Comparing Performance of Formal Concept Analysis and Closed Frequent Itemset Mining Algorithms on Real Data

Lenka Pisková, Tomáš Horváth

University of Pavol Jozef Šafárik, Košice, Slovakia
lenka.piskova@student.upjs.sk, tomas.horvath@upjs.sk

Abstract. In this paper, an experimental comparison of publicly available algorithms for computing intents of all formal concepts and mining frequent closed itemsets is provided. Experiments are performed on real data sets from UCI Machine Learning Repository and FIMI Repository. Results of experiments are discussed at the end of the paper.

1 Introduction

Formal Concept Analysis (FCA) [9] is a method for analysing data in the form of a table with applications in many disciplines.A formal concept is a formalization of the concept of a "concept" which consists of two parts, a set of objects which forms its extension and a set of attributes which forms its intension [25]. Formal concepts can be ordered according to the subconcept-superconcept relation resulting in a concept lattice.

Frequent itemset mining (FIM) introduced in [1] was proposed as a method for market basket analysis. The identification of sets of items (itemsets) which often occur together in a database (the frequency is not less than a user defined minimum support threshold) is one of the basic tasks in Data Mining. When the minimum support is set low, a huge number of itemsets is generated. To overcome this problem, closed and maximal frequent itemsets were proposed.

FCA and FIM are two research fields that are closely related to each other [20]. Naturally, they address similar problems, e.g. selecting important concepts versus finding interesting patterns in data. Moreover, they inspire each other (Iceberg concept lattice which is the set of all frequent concepts connected with the subconcept-superconcept relation [23]).

Finding the set of all intents (of formal concepts) is equivalent to finding the set of all closed frequent itemsets using a minimum support equal to zero [20]. Nonetheless, there is no experimental comparison between algorithms for computing formal concepts and algorithms for mining frequent closed itemsets. The aim of this paper is to provide such comparison on real-world data.

2 Compared Algorithms

The problem of generating formal concepts and/or a concept lattice has been well studied and many algorithms have been proposed [8], [15], [17], [21], [22]. A

 Published in Manuel Ojeda-Aciego, Jan Outrata (Eds.): CLA 2013, pp. 299–304, ISBN 978–2–7466–6566–8, Laboratory L3i, University of La Rochelle, 2013.

comparative performance study of algorithms for building concept lattices can be found in [10] and [16]. In this paper we will focus only on those algorithms which compute the set of all formal concepts (frequent closed itemsets) only. Therefore, we do not compare our results with [10] and [16].

The fastest algorithms for computing formal concepts are FCbO [14] and In-Close [2] which are based on the CbO algorithm [15]. In the competition between FCA algorithms at ICCS 2009[1] FCbO took the first place and the runner-up was In-Close. The improvement of In-Close algorithm [3] was developed in response to the competition to outperform FCbO, but our results show that FCbO still performs better. A parallel variant of FCbO was also proposed [14], however, we consider only the serial version in this paper.

Implementations of algorithms for mining closed frequent itemsets were experimentally compared[2] and presented at FIMI'03 and FIMI'04 workshops [12]. The best of the tested algorithms ([5], [13], [18], [19], [24], [26]) was FP-Close [13] although it gave a segmentation fault for 4 out of 14 data sets.

In this paper, we provide an experimental comparison of 10 algorithms on real-world data sets whose implementations are publicly available, two of them compute formal concepts (FCbO and In-Close2) and the remaining 8 generate closed frequent itemsets ([5], [6], [7], [13], [18], [19], [24]).

3 Experimental Evaluation

We have carried out a number of experiments for several real-world data sets to compare FCA and FIM algorithms that are publicly available. The characteristics of selected data sets [4], [11] are shown in the table 1.

Table 1. The characteristics of data sets.

Dataset	# Transactions	# Items	Density (%)	Small/Big
Accidents	340183	468	33.8	Big
Car Evaluation	1728	25	28	Small
Connect	67557	129	43	Big
Kosarak	990002	41270	8.1	Big
Mushroom	8124	119	23	Small
Retail	88162	16469	10.3	Big
Tic-tac-toe	958	29	34	Small

The experiments were conducted on the computing node with 16 cores equipped with 24 GB RAM memory running GNU/Linux openSUSE 12.1.

The measured times are CPU times. Each algorithm was run three times for each data set and the given minimum support threshold value to get the most accurate results. All reported times are the average times of the three

[1] http://www.upriss.org.uk/fca/fcaalgorithms.html
[2] http://fimi.ua.ac.be/experiments/

runs. The output was turned off, i.e. the results of algorithms (intents of formal concepts/frequent closed itemsets) were neither written to a file nor to the screen.

Some of the algorithms were originally developed to mine closed frequent itemsets. For Apriori, Carpenter and Eclat we have set the flag -tc to mine closed frequent itemsets. Similarly, we have used the flag -fci for Mafia.

The input file of In-Close2 is in the cxt (formal context) format while the input of other algorithms is in the standard FIMI format - each line represents a list of attributes of an object/a list of items in a transaction. The disadvantage of In-Close2 is that unlike other algorithms it also computes extents of formal concepts (in addition to their intents).

Some algorithms had problems on certain data sets. For mushroom, Apriori gets killed, Carpenter outputs an incorrect number of closed frequent itemsets (238827), DCI_Closed does not calculate the result in a reasonable time (we have stopped the computation after a few hours). In-Close2 gives an incorrect number of formal concepts (59343) for tic-tac-toe.

For kosarak, FPClose is either aborted due to the invalid pointer or gives segmentation fault for the support 0.8% and supports less than or equal to 0.6%. For retail with $minsup = 0$, Apriori gets killed, Carpenter gives an incorrect number of closed frequent itemsets (2186693) and DCI_Closed is aborted. In-Close2 gives segmentation fault for the supports lower or equal to 60% on accidents and for all supports except for 90% on connect.

Table 2. Performance of algorithms for mining closed frequent itemsets with the minimum support equal to 0 (CPU time in seconds).

# Formal concepts	Car Evaluation 12640	Mushroom 238710	Tic-tac-toe 59505
Afopt	0.13	5.083	1.006
Apriori	0.04	-	0.25
Carpenter	0.71	-	1.646
DCI_Closed	0.023	-	0.05
Eclat	0.02	0.976	0.143
FCbO	0.02	0.803	0.13
FPClose	0.056	1.586	0.36
In-Close2	0.043	2.583	-
LCM	0.02	1.363	0.086
Mafia	0.243	39.746	2.64

We have compared the performance of the algorithms for mining intents of all formal concepts, i.e. closed frequent itemsets using $minsup = 0$ (typical task in FCA) on small data sets. The results are depicted in the table 2. Arguably, FCbO is the best algorithm for the given task, it is the fastest algorithm for car and mushroom and the third fastest for tic-tac-toe. LCM and Eclat perform well on these data sets, too. Considering also the results on retail with $minsup = 0$, LCM is the fastest algorithm and the runner-up are Eclat and FPClose.

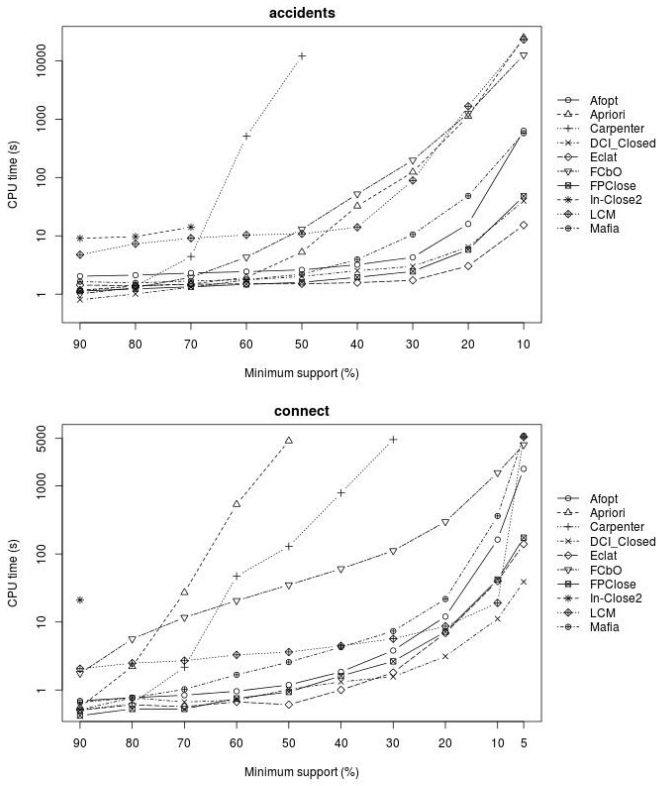

Fig. 1. Performance of algorithms on *accidents* and *connect* data sets for various minimum support values (CPU time in seconds are measured).

Other tests were performed on big data sets and the performance of algorithms was tested for various values of support. Figures 1 and 2 show the timings for the algorithms on the accidents, connect, kosarak and retail data sets. The performance of the FCbO algorithm is average on big data sets except for kosarak. Eclat, DCI_Closed and FPClose are good choice in the case of dense data sets (accidents, connect). However, the runtime of most algorithms increases dramatically with decreasing minimum support on these data sets. Afopt, DCI_Closed and LCM are suitable for sparse data sets (kosarak, retail) although the Afopt algorithm is not able to handle the retail data set for $minsup = 0$. The timings for In-Close2 on kosarak are not included, because the computation took several hours just for high values of support .

For kosarak, in our experimental testing FPClose failed while FPClose was the fastest algorithm for low as well as high values of support in [12]. Our results on other data sets correspond to some extent to the results in [12].

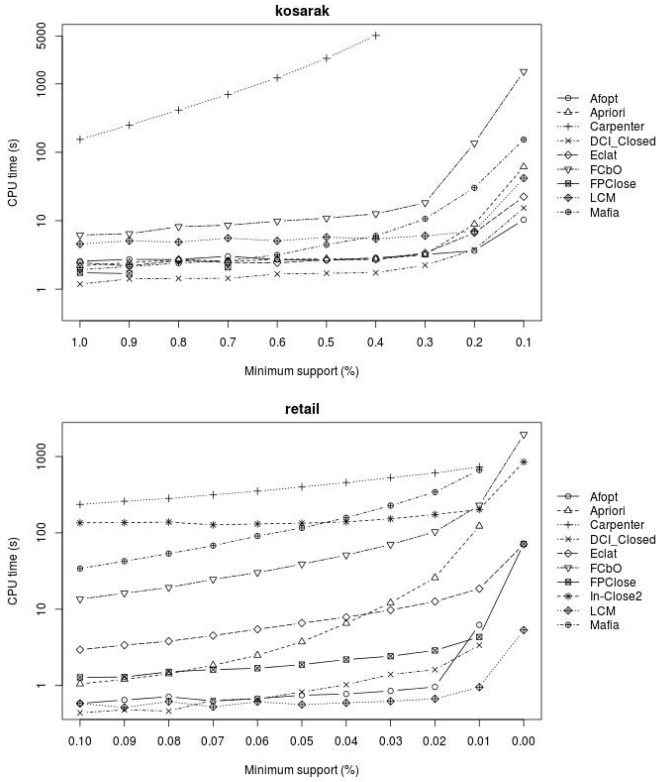

Fig. 2. Performance of algorithms on *kosarak* and *retail* data sets for various minimum support values (CPU time in seconds are measured).

4 Conclusion

We have experimentally compared algorithms for computing intents of formal concepts and algorithms for mining closed frequent itemsets on real-world data. Our experimental testing has no clear winner for different data sets and minimum support threshold setting. In our opinion, DCI_Closed behaves well although it had some problems on the mushroom data set and the retail data set with $minsup = 0$. On small data sets, the fastest algorithm is FCbO.

Acknowledgements: This work was partially supported by the research grants VEGA 1/0832/12 and the Center of knowledge and information systems in Košice (ITMS 26220220158).

References

1. R. Agrawal, T. Imielinski, A. Swami: Mining association rules between sets of items in large databases. SIGMOD, 1993.

2. S. Andrews: In-Close, a fast algorithm for computing formal concepts. ICCS 2009.
3. S. Andrews: In-Close2, a High Performance Formal Concept Miner. ICCS 2011.
4. K. Bache, M. Lichman: UCI Machine Learning Repository, 2013, http://archive.ics.uci.edu/ml, University of California, Irvine, School of Information and Computer Sciences.
5. Ch. Borgelt: Efficient Implementations of Apriori and Eclat. IEEE ICDM Workshop on FIMI (2003).
6. Ch. Borgelt, X. Yang, R. Nogales-Cadenas, P. Carmona-Saez, A. Pascual-Montano: Finding Closed Frequent Item Sets by Intersecting Transactions. EDBT 2011.
7. D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, T. Yiu: MAFIA: A Performance Study of Mining Maximal Frequent Itemsets. IEEE ICDM Workshop on FIMI (2003).
8. B. Ganter: Two basic algorithms in concept analysis. (Technical Report FB4-Preprint No. 831). TH Darmstadt, 1984.
9. B. Ganter, R. Wille: Formal concept analysis: Mathematical foundations. Springer, 1999.
10. R. Godin, R. Missaoui, H. Alaoui: Incremental concept formation algorithms based on Galois (concept) lattice. Computational Intelligence, vol. 1(2), 1995.
11. FIMI Repository, http://fimi.ua.ac.be/data/.
12. B. Goethals, M. J. Zaki: Advances in Frequent Itemset Mining Implementations: Report on FIMI03. SIGKDD Explorations, vol. 6 (1), 2004.
13. G. Grahne, J. Zhu: Efficiently Using Prefix-trees in Mining Frequent Itemsets. IEEE ICDM Workshop on FIMI (2003).
14. P. Krajca, J. Outrata, V. Vychodil: Advances in algorithms based on CbO. CLA 2010.
15. S. O. Kuznetsov: A Fast Algorithm for Computing All Intersections of Objects in a Finite Semi-lattice. Automatic Documentation and Mathematical Linguistics, vol. 27 (5), 1993.
16. S. O. Kuznetsov, S. A. Obiedkov: Comparing Performance of Algorithms for Generating Concept Lattices. Journal of Experimental and Theoretical Artificial Intelligence, vol. 14 (2-3), 2002.
17. Ch. Lindig: Fast Concept Analysis. Working with Conceptual Structures – Contributions to ICCS 2000, 2000.
18. G. Liu, H. Lu, J. X. Yu, W. Wei, X. Xiao: AFOPT: An Efficient Implementation of Pattern Growth Approach. IEEE ICDM Workshop on FIMI (2003).
19. C. Lucchese, S. Orlando, R. Perego: DCI_Closed: A Fast and Memory Efficient Algorithm to Mine Frequent Closed Itemsets. IEEE ICDM Workshop on FIMI (2004).
20. B. Martin, P. Euklund: From Concepts to Concept Lattice: A Border Algorithm for Making Covers Explicit. ICFCA, 2008.
21. E. M. Norris: An Algorithm for Computing the Maximal Rectangles in a Binary Relation. Revue Roumaine de Mathématiques Pures et Appliquées, vol. 23(2), 1978.
22. L. Nourine, O. Raynaud: A fast algorithm for building lattices. Information Processing Letters, vol. 71, 1999.
23. G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal: Computing Iceberg Concept Lattices with Titanic. Journal on Knowledge and Data Engineering, vol. 42(2), 2002.
24. T. Uno, T. Asai, Y. Uchida, H. Arimura: LCM: An Efficient Algorithm for Enumerating Frequent Closed Item Sets. IEEE ICDM Workshop on FIMI (2003).
25. R. Wille: Restructuring lattice theory: An approach based on hierarchies of concepts. Ordered Sets, vol. 83, 1982.
26. M. J. Zaki: Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engeneering, vol. 12, 2000.