lift(A \Rightarrow B) = \frac{confidence(A \Rightarrow B)}{\text{support}(B)} \quad (3)$$

If the lift is equal to 1, meaning that if the background and the consequent are totally in independent terms, if on the opposite it’s a value > 1 means that the occurrence of the items from {B} will have influence in the probability of occurrence in the items from {A}, now if the value of such Lift is < 1 it means that the frequency of the consequent will have influence in the probabilities of the antecedent not occurring. [10]

An example of a rule in general could be:

Rule1. If cond1 ˄ cond2 ˄ … $\Longrightarrow$ conclusion
[support, confidence, lift]

3 Data Association Algorithms

3.1 A PRIORI Algorithm

The A priori, the precursor of the algorithms based on data association, performs multiple interactions in the dataset with the purpose of determining the itemset with frequent items, in first place, the minimum support is established, afterwards the algorithm obtains the items which frequency exceeds such threshold, obtaining this way the item set L[1] of frequent items. Starting from the last itemset L[k] of frequent items obtained, it generates the itemset [k+1] of potentially frequent items named candidate items C[k] and the frequency of these items is obtained to choose those that are frequent only, such items will be included in the itemset L [k+1], this process will be repeated until no more frequent items are found [11].In table 1 the algorithm is shown.

A priori algorithm. Finds the frequent items.
Entries: Transactions from a data base D; min_sop
Exits: L, frequent itemsets from database D.
Method:
(1) L_1= Find_1-itemsets_frequent(D);
(2) $for\ (k = 2; L_{k-1}$ 1Æ$; k++)$
(3) $C_{k=}$generate A priori $(L_{k-1}\ ,\min\_sop)$;
(4) $for\ each\ transaction\ t$ Î D {// examine D for the counter

```
    (5) $C_{t}$= sub-itemsets $(C_{k-t})$ // obtain the sub itemset t that are candidates
    (6) $for\ each\ candidate\ c\ Î\ C_t$
    (7) $c.counter++$
    (8) $end\ to$
    (9) $L_{k}= \{c\ Î\ C_k | c.counter^3 \min\_sop\}$
   (10) End to
   (11) $Return\ L = U_k L_k;$
Function generate A priori $(L_{k-1}:(k-1) - frequent\ itemsets; \min\_sop$ : minimum support)
    (1) $for\ each\ itemset\ l_1\ Î\ L_{k-1}$
    (2) $for\ each\ itemset\ l_2\ Î\ L_{k-1}$
    (3) $si\ \left(l_1[1] = l_2[1]\ Ù\ (l_1[2] = l_2[2])\ Ù\ ...\ Ù\ (l_1[k-2] = l_2[k-2])Ù\ (l_1[k-1] < l_2[k-1])\right)$
Then{
    (4) $c = l_1\ X\ l_2;$ // generate Candidates
    (5) If sub-itemset_unfrequent $(c, L_{k-1})$ so
    (6) Eliminate $c$;
    (7) If not
    (8) Add $c\ to\ C_k$
    (9) End- to
  (10)Return $C_k$;
Function: sub-itemset_unfrequent $(c: k - candidate\ itemset;\ L_{k-1} : (k-1) - frequent\ itemset)$
   (1) For each $(k-1) - subitemset\ from\ c$
   (2) If $(s\ Ï\ L_{k-1})$ so
   (3) $Return\ truth$;
   (4) $Return\ false$;
```

Tabla 1. Algoritmo A priori [11].

For example to an item set A= {0, 1, 2, 3}. This data itemset grows exponentially; such way if the number of items is n, then it is possible to generate $2^n + 1$ itemsets. This means that, if there is an item set with 100 items, it will exist $1{,}26 \times 10^{30}$ candidate itemsets C [K]; so the algorithm must go through the total number of transactions [12].

At the time to perform the rules generation process the A priori algorithm receives the itemsets collection of frequent items, analyzes its sizes and obtains all the possible sub-itemsets with the purpose of determining each of the backgrounds and consequents of each rule, if and only if meets the minimum confidence requirement.

At the moment of establishing the algorithm entry parameters, is suggested to set a low value for the support and a higher value for the confidence, this way a big number of rules will be produced, to be then examined by the confidence threshold. It's worth to mention that a rule established with a low confidence value will not externalize a behavior pattern in the database, however if the support value is significantly high, in consequence patterns, it will be lost.

The A priori Algorithm complexity is bounded for $O(C_{sum} \times |T|)$ where C_{sum} is the sum of the total of the candidate itemsets considered and $|T|$ denotes the size of the transactions itemset [12].

3.2 FP-GROWTH Algorithm

It's an algorithm with better performance than the A priori algorithm because it stores the information in an efficient way that allows it to compress the itemsets transaction to accomplish more accurate consultations, as it was proposed by [13]. Its structure is based on a FP-Tree (frequent pattern tree), this consist in a root node, that gives place to sub-trees which nodes make reference to the transactions, at this point it can be highlighted by watching that each node includes three fields: item information, a counter for the number of transactions that goes through the root branch to this node and finally a pointer to the next node that contains information about the same item if it exists, otherwise it will be an empty pointer. Additionally, this algorithm uses the hash table that accumulates information about the frequent items.

In the first instance the algorithm (see table 2) obtains the frequent itemsets from size 1, afterwards it inserts each transaction considering the occurrence of the item from higher to lower, according to its support in the FP-tree [14], this way the algorithm made itemsets recursively of frequent items while it goes through the tree, such tour begins with the item of less frequency in the hash table and this recursively its obtained from a conditional base pattern of frequent itemsets the algorithm obtains the association rules in an equivalent way to the A priori Algorithm [15].

Input: A database DB, represented by FP-tree constructed, and a minimum support threshold ξ.
Output: The complete set of frequent patterns.
Method: call FP-growth (FP-tree, null).
Procedure FP-growth(Tree, α){
(1) if Tree contains a single prefix path // Mining single prefix-path FP-tree
(2) then {
(3) let P be the single prefix-path part of Tree;
(4) let Q be the multipath part with the top branching node replaced by a null root;
(5) for each combination (denoted as β) of the nodes in the path P do

(6) generate pattern $\beta \cup \alpha$ with support = minimum support of nodes in β;
(7) let freq pattern set(P) be the set of patterns so generated; }
(8) else let Q be Tree;
(9) for each item ai in Q do { // Mining multipath FP-tree
(10) generate pattern β = ai $\cup$ α with support = ai .support;
(11) construct β's conditional pattern-base and then β's conditional FP-tree Treeβ ;
(12) if Treeβ = $\emptyset$
(13) then call FP-growth(Treeβ , β);
(14) let freq pattern set(Q) be the set of patterns so generated; }
(15) return(freq pattern set(P) $\cup$ freq pattern set(Q) $\cup$ (freq pattern set(P)
× freq pattern set(Q)))}

Table 2. FP-Growth Algorithm. Source: own.

The algorithms complexity depends on the trees depth and the number of items in hash table and it's given by the next formula $O(Nt \times Dt)$ where Nt is the number of items in the header chart and Dt is the tree's maximum deepness. [15]

3.3 ECLAT Algorithm

ECLAT (Equivalence Class Transformation) introduced in 1997; performs a deep search and adopts a vertical position to represent the database itemset, in which each component is represented by a transactional ID called TIDset and its transactions contains the items [16]. This algorithm reduces significantly the I/O operations because it checks the database once. It is based in performing a clustering among the items to get close to the frequent itemsets reducing the candidate itemset generation process.

It is proposed for the grouping two method that they are used after finding the frequent itemsets of two elements:

1. By equivalent classes: this technique, groups' itemsets that have the same first item.
2. By searching maximal cliques: an equivalent graph is generated which nodes are the items and the arches connect the items from the 2 frequent itemsets. The items are grouped by those who make maximal cliques [17]

Eclat represents the transactions in a bit matrix, in which each row corresponds to an item and each column to a transaction, a bit is placed in this matrix if the item correspondent to the row is included in the column corresponding to the transaction, if the item does not match the criteria then such item is eliminated [18]. The support of each sub-itemset is calculated by the intersection of the lists of the identifiers from each one of the transactions, this process is repeated for all of the transactions.

Fecha de envío: 12 Febrero de 2016
Fecha de recepción: 13 Marzo 2016
Fecha de aceptación: 22 Abril2016

Table 3 the algorithm is shown

```
ECLAT TIDset $( T, \beta )$
Input: T is vertical dataset and $\beta$ is support value.
Output: List of frequent item set
    1. Repeat step  for $i = 1\ to\ Li\ ! = NULL$
        (a)  Initialize $P_i\ = \{Li, T(Li)\ fall\ itemsets\ of\ T\ \}$
          $Li\ refers\ to\ the\ item\ name$
          $T(Li)\ refers\ to\ the\ transactions\ \ to$
$that\ particular\ transactions\ .$
        (b) $i = i + 1$
    2. Repeat  $for\ j = 1\ to\ Pj\ ! = NULL$
        (a)  $Xj = Lj\ and\ \ Yj = T(Lj)$
        (b)  $j = j + 1$
    3. Set value of $i = 1$  and  $k = 1$
    4. Repeat for ( $Xj\ ! = NULL$  and $(Yj\ ! = NULL$
        ☎♋✆  $J = i + 1$
        ☎♌✆  $Nij = Xi\ \cup Xj\ \ and\ T(Nij)\ Yi\ \cap Yj$
        ☎♍✆  $if\ count\ of\ T\ (Nij) >\ \beta\ then$
$Bk = Nij\ and\ k = k + 1$
        ☎♎✆  $i = i + 1$
    5. Print $(Bk)$
    6. End
```

Table 3. ECLAT Algorithm [19].

This algorithm complexity is given by the equation $O(k(k+1))$ in which k is the number of items [19].

4 Cases of Study

To obtain useful results out of the knowledge extraction process KDD is important to make an accurate pre data processing [20], this way at the moment the data is entered to perform the association, and these can be found with no noise; for this data association particular case in which the data type will be categorical. Afterwards the algorithms are compared using tree datasets: Titanic, Bank and Nursery (Taken from www.rstudio.com and archive.ics.uci.edu/ml).

4.1 Data Preprocessing

4.1.1 Titanic Dataset

This dataset has information about the people on board the Titanic ship, which contains 2201 entries and these at the same time have 4 attributes: Class, Gender, Age and Survived (Table 4).

Class	Sex	Age	Survived
1st	Female	Child	yes
2nd	Male	Adult	No
3rd	Female	Child	No
2nd	Male	Child	No
3rd	Male	Adult	yes
Crew	Female	Adult	yes
Crew	Male	Adult	No

Tabla 4. Titanic dataset information. Source: own.

4.1.2 Bank Dataset

This dataset has 41188 entries and 21 attributes, it has information on previous telephonic campaigns; whit the purpose of linking possible customer to a long term money deposit. Among other customer information, it has: age, marital status, banking questions and linking.

It was noticed that there was a redundancy in the data, so it was necessary to reduce it to nine attributes, selecting the most relevant ones. The same way it was necessary to create discreet variables for some attributes.

4.1.3 Nursery Dataset

Includes information about requests made to children schools around the decade of the 80s; they are based in: parent's occupation, children in day care service and family's health and social image.

Nursery has 12960 entries and 9 attributes

5 Results

Up next the process that leads to the generation of the association rules is detailed using the tree mentioned algorithms.

For each of the tests values of 30% and 80% were stablished as minimum support and minimum value respectively for the Titanic and Bank.

For the Nursery dataset the evaluation criteria was modified given the magnitude of the entries and the high numeric register. It was assigned a minimum support value of 10% and 60% for the minimum support.

5. 1 Rules generated for the Titanic dataset

Rule 1. if Class = 3rd ⟹Age = Adult

[28.48% / 627, 88.81%, 0.93]

Rule 2. if Survived = Yes ⟹Age = Adult

[29.71% / 654, 91.98%, 0.96]

Rule 3. If Class= Crew ʌ Survived = No ʌ Age = Adult ⟹ Sex = Male

[30.44% / 670, 99.55%, 1.26]

This set's mode is the "Adult" data, it can be noticed that is present in all tree rules, being consistent to the frequent dataset generated.

5.2 Rules generated for the Bank Dataset

Rule 4. if Education = secondary ∧ loan = no ∧ age=major ⟹ suscription = no

[33.66% / 15219, 90.20%, 1.02]

Rule 5. if Housing = yes ∧ marital = married ∧ subscription = no ⟹ age = major

[30.39% / 13743, 96.02%, 1.07]

Rule 6. If Marital = married ∧ duration = low ∧ loan=no ∧ age = major ∧ default = no ⟹

subscription = no

[31.50% / 14245, 95.69%, 1.08]

Based on the lift theory it is inferred that for the previous rules the frequency of the consequent influences in the occurrence of the antecedent, Example rule 5, Age = Higher influence in the Housing occurrence = yes ∧ marital = married ∧ subscription.

5.3 Rules generated for the Nursery Dataset

Rule 7. If Health = priority ^ finance = inconv⟹ diagnostic = spec_prior

[10.09% / 1308, 60.55%, 1.94065]

Rule 8. If Health = not_recom ^ finance = inconv ⟹ diagnostic = not_recom

[16.66% / 2160, 100%, 3]

Rule 9. Health = not_recom ^ finance = convenient ⟹ diagnostic = not_recom

[16.66% / 2160, 100%, 3]

The reason why the algorithms generate equal rules is because the base algorithm for each is the same; that is to say that the tree algorithms are based in the frequent items generation. This means that the algorithms have the same rate of effectiveness; the difference lies in the amount of time each algorithm takes to build the rules. Table 5 shows the time it takes for each algorithm as well as its algorithm complexity, where is noticeable that the algorithm (A priori algorithm) with the highest complexity is the one that takes the longest to generate the rules and the one with less complexity (Eclat algorithm) takes fewer time; for further illustration it is shown in figure 1.

Algorithm	Complexity	Runtime (milliseconds)		
		Titanic	Nursery	Bank
A priori	$O\left(C_{sum} \times \|T\|\right)$	2061	2033	2134

Fp-Growth	$O(Nt \times Dt)$	1817	1917	2034
Eclat	$O(k(k+1))$	1684	1778	1884

Table 5. Complexity and runtime. Source: own.

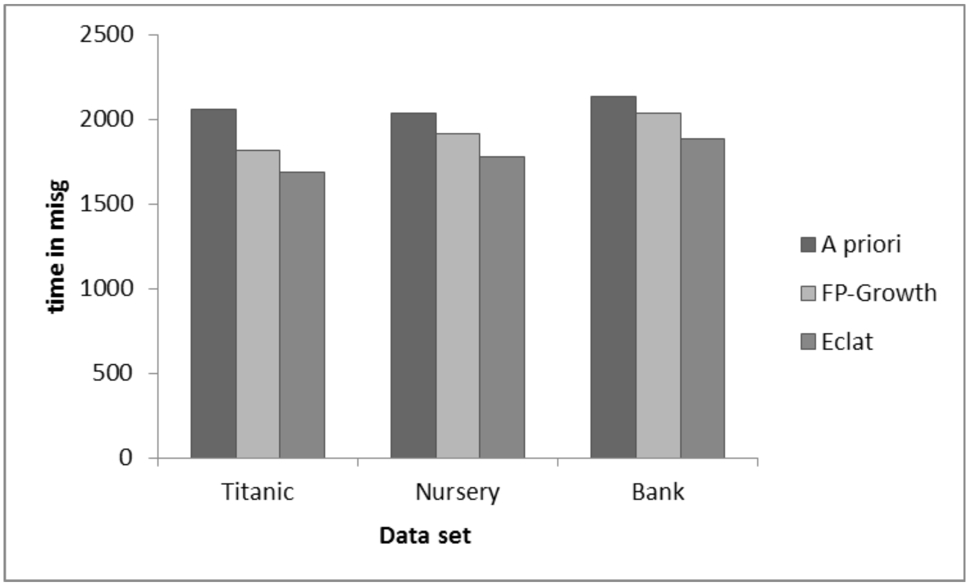

Figure **1. Runtime. Source: own.**

6 Conclusions

The obtained results proved the importance of this field of investigation, nevertheless, it is required a bigger study in order to optimize the methods currently used, for example, the data association as a data mining technique is an alternative that allows to build experience and contributes to the decision making process, optimizes the processes that currently exits and makes easier to extract the relations among data.

It was noticed within the induction process that the association rules is minor in the FP-Growth algorithm than on the A priori because the search of candidate itemsets that is performed in the FP-tree structure and not in the whole dataset, the FP-Growth uses the divide and rule strategy which allows it to be a faster and scalable; nevertheless computationally the Eclat algorithm is more efficient than the FP-Growth, even though it performs additional grouping tasks that requires additional steps within its functioning.

Its recommended to use the Eclat algorithm for the association rules generation, considering that the algorithm complexity and the execution time was less that the ones for the A priori and FP-Growth when performing the KDD process, demonstrated in the tests and results analysis in this article, also that the datasets have different behaviors and that the minimum support and minimum confidence values must stablished according to the features that these presents.

7 Future works

It is proposed to try other algorithms such as the Generec Fuzzy Apriority, whit the purpose of comparing the results generated in this article.

References

[1] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus. “Knowledge Discovery in Databases: An Overview,” *AI Magazine*, vol. 13, no. 3, pp. 57-70, 1992.

[2] Z. Tang and J. McLennan, “Data Mining with SQL Server 2005”, United States of America: Wiley, 2005.

[3] M. Kantardzic, “Data Mining: concepts, models, methods, and algorithms”, 2nd Ed. USA: Wiley-IEEE Press, 2011.

[4] J. Hernández, J. Ramírez and C. Ferri, “Introducción a la Minería de Datos,” in Introducción a la Mineria de Datos”, Madrid: Pearson, pp 239-243, 2004.

[5] A. A. Morales, M. L. Rodríguez and J. E. Rodríguez, “Prototipo de software para la asociación de datos ‘UDAssociate ’,” *Revista Científica*, vol. 7, pp. 567–588, 2005.

[6] K. Khurana and Mrs. S. Sharma, “A Comparative Analysis of Association Rules Mining Algorithms,” *International Journal of Scientific and Research Publications*, vol 3, Issue 5, May. 2013.

[7] M, Berry and G. Linoff, “Data Mining Techniques: For Marketing, Sales, and Customer Support”. USA: Wiley, Jun 1997.

[8] R. Agrawal, T. Imielinski, and A. Swami, “Mining Association Rules between Sets of Items in Large Databases,” ACM SIGMOD Conference Washington DC, USA, pp. 207–216, 1993.

[9] A. Y, González, J. F. Martínez, J. A. Carrasco, J. Ruiz, “Minería de Reglas de Asociación sobre Datos Mezclados,” Instituto Nacional de Astrofísica, Óptica y Electrónica, Mar 2009. [Online]. Available: http://ccc.inaoep.mx/portalfiles/file/CCC-09-001.pdf

[10] J. P. Lucas, “Métodos de clasificación basados en asociación aplicados a sistemas de recomendación,” Tesis Dr. Universidad Salamanca, 2010. [Online]. Available: http://gredos.usal.es/xmlui/bitstream/handle/10366/83342/DIA_PinhoLucasJ_M%C3%A9todosdeclasificaci%C3%B3n.pdf?sequence=1

[11] M. E. De Moya Amaris and J. E. Rodríguez, “La contribución de las reglas de asociación a la minería de datos,” *Tecnura*, vol. 7, no. 13, pp. 94–109, 2003.

[12] M.Castillo and N. Martín, “Reglas de asociación para líneas espectrales,” Universidad de Chile 2015. [Online]. Available: http://repositorio.uchile.cl/handle/2250/133022

[13] J. Han, J. Pei, Y. Yin, and R. Mao, “Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach,” *Data Min. Knowl. Discov*, vol. 8, no. 1, pp. 53–87, 2004.

[14] J. Han, J. Pei, and Y. Yin. “Mining frequent patterns without candidate generation,” In Weidong Chen, Jeffrey Naughton, and Philip A. Bernstein, editors, ACM SIGMOD Intl. Conference on Management of Data, ACM Press, pp. 1–12, May 2000.

[15] J. Panjwani, “APPLICATION OF FP TREE GROWTH IN TEXT MINING”, India, 2008. [Online]. Available: http://dspace.jdvu.ac.in/bitstream/123456789/20202/1/Acc.%20No.%20DC%20388.pdf

[16] M.J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, "New algorithms for fast discovery of association rules," in Third International Conference on Knowledge Discovery and Data Mining, AAAI ,1997. [Online]. Available: http://www.aaai.org/Papers/KDD/1997/KDD97-060.pdf

[17] R. Danger and R. Berlanga. Llavori, “Búsqueda de Reglas de Asociación en bases de datos y colecciones de textos,” Universitat Jaume, april, 2001. [Online]. Available: https://www.researchgate.net/publication/228849142_Busqueda_de_Reglas_de_Asociacion_en_bases_de_datos_y_colecciones_de_textos

[18] C. Borgelt, “Efficient implementations of apriori and eclat,” Department of Knowledge Processing and Language Engineering School of Computer Science, Otto-von-Guericke-University of Magdeburg Universitatsplatz, Magdeburg, Germany, 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.162.291&rep=rep1&type=pdf

[19] S.C Satapathy, “Information Systems Desing and Intelligent Applications”, *Advances in intelligent System and Computing,* vol 2, pp. 219-223, India, 2016.

[20] J. J. Aguilera, M. J. Del Jesus, P. González, F. Herrera, M. Navío, and J. Sáinz, “Extracción Evolutiva de Reglas de Asociación en un Servicio de Urgencias Psiquiátricas,” Keel, 2002. [Online]. Available: http://sci2s.ugr.es/keel/pdf/keel/congreso/agu03_maeb03.pdf