Journal of Machine Learning Research 1 (2014) 1-5 Submitted 4/12; Published 10/14

SPMF: a Java Open-Source Pattern Mining Library

Philippe Fournier-Viger[†] PHILIPPE.FOURNIER-VIGER@UMONCTON.CA
Department of Computer Science
University of Moncton, Moncton, NB E1A 3E9, Canada
Antonio Gomariz AGOMARIZ@UM.ES
Department of Information and Communication Engineering
University of Murcia, Murcia 30100, Spain
Ted Gueniche ETG8697@UMONCTON.CA
Azadeh Soltani SOLTANI.AZ@STU-MAIL.UM.AC.IR
Department of Computer Engineering
Ferdowsi University of Mashhad, Iran
Cheng-Wei Wu SILVEMOONFOX@IDB.CSIE.NCKU.EDU.TW,
Vincent S. Tseng TSENGSM@MAIL.NCKU.EDU.TW,
Department of Computer Science and Information Engineering
National Cheng Kung University, Taiwan

Editor: Balázs Kégl

Abstract

We present SPMF, an open-source data mining library offering implementations of more than 55 data mining algorithms. SPMF is a cross-platform library implemented in Java, specialized for discovering patterns in transaction and sequence databases such as frequent itemsets, association rules and sequential patterns. The source code can be integrated in other Java programs. Moreover, SPMF offers a command line interface and a simple graphical interface for quick testing. The source code is available under the GNU General Public License, version 3. The website of the project offers several resources such as documentation with examples of how to run each algorithm, a developer's guide, performance comparisons of algorithms, data sets, an active forum, a FAQ and a mailing list.

Keywords: data mining, library, frequent pattern mining, sequence database, transaction database, open-source

1. Introduction

In this paper, we present SPMF (Sequential Pattern Mining Framework), a data mining library that is specialized in *frequent pattern mining*, an important subfield of data mining that aims at discovering interesting patterns and associations in databases. SPMF is an open-source project, started in 2009 to address the lack of large open-source data mining library specialized in frequent pattern mining. There exist several general purpose open-source data mining libraries such as Weka (Witten et al., 2005), Mahout (Mahout, 2013) and Knime (Knime, 2013), which provide a wide range of data mining techniques. However, they offer a very limited set of algorithms for frequent pattern mining. Weka, Knime and Mahout

offer only a few popular pattern mining algorithms such as Apriori (Agrawal and Srikant, 1994), GSP (Srikant et al., 1996) and FPGrowth (Han et al., 2004). Some specialized platforms like Coron (Coron, 2013), LUCS-KDD (LUCS-KDD, 2013) and Illimine (Illimine, 2013) offer a slightly larger choice of pattern mining algorithms. However, the source code of Coron is not public, Illimine provides the source code of only one of its pattern mining algorithms and LUCS-KDD source code cannot be used for commercial purposes. SPMF provides more than 55 algorithms for pattern mining. Implementations of most of these algorithms can only be found in SPMF. For example, only three algorithms from SPMF appear in Weka and Knime (Apriori, FPGrowth and GSP), only one in Mahout (FPGrowth), two in LUCS-KDD (Apriori, FPGrowth), and eight in Coron. Another related project is Galicia (Galicia, 2013), an open-source software focusing on mining lattice-based patterns and visualizing lattices. It has only one algorithm in common with SPMF. Another distinctive feature of SPMF is that it offers more than 17 algorithms for mining sequential patterns, while Weka and Knime only offer a single algorithm (GSP), and other previously mentioned software offer none. Moreover, note that Galicia, Coron, Illimine and LUCS-KDD are projects that have been inactive for several years.

Since its first major release in 2010, SPMF has been used in more than 70 research projects in various domains such as web usage mining, analyzing learner behavior in e-learning, clinical text retrieval, sales forecasting, restaurant recommendation, analyzing nucleic acids sequences, anomaly detection in medical treatment and forecasting crime incidents (see the SPMF website for an up-to-date list of applications). Algorithms offered in SPMF can be applied to two main types of data:

- A *transaction database* (a.k.a. *binary context*) is a set of transactions $T = \{T_1, T_2, ... T_n\}$ and a set of items $I = \{i_1, i_2, ...i_m\}$, where $T_x \subseteq I$ for $1 \leq x \leq n$. For example, each transaction of a transaction database could represent a set of items purchased by a customer at a store, or a set of words appearing in a text document.

- A *sequence database* is a generalization of a transaction database. It is a set of sequences $S = \{S_1, S_2, ...S_p\}$ and a set of items $I = \{i_1, i_2, ...i_q\}$. A sequence is a list of transactions $< T_1, T_2, ...T_r >$ where $T_x \subseteq I$ for $1 \leq x \leq r$. Examples of real-life data that can be represented as sequence databases are sequences of webpages visited by users, bioinformatics data (e.g., protein sequences, microarray data and DNA sequences), stock market data, weather observations and sensor data.

Three main data mining tasks can be performed with SPMF.

- *frequent itemset mining* (Agrawal and Srikant, 1994) consists of discovering frequent itemsets, i.e., sets of items appearing in more than *minsup* transactions of a transaction database, where *minsup* is a parameter set by the user.

- *association rule mining* (Agrawal and Srikant, 1994) consists of discovering the association rules respecting some thresholds *minsup* and $minconf$ in a transaction database. An association rule $X \Rightarrow Y$ is an association between two sets of items X and Y such that $X \cap Y = \emptyset$, $X \cup Y$ appears in more than *minsup* transactions, and that the number of transactions containing $X \cup Y$ divided by the number of transactions containing X is higher than $minconf$.

- *sequential pattern mining* (Agrawal and Srikant, 1995) consists of discovering frequent sequential patterns, i.e., subsequences appearing in more than *minsup* sequences of a sequence database, where *minsup* is a parameter set by the user.

For these three classical data mining tasks with wide applications, SPMF offers implementations of popular algorithms such as Apriori, Eclat (Zaki, M. J.), FPGrowth, GSP, PrefixSpan (Pei et al., 2004), SPAM (Ayres et al., 2000) and BIDE (Wang et al., 2007). But it also offers several algorithms for variations of these problems, for example to discover rare itemsets, closed itemsets, non-redundant association rules, indirect association rules, top-k association rules, to deal with uncertain data or database containing quantity and profit information and to discover *sequential rules* (Fournier-Viger et al., 2011). SPMF offers both classical algorithms and recent state-of-the-art algorithms such as Hui-Miner (Liu et al., 2012), ClaSP (Gomariz et al., 2013) and RuleGrowth (Fournier-Viger et al., 2011).

2. Using SPMF

SPMF is implemented in Java and is cross-platform. The only requirement to run SPMF is to have Java 7 or higher installed. There are two versions of SPMF. The *source code version* offers all algorithms from SPMF. The documentation provides an example of how to run each algorithm. It explains the input and output of each algorithm, its main characteristics and where to obtain more information about the algorithm. Moreover, a sample program and input file is provided in the source code of SPMF to show how to execute each example from Java code. Running an algorithm is just a few lines of code. One needs to create an instance of the algorithm, specify its parameter(s), input file and an output file path (if the result is to be saved to a file). For example, the following code runs the Apriori algorithm on a file "input.txt" with its *minsup* parameter set to 0.4.

```
AlgoApriori apriori = new AlgoApriori();
apriori.runAlgorithm(0.4, "input.txt", "output.txt");
```

The source code can be easily integrated into other Java software programs since (1) the source code of each algorithm implementation is located in its own subpackage and (2) there is no dependency on any other software or library. To support developers and users, extensive resources are provided on the website of SPMF such as an active forum, a FAQ, a developers' guide and a mailing list to be informed of the latest updates to SPMF. The website also provides a set of more than 40 large real-life data sets that can be used with the algorithms offered in SPMF. This can be useful for educational purpose (e.g., for a data mining course) or for comparing the performance of algorithms (for data mining researchers). Finally, the website also provides several performance comparisons of algorithms offered in SPMF, for various data sets, to give a good idea of the relative performance of the algorithms designed for the same task.

The *release version* of SPMF is a runnable JAR file that can be launched with a double-click. It provides a minimalistic user interface (see Figure 1), designed to allow quickly testing the behavior of the algorithms. The graphical user interface allows one to select an input file and an output file, choose an algorithm, enter its parameters and run it. For each algorithm, a sample input file is provided and an example is described in the documentation.

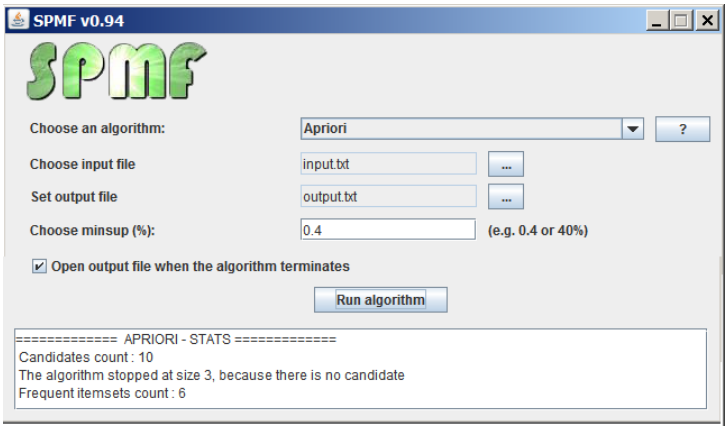

Figure 1: SPMF graphical user interface

The runnable JAR file can also be used to run algorithms from the command line. For example, to run Apriori, the following command can be used:

```
java -jar spmf.jar run Apriori input.txt output.txt 0.4
```

Input files for the algorithms are text files. The format that is used is the one from frequent pattern mining competitions such as FIMI (Boyardo et al., 2004) and used by researchers in this domain (files where items are represented by integers). But, to allow greater interoperability, the GUI and command line version of SPMF can also read the popular ARFF file format for itemset and association rule mining (used by Weka and Knime), and tools are provided to convert some selected formats to the SPMF format.

3. Conclusion

We have presented SPMF, a data mining library specialized in frequent pattern mining. The project is active and latest releases can be found on its website. SPMF has been applied in more than 70 research projects. Code submissions are reviewed by the project founder and contributors to see if they meet the requirements before being integrated in SPMF.

Acknowledgments

The authors thank the Natural Sciences and Engineering Research Council for its financial support and users who provided feedback and applied SPMF in their research projects.

References

R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the Twentieth International Conference on Very Large Data Bases*, pages 487-499, Santiago, Chile, 1994.

R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, pages 3-14, Taipei, Taiwan, 1995.

J. Ayres, J. Gehrke, T, Yiu, J. Flannick. Sequential pattern mining using bitmaps. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 429-435, Edmonton, Alberta, Canada, July 2002.

R. Bayardo, B. Goethals, M. J. Zaki, Proceedings of the IEEE Internation Conference on Data Mining Workshop on Frequent Itemset Mining Implementations Brighton, UK, 2004.

Coron. Software available at: `http://coron.loria.fr/site/index.php`

P. Fournier-Viger, R. Nkambou and V.S. Tseng. RuleGrowth: mining sequential rules common to several sequences by pattern-growth. In *Proceedings of the 26th Symposium on Applied Computing.*, ACM Press, pages 954-959, Taitung, Taiwan, 2011.

J. Han, J. Pei, Y. Yin, R. Mao. *Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Mining and Know. Discovery*, 8(1):53-87 (2004)

Galicia. Software available at: `http://www.iro.umontreal.ca/~galicia/`

A. Gomariz, M. Campos, R. Marin, B. Goethals. ClaSP: An efficient algorithm for mining frequent closed sequences. *Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 50-61, Gold Coast, Australia, 2013.

Illimine. Software available at: `http://illimine.cs.uiuc.edu/`

Knime. Software available at: `http://www.knime.org/`

M. Liu, J.-F. Qu. Mining high utility itemsets without candidate generation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 55-64, Maui, HI, USA, 2012

LUCS-KDD. Software available at: `http://cgi.csc.liv.ac.uk/~frans/KDD/Software/`

Mahout. Software available at: `http://mahout.apache.org/`

J. Pei, J. Han et al. Mining sequential patterns by pattern-growth: the prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(10):117, 2004.

R. Srikant, R. Agrawal. Mining sequential patterns: generalizations and performance improvements. *Proceedings of the 5th International Conference on Extending Database Technology*, pages 3-17, Avignon, France, 1996.

J. Wang, J. Han, C. Li. Frequent closed sequence mining without candidate maintenance. *IEEE Transactions on Knowledge and Data Engineering*, 19(8):1042–1056, 2007.

I. H. Witten and E. Frank. *Data mining: practical machine learning tools and techniques.* Morgan Kaufmann, 2005.

M. J. Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):372390.