Approximation of Frequent Itemset Border by Computing Approximate Minimal Hypergraph Transversals

Nicolas Durand and Mohamed Quafafou

Aix-Marseille Université, CNRS, LSIS UMR 7296, 13397, Marseille, France
{nicolas.durand,mohamed.quafafou}@univ-amu.fr
http://www.lsis.org

Abstract. In this paper, we present a new approach to approximate the negative border and the positive border of frequent itemsets. This approach is based on the transition from a border to the other one by computing the minimal transversals of a hypergraph. We also propose a new method to compute approximate minimal hypergraph transversals based on hypergraph reduction. The experiments realized on different data sets show that our propositions to approximate frequent itemset borders produce good results.

Keywords: frequent itemsets, borders, hypergraph transversals, approximation.

1 Introduction

The discovery of frequent itemsets has quickly become an important task of data mining [1]. This corresponds to find the sets of items (i.e. attribute values) which appear together in at least a certain number of transactions (i.e. objects) recorded in a database. These sets of items are called frequent itemsets. The main use of the frequent itemsets is the generation of association rules. Nevertheless, the uses have been extended to other tasks of data mining such as supervised classification and clustering [1]. Two points are important in the discovery of frequent itemsets. The first point is the reduction of the search space due to combinatorial explosion. The second point is the reduction of the number of generated itemsets to make easier the exploitation. In this paper, we focus on the second point. To reduce the number of itemsets, some condensed representations of frequent itemsets, such as the frequent closed itemsets, have been proposed [1]. The maximal frequent itemsets (w.r.t. set inclusion) also represent a reduced collection of itemsets. They correspond to a subset of the set of the frequent closed itemsets. The regeneration of all the frequent itemsets is possible from the maximal frequent itemsets but they are not considered as a condensed representation of frequent itemsets because the database must be read to compute the frequencies. The maximal frequent itemsets and the minimal infrequent itemsets correspond respectively to the positive border and the negative border

L. Bellatreche and M.K. Mohania (Eds.): DaWaK 2014, LNCS 8646, pp. 357–368, 2014.

of the set of the frequent itemsets [2]. These two borders are linked together by the computation of minimal hypergraph transversals (also called ”minimal hitting sets”) [2,3]. Thus, it is possible to switch to a border from the other one. The number of itemsets of the borders can still be huge.

In this paper, we propose a new approach to approximate the positive border of frequent itemsets in order to reduce the size of the border. This approach is based on the transition from a border to the other one by computing minimal hypergraph transversals. The approximation is obtained by the computation of approximate minimal transversals. Through the approximation, we also want to find new items which could be interesting for some applications like document recommendation. Another contribution is the proposition of a new method to approximate the minimal transversals of a hypergraph by reducing the hypergraph. To the best of our knowledge, this is the first time that such approaches are proposed. Some experiments have been performed on different data sets in order to evaluate the number of generated itemsets and the distance between the computed approximate borders and the exact borders. We focus on the comparison between our method and two other algorithms which compute approximate minimal transversals, in considering our approach of border approximation.

The rest of this paper is organized as follows. Section 2 defines the notations and the notions necessary for understanding the paper. The proposed approach of border approximation is detailed in Section 3. In Section 4, we present our method to compute approximate minimal hypergraph transversals. Related works are discussed in Section 5. The experiments and the results are presented in Section 6. We conclude in Section 7.

2 Preliminaries

Let $\mathcal{D} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$ be a data mining context, $\mathcal{T}$ a set of transactions, $\mathcal{I}$ a set of items (denoted by capital letters), and $\mathcal{R} \subseteq \mathcal{T} \times \mathcal{I}$ is a binary relation between transactions and items. Each couple $(t, i) \in \mathcal{R}$ denotes the fact that the transaction t is related to the item i. A transactional database is a finite and nonempty multi-set of transactions. Table 1 provides an example of a transactional database consisting of 6 transactions (each one identified by its ”Id”) and 8 items (denoted $A \ldots H$).

Table 1. Example of transactional database

Id	Items							
t_1	A		C		E		G	
t_2		B	C		E		G	
t_3	A		C		E			H
t_4	A			D		F		H
t_5		B	C			F		H
t_6		B	C		E	F		H

An ***itemset*** is a subset of $\mathcal{I}$ (note that we use a string notation for sets, e.g., AB for $\{A, B\}$). The complement of an itemset X (according to $\mathcal{I}$) is noted $\overline{X}$. A transaction t supports an itemset X iff $\forall i \in X, (t, i) \in \mathcal{R}$. An itemset X is ***frequent*** if the number of transactions which support it, is greater than (or is equal to) a miminum threshold value, noted *minsup*. The set of all-frequent itemsets is noted S. Let us take the example of Table 1, if *minsup*=3 then the itemset H is frequent because 4 transactions support it (t_3, t_4, t_5 and t_6). AE is not frequent because only t_1 and t_3 support it.

The set of all maximal frequent itemsets (resp. minimal infrequent itemsets), w.r.t. set inclusion, in $\mathcal{D}$ is the ***positive border*** (resp. ***negative border***) [2] of S and is noted $Bd^+(S)$ (resp. $Bd^-(S)$): $Bd^+(S) = \{X \in S \mid \forall Y\, tq\, X \subset Y,\, Y \notin S\}$ and $Bd^-(S) = \{X \in 2^{\mathcal{I}} \setminus S \mid \forall Y\, tq\, Y \subset X,\, Y \in S\}$. Let us take the example of Table 1, if *minsup*=3, $Bd^+(S) = \{A, BC, CE, CH, FH\}$ and $Bd^-(S) = \{D, G, AB, AC, AE, AF, AH, BE, BF, BH, CF, EF, EH\}$.

Before the presentation of the relationship between the positive border and the negative border of frequent itemsets, we need to introduce the notion of minimal transversals of a hypergraph. A ***hypergraph*** $\mathcal{H} = (V, E)$ is composed of a set V of vertices and a set E of hyperedges [4]. Each hyperedge $e \in E$ is a set of vertices included or equal to V. The ***degree*** of a vertex v in $\mathcal{H}$, denoted $deg_{\mathcal{H}}(v)$, is the number of hyperedges of $\mathcal{H}$ containing v. Let τ be a set of vertices ($\tau \subseteq V$). τ is a ***transversal*** of $\mathcal{H}$ if it intersects all the hyperedges of $\mathcal{H}$. A transversal is also called a "hitting set". The set of all the transversals of $\mathcal{H}$ is: $Tr(\mathcal{H}) = \{\tau \subseteq V \mid \forall e_i \in E, \tau \cap e_i \neq \emptyset\}$. A transversal τ of $\mathcal{H}$ is ***minimal*** if no proper subset is a transversal of $\mathcal{H}$. The set of all minimal transversals of $\mathcal{H}$ is noted $MinTr(\mathcal{H})$. The relationship between the notion of borders and minimal transversals has been presented in [2] and [3].

In [2], the following property has been showed:

$$Bd^-(S) = MinTr(\overline{Bd^+(S)})$$

where $\overline{Bd^+(S)}$ is the hypergraph formed by the items of $\mathcal{I}$ (i.e. the vertices) and the complement of the itemsets of the positive border of S (i.e. the hyperedges).

In [3], the following property has been showed:

$$Bd^+(S) = \overline{MinTr(Bd^-(S))}$$

where $Bd^-(S)$ is the hypergraph formed by the items of $\mathcal{I}$ (i.e. the vertices) and the itemsets of the negative border of S (i.e. the hyperedges).

The term ***dualization*** refers to the use of the previous properties to compute the negative border from the positive border, and vice versa. The size of the borders can be huge according to *minsup*. In the two next sections, we propose a new approach to approximate the borders and to reduce their size. In this way, the exploitation of the itemsets of the borders will be easier.

3 Proposed Approach of Border Approximation

The proposed approach of border approximation exploits the dualizations between the positive border and the negative border. Let f and g be the functions that allow to compute respectively the negative border from the positive border and the positive border from the negative border:

$$f : \left| \begin{array}{l} 2^{\mathcal{I}} \to 2^{\mathcal{I}} \\ x \mapsto MinTr(\overline{x}) \end{array} \right. \qquad g : \left| \begin{array}{l} 2^{\mathcal{I}} \to 2^{\mathcal{I}} \\ x \mapsto \overline{MinTr(x)} \end{array} \right.$$

The principle of the proposed approach is to replace the function f by a function $\widetilde{f}$ which performs an approximate computation of the negative border. The new function $\widetilde{f}$ uses an approximate minimal transversals computation, noted $\widetilde{MinTr}$:

$$\widetilde{f} : \left| \begin{array}{l} 2^{\mathcal{I}} \to 2^{\mathcal{I}} \\ x \mapsto \widetilde{MinTr}(\overline{x}) \end{array} \right.$$

From the positive border, the approach computes an approximate negative border (noted $\widetilde{Bd^-}(S)$): $\widetilde{f}(Bd^+(S)) = \widetilde{MinTr}(\overline{Bd^+(S)}) = \widetilde{Bd^-}(S)$. The return to a positive border (via the function g) allows to obtain an approximate positive border (noted $\widetilde{Bd^+}(S)$): $g(\widetilde{Bd^-}(S)) = \overline{MinTr(\widetilde{Bd^-}(S))} = \widetilde{Bd^+}(S)$. Thus, our approach produces the approximate negative border $\widetilde{Bd^-}(S)$ and the corresponding approximate positive border $\widetilde{Bd^+}(S)$. Let us take the example of Table 1 and let us compute the approximate border: $\widetilde{Bd^-}(S) = \widetilde{f}(Bd^+(S)) = \widetilde{MinTr}(\overline{Bd^+(S)}) = \widetilde{MinTr}(\{\overline{A}, \overline{BC}, \overline{CE}, \overline{CH}, \overline{FH}\} = \widetilde{MinTr}(\{BCDEFGH, ADEFGH, ABDFGH, ABDEFG, ABCDEG\})$. Let us assume that the approximate minimal transversals computation provides the following result: $\widetilde{Bd^-}(S) = \{D, E, G, AF, AH, BF, BH\}$. The approximate positive border is obtained by dualization: $\widetilde{Bd^+}(S) = g(\widetilde{Bd^-}(S)) = \overline{MinTr(\widetilde{Bd^-}(S))} = \{\overline{ABDEG}, \overline{DEFGH}\} = \{CFH, ABC\}$. We can remark that A, B, C and BC are frequent itemsets (according to $minsup = 3$) and here ABC is considered as a frequent itemset. CFH is not frequent (its support is equal to 2) but it is almost frequent. These two itemsets can be interesting for applications like document recommendation. For instance, without our approach, CF is frequent and CFH is not frequent. The item H is potentially interesting. If the items are documents, with our approach, the item H can be recommended to a user.

4 Computation of Approximate Minimal Transversals

In order to complete the approach presented in the previous section, we proposed a method to compute the approximated minimal transversals of a hypergraph. The method is based on the reduction of the initial hypergraph. The aim is to compute the minimal transversals on the reduced hypergraph (smaller than the initial hypergraph). The proposed algorithm of reduction is specially designed to compute minimal transversals. It exploits the fact that the hyperedges formed by the complements of the itemsets of the positive border, strongly intersect (i.e. the average degree of a vertex is high). Indeed, in the example this hypergraph is: $\{BCDEFGH, ADEFGH, ABDFGH, ABDEFG, ABCDEG\}$. The proposed method is composed of two steps: (1) Reduction of the hypergraph, (2) Computation of the (exact) minimal transversals of the reduced hypergraph. At the end, the minimal transversals obtained from the reduced hypergraph are declared as the approximate minimal transversals of the initial hypergraph.

4.1 Reduction of the Hypergraph

The hypergraph reduction of the initial hypergraph is based on the intersections of its hyperedges and on the degree of each vertex. The representative graph [4] (also called "line-graph") of the hypergraph is thus generated. Let us recall that the representative graph of the hypergraph $\mathcal{H}$ is a graph whose vertices represent the hyperedges of $\mathcal{H}$ and two vertices are adjacent if and only if the corresponding hyperedges in $\mathcal{H}$ intersect. In our algorithm, we add values to the edges of the representative graph. Algorithm 1 presents the reduction of a hypergraph $\mathcal{H}$. The algorithm is composed of three steps: (1) Computation of the degree of each vertex in $\mathcal{H}$ (lines 1-3), (2) Generation of the valued representative graph of $\mathcal{H}$ (lines 4-9), (3) Generation of the reduced hypergraph from the valued representative graph (lines 10-17). The complexity of the algorithm is in $O(m^2)$ where m is the number of hyperedges of the initial hypergraph.

Algorithm 1. HR (Hypergraph Reduction)

```
Require: a hypergraph $\mathcal{H}$=($V$, $E$) where $|V|$=$n$ and $|E|$=$m$
Ensure: the reduced hypergraph $\mathcal{H}_R$
 1: for all $v \in V$ do
 2:     Compute $deg_{\mathcal{H}}(v)$
 3: end for
 4: $V' \leftarrow \{v'_i\}$ $i = 1, \ldots, m$; {each $v'_i \in V'$ represents $e_i \in E$}
 5: $E' \leftarrow \{\}$;
 6: for all $v'_i \cap v'_j \neq \emptyset$ do
 7:     $E' \leftarrow E' \cup \{(v'_i, v'_j)\}$;
 8:     $w_{(v'_i,v'_j)} \leftarrow \sum_{v\in\{\psi^{-1}(v'_i)\cap\psi^{-1}(v'_j)\}} deg_{\mathcal{H}}(v)$;
 9: end for
10: $V_R \leftarrow \{\}$;
11: $E_R \leftarrow \{\}$;
12: while $E' \neq \emptyset$ do
13:     Select $e'max = (v'max_i, v'max_j)$ having the maximal weight value
14:     $V_R \leftarrow V_R \cup \{\psi^{-1}(v'max_i) \cap \psi^{-1}(v'max_j)\}$;
15:     $E_R \leftarrow E_R \cup \{\{\psi^{-1}(v'max_i) \cap \psi^{-1}(v'max_j)\}\}$;
16:     Delete the edges $e' \in E'$ where $v'max_i$ or $v'max_j$ is present
17: end while
18: return $\mathcal{H}_R$;
```

Valued Representative Graph Generation (lines 1-9). Let be $\mathcal{H} = (V, E)$ a hypergraph ($|V| = n$ and $|E| = m$). The algorithm constructs a valued graph G=(V', E') where $V' = \{v'_i\}$ ($i = 1, \ldots, m$) and $E' = \{e'_k\}$ ($k = 1, \ldots, l$). A vertex v'_i represents a hyperedge e_i from $\mathcal{H}$. Let be $\psi : E \to V'$ the bijective function who associates a hyperedge e_i to a vertex v'_i. A hyperedge between v'_i and v'_j shows that the intersection between the hyperedges $\psi^{-1}(v'_i)$ and $\psi^{-1}(v'_j)$ (e_i and e_j from $\mathcal{H}$) is not empty. The weight of an edge is based on the degree of each vertex in the corresponding intersection. To evaluate the weight of a generated edge, we use the degree of each vertex from the initial hypergraph.

The idea is that a vertex very present has a good chance to be in a minimal transversal. This expresses a "degree" of transversality. If the degree of a vertex is equal to the number of hyperedges then this vertex is a minimal tranversal. Let us note that this heuristic is used by several algorithms that compute transversals [5,6]. The weight of an edge $e'_k = (v'_i, v'_j)$, noted $w_{e'_k}$, is the sum of the degree of the vertices present in the intersection which has led to create this edge (see (1)).

$$w_{e'_k} = \sum_{v \in \{\psi^{-1}(v'_i) \cap \psi^{-1}(v'_j)\}} deg_{\mathcal{H}}(v). \tag{1}$$

Generation of the Reduced Hypergraph (lines 10-17). After the creation of the valued representative graph, the algorithm performs a selection of edges with a greedy approach. It selects the edge having the higher weight value while there are edges left in G. Each selected edge is transformed to a hyperedge of the reduced hypergraph. This hyperedge contains the vertices from $\mathcal{H}$ corresponding to the intersection of the two vertices of the edge. We obtain, at the end, a set of hyperedges corresponding to the reduced hypergraph $\mathcal{H}_R$=(V_R,E_R). Let us remark that if several edges have the same weight, the first found edge is selected.

Let us consider the example of Table 1 as a hypergraph $\mathcal{H}$ (6 hyperedges, 8 vertices). The reduced hypergraph is $\mathcal{H}_R = (V_R, E_R)$ where $V_R = \{A, B, C, E, F, H\}$ and $E_R = \{\{A, C, E\}, \{B, C, F, H\}\})$.

4.2 Minimal Transversal Computation

The last step is the computation of the (exact) minimal transversals of the reduced hypergraph. These transversals correspond to the approximate minimal transversals of the initial hypergraph: $\widetilde{MinTr}(\mathcal{H}) = MinTr(\mathcal{H}_R)$.

Let us take our example, the minimal transversals of $\mathcal{H}_R$ are: $\{C, AB, AF, AH, BE, EF, EH\}$. We consider them as the approximate minimal transversals of $\mathcal{H}$. Let us remark that the (exact) minimal transversals of $\mathcal{H}$ are: $\{AB, AC, CD, CF, CH, EF, EH, GH, AFG, BDE\}$.

5 Related Works

Numerous methods have been proposed to reduce the number of itemsets. In [7], the authors have studied the problem of randomly sampling maximal itemsets without explicit enumeration of the complete search space. They have employed a simple random walk that only allows additions of singletons to the current set untill a maximal set is found. In [8], the authors have used the Minimum Description Length (MDL) principle: the best set of itemsets is that set that compresses the database best. In [9], the approximation of a collection of frequent itemsets by the k best covering sets has been studied. The proposed algorithm takes in input the whole collection of the frequent itemsets or the positive border. The authors have explained the difficulties to use a greedy algorithm to obtain, from the positive border, k covering sets belonging to the initial collection. Our approach computes an approximate border from a border given completely in input. In that respect, we are close to the works presented in [9]. The exact positive border is the algorithm's input. Nevertheless, we do not want to find some covering sets and the itemsets of the approximate border do not necessarily belong to the initial collection. Our approach is more controllable than MDL used in [8]. We have the possibility to have several different methods to approximate the negative border ($\widetilde{f}$). Moreover, we have an understanding mapping between the exact border and the approximation border.

The computation of minimal transversals is a central point in hypergraph theory [4]. The algorithms to compute the minimal transversals come from different domains: graph theory, logic and data mining [10]. This is a NP-hard problem. The algorithms of approximation of minimal transversals are rare. Some works approximate the minimal transversals in order to obtain some ones or only one [6]. Some works are based on an evolutionary computation [11] where the transversality and the minimality are transcribed in a fitness function where a parameter, noted ϵ, is the fraction of hyperedges needed to intersect by any generated transversals. In [5], the *Staccato* algorithm computes low-cost approximate minimal transversals with a depth-first search strategy. It has been designed for model-based diagnosis. We have adapted *Staccato* in order to compute approximated minimal transversals in general. *Staccato* sorts the vertices according to their degree in increasing order. At each step, only the first λ (%) vertices of the remaining hypergraph are used. The more the λ value is high, the more the result is close to the set of the minimal transversals. The algorithm presented in [12], that we call δ-*MTminer*, produces minimal transversals which can miss at most δ hyperedges. It uses a breadth-first search strategy and several itemset discovery techniques. We have a different approach to compute approximate minimal transversals. We propose to apply a hypergraph reduction and then to compute the minimal transversals of the reduced hypergraph. These transversals are considered as the approximate minimal transversals of the initial hypergraph. Moreover, we don't need to set any parameters.

6 Experiments

6.1 Data and Protocol

Four data sets have been used: Mushroom, Chess, Connect and Kosarak. They have been downloaded from the FIMI web site[1]. Mushroom contains data on 23 species of gilled mushrooms. Chess contains some strategies for chess sets. Connect contains strategies for the game of connect-4. Kosarak contains anonymized click-stream data of a hungarian on-line news portal. The data sets (see Table 2) have been chosen to cover the different types of existing data sets according to the classification proposed by Gouda & Zaki [13].

The protocol of the experiments is as follows: For each data set and for some minimum support threshold values, (1) Compute the positive border according the minimum support threshold value, with *IBE* [14], (2) Compute the (exact)

[1] Frequent Itemset Mining Implementations, `http://fimi.ua.ac.be/data/`

Table 2. Data sets used in the experiments

Dataset	#transactions	#items	Avg. size of a trans.	Gouda & Zaki
Mushroom	8124	119	23	type 4
Chess	3196	75	37	type 1
Connect	67557	129	43	type 2
Kosarak	990002	41270	8,1	type 3

negative border with *Border-Diff* [15], the approximate negative border with 1-*MTminer* [12], the approximate negative border with *Staccato* [5], and the approximate negative border with our method (noted *AMTHR* - *Approximate Minimal Transversals by Hypergraph Reduction*), (3) Dualize to the positive borders (1 exact border and 3 approximate borders) with the *Border-Diff* algorithm which computes minimal transversals. For δ-*MTminer* (cf. Section 5), we have set δ to 1 because this value has produced the best results for δ-*MTminer*. For *Staccato* (cf. Section 5), we have chosen the highest values of λ before being impraticable: λ=0.8 for Mushroom, λ=0.65 for Chess, λ=0.7 for Connect, and λ=0.95 for Kosarak. In Steps 2 and 3, some statistics are computed: the number of itemsets of the computed border, the average size of the itemsets of the computed border, and the distance between the set of the itemsets of the computed border and the set of itemsets of the exact border. To evaluate the distance between two borders, we have used the distance of Karonski & Palka based on the Hausdorff distance. The cosine distance (see (2)) have been chosen to compute the distance between two elements (i.e. two itemsets). The distance D between two set of itemsets $\mathcal{X}$ and $\mathcal{Y}$ is defined in (3).

$$d(X,Y) = 1 - \frac{|X \cap Y|}{\sqrt{|X| \times |Y|}}. \tag{2}$$

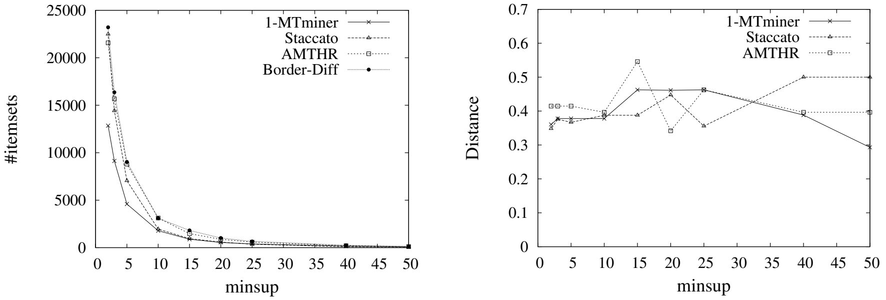
$$D(\mathcal{X},\mathcal{Y}) = \frac{1}{2}\{h(\mathcal{X},\mathcal{Y}), h(\mathcal{Y},\mathcal{X})\} \text{ where } h(\mathcal{X},\mathcal{Y}) = \max_{X\in\mathcal{X}} \{\min_{Y\in\mathcal{Y}} d(X,Y)\}. \tag{3}$$

6.2 Results and Discussion

Due to space constraints, Figures about the average size of an itemset of the computed borders, are not presented in the paper. All the figures are available online[2]. Moreover, we do not present the execution times because this is not our main objective. For information, the computation of $\widetilde{Bd^-(S)}$ with *AMTHR* is longer than with the other algorithms. The computation of $\widetilde{Bd^+(S)}$ is the fastest with *AMTHR*.

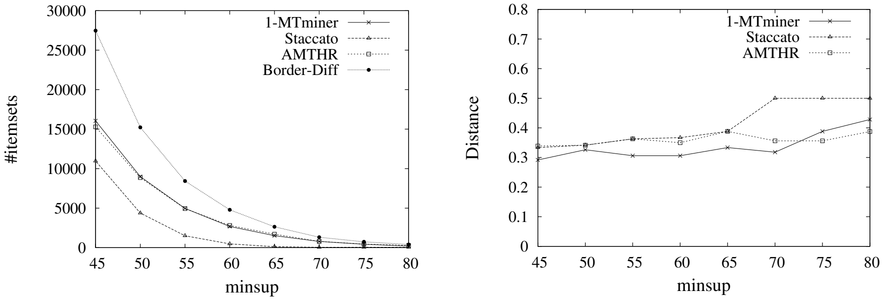
Fig. 1, 2, 3 and 4 present, for each data sets, the number of itemsets of the computed negative borders and the distance between the computed negative borders and the exact negative borders. We can observe that the cardinality of $\widetilde{Bd^-(S)}$ is lower than the cardinality of $Bd^-(S)$ for each data sets. For information, the itemsets of $\widetilde{Bd^-(S)}$ are shorter than the itemsets of $Bd^-(S)$. They

[2] `http://nicolas.durand.perso.luminy.univ-amu.fr/amthr/`



(a) Number of itemsets (b) Distance to $Bd^-(S)$

Fig. 1. Negative borders computed on Mushroom (according to *minsup*)

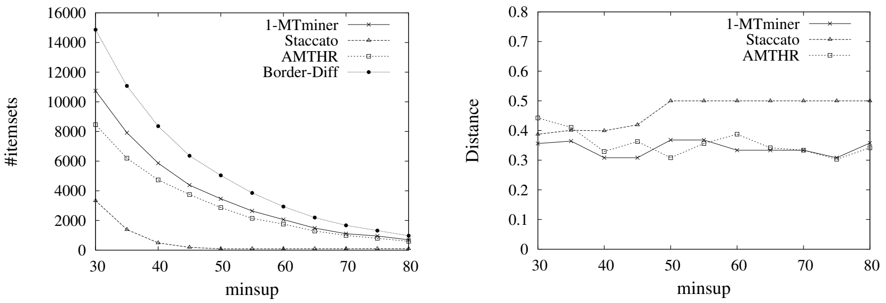


(a) Number of itemsets (b) Distance to $Bd^-(S)$

Fig. 2. Negative borders computed on Chess (according to *minsup*)

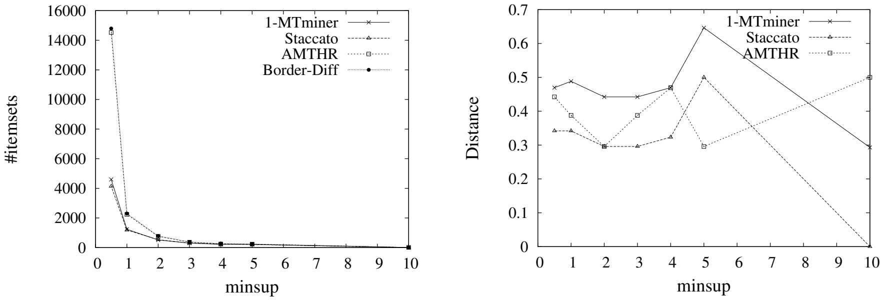
are the shortest with *Staccato* for each data sets. On Mushroom and Kosarak, the number of itemsets of $\widetilde{Bd^-(S)}$ produced by $AMTHR$ is very close to the number of itemsets of $Bd^-(S)$. The generated itemsets with $AMTHR$ are a little shorter than for the exact borders on Mushroom and Kosarak. Nevertheless, the itemsets of $\widetilde{Bd^-(S)}$ are different in view of the observed distances. This is an interesting remark. Some itemsets have been changed and they can be potentially interesting items. On Chess and Connect, $AMTHR$ and 1-*MTminer* have produced a similar number of itemsets. These itemsets have a very close average size. Regarding the distance (between $\widetilde{Bd^-(S)}$ and $Bd^-(S)$), *Staccato* has obtained the closest borders on Mushroom and Kosarak. 1-*MTminer* has produced the closest borders on Chess and Connect. Nevertheless, we can observe that $AMTHR$ is close to the best algorithm for each data sets.

Fig. 5, 6, 7 and 8 present, for each data sets, the number of itemsets of the computed positive borders and the distance between the computed positive borders and the exact positive borders. For information, the itemsets of $\widetilde{Bd^+(S)}$ are longer than the itemsets of $Bd^+(S)$. They are the longest with *Staccato* or $AMTHR$ on each data sets. The number of itemsets of $\widetilde{Bd^+(S)}$ with $AMTHR$ is the lowest on Mushroom. On the other data sets, *Staccato* has generated



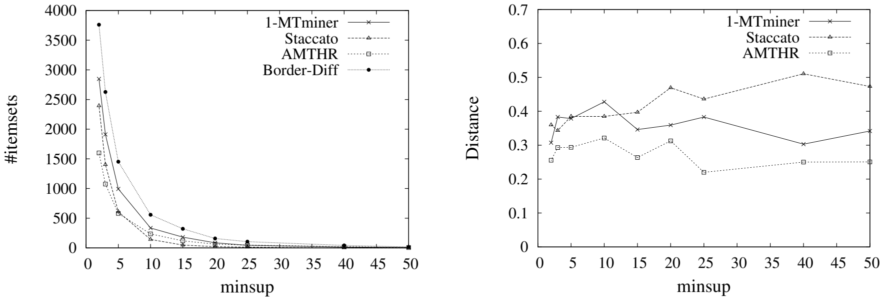
(a) Number of itemsets (b) Distance to $Bd^-(S)$

Fig. 3. Negative borders computed on Connect (according to *minsup*)



(a) Number of itemsets (b) Distance to $Bd^-(S)$

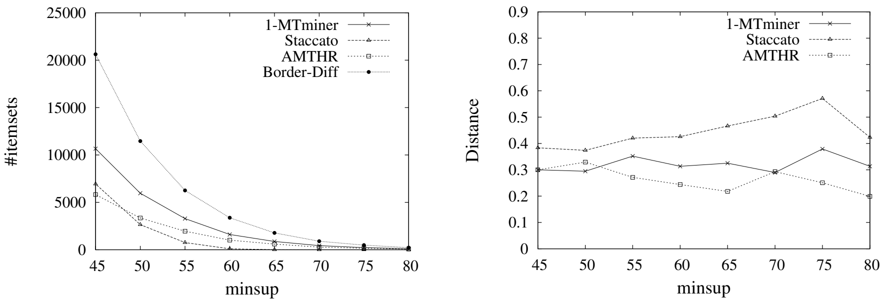
Fig. 4. Negative borders computed on Kosarak (according to *minsup*)



(a) Number of itemsets (b) Distance to $Bd^+(S)$

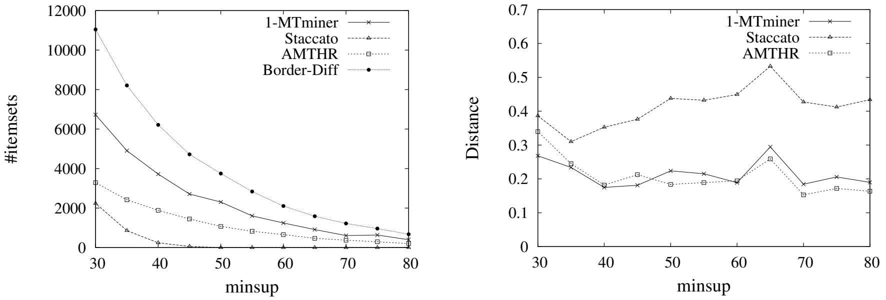
Fig. 5. Positive borders computed on Mushroom (according to *minsup*)

the lowest number of itemsets. 1-*MTminer* have produced more itemsets than *AMTHR*, except for Kosarak. *AMTHR* has obtained the closest $\widetilde{Bd^+(S)}$ to $Bd^+(S)$ on Mushroom, Chess and Kosarak. On Connect, 1-*MTminer* has also obtained good results. On Kosarak, *Staccato* and δ-*MTminer* have produced bad results.



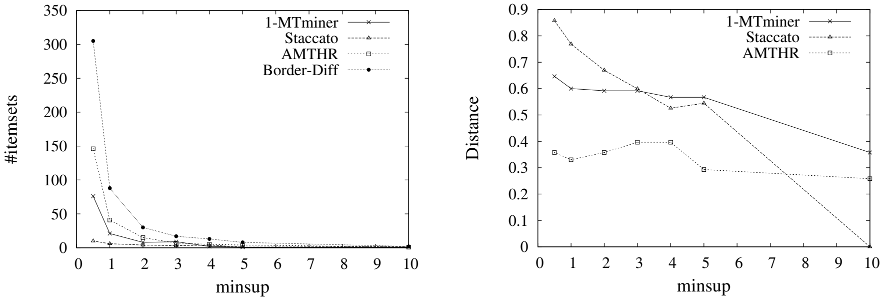
(a) Number of itemsets (c) Distance to $Bd^{+}(S)$

Fig. 6. Positive borders computed on Chess (according to *minsup*)



(a) Number of itemsets (b) Distance to $Bd^{+}(S)$

Fig. 7. Positive borders computed on Connect (according to *minsup*)



(a) Number of itemsets (b) Distance to $Bd^{+}(S)$

Fig. 8. Positive borders computed on Kosarak (according to *minsup*)

To resume, we can note that the proposed method, *AMTHR*, has reduced the number of itemsets of the generated positive borders, while keeping a reasonable distance to the exact positive borders. Moreover, our method seems to be robust according to the different types of data sets.

7 Conclusion

We have proposed a new approach of approximation of frequent itemset borders based on the computation of approximate minimal hypergraph transversals. From the exact positive border, an approximate negative border is computed and then the corresponding approximate positive border is generated by dualization. The proposed computation of approximate minimal transversals is based on hypergraph reduction. There is no need to set any parameters. In the experiments, we have showed that our method produces an approximate positive border smaller than the exact positive border, while keeping a reasonable distance with the exact border. These results confirm that the proposed method is interesting to find potentially interesting new items. In the future, we will thus develop a recommendation system using the approximate positive borders. In that way, we will able to evaluate the quality of the approximation.

References

1. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Mining and Knowledge Discovery 15, 55–86 (2007)
2. Mannila, H., Toivonen, H.: Levelwise Search and Borders of Theories in Knowledge Discovery. Data Mining and Knowledge Discovery 1(3), 241–258 (1997)
3. De Marchi, F., Petit, J.: Zigzag: a New Algorithm for Mining Large Inclusion Dependencies in Database. In: ICDM 2003, pp. 27–34 (November 2003)
4. Berge, C.: Hypergraphs: Combinatorics of Finite Sets, vol. 45. North Holland Mathematical Library (1989)
5. Abreu, R., van Gemund, A.: A Low-Cost Approximate Minimal Hitting Set Algorithm and its Application to Model-Based Diagnosis. In: SARA 2009 (July 2009)
6. Ruchkys, D.P., Song, S.W.: A Parallel Approximation Hitting Set Algorithm for Gene Expression Analysis. In: SBAC-PAD 2002, pp. 75–81 (2002)
7. Moens, S., Goethals, B.: Randomly Sampling Maximal Itemsets. In: IDEA 2013, Chicago, Illinois, USA, pp. 79–86 (2013)
8. Vreeken, J., van Leeuwen, M., Siebes, A.: Krimp: Mining Itemsets that Compress. Data Mining and Knowledge Discovery 23(1) (2011)
9. Afrati, F., Gionis, A., Mannila, H.: Approximating a Collection of Frequent Sets. In: KDD 2004, pp. 12–19 (August 2004)
10. Hagen, M.: Algorithmic and Computational Complexity Issues of MONET. PhD thesis, Friedrich-Schiller-Universitat Jena (November 2008)
11. Vinterbo, S., Øhrn, A.: Minimal Approximate Hitting Sets and Rule Templates. Approximate Reasoning 25, 123–143 (2000)
12. Rioult, F., Zanuttini, B., Crémilleux, B.: Nonredundant Generalized Rules and Their Impact in Classification. In: Ras, Z.W., Tsay, L.-S. (eds.) Advances in Intelligent Information Systems. SCI, vol. 265, pp. 3–25. Springer, Heidelberg (2010)
13. Gouda, K., Zaki, M.J.: Efficiently Mining Maximal Frequent Itemsets. In: ICDM 2001, pp. 163–170 (November 2001)
14. Satoh, K., Uno, T.: Enumerating Maximal Frequent Sets Using Irredundant Dualization. In: Grieser, G., Tanaka, Y., Yamamoto, A. (eds.) DS 2003. LNCS (LNAI), vol. 2843, pp. 256–268. Springer, Heidelberg (2003)
15. Dong, G., Li, J.: Mining Border Descriptions of Emerging Patterns from Dataset-Pairs. Knowledge and Information Systems 8(2), 178–202 (2005)