Frequent closed itemset based algorithms: A thorough structural and analytical survey

Authors replies to the comments and suggestions of the reviewers

First of all, we like to express our gratitude for the useful comments of the referees. We tried to comply with their suggestions as far as possible.

S. Ben Yahia, T. Hamrouni, and E. Mephu Nguifo
Tunis/Lens February 27th 2006

Reviewer 1

1. The paper does a decent job of comparing the FIMI methods (Charm, FP-Close, LCM), etc, and some of the methods from the FCA community (such as Close, Titanic, etc.). What I found missing is some of the other classic algorithms in FCA like Next-Closure of Ganter, and other methods studied in the paper by Kuznetsov. It would be really great if some of those classic methods can be added to this paper, since that has not been done previously, and is sorely needed.

   Answer: We think that even if Close and Titanic can be considered from the FCA community, they benefit by a great performance improvement using the minimum support pruning strategy. Indeed, imposing a statistical constraint was of help to sweep a large search space, especially for low minsup requirements. Furthermore, these two algorithms do not save the extension part of a given formal concept (its extension is reduced to its cardinality) and hence are able to save memory as much as possible. In contrary, classic algorithms in the FCA like Next-Closure, Bordat or Chein algorithms, etc, are devoted to closed itemsets and not to frequent closed itemsets (FCIs). Hence, they extract all formal concepts of a given dataset since the minimum support threshold *minsup* is not of use. This makes them really in the need of a very important amount of memory if we run them on datasets like we used in our survey. This fact is argued by the dataset choice made by for example, on the one hand, Kuznetsov and Ob"edkov [4] and, on the other hand, Fu and Mephu Nguifo [2]. Indeed, the former used small randomly generated datasets while the latter compared them on relatively small datasets from the UC Irvine Machine Learning Database Repository[1].

2. The English is a bit awkward in several places. A careful re-reading is required to correct some sentences.

   Answer: This remark was taken into consideration. A thorough review helped us to clear up grammar and phrasing problems.

3. Table 1 is not really understandable, in that I am not sure what it is trying to classify. A clearer presentation and where each algo belongs needs to be discussed.

   Answer: Table 1 tries to decompose, into 10 features, the structural differences among the four categories of FCI based algorithms (described in pages 2 and 3). Another entry is added showing the belonging of each algorithm to the associated category.

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html.

4. The Authors claim in section 2 that the Hasse diagram can be built only for Titanic, DCI-Closed and LCM. I think this statement is not correct. For example, Charm-L (Zaki, tkde 2005) explicitly builds the lattice. Also I don't see what is the big deal about adding the empty min-gen concept. That is a straightforward thing to do and thus any lattice building algorithm can be used to build the lattice.

   Answer: Since there are some cases where the closure of the empty min-gen concept is not null (like for the Mushroom dataset downloadable from `http//fimi.cs.helsinki.fi/data` where its closure is equal to the item "85"), additional treatments are required for algorithms not considering the empty set as a minimal generator (like for the Close algorithm) to find its real closure. The corresponding statement was written to remove the ambiguity.

5. While the experimental comparison is quite detailed, I think a few key characteristics of the algorithms, which are responsible for their runtimes, should be more explicitly spelt out.

   Answer: This remark was taken into consideration in the revised version.

6. For T10I4 data, the authors should check why LCM does not match the correct output? The diff for Titanic and DCI-Closed is explained.

   Answer: For the T10I4D100K dataset and for a *minsup* equal to 0.02%, the LCM algorithm gives two "incorrect" FCIs (in comparison with the output of the other algorithms). We tried to check why it does not match the correct output. However, we encountered two main problems:
   1. The large number of extracted FCIs (more than 100 thousands), so we were not able to check it by hand.
   2. Using LCM with the option allowing to write the output on a disk-resident file, LCM sometimes writes a space to indicate the item "0".

Reviewer 2

1. The main comparison is against speed and memory using some of the FIMI data sets, which makes most of the proposed comparison already published and the results are known especially from FIMI 2003 and FIMI 2004 workshops. The authors did not try to test algorithms that are not already tested in FIMI or at least test new datasets that never been reported the same workshop. (It is true that there were some test on the worst case artificially made datasets, but yet this did not add any big values as most of the existing algorithms behaved very badly in such cases, which is expected.

   Answer: It is true that the datasets we used in this survey are from the FIMI web site. However, this can be explained by the fact that almost all recent papers use such datasets in their performance assessments. Hence, these datasets become standard ones. However, in FIMI 2003 and FIMI 2004 workshops, only performance curves are shown which is not the aim of our survey. Indeed, we tried not to limit our survey to curves but mainly to give the real face of each algorithm (or in general of each category of algorithms). About worst case datasets, we believe that such dataset represent a successful way to test algorithm scalability since they are not randomly or synthetically generated. Indeed, when we pass from a worst case dataset of size $n\times(n+1)$ to another of size $(n+1)\times(n+2)$, we know that the number of FCIs grows up exactly by a factor of 2. Hence, using such datasets, we can know if a given algorithm performs exponentially or linearly with the evolution of the FCI number. Such fact confirms, for example, that the runtime of DCI-Closed and LCM is a linear function in the number of FCIs.

2. Authors mentioned in many places, how some algorithms use lots of memory, however they applied most of their algorithms in the same hardware. Readers would be interested in knowing the behavior

changes once the same experiments are executed on machines with limited memory sizes, or at least the effect of changing memory sizes on the behavior of these algorithms.

Answer: As mentioned by Goethals and Zaki [3], only minor differences happen when we use different machines. This can be explained by the fact that the memory consumption of a given algorithm is, in general, constant: the more the machine has a RAM capacity, better algorithm performance are since this delays as much as possible the use of the swap space (known to be slower than the RAM space).

3. FIMI workshop never recorded the behavior of such algorithms while mining extremely large data bases (i.e. millions or even hundreds of millions of transactions). Applying such tested will add real values for this survey.

   Answer: In this survey, we tried to use the most common datasets of the different performance studies that recently appeared. Indeed, we tried to give an external point of view of the surveyed algorithms on datasets that were run before (in their respective papers or in others). Hence, this survey can be considered as a checkpoint of the most known FCI-based algorithms on the most used datasets in the performance studies. Nevertheless, for extremely large datasets, a recent algorithm called COFI-Closed [1] was proposed to mainly tackle such datasets.

4. The survey is exclusive in the sense that some new algorithms are not covered. For example one method is authored by one of the editors of Explorations. I wonder if there are others. How do they differ? Why are they excluded?
   M. El-Hajj, O. Zaiane, Finding All Frequent Patterns Starting From The Closure, International Conference on Advanced Data Mining and Applications, pp 67-74, Wuhan, China, July 2005.

   Answer: This algorithm is mentioned in the survey.

Reviewer 3

1. Abstract. "go further beyond the top of the Iceberg". What do you really mean here?

   Answer: We mean that Data mining tools allow to extract by far more valuable information than traditional tools. The abstract was improved.

2. Section 3 should be shortened and summarized. The current text repeats the curves in many places. The paper should summarize and highlight the differences among the methods.

   Answer: Actually, we find that Section 3 presents the core of the paper. It tries to explain the main differences of the algorithms by scrutinizing their performances on regular intervals of minimum support. This performance analysis is presented using a conjunction of different types of information, *e.g.*, utilized heuristics, number of candidates and amount of handled main memory.

3. In the abstract, it is better to mention the algorithms in four categories on two main groups of benchmarks, sparse and dense.

   Answer: This remark was taken into consideration in the revised version.

4. In the introduction, the authors use the terms "equivalence classes" and "minimal generators". Brief introduction of frequent closed itemset, equivalence classes and minimal generators may help to make the paper self-contained. In addition, the term "upper cover" is used without any explanation. (page1, second column).

Answer: This remark was taken into consideration. Indeed, a new section having for purpose to recall such notions is added.

5. For analyzing the results, it would be better to use the characteristics mentioned before for each category. For example, instead of mentioning that DCI and LCM both outperform Charm according to their duplicate detection strategy, it is better to mention that they outperform Charm according to their characteristic mentioned before on detecting the redundant closure computation where they use order preserving strategy. Instead of comparing the algorithms they can compare the categories with each other.

   Answer: This remark was taken into consideration in the revised version.

6. The direct comparison between CLOSET+ on Windows and other methods on Linus should be justified. Windows platform may introduce a higher cost in both runtime and main memory usage.

   Answer: Answer: It is true that most recent algorithms are designed for a Linux platform. However, we were obliged to run the Closet+ algorithm on a Windows distribution since only a Windows binary executable was provided by its authors.

7. There are a few previous surveys highly related to this paper.
   T. Calders, C. Rigotti, and J-F. Boulicaut. A Survey on Condensed Representations for Frequent Sets. In Constraint-Based Mining; Springer; Vol. 3848
   Charu C. Aggarwal: Towards Long Pattern Generation in Dense Databases. SIGKDD Explorations 3(1): 20-26 (2001).

   Answer: We added both papers since they are closely related to the survey.

8. Reference [17] is wrong.

   Answer: The reference of the Closet algorithm has been corrected.

References

1. M. El-Hajj and O. Zaiane. Finding all frequent patterns starting from the closure. In *the International Conference on Advanced Data Mining and Applications, Wuhan, China*, pages 67–74, July 2005.
2. H. Fu and E. Mephu Nguifo. Etude et conception d'algorithmes de génération de concepts formels. In J.-F. Boulicault and B. Crémilleux, editors, *Revue d'Ingénierie des Systèmes d'Information (ISI), Hermès-Lavoisier*, volume 9, pages 109–132, 2004.
3. B. Goethals and M. J. Zaki. FIMI'03: Workshop on frequent itemset mining implementations. In B. Goethals and M. J. Zaki, editors, *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI 2003)*, volume 90 of *CEUR Workshop Proceedings, Melbourne, Florida, USA*, 19 November 2003.
4. S. O. Kuznetsov and S. A. Ob"edkov. Comparing performance of algorithms for generating concept lattices. In *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)*, volume 14(2-3), pages 189–216, 2002.