LINMA2120 - Applied Mathematics Seminars

Seminar Report

Name: Breno Tiburcio Noma:7404-20-00

16th April, 2021

Project2 Report: Implementing Sequence Mining

1 Technical summary

1.1 Objective

The project objective is to implement different sequence mining algorithms and compare its results regarding the methodology behind each of them. Also, we make another comparison regarding the time running in regards to different top-ranked sequences and among different algorithms. Unlike the first project, we are dealing with a supervised pattern where we can evaluate how each sequence contributes to a particular classification (positive or negative).

1.2 Implementation

The algorithm chosen was the Prefix-Span. As seen in the course class, this algorithm has the advantage over the SPADE of not generating so many candidates, which can increase the algorithm's running time.

The algorithm follows a similar structure where a class Database is created to manage the transitions files. Although not requires by the project statement, the class in the algorithm can work multi-files, being each one for a possible class (not only positive or negative). This class consolidates the files into a unique data-set that can retrieve class' particular values as long as necessary, following the model type: supervised, closed, weighted or information gain. In each algorithm file, it is commented the mining model algorithm accompanied by general functions (auxiliary) that are slightly modified or increased according to each model premises.

All rely on the recursion method where at each level, a new database is projected. At each database, we calculate the recurrence of the prefix using a cover function. Global variables are created to store prefix and their respective recursions, enabling us to assemble new prefixes. For the first algorithm, the prefixes were ranked according to the sum of occurrences between the classes positive and negative. For the following three algorithms, algorithm the weighted relative accuracy is used to rank the most frequent prefixes.

$$Wracc(x) = (\frac{P}{P+N} * \frac{N}{P+N})(\frac{p(x)}{P} - \frac{n(x)}{N}) \tag{1}$$

As expected, the performance closed algorithm is superior because it reduces the number of false positives generated and improves the mining efficiency, especially in the presence of large, frequent free tree patterns in the graph database.

The absolute **wracc** consider the module of the scorer. By intuition, a magnitude negative is such an important pattern as a magnitude positive when it comes down to differentiate classes. The number of computation increases to a particular value of K in regards to the non-absolute algorithm.

The latest algorithm is based on the information gain. The algorithm opted was the entropy where takes into consideration how unbalanced the patterns are between the classes. It has an implementation close to the non-absolute weighted relative score mainly differentiated by considering both class recurrence list at each level recursion level to calculate the impurity.

1.3 Results

We have applied those five algorithms into different values of K top-ranking items based on the medium-sized database. Due to simplicity purposes, the sequence frequent algorithm bases on the sum of the negative and positive recurrences is faster regardless of K. As briefly commented above, the closed algorithm is more efficient because it saves us time exploring unnecessary sequences.

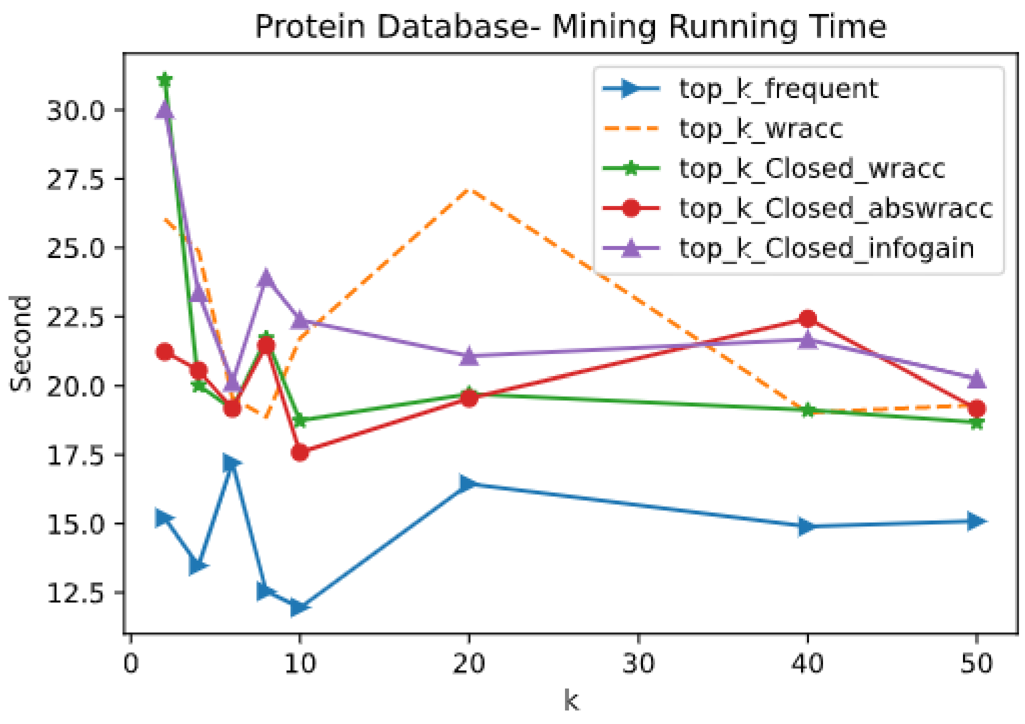

Fig. 1. Mining sequence algorithms running time simulations.

Naturally, each algorithm presents divergent results. It is essential to mention the divergence magnitude varies according to the data-set but is notably equality of results between the algorithms **Wracc Wracc Closed** since the scoring metric is the same. The **Wracc Absolute** and **Info Gain** presents some similar results once they privilege the contrast between the two classes. Lastly, the **Frequent Miner** prioritize those values according to their pure appearance, leading us to a unique prefix more easily.

2 Conclusion

Unfortunately, the algorithm implementation presented Timeout for the most extensive data sets. It happens because the mining pattern was implemented for multiple classes, not only to positive and negatives result came to a cost. The results are correct for the two most miniature data sets tested in the Ingenious.

In conclusion, this project helps us visualize the importance of understanding the metric we are using. It may be of relevant effect to the type of data we are mining. The trade-off between trade precision and recall is a conscious decision and might impact our results.