Mining Patterns in Data

Implementing Sequence Mining

Gilles Peiffer (23421600)
Université catholique de Louvain
Place de l'Université 1
Louvain-la-Neuve, Belgium
gilles.peiffer@student.uclouvain.be

Liliya Semerikova (64811600)
Université catholique de Louvain
Place de l'Université 1
Louvain-la-Neuve, Belgium
liliya.semerikova@student.uclouvain.be

ABSTRACT

The following paper contains a detailed analysis of the performance and results of using various sequential pattern mining algorithms to fulfill several important tasks in the field of data mining. In a first part of the paper, algorithms existing in the literature are described, which are then compared on different tasks. The second part of the paper looks at the results of these mining tasks and gathers insights based on them.

Keywords

Data mining, machine learning, top-k frequent pattern, sequential pattern, closed pattern, supervised learning, performance analysis.

1. INTRODUCTION

Frequent sequential pattern mining is an active area of research in data mining with broad applications. Finding efficient algorithms for this task is thus majorly important, and with the rise of machine learning techniques such as supervised learning, it is interesting to consider which algorithms are able to combine both tasks well.
Additionally, some large datasets contain a lot of redundant information: to take a pathological example, consider the database made of the single sequence $\langle (a_1)(a_2)\ldots(a_{100})\rangle$. With a minimum support of 1, it will generate $2^{100}-1$ frequent subsequences, all of which are redundant except for the last one because they have the same support. For this reason, we also consider algorithms which perform well when mining closed sequential patterns.

2. TASKS

2.1 Frequent Sequence Mining

The goal of this task is, given two datasets of respectively positive and negative examples, to find the top-k most frequent sequential patterns across both of them. If multiple patterns obtain the same total support, all of them should only count for 1 in the value of k.

2.2 Supervised Sequence Mining

In supervised sequence mining, the aim is still to find top-k most frequent patterns, but with a new scoring function instead of the total support. Let $p(\alpha)$ and $n(\alpha)$ be the support of sequential pattern α in both datasets, and P and N be the number of transactions in each dataset; in that case, the *weighted relative accuracy* is given by

$$\mathrm{WRAcc}(\alpha) = \frac{PN}{(P+N)^2}\left(\frac{p(\alpha)}{P} - \frac{n(\alpha)}{N}\right). \tag{1}$$

In order to search for frequent patterns efficiently, an upper bounding procedure is necessary. By computing this bound, one can prune the search tree as soon as the bound does not exceed or equal the lowest score found amongst the current top-k sequential patterns. It is easy to see that for a sequential pattern $\beta \sqsupseteq \alpha$, the highest possible WRAcc score that can be attained is bounded by

$$\mathrm{WRAcc}(\beta) \leqslant \frac{Np(\alpha)}{(P+N)^2}, \quad \forall \beta \sqsupseteq \alpha, \tag{2}$$

where the assumption is made that all transactions containing α in the positive dataset also contain β, yet none of the transactions in the negative dataset do.

2.3 Supervised Closed Sequence Mining

For our purposes, we define a closed sequential pattern α as a sequence such that for any sequence $\beta \sqsupsetneq \alpha$, $p(\alpha) > p(\beta)$ or $n(\alpha) > n(\beta)$, that is, no supersequence exists such that both have the same support in both datasets.
The task of supervised closed sequence mining is applied using three different scoring functions:

- The WRAcc scoring function described earlier.
- The AbsWRAcc scoring function, defined as

  $$\mathrm{AbsWRAcc}(\alpha) = |\mathrm{WRAcc}(\alpha)|. \tag{3}$$

  Upper bounding can then be done as follows: for a sequential pattern $\beta \sqsupseteq \alpha$, the highest possible AbsWRAcc score that can be attained is bounded by

  $$\mathrm{AbsWRAcc}(\beta) \leqslant \frac{\max\{Np(\alpha), Pn(\alpha)\}}{(P+N)^2}, \ \forall \beta \sqsupseteq \alpha. \tag{4}$$

- The information gain function [3] (where p and n are used instead of $p(\alpha)$ and $n(\alpha)$, to alleviate notations):

  $$\mathrm{IG}(\alpha) = \mathrm{imp}\left(\frac{P}{P+N}\right) - \frac{p+n}{P+N}\,\mathrm{imp}\left(\frac{p}{p+n}\right) - \frac{P+N-p-n}{P+N}\,\mathrm{imp}\left(\frac{P-p}{P+N-p-n}\right), \tag{5}$$

where imp is the entropy, defined as

$$\operatorname{imp}(x) = -x \lg x - (1-x)\lg(1-x). \tag{6}$$

Computing a bound for this scoring function is harder to do analytically. To compute the maximum score for a sequential pattern $\beta \sqsupseteq \alpha$, one has the following relationship:

$$\operatorname{IG}(\beta) \leqslant \max_{\substack{0 \leqslant \pi \leqslant p(\alpha) \\ 0 \leqslant \nu \leqslant n(\alpha)}} \operatorname{IG}(\pi, \nu), \quad \forall \beta \sqsupseteq \alpha, \tag{7}$$

where we have used another definition of the information gain function directly taking the supports of the sequence as arguments. By precomputing the information gain for all pairs of values, this bound can be computed as a cumulative maximum.

3. ALGORITHMS

Various algorithms and implementations were used to complete the tasks outlined in Section 2.

3.1 PrefixSpan

The PrefixSpan algorithm was proposed by Pei et al. [2, 1] in order to mine sequential patterns using a pattern-growth approach. This algorithm was used for tasks 2.1 and 2.2. For our purposes, two implementations of this algorithm were written, one using a priority queue to store the top-k patterns and one using a sorted list.

For the tasks using alternate scoring functions, the algorithm is ran with a value of k starting from 1 all the way up to the original value, since this strategy allowed for faster pruning on large datasets; for larger values of k, the algorithm would occasionally insert low-scoring patterns in its results list, which would then prevent the algorithm from efficiently pruning its search tree. By using an incremental strategy, we were able to stop this from happening, with minimal overhead on easier datasets.

3.2 SPADE

Attendre que débilemobile termine son code…

3.3 CloSpan

The CloSpan algorithm is an adaptation of PrefixSpan designed specifically to mine closed sequential patterns proposed by Yan et al. [4], as in task 2.3. In order to do so, the CloSpan algorithm runs in two phases:

- A *search* phase, during which PrefixSpan is run with the additional constraint of computing a score defined as

  $$\sum_{(t,p) \in \mathcal{D}|_\alpha} \mathcal{D}[t] - p + 1, \tag{8}$$

  where $\mathcal{D}|_\alpha$ is the α-projected database. For any pattern, we check whether a pattern has already been seen that has the same score while also being a supersequence. If so, we cut the search tree and backtrack.

- A *post-processing* phase, where all patterns which are not closed are removed.

As with PrefixSpan, in order to avoid exploring the search tree too much because of inefficient bounding, the algorithm is run incrementally.

4. PERFORMANCE AND ANALYSIS OF RESULTING PATTERNS

4.1 Frequent Sequence Mining

4.2 Supervised Sequence Mining

4.3 Supervised Closed Sequence Mining

5. CONCLUSION

References

[1] Jian Pei, Jiawei Han, B. Mortazavi-Asl, Jianyong Wang, H. Pinto, Qiming Chen, U. Dayal, and Mei-Chun Hsu. Mining sequential patterns by pattern-growth: the PrefixSpan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1424–1440, 2004.

[2] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth. In *Proceedings of the 17th International Conference on Data Engineering*, pages 215–224, USA. IEEE Computer Society, 2001. ISBN: 0769510019.

[3] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986. ISSN: 1573-0565. DOI: 10.1007/BF00116251. URL: https://doi.org/10.1007/BF00116251.

[4] X. Yan, J. Han, and R. Afshar. CloSpan: Mining: Closed Sequential Patterns in Large Datasets. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 166–177. DOI: 10.1137/1.9781611972733.15. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611972733.15. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611972733.15.