

# Efficient Discovery of Interesting Patterns Based on Strong Closedness

Mario Boley      Tamás Horváth      Stefan Wrobel

Fraunhofer IAIS, Sankt Augustin, Germany  
Dept. of Computer Science, University of Bonn, Germany

{mario.bole, tamas.horvath, stefan.wrobel}@iais.fraunhofer.de

## Abstract

Finding patterns that are interesting to a user in a certain application context is one of the central goals of Data Mining research. Regarding all patterns above a certain frequency threshold as interesting is one way of defining interestingness. In this paper, however, we argue that in many applications, a different notion of interestingness is required in order to be able to capture “long”, and thus particularly informative, patterns that are correspondingly of low frequency. To identify such patterns, our proposed measure of interestingness is based on the degree or strength of closedness of the patterns. We show that (a) indeed this definition selects long interesting patterns that are difficult to identify with frequency-based approaches, and (b) that it selects patterns that are robust against noise and/or dynamic changes. We prove that the family of interesting patterns proposed here forms a closure system and use the corresponding closure operator to design a mining algorithm listing these patterns in amortized quadratic time. In particular, for non-sparse datasets its time complexity is  $O(nm)$  per pattern, where  $n$  denotes the number of items and  $m$  the size of the database. This is equal to the best known time bound for listing ordinary closed frequent sets, which is a special case of our problem. We also report empirical results with real-world datasets.

## 1 Introduction

Most enumeration problems in data mining can be viewed as parameterized theory extraction problems, i.e., special cases of the generic problem of listing all patterns that are “interesting” with respect to a given database [8]. In frequent set mining, in particular, the database is a set of transactions, each being a subset of a fixed set of items. We are then usually looking for all sets that occur in more than a specified number of the transactions (frequency threshold). That is, interestingness for this particular theory extraction problem is defined by frequency.

A common problem with frequency as interestingness notion is that it may exclude “long”, and thus particularly informative, patterns that are correspondingly of low frequency. Mining long informative patterns thus leads to the problem that very large output sets can easily occur, essentially forcing the user to stop the algorithm after the allocated runtime is over, leaving him with an incomplete result set that will depend on the internal search ordering of the particular algorithm.

Motivated by the above inherent weakness of interestingness based on minimum frequency, in this paper we propose a new definition of interestingness expressing the *degree* or *strength* of closedness, where by closedness we mean the standard notion used in data mining or formal concept analysis. In this respect, our definition is also motivated by other theory extraction problems such as formal concept analysis [4] or core discovery of cyber-communities [7], where *closedness* is the natural, semantically reasonable interestingness notion.

We present a parameterized pattern class based on the principle of strongly closed or  $\Delta$ -closed sets. A  $\Delta$ -closed set is a set that cannot be augmented by any further item without reducing its support by at least  $\Delta$ . Thus, strongly closed sets are sets that are at the boundary of a sharp drop in frequency. As an example, a concept in formal concept analysis is  $\Delta$ -closed if all of its proper subconcepts contain less than or equal to  $\Delta$  objects from the concept. Our experimental results with different datasets clearly indicate two desirable properties of strongly closed sets in applications: (a) they are indeed able to capture *long* patterns in reasonable time, even when this problem is difficult for frequency-based approaches and (b) they are *stable*, i.e., robust against noise and/or dynamic changes in the data.

The above definition implies that strongly closed sets provide a *granularity* of ordinary closed sets [9] (these are 1-closed), so their use in addition builds on the interesting formal and practical results of the use of closed sets in frequent pattern mining. However, in

contrast to ordinary closed sets, strongly closed sets generally do not provide a lossless compact representation of frequent sets. On the other hand, we prove that they always form a closure system just as the ordinary closed sets. We use the corresponding closure operator to design a mining algorithm and prove that it lists all strongly closed sets in amortized quadratic time. In particular, for non-sparse datasets its time complexity is  $O(nm)$  per pattern, where  $n$  denotes the number of items and  $m$  the size of the database. This is equal to the best known time bound for listing ordinary closed frequent sets [11]—a notable fact as our notion subsumes the latter one. Moreover, the interestingness notion proposed in this work can be combined with frequency in a very natural way without changing the time complexity of our algorithm. Thus, it can also be considered as an alternative approach to iceberg concept lattices [10], where a different hierarchy on closed sets is defined by frequency.

In the remainder of the paper, in Section 2, we first precisely define the notion of  $\Delta$ -closed sets and demonstrate that by using strongly closed sets, it is possible to arrive at semantically meaningful and stable result sets containing long patterns, without using support threshold parameters. In Section 3, we show that the family of strongly closed sets can be generated by a closure operator that can be computed efficiently. In Section 4, we then give the algorithm enumerating strongly closed sets and investigate its computational properties. Moreover, we provide an empirical evaluation of the use of strongly closed sets on real-world data. Section 5 discusses important related work and Section 6 concludes.

All datasets involved in our experiments and the implementation of our algorithm are publicly available (for the implementation see [http://www-kd.iai.uni-bonn.de/index.php?page=people\\_details&id=16](http://www-kd.iai.uni-bonn.de/index.php?page=people_details&id=16) or the corresponding main page).

## 2 Strong Closedness as Interestingness

The primary goal of this paper is to design a semantically reasonable interestingness measure able to effectively capture long patterns. In addition, we require the patterns to be robust against noise and/or dynamic changes in the dataset, which arguably is a desirable property in applications. As starting point, we have selected *closedness*, as defined in frequent itemset mining. Beside its semantic there, closedness is also the straightforward, semantically reasonable interestingness predicate, e.g., in formal concept analysis [4], where the task is to list all maximal submatrices of a binary matrix that consist of only 1 entries, and in core discovery of cyber-communities [7], where the task is to list all maximal bipartite subgraphs of a directed graph. In

contrast to interestingness based on frequency, the likelihood of a pattern being closed does not decrease with its length. Finally, closed sets are efficiently enumerable due to their property of forming a closure system. Our interestingness measure, i.e.,  $\Delta$ -closedness, is then obtained by expressing the degree or strength of closedness.

In the rest of this section, we first give the precise definition of  $\Delta$ -closedness and then report empirical experiments demonstrating that it indeed possesses the two desirable properties mentioned above. We also note that, besides these properties,  $\Delta$ -closedness might be a semantically reasonable interestingness predicate in such applications of formal concept analysis or core discovery of cyber-communities (e.g., visualization of concept lattices, ontology engineering, analysis of social communities, recommendation systems etc.), where we are interested only in such closed patterns which have a specified distance from other closed patterns.

**2.1 Definition** In this paper we always assume  $E = \{e_1, \dots, e_n\}$  to be a finite ground set and  $\mathcal{D}$  a dataset over  $E$ , i.e.,  $\mathcal{D}$  is a finite multiset  $\{D_1, \dots, D_m\} \subseteq \mathcal{P}(E)$ , where  $\mathcal{P}(E)$  denotes the power set of  $E$ . A set  $\{e_{i_1}, \dots, e_{i_k}\}$  will sometimes be abbreviated by  $e_{i_1} \dots e_{i_k}$ . In pattern mining,  $E$  and  $\mathcal{D}$  are commonly referred to as *items* and *transaction database*, respectively. Since some of the standard problems in closed itemset mining [9] are also relevant to other application fields, like for instance formal concept analysis [4] or core discovery of cyber-communities [7], we deliberately use a neutral terminology in this paper. We define the *size* of  $\mathcal{D}$  as the sum of its transaction sizes, i.e.,  $\|\mathcal{D}\| = \sum_{i=1}^m |D_i|$ . For  $X \subseteq E$ , we denote by  $\mathcal{D}[X] = \{D \in \mathcal{D} : X \subseteq D\}$  the *support set* of  $X$  in  $\mathcal{D}$  and by  $\bar{\mathcal{D}}[X] = \mathcal{D} \setminus \mathcal{D}[X]$  the complement of the support set of  $X$ . For an integer threshold  $t \geq 0$ , a set  $F \subseteq E$  is *t-frequent* or shortly, *frequent*, if  $|\mathcal{D}[F]| \geq t$ . The family of *t-frequent* sets is denoted by  $\mathcal{F}_{t,\mathcal{D}}$ . Using the above notations, we define *strongly closed* or  $\Delta$ -*closed sets*, the central notion in this work for interestingness, as follows:

**DEFINITION 1.** For an integer  $\Delta \geq 0$ , a set  $F \subseteq E$  is  $\Delta$ -closed if  $|\mathcal{D}[F']| \leq |\mathcal{D}[F]| - \Delta$  holds for every  $F'$  with  $F \subsetneq F' \subseteq E$ .

That is, a set  $F$  is  $\Delta$ -closed if any augmentation reduces its support by at least  $\Delta$ .

Though our interestingness notion is independent of frequency, for technical reasons discussed later, we need the following notation. The family of *frequent*  $\Delta$ -closed sets is denoted by  $\mathcal{C}_{\Delta,t,\mathcal{D}}$ , i.e.,

$$\mathcal{C}_{\Delta,t,\mathcal{D}} = \{X \in \mathcal{F}_{t,\mathcal{D}} : X \text{ is } \Delta\text{-closed}\} .$$

If  $\mathcal{D}$  or  $t$  are clear from the context then they are omitted from the indices in  $\mathcal{C}_{\Delta,t,\mathcal{D}}$  and  $\mathcal{F}_{t,\mathcal{D}}$ .

The name “ $\Delta$ -closed” is justified by the fact that  $\Delta$  measures the degree of closedness. In particular, for  $\Delta = 1$  we obtain the definition of ordinary closed sets. Therefore, throughout the remainder of this article we will use these terms interchangeably. The following proposition describes further direct implications of the definition.

PROPOSITION 2.1. *For every dataset  $\mathcal{D}$  and all integers  $\Delta, t, k \geq 0$  it holds that*

- (i)  $E \in \mathcal{C}_{\Delta,0}$  ,
- (ii)  $\mathcal{C}_{|\mathcal{D}|+k,t} = \mathcal{C}_{|\mathcal{D}|,t} \subseteq \mathcal{C}_{|\mathcal{D}|-1,t} \subseteq \dots \subseteq \mathcal{C}_{0,t} = \mathcal{F}_t$  ,
- (iii)  $\forall F \subsetneq E, F \in \mathcal{C}_{\Delta,0} \Rightarrow F \in \mathcal{F}_{\Delta}$  .

We now give an example illustrating  $\Delta$ -closed sets. Consider the dataset  $\mathcal{D}$  over  $E = \{a, b, \dots, f\}$  given in

	a	b	c	d	e	f
1	0	1	1	0	1	1
2	0	0	1	1	1	1
3	1	0	0	0	1	1
4	1	1	0	0	1	1
5	1	1	1	1	0	1
6	1	1	1	0	0	0
7	1	1	1	1	0	0
8	1	1	1	1	1	0

Figure 1: example dataset

Figure 1. The set  $ae$  is 1-closed because  $ae = \bigcap \mathcal{D}[ae]$ . Since in addition  $|\mathcal{D}[ae]| = 3$ , we have  $ae \in \mathcal{C}_{1,3}$ . However, it is not 2-closed because for  $b$  we have

$$2 = |\mathcal{D}[abe]| > |\mathcal{D}[ae]| - 2 = 1 .$$

On the other hand,  $ef$  is a 2-closed set with support count 4 because

$$\begin{aligned} 2 &= |\mathcal{D}[aef]| \leq |\mathcal{D}[ef]| - 2 = 2 \\ 2 &= |\mathcal{D}[bef]| \leq |\mathcal{D}[ef]| - 2 = 2 \\ 2 &= |\mathcal{D}[cef]| \leq |\mathcal{D}[ef]| - 2 = 2 \\ 1 &= |\mathcal{D}[def]| \leq |\mathcal{D}[ef]| - 2 = 2 \end{aligned}$$

as required. One can check that for this dataset we have

$$\begin{aligned} \mathcal{C}_{1,1} &= \{\emptyset, a, ab, abc, abcd, abcde, abcdf, abcde, \dots\} \\ \mathcal{C}_{2,1} &= \{\emptyset, abcd, ef\} \\ \mathcal{C}_{3,1} &= \{\emptyset\} \end{aligned}$$

(note that  $\mathcal{D}[\emptyset] = \mathcal{D}$ ). The rest of this section is devoted to discussing two important features of strongly closed sets.

**2.2 Stability** A first distinctive feature of  $\Delta$ -closed sets is that they exhibit a certain robustness against changes in the input dataset. We first state a straightforward proposition interpreting the definition of strong closedness with respect to ordinary closedness.

PROPOSITION 2.2. *Let  $C \in \mathcal{C}_{\Delta,1,\mathcal{D}}$  and  $\mathcal{D}'$  be a dataset with  $|\mathcal{D} \setminus \mathcal{D}'| < \Delta$ . Then  $C \in \mathcal{C}_{1,1,\mathcal{D}'}$ .*

So a set  $F$  that is  $\Delta$ -closed in  $\mathcal{D}$  remains closed (i.e., 1-closed) after deleting or changing any  $\Delta - 1$  transactions from  $\mathcal{D}$ . In this sense, the feature of  $F$  to be closed in  $\mathcal{D}$  is *stable*.

In the following we describe an experiment with real-world Web data indicating that strong closedness itself possesses a similar stability, i.e., a  $\Delta$ -closed set is likely to remain  $\Delta$ -closed in face of (dynamic) changes of the dataset. The dataset used in this experiment is “Anonymous web data from www.microsoft.com” (msweb) from the UCI Machine Learning repository. It contains the areas of www.microsoft.com in 1998 as items and 38,000 randomly selected users as transactions. Each user “contains” a web site if he visited it during some fixed one week timeframe. We added noise to the data by applying the following geometric perturbation procedure to each transaction:

1. Flip an unfair coin with some fixed success probability  $p$ .
2. Stop if the flip fails, otherwise flip a fair coin to determine whether to remove or to add an element from the transaction. This element is chosen uniformly at random from the transaction respectively from its complement.
3. Go back to 1.

In the example of cyber-communities, this operation corresponds to the event that a user changes his site preferences: A few new Web sites may attract his attention while he abandons a few old ones. Furthermore, the expected transaction sizes do not differ from the original data and thus, the dataset keeps its level of density. For the cyber-community example this means that a user’s “capacity” is not expected to change drastically. A uniform perturbation operator that changes every bit with a certain probability would not satisfy these intuitions.

With this procedure we created a dataset  $\mathcal{D}'$  from the original msweb dataset  $\mathcal{D}$  using the relatively high success probability 0.9. This results in roughly 9 changes on average per transaction. We then investigated the following question: How many  $\Delta$ -closed sets from  $\mathcal{C}_{\Delta,1,\mathcal{D}}$  will also appear in  $\mathcal{C}_{\Delta,1,\mathcal{D}'}$  of the perturbed dataset? For comparison, how many frequent closed

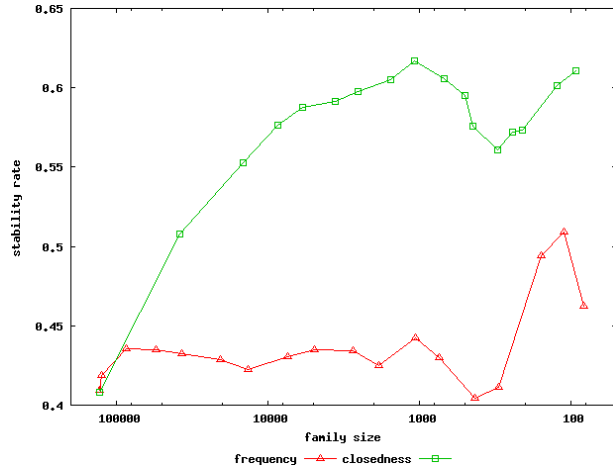


Figure 2: stability rates of closed frequent sets respectively strongly closed sets

sets from  $\mathcal{C}_{1,t,\mathcal{D}}$  will also appear in  $\mathcal{C}_{1,t,\mathcal{D}'}$ ? Since the families  $\mathcal{C}_{\Delta,t,\mathcal{D}}$  have strongly varying cardinalities for different parameter values, we are interested in the “stability rate”  $|\mathcal{C}_{\Delta,t,\mathcal{D}} \cap \mathcal{C}_{\Delta,t,\mathcal{D}'}| / |\mathcal{C}_{\Delta,t,\mathcal{D}}|$ , i.e., the fraction of closed sets “surviving” the perturbation, instead of absolute numbers. The averaged results over 50 repetitions of this experiment are presented in Figure 2. It shows the stability rates  $|\mathcal{C}_{\Delta,t,\mathcal{D}} \cap \mathcal{C}_{\Delta,t,\mathcal{D}'}| / |\mathcal{C}_{\Delta,t,\mathcal{D}}|$  (y-axis) against the cardinalities  $|\mathcal{C}_{\Delta,t,\mathcal{D}}|$  using a reversed logarithmic scale (x-axis). In particular it depicts the families  $\mathcal{C}_{\Delta,1,\mathcal{D}}$  for various values of  $\Delta$  between 1 and 256 (boxes) and the families  $\mathcal{C}_{1,t,\mathcal{D}}$  for values of  $t$  between 1 and 640 (triangles). As expected, one can observe better stability rates for shrinking families corresponding to greater values of  $\Delta$  and  $t$ . It is notable that the recalls do not increase anti-monotonically to the decreasing family sizes but show some fluctuations. For family sizes below 100 the behavior becomes a bit discontinuous. Most importantly, for cardinalities  $l \approx 100,000$  and lower we can always find values for  $\Delta$  such that the resulting family  $\mathcal{C}_{\Delta,1,\mathcal{D}}$  is roughly of cardinality  $l$  and has members that are more resistant to data changes when compared to the members of the family  $\mathcal{C}_{1,t,\mathcal{D}}$  with cardinality closest to  $l$ .

**2.3 Pattern Complexity** Another important reason to consider strongly closed sets is that they often contain rather complex patterns even for large values of  $\Delta$ , while there are only a small number of patterns in the output. To achieve the same degree of output reduction using only a frequency threshold, one usually ends up with less interesting or even only trivial patterns. Again the msweb dataset serves as an illustration. Figure 3 shows a small excerpt from the  $\Delta$ -closed

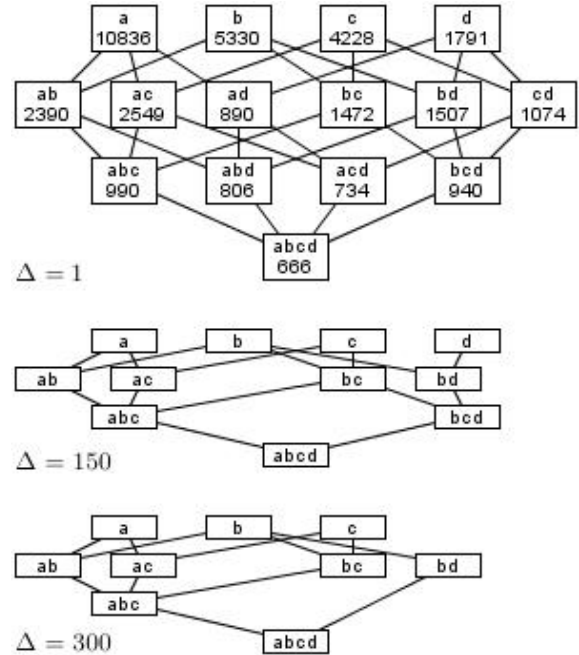


Figure 3: excerpts from closed set lattices of “msweb” for different  $\Delta$

set lattice that is successively reduced by increasing  $\Delta$ . The semantic of the items is a: ‘free downloads’, b: ‘isapi’, c: ‘Windows Family of OSs’, and d: ‘Win95 support’. The numbers in the first graph below the sets denote their support count. Note that  $abcd$  is a member of  $\mathcal{C}_{300,1}$  while its subsets  $d$ ,  $ad$ ,  $cd$ ,  $abd$ ,  $acd$ , and  $bcd$  are not. Output families resulting only from a frequency threshold can never possess such a constellation because of the anti-monotonicity of support. In this sense, a strength threshold allows more interesting output families to emerge. This is even more evident by the results presented in Figure 4. Here, the dataset “chess” was used from the workshop on Frequent Itemset Mining Implementations (FIMI) where it acted as a benchmark dataset [6]. In this dataset the transactions are a collection of chess endgame positions and the items represent different chess pieces together with their board coordinates. Its main characteristic is that, while it contains only 75 items and 3179 transactions, it is very dense and contains a huge number of closed sets (see Section 4.3). The figure shows the largest closed set family with ten or less elements that results from a frequency threshold (upper half) and the largest such family resulting from a strength threshold (lower half). While the longest pattern among the 10 most frequent 1-closed sets has only length 3, there is a pattern of length 21 among the 10 most strongly closed sets.

$\mathcal{C}_{1,3169}$	$ \mathcal{D}[C] $
$\{\}$	(3197)
$\{29\}$	(3181)
$\{29,52\}$	(3170)
$\{29,52,58\}$	(3169)
$\{29,58\}$	(3180)
$\{40\}$	(3170)
$\{40,58\}$	(3169)
$\{52\}$	(3185)
$\{52,58\}$	(3184)
$\{58\}$	(3195)
$\mathcal{C}_{197,1}$	
$\{5,7,29,34,36,40,48,52,56,58,60,62,66\}$	(2244)
$\{3,5,7,9,29,34,36,40,48,52,56,58,60,62,66\}$	(1849)
$\{3,5,7,9,17,29,34,36,40,48,52,56,58,60,62,66\}$	(1442)
$\{3,5,7,9,23,25,29,31,34,36,40,42,48,52,56,58,60,62,64,66,72\}$	(906)
$\{3,5,7,9,25,29,34,36,40,48,52,56,58,60,62,66\}$	(1643)
$\{3,5,7,9,25,27,29,34,36,40,48,52,56,58,60,62,66\}$	(1145)
$\{3,5,7,9,25,29,34,36,40,48,52,56,58,60,62,66,74\}$	(1252)
$\{5,7,25,29,34,36,40,48,52,56,58,60,62,66\}$	(2035)

Figure 4: top closed sets of *chess* w.r.t. frequency (top) and strength (bottom)

### 3 Closure Operator

In this section we show that for all datasets  $\mathcal{D} \subseteq \mathcal{P}(E)$  and every integer  $\Delta \geq 0$ ,  $\mathcal{C}_{\Delta,0,\mathcal{D}}$  is a *closure system*. That is,  $\mathcal{C}_{\Delta,0,\mathcal{D}}$  always contains  $E$  and is closed under intersection; or equivalently, there is a *closure operator*  $\sigma : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$  that *generates*  $\mathcal{C}_{\Delta}$  (for the rest of this section, we omit the indices 0 and  $\mathcal{D}$ ), i.e., its fixpoints are exactly the strongly closed sets ( $F \in \mathcal{C}_{\Delta} \Leftrightarrow \sigma(F) = F$ ). The mapping  $\sigma$  is called closure operator if it is

- extensive:  $\forall X \subseteq E, X \subseteq \sigma(X)$ ,
- monotone:  $\forall X, Y \subseteq E, X \subseteq Y \Rightarrow \sigma(X) \subseteq \sigma(Y)$ ,
- and idempotent:  $\forall X \subseteq E, \sigma(X) = \sigma(\sigma(X))$ .

Given a closure operator there are several listing algorithms (e.g., [3, 5]) that enumerate the corresponding closure system using explicit closure computations. They have in common that their listing strategy is efficient, i.e., has polynomial delay, as long as the given closure operator is computed efficiently.

In the following we construct such an operator in two steps. First we define a preclosure operator  $\hat{\sigma}_{\Delta}$ , i.e., an extensive and monotone mapping, and show that  $\hat{\sigma}_{\Delta}$  does already generate the family of strongly closed sets. In a second step we transform this preclosure operator in a straightforward way to a proper closure operator  $\sigma_{\Delta}$ . We start with the definition of  $\hat{\sigma}_{\Delta}$ .

DEFINITION 2. Define  $\hat{\sigma}_{\Delta} : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$  by

$$\hat{\sigma}_{\Delta} : F \mapsto \{x \in E : |\mathcal{D}[F \cup \{x\}]| > |\mathcal{D}[F]| - \Delta\} .$$

This definition already possesses some of the desired properties: The defined mapping is extensive and it can be computed in time  $O(\|\mathcal{D}[F]\|)$  by a simple iteration over all transactions in  $\mathcal{D}[F]$ , during which  $|\mathcal{D}[F \cup \{x\}]|$  can be computed for all  $x \in E \setminus F$ . Moreover, its fixpoints are exactly the  $\Delta$ -closed sets, i.e., it does indeed generate  $\mathcal{C}_{\Delta}$ . This is shown in the following lemma.

LEMMA 3.1.  $F \in \mathcal{C}_{\Delta} \Leftrightarrow \hat{\sigma}_{\Delta}(F) = F$ .

*Proof.* Let  $F$  be a fixpoint of  $\hat{\sigma}_{\Delta}$ . Then

$$(3.1) \quad \forall x \in E \setminus F, |\mathcal{D}[F \cup \{x\}]| \leq |\mathcal{D}[F]| - \Delta .$$

Assume there is an  $F' \supsetneq F$  with  $|\mathcal{D}[F']| > |\mathcal{D}[F]| - \Delta$ . Let  $x \in F' \setminus F$ . It follows by the anti-monotonicity of support that  $|\mathcal{D}[F \cup \{x\}]| \geq |\mathcal{D}[F']| > |\mathcal{D}[F]| - \Delta$ , which contradicts (3.1). Hence,  $F$  is  $\Delta$ -closed.

Conversely, suppose  $F \neq \hat{\sigma}_{\Delta}(F)$ . Then, as  $\hat{\sigma}_{\Delta}$  is extensive, there must be an  $x \in \hat{\sigma}_{\Delta}(F) \setminus F$ . Then  $|\mathcal{D}[F \cup \{x\}]| > |\mathcal{D}[F]| - \Delta$  and thus  $F \notin \mathcal{C}_{\Delta}$ .  $\square$

In general, however,  $\hat{\sigma}_{\Delta}$  is not idempotent and thus not a closure operator. This can be observed for instance in the example in Figure 1, where for the set  $\{a\}$  we have

$$\hat{\sigma}_2(a) = ab \neq abc = \hat{\sigma}_2(ab) = \hat{\sigma}_2(\hat{\sigma}_2(a)) .$$

Although  $\hat{\sigma}_{\Delta}$  is not idempotent, in the following lemma we show that it is a preclosure operator.

LEMMA 3.2.  $\hat{\sigma}_{\Delta}$  is a preclosure operator.

*Proof.* Extensivity is an easy implication of the definition. In order to show monotonicity let  $F' \subseteq F \subseteq E$  and  $x \in E$ . Now suppose  $x \notin \hat{\sigma}_{\Delta}(F)$ , i.e.,

$$(3.2) \quad |\mathcal{D}[F]| - |\mathcal{D}[F \cup \{x\}]| \geq \Delta .$$

Then it follows for the support of  $F' \cup \{x\}$  that

$$\begin{aligned} |\mathcal{D}[F' \cup \{x\}]| &= |\mathcal{D}[F']| - |\mathcal{D}[F'] \setminus \mathcal{D}[\{x\}]| \\ &\leq |\mathcal{D}[F']| - |\mathcal{D}[F] \setminus \mathcal{D}[\{x\}]| \\ &= |\mathcal{D}[F']| - (|\mathcal{D}[F]| - |\mathcal{D}[F \cup \{x\}]|) \end{aligned}$$

and because of (3.2)

$$\leq |\mathcal{D}[F']| - \Delta .$$

This implies that  $x \notin \hat{\sigma}_{\Delta}(F')$  and consequently  $\hat{\sigma}_{\Delta}(F') \subseteq \hat{\sigma}_{\Delta}(F)$ .  $\square$

As  $\hat{\sigma}_{\Delta}$  is a preclosure operator over a finite domain there is a canonical way of turning it into a closure operator: Since for all  $F \subseteq E$  the sequence

$F, \hat{\sigma}_\Delta(F), \hat{\sigma}_\Delta(\hat{\sigma}_\Delta(F)), \dots$  is monotone and bounded by  $E$ , it has a fixpoint. Thus, assigning this fixpoint to  $F$  is a well-defined operation, which, in addition, is idempotent. Moreover, it is easy to see that the thus attained operation inherits extensivity, monotonicity, and the fixpoints from  $\hat{\sigma}_\Delta$ .

**THEOREM 3.1.** Define  $\sigma_\Delta : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$  by

$$\begin{aligned} \hat{\sigma}_\Delta^1 &: F \mapsto \hat{\sigma}_\Delta(F) \\ \hat{\sigma}_\Delta^{i+1} &: F \mapsto \hat{\sigma}_\Delta(\hat{\sigma}_\Delta^i(F)) \\ \sigma_\Delta &: F \mapsto \hat{\sigma}_\Delta^k(F), k = \min\{i: \hat{\sigma}_\Delta^i(F) = \hat{\sigma}_\Delta^{i+1}(F)\} . \end{aligned}$$

Then  $\sigma_\Delta$  is a closure operator that generates  $C_\Delta$ , i.e., for all  $F \subseteq E$  it holds that  $F \in C_\Delta \Leftrightarrow F = \sigma_\Delta(F)$ .

Note that for a given  $F \subseteq E$ , the value  $k$  of the closure  $\sigma_\Delta(F)$  as defined in Theorem 3.1 is bounded by  $|E \setminus F|$ . This is because  $\sigma_\Delta(F) \subseteq E$  and in every iteration  $i < k$ , at least one element is added, i.e.,  $|\hat{\sigma}_\Delta^i(F) \setminus \hat{\sigma}_\Delta^{i-1}(F)| \geq 1$ . We call the iteration length  $k$  the *stair number* of  $F$ . So computing  $\sigma_\Delta$  by an iterative application of  $\hat{\sigma}_\Delta$  has time complexity  $O(k \|\mathcal{D}[F]\|) = O(|E \setminus F| \|\mathcal{D}[F]\|)$ . Advantages and disadvantages of this trivial implementation of  $\sigma_\Delta$  and other algorithmic issues are discussed in the next section.

## 4 Algorithms

In this section we first give an efficient algorithm computing  $\sigma_\Delta$  and then integrate it into a generic algorithm listing the closed sets of a closure operator. As mentioned earlier strong closedness can easily be combined with frequency. Thus, we directly incorporate it into our listing algorithm.

**4.1 Closure Computation** As stated in Section 3, the closure  $\sigma_\Delta(F)$  of a set  $F \subseteq E$  with stair number  $k$  can be computed in time  $O(k \|\mathcal{D}[F]\|)$  by iteratively computing the preclosure  $\hat{\sigma}_\Delta$ . Algorithm 1 almost exactly implements this naive strategy with the difference that it scans the dataset column-wise instead of row-wise. While this modification does not affect the computation time, it sometimes allows to compute more than one iteration of the preclosure operator during a single iteration of the outer loop.

For real-world datasets, Algorithm 1 turned out to be effective (see Section 4.3). The reason is that the number of iterations of the outer loop  $k$  was very small (in average close to 1) for the vast majority of closure computations. However, it is easy to construct worst-case examples for which  $k = \Omega(|E|)$  and thus the overall complexity becomes  $\Theta(|E \setminus F| \|\mathcal{D}[F]\|)$ , i.e., the worst-case bound is sharp. For instance, we can construct for any positive integer  $n$  a dataset  $\mathcal{D}_n = \{D_1, \dots, D_{n+1}\}$

---

### Algorithm 1 Compute Closure

---

Input : set  $F \subseteq E$ , integer  $\Delta > 0$   
 Require:  $\mathcal{D}$  is a dataset over  $E$  and  $\mathcal{D}' = \mathcal{D}[F]$   
 Output:  $C$  equal to  $\sigma_\Delta(F)$

1.  $C \leftarrow F$
  2. **repeat**
  3.   **for all**  $e \in E \setminus C$  **do**
  4.     **if**  $|\bar{\mathcal{D}}'[\{e\}]| < \Delta$  **then**
  5.        $C \leftarrow C \cup \{e\}; \mathcal{D}' \leftarrow \mathcal{D}'[\{e\}]$
  6.   **until**  $\mathcal{D}'$  was not changed during step 5
  7. **return**  $C$
- 

on elements  $\{e_1, \dots, e_n\}$  with  $D_i = \{e_i, \dots, e_n\}$  for  $1 \leq i \leq n$  and  $D_{n+1} = \emptyset$ . This results in an incidence matrix of the following form:

$$\begin{array}{cccc} 1 & \dots & 1 & 1 \\ 0 & 1 & \dots & 1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 \end{array}$$

For  $\mathcal{D}_n$ ,  $\Delta = 1$ , and  $F = \emptyset$ , Algorithm 1 will only select one element per iteration, namely  $e_{n-i+1}$  in iteration  $i$ . Until the solution  $\sigma_\Delta(\emptyset) = E$  is completed, the algorithm will run through  $n$  iterations where iteration  $i$  takes time  $(n-i)(|\mathcal{D}_n| - i)$ . Thus, the overall running time for these examples is in  $\Theta(n^3)$ .

It is a natural question, whether there is an algorithm computing  $\hat{\sigma}_\Delta$  that has a better worst-case bound, and in fact, such an algorithm exists—at least for non-sparse datasets. We can regard the input dataset as bipartite graph  $G_{\mathcal{D}}$  with vertices  $\mathcal{D} \cup E$  and edges  $Z = \{\{e, D\} : e \notin D\}$ , i.e., there is an edge for every zero in the incidence matrix of  $\mathcal{D}$ . In order to compute the closure  $\sigma_\Delta(\emptyset)$  of the empty set, a traversal of  $G_{\mathcal{D}}$  can be used as follows:

1. starting from all elements  $e \in E$  having degree less than  $\Delta$  perform a traversal of  $G_{\mathcal{D}}$ :
  - if the current vertex is an element  $e \in E$ , add  $e$  to  $C$ , delete all edges incident to  $e$ , and visit all prior neighbors of  $e$
  - if the current vertex is an element  $D \in \mathcal{D}$ , delete all edges incident to  $D$ , and visit all prior neighbors of  $D$  that have degree less than  $\Delta$
2. return  $C$

For other closure computations we consider the corresponding reduced graph. Since this algorithm successively deletes edges that correspond to zeros in the incidence matrix, we call it *Zero-Elimination* algorithm. Let  $\|\mathcal{D}\|_0 = \sum_{i=1}^m |E \setminus D_i|$  denote the sum of all transaction complements of  $\mathcal{D}$ , i.e., the number of zero entries in the incidence matrix representation of  $\mathcal{D}$ . Clearly, the size of the graph  $G_{\mathcal{D}}$  is bounded by  $\|\mathcal{D}\|_0$  and it can also be computed in this time. Since a depth-first traversal of a graph  $G$  can be implemented to run in time linear in the size of  $G$ , Zero-Elimination has a worst-case complexity of  $O(\|\mathcal{D}\|_0)$ . We refer to Appendix A for a more realistic in-place formulation of this algorithm and the proof of the following theorem.

**THEOREM 4.1.** *Let  $\mathcal{D}$  be a dataset over  $E$ . Given  $F \subseteq E$ ,  $\mathcal{D}[F]$  represented by the complements of its transactions, and  $\Delta \leq m$ , the closure  $\sigma_{\Delta}(F)$  can be computed with time and space  $O(\|\mathcal{D}[F]\|_0)$ .*

For non-sparse datasets the worst-case complexity of the Zero-Elimination algorithm is better than that of Algorithm 1 by a factor  $k$ . In practice, however, we observed that Algorithm 1 strictly outperformed the Zero-Elimination algorithm. This has two reasons: Most benchmark datasets are relatively sparse, and  $k$  is on average very small (see Section 4.3). For that reason we use Algorithm 1 for closure computations in the subsequent discussion.

**4.2 Integration into Listing** By Theorem 3.1, the family  $\mathcal{C}_{\Delta,0,\mathcal{D}}$  of  $\Delta$ -closed sets is a closure system for any  $\Delta$  and  $\mathcal{D}$ . We make use of this fact and list the elements of  $\mathcal{C}_{\Delta,t,\mathcal{D}} = \mathcal{C}_{\Delta,0,\mathcal{D}} \cap \mathcal{F}_{t,\mathcal{D}}$  for any  $t \geq 0$  with the following general technique: (i) Using some *generic* algorithm that lists the closed sets of a closure operator, enumerate an (as small as possible) subset of  $\mathcal{C}_{\Delta,0,\mathcal{D}}$  that contains  $\mathcal{C}_{\Delta,t,\mathcal{D}}$  and (ii) print those sets that are frequent. Among the generic algorithms solving the first task, we have considered the algorithms of Ganter and Reuter [3] and of Gély [5].

The algorithm of Ganter and Reuter lists the closed sets with delay<sup>1</sup>  $O(|E| T)$  resulting in a total time of  $O(|E| |C| T)$  and  $O(|E| + S)$  space, where  $E$  denotes the ground set and  $T$  and  $S$  the time respectively space complexities of computing the closure operator. The algorithm is based on a total order (reverse lexicographic) on the power set of the ground set  $E$ . The closed sets are listed increasingly with respect to this total order. For this feature, however, it is difficult to combine this

algorithm with techniques pruning infrequent  $\Delta$ -closed sets.

In contrast, this is not a problem with Gély's algorithm, which essentially traverses the closed sets in a DFS-order realized by a "divide and conquer" formulation. While it has a worse worst-case delay and space complexity of  $O(|E|^2 T)$  respectively  $O(|E| + S)$ , it has the same total time complexity of  $O(|E| |C| T)$ . Moreover, it is easy to understand and implement. In a nutshell, the algorithm recursively lists all closed sets containing an element, say  $a$ , and then all closed sets not containing  $a$ . This listing strategy can indeed be combined with pruning in the obvious way: if a closed set is infrequent then do not expand it. More precisely, let  $C \subseteq E$  be the last closed set computed by the algorithm and  $N \subseteq E$  be the set that must be disjoint with every closed set derived from  $C$ . The algorithm works as follows:

1. select an element  $a$  from  $X = E \setminus (C \cup N)$  if  $X \neq \emptyset$ ; otherwise return
2. compute the closure  $C'$  of  $C \cup \{a\}$
3. if  $C'$  is frequent and disjoint with  $N$  then print  $C'$  and list recursively the closed supersets of  $C'$ , which are disjoint with  $N$
4. lists recursively the closed supersets of  $C$ , which are disjoint with  $N \cup \{a\}$ .

One can show that the above algorithm is correct and complete, i.e., it lists precisely  $\mathcal{C}_{\Delta,t,\mathcal{D}}$ . Furthermore, the listing is non-redundant, i.e., each  $\Delta$ -closed frequent set is listed only once. The generic algorithm supports another simple still powerful optimization strategy. Since the closure is computed iteratively by augmenting the current set with one new element in each step (see loop 3-5 in Algorithm 1), the closure computation algorithm can immediately be aborted if an element of  $N$  has been added to the closure.

The pseudo code of the algorithm we have implemented is given in Algorithm 2. The algorithm combines the generic listing algorithm with the closure function algorithm as described above. Technically, we consider some total order on the ground set  $E$  and select the next augmenting element by this order (see lines 1-2 of DFS( $n$ )). Combining the complexities of the generic listing algorithm with that of the Zero-Elimination algorithm given in Theorem 4.1 and the complexity of Algorithm 1 we have the following result.

**THEOREM 4.2.** *Given a dataset  $\mathcal{D}$  over  $E$  and thresholds  $\Delta, t$ , the family  $\mathcal{C}_{\mathcal{D},\Delta,t}$  of  $\Delta$ -closed  $t$ -frequent sets can be listed with total time*

$$O(|E| |\mathcal{C}_{\Delta,t,\mathcal{D}}| \min\{\|\mathcal{D}\|_0, |E| \|\mathcal{D}\|\}) ,$$

<sup>1</sup>The delay of a listing algorithm  $\mathcal{A}$  is defined as the maximum number of steps  $\mathcal{A}$  performs between printing two consecutive elements of its output respectively before printing the first and after printing the last element.

---

**Algorithm 2** List Strongly Closed Frequent Sets

---

Input : dataset  $\mathcal{D}$  over  $E$ ,  
strength parameter  $\Delta$ ,  
and frequency threshold  $t$   
Output: family  $\mathcal{C}_{\Delta,t,\mathcal{D}}$  of  $\Delta$ -closed  $t$ -frequent sets

**main:**

1. **global variables**  $C, \leftarrow \emptyset, \mathcal{D}' \leftarrow \mathcal{D}$
2.  $\text{closure}(\emptyset)$
3. **if**  $|\mathcal{D}'| \geq t$  **then**
4.   **print**  $C$
5.   DFS( $\emptyset$ )

**closure( $N$ ):**

1. **repeat**
2.   **for all**  $e \in E \setminus C$  **do**
3.     **if**  $|\mathcal{D}'[\{e\}]| < \Delta$  **then**
4.        $C \leftarrow C \cup \{e\}; \mathcal{D}' \leftarrow \mathcal{D}'[\{e\}]$
5.     **if**  $e \in N$  **then return**
6. **until**  $\mathcal{D}'$  was not changed during step 4

**DFS( $N$ ):**

1. **if**  $\exists j, e_j \in E \setminus (C \cup N)$  **then**
  2.    $j^* \leftarrow$  minimal such  $j$
  3. **else return**
  4.    $C \leftarrow C \cup \{e_{j^*}\}, \mathcal{D}' \leftarrow \mathcal{D}'[\{e_{j^*}\}]$
  5.    $\text{closure}(N)$
  6. **if**  $|\mathcal{D}'| \geq t$  **and**  $C \cap N = \emptyset$  **then**
  7.   **print**  $C$
  8.   DFS( $N$ )
  9.   undo all changes done to  $C$  and  $\mathcal{D}'$  since line 5
  10. DFS( $N \cup \{e_{j^*}\}$ )
- 

*delay*

$$O(|E|^2 \min\{\|\mathcal{D}\|_0, |E| \|\mathcal{D}\|\}) ,$$

*and space*

$$O(|E| + \min\{\|\mathcal{D}\|_0, \|\mathcal{D}\|\}) .$$

Thus, for datasets that are neither sparse nor dense, i.e., datasets  $\mathcal{D}$  with  $\|\mathcal{D}\|_0 \in O(\|\mathcal{D}\|)$  the above total time bound is equal to  $O(|E| \|\mathcal{D}\|_0 |\mathcal{C}_{\mathcal{D},\Delta,t}|)$ . This is equal to the best known theoretical time bound of a closed frequent itemset mining algorithm, namely the LCM-algorithm [11]. This is a notable fact as the family of  $\Delta$ -closed (frequent) sets is a generalization of the usual closed (frequent) itemsets. Whether this bound can also be achieved for sparse datasets is an open question.

**4.3 Experiments** In this section we present some experimental results of mining strongly closed sets on three real-world datasets. Besides the “msweb”

and “chess” datasets introduced in Section 2.2 and Section 2.3, respectively, in our experiments we have also used the datasets “mushroom”, “retail”, “T1014D100K”. They are available from the FIMI repository and contain 119 items and 8125 transactions (*mushroom*), 16470/88163 (*retail*), and 1000/100000 (*T1014D100K*).

The implementation used in this empirical study is not sophisticated. As such it is not aimed towards competing with the highly elaborated closed frequent itemset mining implementations for the special case of  $\Delta = 1$ . Instead it is meant to a) show the influence of the strength parameter on the number of strongly closed sets and to b) demonstrate that the algorithm indeed scales linearly in the number of strongly closed sets. In order to improve its practical performance, standard optimization techniques like *occurrence deliver* and *anytime database reduction* (see for instance [11]) can be applied.

In Table 1 we present the results of our study. On all five datasets we experienced a sharp exponential drop of the number of strongly closed sets with growing strength threshold. Moreover, the running time of the algorithm was linear in the number of sets produced. For all performed listing tasks the average number of iterations of the outer loop during closure computation was very close to 1 (for example around 1.00011 for msweb, 1.00503 for chess, and 1.01 for mushroom) and the maximum experienced numbers were 3 (msweb), 4 (mushroom), and 6 (chess). Thus, Algorithm 1 computed the closures quasi in linear time with a factor close to 1. This is much better than the Zero-Elimination algorithm does perform on these datasets. Table 2 shows the number of closure computations that were called for chess. The numbers emphasize the importance of the early abort check in Line 5 of the closure procedure within Algorithm 2 and of an overall efficient implementation of  $\sigma_\Delta$ .

## 5 Related Work

**$\delta$ -Tolerant Closed Frequent Itemsets** The interestingness notion considered in this paper is defined with respect to an additive *absolute* value (i.e.,  $\Delta$ ). In some of the applications, however, one might be interested in *relative* strength of frequent sets. In Figure 5 we give an example motivating this notion. It shows the family  $\mathcal{C}_0$  based on a fictitious dataset over elements  $\{a, b, c\}$  containing 9089 times  $ab$ , nine times  $ac$ , and

$\Delta$	100	50	20	10	8
#clos	15380	740893	$22 \cdot 10^6$	$136 \cdot 10^6$	$224 \cdot 10^6$

Table 2: closure computations on “chess”

$\Delta$	msweb		chess		mushroom		retail		T1014D100K	
	$ \mathcal{C}_{\Delta,1} $	time	$ \mathcal{C}_{\Delta,1} $	time	$ \mathcal{C}_{\Delta,1} $	time	$ \mathcal{C}_{\Delta,1} $	time	$ \mathcal{C}_{\Delta,1} $	time
300	75	0s	0	0s	220	1s	191	10s	673	36s
200	117	0s	7	0s	717	2s	405	19s	1024	42s
100	284	1s	834	3s	2988	6s	1329	62s	2592	55s
50	652	2s	64082	117s	9469	11s	4129	174s	8018	75s
20	1948	3s	$3 \cdot 10^6$	41m	36117	21s	14531	497s	31814	135s
10	4487	5s	$19 \cdot 10^6$	3.5h	71050	26s	35203	15m	75820	218s
5	10841	8s	$91 \cdot 10^6$	10h	119710	32s	91350	27m	177661	328s

Table 1: empirical study demonstrating linear scaling in the number of strongly closed sets

once each of  $a$  and  $abc$ . In a relative sense the set  $ac$  is “much more closed” than the set  $a$  because augmenting  $a$  to  $ab$  would reduce the support count by a factor of only 0.001 while augmenting  $ac$  to  $abc$  would reduce the support count by 0.9. However, there is no value of  $\Delta$  such that  $\mathcal{C}_\Delta$  contains  $ac$  but not  $a$ .

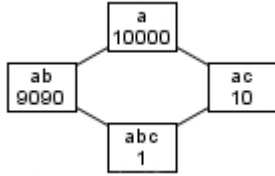


Figure 5: example motivating relative strength

The notion of relative strength has already been investigated in [2], referred to as  $\delta$ -tolerant closed frequent itemsets. We recall the definition from [2] in a slightly different, but equivalent form. Let  $E$  and  $\mathcal{D}$  be again a ground set and a dataset over  $E$ , respectively. For a real number  $\delta \in (0, 1]$ , a set  $F$  is  $\delta$ -tolerant closed if

$$|\mathcal{D}[F']| < \delta \cdot |\mathcal{D}[F]|$$

holds for every  $F'$  satisfying  $F \subsetneq F' \subseteq E$ . The notion can be combined with the frequency property in the standard way; for an integer frequency threshold  $t \geq 0$ , the family of  $\delta$ -tolerant closed frequent itemsets is denoted by  $\mathcal{C}_{\delta,t,\mathcal{D}}^{\text{rel}}$ .

In [2], the authors discuss the relationship between the families of  $\delta$ -tolerant closed frequent and closed frequent itemsets; a set  $F \subseteq E$  is closed frequent if and only if it is 1-tolerant closed frequent, i.e.,  $F \in \mathcal{C}_{1,t,\mathcal{D}}^{\text{rel}}$ . Since  $\mathcal{C}_{\delta_1,t,\mathcal{D}}^{\text{rel}} \subseteq \mathcal{C}_{\delta_2,t,\mathcal{D}}^{\text{rel}}$  holds for every  $0 < \delta_1 \leq \delta_2 \leq 1$ , each  $\delta$ -tolerant closed frequent itemset is also a closed frequent itemset for every  $\delta \in (0, 1]$ . One might ask, whether for any  $\delta \in (0, 1]$ ,  $\delta$ -tolerant closed frequent sets form a closure system and thus, a sublattice of the lattice of closed sets. The answer to this question is negative; no generating closure operator exists for these families. To see this consider again the example in Figure 5 and assume that there is a closure operator

$\sigma'_\delta$  that generates  $\mathcal{C}_{1/2,0,\mathcal{D}}^{\text{rel}} = \{ab, ac, abc\}$ . Then  $\sigma'_\delta(a)$  must be either  $ab$  or  $ac$ . But both options would violate monotonicity of  $\sigma'_\delta$ : In case  $\sigma'_\delta(a) = ab$  we have  $a \subseteq ac$  but  $\sigma'_\delta(a) = ab$  is not a subset of  $\sigma'_\delta(ac)$  which has to be  $ac$  because of extensivity. In case  $\sigma'_\delta(a) = ac$  the argument is symmetric. Thus, the listing approach based on explicit closure computations we used in this work cannot be applied to enumerate  $\mathcal{C}'_\delta$ .

**$\delta$ -Free Sets** The related notion of *free sets* or  *$\delta$ -free sets* [1] is based on the concept of  $\delta$ -strong rules. Although  $\delta$ -strong rules are also defined by bounding the difference between the support count of a set and that of its supersets,  $\delta$ -freeness and  $\Delta$ -closedness are incomparable properties with respect to entailment. More precisely, let  $E$  be a ground set and  $\mathcal{D}$  be a dataset over  $E$ . For an integer  $\delta \geq 0$ , a  $\delta$ -strong rule is an association rule of the form  $X \Rightarrow Y$ , where  $X$  and  $Y \neq \emptyset$  are disjoint subsets of  $E$  satisfying

$$|\mathcal{D}[X \cup Y]| \geq |\mathcal{D}[X]| - \delta.$$

The two properties can be defined in terms of strong rules as follows: A set  $U \subseteq E$  is

$\delta$ -free if and only if  $X \Rightarrow Y$  is *not*  $\delta$ -strong for every disjoint subsets  $X, Y \subseteq U$  satisfying  $Y \neq \emptyset$  and

$\Delta$ -closed if and only if  $U \Rightarrow a$  is *not*  $(\Delta - 1)$ -strong for every  $a \in E \setminus U$ .

Thus, while  $\delta$ -freeness depends only on the support sets of the subsets of  $U$ ,  $\Delta$ -closedness is a function of the support set of  $U$  and the support sets of the elements in  $E \setminus U$ . This implies that the two properties are incomparable; for example, for the dataset given in Figure 1 we have that  $d$  is  $\delta$ -free for every  $\delta \leq 3$ , but it is not  $\Delta$ -closed for every  $\Delta \geq 1$ . On the other hand,  $abcd$  is  $\Delta$ -closed for every  $\Delta \leq 2$ , but it is not  $\delta$ -closed for every  $\delta \geq 0$  for the rule  $bd \Rightarrow c$ .

We also note that in contrast to  $\Delta$ -closedness,  $\delta$ -freeness is an anti-monotone (or hereditary) property [1] (i.e., every subset of a  $\delta$ -free set is also  $\delta$ -free).

**Pattern Compression** It is well-known that closed sets provide a lossless compact representation of

frequent sets (see, e.g., [9]). However, as mentioned earlier,  $\Delta$ -closed sets do *not* provide a lossless compact representation of frequent sets for any  $\Delta > 1$ . To see this e.g. for  $\Delta = 2$ , consider again the example given in Figure 1. For frequency threshold  $t = 1$ , we have  $\mathcal{C}_{2,1} = \{\emptyset, abcd, ef\}$ , from which the frequent set  $de$  cannot be derived.

Beside lossless compression, there is also an interest in approximations of frequent sets. In [12], for example, such an approximation based on frequent set clustering is proposed. More precisely, the authors define a normalized distance measure  $\mu$  between itemsets and for a given distance threshold  $\delta \in [0, 1]$ , they compute a potentially small family  $\mathcal{P}$  of *representative sets* such that for each frequent set  $F$  there is a representative set  $P \in \mathcal{P}$  satisfying  $F \subseteq P$  and  $\mu(F, P) \leq \delta$ . The family of frequent sets is then approximated by the family of sets covered by the union of the  $\delta$ -balls of the representative sets.

One might ask, whether the family of  $\Delta$ -closed sets provide such a family of representative sets for some appropriate distance function. The answer to this question appears negative, as it may happen that the empty set is the only  $\Delta$ -closed set (see, e.g., Figure 1 for  $\Delta = 3$ ) which may have an arbitrary distance to any other set with respect to any reasonable distance function.

**Error-Tolerant Frequent Sets** One motivation for our notion of interestingness was that it exhibits a certain robustness against noise. There are several notions of error- or fault-tolerant frequent sets, e.g., the one introduced in [13] that are aiming for a similar goal. They have in common that they relax the minimum frequency constraint resulting in a certain *superset* of the family of all frequent sets. In contrast, our method induces a *subset* of that family, i.e, we strengthen the usual constraint.

## 6 Discussion

In this paper, we have introduced a novel interestingness measure able to effectively capture long patterns, which is, at the same time, robust against noise and/or dynamic changes in the dataset. Our interestingness predicate is a strict generalization of closedness, and captures the semantic notion of those patterns that are at the boundary of a sharp drop in support when augmenting them with additional items. More intuitively, the closedness of the sets is strong even if the underlying database were changed. The class of strongly closed sets is parameterized by a numerical parameter specifying the magnitude of the drop in support when augmenting the closed set. By selecting this parameter appropriately, the user can thus control the size of the resulting

output while at the same time having a semantically meaningful specification and guarantee of what kinds of patterns to expect in the output set. This is especially useful in applications where increasing the support threshold would lead to an unacceptable loss of less frequent, but still interesting long patterns. By building on closed sets, our approach directly ties into previous research on this topic. As we have shown in the paper, it is possible to generate strongly closed sets using a suitable closure operator. We have used this operator to design an efficient mining algorithm. In particular, for non-sparse datasets its time complexity is equal to the best known time bound for listing ordinary closed frequent sets.

Our experimental evaluation has shown that strongly closed sets indeed give rise to small but interesting solution sets, and thus offer a viable alternative to the use of the support threshold parameter.

We close this section with a problem for future work. The interestingness notion used in this paper can be generalized as follows: Let  $\mathcal{D}$  be a dataset over a set  $E$  (columns) and let  $T$  denote the transaction identifiers of  $\mathcal{D}$  (rows). For integers  $\Delta_1, \Delta_2 \geq 0$ , a pair  $(A, B)$  with  $A \subseteq T$  and  $B \subseteq E$  is a  $(\Delta_1, \Delta_2)$ -*concept* if  $A$  is  $\Delta_1$ -closed for the transpose matrix of  $\mathcal{D}$  and  $B$  is  $\Delta_2$ -closed (for  $\mathcal{D}$ ). Notice that ordinary concepts defined in formal concepts analysis (see, e.g., [4]) are  $(1, 1)$ -concepts. Given dataset  $\mathcal{D}$ , and integers  $\Delta_1, \Delta_2$ , the question is whether the set of  $(\Delta_1, \Delta_2)$ -concepts can also be listed with polynomial delay.

## Acknowledgement

Tamás Horváth was partially supported by the German Federal Ministry of Economy and Technology under the Theseus Project.

## References

- [1] Jean-François Boulicaut, Artur Bykowski, and Christophe 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.
- [2] James Cheng, Yiping Ke, and Wilfred Ng.  $\delta$ -tolerance closed frequent itemsets. In *Proc. of the 6th IEEE Int. Conf. on Data Mining (ICDM)*, pages 139–148. IEEE Computer Society, 2006.
- [3] B. Ganter and K. Reuter. Finding all closed sets: A general approach. *Order*, 8(3):283–280, 1991.
- [4] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer Verlag, 1999.
- [5] Alain Gély. A generic algorithm for generating closed sets of a binary relation. In *Proc. of 3rd Int. Conf.*

on Formal Concept Analysis (ICFCA), volume 3403 of LNCS, pages 223–234. Springer, 2005.

- [6] Bart Goethals and Mohammed Javeed Zaki. Advances in frequent itemset mining implementations: report on fimi'03. *SIGKDD Explorations*, 6(1):109–117, 2004.
- [7] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the web for emerging cyber-communities. *Computer Networks*, 31(11-16):1481–1493, 1999.
- [8] Heikki Mannila and Hannu Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.
- [9] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1):25–46, 1999.
- [10] Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier, and Lotfi Lakhal. Computing iceberg concept lattices with t. *Data & Knowledge Engineering*, 42(2):189–222, 2002.
- [11] Takeaki Uno, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura. An efficient algorithm for enumerating closed patterns in transaction databases. In *Discovery Science*, pages 16–31, 2004.
- [12] Dong Xin, Jiawei Han, Xifeng Yan, and Hong Cheng. Mining compressed frequent-pattern sets. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 709–720. VLDB Endowment, 2005.
- [13] Cheng Yang, Usama Fayyad, and Paul S. Bradley. Efficient discovery of error-tolerant frequent itemsets in high dimensions. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 194–203, New York, NY, USA, 2001. ACM.

## A Zero-Elimination Algorithm

LEMMA A.1. *After the execution of line 4 the following invariants hold for all  $e \in E \setminus F$  throughout the remaining computation:*

$$\Sigma(e) \geq |\bar{\mathcal{D}}[\{e\}] \cap \mathcal{D}[\sigma_\Delta(F)]| + |\{D \in \bar{\mathcal{D}}[\{e\}]: D \in \bar{\mathcal{D}}[\sigma_\Delta(F)] \wedge D \notin X\}| .$$

Moreover, for all  $e$  added to  $C$  it holds that  $e \in \sigma_\Delta(F)$ .

*Proof.* After line 4 the values  $\Sigma(e)$  are initialized to  $|\{D \in \mathcal{D}[F]: e \notin D\}|$ . Thus the invariants hold at this point and no  $e$  has been added to  $C$  that could violate  $e \in \sigma_\Delta(F)$ . Subsequently  $\Sigma$  and  $C$  are only modified during deleteColumn and deleteRow. We prove correct behavior of these procedures by induction on their calling order. Consider a call deleteRow( $D$ ). By induction before the execution of line 1 the invariant holds. Hence, after the addition of  $D$  to  $X$  for all  $e \notin D$ ,

---

### Algorithm 3 Zero-Elimination

---

Input : subset  $F \subseteq E$ ,  
positive integer  $\Delta$ ,  
dataset  $\mathcal{D}$  over  $E$  restricted to  $\mathcal{D}[F] \times (E \setminus F)$   
given by incidence list complements  
 $\bar{D} = \{e \in E \setminus F: e \notin D\}$ , for all  $D \in \mathcal{D}[F]$   
Output :  $C$  equal to  $\sigma_\Delta(F)$

**deleteColumn( $e$ ):**

1.  $C \leftarrow C \cup \{e\}$
2. **for all**  $D \in L_e$  **do**
3.   **if**  $D \notin X$  **then** deleteRow( $D$ )

**deleteRow( $D$ ):**

1.  $X \leftarrow X \cup \{D\}$
2. **for all**  $e \in \bar{D}$  **do**
3.    $\Sigma(e) \leftarrow \Sigma(e) - 1$
4.   **if**  $(\Sigma(e) < \Delta$  **and**  $e \notin C)$  **then** deleteColumn( $e$ )

**main:**

1.  $C \leftarrow F, X \leftarrow \emptyset$
  2. **for all**  $e \in E \setminus F$  **do**
  3.    $L_e \leftarrow \{D \in \mathcal{D}[F]: e \notin D\}$
  4.    $\Sigma(e) \leftarrow |L_e|$
  5. **for all**  $e \in E \setminus F$  **do**
  6.   **if**  $\Sigma(e) < \Delta$  **and**  $e \notin C$  **then** deleteColumn( $e$ )
  7. **return**  $C$
-

$\Sigma(e) - 1$  is greater or equal to

$$|\bar{\mathcal{D}}[\{e\}] \cap \mathcal{D}[\sigma_\Delta(F)]| + |\{D \in \bar{\mathcal{D}}[\{e\}]: D \in \bar{\mathcal{D}}[\sigma_\Delta(F)] \wedge D \notin X\}| .$$

Since in line 3 the number  $\Sigma(e)$  is decremented by only one for such an  $e$ , this step does not violate any invariant.

Furthermore, for a call `deleteColumn(e)` we know that  $\Sigma(e) \leq \Delta$  and thus by induction that  $|\bar{\mathcal{D}}[\{e\}] \cap \mathcal{D}[\sigma_\Delta(F)]|$  is not greater than  $\Delta$ . It follows that  $e \in \sigma_\Delta(F)$ .  $\square$

LEMMA A.2. *Let  $e \in (\sigma_\Delta(F) \setminus F)$ . Then Algorithm 3 will add  $e$  to  $C$  eventually.*

*Proof.* Since  $e \in \sigma_\Delta(F)$  there is a  $k$  such that  $e \in \hat{\sigma}_\Delta^k(F) \setminus \hat{\sigma}_\Delta^{k-1}(F)$ . We prove the claim by induction on  $k$ . In case  $k = 1$ , `deleteColumn(e)` is called in Line 6 of the main procedure and subsequently added to  $C$ . Otherwise by induction there is a state of the algorithm in which  $C \supseteq \hat{\sigma}_\Delta^{k-1}(F)$  and so there is a subsequent state in which `deleteRow(D)` has been called for all  $D \in \mathcal{D}[\hat{\sigma}_\Delta^{k-1}(F)]$ . Since  $|\mathcal{D}[\hat{\sigma}_\Delta^{k-1}(F)]| - |\mathcal{D}[\hat{\sigma}_\Delta^{k-1}(F) \cup \{e\}]| < \Delta$  the counter  $\Sigma(e)$  will have been set to a value smaller than  $\Delta$  at this state and consequently  $e$  will be added to  $C$ .  $\square$

*Proof.* [Proof of Theorem 4.1.] We show that Algorithm 3 can be used to compute  $\sigma_\Delta$  with the claimed complexity. For the running time we treat each of the three procedures separately. The first loop in line 2 can be realized by one pass through the data in time  $O(\|\mathcal{D}[F]\|_0)$ . The same holds for the second for-loop constituting a total time of  $O(\|\mathcal{D}[F]\|_0)$  spend in the main procedure.

For the `deleteColumn` procedures observe that it is called at most once for each  $e \in E \setminus F$  and contains a loop over all transactions  $D \in \mathcal{D}[F]$  with  $e \notin D$  realized by one traversal of the list  $L_e$  in time  $O(|L_e|)$ . So the total time spend within this procedure is bounded by  $O(\sum_{e \in E \setminus F} |L_e|) = O\|\mathcal{D}[F]\|_0$ . Similarly, the `deleteRow` procedure is called at most once for each element  $D \in \mathcal{D}[F]$  and contains a loop over all  $e \in \bar{D}$ . Thus the time spend in this procedure and thus the overall time complexity of the whole algorithm is bounded by  $O(\|\mathcal{D}[F]\|_0)$ .

For the correctness, Lemma A.1 implies that always  $C \subseteq \sigma_\Delta(F)$ . Together with Lemma A.2 it follows that  $C$  as returned in line 7 of the main procedure is equal to  $\sigma_\Delta(F)$ .  $\square$