Open Source Data Mining: Workshop Report

Bart Goethals
University of Antwerp
Antwerp, Belgium
bart.goethals@ua.ac.be

Siegfried Nijssen
LIACS, Universiteit Leiden
Leiden, The Netherlands
snijssen@liacs.nl

Mohammed J. Zaki
Rensselaer Polytechnic Institute
Troy, NY, USA
zaki@cs.rpi.edu

1. INTRODUCTION

Over the past decade tremendous progress has been made in data mining methods like clustering, classification, frequent pattern mining, and so on. Unfortunately, however, the advanced implementations are often not made publicly available, and thus the results cannot be independently verified. We believe that this hampers the rapid advances in the field. With this workshop we intended to promote open source data mining (OSDM) by creating a first meeting place to discuss open source data mining methods.

The first steps towards an open source data mining workshop were set in previous years by the Frequent Itemset Mining Implementations workshops (FIMI), which enjoyed a large popularity [1; 2]. The OSDM workshop was held in the same spirit as these earlier workshops, and, in its first edition, the workshop therefore had a special focus on implementations of frequent pattern mining algorithms. We hope that in the next years the workshop will also focus on open source implementations for other data mining problems like clustering, classification, outlier detection, and so on.

Frequent pattern mining is a core field of research in data mining encompassing the discovery of patterns such as itemsets, sequences, trees, graphs, and many other structures. Varied approaches to these problems appear in numerous papers across all data mining conferences. Generally speaking, the problem involves the identification of items, products, symptoms, and so forth, that often occur together in a given dataset. As a fundamental operation in data mining, algorithms for FPM can be used as a building block for other, more sophisticated data mining processes. During the last decade, a huge number of algorithms have been developed in order to efficiently solve all kinds of FPM problems.

All submissions to this workshop were necessarily accompanied by source code. This source code can also be found on the homepage of the OSDM 2005 workshop:

`http://osdm.ua.ac.be/`.

All papers were independently reviewed by the members of the program committee, consisting of: Charu Aggarwal, Christian Borgelt, Mohammad El-Hajj, Lawrence B. Holder, Akihiro Inokuchi, George Karypis, Sergei Kuznetsov, Sergei Obiedkov, Jian Pei, Hannu Toivonen, and Takeaki Uno. We wish to thank all members of this committee for their effort. Also, we would like to thank all the authors, the invited speaker, and all attendees for contributing to the success of the workshop.

2. KEYNOTE TALK

Geoff Webb gave an invited talk about "Finding the Real Patterns". Pattern discovery typically explores a massive space of potential patterns to identify those that satisfy some user-specified set of criteria. This process entails a huge risk (in many cases a near certainty) that many patterns will be false discoveries. These are patterns that satisfy the specified criteria with respect to the sample data but do not satisfy those criteria with respect to the population from which those data are drawn. The talk discussed the problem of false discoveries, and presented techniques for avoiding them.

3. CONTRIBUTIONS

The workshop started off with a paper about "Benchmarking Frequent Itemset Mining Algorithms", by Balázs Rácz, Ferenc Bodon and Lars Schmidt-Thieme. Traditionally, publications about frequent pattern mining have a large stress on the efficiency of algorithms. To prove that an algorithm is efficient, it is common practice to perform experiments in which the run-time behavior of several frequent pattern mining implementations are compared. Rácz *et al.* argued that such results should be considered with caution. They showed that differences in implementation can sometimes have a large influence on run-time behavior: FP-Trees stored in arrays sometimes result in almost 10 times faster execution than FP-Trees stored using separately allocated objects. Rácz *et al.* therefore argued that it could be beneficial to develop a frequent pattern mining library; this would allow for a better comparison of the merits of *algorithms* in stead of *implementations*. The authors proposed such a library and discussed some of its details; they provided an experimental comparison of several algorithms, obtained through this library.

The author of the next paper, Christian Borgelt with "Keeping Things Simple: Finding Frequent Item Sets by Recursive Elimination", also concentrated on a different aspect than efficiency in his presentation about the Recursive Elimination (RELIM) algorithm. He introduced a new frequent itemset mining algorithm whose main purpose is to be simple. The algorithm is similar to well-known algorithms such as FP-Growth, but uses a simpler datastructure to store transactions during the recursive search. Still, it was shown that an implementation of this new algorithm sometimes outperforms implementations of other well-known algorithms.

In another presentation, "An Implementation of the FP-growth Algorithm", Christian Borgelt elaborated on the details of his implementation of FP-Growth. The distinguish-

ing feature of FP-Growth, in comparison with RELIM, is the FP-Tree. This is an important difference as the construction of FP-Trees through projections is usually the most time consuming part of FP-Growth. Christian Borgelt discussed several alternatives to built FP-Trees — level-wise or branch-wise — and showed that the branch-wise approach seems to be the more favorable.

It is well-known that every datastructure has its own merits — some allow for quick set inclusion tests in dense sets (like bitmaps), while others are more space efficient in the case of sparse sets (like lists). An algorithm which makes a predefined choice for one datastructure, is bound to perform efficient on one kind of dataset, but less efficient on another. This issue was studied by Takeaki Uno, Masashi Kiyomi and Hiroki Arimura, who introduce the 3rd incarnation of their LCM algorithm in "LCM ver 3.: Collaboration of Array, Bitmap and Prefix Tree for Frequent Itemset Mining". This new algorithm tries to choose a suitable datastructure more dynamically, and is therefore shown to perform well on a wider variety of datasets.

For dense datasets the number of frequent itemsets can often become excessively large. In recent years this observation has led to the development of condensed representations. A condensed representation is a compact representation of the set of frequent itemsets, from which still all frequent itemsets can be deduced. One such deduction procedure relies on the *inclusion-exclusion* principle for sets. Although the inclusion-exclusion principle can reduce the number of patterns very effectively, the computation of this condensed representation can be very costly. Bassem Sayra, Dirk Van Gucht and Paul W. Purdom presented results "On the Effectiveness and Efficiency of Computing Bounds on the Support of ItemSets in the Frequent ItemSets Mining Problem". They provide theoretical and practical results involving heuristics for determining the inclusion-exclusion test more efficiently.

Another condensed representation is based on maximal frequent itemsets, which are frequent itemsets that do not have a frequent superset. Mohammad El-Hajj and Osmar R. Zaïane presented their work on "Implementing Leap Traversals of the Itemset Lattice". They introduce a strategy for determining the maximal frequent itemsets more quickly, thus possibly reducing the size of the search space. The approach can be integrated in a modification of FP-Growth, or in the COFI algorithm, which is earlier work of the authors.

As can be seen from the list of presentations up to now, the workshop had a strong focus on mining itemsets. However, also other pattern domains were present at the workshop.

First, there was a presentation by Christial Borgelt, Thorsten Meinl and Michael Berthold about "MOSS: A Program for Molecular Substructure Mining". The database that is mined in this work consists of a set of molecules in a graph representation. The problem is that of discovering frequent subgraphs. To make the patterns more meaningful, the authors consider several extensions of 'basic' frequent subgraph mining: they study the use of wildcard labels and extensions with ring structures. Also the possibility of mining closed subgraphs was discussed.

Details of another graph miner, Subdue, were presented by Nikhil S. Ketkar, Lawrence B. Holder and Diane J. Cook, in "Subdue: Compression-Based Frequent Pattern Discovery in Graph Data". Subdue differs essentially from the other pattern mining algorithms in the sense that it does not concentrate on discovering frequent patterns; rather, it searches for patterns that achieve a high compression of the data, and relies on beam search to find those patterns. The output of Subdue is smaller than that of other pattern mining algorithms, but therefore also less complete. The authors provided an experimental comparison which showed that the number of discovered patterns is indeed smaller, and therefore possibly more useful; the run-time experiments showed however that Subdue required more time to find this smaller set of patterns.

The third pattern domain that was studied at the workshop, was that of mining sequences. The interesting property of this pattern domain is that it is very simple, but, in some cases, still generalizes slightly over the problem of mining frequent itemsets, and thus provides new challenges for frequent pattern mining.

The first paper about sequence mining, "PLWAP Sequential Mining: Open Source Code", by Christie Ezeife, Yi Lu and Yi Liu, provided an extensive comparison between implementations of the PLWAP, WAP and GSP sequence mining algorithms. Both WAP and PLWAP are very similar to FP-Growth; GSP ressembles the original Apriori algorithm. When using FP-Tree-like structures in sequence mining, the challenge is to encode positions and orders of items in the tree. WAP and PLWAP differ mainly in their solutions to this problem. In terms of run-time, the PLWAP algorithm was shown to perform better.

The second paper about sequence mining was presented by Ferenc Bodon, and discussed "A Trie-based APRIORI Implementation for Mining Frequent Item sequences". Whereas the paper of Ezeife *et al.* concentrated mainly on depth-first tree mining, Bodon considered the details of an Apriori-like algorithm. Special attention was devoted to implementation issues, thus taking care of the earlier remark that implementation details can be important.

4. CONCLUSION

The workshop covered a broad range of relevant topics. Among others, implementation issues, condensed representations and varieties in datastructures and pattern domains were studied. Everybody who is working on frequent pattern mining problems has to deal with some of these issues. We believe the workshop was very successfull in bringing all these related topics together, and hope that the open source implementations of the presented algorithms may help many researchers in the development of their own frequent pattern mining algorithms and implementations.

5. REFERENCES

[1] Bart Goethals and Mohammed Javeed Zaki, editors. *FIMI '03, Frequent Itemset Mining Implementations, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA*, volume 90 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[2] Roberto J. Bayardo Jr., Bart Goethals, and Mohammed Javeed Zaki, editors. *FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Brighton, UK, November 1, 2004*, volume 126 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.