A Network Algorithm to Discover Sequential Patterns

Luís Cavique

ESCS-IPL, Portugal
lcavique@escs.ipl.pt

Abstract. This paper addresses the discovery of sequential patterns in very large databases. Most of the existing algorithms use lattice structures in the space search that are very demanding computationally. The output of these algorithms generates a large number of rules. The aim of this work is to create a swift algorithm for the discovery of sequential patterns with a low time complexity. In this work, we also want to define tools that allow us to simplify the work of the final user, by offering a new visualization of the sequences, while bypassing the analysis of thousands of association rules.

Keywords: data mining, frequent sequence mining, graph theory.

1 Introduction

Sequential Pattern Discovery is a very important data mining subject with a wide range of applications. Sequence analysis is used to determine data patterns throughout a sequence of temporal states. Nowadays, sequence analysis is widely applied to find click stream data in the web sites. Several authors have been trying to find customers' purchase patterns of goods in the retail [3], people's careers [5] and in the financial sectors [11].

The Sequential Patterns Discovery, which extends itemset mining, is a complex task for the user. The user must specify a minimum support threshold to find the desired patterns. A useless output can be expected by pruning either too many or too few items. The process must be repeated interactively, which becomes very time consuming for large databases.

The association rules' systems that support the itemset and sequence mining, usually generate a huge number of rules, and therefore, it is difficult for the user to decide which rules to use.

Since most of the existing algorithms use a lattice structure in the search space and need to scan the database more than once, they are not compatible with very large databases.

These three handicaps are strong challenges to be overcome in this research area. In this work we present an algorithm that can discover sequence patterns with low time complexity and that can cut down on the work of the user.

In section 2 the related work with Frequent Pattern Mining is presented using a taxonomy of algorithms that distinguish between the Frequent Itemset Mining and the

J. Neves, M. Santos, and J. Machado (Eds.): EPIA 2007, LNAI 4874, pp. 406–414, 2007.

Frequent Sequence Mining. In section 3 the new network based algorithm, the Ramex algorithm, is presented. The sub-section 3.1 reports some network concepts in graph theory and the sub-section 3.2 describes the algorithm. In section 4, computational experience is reported using a web click stream dataset. Finally, in section 5, we draw some conclusions.

2 Related Work

Data Mining includes a wide range of procedures and algorithms, although, in the most recent literature three subjects belong to the core topics: the clustering (with strong origins in statistics), the classification and the frequent pattern mining. In frequent pattern mining we include the frequent itemset mining (or market basket analysis) and the frequent sequence mining.

In Frequent Pattern Mining two approaches can be used: The Frequent Itemset Mining (FIM) only needs two attributes as input: the transaction and the item. This subject is also known as the Market Basket Analysis. On the other hand the Frequent Sequential Mining (FSM) usually needs three attributes as input: the customer, the date and the item. The information retrieved by the FSM, that includes the variable time, offers a dynamic view of the purchasing process [12].

Most of the Frequent Pattern algorithms use lattice structures in the search space. In the search two methods can be used, the breadth-first search and the depth-first search. A third approach can be carried out by transforming the problem and condensing the data.

Table 1. Taxonomy of the Frequent Pattern Mining

	Frequent Itemset Mining (FIM)	**Frequent Sequential Mining (FSM)**
Lattice, breadth-first search	Apriori	ApprioriAll GSP
Lattice, depth-first search	FP-growth	SPADE
Transformed Structures	Similis	Markov Chain Model Ramex

In table 1, a taxonomy of the Frequent Pattern Mining algorithms is presented. The Frequent Itemset Mining algorithms include for the breadth-first search, the Apriori algorithm [2] and for the depth-first, search the FP-growth [9]. The FIM deals with static itemsets, i.e., they don't change with time.

On the other hand, the Frequent Sequence Mining involves a sequence of steps in time. A typical purchasing sequence starts when a customer buys, for instance, a laptop. There are lots of other items that can be bought afterward, like a mouse, then a CD with software, a bag, a memory stick, a floppy disk drive, a printer/scanner, a modem/router and other multimedia items.

The Frequent Sequence Mining, FSM, algorithms include for the breadth-first search, the ApprioriAll algorithm [1] and the GSP (Generalized Sequential Pattern)

[13] and for the depth-first search the SPADE (Sequential PAttern Discovery using Equivalence classes) [14]. They use a lattice structure in the search space resulting in high computation time when dealing with large databases.

The transformation method shrinks the data into new data structures, and afterward it uses known techniques to extract the patterns. The Similis algorithm [4] transforms the database into a weighted graph and heuristic search techniques discover complete sub-graphs that correspond to itemsets. For Sequential Pattern Mining the Markov Chain Models and the proposed algorithm Ramex use network structures to condense the data. With the Markov Chain Model a global view is possible since all items are taken into consideration. On the other hand, the Association Rules usually present a set of rules that only show the items with high support.

Most of the existing software for cross-selling, using association rules, generates thousands of rules, and therefore it is difficult for the marketeer (or user) to predict the next-item that each customer will buy. To implement cross-selling strategies we want an algorithm that can discover the next item, that reduces the work of the user and with a low time complexity algorithms.

3 Ramex Algorithm

The aim of this approach is to create a tree of sequences, with as many branches as needed to visit all the vertexes, starting by a vertex, called root. Ramex is the name of present algorithm that means branch in Latin. This term was chosen just like Apriori and Similis, since they are all Latin names.

3.1 Network Concepts

In this sub-section a short bibliographic overview in graph theory is given to be reused in the next sub-section.

Given a connected undirected graph, a spanning tree of the graph is a sub-graph that connects all the vertexes. A minimum weight spanning tree is the spanning tree with a weight that is lower than or equal to the weight of every other spanning tree. This problem is easily solved using a greedy algorithm. The algorithms proposed by Kruskal and Prim are well known examples. In Kruskal's algorithm, the edges are chosen without worrying about the connections to previous edges, but avoiding the cycles. In Prim's algorithm the tree grows from an arbitrary root.

In a connected directed graph G, an acyclic sub-graph B is a branch in G if the in-degree of each vertex is at the most 1. Let w(B) be the sum of the weighted arcs in branch B. The maximum weight branching problem is the optimization problem that finds the largest branch B possible. Edmonds' branching algorithm [6], proved by Karp [10] can be described in two steps, as follows:

i) The condensation process: the input is the weighted directed graph G, let G_1 be equal to G, repeat the process by condensing the cycles, G_k is the digraph which is acyclic.
ii) The unraveling process: Let B_k=G_k, then constructs B_{k-1} from B_k by expanding the condensed cycles, until a tree is obtained.

Fulkerson [7] presents the same problem with an additional constraint, the branch must start in a vertex called the root. His algorithm for the maximum packing rooted directed cuts in a graph is equal to the weight of the minimum spanning tree directed away from the root. The algorithm is also described in two steps: the condensation process, to remove the cycles, and the unraveling process where the branch is created.

3.2 Ramex Description

The Ramex approach has strong connections with the Markov Chain Models. In the Markov Chain Model each state is an item and in the transition matrix, each *Pij* is the probability of moving from state *i* to state *j* in one step**.** The Ramex, like the Markov Chain Models, also presents a global view, since all the items are taken into consideration. In the Ramex approach instead of the relative frequencies, the absolute frequencies are used. In each transition the number of times from an item to the next-item is reported.

The Ramex algorithm is presented in two main steps: the transformation of the problem and the search of the sequences.

Ramex Algorithm
Input: a database (customer, date, item);
Output: a sequence of items;
1) Network Transformation
1.1) Sort data by customer, date and item;
1.2) Create a new attribute next-item;
1.3) Build a state transition network G;
2) Find highly probable branch sequence B;
2.1) Condensation process;
2.2) Unraveling process.

In the transformation of a database into a network, the raw data (customer, date, item) must be sorted in such a way that each customer sequence can be identified. For each line in the table the new attribute next-item is created. Then a network, i.e. a graph G with a source and a sink is created, where each customer sequence has an initial node called source (or root) and a final node called sink.

The network, where cycles are allowed, condenses the information of the database by incorporating all the customer sequences. In the network G each state corresponds to an item and each transition represents the sequence from one item to the next-item. The weight of each arc corresponds to the number of times that one item precedes the next-item.

The second step is to find in the network G the highly probable branch sequence B, given a root vertex, where the Maximum Weight Rooted Branching Algorithm is applied, as defined in Fulkerson [7]. Like Edmonds' branching algorithm [6], Fulkerson's algorithm has $\Theta(N^2)$ in the worst case time complexity, where N is the number of vertexes.

In the original table a new attribute "next-item" must be included. Afterward, using the feature Cross Reference Query, that exists in most of the databases, the adjacency matrix can be obtained. The adjacency matrix of G indicates the number of times that a transition occurs from one item to the next-item.

Table 2. Adjacency matrix

	A	laptop	mouse	bag	CD	mem	Z
A	-	3					
laptop		-	2	1			
mouse			-	2	2		
bag				-		3	
CD					-		2
mem			2			-	1
Z							-

Numeric Example. Using the adjacency matrix presented in table 2, a network in figure 1-I can be built. In the network the source is represented by A and the sink is represented by Z. In this type of network, cycles may be present. A cycle can be identified: mouse, bag, memory and mouse again. Starting at the root A, all vertexes must be visited, except Z, avoiding cycles, where the sum of the weighted arcs is as heavy as possible.

In the condensation phase, figure 1-II, the existing cycle is shrunk, obtaining a directed acyclic graph. In the unraveling phase, figure 1-III, the nodes are expanded, obtaining the tree in figure 1, where the dotted line represents the forbidden arcs.

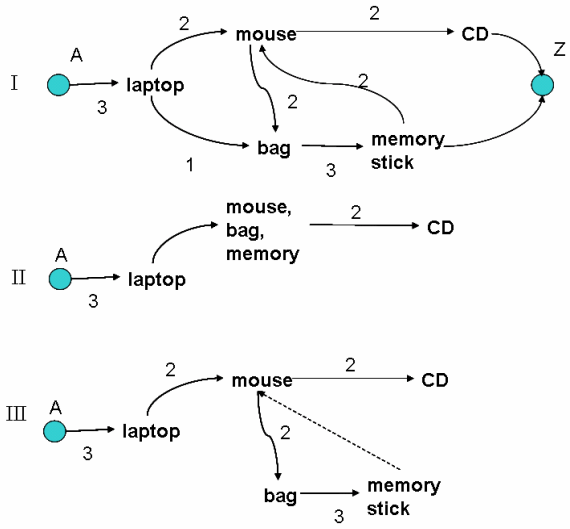

Fig. 1. Phases of the Maximum Weight Rooted Branching Algorithm

So, this optimization problem is solved by the Maximum Weight Rooted Branching Algorithm and the solution is a tree, as represented in fig.1-III. In this figure the sum of the weights is equal to 3+2+2+2+3=12. In comparison, the value of

the Minimum Weight Rooted Branching is 3+1+3+2+2=11, which uses the arc (laptop, bag).

4 Computational Experience

To validate the Ramex algorithm, a dataset named Web Click-Stream [8] was used. This file has 250,711 observations, from 22,527 web visitors and 36 web pages. The dataset has three attributes: client identifier (cookie), date/ time and web page. In this case a sequence is given by the clicks a client produces on a particular date.

Table 3. Statistic Measure of the dataset

Statistic Measure	Number Clicks
Average	11.1
Mode	5
Minimum	1
First quartile	6
Median	8
Third quartile	13
Maximum	192
Skewness	4.7
Kurtosis	40.8

Before using the algorithm it is important to make a brief exploratory data analysis. Table 1 shows some statistic measures for the number of clicks. The average number of clicks per visit is 11, the mode is 5 and the median 6. The minimum number of clicks is 1 and the maximum is 192, which seems far-fetched. The distribution is heavy skewed to the right and presents a high kurtosis equal to 41.

The algorithm Ramex-phase-1 transforms the dataset and generates a network with 36 nodes, 539 arcs with 42% of density.

Afterward, in the Ramex-phase-2, the Maximum Weight Rooted Branching Algorithm is applied, as defined in Fulkerson [7]. The input of the algorithm needs an initial node (root), in this case it is number 1, and the Ramex output is a tree as shown in figure 2.

We think that a tree is the best way to represent the most frequent sequences. In this way, we can find many branches in the tree, that includes all the vertexes, with short branches and long branches. Each branch corresponds to a weighted sequence of clicks. By looking closely at the tree we can see different clusters, after node 12-home. The sequence 17, 8, 9 corresponds to news, feedback, feed-post that represents the users who are looking for news. The branch that starts in 13-login represents users who login, logout and register. The sequence 5 and 28 represents users of the catalog. The branch that starts with 22-program is divided into two sub-branches. Sub-branch 18, 23, 33 represents the users that want information and also want to test the

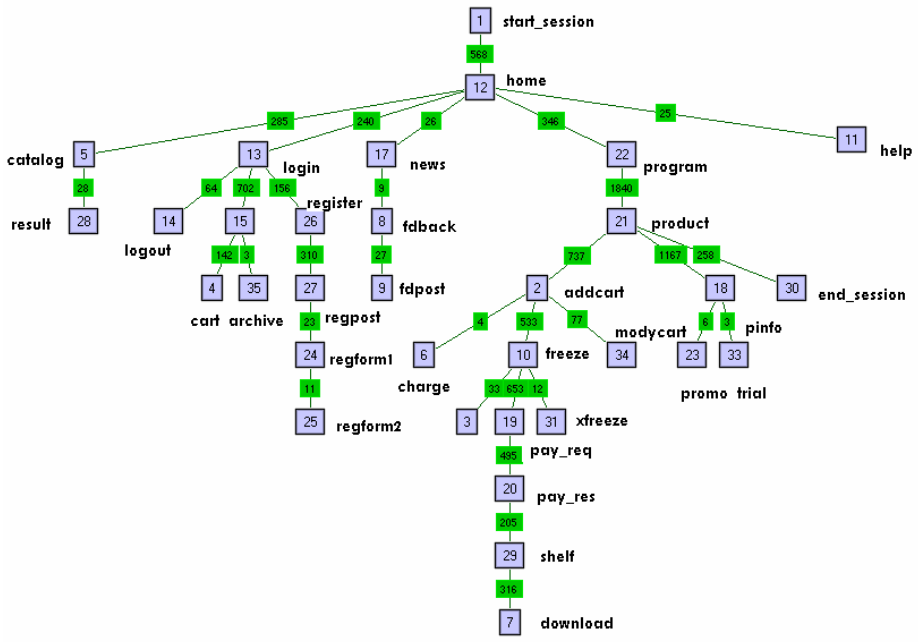

Fig. 2. Ramex Output

product. The second sub-branch, that starts with 2-add-cart, represents the users that add a product to the basket cart. Afterward, the users pay and finally download the product. The main advantage is that we can find long sequences of clicks and even clusters of clicks using a low complexity time algorithm.

It is hard to make comparisons among different methods that use completely different approaches and return different outputs. However, in this section we will report the advantages and the disadvantages of Ramex, compared to the Markov Chain Model, the Link Analysis and the Association Rules, keeping in mind that the sequence patterns must be found.

Markov Chain Models. Using a Markov Chain Model the path of the most likely transitions is given in table 4 [8]. These results are identical to the solution given by the Ramex algorithm. However, the Ramex algorithm returns a more complete solution, presenting the needed branches that cover all the vertexes.

Table 4. Markov Chain Model the forward path

Page i	Page j	Prob (i, j)
Start_session	Home	45.81%
Home	Program	17.80%
Program	Product	70.18%
Product	P_info	26.73%

Link Analysis. Link Analysis usually uses a circular graph, where the clicks between two pages can be shown. Strong links can be found between some pages, like 21-product, 18-pinfo and 21-product, 22-program. An extension of this approach is the search of complete sub-graphs, in order to find strong connected pages. The drawback is that the time sequence is lost. When studying the click-stream, the visualization is critical. The visualization with a circular graph, used in the link analysis, is recommended when the time is not relevant.

Association Rules. In the market basket analysis, the Soul-Mate algorithm compares a k-itemset with an identical k-itemset of a single customer. On the other hand, Apriori-like algorithms find all the exact k-itemsets. In the sequence pattern discovery the AprioriAll counts the exact number of sequences. In a more general way, the Ramex algorithm, like the Markov Chain Models, mixes all the sequences in the digraph going further than a single comparison or an exact count, by creating a pattern. Emerging from the network pattern a weighted tree can be revealed, offering new directions for the decision maker.

5 Conclusions

Knowledge Discovery in Databases (KDD) and Data Mining appear in the literature so tied to each other that sometimes they seem to be the same. However, Data Mining is part of Knowledge Discovery. Knowledge Discovery in Databases can be divided in three sub-process: the data pre-processing, the data mining and the post-processing. The pre-processing includes the ETL, extraction, transformation and loading the data into the data warehouses, and involves subjects like data reduction, normalization and data quality. On the other hand, the data post-processing includes the visualization and patterns interpretation for decision making.

In this paper a new approach in Data Mining to discover Sequence Patterns with a low time complexity algorithm is presented. This work also includes the post-processing dimension of KDD, the visualization of the sequences using a tree seems easier than using association rules or using the link analysis. Once the branches of items are known the user can easily decide what is the next-item for each customer.

In Data Mining the time complexity of the algorithms is very important. To discover sequence patterns we propose the Ramex algorithm. To run the algorithm no parameter are needed. On the other hand, other algorithms use the min-support as a parameter to control the combinatorial expansion. This algorithm doesn't use the lattice structure in the search space, but it uses data condensation in a network. The procedure that returns the output is a low time complexity algorithm with $\Theta(N^2)$ in the worst case, where N is the number of items of the condensed network.

The Ramex algorithm finds large branch patterns that sometimes do not correspond to the exact sequence count, but express the combination of several sequences, as patterns tend to do.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings 11th International Conference Data Engineering, ICDE, pp. 3–14. IEEE Press, Los Alamitos (1995)
2. Agrawal, R., Srikan, R.: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Databases, pp. 478–499 (1994)
3. Berry, M., Linoff, G.: Data Mining Techniques for Marketing, Sales and Customer Support. John Wiley and Sons, Chichester (1997)
4. Cavique, L.: A Scalable Algorithm for the Market Basket Analysis. Journal of Retailing and Consumer Services, Special Issue on Data Mining in Retailing and Consumer Services (accepted paper, 2007)
5. Dunham, M.H.: Data Mining: Introductory and Advanced Topics. Prentice Hall, Pearson Education Inc. (2003)
6. Edmonds, J.: Optimum branchings. J. Research of the National Bureau of Standards 71B, 233–240 (1967)
7. Fulkerson, D.R.: Packing rooted directed cuts in a weighted directed graph. Mathematical Programming 6, 1–13 (1974)
8. Giudici, P.: Applied Data Mining: Statistical Methods for Business and Industry. John Wiley and Sons, Chichester (2003)
9. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, United States, pp. 1–12 (2000)
10. Karp, R.M.: A simple derivation of Edmonds' algorithm for optimum branchings. Network 1, 265 (1971)
11. Prinzie, A., Van den Poel, D.: Investigating Purchasing Patterns for Financial Services using Markov, MTD and MTDg Models. In: Working Papers of Faculty of Economics and Business Administration, Ghent University, Belgium 03/213, Ghent University, Faculty of Economics and Business (2003)
12. Silvestri, C.: Distributed and Stream Data Mining Algorithms for Frequent Pattern Discovery, Universit'a Ca' Foscari di Venezia, Dipartimento di Informatica, Dottorato di Ricerca in Informatica, Ph.D. Thesis TD-2006-4 (2006)
13. Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 3–17. Springer, Heidelberg (1996)
14. Zaki, M.J.: Spade: An efficient algorithm for mining frequent sequences. Machine Learning 42, 31–60 (2001)