A New Algorithm for Mining Frequent Itemsets from Evidential Databases

Mohamed Anis Bach Tobji
LARODEC Laboratory
ISG of Tunis - Tunisia
anis.bach@isg.rnu.tn

Boutheina Ben Yaghlane
LARODEC Laboratory
IHEC Carthage - Tunisia
boutheina.yaghlane@ihec.rnu.tn

Khaled Mellouli
LARODEC Laboratory
IHEC Carthage - Tunisia
khaled.mellouli@ihec.rnu.tn

Abstract

Association rule mining (ARM) problem has been extensively tackled in the context of perfect data. However, real applications showed that data are often imperfect (incomplete and/or uncertain) which leads to the need of ARM algorithms that process imperfect databases. In this paper we propose a new algorithm for mining frequent itemsets from evidential databases. We introduce a new structure called *RidLists* that is the vertical representation of the evidential database. Our structure is adapted to itemsets belief computation which makes the mining algorithm more efficient. Experimental results showed that our proposed algorithm is efficient in comparison with the only evidential ARM algorithm in the literature [10].

Keywords: Association Rule, Evidential Database, Dempster-Shafer Theory.

1 Introduction

Since its emergence, the field of knowledge discovery from databases (KDD) has been especially focused on perfect data, assuming that data are complete and certain. Association rule mining (ARM) being a subfield of KDD, most of ARM algorithms were constructed on the assumption that data are perfect ([2], [9] and [12]). Nevertheless, real applications showed that data are often incomplete and uncertain. That is why recently, some works focused on the problem of ARM from uncertain databases. In most of such works, mined data are *probabilistic* ([11] and [5]), *possibilistic* [8] or *fuzzy* [4]. In spite of *evidence theory* [13] importance, there is a lack of works on ARM from evidential databases. *Evidential databases* allow storage of uncertain data where each attribute could have a basic belief assignment. Evidence theory is a generalized theory for modelling data uncertainty. Thus, an algorithm that mines association rules from evidential data, could also mine association rules from probabilistic and possibilistic data. In the literature, only the work of [10] tackled ARM problem from evidential databases.

In this paper, we present a new algorithm for mining frequent itemsets from evidential databases. Our proposed algorithm and the algorithm of [10] mine exactly the same association rules since the two methods are exact and are based on the same uncertain ARM model, i.e., the evidential one. The algorithm of [10] used a data structure, called *belief itemset tree* (BIT) to efficiently extract frequent itemsets and generate association rules from evidential databases. On the other hand, our proposed algorithm is based on the *record identifier lists* (RidLists) data structure. Our structure is a vertical representation [15] of the evidential database that accelerates support itemsets counting. We led experimentations on algorithms performances and results

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 1535–1542
Torremolinos (Málaga), June 22–27, 2008

showed that our algorithm is more efficient especially in the case of ARM from sparse data.

The paper is organized as follows: Section 2 introduces the evidence theory. Section 3 presents the evidential databases. In section 4, we recall the problem of ARM in perfect databases, and then we present frequent evidential itemset mining in Section 5. Section 6 shows experimentation results and discusses some observations. Finally, in Section 7, we conclude our work and present some perspectives.

2 Evidence Theory

2.1 Formalism

Evidence theory, also called Dempster-Shafer (DS) theory or belief functions theory, was introduced in ([6] and [7]). It was mathematically formalized in [13]. DS theory is often described as a generalization of the Bayesian theory since it manipulates events that are not necessarily exclusive. We present here formal concepts of this theory.

Let $\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$ be a finite non empty set of all elementary exhaustive and mutually exclusive events related to a given problem. Θ is called *frame of discernment* of the given problem.

The *basic belief assignment* (bba) is defined on the set of all subsets of Θ, namely 2^Θ. The bba m is the function $m : 2^\Theta \rightarrow [0, 1]$ that satisfies: $m(\emptyset) = 0$ and $\sum_{X \subseteq \Theta} m(X) = 1$

The *mass function* (m) allows someone to affect a partial belief value to a subset of Θ. Thus, $m(X)$ represents belief value placed exactly in the subset X and non distributed to subsets of X. Subsets of Θ with masses strictly positive are called *focal elements* of the bba m, focal elements set is denoted by F. The triplet $\{\Theta, F, m\}$ is called a *body of evidence* and denoted by BoE.

The *belief function* (*bel*) is defined and computed from the bba function m. X being an event, $bel(X)$ reflects total belief committed to X, i.e., total mass for all subsets of X.

$$bel(X) = \sum_{Y \subseteq X} m(Y)$$

The *plausibility function* (*pl*) quantifies amount of belief that could be given to a subset X of Θ. It is the sum of all masses of subsets Y that are compatible with X.

$$pl(X) = \sum_{Y \cap X \neq \emptyset} m(Y)$$

2.2 Conjunctive Rule of Combination

Let m_1 and m_2 be two bba's defined on the same frame of discernment Θ and provided by two 'independent' BoE's. The conjunctive rule of combination [14] is applicable when both sources of information (of combined bba's) are fully reliable. Conjunctive rule of combination is naturally applicable to more than two bba's.

$$m_1 \textcircled{\cap} m_2(Z) = \sum_{X,Y \subseteq \Theta : X \cap Y = \emptyset} m_1(X) \times m_2(Y)$$

3 Evidential Database

3.1 Definition

An evidential database, also called DS database or belief database stores data that could be perfect or imperfect. It allows users to set null (missing) values and also uncertain values. Uncertainty in such database is expressed via evidence theory presented in section 2. An evidential database is defined as follows:

It is a database denoted by EDB with n columns and d lines. Each column i ($1 \leq i \leq n$) has a domain D_i of discrete values. Cell of line j and column i contains an *evidential value* V_{ij} which is a bba defined as follows:

Definition 1 (Evidential value) *Let V_{ij} be an evidential value of column i and line j. V_{ij} is a BoE defined by a frame of discernment D_i, a set of focal elements F and mass function m_{ij} defined as follows:*

$$m_{ij} : 2^{D_i} \rightarrow [0, 1] \text{ with:}$$

$$m_{ij}(\emptyset) = 0 \text{ and } \sum_{x \subseteq D_i} m_{ij}(x) = 1$$

3.2 Data Imperfections

Evidential databases process different kinds of data imperfections thanks to evidence theory. Indeed, such theory allows us to represent *perfect information* when evidential value *BoE* includes only one focal element that is singleton with mass equal to one. For example, in the following evidential database sample (table 1), B_1 is a perfect value in column B in lines 1 and 2.

Table 1: Evidential database example

id	A	B	C
1	$A_1(0.6)$ $A_2(0.4)$	B_1	$C_2(0.2)$ $\{C1, C2\}(0.8)$
2	$A_1(0.2)$ $\{A_2, A_3\}(0.8)$	B_1	$C_1(0.5)$ $C_2(0.5)$

Probabilistic information is represented by evidential value with several focal elements that are singleton. In our database example, values of column C in the second line are probabilistic.

Possibilistic information can be also represented since possibility distribution could be converted into valid *BoE*. In our example, values of column C in first line are possibilistic with $\pi(C_1) = 0.8$ and $\pi(C_2) = 1$ (we recall that function π in possibility theory corresponds to *pl* in evidence one).

Missing information corresponds to a *BoE* with only one focal element that includes all column domain D_i values with mass equal to one. For example, if the value of column B is missing for one line, then Evidential value will be composed of only one focal element that is D_B with $m(D_B) = 1$.

Finally any *evidential information* could be represented. In our example, the value of A in the second line is an evidential information, that is neither perfect, nor probabilistic, nor possibilistic nor missing one.

In the next section, we recall briefly ARM problem in perfect databases before presenting the same problem in the context of imperfect databases.

4 Association Rule Mining from Perfect Databases

ARM problem has been introduced in [1] and has received a lot of attention thanks to its applicability in several fields. ARM problem is defined as follows:

Let I=$\{i_1$,i_2,...,$i_n\}$ be a set of n items. Let DB be a perfect database of D transactions with scheme $< tid, items >$. Each transaction is identified by a transaction identifier *tid* and is included in I (*items*). An association rule is $X \rightarrow Y$ with $X, Y \subseteq I$, $X \neq \emptyset$ and $X \cap Y = \emptyset$. *Support* of the association rule $X \rightarrow Y$ is the occurrence number of $Z = X \cup Y$ in DB denoted by *support*(Z). Its *confidence* is the ratio *support*(Z)/*support*(X). Given a support threshold min_{sup} and a confidence threshold min_{conf}, ARM problem consists in computing association rules with supports exceeding min_{sup}% and confidences exceeding min_{conf}%. An itemset is a set of items, it is said to be *frequent* in DB if its support exceeds min_{sup}%. ARM problem is divided into two subproblems:

1. Frequent itemsets generation.
2. Association rules computation from frequent itemsets.

The whole of the association rule problem is often reduced to frequent itemset mining, because once frequent itemsets are generated, association rules computation becomes a straightforward problem that is less costly comparing to the first subproblem [2]. In our work we focus only on frequent itemsets mining step.

5 Frequent Itemsets Mining from Evidential Databases

This section is an adaptation of the model of ARM from perfect databases to imperfect ones. Item and itemset notions are modified to support evidential values, support notion is also modified to take into account masses placed on evidential information which produces more accurate association. The preliminaries of frequent evidential itemsets mining model are the following:

5.1 Preliminaries

Definition 2 (Evidential Item) *An evidential item denoted by* iv_i *is one focal element in a body of evidence* V_{ij} *corresponding to column i. Thus it is defined as a subset of* D_i *(*$iv_i \in 2^{D_i}$*)*

Example 1 *In our database example (Table 1),* A_1 *is an item,* $\{A_2, A_3\}$ *too.*

Definition 3 (Evidential Itemset) *An evidential itemset is a set of evidential items that correspond to different columns domains. Formally, evidential itemset X is defined as:*

$$X \in \prod_{i \in \{1,.,n\}} 2^{D_i}$$

Example 2 *The itemset* A_1A_3 *is not a valid evidential itemset because the two items correspond to the same column A. The evidential itemset* $A_1\{C_1, C_2\}$ *is a valid one.*

We also define inclusion operator for evidential itemsets.

Definition 4 (Evidential Itemset Inclusion) *Let X and Y be two evidential itemsets. The* i^{th} *items of X and Y are respectively denoted by* i_x *and* i_y*.*

$$X \subseteq Y \text{ if and only if: } \forall i_X \in X, i_X \subseteq i_Y$$

Example 3 *The itemset* $\{A_1, A_2, A_3\}C_2$ *includes itemset* $\{A_1, A_2\}C_2$*.*

Now, we define the *line body of evidence* thanks to conjunctive rule of combination. A line body of evidence is computed from evidential values composing the line:

Definition 5 (Line BoE) *The frame of discernment of a line* BoE *is the cross product of all columns domains denoted by* $\Theta = \prod_{i \leq n} D_i$*. Focal elements are subsets of* Θ*, and thus vectors of the form* $X = \{x_1, x_2, \ldots, x_n\}$ *where* $x_i \subseteq D_i$*. The mass of a vector X in a line j is computed by conjunctive rule of combination of the bba's evidential values.*

$$m_j : \Theta \to [0, 1]$$

$$m_j(\emptyset) = 0$$

$$m_j(X) = \circledcirc_{i \leq n} m_{ij}(X) = \prod_{iv_i \in X} m_{ij}(iv_i)$$

Example 4 *To illustrate this late definition, we present here the first line body of evidence in our database example (table 1). The frame of discernment is* Θ *which is the cross product of all columns domains and the frame of discernment of all bodies of evidence of the database lines. Focal elements are combinations of all evidential items in the line, and thus all possible evidential itemsets in the record that are* $A_1B_1C_2$*,* $A_1B_1\{C_1, C_2\}$*,* $A_2B_1C_2$ *and* $A_2B_1\{C_1, C_2\}$*. The resulting mass distributions are the following:* $m_1(A_1B_1C_2) = 0.12$*,* $m_1(A_1B_1\{C_1, C_2\}) = 0.48$*,* $m_1(A_2B_1C_2) = 0.08$ *and* $m_1(A_2B_1\{C_1, C_2\}) = 0.32$*.*

Now, we introduce the notion of *evidential database body of evidence* which is induced from line body of evidence notion since database is a set of lines.

Definition 6 (Evidential Database BoE) *Body of evidence of evidential database EDB is defined on the frame of discernment* Θ*, the set of focal elements is composed of all possible evidential itemsets existing in the database and the mass function* m_{EDB} *is defined as follows: Let X be an evidential itemset and d be the size of EDB:*

$$m_{DB} : \Theta \to [0, 1] \text{ with } m_{DB}(X) = \frac{1}{d} \sum_{j=1}^{d} m_j(X)$$

Belief and Plausibility functions are naturally defined as follows:

$$Bel_{DB}(X) = \sum_{Y \subseteq X} m_{DB}(Y)$$

$$Pl_{DB}(X) = \sum_{Y \cap X \neq \emptyset} m_{DB}(Y)$$

Example 5 *In our database (table 1) the mass of evidential itemset* $\{A_2, A_3\}B_1C_2$ *is the sum of its line masses in the database divided by* $d = 2$ *so* $m_{BD}(\{A_2, A_3\}B_1C_2) = 0.2$*. Its belief in the database is the sum of all* *database masses of evidential itemsets that are included in, which are* $\{A_2, A_3\}B_1C_2$ *and* $A_2B_1C_2$ *so* $Bel_{BD}(\{A_2, A_3\}B_1C_2) = 0.24$.

The mass of an evidential itemset X in evidential database EDB is the partial belief attributed to X. The total belief of X is the sum of masses of evidential itemsets Y included in X which corresponds to belief of X in database body of evidence. Thus the support of X in EDB is simply its belief in EDB's body of evidence. The function *support* is monotone like in perfect databases [1] since the function belief itself is monotone [13]. Monotony property of support function in frequent itemsets mining from imperfect databases model is very important since it allows us to adapt the panoply of methods in perfect ARM literature [3].

5.2 Frequent Evidential Itemsets Mining Algorithm

Now we present our algorithm for mining frequent evidential itemsets from evidential databases under support threshold min_{sup}.

Let EDB be an evidential database, X be an evidential itemset and Θ be the cross product of all attribute domains. F is the set of frequent evidential itemsets in EDB under min_{sup}. F is formally defined as follows:

$$F = \{X \subseteq \Theta / support(X) \geq min_{sup}\}$$

Our algorithm proceeds in two major steps. In the first one we generate a data structure that stores the evidential database in a vertical way [15]. Then, we scan this data structure to generate frequent evidential itemsets. The two steps are described in detail in the following:

In the first one, we generate for every evidential item in EDB the *records identifiers* (*rids*) that include it. For example, the *rids* list of item A_1 in the database example is {1,2}. However, *rids* of evidential items are not enough to compute their supports, in opposition to the perfect databases context, where support of A_1 would be 2 (cardinal of *rids* list) since A_1 would be "surely" (not "probably") in records 1 and 2. That is why, *rid* information has to be extended by the item mass in the record denoted by *mir*. Thus, for each evidential item we construct the list of the couples (rid, mir). The set of (rid, mir) lists is the data structure that represents the database EDB. It is denoted by *RidLists*. Table 2 shows the *RidLists* corresponding to our database example:

Table 2: *RidLists* of database example

item	***rid*** **list**
A_1	$(1, 0.3)(2, 0.1)$
A_2	$(1, 0.2)$
$\{A_2, A_3\}$	$(2, 0.4)$
B_1	$(1, 0.5)(2, 0.5)$
C_1	$(2, 0.25)$
C_2	$(1, 0.1)(2, 0.25)$
$\{C_1, C_2\}$	$(1, 0.4)$

Procedure of construction of the *RidLists* denoted by *ConstructRIDLISTS* is described in the pseudo-code below (procedure 1). Notations of ambiguous objects used in all procedures of the paper are presented in table 3:

Procedure 1: *ConstructRIDLISTS(in EDB, out RidLists)*

```
For each record r in EDB
  For each cell c in r
    For each item i in c
      If i doesn't exists in ridlists then
        Add new rid list for i
      Add to i's rid list the couple (r.id, i.mass)
End ConstructRIDLISTS
```

The (rid, mir) list information allows us to compute the mass of one item in the database's *BoE* (it is the sum of masses in its *rid* list). However support of one evidential item is its *belief* in the database and not its *mass*. That is why we need to update the structure *RidLists* such that every item will have its list of couples *rid* and its **belief value** in that record. For this purpose, we scan every couple of items i_1 and i_2 in the *RidLists*. Next, if i_1 includes i_2 then i_1's *rid* list have to be updated by adding it all i_2 couples (rid, mir). The *RidLists* of our database example is updated in table 4.

Table 3: Notations of procedures objects

Object	Signification
Cell	Set of evidential items composing one evidential value. A record is a set of cells.
r.id	Identifier of the record in the evidential database.
i.mass	The mass of an item in evidential value.
$bel_r(i)$	Belief of item/itemset i in the record. r.

Table 4: The *RidLists* updated

item	***rid*** **list**
A_1	$(1, 0.3)(2, 0.1)$
A_2	$(1, 0.2)$
$\{A_2, A_3\}$	$(1, 0.2)(2, 0.4)$
B_1	$(1, 0.5)(2, 0.5)$
C_1	$(2, 0.25)$
C_2	$(1, 0.1)(2, 0.25)$
$\{C_1, C_2\}$	$(1, 0.5)(2, 0.5)$

The *RidLists* is updated by the following procedure called *UpdateBeliefsRIDLISTS*:

Procedure 2: *UpdateBeliefsRIDLISTS(inout* *RidLists)*

```
For each couple of items (i_1, i_2) ∈
RidLists × RidLists
  If item i_1 includes item i_2 Then
    For each (rid, mass) in i_2.list
      If rid ∈ i_1.list Then
        Add i_2.mass to i_1.mass for the same rid
      Else Add (rid, mass) in i_1.list
End UpdateBeliefsRIDLISTS
```

Once we have *RidLists* structure with couples $(rid, belief)$ for each item, the second step could start. It consists in generating frequent evidential itemsets from the *RidLists*. For this purpose, we must define a computation method of evidential itemsets. Indeed, the support of any evidential itemset is computed from intersection of *rid* lists of all its items. Let X be an evidential itemset. X exists in all records that include its evidential items. Belief value of X in each record is the product of its items believes in that record. Belief of X in the whole of the database is the sum of its believes in the database (see definitions 5 and 6). For example, let's compute support (belief) of evidential itemset $X = \{A_2, A_3\}C_2$. First of all, we compute intersection of $\{A_2, A_3\}$ list and C_2 list. The resulting list contains the *rids* 1 and 2. Then, we compute beliefs of X in each record. The list of $X = \{(1, 0.04), (2, 0.25)\}$. Then, we deduce that $bel(X) = 0.29$.

The following procedure compute belief value of any evidential itemset. We call it *Bel*:

Procedure 3: *Bel(in* *RidLists*, *in* X*)*

```
Compute list l = ⋂_{i∈X} RidLists(i)
For each rid r in l
    bel_r(X) = ∏_{i∈X} bel_r(i)
Define l as the rid list of X
For each rid r in l
  Bel(RidLists, X) = Bel(RidLists, X) + bel_r(X)
End Bel
```

We present now our main procedure called *ComputeFrequentItemsets* that mines frequent itemsets from the database EDB. The procedure is iterative since it generates candidate itemsets of size k via *Apriori_Gen* function [2], computes their supports in the database, then keeps only frequent ones and goes to next level $k + 1$ to generate candidate itemsets of size $k + 1$, etc. It starts obviously from evidential items (itemsets of size one). The pseudo-code of the procedure is presented in the following:

Procedure 4: *ComputeFrequentItemsets(in* EDB, *in* min_{sup}, *out* L*)*

```
ConstructRIDLISTS(EDB, RidLists)
UpdateBeliefsRIDLISTS(RidLists)
k = 1
For each item i in RidLists
  If Bel(RidLists, i) ≥ min_sup Then
    Add i to the set L_k
  Else Remove i's list from RidLists
L = L ∪ L_k
```

Do while $L_k \neq \emptyset$
  $C_{k+1} = Apriori\_Gen(L_k)$
  $k = k + 1$; $L_k = \emptyset$
  For each itemset $itmst$ in C_{k+1}
    If $Bel(RidLists, itmst) \geq min_{sup}$ Then
      Add $itmst$ to F_k
  $L = L \cup L_k$
Loop
End *ComputeFrequentItemsets*

6 Experimentation

To assess our method performance, we implemented our proposed algorithm and the algorithm of [10] to compare them. Then we implemented an algorithm that generates synthetic databases with the following parameters: D (database size), I (number of items in all columns domains), C (number of columns) and $\%U$ (percentage of records including evidential values). We generated several synthetic databases on which we tested the two algorithms, but we present here only tests led on the database $D = 5000, I = 800, C = 5, \%U = 10$.

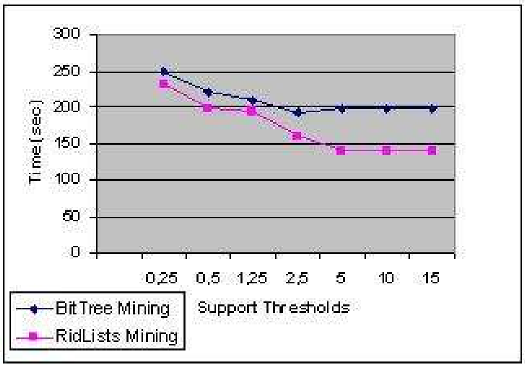

Figure 1: Comparison performance between BitTree Mining and RidLists Mining

Figure 1 shows that RidLists mining is more efficient than BitTree mining for the synthetic database $D5000I800C5\%U10$. Note that more I decreases (or D increases), more the number of frequent itemsets is important. For example, assume that we generate 1000 records randomly with only 20 items; we will obtain a dense (correlated) database since the twenty items will be repeated in the 1000 records. Experimentations led on various synthetic databases (with different generation parameters) showed that more I decreases (or D increases), more algorithms performances approach (figure 2). It simply means that our algorithm is especially efficient in sparse data, since its performance grow when I increases.

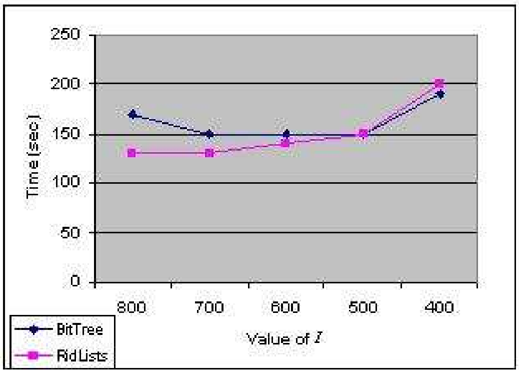

Figure 2: Algorithms performance for various values of I

We also evaluated performance algorithms for various uncertainty degree (figure 3). Experimentation results shows that performance algorithm decreases when uncertainty degree increases. That is logical since when uncertainty degree increases, belief computations increases:

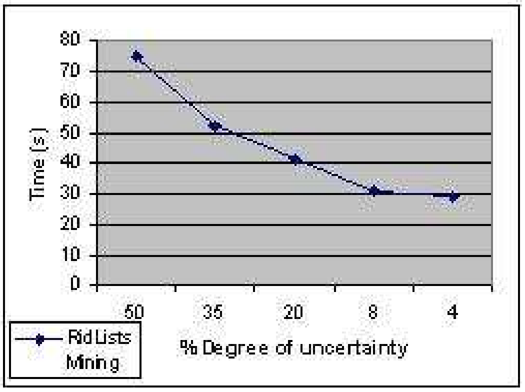

Figure 3: Algorithm performance for various % uncertainty degrees

7 Conclusion

In this paper we propose an itemset mining algorithm in context of evidential databases. This kind of databases provides a large field of uncertainty expression to end-user since they allow storage of probabilistic, possibilistic, evidential information and even missing values. All these imperfect information are processed by our proposed algorithm which makes mined patterns more accurate compared with real behavior of data.

Our work could be extended by *plausible* pattern mining, since mined itemsets in our model are *credible*, but we have no information about their plausibility. In other words, it will be interesting if we compute plausibilities of frequent (credible) evidential itemsets. We can even find infrequent itemsets that are more plausible than frequent ones. A study of this measurement will be interesting.

Finally, further studies on the construction of a complete framework of pattern *mining* and *maintenance* have to be performed, especially as our data structure *RidLists* is easily maintained in case of insertion/deletion of records. Maintenance of pattern is useful when databases are dynamic, i.e., updated frequently.

References

[1] R. Agrawal, T. Imielinski, A. Swami (1993). Mining Association Rules Between Sets of Items in Large Databases. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pp207-216, 1993.

[2] R. Agrawal, R. Srikant (1994). Fast Algorithms for Mining Association Rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pp 487-499, 1994.

[3] M-C. Tseng, W-Y. Lin (2005). Incremental Mining on Association Rules. In *Studies in Fuzziness and Soft Computing*, Volume 180, pp125-162, 2005.

[4] G. Chen, Q. Wei (2002). Fuzzy association rules and the extended mining algorithms. In *Information Sciences-Informatics and Computer Science*, Vol 147, Issue 1-4, pp201-228, 2002.

[5] C-K. Chui, B. Kao, E. Hung (2007). Mining Frequent Itemsets from Uncertain Data. In *The 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2007)*, pp 47-58, 2007

[6] A.P. Dempster (1967). Upper and Lower Probabilities Induced by a Multivalue Mapping. In *Annals of Mathematical Statistics*, volume 38, No 2, pp325-339, 1967.

[7] A.P. Dempster (1968). A Generalization of Bayesian Inference. In *Journal of the Royal Statistical Society*, volume 30, No 2, pp205-247, 1968.

[8] Y. Djouadi, S. Redaoui, K. Amroun (2007). Mining Association Rules under Imprecision and Vagueness: towards a Possibilistic Approach. In *Fuzzy Systems Conference*. Int. Vol, Issue 23-26, pp1-6, 2007.

[9] J. Han, J. Pei, Y. Yin (2000). Mining Frequent Patterns without Candidate Generation. In *Proceedings of the 2000 ACM-SIGMOD International Conference on Management of Data*, pp1-12, 2000.

[10] K. K. R. G. K. Hewawasam, K. Premaratne, S. P. Subasingha, M-L Shyu (2005). Rule Mining and Classification in Imperfect Databases. In *Proceedings of the Seventh International Conference on Information Fusion*, pp 661-668, 2005.

[11] C. K-S. Leung, C. L. Carmichael, B. Hao (2007). Efficient Mining of Frequent Patterns from Uncertain Data. In *Proceedings of the Seventh IEEE International Conference on Data Mining*, pp 489-494, 2007.

[12] C. Lucchese, S. Orlando, R. Perego, P. Palmerini, R. Perego, F. Silvestri (2003). kDCI: A Multi-Strategy Algorithm for Mining Frequent Sets. In *Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations* 2003.

[13] G. Shafer (1976). A Mathematical Theory of Evidence. In *Princeton University Press*, 1976.

[14] P. Smets (1998). The Application of the Transferable Belief Model to Diagnostic Problems. In *Int. J. Intelligent Systems*, volume 13, pp 127-158, 1998.

[15] S-J. Yen, A.L.P. Chen (1996). An efficient approach to discovering knowledge from large databases. In *Proc. of 4th Intl. Conf. on Parallel and Distributed Information Systems*, pp8-18, 1996.