Likely-Occurring Itemsets for Pattern Mining

Tatiana Makhalova, Sergei O. Kuznetsov, and Amedeo Napoli

[1] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
`tatiana.makhalova@inria.fr`
[2] National Research University Higher School of Economics, Moscow, Russia
`skuznetsov@hse.ru`
[3] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
`amedeo.napoli@loria.fr`

Abstract. We consider the itemset mining problem in general settings, e.g., mining association rules and itemset selection. We introduce the notion of likely-occurring itemsets and propose a greedy approach to itemset search space discovery that allows for reducing the number of arbitrary or closed itemsets. This method provides itemsets that are useful for different objectives and can be used as an additional constraint to curb the itemset explosion. In experiments, we show that the method is useful both for compression-based itemset mining and for computing good-quality association rules.

1 Introduction

A generic objective of itemset mining is to discover a small set of non-redundant and interesting itemsets that describe together a large portion of data and that can be easily interpreted [1].

Itemset mining can be summarized into two steps: (i) discovering itemset search space and (ii) selecting interesting itemsets among the discovered ones.

This paper is devoted to the first step, i.e., the itemset search space discovery. Since the itemset search space contains exponentially many elements, it is important to discover as few useless itemsets as possible.

There are several approaches to discover the itemset search space: (i) an exhaustive enumeration of all itemsets followed by a selection of those satisfying imposed constraints [19], (ii) a gradual enumeration of some itemsets guided by an objective (or by constraints) [17], (iii) mining top-k itemsets w.r.t. constraints [15], (iv) sampling a subset of itemsets w.r.t. a probability distribution that conforms to an interestingness measure [6,7]. To reduce redundancy when enumerating itemsets, the search space can be shrunk to *closed itemsets*, i.e., the maximal itemsets among those that are associated with a given set of objects (support).

The exhaustive enumeration is the most universal way to discover itemset search space. There exists a lot of very efficient algorithms for its enumeration, e.g., CBO [12], IN-CLOSE [3], LCM [18], ALPINE [11], and others [13].

Despite its wide usage and applicability for a large spectrum of interestingness measures, the exhaustive enumerators usually mine itemsets w.r.t. frequency, which results in the following issues: using too high frequency threshold results in a considerable amount of not interesting itemsets, while too low frequency threshold results in itemset explosion and intractability of itemset mining methods in practice.

However, considering the itemset mining problem in more general settings, e.g., mining association rules and implications, the exhaustive enumeration of frequent itemsets is usually the only (universal) remedy for the pattern explosion problem.

In this paper, we revisit the notion of *likely-occurring* itemsets introduced in [14] and propose a greedy approach to itemset search space discovery that allows for reducing the number of closed itemsets. This method provides itemsets that are useful for different objectives and can be used as an additional constraint to curb the itemset explosion. In experiments we show that the method is useful both for compression-based itemset mining and for computing good-quality association rules.

2 Preliminaries

We deal with binary datasets within the FCA framework [10].

A *formal context* is a triple $\mathbb{K} = (G, M, I)$, where G is a set of objects, M is a set of attributes and $I \subseteq G \times M$ is the incidence relation, i.e., $(g, m) \in I$ if object g has attribute m.

Two derivation operators $(\cdot)'$ are defined for $A \subseteq G$ and $B \subseteq M$ as follows:

$$A' = \{m \in M \mid \forall g \in A : gIm\}, \ B' = \{g \in G \mid \forall m \in B : gIm\}.$$

For $A \subseteq G$, $B \subseteq M$, a pair (A, B) such that $A' = B$ and $B' = A$, is called a *formal concept*, then A and B are closed sets and called *extent* and *intent* (or *closed itemsets*), respectively.

The (empirical or observed) probability of an itemset $X \subseteq M$ is given by $P(X) = fr(X) = |X'|/|G|$.

3 Likely-occurring itemsets

To reduce the itemset search space, we propose an additional constraint that consists in considering only the itemsets whose observed probability is greater than the estimated one. The estimated probability is computed under the independence model. We give the details on the chosen independence model below.

Definition 1. *A closed itemset $X \subseteq M$ is called* likely-occurring closed *(LOC) if there exists $m \in X$ and $Y \subseteq X \setminus \{m\}$, $(Y \cup \{m\})'' = X$ such that $P(X) > Q \cdot P(Y) \cdot P(\{m\})$, and $Q \geq 1$.*
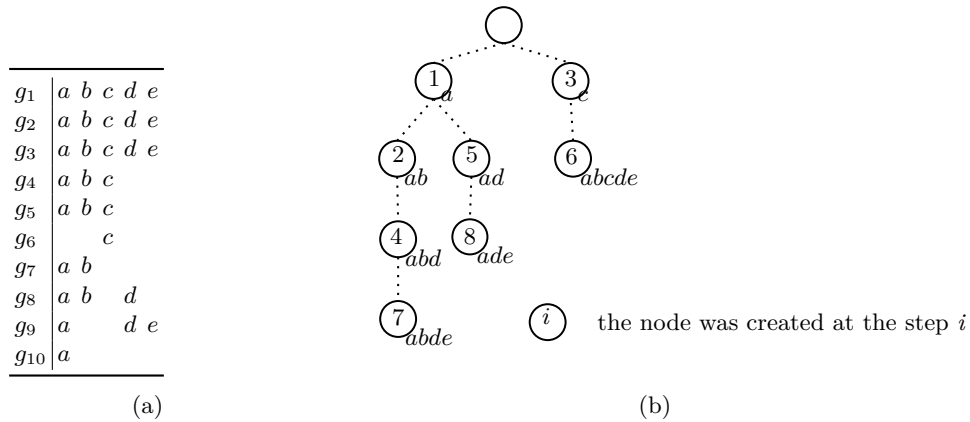
g_1	a	b	c	d	e	
g_2	a	b	c	d	e	
g_3	a	b	c	d	e	
g_4	a	b	c			
g_5	a	b	c			
g_6			c			
g_7	a	b				
g_8	a	b		d		
g_9	a			d	e	
g_{10}	a					

(a)　　　　　　　　　　　　　　　　　(b)

Fig. 1: A binary dataset and the execution tree of Algorithm 1 for this dataset

The empty itemset $\emptyset$ is considered to be likely-occurring by default. The parameter Q controls how large the difference between the observed probability $P(X)$ and the estimated probability $P(Y) \cdot P(\{m\})$, $Y \subseteq X \setminus \{m\}$ of the itemset X may be. The least restrictive constraint, i.e., $Q = 1$, requires the observed probability to be greater than the estimated one. The larger values of Q are more restrictive, i.e., they require the observed probability to be much larger than the estimated one.

According to the definition above, at most $|X|$ splittings should be enumerated to check whether an itemset X is LOC or not. To make it more tractable in practice, we propose a relaxation of the LOC itemset and a greedy approach for its computing, where one needs to check only one splitting per itemset. Let us proceed to this definition.

Definition 2. *Let $\{m_1, m_2, \cdots, m_k\}$ be a set of attributes arranged in order of decreasing frequency, i.e., $fr(m_i) \geq fr(m_j)$ for any $i \leq j$. A closed itemset X is* likely-occurring closed *(LOC) if there exists a LOC itemset $Y \subset X$ and $m \in X \setminus Y$ such that $fr(m) \geq \min_{m^* \in Y} fr(m^*)$, $X = (Y \cup \{m\})''$ and $P(X) > Q \cdot P(Y) \cdot P(\{m\})$.*

Example. Let us consider a running example from Fig. 1a, where the attributes are arranged by decreasing frequency. Itemset ab is an LOC itemset because a is an LOC itemset and $P(ab) > P(a) \cdot P(\{b\})$, the same for abd, namely, abd is an LOC itemset because ab is an LOC itemset and $P(abd) > P(ab) \cdot P(\{d\})$, etc.

We propose an algorithm to compute LOC itemsets using Definition 2, its pseudocode is given in Algorithm 1. This algorithm computes gradually LOC itemsets by considering one by one attributes of decreasing frequency. Apart from the threshold Q on the difference in probabilities, the algorithm also supports threshold F on frequency. By default, we use minimal restrictions, namely $Q = 1$ (we require the observed probability to be greater than the estimated one) and $F = 0$ (we do not impose any frequency constraints).

Algorithm 1 C">Compute</small>LOC

Procedure M<small>ERGE</small> (*node*,*candidate*)

Input: *node*, current node
    candidate, candidate node
1: $I_n \leftarrow node.itemset$
2: $I_c \leftarrow candidate.itemset$
3: **if** $|I_n \setminus I_c| > 0$ **then**
4:    $extent \leftarrow (I_c \cup I_n)'$ {computing shared objects}
5:    **if** $extent = I_c'$ **then**
6:      $I_n \leftarrow I_n \cup I_c$ {if I_n is included into all objects as I_n, just extend I_c}
7:    **else if** $\left(\frac{|extent|}{|G|} > Q \cdot \frac{|I_c'|}{|G|} \frac{|I_n'|}{|G|} \right)$ **and** $|extent| \geq F$ **then**
8:      **for all** $child \in node.children$ **do**
9:        $merge(child, candidate)$
10:     **end for**
11:      **if** $candidate \notin \mathcal{T}$ **then**
12:        $node.children.add(candidate)$
13:     **end if**
14:    **end if**
15: **end if**

Input: (G, M, I) formal context
    F, frequency threshold
    Q, threshold on LOC
Output: $\mathcal{T}$, a tree composed of LO/LOC-itemsets
1: $\mathcal{T} \leftarrow createEmptyTree()$
2: $root \leftarrow \mathcal{T}.getRoot()$
3: $M^* \leftarrow sortByDescendingFrequency(M)$
4: **for all** $m \in M^*$ **do**
5:    $candidate \leftarrow m''$ {for LOC itemsets}
6:    **for all** $child \in root.children$ **do**
7:      $merge(child, candidate)$
8:      **if** $candidate \notin \mathcal{T}$ **then**
9:        $root.children.add(candidate)$
10:     **end if**
11:    **end for**
12: **end for**
13: **return** $\mathcal{T}$

Example. Let us consider the execution tree of the algorithm for a dataset from Fig. 1a. The algorithm starts constructing a tree adding the attributes of decreasing frequency. The order in which itemsets are enumerated is specified in the corresponding nodes.

4 Likely-occurring itemsets and related notions

Probability-based models are common in itemset and association rule mining. In this section we consider two widespread approaches to assess itemsets and

association rules, and discuss how they are related to likely-occurring closed itemsets.

Independence model and lift. The models based on the comparison of estimated and observed probabilities of itemsets are quite common in the scientific literature. The simplest model is the attribute independence model. Under this model, all items (attributes) are assumed to be independent. Attribute probability is approximated straightforwardly using the attribute frequency. Then, the probability of an itemset X is computed as follows:

$$P_{ind}(X) = \prod_{x \in X} P(x) = \prod_{x \in X} fr(x).$$

Despite its simplicity, this model is widely used in machine learning, e.g., Naïve Bayes classifiers are based on it. A natural extension of the attribute independence model is the partition independence model, where some partitions of X are assumed to be independent. Lift [8] is one of the most common measures to assess association rules under the partition independence model.

Definition 3. *Let $X \to Y$ be an association rule, then* lift *is given by*

$$lift(X \to Y) = \frac{P(XY)}{P(X)P(Y)} = \frac{fr(XY)}{fr(X)fr(Y)}.$$

Apart from lift, there is a lot of other measures (indices) based on the comparison of the antecedent and consequent supports, e.g., redundancy constraints [4,22], minimum improvement [5], etc. They are commonly used to select association rules.

The notion of lift can be also adapted in different ways for itemset assessment. For example, one may assess the probability of an itemset under the assumption that any partition of the itemset into two disjoint sets is independent. If the observed probability is greater than all the estimated probabilities obtained under this model, then the itemset is called *productive* [21].

The introduced above LOC itemsets, in a certain sense, represent a particular case of productive itemsets. Instead of considering all possible partitions of X into two sets of items, we consider only its proper subset Y and attribute $m \in X \backslash Y$. Reformulating the definition of LOC in terms of lift (for association rules), LOC itemset X is an itemset that consists of LOC itemset Y and attribute m such that $Y \cup \{m\}$ is the generator of X, and $lift(X \to m) > Q$, $Q \geq 1$. Since Y is also LOC, this reasoning can be done recursively.

If it is needed, one may reduce further the size of the discovered LOC itemsets by putting more tighter constraints, i.e., setting higher values for Q (in line 7 of the MERGE procedure given in Algorithm 1):

$$\frac{|(I_c \cup I_n)'|}{|G|} > Q \cdot \frac{|I_c'|}{|G|} \cdot \frac{|I_n'|}{|G|}.$$

The constraint above is equivalent to the constraint on lift of the association rule $I_n \rightarrow I_c$, i.e.,

$$lift(I_n \rightarrow I_c) = \frac{P(I_n \cup I_c)}{P(I_n) \cdot P(I_c)} > Q.$$

Moreover, because of the greedy strategy, the constraints hold recursively, i.e., there exist two disjoint subsets I_n^*, $I_c^* \subseteq I_n$ such that $lift(I_n^* \rightarrow I_c^*) > Q$. In experiments we consider how the proposed greedy strategy works for mining association rules on real-world datasets. Since the computing strategy is greedy, there are no guarantees that all LOC itemsets (see Definitions 1) will be enumerated.

Itemset mining through compression Likely-occurring itemsets may be also useful for selection of itemsets. We consider the relation between the itemsets selected by a compression-based itemset miner KRIMP [20] and LOC itemsets.

In KRIMP, and similar methods, the length of the code word corresponding to an itemset X is given by $length(X) = -\log P(X)$. Hence the compression is achieved by replacing several code words representing the itemsets B with a single code word, such that $length(B) < \sum_{X \in cover(B)} length(X)$. The latter is equivalent to $\log P(B) > log(\prod_{X \in cover(B)} P(X))$. Thus, we obtain the inequality $P(B) > \prod_{X \in cover(B)} P(X)$, which is very similar to one from the definition of the LOC itemsets.

Intuitively, in both cases, an itemset is considered optimal if its observed probability is greater than the estimated one. However, there are important differences between the models underlying the definition of "itemset optimality" (for the LOC itemsets and the model used in KRIMP):

1. the both methods use different probability estimates of itemsets, namely, $P(X) = fr(X)$ (for the LOC estimates) and $P(X) = \frac{usage(X)}{\sum_{Y \in \mathcal{P}} usage(Y)}$ (for the KRIMP-like models), where $usage(X)$ is frequency of X in the coverage, and $\mathcal{P}$ is the set of patterns;

2. the "optimality" of an itemset X in the compression-based model used in KRIMP is evaluated not only w.r.t. the dataset but also w.r.t. the other itemsets selected so far.

Thus, LOC itemsets may provide better results than the commonly used frequent closed itemsets, which are used by KRIMP. We compare different strategies for discovering itemset search space on real-world datasets in the next section.

5 Experiments

We use the discretized datasets from the LUCS-KDD repository [9] and study LOC itemsets[4] for two tasks, namely association rule and itemset mining.

Association rule mining. Frequent (closed) itemsets are commonly used to mine association rules. We study how useful LOC itemsets compared to frequent closed itemsets. In experiments we use 2 different sets of itemsets to compute rules: frequent closed (FR.CL.) and likely-occurring closed (LOC) itemsets. The itemsets are evaluated on 10 datasets, their parameters are given in Table 1. The number of objects and attributes is denoted by $|G|$ and $|M|$, respectively. The density of datasets (the ratio of 1's) is given in the column "density". The total number of closed itemsets is reported in the column "#CL". The total number of arbitrary itemset has not been computed.

Table 1: Parameters of datasets and the studied sets of itemsets

name	data description				closed itemsets			time (for itemsets)		#rules	
	$|G|$	$|M|$	density	#CL	#LOC	#FR.CL.	fr.thr.	LOC	FR	LOC	FR.CL
breast	699	14	0.64	360	74	74	0.33	0.01	0.12	4292	3980
ecoli	327	24	0.29	424	120	120	0.06	0.02	0.60	4768	2950
glass	214	40	0.22	3211	887	955	0.06	0.10	10.85	55262	18454
heart-dis.	303	45	0.29	25511	1928	1973	0.17	0.28	1.43	862252	22222
iris	150	16	0.25	106	45	47	0.05	0.00	0.03	274	320
led7	3200	14	0.50	1936	150	150	0.19	0.01	0.05	1484	1120
pima	768	36	0.22	1608	317	317	0.10	0.06	4.57	21294	7528
ticTacToe	958	27	0.33	42684	1880	1908	0.03	0.11	14.90	53816	13016
wine	178	65	0.20	13169	4914	5647	0.03	1.20	520.43	1771852	189378
zoo	101	35	0.46	4552	610	621	0.33	0.10	1.14	1609108	24736
average	**690**	**32**	**0.34**	**9356**	**1093**	**1181**	**0.14**	**0.19**	**55.41**	**438440**	**28370**

For each dataset we generate the whole set of LOC itemsets ($Q = 1$, $F = 0$), the sizes of these sets are indicated in the column "#LOC".

We chose the frequency threshold for closed itemsets in such a way that the number of closed itemsets is equal to the number of the LOC itemsets. The frequency threshold is indicated in the column "fr.thr." for closed itemsets. The frequency threshold varies a lot from dataset to dataset. For example, the smallest threshold is 0.06 for "ecoli" and "glass" datasets and the largest one is 0.33 for "breast" and "zoo" dataset. As we can see from the table, the sizes of "#LOC" and "#FR.CL." are quite close one to another.

To compute association rules we use a miner from MLxTENT library implemented in Python[5]. The number of rules generated based on LOC and frequent closed (FR.CL.) is reported in the column "#rules".

---

[4] The source code for computing LOC itemsets is available at `https://github.com/makhalova/pattern_mining_tools/blob/master/modules/binary/likely_occurring_itemsets.py`

[5] `http://rasbt.github.io/mlxtend/`

The number of rules generated based on the LOC itemsets is higher than the number of rules generated based on frequent closed itemsets. For example, for the "ecoli" dataset, the number of rules computed on 120 LOC and 120 frequent closed itemsets is 4768 and 2950, respectively. It can be explained by the fact that the size of the LOC itemsets is usually larger than the size of frequent closed itemsets. Thus, a larger amount of rules can be built on LOC itemsets by splitting each itemset into an antecedent and consequent.

To evaluate their quality, we consider the most common quality measures for the association rules, namely *support*, *confidence*, *lift*, *leverage*, and *conviction*. We recall them below.

Let $X \to Y$ be an association rule with the antecedent X and the consequent Y, then the rule support is given by

$$support(X \to Y) = support(X \cup Y) = \frac{(X \cup Y)'}{|G|} \in [0, 1].$$

Confidence [2] of a rule $X \to Y$ is the conditional probability of $X \cup Y$ given X. It is defined as follows:

$$confidence(X \to Y) = \frac{support(X \to Y)}{support(X)} \in [0, 1].$$

The maximal value is achieved when Y always occurs with X.

Lift [8] was discussed in the previous section. We recall it below. For a rule $X \to Y$ lift is given by

$$lift(X \to Y) = \frac{support(X \to Y)}{support(X) \cdot support(Y)} \in [0, \infty).$$

Leverage [16], like lift, is based on the comparison of the observed probability (frequency) of the rule and the probability estimated under the assumption that the antecedent and consequent are independent. Leverage is given as follows:

$$leverage(X \to Y) = support(X \to Y) - support(X) \cdot support(Y) \in [-1, 1].$$

For independent X and Y leverage is equal to 0.

Let us proceed to the results of the experiments.

For the generated rules we consider mean values of the aforementioned quality measures as well as the 75th, 90th, and 95th percentiles. Considering the percentiles allows us to focus on the quality of the best itemsets, which are usually of interest to analysts. The averaged over 10 dataset values are reported in Fig. 2.

Since we do not set any frequency threshold for LOC, the support of LOC-based rules, as expected, is lower than the support of the rules based on frequent closed itemsets (FR.CL.). The top $n\%$ of LOC-based rules have higher values than the top $n\%$ of FR.CL.-based ones. For example, the top 10% values (the 90th percentile) of confidence are at least 0.935 for the LOC-based rules, and
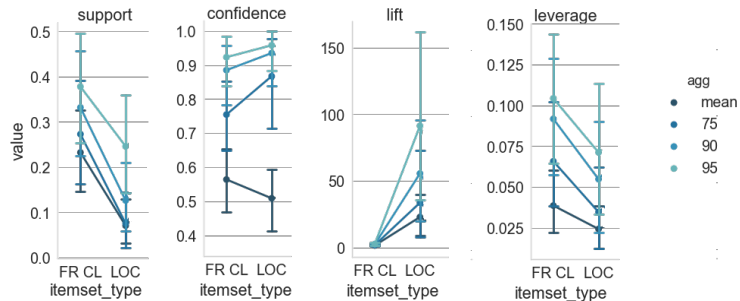
Fig. 2: The averaged quality for 2 types of rules: computed based on frequent closed (FR.CL.) and LOC itemsets. The quality is measured by support, confidence, lift, and leverage. For each type of rules and each quality measure, the average values of mean, the 75th, 90th, and 95th percentiles over 10 datasets from Table 1 are reported

only 0.885 for FR.CL.-based rules, respectively. Thus, considering the top rules, the LOC-based rules have higher confidence.

Regarding lift, LOC-based rules provide the best results. The difference in values is especially noticeable for the top 5% of rules (the 95th percentile). Top 5% LOC-rules have the highest values of lift, on average, 91.38. However, the lift values of the top 5% of rules vary a lot from dataset to dataset (the standard deviation is shown in plots by horizontal lines). Nevertheless, the quality, measured by lift, is consistently higher for LOC-based rules than for FR.CL.-based rules.

The leverage is higher for FR.Cl.-based rules. Despite the fact that lift and leverage differ only in the mathematical operations they use to compare the observed and estimated supports of rules and their parts, the analysis of rules based on these measures may lead to very different results. The high values of leverage (and low values of lift) for FR.CL.-based rules are caused by a different order of magnitude of the supports. Very low supports (that is the case of LOC-based rules) result in high values of lift and low values of leverage.

Thus, the analysis of the generated rules allows us to conclude that rules generated based on LOC itemsets have better quality than the rules generated using roughly the same amount of frequent arbitrary and closed itemsets, respectively.

Compression quality. In Section 4 we discussed the relation between LOC itemsets and the itemsets ensuring good compression in KRIMP.

In this section we study the applicability of LOC itemsets for this task and compare them with closed itemsets (used in the original version of KRIMP. We emphasize that, in the compared approaches, the itemset search space is discovered independently of the itemset mining process.

To evaluate the ability of the itemsets to compress data, we consider how many itemsets we need to obtain a certain compression ratio. Fig. 3 shows how the compression ratio changes w.r.t. the number of considered itemsets. The
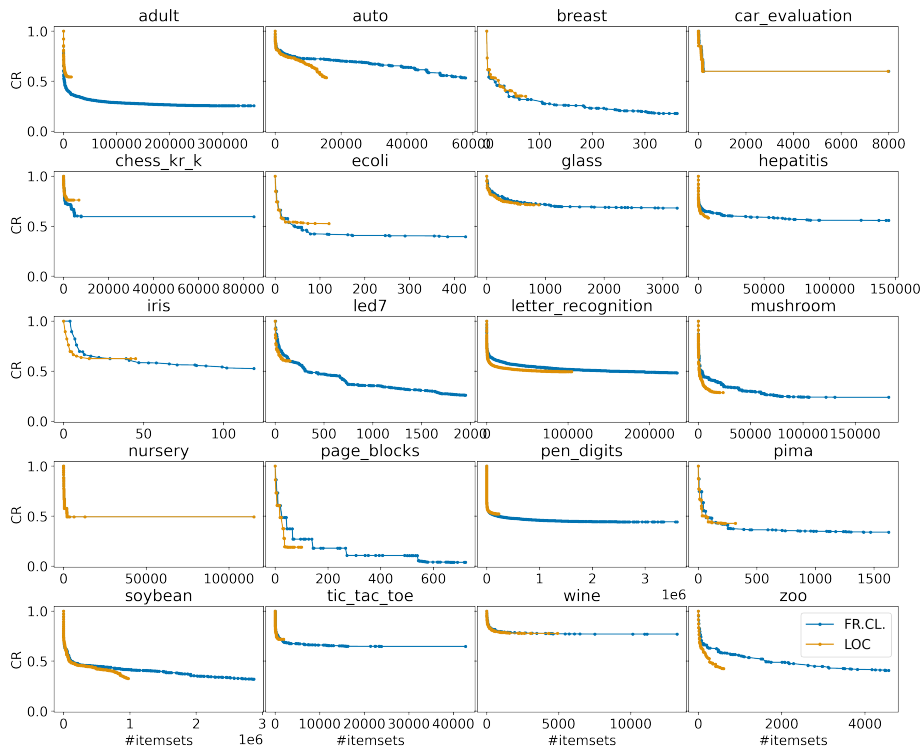
Fig. 3: Compression quality of closed (FR.CL.) and LOC itemsets. The lower values are better

initial state corresponds to the point (0,1), meaning that 0 itemsets have been used to compress data, and the compression ratio is maximal and equal to 1. The curves that are closer to the point (0,0) correspond to the best strategies of itemset search space discovery (i.e., the itemset set allows for compressing data better with a lower number of itemsets). The experiments show that for "car evaluation", "wine" and "nursery" datasets the LOC itemsets do not provide any benefits over the closed itemsets. For the majority of datasets, the number of LOC is too small to ensure as good compression as with the whole set of closed itemsets, e.g., "adult", "breast", "led7", and others. Among some of these datasets, we may still observe better behavior of LOC itemsets, e.g., for "hepatitis", "mushroom", "letter recognition", and "page blocks". There are also datasets where with the LOC itemsets we achieve as good compression as with the closed ones, but use a much lower number of itemsets, e.g., "auto", "hepatitis", "soybean", "zoo".

In general, LOC itemsets may be quite useful for itemset selection based on compression.

6 Conclusion

In this paper we studied likely-occurring closed itemsets in the context of association rule mining and itemset selection. In our experiments we show that the number of frequent enumerated LOC itemsets is much lower than the number of frequent closed itemsets. However, with LOC itemsets, we obtain association rules of better quality. The proposed approach may be useful for compression as well, however, it does not outperform the methods where itemsets are discovered towards the direction minimizing the total description length.

References

1. Aggarwal, C.C., Han, J. (eds.): Frequent Pattern Mining. Springer (2014)
2. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the International Conference on Management of Data. vol. 22, pp. 207–216. ACM SIGMOD (1993)
3. Andrews, S.: A partial-closure canonicity test to increase the efficiency of CbO-type algorithms. In: International Conference on Conceptual Structures. pp. 37–50. Springer (2014)
4. Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: Mining frequent patterns with counting inference. In: ACM SIGKDD Explorations Newsletter. vol. 2. ACM SIGKDD (2000)
5. Bayardo, R.J., Agrawal, R., Gunopulos, D.: Constraint-based rule mining in large, dense databases. Data Mining and Knowledge Discovery **4**(2-3), 217–240 (2000)
6. Boley, M., Lucchese, C., Paurat, D., Gärtner, T.: Direct local pattern sampling by efficient two-step random procedures. In: Proceedings of the 17th International Conference on Knowledge discovery and Data Mining. pp. 582–590. ACM SIGKDD (2011)
7. Boley, M., Moens, S., Gärtner, T.: Linear space direct pattern sampling using coupling from the past. In: Proceedings of the 18th International Conference on Knowledge Discovery and Data Mining. pp. 69–77. ACM (2012)
8. Brin, S., Motwani, R., Ullman, J.D., Tsur, S.: Dynamic itemset counting and implication rules for market basket data. In: Proceedings of the International Conference on Management of Data. pp. 255–264. ACM SIGMOD (1997)
9. Coenen, F.: The LUCS-KDD discretised/normalised ARM and CARM data library. `http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS_KDD_DN` (2003)
10. Ganter, B., Wille, R.: Formal Concept Analysis. Springer Berlin Heidelberg (1999). https://doi.org/10.1007/978-3-642-59830-2, `http://dx.doi.org/10.1007/978-3-642-59830-2`
11. Hu, Q., Imielinski, T.: Alpine: Progressive itemset mining with definite guarantees. In: Proceedings of the International Conference on Data Mining. pp. 63–71. SIAM (2017)
12. Kuznetsov, S.O.: A fast algorithm for computing all intersections of objects from an arbitrary semilattice. Nauchno-Tekhnicheskaya Informatsiya Seriya 2-Informatsionnye Protsessy i Sistemy (1), 17–20 (1993)
13. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. Journal of Experimental & Theoretical Artificial Intelligence **14**(2-3), 189–216 (2002)

14. Makhalova, T., Kuznetsov, S.O., Napoli, A.: On coupling FCA and MDL in pattern mining. In: Proceedings of the 15th International Conference on Formal Concept Analysis. pp. 332–340. Springer (2019)
15. Mampaey, M., Vreeken, J., Tatti, N.: Summarizing data succinctly with the most informative itemsets. ACM Transactions on Knowledge Discovery from Data **6**(4), 16 (2012)
16. Piatetsky-Shapiro, G.: Discovery, analysis, and presentation of strong rules. Knowledge Discovery in Databases pp. 229–238 (1991)
17. Smets, K., Vreeken, J.: Slim: Directly mining descriptive patterns. In: Proceedings of the International Conference on Data Mining. pp. 236–247. SIAM (2012)
18. Uno, T., Asai, T., Uchida, Y., Arimura, H.: An efficient algorithm for enumerating closed patterns in transaction databases. In: Proceedings of the 7th International Conference on Discovery Science. pp. 16–31. Springer (2004)
19. Vreeken, J., Tatti, N.: Interesting patterns. In: Aggarwal, C.C., Han, J. (eds.) Frequent Pattern Mining, pp. 105–134. Springer (2014)
20. Vreeken, J., Van Leeuwen, M., Siebes, A.: Krimp: mining itemsets that compress. Data Mining and Knowledge Discovery **23**(1), 169–214 (2011)
21. Webb, G.I.: Self-sufficient itemsets: An approach to screening potentially interesting associations between items. ACM Transactions on Knowledge Discovery from Data **4**(1), 1–20 (2010)
22. Zaki, M.J.: Generating non-redundant association rules. In: Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining. pp. 34–43. ACM SIGKDD (2000)