LINGI2364: Mining Patterns in Data

Implementing Sequence Mining

Jiayue Xue – NOMA: 04231800
Pierre Lambert – NOMA: 22661000

19th November 2018

CONTENTS

CHAPTER 1

OVERVIEW

In this assignment, our task can be divided into four parts.
Our first objective is to mine the k most frequent patterns by a PrefixSpan algorithm.
After that, given a supervised dataset, we modify the frequent sequence mining algorithm to find out the k best patterns according to the "Weighted Relative Accuracy" scoring function.
For the third part, we are required to adapt our supervised sequence mining algorithm in order to return only closed patterns.
Finally, we use two alternative scoring functions: "Absolute Weighted Relative Accuracy" scoring function and "Information Gain" scoring function to implement closed supervised sequence mining algorithm. Thus, with three different versions of closed supervised sequence mining algorithms, we can also compare them.

CHAPTER 2

IMPLEMENTATION DETAILS

2.1 Frequent Sequence Mining

PrefixSpan is one type of depth-first-search algorithm which efficiently finds out all the frequent sequences in the given transactions. Compared to ordinary task which finds out patterns according to a given minimum support threshold, this time we are required to output the top k frequent sequences considering the summation of the supports in positive and negative datasets. Regarding these requirements, we first combine two dataset into a bigger one so that the summation can be easily derived. Furthermore, in order to avoid searching on unnecessary nodes, we initialize the depth first search with a high minimum support threshold. If the output of the first iteration does not meet the top k constraint, we carefully adjust the minimum support threshold and continue the depth first search based on existing nodes to find the remaining results. Note that, by continuing on the former search tree, it is much faster than starting from the root again.

2.2 Supervised Sequence Mining with Wracc

In this section, we make full use of the first frequent sequence mining algorithm and make some paramount changes to implement the supervised sequence mining algorithm.
First, we continue the search in the tree in three conditions:

1. we do not have k results yet

2. the score of the current node is equal or larger than the lowest score of the found patterns

3. the score of the current node is smaller than the lowest score of the found patterns but its support in the positive dataset is larger than the lower bound

Second, we have a special strategy to update the lower bound dynamically so that we can greatly prune the search tree. To be more concrete, lower bound is computed as following:

$$Bound - P = P * min - score/coefficient, \tag{2.1}$$

where P is the total number of the positive transactions, N is the total number of the negative transactions, $min - score$ is the lowest score of the found patterns and $coefficient$ is the constant value in the scoring function: $(P/(P+N)) * (N/(P+N))$.

2.3 Supervised Closed Sequence Mining with Wracc

Based on the supervised sequence mining algorithm, we add another constraint on the search algorithmm: the closed constraint. Our choice to deal with the closeness constraint is to check its requirements only when inserting a pattern in the set of k best patterns by searching for and removing eventual sub-patterns with the same supports. Thanks to our appropriate data structure, we can realize this constraint easily. Since we store all the possible patterns in a dictionary and the patterns are stored in the same list if them have the same score. Although having the same score cannot necessarily conclude that the two patterns have the same support, this condition can provide useful information so that we can dramatically decrease the search space. To be more specified, if two patterns have the same support, they must have the same score no matter what kind of scoring function. In that, we can only search all the patterns having the same score to find out if any of them is not closed. Finally, all the patterns having the same score are stored together in our dictionary.

2.4 Supervised Sequence Mining with different scoring function

2.4.1 Absolute Weighted Relative Accuracy

For this task, we just need to add another lower bound constraint: $Bound - N = N * min - score/coefficient$. Note that we stop the search process only if the support of current node is both smaller than the two lower bound constraints. For one node, if its support in positive dataset is larger than $Bound - P$ or its support in negative dataset is larger than $Bound - N$, it stil has a chance to be qualified.

2.4.2 Information Gain

For this task, we just need to adopt another method to define the lower bound constraint. In this case, $lower-bound-P = Information-Gain(P, N, p, 0)$ and $lower - bound - N = Information - Gain(P, N, 0, n)$.

2.5 Different Scoring Functions Analysis

When comparing the different scoring functions, we can find out that the numbers of resulting elements of the information gain scoring function and the absolute Wracc are often similar. In addition, the output of the absolute Wracc are included in the results from information gain scoring function. Finally, we can also observe that the program using the information gain scoring function is slower than the other two programs.

Table 2.1: k = 6

Number of Output / Scoring function / Dataset	Wracc	Abs Wracc	Info. Gain
test	58	20	23
protein	6	6	6
reuters	timeout	timeout	timeout