CORON: A Framework for Levelwise Itemset Mining Algorithms

Laszlo Szathmary and Amedeo Napoli

LORIA, Campus Scientifique - BP 239
54506 Vandœuvre-lès-Nancy Cedex, France
{szathmar, napoli}@loria.fr

Abstract. CORON[1] is a framework for levelwise algorithms that are designed to find frequent and/or frequent closed itemsets in binary contexts. Datasets can be very different in size, number of objects, number of attributes, density, etc. As there is no one best algorithm for arbitrary datasets, we want to give a possibility for users to try different algorithms and choose the one that best suits their needs.

Keywords: frequent itemsets, closed frequent itemsets, itemset mining, algorithms, association rules, data mining.

1 Introduction

Finding association rules is one of the most important tasks in data mining [1]. It is possible to identify hidden relations based on itemsets in large databases. However, for generating association rules, first we need to discover frequent and/or frequent closed itemsets (FIs/FCIs). As input data, like in formal concept analysis [2], we consider a binary relation between a set of objects (O) and a set of attributes (A). A formal *context* is a triple (O, A, R), where $R \subseteq O \times A$. The notation $R(o, a)$ means that the object o has the attribute a. An *itemset* is an arbitrary set of attributes. The *support* of an itemset indicates how many objects are described by the itemset. An itemset is called *frequent* if its support exceeds a given *minimal support* threshold. An itemset X is called *frequent closed* if there exists no proper superset Y ($X \subset Y$) with the same support. The set of frequent closed itemsets is a much smaller subset of frequent itemsets. Furthermore, FCIs are a lossless representation of FIs, because it is possible to find all FIs from the set of FCIs.

The problem is whether to 1) find all FIs, or 2) find all FCIs, or 3) find both FIs and FCIs. The CORON framework exactly addresses this problem by providing different levelwise algorithms for each problem. CORON has the following main characteristics. First, any of its algorithms can be called from command line as a standalone program. Second, due to its Java API, it can easily be integrated in other projects. Three, since CORON is a framework, large parts of it can be reused to implement a new levelwise algorithm.

[1] This research work is carried out in the frame of the French-Hungarian research program Balaton (Balaton F-23/03).

2 Algorithms

The currently implemented variations comprise the following algorithms: Apriori, Apriori-Close, Close, Pascal and Titanic. The following table (Table 1) shows the main properties of the algorithms.

	FI	FCI	count. inf.
Apriori	X		
Apriori-Close	X	X	
Close		X	
Pascal	X		X
Titanic		X	X

Table 1. FI – discovers frequent itemsets, FCI – discovers frequent closed itemsets, count. inf. – uses counting inference

Apriori [3] is the base for most of the proposed itemset search algorithms. Apriori performs a levelwise search and is based on two principles. First, every subset of a frequent itemset is also frequent (also called *downward closure* property). Second, every superset of a nonfrequent itemset is also nonfrequent. Apriori-Close [4] is an extension of the original Apriori algorithm and allows the computation of frequent and frequent closed itemsets simultaneously. It also starts enumerating frequent itemsets, and in the i^{th} iteration it can determine frequent closed itemsets among the itemsets produced in the $(i-1)^{th}$ iteration. Close [5], instead of mining the complete set of frequent itemsets, only finds the frequent closed itemsets. Close has proved to be more efficient for dense datasets. Pascal [6] introduces the notion of *key patterns* and shows that the support of lots of frequent patterns can be inferred from their key patterns. As a consequence, Pascal can reduce significantly the number of database passes. Titanic [7], similarly to Pascal, uses key patterns too to find frequent closed itemsets.

3 Significance of Itemset Mining

Frequent (closed) itemsets can be used for different purposes as explained hereafter.

3.1 Building Concept Lattices

Frequent closed itemset mining can be used for the construction of concept (or Galois) lattices. By finding the FCIs we obtain the *intents* of the corresponding concepts. The CORON framework is extended with the feature to find the *extent* part of the intents, and find the order between the previously identified concepts. This approach can be used to build both complete and iceberg concept lattices [7]. The higher the minimum support is, the less concepts the iceberg lattice has. Changing the support various-sized iceberg concept lattices can be constructed.

Though CORON has a command-line interface and the built concept/iceberg lattices cannot be visualized yet, it is still possible to process and work with the lattices in the memory.

3.2 Association Rules

In [3], Agrawal et al. showed how to generate association rules from FIs. The problem is that this method produces a huge number of association rules, leading to a new data mining problem, namely "rule mining". Recent studies [5] have shown that association mining only needs to find FCIs and their corresponding rules. This way less redundant and more valuable rules are found, since the larger set of association rules generated from FIs can be inferred from this smaller set.

Another project in our team, called ASSRULEX (Association Rule eXtractor) and built on CORON, allows the generation of association rules from both frequent and frequent closed itemsets.

Let us consider the following toy dataset: $D = \{t_1, t_2, t_3, t_4, t_5\}$, where $t_1 = \{a, c, d\}$, $t_2 = \{b, c, e\}$, $t_3 = \{a, b, c, e\}$, $t_4 = \{b, e\}$, $t_5 = \{a, b, c, e\}$. Using for instance CORON's Apriori-Close algorithm with minimum support 3 (60%), we get 9 frequent and only 4 frequent closed itemsets. The FIs: {a (3), b (4), c (4), e (4), ac (3), bc (3), be (4), ce (3), bce (3)}; the FCIs: {c (4), ac (3), be (4), bce (3)}, where the support of the itemsets is indicated in parenthesis.

The following table (Table 2) shows the different association rule sets generated from the previous frequent and frequent closed itemsets. The whole set was produced from the FIs (14 rules), while rules indicated by a "+" sign are produced from the FCIs (10 rules). As we can see, the rules {c} => {b} and {c} => {e} for instance are redundant, because they can be inferred from the rule {c} => {b, e}.

{c, e} => {b} (supp=3; conf=1,000) +	{c} => {b} (supp=3; conf=0,750)
{b, e} => {c} (supp=3; conf=0,750) +	{b} => {c} (supp=3; conf=0,750)
{b, c} => {e} (supp=3; conf=1,000) +	{e} => {b} (supp=4; conf=1,000) +
{e} => {b, c} (supp=3; conf=0,750) +	{b} => {e} (supp=4; conf=1,000) +
{c} => {b, e} (supp=3; conf=0,750) +	{e} => {c} (supp=3; conf=0,750)
{b} => {c, e} (supp=3; conf=0,750) +	{c} => {e} (supp=3; conf=0,750)
	{c} => {a} (supp=3; conf=0,750) +
	{a} => {c} (supp=3; conf=1,000) +

Table 2. Association rules generated from the dataset D with minimum support 3 (60%) and minimum confidence 75%. The "+" sign indicates that the rule was produced from an FCI.

4 Development Environment and Experiments

CORON is implemented entirely in Java, giving full portability to the framework. We have chosen this language for several reasons. First, one of our aims is to create a

software that can function as a base for other itemset/rule miner projects. Second, Java is flexible enough to allow the extension of the framework easily.

Most researchers in our team use Java, and CORON has already been used for instance in the KASIMIR knowledge-based system [8]. Currently we are making experiments with CORON and ASSRULEX on a real-life biological database [9]. Because of the large size of this latter dataset we had to develop two kinds of filters. The first one is a pre-processor to reduce the size of the database using either horizontal or vertical projections. The second is a post-processor for the association rules to delete/keep rules that satisfy some conditions.

5 Perspectives

In the near future we plan to add some other algorithms to the framework and then we want to make a detailed comparative study of the different algorithms. We also want to continue the development of ASSRULEX to identify different bases, such as the Duquenne-Guigues basis for exact, or the Luxenburger basis for approximate association rules.

References

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco (2001)
2. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer, Berlin/Heidelberg (1999)
3. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Advances in knowledge discovery and data mining. American Association for Artificial Intelligence (1996) 307–328
4. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Closed set based discovery of small covers for association rules. In: Proc. 15emes Journees Bases de Donnees Avancees, BDA. (1999) 361–381
5. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient mining of association rules using closed itemset lattices. Inf. Syst. **24** (1999) 25–46
6. Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: Mining frequent patterns with counting inference. SIGKDD Explor. Newsl. **2** (2000) 66–75
7. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing Iceberg Concept Lattices with TITANIC. Data and Knowledge Engineering **42** (2002) 189–222
8. d'Aquin, M., Brachais, S., Lieber, J., Napoli, A.: Decision support and knowledge management in oncology using hierarchical classification. In Kaiser, K., Miksch, S., Tu, S.W., eds.: Proceedings of the Symposium on Computerized Guidelines and Protocols - CGP-2004, Prague, Czech Republic. Volume 101 of Studies in Health Technology and Informatics., Miksch, S. and Tu, S. W., IOS Press (2004) 16–30
9. Maumus, S., Napoli, A., Szathmary, L., Visvikis-Siest, S.: Exploitation des données de la cohorte STANISLAS par des techniques de fouille de données numériques et symboliques utilisées seules ou en combinaison. In: Workshop on Fouille de données complexes dans un processus d'extraction des connaissances - EGC 2005, Paris, France (to appear). (2005)