A thorough experimental study of datasets for frequent itemsets

Frédéric Flouvat[1], Fabien De Marchi[2,3], Jean-Marc Petit[2,4]

[1] Laboratoire LIMOS, UMR CNRS 6158
Université Clermont-Ferrand II,
63 177 Aubière, France
flouvat@isima.fr

[2] Laboratoire LIRIS, UMR CNRS 5205
[3] Université Lyon I [4] INSA Lyon
69 621 Villeurbanne, France
{fabien.demarchi, jmpetit}@liris.cnrs.fr

Abstract

The discovery of frequent patterns is a famous problem in data mining. While plenty of algorithms have been proposed during the last decade, only a few contributions have tried to understand the influence of datasets on the algorithms behavior. Being able to explain why certain algorithms are likely to perform very well or very poorly on some datasets is still an open question.

In this setting, we describe a thorough experimental study of datasets with respect to frequent itemsets. We study the distribution of frequent itemsets with respect to itemsets size together with the distribution of three concise representations: frequent closed, frequent free and frequent essential itemsets. For each of them, we also study the distribution of their positive and negative borders whenever possible.

From this analysis, we exhibit a new characterization of datasets and some invariants allowing to better predict the behavior of well known algorithms.

The main perspective of this work is to devise adaptive algorithms *with respect to dataset characteristics.*

1 Introduction

The discovery of frequent patterns is a famous problem in data mining, introduced in [2] as a first step for mining association rules. While plenty of algorithms have been proposed during the last decade [3, 10, 20, 22, 35], only a few contributions have tried to understand the influence of dataset characteristics on the algorithms behavior [19, 20, 32]. These studies focus on the number of transactions, average length of transactions, or frequent itemsets distribution, i.e. statistics from frequent itemsets and maximal frequent itemsets are usually given. Nevertheless algorithms could have quite different behaviors for (apparently) similar datasets. Benchmarks comparing algorithms performances have been done on real and synthetic datasets [5, 19] (see FIMI website [18]). Algorithm implementations and datasets are freely available from [18] for mining frequent, frequent closed or frequent maximal itemsets. Even with all these informations, being able to explain why certain algorithms are likely to perform very well or very poorly on some datasets is still an open question.

More generally, studying datasets can provide useful hints for devising adaptive algorithms [17, 31], i.e. algorithms which adapt themselves to data characteristics in order to increase their time or memory efficiency. Adaptive behavior of algorithms is not new in the setting of frequent itemsets mining, for example [7, 10] use heuristics to decide when tries-like data structure, representing datasets and/or itemset collections, have to be rebuilt. The promising results obtained by these algorithms show the interest of applying specific strategies according to dataset features.

Another key point is that some problems have specific invariant characteristics, whatever the studied datasets. Their impact on algorithms could give useful information about the difficulty to solve these problems while giving hints on the more appropriate strategies to cope with these difficulties.

Related works Classical characteristics of datasets were studied in [20], and more particularly a density criteria. Up to our knowledge no formal definition of density does exist. According to [20], a dataset is *dense* when it produces many long frequent itemsets even for high values of minimum support threshold. The authors studied seven datasets, each of them capturing a fairly large range of typical uses. The result of these experimentations is a classification of datasets in four categories according to the density. The density is estimated by using the characteristics of maximal frequent itemsets.

The main problem of their classification concerns its variability with respect to minimum support threshold val-

ues. For example, a dataset could belong to the first category for a given threshold value, and to the second category for another threshold value[1]. Moreover, there is no clear relationship between the proposed classification and algorithms performances. Even worse, a surprising result was obtained in the last FIMI workshop [5]: algorithms seem to be more efficient on some very dense datasets than on some other sparser datasets. Note also that in [37], in order to easily compare different implementations, a tool has been developped from information available at the FIMI website.

Based on the works done in [20], [32] proposed a statistical property of transactional datasets to characterize dataset density. Actually, they consider the dataset as a transaction source and measure an entropy signal, i.e. the transactions produced by such a source. Moreover, they show how such a characterization can be used in many fields, from performance prediction, minimum support threshold range determination, sampling, to strategy decisions. As for the previous work, it does not explain algorithms performances anymore. This may be due to the fact that only frequent itemsets are used to calculate the entropy measure.

In [27], the positive border distribution (i.e. the number of maximal elements in each level) is considered as a key parameter to characterize transaction databases. It is proved that any distribution is "feasible", and thus susceptible to be met in practice. Moreover, a constructive theorem is proposed to compute a synthetic transaction database given a positive border distribution as input. Nevertheless, the negative border is never considered and as a result, such synthetic databases do not match the "complexity" of real-world datasets.

Contribution In this setting, we describe a thorough experimental study of datasets with respect to frequent itemsets. We study the *distribution* of frequent itemsets with respect to itemsets size together with the distribution of three concise representations: frequent closed, frequent free and frequent essential itemsets. For each of them, we also study the distribution of their positive and negative borders whenever possible. From this analysis, we exhibit a new classification of datasets and some invariants allowing to better predict the behavior of well known algorithms.

The main perspective of this work is to devise *adaptive algorithms* with respect to dataset/problem characteristics.

Paper organization In section 2, we introduce some preliminaries. Experimental study of datasets is given in section 3, including usual representations of frequent itemsets, experimental protocol, results and analysis. The section 4 presents the main results of this work: a new dataset classification and a study of the influence of anti-monotone predicate on the resolution of some problems. Finally, we conclude and give some perspectives for this work.

[1] As a concrete example, this case arises with $Pumsb*$ dataset with minimum support threshold values equal to 15% and 25% respectively. Other examples are given in [16].

2 Preliminaries

Let R be a set of symbols called *items*, and r a database of subsets of R. The elements of r are called transactions. An *itemset* X is a set of some items of R. The support of X is the number of transactions in r that contain all items of X. An itemset is frequent if its support in r exceeds a minimum support threshold value, called $minsup$. Given a minimal support threshold and a database, the goal is to find all frequent itemsets.

We recall the notion of borders of a set using notations given in [30]. Let $(I, \preceq)$ be a partially ordered set of elements. A set $S \subseteq I$ is *closed downwards* if, for all $X \in S$, all subsets of X are also in S. S can be represented by its *positive border* $\mathcal{B}d^+(S)$ or its *negative border* $\mathcal{B}d^-(S)$ defined by:

$$\mathcal{B}d^+(S) = max_{\subseteq}\{X \in S\}$$
$$\mathcal{B}d^-(S) = min_{\subseteq}\{Y \in I - S\}$$

Let p be an anti-monotone predicate on $(I, \preceq)$, i.e. $\forall X, Y \in I, X \preceq Y$, if $p(Y)$ is true, then $p(X)$ is true. If S is the set of elements of I satisfying p, then S is closed downwards.

For instance, a set of frequent itemsets FI in a database with respect to a given minimum support threshold value is closed downwards. In this case, $\mathcal{B}d^+(FI)$ is often called the set of *maximal frequent itemsets*.

3 Thorough experimental study of datasets

In order to introduce our experimental study, we first describe three classical representations of frequent itemsets. Then, our experimental protocol is explained and our experimental results are given and discussed. To end up, a relationship between these results and algorithms performances is also pointed out.

3.1 Usual representation of frequent itemsets

Several concise (or condensed) representations of frequent itemsets have been studied [12, 29]. Their goal is twofold: improving efficiency of frequent itemsets mining whenever possible, and compacting the storage of frequent itemsets for future usages.

Formally, a condensed representation must be equivalent to frequent itemsets: one can retrieve each frequent itemset *together with its frequency* without accessing data [12].

Such a representation is known as *closed sets* [33, 34, 38]. Two other representations are considered in this paper: frequent free itemsets [4, 8] and frequent essential itemsets [13]. Notice that these sets are not exactly sufficient to represent frequent sets, since they need a subset of the frequent itemsets border to become condensed representations [12].

We briefly describe these representations in the rest of this section.

Frequent Closed sets Given an itemset X, *the closure* of X is the set of all items that appear in all transactions where X appears. Formally, given a transaction database r:

$$Cl(X) = \bigcap \{t \in r | X \subseteq t\}$$

If $Cl(X) = X$ then X is said to be closed.

Frequent free itemsets An itemset X is said to be free if there is no exact rule of the form $X_1 \rightarrow X_2$ where X_1 and X_2 are distinct subsets of X. Free sets can be efficiently detected through the following property:

$$X \text{ is free } \iff \forall x \in X, sup(X) < sup(X - x)$$

Frequent essential itemsets The notion of essential itemsets has been defined recently in [13]. It is based on the notion of disjunctive rule [11, 25]. A *disjunctive rule* is of the form $X \rightarrow A_1 \vee A_2 ... \vee A_n$. Such a rule is satisfied if, every transaction that contains X contains at least one of the elements $A_1, ..., A_n$.

An itemset X is said to be essential if there is no *disjunctive rule* of the form $A_1 \rightarrow A_2 \vee \ldots \vee A_k$, where $(A_i)_{i=1..k}$ are distinct elements in X. As for free sets, they can be efficiently tested exploiting the following property:

$$X \text{ is essential } \iff \forall x \in X, sup_{dij}(X) > sup_{dij}(X - x)$$

where $sup_{dij}(X) = |\{t \in r \mid t \cap X \neq \emptyset\}|$

The predicates "being a frequent free itemset" and "being a frequent essential itemset" are anti-monotone w.r.t. set inclusion. In the following, we study the distributions of these three collections w.r.t. itemsets size.

Other concise representations based on the notion of disjunctive rules have been defined, the reader is referred to the general framework proposed in [12] for more details.

To end up, we believe that our choice of concise representations covers a fairly large range of typical cases.

3.2 Experimental protocol

For frequent itemsets, a benchmark of fourteen datasets is commonly used [18]. Most of them are real-life datasets, only two being synthetic ones, generated by the generator from the IBM Almaden Quest research group [1]. All experiments have been done on these datasets. Each dataset has been studied for many representative minimum support thresholds, from very high to very low values. For each one, frequent itemsets, frequent closed, frequent free and frequent essential itemsets, have been collected. We have studied their distribution with respect to itemsets size, i.e. the number of elements in each level (from one to the size of the largest itemsets). Moreover, we have studied the positive and *negative* borders distributions of frequent, frequent free and frequent essential itemsets [2].

To perform these tests, we used algorithms available at the FIMI website [18]. The discovery of frequent itemsets and frequent closed itemsets has been done using $FPClose$ and $FP-growth*$ algorithms from [21]. ABS [17] has been updated to find frequent free and frequent essential itemsets.

To the best of our knowledge, this work is the first one to address the understanding of datasets for frequent itemsets and other concise representations by using their negative borders.

3.3 Experimental results

In order to perform a fair comparison with [20], results given in this paper focus on the same datasets, i.e. $Chess$, $Pumsb$, $Connect$, $Pumsb*$, $Mushroom$ and $T10I4D100K$. Notations used in the sequel are reported in Table 1.

FI	frequent itemsets
FCI	frequent closed itemsets
FFI	frequent free itemsets
FEI	frequent essential itemsets

Table 1. Notations

Given a dataset and a minimum support threshold value, the Table 2 describes a typical example of our experimental results. Due to space limitations, the reader is referred to [16] for comprehensive results from which the analysis made in this paper has been performed. A wider range of minimum support threshold values and other datasets are also described in [16].

In the rest of this section, we discuss our experimental results with respect to three main axes: borders distribution of frequent itemsets, stability of borders distribution with respect to support threshold values and borders distribution of frequent free and essential itemsets.

[2] The set of closed itemsets is not closed downwards, and thus the notion of borders does not apply.

Itemset size	FI	FCI	FFI	FEI	Bd-(FI)	Bd+(FI)	Bd-(FFI)	Bd+(FFI)	Bd-(FEI)	Bd+(FEI)
1	50	27	50	50	25		25	1	25	1
2	896	338	828	828	329		397	2	397	149
3	9049	2568	7628	4240	928	1	988	34	4376	853
4	59589	13221	44096	6283	3371	9	3440	268	8519	3178
5	273069	49002	170161	1635	10118	89	10178	1343	1764	1186
6	907800	137564	456826	116	21405	439	21416	4876	36	109
7	2255159	303661	875938	1	33711	1369	33720	11963		1
8	4276852	540861	1216501		39910	3686	39910	22521		
9	6291848	787143	1231162		33890	8200	33890	31137		
10	7263312	940504	903996		21894	14804	21894	32243		
11	6626801	923310	474618		10160	21183	10160	25491		
12	4790827	740773	172688		3507	24638	3507	15326		
13	2738089	481499	41186		791	23766	791	6403		
14	1227702	250715	5787		114	18088	114	1951		
15	425896	102977	360		8	10934	8	314		
16	111726	32875	3			5085		3		
17	21328	7908				1734				
18	2757	1370				496				
19	206	145				97				
20	6	6				6				
total	37282962	5316467	5601828	13153	180161	134624	180438	153876	15117	5477

Table 2. $Chess$ **dataset,** $minsup = 30\%$

Borders distribution of frequent itemsets Consider the positive and negative borders of frequent itemsets from five datasets as given in Figure 1. First of all, we observe "bell curve" distributions for the two borders in almost all datasets. Since every distribution of positive border is feasible in theory [27], other properties should exist to explain these distributions. Moreover, the negative and positive borders seem to follow the same behavior even if the negative border is always "lower" than its corresponding positive border. From [30], we also know that the negative border may have elements just one level after the positive border. This case never occurs in our experiments for frequent itemsets.

Moreover, we denote two different behaviors of the "distance" between the two borders. For $Chess, Pumsb$ and $T10I4D100K$ (Figure 1), the borders distributions are very close, i.e. the mean of the negative border curve is only a few levels before the mean of the positive border curve. The dataset $T10I4D100K$ is different from the two others since its borders are made of small itemsets.
For datasets $Connect$, $Pumsb*$ and $Mushroom$ in Figure 1, a larger distance between the borders exists.

These simple observations will be used as predictors of the hardness of a dataset in the sequel.

Stability of borders distribution Now, we study the variation of minimum support threshold values on borders distribution of frequent itemsets. To do that, we consider $Chess$ and $Connect$ for the following minimum support threshold values: 30%, 50% and 80%. For the first minimum support threshold value, the results are in Figure 1 and for the two others, in Figure 2.

A surprising observation is that the relative position of borders distributions is *stable* w.r.t. variation of minimum support threshold values.

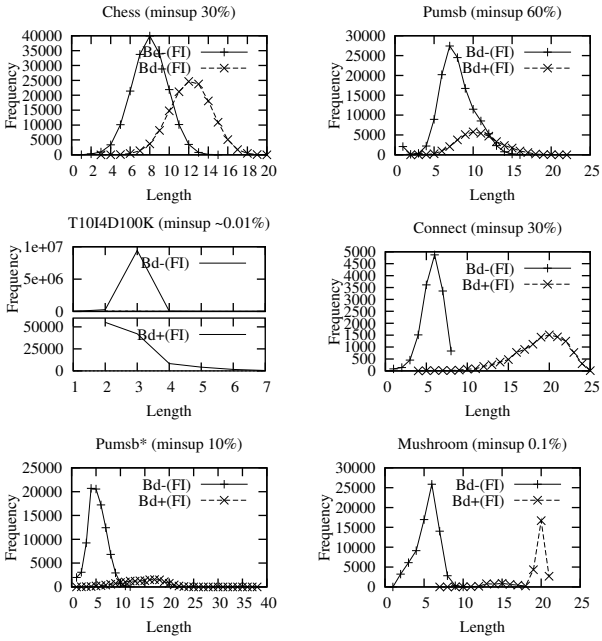

Figure 1. Borders of frequent itemsets

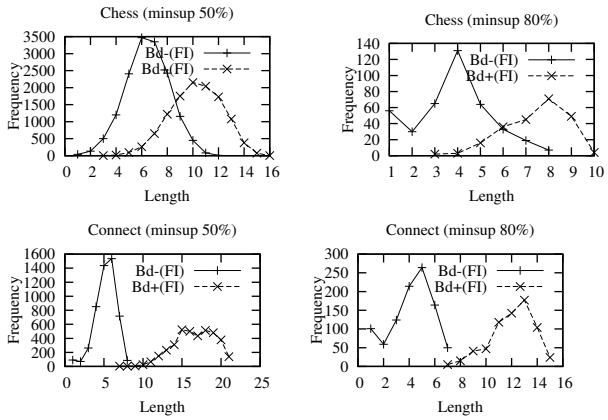

Figure 2. Borders of frequent itemsets with a different minimum support threshold

In other words, this observation suggests a kind of *global structure* for frequent itemsets borders distribution invariant to variation of minimum support threshold values.

Borders of concise representations Now, we consider the positive and negative borders of *frequent free itemsets* and *frequent essential itemsets* on $Chess$ and $Connect$ given in Figure 3 and 4.

From these two figures, one can remark that distributions of the two borders look like "bell curves". Recall that the same behavior has already been pointed out for frequent itemsets, suggesting that such kind of curves is almost independent of the considered anti-monotone predicate.

Moreover, the distance between the mean of the negative and positive borders appears to be small for each concise representation.

The same behavior has been observed in all our experiments [16].

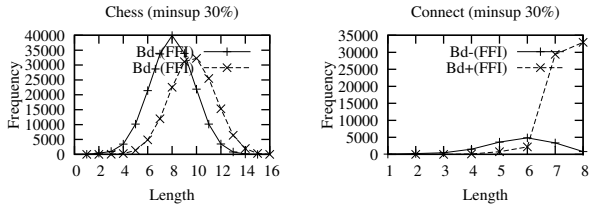

Figure 3. Borders of frequent free itemsets

3.4 Impact on algorithms performances

We focus on the discovery of *maximal frequent itemsets*, and we study the performances of implementations available at the FIMI website [18]. Let us consider results

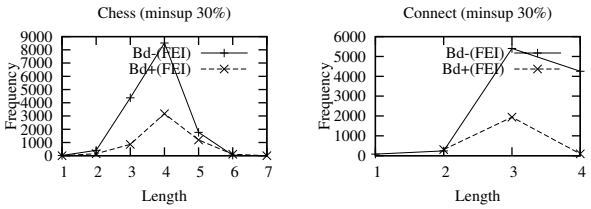

Figure 4. Borders of frequent essential itemsets

given in Figure 5 showing algorithms execution times on four datasets.

On $Chess$ dataset, for every implementation given in Figure 5 (upper-left corner), algorithms execution times increase exponentially with decreasing minimum support threshold values, whereas for $Connect$ (upper-right corner) they appear to be almost linear for $Mafia$ [9], $fp-zhu$ [21], LCM [36] and $afopt$ [26]. Moreover, recall that $Connect$ has more and longer transactions, and more items than $Chess$.

The same kind of behavior can be noticed for datasets such as $Pumsb$ and $Pumsb*$ (Figure 5). These two datasets are very similar w.r.t. the transactions and number of items, but their borders distribution is very different (Figure 1). Algorithms for $Pumsb*$ are still very effective for very low minimum support threshold, whereas for $Pumsb$, algorithms do not perform very well for relatively high minimum support threshold values.

Therefore, we deduce that *pruning strategies are much more efficient on datasets having a "large" distance between their positive and negative borders.*

A possible explanation could be obtained by looking at algorithms pruning strategies since most of them take advantage of minimal unfrequent itemsets to find maximal frequent itemsets and prune the search space.

4 Toward new classifications

Observations described in previous section lead us to devise a new classification for datasets w.r.t. borders distribution. We also intent to use these results for other data mining problems, i.e. those problems said to be "representable as sets" [30].

4.1 A new dataset classification

This new classification differs from the classification given in [20] since it takes into account both the negative border and the positive border of frequent itemsets.

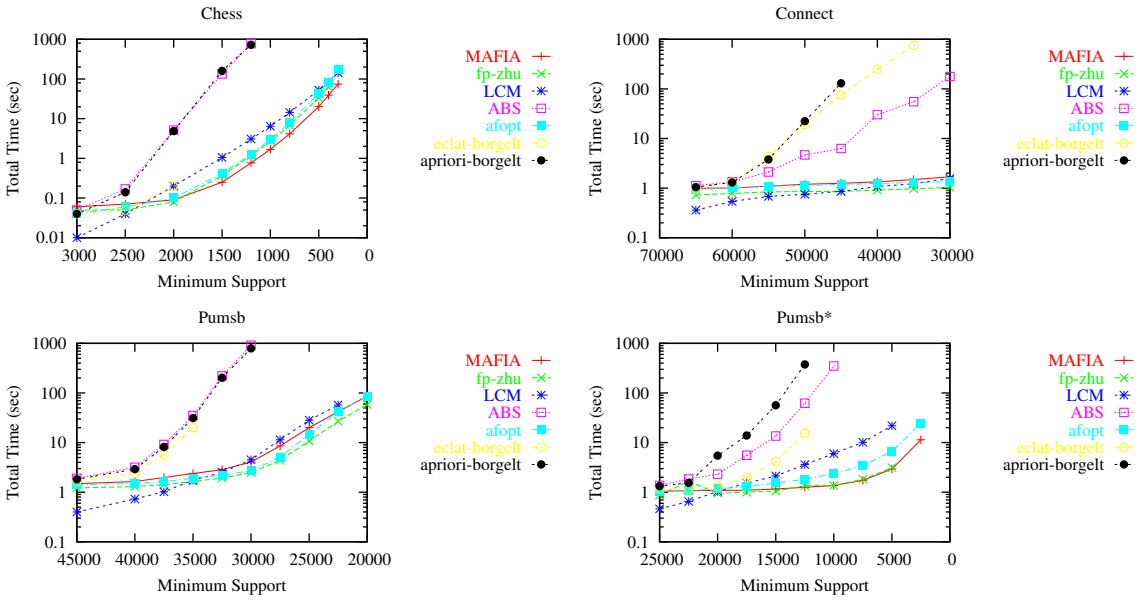

Figure 5. Algorithms performances

This classification follows from remarks done in the previous section. Its main interests are:

- a better correspondence between algorithms performances and the classification. In other words, this classification is a first attempt in order to evaluate the "hardness" of a dataset.
- a stability w.r.t. the variation of minimum support threshold.

Based on the "distance" between positive and negative borders distributions of frequent itemsets, different types of datasets have been identified. As a consequence, we introduce a new classification of datasets made of three types:

- Type I datasets are datasets where borders distributions are very close, i.e. the mean of the negative border curve is not far from the mean of the positive border curve. In other words, most of the itemsets in the two borders have approximately the same size. $Chess$ and $Pumsb$ fall into this category; such datasets can be expected to be hard for frequent itemsets mining.
- Type II datasets are datasets where there is a large distance between the two borders distributions. In other words, the itemsets in the negative border are much smaller than those of the positive border. $Connect$, $Pumsb*$ and $Mushroom$ fall into this category; in practice this type is easier than the previous one.
- Type III is a very special case of type I: the two distributions are very close, but they are concentrated in very low levels. This type allows to catch the notion of sparseness (for example $T10I4D100K$). It might be the most easy dataset type in practice.

This classification is simpler than the one presented in [20], while being very stable w.r.t. variation of minimum support threshold. In addition to classical characteristics of datasets, the "distance" between the mean of the negative and positive border distributions makes possible a better evaluation of the difficulty of a dataset.

For the two other concise representations previously described, this classification suggests that almost all datasets belong to type I or III.

4.2 Predicate classification

In the setting of this paper, we focus our analysis on datasets with respect to frequent itemsets. In our experiments, we studied three anti-monotone predicates, one for frequent itemsets, another one for frequent free itemsets and the last one for frequent essential itemsets. These three predicates exhibit very different behaviors on the same datasets (see Figure 1 to 4 on $Connect$ and $Chess$ for different minimum support threshold values).

Quite clearly, this work could be generalized to other data mining problems, i.e. those which are *representable as sets* [30]. We argue that the study of both positive and negative borders for a given anti-monotone predicate may allow us to come up with some general results.

From the previous sections, we deduced that studying the

$$\left.\begin{array}{c} R[XY] \subseteq S[UV] \\ R[XZ] \subseteq S[UW] \\ S : U \rightarrow V \end{array}\right| \Rightarrow \quad R[XYZ] \subseteq S[UVW]$$

Figure 6. An interaction between FD and IND

gap between the negative and positive borders may be very insightful to explain the behavior of algorithms and may also give some hints to guess the existence of properties associated with anti-monotone predicates. In spite of the huge amount of work done for frequent itemset mining, we are not aware of such kind of contributions. Nevertheless, we introduce in the sequel another data mining problem known to be representable as sets where such properties have been clearly identified [15].

Application to inclusion dependency mining Inclusion dependencies (IND) are fundamental semantic constraints for relational databases [28]. Let r and s be two relations over schemas R and S, and X and Y be sequences of attributes into R and S respectively. The IND $R[X] \subseteq S[Y]$ is true in (r, s) if all the values of X in r are also values of Y in s. This notion generalizes foreign keys constraints, very popular in practice.

The underlying data mining problem can be stated as follows: "Given a database, find all inclusion dependencies satisfied in this database" [23, 30, 24, 15]. From [30], the set of IND candidates can be organized in a levelwise manner; a given level, say k, corresponds to INDs whose arity is equal to k. Moreover, a partial order for INDs can be defined as follows: if i and j are two INDs, $j \preceq i$ if j can be obtained by performing the same projection on the two sides of i. For example, $R[AB] \subseteq S[EF] \preceq R[ABC] \subseteq S[EFG]$. In this setting, the predicate "being satisfied in a database" is anti-monotone with respect to $\preceq$ [30].

Consider now the well known inference rule for inclusion dependencies together with functional dependencies [14] given in Figure 6. Intuitively, consider an inclusion dependency $i = R[XAB] \subseteq S[YEF]$ where X and Y are attribute sequences and A, B, E and F are single attributes. Suppose that every IND j such that $j \preceq i$ is satisfied, and let $j_1 = R[XA] \subseteq S[YE]$ and $j_2 = R[XB] \subseteq S[YF]$ be two of them. The more $|Y|$ is large, the more Y is likely to determine E or F. In other words, i is likely to be satisfied (from inference rule of Figure 6).

From this result, one may logically expect that large INDs should never appear in the negative border, even if large INDs exist. It implies a potentially large gap between the two borders distribution, like for type II datasets for frequent itemsets.

All our experiments corroborate this hypothesis; We tested three synthetic databases built using the *chase* procedure [6]. We enforced large INDs in their positive border, until size 18. For all databases, INDs in the negative border were all of size lower than 3.

This particular behavior of the positive border of INDs justifies an algorithm based on the negative border discovery [15].

5 Conclusion and perspectives

In this paper, we have thoroughly studied datasets for problems related to frequent itemset mining. We have shown that the distribution of the negative and positive borders have an important impact on datasets classification and algorithms performances. For frequent itemsets mining, a new classification of datasets has been proposed. This work is a first step toward a better understanding of the behavior of algorithms with respect to the search space to be discovered.

This work has two main perspectives. The former is to find out theoretical foundation of "bell curves" and stability obtained for the distributions in most of our experiments. The latter is the design of adaptive algorithms with respect to dataset characteristics, i.e. changing dynamically their strategy during runtime.

References

[1] Synthetic data generation code for associations and sequential patterns. Intelligent information systems, IBM almaden research center. http://www.almaden.ibm.com/software/quest/resources/.

[2] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *SIGMOD conference, Washington*, pages 207–216. ACM Press, 1993.

[3] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *VLDB conference, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994.

[4] Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhal. Mining frequent patterns with counting inference. In *SIGKDD Explorations 2(2)*, pages 66–75, 2000.

[5] R. J. Bayardo, B. Goethals, and M. J. Zaki, editors. *FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, volume 126 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[6] C. Beeri and M. Vardi. A proof procedure for data dependencies. *Journal of the ACM*, 31(4):718–741, 1984.

[7] C. Borgelt. Efficient implementations of Apriori and Eclat. In *FIMI '03, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, November 2003.

[8] J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Free-sets: A condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery*, 7(1):5–22, 2003.

[9] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu. MAFIA: A performance study of mining maximal frequent itemsets. In *FIMI '03, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, 2003.

[10] D. Burdick, M. Calimlim, and J. Gehrke. MAFIA: A maximal frequent itemset algorithm for transactional databases. In *ICDE conference, Heidelberg, Germany*, pages 443–452. IEEE CS, 2001.

[11] A. Bykowski and C. Rigotti. A condensed representation to find frequent patterns. In *PODS'01, Santa Barbara, California, USA*. ACM, 2001.

[12] T. Calders and B. Goethals. Minimal k -free representations of frequent sets. In *PKDD*, pages 71–82, 2003.

[13] A. Casali, R. Cicchetti, and L. Lakhal. Essential patterns: A perfect cover of frequent patterns. In *DaWaK conference, Copenhagen, Denmark*, Lecture Notes in Computer Science, 2005.

[14] M. Casanova, R. Fagin, and C. Papadimitriou. Inclusion dependencies and their interaction with functional dependencies. *Journal of Computer and System Sciences*, 24(1):29–59, 1984.

[15] F. De Marchi and J.-M. Petit. Zigzag : a new algorithm for discovering large inclusion dependencies in relational databases. In *ICDM conference, Melbourne, USA*, pages 27–34. IEEE Computer Society, 2003.

[16] F. Flouvat. Experimental study of frequent itemsets datasets. Technical report, LIMOS, France, http://www.isima.fr/flouvat/papers/rr05-ExpStudyDatasets.pdf, june 2005.

[17] F. Flouvat, F. De Marchi, and J.-M. Petit. ABS: Adaptive borders search of frequent itemsets. In *FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*.

[18] B. Goethals. Frequent itemset mining implementations repository, http://fimi.cs.helsinki.fi/, 2004.

[19] B. Goethals and M. J. Zaki, editors. *FIMI '03, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA*, volume 90 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[20] K. Gouda and M. J. Zaki. Efficiently mining maximal frequent itemsets. In N. Cercone, T. Y. Lin, and X. Wu, editors, *ICDM conference, San Jose, USA*, pages 163–170. IEEE Computer Society, 2001.

[21] G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *FIMI '03, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, November 2003.

[22] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *SIGMOD Conference*, pages 1–12, 2000.

[23] M. Kantola, H. Mannila, K. J. Rih, and H. Siirtola. Discovering functional and inclusion dependencies in relational databases. *International Journal of Intelligent Systems*, 7:591–607, 1992.

[24] A. Koeller and E. A. Rundensteiner. Discovery of high-dimentional inclusion dependencies (poster). In *Poster session of ICDE conference*. IEEE Computer Society, 2003.

[25] M. Kryszkiewicz and M. Gajek. Concise representation of frequent patterns based on generalized disjunction-free generators. In *PAKDD'02, Taipei, Taiwan*, volume 2336 of *Lecture Notes in Computer Science*, pages 159–171. Springer, 2002.

[26] G. Liu, H. Lu, J. X. Yu, W. Wei, and X. Xiao. AFOPT: An efficient implementation of pattern growth approach. In *FIMI '03, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, 2003.

[27] W. A. Maniatty, G. Ramesh, and M. J. Zaki. Feasible itemset distributions in data mining: Theory and application. In *SIGMOD conference, San Diego, USA*, pages 284–295. ACM, June 2003.

[28] H. Mannila and K. J. Räihä. *The Design of Relational Databases*. Addison-Wesley, second edition, 1994.

[29] H. Mannila and H. Toivonen. Multiple uses of frequent sets and condensed representations (extended abstract). In *KDD conference,Portland, USA*, pages 189–194. AAAI Press, 1996.

[30] H. Mannila and H. Toivonen. Levelwise Search and Borders of Theories in Knowledge Discovery. *Data Mining and Knowledge Discovery*, 1(1):241–258, 1997.

[31] S. Orlando, C. Lucchese, P. Palmerini, R. Perego, and F. Silvestri. kDCI: a multi-strategy algorithm for mining frequent sets. In *FIMI '03, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, November 2003.

[32] P. Palmerini, S. Orlando, and R. Perego. Statistical properties of transactional databases. In *ACM symposium on Applied computing*, pages 515–519, New York, USA, 2004. ACM Press.

[33] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1):25–46, 1999.

[34] J. Pei, J. Han, and R. Mao. Closet: An efficient algorithm for mining frequent closed itemsets. In *SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[35] T. Uno, T. Asai, Y. Uchida, and H. Arimura. LCM: An efficient algorithm for enumerating frequent closed item sets. In *FIMI '03, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, November 2003.

[36] T. Uno, M. Kiyomi, and H. Arimura. LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets. In *FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, 2004.

[37] O. R. Zaïane, M. El-Hajj, Y. Li, and S. Luk. Scrutinizing frequent pattern discovery performance. In *ICDE*, pages 1109–1110, 2005.

[38] M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemset mining. In *SIAM International Conference on Data Mining*, 2002.