PREVENT: An Algorithm for Mining Inter-transactional Patterns for the Prediction of Rare Events

Christos BERBERIDIS[1], Lefteris ANGELIS[1] and Ioannis VLAHAVAS[1]

[1]*Department of Informatics, Aristotle University of Thessaloniki,*
54124 Thessaloniki, Greece
{berber, lef, vlahavas}@csd.auth.gr

Abstract. In this paper we propose a data mining technique for the efficient prediction of rare events, such as heat waves, network intrusions and engine failures, using inter transactional patterns. Data mining is a research area that attempts to assist the decision makers with a set of tools to treat a wide range of real world problems that the traditional statistical and mathematical approaches are not enough in terms of efficiency and computational performance. Transaction databases, such as the ones in this paper that contain sets of events, require special approaches in order to extract valuable temporal knowledge. We utilize the framework of inter-transaction association rules, which associate events across a window of transactions. We propose an approach that extends sequential analysis to predict rare events in transaction databases. We formulate the problem of rare events prediction and we propose PREVENT, an algorithm that produces inter-transactional patterns for the fast and accurate prediction of a user-specified rare event. Finally, we provide experimental results and suggest some ideas for future research.

Keywords: Data mining, rare events prediction, association rules, sequence analysis.

1. Introduction

The prediction of rare events from data is a particularly interesting problem, because the result not only has to be accurate but it also has to be delivered in time. By the term "rare events" we mean events of a certain domain that do not happen regularly but they have a special meaning or play an important role in the system and they are usually hard to predict. Examples of such events are network intrusions, engine failures, earthquakes and meteorological events such as hail and heat waves. In this paper we propose a novel technique for the prediction of rare events, which is based on the inter-transactional association rules framework. We describe PREVENT (Prediction of Rare EVENTs), an FP-Tree based algorithm that outperforms Apriori-based approaches for inter-transaction association rules mining, since it requires only 2 database scans.

Event prediction is very similar to time series prediction. Classical time series prediction, which has been studied extensively within the field of statistics, involves predicting the next *n* successive observations from a history of past observations [12]. These statistical techniques involve the building of mathematical probabilistic models, which are based on

specific data, since they are strongly dependent on various theoretical assumptions regarding the underlying nature of variation (probability distributions etc). However, this is not our case. First, we are interested in extracting knowledge from a very broad class of large transaction databases, without any prior information on the variability of the data and therefore without having to state theoretical assumptions. Second, our main goal is not to build certain mathematical models, but to discover patterns and rules, which are related to certain critical events and which are going to provide us an alarm for the early identification of such events.

The typical problem of mining association rules concerns the search of associations among discrete items in a database. The traditional association analysis is intra-transactional because it concerns items within the same transaction. The goal is to find all sets of items that occur frequently in the same transaction and from those sets to derive rules that one subset of an itemset implies another. The notion of transaction is very general and includes items bought by the same customer, requests from same user, events happened on the same day, etc. Inter-transactional association rules are a new kind of association rules that associate items within a window of many transactions. In that sense, intra-transactional rules are a subset of the inter-transactional ones. Although transactions occur under certain contexts such as time, space, customers, etc., such contextual information has been ignored because this task was intra-transactional. Association rules aggregate information from a large number of transactions into a rule for a single transaction: “In the millions of transactions of a supermarket, there is a 60% probability to find beers and diapers in the same transaction”. However, rules like "If the prices of IBM and SUN go up, Microsoft's will most likely (80% of the time) goes up 2 days later" cannot be captured by the intra-transactional approaches. This kind of rule associates itemsets among different transactions, along the axis of time. The contextual information here is time, which is the dimensional attribute. Those rules are called inter-transactional and they can be single or multi dimensional.

In this paper we propose PREVENT, an algorithm for efficiently and accurately predicting rare events in a transaction database. We utilize the predictive power of inter-transactional association rules to discover predictive patterns from meteorological data, for the prediction of heat waves. The major advantage of inter-transactional association rules is that besides description they can also facilitate prediction, providing the user with explicit dimensional (in the case of prediction, temporal) information. It is often useful to know when to expect something to happen with accuracy (e.g. "five days later") instead of a fuzzy temporal window (e.g. "some day within 1 week") or a sequence (e.g. B and C will happen after A). Unlike other approaches, our approach does not concern the discovery of rules but searches for patterns that contain the temporal information required for the task of prediction.

The paper is organized as follows: The next section presents a review of the literature regarding inter-transaction association rules. In section 3 we provide the mathematical formulation of the problem, including definitions and theoretical background of our approach. The algorithm we propose is described in section 4, along with a brief discussion about its computational complexity and performance. In section 5 we present the experiments we conducted in order to test and verify the performance of the proposed algorithm and finally, in section 6 we present our conclusions and propose our ideas for further work.

2. Related Work

Having a temporal database, we can mine for various types of association rules. One approach is to cluster the data based on time and then discover association rules from each cluster, in order to track how the model changes over time [9]. Traditional association rule

analysis was extended to *sequence analysis*, where the members of the series are sets of individual *items*, called *itemsets*, from some underlying domain (alphabet). Given a set E of events, an *event sequence* s is a sequence of pairs (e, t), where $e \in E$ and t is an integer, the occurrence time of the event of type e. Unlike time series, sequences do not require any explicit relationship with time, only that the itemsets are totally ordered.

The basic difference between the two concepts, then, is that a time series is a list of ordered values, while a sequence is a list of ordered itemsets or values. Sequential pattern mining aims to discover patterns such as {{A}, {B}, {C, D}}, where {A}, {B} and {C, D} are itemsets in different transactions, within a user-defined time window. Finding the most frequent maximal patters is a particularly useful task that provides the user with valuable insight about the temporal nature of the data. However, the predictive power of sequential association rules is questionable. Sequence analysis or sequential pattern mining was extensively studied initially by Agrawal et al. [6, 7], where the notions of sequence and subsequence were defined.

An *episode rule* [8] is a generalization of association rules applied to sequences of events. An *event sequence S* is an ordered list of events, each one occurring at a particular time. Thus, it can be viewed as a special type of time series. Given the above definitions, an *episode a* is a partial order of event types. Episodes can be viewed as directed acyclic graphs. There are serial, parallel and non-serial and non-parallel episodes. Episode mining algorithms are searching for episodes or episode rules within a sliding window of user-defined size. What is captured here is the temporal relationship among events that occur within the same window, e.g. "C comes after A and B within a window of size *w*".

Inter-transactional association rules were introduced in [1] and [2]. The authors extend the notion of inter-transactional association rules to the multidimensional space and propose EH-Apriori, an Apriori-based algorithm, for mining such rules. The authors also propose the use of templates and concept hierarchies as a means to reduce the large number of the produced rules. A new set of algorithms is introduced in [3], called FITI (an acronym for "First Intra then Inter"), which outperforms EH-Apriori. In [4] and [5], the authors use inter-transactional association rules for prediction on meteorological and stock market data, correspondingly.

3. Problem Formulation - Definitions

In our setup we have the following notions:

- The set of *items* $I = \{i1, i2, \ldots, iv\}$ representing the possible activities we want to keep record of (e.g. items sold in a store or responses to requests by a server).
- The *dimensional variable T* describing the time properties associated with the items. We assume that the variable takes ordinal values representing intervals of equal length (e.g. day, week, month etc.). Note that this variable can be defined to represent various other ordinal measurements such as length, height, etc. It is also possible to have many of these variables (time, distance, etc.), simultaneously describing our data, but in our context we consider only one. Without loss of generality we denote the values of T by integers 0, 1, 2, ...
- The *transactions* which are records of the form $J(t)$ where t is a value of the time variable T and $J(t) \subseteq I$. So, each transaction is represented by a set of certain activities from I recorded in time t.
- The *transaction database* containing all the transactions recorded over a (usually long) period of time.

- The *transaction sequence*, a time-ordered sequence of transactions, denoted by $S = J(t1), J(t2), \ldots, J(tn)$, which includes n transactions recorded in the time interval $[t1, tn]$.
- The *target item*, $i^* \in I$, which represents an activity that we are particularly interested in predicting it, e.g. failure of a system to respond, network fault, etc. Such an item occurs infrequently with respect to the other items while its occurrence is much more critical than the others'. Let us also denote by t^* the time interval when the target item occurs.
- The *target transactions* which are transactions containing the target item. Note that for our work here we do not need to consider target items at all but more generally target transactions, since a target transaction may have the meaning of an infrequent combination of items that we are interested to predict.

So, the problem we consider here is to derive inter-transactional patterns that can be used as alarm messages in order to predict the target transactions within a reasonable period of time before the critical target item occurs. For this purpose we associate with every target transaction $J(t^*)$:

- A *prediction period*, which is a time period preceding the target transaction of fixed length defined as $[t^*-m, t^*-w]$ where m is the *monitoring time* and m is the *warning time*. We assume that $m > w$.
- A *target - preceding window* W^*, which is a block of $m-w+1$ continuous time intervals included in the prediction period of the target transaction. Thus, the window consists of all the time intervals from t^*-m to t^*-w. Note that it is not necessary for each interval to contain a transaction. These intervals within a window are called *target - preceding subwindows* of W^*. We will use non-negative integers to denote the subwindows. So, we denote the subwindow in the beginning of the prediction period by $W^*(1)$, and the following ones by $W^*(1), \ldots, W^*(m-w+1)$. We will also use the same indices to denote the items in each subwindow. Thus, if the item i_k $(1 \le k \le v)$ occurs in *target - preceding subwindow* $W^*(x)$ $(1 \le x \le m-w+1)$, it will be denoted by $i_k(x)$. Such items are called *extended items*. We denote the set of all possible extended items by
$$I^* = \{ i_k(x): \ 1 \le k \le v, \ 1 \le x \le m-w+1 \}.$$
- A *target megatransaction* $M^* \subseteq I^*$ defined as the set of all extended items within W^*, i.e.
$$M^* = \{ i_k(x): i_k \in W^*(x), 1 \le k \le v, \ 1 \le x \le m-w+1 \}$$
- An *intertransaction association rule predicting the target transaction* is an implication of the form $F \Rightarrow G$ where $F \subseteq I^*$ and G is the target transaction.
- Two *measures* of intertransaction association rules $F \Rightarrow G$:
$$\textit{support of } F: \ s = \frac{N_F^*}{N^*} \qquad (1)$$
where N^* is the number of all the target megatransactions in the database and N_F^* is the number of all target megatransactions that contain the set F. We can characterize a set as *frequent* if s exceeds a lower bound, defined by the user.
$$\textit{confidence of } F: \ c = \frac{N_F^*}{N_F} \qquad (2)$$
where N_F is the set of all megatransactions of size $m-w+1$ in the database that contain the set F. We can characterize a set as *accurate* if c exceeds a lower bound, defined by the user.

The purpose of the search is to find all frequent and accurate sets of extended items that contain the temporal information required for the prediction task. Those are temporal patterns that can be used for prediction.

4. Our approach: PREVENT

In this paper we propose PREVENT (Prediction of Rare EVENTs), an algorithm based on the inter-transactional association rules framework, which provides an excellent basis for producing predictive patterns. The general strategy we follow to predict rare events takes into account the fact that it is highly important that a prediction is given in time. Therefore, we assume that there is a time period preceding a target event X_t, when the prediction can be useful (prediction period or monitoring window). This period starts with a time point that denotes the beginning of the period when the user is interested in having a prediction and ends with a time point after which it is too late and the prediction has no practical meaning (warning time) [11]. The concept is illustrated below:

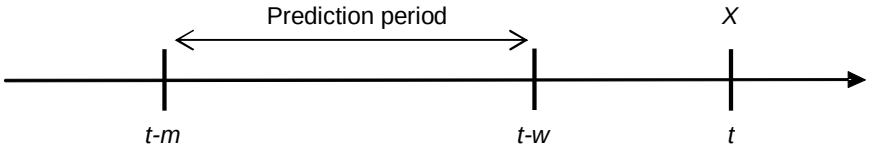

Figure 1. The prediction period.

Given that the warning time is always *w* time points before the target event X_t, we propose an efficient method for mining predictive patterns within the prediction time period. Alternatively to the original inter-transactional association rules paradigm, we propose the use of patterns instead of the typical If-Then rules for the prediction task. A temporal inter-transactional pattern contains all the necessary information, except from the notion of causality, which, even using rules as a representation form, can be elusive and hard to prove anyway. Consider a typical inter-transactional association rule, such as $A(t_1) \rightarrow X(t_n)$, where $X(t_2)$ is the target event, $A(t_1...t_k)$ is a set of extended items (events taking occurring at time points $t_1..t_k$) and $t_n > t_k$. In the proposed approach we perform prediction using the frequent extended itemsets derived from the prediction periods of *X*. Since we are interested only in the rules that have *X* in the consequent, extracting only the frequent extended itemsets from the monitoring windows would be enough for predicting the target event. This makes the whole process simpler and faster.

One of the major advantages of PREVENT is that it requires only 2 database scans. We perform one scan over the database in order to capture and store only the transactions associated with those periods, using a sliding window. The number of such periods (windows) stored is equal to the number of occurrences of the target event, which, in our case is rare. In other words, we capture the corresponding monitoring window of every occurrence of the target event in order to extract the desired knowledge. While capturing those windows, a database transformation takes place in order to map the relative temporal information of every item within the window. The transformation is done according to the definitions given in the previous paragraph, based on the inter-transactional association rules framework. An example of such transformation can be seen in the following figure.

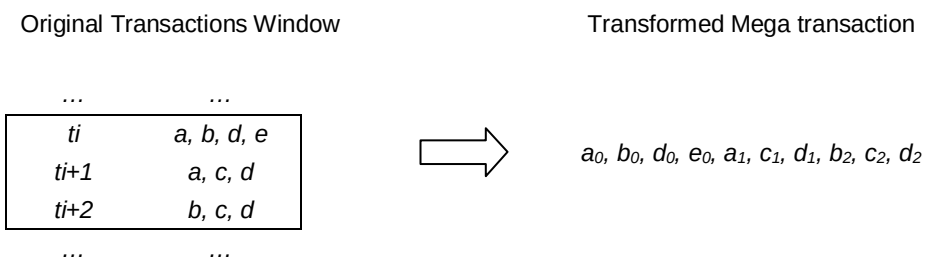

Figure 2. A data transformation example.

For memory efficiency purposes, we map every item instance i_t to an integer and keep the index in order to be able to backtrack later to the original data. An example of such mapping is the following:

Example. Assume that the size of the monitoring window is 4 transactions and the set of literals in the database is *{a, b, c, d}*. The corresponding set of extended items and their integer mapping are depicted in the table below.

Table 1. Integer mapping example

Set of Extended Items	a_0	a_1	a_2	a_3	b_0	b_1	b_2	b_3	c_0	c_1	c_2	c_3	d_0	d_1	d_2	d_3
Integer Mapping:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

The following step is the mining of the frequent itemsets from the transformed data. At this point we provide the reader with useful information about the frequent itemset mining algorithm we use, FP-Growth, taken from [10], where it was introduced. FP-Growth builds an FP-tree (Frequent Pattern-tree), which is an extended prefix tree structure that stores crucial information about frequent patterns. FP-tree is actually an efficient way to compress the original database into a much smaller structure that is cost-effective to mine. Each node contains a frequent item (itemset of length 1). Each transaction contributes at most one path to the FP-tree, with length equal to the number of frequent items in that transaction. Quoting from [10]: "The tree nodes are arranged in such a way that more frequently occurring nodes will have better chances of sharing nodes than less frequently occurring ones. Our experiments show that such a tree is highly compact, usually orders of magnitude smaller than the original database." Its major advantage is that it reduces the number of database scans to only 2 in order to construct the FP-tree.

FP-Growth, a divide and conquer algorithm, is used to mine the patterns from the FP-tree. It scans the FP-tree once to build a small pattern base for each frequent item a_i, each consisting of the set of transformed prefix paths of a_i. Frequent pattern mining is then recursively performed on the small pattern bases. Pattern bases are usually much smaller than the original FP-tree. While Apriori-based approaches require a large number of repeated scans and the generation of a very large number of candidate sets, which often reaches the levels of a combinatorial explosion, FP-Growth requires only 2 database scans to create the FP-tree. Then it reduces the problem of mining the frequent k-itemsets into a sequence of k frequent 1-itemset mining problems. FP-Growth avoids the costly generation of candidate itemsets that Apriori-based approaches require. Especially in our setup, where the number of different (extended) items is usually large (Monitoring Window Size x Alphabet Size), the application of an Apriori-based algorithm would be extremely costly.

We do not provide further information or examples of FP-tree algorithm, because it is not within the scope of this paper. For more information about the FP-Tree algorithm we refer the reader to [10].

Summarizing, the steps of our algorithm are outlined below:

```
1) Move a sliding window across the transactions of the database until
   the next occurrence of the target item is found.
   a) For every such occurrence, capture the corresponding monitoring
      window, transform it as described above and store it in a new da-
      tabase file.
   b) Store the integer-mapping index.
   c) Count the support of the extended 1-itemsets.
2) Build the FP-Tree
3) Extract the frequent extended itemsets (predictive patterns).
4) Using the integer-mapping index, convert the items of the rules,
   from integer numbers into their original form.
```

***Figure 3**. PREVENT*

4.1. Algorithm analysis and discussion

When speaking about computational complexity within the data mining context, what is mostly important is the number of database scans. When the main memory is not enough to fit the data, main memory based operations are insignificant compared to operations that require hard disk access. The major advantage of our algorithm is that it requires only 2 scans over the original database, regardless of the size of the database or the number of literals; one for the first step (sliding window) and one for the construction of the FP-tree [10]. The database transformation has been embedded into the first pass of FP-Tree.

Moreover, the main memory structures used are small and pose no additional overhead. What is important here is that the shrinking factor achieved by the transformation of the original database. In [10] the authors claim that while the shrinking factor of the first FP-tree normally ranges from 20 to 100, the shrinking factor from this FP-tree to the pattern bases is expected to be another hundreds of times reduction for constructing each conditional FP-tree from its already quite small conditional frequent pattern-base. The size of the sliding window is $m \times MaxTransactionSize$, and the size of the integer mapping index is (Monitoring Window Size) x (Size of the set of extended items), both of which are appropriate for the main memory. Since there are often a lot of sharing of frequent items among transactions, the size of the tree is usually much smaller than its original database and that of the candidate sets generated in the Apriori-based approaches.

4.2. Implementation and Performance Results

We implemented our algorithm in C++ and tested it against a number of data sets of different sizes. The datasets were created using a MATLAB routine, according to a set of probabilistic pseudo-random parameters, such as the frequency of the rare event, the Monitoring Time and the Warning. The performance of the algorithm depends on the size of the monitoring window, the number of different items and the frequency of the target event. Below, we present an experimental setup that has the following configuration: There are eleven different items in the database, including the target item. The Monitoring Time was set to 10 and the Warning Time to 5, which means that, according to the m-w+1 formula the size of the Monitoring Window was 6. Therefore, the transformed database contains 66 different extended items. The target event frequency was set between 9% and 10%, therefore, the transformed database contained approximately $0.1 \times DatabaseSize$ Mega Transactions.

The experiments were taken on a Pentium 3, 1GHz computer with 512MB of RAM and a SCSI hard disk. Figure 3 illustrates the performance of our algorithm with respect to the number of Mega Transactions of the transformed database. The times shown below do not include the run time of the frequent itemset mining algorithm we used (FP-Tree), since any such algorithm can be used, are indicative however of the efficiency of our algorithm. More information about FP-Tree can be found in [10].

DB Size (in transactions)	Mega Transactions	Run Time (seconds)
5000	479	0.08
10000	962	0.12
50000	4885	0.6
100000	9424	1.22

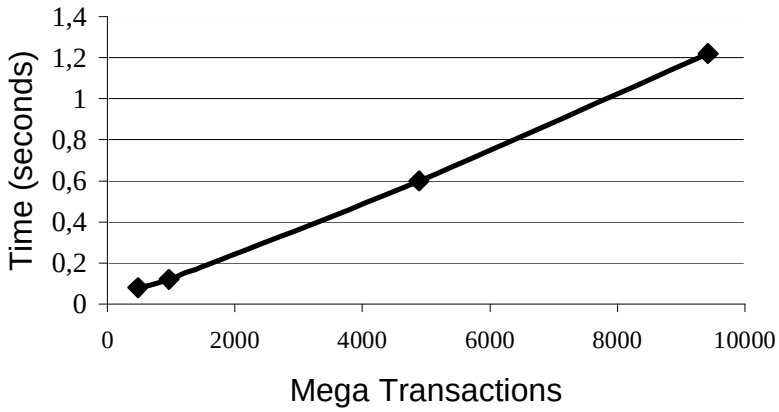

Fig. 4. Run time against the number of Mega Transactions

In all cases, the expected rules predicting the target event were successfully discovered. Our approach is complete due to the completeness of FP-Tree. In our experimental setup, a frequent extended itemset starting from time point 0 and ending at time point 4 can predict an event that will happen at time point 11. For example, given that the target item is *x*, a frequent extended itemset such as $\{a_0, b_1, b_2, d_4\}$ can be used for the prediction of *x* at time point 11 with a level of support.

The candidate rules produced from the frequent extended itemsets are required to satisfy the minimum confidence criterion set by the user. The confidence of the rule is calculated as defined in formula (2) in paragraph 3.

5. Conclusions and Further Research

In this paper we proposed PREVENT, a novel data mining approach for predicting rare events in transaction databases fast and accurately. Our approach is based on the inter-transactional association rules framework and utilizes a state-of-the-art algorithm for classical association rules mining, namely FP-Tree, in order to produce predictive patterns. It involves a database transformation in order to extract only the required information before mining for the predictive patterns. We formulated the problem, proposed a novel algorithm and conducted experiments to test and verify its performance.

PREVENT features low computational cost, since it requires only two scans over the database. Additionally it is also memory efficient as it uses small memory based structures,

except from the FP-tree, which in extreme cases can be relatively large. However, the FP-tree approach is a well established, complete and one of the most efficient approaches in the literature, which is the reason we selected it. The overall approach is complete due to the completeness of the frequent itemset algorithm.

For future research we consider conducting a large series of real world experiments, compare PREVENT to other approaches and finally, extend our approach for distributed databases, such as web databases.

References

[1] Hongjun Lu, Ling Feng, Jiawei Han, Beyond intratransaction association analysis: mining multidimensional intertransaction association rules, ACM Transactions on Information Systems (TOIS) Volume 18 , Issue 4 (October 2000) , ACM Press, NY, USA

[2] Anthony K. H. Tung, Hongjun Lu, Jiawei Han, Ling Feng, Breaking the barrier of transactions: Mining Inter-Transaction Association Rules, KDD '99, ISSN:1046-8188, Pages: 423 - 454

[3] Anthony K. H. Tung, Hongjun Lu, Jiawei Han, Ling Feng, Efficient Mining of Inter-transaction Association Rules, IEEE Transactions On Knowledge And Data Engineering, Vol. 15, No. 1; January/February 2003, pp. 43-56

[4] Ling Feng, Tharam S. Dillon, James Liu: Inter-transactional association rules for multi-dimensional contexts for prediction and their application to studying meteorological data, Data and Knowledge Engineering, Vol. 37, Issue 1, April 2001, Pages 85-115

[5] H. Lu, J. Han, and L. Feng. Stock movement prediction and n-dimensional inter-transaction association rules. In Proc. ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, pages 12:1--12:7, Seattle, Washington, June 1998

[6] Rakesh Agrawal and Ramakrishnan Srikant. Mining Sequential Patterns. In Proc. of the 11th Int'l Conference on Data Engineering, Taipei, Taiwan, March 1995

[7] R. Agrawal, M. Mehta, J. Shafer, R. Srikant, A. Arning, and T. Bollinger. The Quest data mining system. In Proc. 1996 Int. Conf. Data Mining and Knowledge Discovery (KDD'96), pages 244 -249, Portland, Oregon, August 1996.

[8] H. Mannila and H. Toivonen and A. I. Verkamo. Discovering Frequent Episodes in Sequences. In Proc. First International Conference on Knowledge Discovery and Data Mining (KDD-95). AAAI Press. Montreal, Eds. U. M. Fayyad and R. Uthurusamy , Canada, 1995.

[9] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule Discovery from time series. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98). AAAI Press, 1998.

[10] J. Han, J. Pei, and Y. Yin, Mining Frequent Patterns without Candidate Generation, Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00), Dallas, TX, May 2000.

[11] G. M. Weiss and H. Hirsh, Learning to Predict Rare Events in Event Sequences, In Proc. 4th International Conference on Knowledge discovery and Data Mining, AAAI Press, 1998, 359-363,

[12] P. Brockwell and R. Davis, Introduction to Time Series and Forecasting. Springer-Verlag New York, 1996.