MEDINFO 2001
V. Patel et al. (Eds)
Amsterdam: IOS Press

Mining Association Rules from Clinical Databases: An Intelligent Diagnostic Process in Healthcare

S. Stilou[1], P.D. Bamidis[1,2], N. Maglaveras[1], C. Pappas[1]

[1]*Lab of Medical Informatics, The Medical School, Aristotelian University, Thessaloniki, Macedonia, Greece*
[2]*Dept. of Computer Science, CityLiberal Studies, Affiliated Institution of the University of Sheffield, Thessaloniki, Greece*

Abstract

Data mining is the process of discovering interesting knowledge, such as patterns, associations, changes, anomalies and significant structures, from large amounts of data stored in databases, data warehouses, or other information repositories. Mining Associations is one of the techniques involved in the process mentioned above and used in this paper. Association is the discovery of association relationships or correlations among a set of items. The algorithm that was implemented is a basic algorithm for mining association rules, known as a priori. In Healthcare, association rules are considered to be quite useful as they offer the possibility to conduct intelligent diagnosis and extract invaluable information and build important knowledge bases quickly and automatically. The problem of identifying new, unexpected and interesting patterns in medical databases in general, and diabetic data repositories in specific, is considered in this paper. We have applied the apriori algorithm to a database containing records of diabetic patients and attempted to extract association rules from the stored real parameters. The results indicate that the methodology followed may be of good value to the diagnostic procedure, especially when large data volumes are involved. The followed process and the implemented system offer an efficient and effective tool in the management of diabetes. Their clinical relevance and utility await the results of prospective clinical studies currently under investigation.

Keywords:

data mining, association rules, medical databases, apriori algorithm

Introduction

In the past few years, the collection of clinical data has become so huge, that unless we introduce new techniques and tools, accuracy may be severely affected. This explosive growth in data and databases has generated an urgent need for new techniques and tools that can intelligently and automatically transform the processed data into useful information and knowledge [1]. Data mining, which is also known as *knowledge discovery in databases*, is a process of nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. In general, data mining is an essential process in knowledge discovery where intelligent methods are applied in order to extract data patterns.

Clinical databases belong to a domain where the process of data mining has become a requirement because of the consecutive increase of medical and research clinical data. Data mining can be used as an intelligent diagnostic tool in Healthcare. In medical data, it is possible to extract knowledge and information about a disease from the patient specific stored measurements. In addition, in research data the extraction knowledge could be the information about an unknown virus that looks like some other kind of known viruses. Consequently, *data mining* has become an important research domain in Healthcare.

In this paper, we implement one of the basic algorithms for mining association rules, namely the *a priori* algorithm, and apply it in extracting knowledge from a clinical database with records from diabetic patients.

Materials and methods

Data mining is an application-dependent issue that copes with different mining techniques. One of these techniques is the mining of *association rules* from transactional or relational databases [2,3,4]. The task is to derive a set of strong association rules [5] in the form of "$A_1 \wedge \ldots \wedge A_m \Rightarrow B_1 \wedge \ldots \wedge B_n$", where A_i ($i \in \{1, m\}$) and B_j ($j \in \{1, n\}$) are sets of attribute-values from the relevant data sets in a database. For example, in a large set of transaction data, one may find an association rule like the following: "if a patient smokes a lot and is overweighed, he/she has more possibilities to get a stroke".

$I = \{i_1, i_2, \ldots, i_m\}$ is the set of items and D is a set of transactions where each transaction T is a set of items such that $T \subseteq I$. A transaction T contains X, a set of some items in I, if $X \subseteq T$. An association rule is of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \varnothing$. The rule $X \Rightarrow Y$ holds in the transaction set D with *confidence c*, if c% of

transactions in D that contain X also contain Y. The rule X $\Rightarrow$ Y has *support s*, in the transaction set D, if s% of transactions in D contain X $\cup$ Y [2].

If a set D exists, mining association rules is the problem of generating all association rules that have support which is greater than the user defined minimum support, and have confidence that is greater than the user defined minimum confidence.

Discovering association rules in [3] is split into two sub-tasks, that of finding all sets of items that have support above the user defined support, which are known as large item sets, and that of using these sets to generate the rules. An algorithm, known as Apriori [2, 3], which makes multiple passes over the data, achieves that. The Apriori algorithm used to mine association rules from the data is described below:

1) L_1 = {large 1-itemsets}
2) **for** ($k = 2$; $L_{k-1} \neq \emptyset$; k++) **do begin**
3) C_k = apriori-gen(L_{k-1})
// New candidates
4) **forall** transactions $t \in D$ **do begin**
5) C_t = subset (C_k , t)
//Candidates contained in t
6) **forall** candidates $c \in C_t$ **do**
7) c.count++
8) **end**
9) $L_k = \{c \in C_k \mid c.\text{count} \geq \text{minsup}\}$
10) **end**
11) Answer = $\cup_k L_k$

The first pass is used to determine the support for individual sets of items and to ensure that this support is above the user defined level, these sets are called large 1-itemsets, L_1 as in line 1. From this point the algorithm works with a number of iterations k (see line 2). Each subsequent pass uses the item sets discovered in the previous pass whose items have support greater than the user defined level. This set of item sets is used to generate new item sets called "candidate" item sets C_k. This is done using the apriori-gen candidate generation algorithm. The idea of the apriori-gen algorithm is to create all the supersets of the large k-item sets from all the (k-1)-item sets as described in line 3. The apriori-gen algorithm [3] works by generating the "candidate" item set by two steps, a *join step* and a *prune step*. The join phase is described in pseudo SQL is given below:

insert into C_k

select p.item$_1$, p.item$_2$, …., p.item$_{k-1}$, q.item$_{k-1}$

from L_{k-1} p, L_{k-1} q

where p.item$_1$ = q.item$_1$,…., p.item$_{k-2}$ = q.item$_{k-2}$, p.item$_{k-1}$ < q.item$_{k-1}$;

The join step simply joins the large sets to form a "pre-pruned candidate" item set. The effects of this join would be to create new k-item sets by joining (k-1)-item sets, such that each element of the two (k-1)-item sets being joined match except for the last member of each set. The last member of the first set being joined should be less than that of the last member of the second set being joined. After the join phase the candidate item sets C_k are generated. The second step is the prune step of the a priori-gen algorithm, which can be seen below:

forall itemsets $c \in Ck$ **do**
  forall (k-1)-subsets s of c **do**
    if ($s \notin L_{k-1}$) **then**
      delete c from Ck;

The prune step simply involves removing from the "pre-pruned candidate" item set those item sets that contain sub item sets that did not exist in the set of the previous pass. In other words the purpose is to delete all item sets $c \in C_k$ such that some (k-1)-subset of c is not in L_{k-1}, where L_{k-1} are the large item sets before the join.

The resultant candidate C_k is then used to generate a large k-item set for the next iteration of the algorithm as described in lines 4 to 9. This is done removing the item sets whose support is measured at less than that of the user-defined level. This process is continued until the "candidate" item set is an empty set and the generated large item sets are used for the rule generation process. This is achieved through the use of the user-defined confidence. For every large item set rules of the form $a \Rightarrow (m - a)$ are produced, where 'm' is the large item set and 'a' is some non empty subset of 'm'. The confidence of a rule is given by the ration of the support(m) to support(a). If this ratio falls below the user defined confidence then the rule is discarded. The remaining rules will all therefore have confidence greater than or equal to the user defined minimum confidence. These would be the rules that would be the output to the user.

In order to mine association rules the first requirement is to create some data that are used from the Apriori algorithm given in [3]. It is important to mention that the nature of medical data is categorical, quantitative or Boolean. Data mining with association rules as described in [3] are concerned only with the mining of Boolean data. Therefore, it was necessary to transform categorical and quantitative data to Boolean and then apply the algorithm discover association rules. In essence the transformation is achieved by creating additional attributes from existing categorical or quantitative data through partitioning.

In the case of categorical data, a new partition is created for each unique occurrence of an attribute, so for example the terminology of the diseases, which is a categorical attribute could be partitioned into attributes representing each occurrence of a major disease. Thus, if we have the following data set of the field "Disease", the result of the transformation to Boolean formats, or coding process would be:

Patient 1 - Stomach Cancer ⇒ 1001

Patient 2 - Stroke ⇒ 1002

Patient 3 - Asthma ⇒ 1003

Patient 4 - Stomach Cancer ⇒ 1001

Where "001" indicates the 1st category of the diseases and "1" the 1st field of the specific table.

Quantitative attributes are concerned partitions are made according to ranges of values over the attribute. The user declares the number of ranges and the ranges itself. For instance, the quantitative attribute "weight" has the following data set, the number of ranges is declared "3", the 1st range "65-80", the 2nd "80-95" and the 3rd "95-110". Consequently, the code that is generated is the following:

Patient 1 - 80 ⇒ 2002

Patient 2 - 67 ⇒ 2001

Patient 3 - 70 ⇒ 2001

Patient 4 - 100 ⇒ 2003

After the transformation the Apriori algorithm can be executed to this kind of data. The generated rules are represented in a coded Boolean format like the previous examples and are of the form: 3001 2003 → 1002. Thus, a decoding process was implemented so that the rules were readable. Therefore, the coding of the field was necessary because we had to know with which field we cope with, to generate the decoded rules, which are of the form: IF Status = "smokes a lot" AND Weight = "95-110" THEN Disease = "Stroke", with confidence *c* and support *s*.

The Apriori algorithm was implemented in C++, but the object-oriented aspect was nevertheless neglected, due to the nature of the algorithm and its associated structures. The data-mining tool was developed in Delphi 4. The user can choose a database from a list, and then table and fields from which the rules will be generated. The remaining steps are data coding, the mining process itself, and finally the decoding of rules.

Results

An example of the mining process was implemented in a medical database with stored real parameters of diabetic patients, which were measured from an expert, consisting of a table with 100 patients. Its structure can be seen in Table 1, and its explanation about the fields and the values in the spreadsheet in Table 2.

We implemented the Apriori algorithm several times to this dataset or to a sub-dataset of the table, and got each time a list of associations between the parameters (association rules). In one particular example we chose from the table the fields: Case_No (patients), DMType (diabetes mellitus type), Age (age in years), Special (special condition like pregnancy, surgery, infection), Previous_Rx (patient's previous regime), Target (desirable diabetes control), Dawn (dawn phenomenon), Unstable (unstable diabetes) and regime (regime proposed by expert). Only the field "Age" is continuous, all the other fields are categorical. The field "Age" has minimum record value 20 and maximum 90, so we declared 7 ranges that are the following: 20-30, 31-40, 41-50, 51-60, 61-70, 71-80, 81-90. After the coding phase of categorical and continuous data, the algorithm was executed with confidence 50% and support 10%, in a data set consisting the parameters of all 100 patients. The rules that were mined are represented in the form in figure 1, where the resultant coded and decoded rules from this data set are shown. One rule that is generated is: " IF *diabetes mellitus type* = 2 AND *special* condition = no AND *target* (desirable diabetes control) = good AND *unstable diabetes* = no THEN *regime* (proposed injection per day) = 2 (which means 2 injection of mix insulin sort and intermediate action, one in the breakfast and one in the afternoon)".

Because of the small confidence (50%) the rules that are generated are not so strong. The support parameter is also small (10%), because of the small number of categories that were generated. The specific data set creates 2 or at most 3 categories for each field. If the confidence and the support were greater, then the generated rules would be fewer or even none.

Table 1 Sample of diabetes parameters

Cases no	DM type	age	Special	Previous Rx	Target	Dawn	Unstable	BG-bre	BG-lun	BG-din	BG-bed	PA-mor	PA-aft	PA-nig	FI-bre	FI-lun	FI-din	FI-bed	Regime
1	1	20	y	i	vg	n	n	hi	hi	hi	hi	s	-	-	y	y	y	y	4
2	1	40	y	i	vg	n	n	hi	nl	nl	nl	s	-	-	y	y	y	n	4
3	1	20	n	i	vg	n	n	nl	nl	nl	nl	h	-	-	y	y	y	y	3
4	1	35	n	i	vg	n	n	nl	hi	nl	nl	h	h	-	y	y	y	y	3
5	1	30	n	i	vg	n	n	nl	nl	hi	nl	L	l	-	y	n	y	n	3

Table 2 Explanation of Table 1 fields.

parameter	values	explanation
DM type	1	diabetes mellitus, type
	2	
age	number	age in years
special	yes	special condition : pregnancy, surgery, infection
	no	
previous Rx	insulin	patient's previous regime
	tablets	
target	fair	desirable diabetes control
	good	
	very good	
dawn	yes	dawn phenomenon
	no	
unstable	yes	unstable diabetes
	no	
BG-bre	normal (nl)	blood glucose - breakfast
	hyperglycaemia (hi)	
	hypoglycaemia (lo)	

BG-lun	normal (nl)	blood glucose – lunch
	hyperglycaemia (hi)	
	hypoglycaemia (lo)	
BG-din	normal (nl)	blood glucose – dinner
	hyperglycaemia (hi)	
	hypoglycaemia (lo)	
BG-bed	normal (nl)	blood glucose – bed
	hyperglycaemia (hi)	
	hypoglycaemia (lo)	
PA-mor	sedentary	physical activity – morning
	light	
	heavy	
PA-aft	sedentary	physical activity – afternoon
	light	
	heavy	
PA-nig	sedentary	physical activity – night
	light	
	heavy	
FI-bre	yes	food intake – breakfast
	no	
FI-lun	yes	food intake – lunch
	no	
FI-din	yes	food intake – dinner
	no	
FI-bed	yes	food intake – bed
	no	
Regime	1	Regime, proposed by expert, Bre(I*)
	2	Bre, Aft (S**+I)
	3	Bre, Lun, Aft, (S); Bed (I)
	4	Bre, Lun, Aft, Bed (S)
		**I=intermediate action*
		***S=short-action*

An alternative example of query is the following: we chose the fields DMType, special, dawn, unstable, BG_bre (blood glucose - breakfast), BG_lun (blood glucose - lunch), BG_din (blood glucose - dinner), BG_bed (blood glucose - bed) and regime. All the mentioned fields are categorical. The algorithm was executed in a data set of the first 50 patients, with confidence 80 and support 20. The rules that are generated are represented in figure 2.

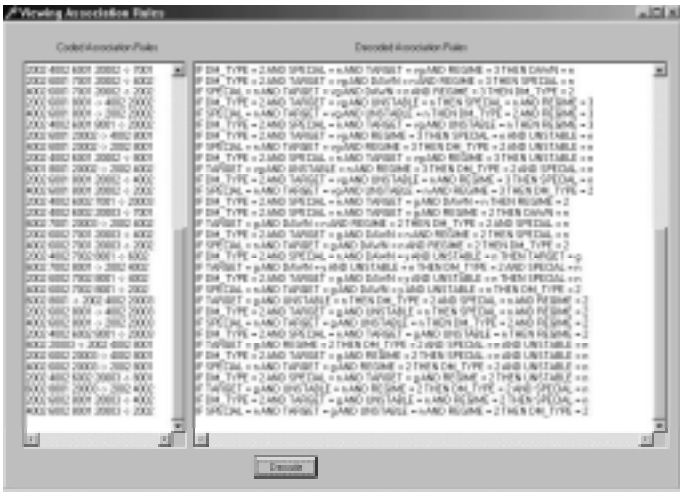

Figure 1 – Form with Coded and Decoded Association Rules / Example 1

Figure 2 – Form with Coded and Decoded Association Rules / Example 2

Discussions and Conclusion

The extraction of some kind of knowledge from medical databases is a challenging and useful technique that offers assistance unavailable before the mining process [6].

The accuracy of the extracted knowledge is an important point that must be considered, but this has to do with the validity of the data and their relevant transformations.

In this paper, the mining process was implemented with categorical, quantitative and Boolean medical data, but a feature perspective is to deal with multimedia data such as images in general (microbiology), digital X-ray images, tumor scans, or signals. Another point that is currently under consideration is the improvement of the algorithm performance.

Acknowledgments

This work has been partly supported by a grant from the Greek Ministry of Education, EPEAEK – PROMESIP.

Advice and suggestions from postgraduate students I.Chouvarda and G.Gogou are gratefully acknowledged.

References

[1] U. M. Fayyad, G.Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.

[2] M. Chen, J. Han, and P. S. Yu. Data Mining: An Overview from Database Perspective. IEEE Trans. Knowl. Dat. Eng., 8(6): 866-883, 1996.

[3] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases", In Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, Chile, August 29-September 1994.

[4] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207-216, Washington, DC, May 26-28 1993.

[5] G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In G. Piatetsky-Shapiro and W. J. Frawley, editors, Knowledge Discovery in Databases, pages 229-238. AAAI/MIT Press, 1991.

[6] S. Brossette, A. P. Sprague, J. M. Hardin, K. B. Waites, W. T. Jones, and S. A. Moser: Association rules and Data Mining in Hospital Infection Control and Public Health Surveillance. JAMIA, 5: 373-381, 1998.

Address for correspondence

Professor Costas Pappas, Aristotelian University, The Medical School, Lab of Medical Informatics – Box 323, 54006, Thessaloniki, Macedonia, GREECE, EMAIL: cpappas@med.auth.gr