USING DATA MINING TECHNIQUES FOR DECISION SUPPORT SYSTEMS

Mirela DANUBIANU

"Stefan cel Mare" University of Suceava
9 University Street, RO-5800, Suceava
mdanub@eed.usv.ro

Abstract. In the following we briefly introduce the basic idea about the relationship between knowledge discovery in databases and management. We point out the implications of understanding KDD as a nontrivial and interactive process and we focus to the data mining step for the association rules generation. Further we consider the real case of a utility supplier company, and we analyze the association rules generated from the data of the marketing department.

Keywords: data mining,, KDD process, CRISP-DM model, associations rules, DSS

1. Introduction

Over the past years, computers have been used to capture details of business transactions such as banking and credit card records, retail sales, manufacturing warranty, telecommunications, utilities, etc. These dates have thumb prints of the key trends that impact various aspects of each business like products that sell together, sources of profits, factors that affect manufacturing quality, etc. For this reason the field of knowledge discovery in databases and data mining, as one of this steps, become a fundamental research area with important applications.
Many organizations view information as one of their most valuable assets and knowledge discovery in databases allows a company to make full use of these information assets.

2. The relationship between data mining and DSS

Many companies realize that to succeed in a fast pace world, their managers need to be able to get information on demand. But there is never enough time to think of all the important questions. In these conditions the computer should do this by itself. It can provide the winning edge in business by exploring the database itself and brings back invaluable nuggets of information.
They also need to be pleasantly surprised by unexpected, but useful, information.
Decision Support is a broad term referring to the use of information as a strategic corporate asset, enabling companies to utilize their databases to make better decisions. Decision support systems, that are indispensable tools for the managers, have traditionally relied on three types of analyses: Query and Reporting: where a user asks a question, OLAP (On Line Analytical Processing): which amounts to the processing of queries along multiple dimensions such as state, month, etc., and Data Mining: which provides influence factors and relationships in data.

3. Data Mining and the KDD Process

The field of data mining and knowledge discovery is emerging as a fundamental research area with important applications to science, engineering, medicine, business, and education. Data mining attempts to formulate, analyze and implement basic induction processes that facilitate the extraction of meaningful information and knowledge from unstructured data. Data mining extracts patterns, changes, associations and anomalies from large data sets. Work in data mining ranges from theoretical work on the principles of learning and mathematical representations of data to building advanced engineering systems that perform information filtering on the web, help understand trends and anomalies in economics and education, and detect network intrusion. Researchers from many intellectual communities have much to contribute to this field. These include the communities of machine learning, statistics, databases, visualization and graphics, optimization, computational mathematics, and the theory of algorithms.
Data mining is the semi-automatic discovery of patterns, associations, changes, anomalies, rules, and statistically significant structures and events in data
The experience of the last years showed that discovering knowledge from huge databases involve much more than simply applying a sophisticated data mining algorithm to a predefined dataset.
One of the most important problems in KDD research is the understanding of KDD as a "nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.**"**[3]

Of this point of view, pattern is meant in a very general way. A pattern is whatever a data mining algorithm may find or generate from the data, like a model than scores customers based on a decision tree or based on a neural network, a clustering of the data, or a set of association rules.

Although are several different between process models [2,3,4], the key message is the same: data mining that is applying a sophisticated mining algorithm to a dataset, is just one of several steps in a KDD process.

Corresponding to the CRISP-DM model [6] , we distinguish the following six tasks:

- ✓ **Business (or Problem) Understanding**. This phase focuses on understanding the project objectives and requirements from a business perspective, and developing initial technical problem definition and a project plan.
- ✓ **Data Understanding** Based on the results from the business point of view the second step is to get familiar with the available data.
- ✓ **Data Preparation.** The next step is to construct the dataset where the mining algorithm is to be run on.
- ✓ **Modeling** Various modeling techniques are selected, applied, and fine-tuned. In this phase the actual data mining takes place. Based on the identified business goal and the assessment of the available data an appropriate mining algorithm is chosen and run on the prepared data.
- ✓ **Evaluation** At this stage there are good models (from a technical point of view). Here we thoroughly evaluate the model, and review the steps executed to construct the model, to check if we did not miss an important business issue and achieves the desired business objectives
- ✓ **Deployment**.

The analysis of these phases show us that the KDD process no means only a push button technology. On the contrary, knowledge discovery is complex, iterative and highly interactive.

Figure 1 presents these phases and the most important interdepencies between them.

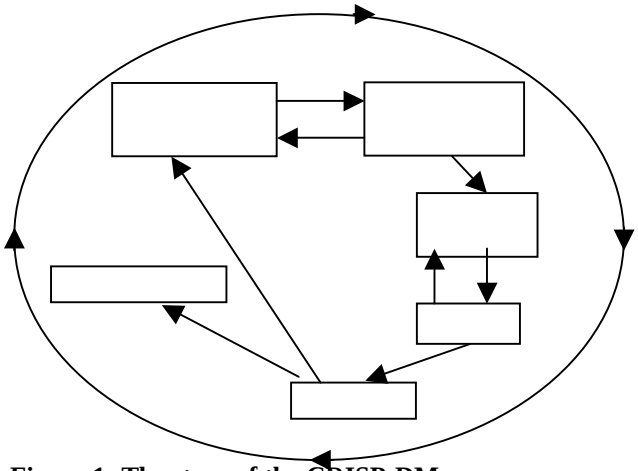

Figure 1: The steps of the CRISP-DM process

4. Association Rules

Association rules model dependencies between items in transactional data.

Let $I = \{x_1 ,.. x_n\}$ be a set of distinct literals, called items. A set $X \subseteq I$ with $k = |X|$ is called a k-itemset or simply an itemset. Let a database D be a multi-set of subsets of I. Each $T \in D$ is called a transaction.

A transaction $T \in D$ supports an itemset $X \subseteq I$ if $X \subseteq T$.

Let $X; Y \subseteq I$ be nonempty itemsets with $X \cap Y = \emptyset$. Then an association rule is an expression :

$$X \rightarrow Y$$

with assumption X, consequent Y , and rule confidence.

$$conf(X \rightarrow Y) = \frac{|\{T \in D \mid X \cup Y \subseteq T\}|}{|\{T \in D \mid X \subseteq T\}|}$$

The confidence can be understood as the conditional probability P(Y|X).

The fraction of transactions T supporting an itemset X with respect to database D is called the support of X

An itemset reaching a predefined threshold for support is called a frequent itemset. The support of a rule $X \rightarrow Y$ is defined as:

$$supp(X \rightarrow Y) = supp(X \cup Y)$$

In practice the support-confidence framework, as described above, shows severe limitations. The reason is that association rules are based on correlations and even for large confidence values do not necessarily imply causation. As consequence supplementary rule quality measures have been developed over the years. One of these measures is lift (interest) [4]:

The measure lift expresses the deviation of the rule confidence from the a priori probability of Y, supp(Y)

$$lift(X \rightarrow Y) = \frac{conf(X \rightarrow Y)}{supp(Y)}$$

That is, in how far does the rule body X “lift" the probability for the rule head Y to occur in the same transaction.

5. Rules generation

Since the introduction of association rules, a broad variety of associations rule mining algorithms have been developed, but the main challenge when mining association rules is the great number of rules to be considered.

The rule set to be generated is typically restricted by minimal thresholds for the rule quality measures support and confidence, called minsupp and minconf respectively. This restriction allows to split the problem into two separate parts: first we must find the set of all frequent itemsets , second, for every frequent itemset on must check the confidence.

An itemset X is called frequent if **supp(X) ≥ minsup**, and

$$F = \{X \subset I \mid \text{supp}(X) \geq \text{minsupp}\}$$

is the set of all frequent itemsets.

F has the so called downward closure property of itemset support, that states that all subsets of a frequent itemset must be also frequent. So before rule generation on must determine F, the set of all frequent itemsets. Unfortunately we are not able to

look at all subsets of *I*, because a linearly growing of number of items
for an itemset still implies an exponential growing number of subsets to be taken into consideration.
The modern association mining algorithms employ a candidate generation and test approach. The idea is to generate an easy to survey set of potential frequent itemsets, called candidates. The support values of these candidates are determined based on the database *D*. The process of candidate generation considers all information on frequency of already investigated candidates. In brief, the procedure is: from the downward closure property of itemset support we know that all subsets of a frequent itemset must be also frequent. This allows us to prune those candidates as infrequent from the search space that have at least one infrequent subset. After candidate generation the designed candidates are counted based on the database and on proceed to the next iteration. The whole process stops as soon as there are no more potentially frequent itemsets that have not be considered as candidates.
The framework for frequent itemset generation allows several algorithms. The most wide-spread of these algorithms is Apriori algorithm.[1] A variant of this algorithm is AprioriTid [1]. The Apriori and AprioriTid algorithms generate the candidate itemsets to be counted in a pass by using only the itemsets found frequent in the previous pass-without considering the transactions in the database. The basic intuition is that any subset of a large itemset must be large. Therefore, the candidate itemsets having k items can be generated by joining frequent itemsets having (k-1) items, and deleting those that contain any subset that is not large. This procedure results in generation of a much smaller number of candidate itemsets. The AprioriTid algorithm has the additional property that the database is not used at all for counting the support of candidate itemsets after the first pass. Rather, an encoding of the candidate itemsets used in the previous pass is employed for this purpose.
In later phases, the size of this encoding can become much smaller than the database, thus saving much reading effort.
In the follow we present the Apriori algorithm. We use the notation below:
D- database of transaction
t- tuple of *D*
k-itemset-an itemset having k items.
L_k- set of frequent k-itemsets (those with minimum support).Each member of this set has two fields:
itemset and support count.
C_k- set of candidate k-itemsets (potentially frequent itemsets).Each member of this set has two fields:
itemset and support count.
C'_k- set of candidate k-itemsets when the TIDs of the generating transactions are kept associated with the candidates.

L_1 = [frequent 1-itemsets];
while $L_{k-1} \neq \Phi$ **do begin**
C_k=gen_apriori(L_{k-1})/New candidates
for $\forall$ t$\in$ D **do begin**
C_t={C_k|$C_k \subset$t)/Candidates contained in t
for $\forall$ c $\in$ C_t **do**
c.count=c.count+1
end
Lk = {c $\in$ Ck | c.count $\geq$ minsup}
k=k+1
end
The **gen-apriori** function takes as argument L_{k-1}, the set of all frequent (k-1)-itemsets. It returns a superset of the set of all frequent k-itemsets. The function works as follows.
- first, in the join step, we join L_{k-1} with L_{k-1}:
insert into C_k
select p.$item_1$,..,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.ite**m $_{k-1}$;**

- in the prune step, we delete all itemsets c$\in$ Ck such that some (k-1)-subset of c is not in L_{k-1}
for $\forall$ itemsets c $\in$ Ck **do**
for$\forall$(k-1-subsets s of c **do**
if (s $\notin$ L_{k-1}) **then**
delete c from Ck **;**

6. Integration with Relational Database Systems

In the real-world applications the data reside in database systems. But the mining algorithms work with the binary encoded datasets, so, one of the key features is a proper integration with relational database systems.
The natural way to store a transaction in a relational table is in (id, item)-tuple form, so each transaction is represented by one or more rows in the table.
The association rule algorithms expect the data prepared as transactions. Generating such transactions from a database may require joining information pieces from all over the database, and sorting on the identification field, that we ensure that each transaction set is described by consecutive rows. Whenever the value in the identification field changes we know that a new transaction starts.

7. Case Study

We consider a database application for an utility supplier, and we want to analyze the marketing activity. The database contain fewer tables, and, as result of the Data Understanding step, for our purpose we use the tables “Contr”(who contains cca. 10.000 records) and “Tipuri” (50 records, each record point to one of the possibile services).
We search the association rules between the contracted services by the subscribers.
For solving this problem we must first prepare the data. So, we must clean the data and we must transform the table for a transaction table generation. Follow this step result a new database- a transaction database- with 3000 records. Then, we use the Apriori algorithm for association rules generation. The next step consist in

check of the confidence of these association rules, and finally we evaluate the results.
In the next rows we present, the phases of this approach.
Let D be common database.
*) **do** clean_db
*) **do** gen_dbt
*) **do** gen_fr_itemset
*) do find_rules

procedure clean_db(D) **do**
***)** prune the records and the fields irrelevant for our purpose
end
Procedure gen_dbt (D) **do**
n=nr_max_serv_contr
*) **do** gen_str_dbt
*) **do** fill_dbt
end
function nr_max_serv_contr (D)
*) count Si / the records for each subscriber
return max $_i(S_i)$
end
procedure gen_str_dbt **do**
*) create the table structure / only the name and the identification field
for i=1,n **do**
*) alter table - add new column
End
Procedure fill_dbt (D,*D*) **do**
*) dimension t[1,n]
*) open D
*) open *D*
id=""
while .not. EOF(D) **do**
read id
id1=id
*) append blank record in *D*
*) repl id withid1 in *D*
while id=id_1 **do**
t[1,i]= tip
i=i+1
*) skip in D
end
*) gather from t into *D*
*) reset t
end
end

Procedure gen_fr_itemset implement the Apriori algorithm, mentioned above, for the frequent k-itemset generation.
Procedure find_rules (L_k)
for i=2,k do
for j=1,i-1 do
$X_{i-1}=\{x_j|x_j\in L_i\}$
$Y=\{y|y\in L_i \wedge y\neq x_j\}$
$conf(X_{i-1}\rightarrow Y)=supp(X_{i-1}\cup Y)/suppX_{i-1}$
$Rij=\{X_{i-1}\rightarrow Y|conf(X_{i-1}\rightarrow Y)>minconf\}$
end
end

For the better results we are executed the aplication using different values for the minsup and minconf.
Finally, the association rules are interpreted and evaluated.
The association rules that are found show that there is not a adequate relationship between the services, that normally should be corelated (e.g. the cold water supply and the sewerage service are not relatted by an association rule).
In this real case we have discovered seriously deficiencies in the analyzed activity, so there is a field for management actions.

8. Conclusions

This paper has focus to the one of the aspects of the relationship between the business management and the knowledge discovery in databases process.
The real life shows that now it is impossible to succeed without the opportunities offered by the information technology.
We have considered association rules, viewed like the result of data mining - step in knowledge discovery in databases process- and we have applied the Apriori algorithm to find the association rules between the services supplied by a utilities company.
The analyze of results have discovered more deficiences in the marketing activity.

References

[1] Agrawal, R. and Srikant, R. (1994) *Fast algorithms for mining association rules*. In Proceedings of the 20th International Conference on Very Large Databases (VLDB '94), Santiago, Chile, June .
[2] Brachman R. J., and Anand. T.(1996) *The process of knowledge discovery in databases: A human centered approach.* In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, chapter 2, pages 37-57. AAAI/MIT Press,
[3] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996)*The KDD process for extracting useful knowledge from volumes of data*. Communications of the ACM, 39(11):27-34
[4] Hipp, J., Untzer, U. G., and Grimmer, U.(2001). *Integrating association rule mining algorithms with relational database systems.* In Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS 2001),pages 130{137, Set_ ubal, Portugal, July 7-10
[5] Pentiuc., Ghe-St, MORARIU, N. Morariu, M, Pentiuc. L (2001) *Intelligent System For Impact Prognosis Of The Economic Decisions At District Level,* Advances in Electrical and Computer Engineering ISSN 1582-7445 - Volume 1(18), Number 1(15), 2001
[6] Wirth, R. and Hipp, (2000) J. *CRISP-DM: Towards a standard process model for data mining.* In Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining, pages 29-39, Manchester, UK.