Proceedings 3rd Int. Conf. on Data Mining Methods and Databases for Engineering, Finance and Other Fields Data Mining 2002, Bologna (I), 25 - 27 September 2002. pp. 655-664. WIT Press.

Signing stock market situations by means of characteristic sequential patterns

M.Leleu[1,2], J.F. Boulicaut[1]
[1]*LISI, INSA Lyon, France*
[2]*Direction de la Stratégie, Informatique Caisse des dépôts, Paris, France*

Abstract

We are studying new techniques for computing similarities between stock market situations. The situations are represented by means of event sequences (tens or hundreds of event types, hundreds or thousands of events). These events are obtained from available financial data (e.g., discretized financial times series, Reuters financial news). Interestingly, event sequences enable to mix quantitative and qualitative information. In order to analyse the situations and, e.g., to overcome direct similarity measure limitations, we compute signatures for the situations. The signatures are made of sequential patterns and we consider different possibilities for the selection/computation of characteristic patterns. Signatures contain not only the so-called frequent sequential patterns but also some infrequent ones (e.g., a sequential pattern that indicates a crash on some values).
We discuss an ongoing application that concerns the study of "market trends". Experts already identified in a collection of situations three kinds of periods (trend to a raise, trend to a decline, stability) and we compute the signatures of these labelled situations, looking for characteristic and discriminant patterns. The ultimate goal is then to classify a new situation by looking at its signature and the similarities with labelled signatures.

1. Introduction

Financial market evolutions are described by both qualitative and quantitative data with a temporal nature. Our data mining approach has to cope with large data streams while taking into account all these aspects. Therefore, we chose to represent financial market situations by means of event sequences. These events are obtained from market data (e.g. discretized financial time series) and from the financial news provided by specialized press-agency (e.g., Reuters or Bloomberg). We are then looking for signatures of sets of market

situations. For instance, in our current application domain, the data is available as a collection of situations that have been labeled by experts as being associated to “trend to raise”, “trend to decline”, or “stable market. To support the analysis of these situations and, as an ultimate goal, to support the classification of new situations, we chose to sign each class with a collection of characteristics sequential patterns. These patterns are mined from the sequences of events.

Representing market situations by events sequences allows an homogeneous view of qualitative and quantitative data streams. Quantitative data (time series) can be transformed into an event sequence by performing, e.g., a discretization approach based on clustering [1]. In [1], an alphabet of events is computed and used to encode the time series as a sequence of symbols over this alphabet. Qualitative information can obviously be considered as a sequence of dated events. These sequences can be merged straightforwardly. Figure 1 is an example of such a sequence where alphabet {A,B,C} is a representation of bound movements whereas events of type N_i are used to represent qualitative data that have been selected by experts. For instance, at time 2, we can observe simultaneously events B and N_{12}, where B could means “important increase of the bounds” and N_{12} “announce of a merge of companies”. Provided sequences can be composed of several hundred events from an alphabet of more than one hundred items. In other terms, scalability of the processing algorithm is a major issue.

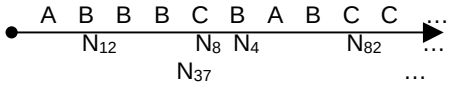

Figure1: Example of a stock market situation as a sequence of events

Once market situations are represented by event sequences, we are looking for similarity measures between these sequences. It could be possible to use direct measures like an edit distance (see, e.g., [2]) but it lacks from context-sensitivity (see Section 2). In this paper, we propose to compute symbolic signatures of these situations. These signatures are collections of sequential patterns that occur in the sequences. They can be understood by experts and then support the analysis of a specific situation.

In the data mining community, the computation of the sequential patterns that satisfy some user-defined constraints has been studied since 1995 [3, 4, 6, 7]. It has lead to several algorithms that can process huge sequences and remain tractable with large alphabets. Recent research focuses on the constraint-based extraction of sequential patterns for various constraints, i.e., when the minimal frequency is only one of the criteria that specifies the relevancy of patterns. Using additional constraints not only can reduce the number of extracted patterns (more focus on potentially interesting patterns given the user needs) but also can be used to reduce drastically the search space. Examples of such constraints are the specification of a minimum or maximum time interval between the occurrences of two events inside a pattern [5], the definition of a regular expression that must match with the

extracted patterns [11], or a constraint of similarity with a consensus pattern [8].

In this paper, we consider various possibilities for signing sequences of events and thus characterizing market situations. The next section introduces the problem. Section 3 describes the different type of patterns that can be used to build the symbolic signatures. Section 4 presents an experimentation on real stock market data, showing that the approach is of practical interest for the financial expert. Section 5 is a conclusion and a presentation of directions for future work.

2. Definitions

2.1. Events Sequences and Sequential Patterns

Let I = $\{i_1, i_2, \ldots, i_m\}$ a set of m distinct items constitutes the alphabet. An *event* (also called *itemset*) is a non empty set of items from I : $(i_1\, i_2 \ldots i_p)$, sorted in the lexicographic order. The size of event is p. A sequence α is an ordered list of events, denoted as $\alpha_1 \rightarrow \alpha_2 \rightarrow \ldots \rightarrow \alpha_q$. The *length* of α is q, that is the number of events in the sequence. Its *width* is the maximum size of all its events. A sequence with k items is called a *k-sequence*. For example, the sequence B $\rightarrow$ ACD $\rightarrow$ CDFG is a 8-sequence, its lengtg is 3 and its width is 4. A *sequential pattern* is a subsequence of a sequence. We will de interested in the so-called frequent sequential patterns occurring in a collection of sequences, i.e., sequential patterns that occur in enough sequences from the collection given a user-defined threshold.

A *prefix* of a k-pattern *m* is a sub pattern of *m* constituted with the k-1 first items of *m*. For example, the prefix *p* of the pattern A $\rightarrow$ BC is the sub pattern A $\rightarrow$ B. A *suffix* of a k-pattern corresponds to its last item. Thus, suffix *s* of A $\rightarrow$ BC is item C.

We distinguish *sequence patterns* from *event patterns*, which depends on the temporal relation between the prefix and the suffix of a pattern. A pattern is called an *event pattern* (ps) if its suffix *s* occurs at the same time than the prefix *p*. If the suffix occurs after the prefix, the pattern is called a *sequence pattern* (p$\rightarrow$ s). For example, pattern m_1 = AB$\rightarrow$C$\rightarrow$BDF has pattern AB$\rightarrow$C$\rightarrow$BD as a prefix *p* and item F as a suffix *s*. This is an *event pattern*. Pattern m_2 = AB$\rightarrow$C whose prefix *p* = AB and suffix *s* = C is a *sequence pattern*.

2.2 Similarity between Events Sequences

Several techniques, e.g., measures of edit distances, can be used to compute similarities between sequences of events (see, e.g., [10] for a recent overview). An edit distance attempts to measure how much work is needed to transform one sequence of events into another one. It is based on the use of edit operations such as insertion, deletion and substitution of events. There are, in general, several possible sequences of edit operations to transform one sequence into another, and the one with a minimal transformation cost is then considered. The cost of an edit operation can be more or less important to penalize some events and give more importance to some others (e.g., rare

events). However, the importance of an event often depends on its context. Thus, an event ε can become important only if it appears near another event ε'. An edit distance cannot directly deal with such a situation and getting such a knowledge from experts is a tedious task. By the computation of signatures, we capture at least partially the context and our thesis is that it is quite useful for context-sensitive measures of similarities between sequences.

2.3. The Proposition

We propose to sign event sequences with a collection of characteristics patterns that are symbolic characterizations of the sequences. All or part of these patterns are sequential patterns to take into account the temporal aspect of the data. Using this kind of representation, we might, e.g., characterize a situation with a pattern such as "Announce of an increase on American base rates has been followed by a raise on CAC40 indices".

3. Presentation of Sequential Patterns Forming Signature

Sequential patterns mining from a sequence or a set of sequences has been studied for a while. First, we consider the search for frequent pattern [3, 9, 4]. Then, we consider the so-called characteristic patterns.

3.1. Frequent Patterns

The main problem for frequent sequential pattern mining concerns the number of candidate patterns. Indeed, the number of patterns with a length l and a width L formed from an alphabet of size n can be in the worst case:

$$\sum_{i=1}^{l} \left(\sum_{j=1}^{L} C_n^j \right)^i$$

This number is exponential w.r.t. the number of events. To reduce the size of the search space, effficient safe pruning strategies have been designed (using the anti-monotonicity of the minimal frequency constraint, the so-called "apriori trick" [12]). We do not detail this well-documented approach [3,4,6,9]. However, in the stock market area, characteristic patterns are not only frequent patterns. Indeed, a rare pattern can really be characteristic of a situation. Furthermore, a frequent pattern is not necessarily a characteristic pattern.

3.2. Characteristic Patterns

Dealing with market sequences, it is not surprising that beyond a certain time span, two financial events have no more impact on each other. For example, experts consider that a firm restructuring project will no more affect CAC40 variations two months after its announcement. This kind of knowledge can be used during the extraction process, under the form of minimum or maximum time span that between two event occurrences. Another similar constraint,

called time window constraints, enables to limit the maximum time between the occurrence of the first event pattern and the last one. In the same way, financial experts can specify that the presence or the absence of a given event have a significant impact on market situations. These kinds of constraints, called domain-dependent constraints, enable the expert-driven definition of the subjective interestingness of patterns, i.e., to soecify characteristic patterns. For example, let uss consider the three sequences represented in Figure 2.

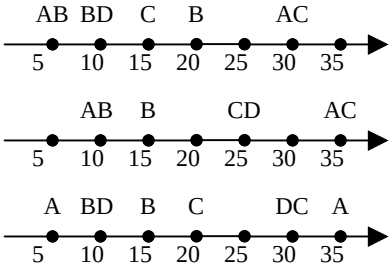

Figure 2: Set of sequences

In these sequences, pattern AB $\rightarrow$ C which is considered as frequent (i.e., it appears often enough among the three sequences), indicates that event C occurs often after event AB has occurred. If a constraint enforces that a maximum time between two events is equal to 10 time units, then occurrence of pattern AB $\rightarrow$ C in the second sequence will not be considered since there are 15 time units between AB and C. The pattern has become infrequent and will not be extracted.

Characteristic patterns might be infrequent. For example, a characterisic pattern could be a pattern "centered" on some rare event (e.g., a stock market crash). Once this important events have been identified, looking at such patterns is not computationally expensive. If they are not defined, it is almost sure that they can not be found by a post-processing step over the collection of frequent sequential patterns.

It is possible to combine the minimal frequency constraint and domain-dependant constraints. Considering a "generate and test" approach ("generate the patterns and then check for the constraints" is generally unfeasible. Fortunately, recent progress enables to use the constraint to optimize the search [9, 5]. Sets of extracted patterns are acceptable and their relevancy increase as well thanks to the user-defined knowledge, i.e., the provided constraints. Notice however that providing the constraints can be tedious when expert users are not available.

Finally, consider our current application (see Section 4) for which we are searching characteristic patterns for labelled sets of sequences, say Class 1 for Dataset 1 and Class 2 for Dataset 2. In that context, characteristic patterns for Class 1 might be such that they are not characteristic patterns for Class 2. A first idea is that these patterns might be frequent in Dataset 1 and infrequent in Dataset 2. This search for discriminant patterns is studied actually in other data mining context like, e.g., molecular fragment finding [13].

3.3. The cSPADE Algorithm

We implemented the cSPADE algorithm [5] to support the computation of signatures. This algorithm enables to integrate various constraints, such as the definition of a maximum/minimum gap and the inclusion/exclusion of given items. This algorithm extends Spade [6]. It uses the equivalence suffix class notion which allows an a pattern generation per class, i.e., it is not necessary to have information about prefix class [A $\rightarrow$ B] when generating patterns in class [A $\rightarrow$ C]. In our implementation, we preferred equivalence prefix classes, that is classes grouping all k-patterns with the same k-1-prefix. For example, class [A] contains all patterns prefixed by event A, such as patterns AB, A $\rightarrow$ B, A $\rightarrow$ C and so on.

The sequences from the input database, are represented by a unique identifier *sid* corresponding to the number of the sequence. The database representation is called *vertical*, i.e., for each pattern X occurring in the data, an occurrence list (*IdList*) denoted by *L(X)*, is associated. This list contains all information needed for the generation process: all pairs of input sequences (*sid*) and event identifier (*eid*) where pattern X occurs. Figure 3 is an example of such a representation:

sid	Time	items
1	10	C D
1	15	A B C
1	20	A B F
1	25	A C D F
2	15	A B F
2	20	E
3	10	A B F
4	10	D G H
4	20	B F
4	25	A G H

A	
sid	*eid*
1	15
1	20
1	25
2	15
3	10
4	25

B	
sid	*eid*
1	15
1	20
2	15
3	10
4	25

D	
sid	*eid*
1	10
1	25
4	10

F	
sid	*eid*
1	20
1	25
2	15
3	10
4	20

Figure 3 : Input Sequences and IdList from items A, B, D and F

The frequency of a pattern X in *L(X)* is obtained by computing the ratio of the number of distinct *sids* present in its IdList by the number of sequences. For example, frequency of pattern D is 50% because it occurs in two sequences (1 and 4) for a total number of 4 sequences.

The set of k-patterns is obtained by performing successive joins between two patterns called *generator patterns*. These patterns are k-1-patterns that share a same k-2-prefix. Two kinds of joins are used for the generation of a pattern:

- *Temporal join*

  This join is used to find all occurrences of a pattern *m1* preceding occurrences of another pattern *m2*. That means that it allows to find all

pairs (s, t_1) from $L(m_1)$ verifying condition s = s' and $t_1 < t_2$ where (s', t_2) is a pair from $L(m_2)$.

- *Equivalent join*

  Performing an equivalent join between two patterns m_1 and m_2, enables to find all occurrences of pattern m_1 and m_2 that occurred at the same time. Thus, for a pair (s, t_1) from $L(m_1)$, it is looking for a pair (s,t_2) from $L(m_2)$ which has the same *sid* s and with $t_1 = t_2$.

Considering two generator patterns m_1 et m_2 and their shared prefix p, the considered join will depend on nature of the patterns (c.f. Section 2.1):

- *Event pattern* with *event pattern* :

  Let m_1 = pB and m_2 = pF. The generated pattern is then pBF and its *IdList* is obtained doing an equivalent join between $L(m_1)$ and $L(m_2)$.

- *Event pattern* with *sequence pattern* :

  If m_1 = pB and m_2 = p $\rightarrow$ A, the generated pattern will be the sequence pattern pB $\rightarrow$ A whose *IdList* is obtained from a *temporal join* between $L(m_1)$ and $L(m_2)$.

- *Sequence pattern* with *sequence pattern*:

  Considering patterns m_1= p$\rightarrow$A et m_2 = p$\rightarrow$F, there are three patterns resulting of their join:

  - An event pattern p$\rightarrow$AF that is an *equivalent join* between $L(m_1)$ and $L(m_2)$.
  - A sequence pattern p$\rightarrow$A$\rightarrow$F that is *a temporal join* between $L(m_1)$ et $L(m_2)$.
  - A sequence pattern p$\rightarrow$F$\rightarrow$A that is *a temporal join* between $L(m_2)$ and $L(m_1)$.

  In case where m_1 = m_2 = p$\rightarrow$A, there is only one generated pattern p$\rightarrow$A$\rightarrow$A which is obtained by a *temporal join* between $L(m_1)$ et $L(m_2)$.

For example, the *IdList* of pattern D$\rightarrow$B is obtained thanks to a temporal join between $L(D)$ et $L(B)$ and is illustrated in Figure 4.

D$\rightarrow$B	
sid	*eid*
1	15
1	20
4	25

Figure 4 : *IdList* $L(D\rightarrow B)$ of pattern D$\rightarrow$B

Our implementation of the cSPADE algorithm decomposes the lattice of all the patterns in smaller independent lattices, each representing a so-called equivalence prefix-class (suffix-class in the original version [5]) and proceeds to a depth-first search on these lattices.

First, the algorithm generates patterns of size 1, i.e., class [∅]. Then, it proceeds to the generation of all equivalence classes of 2-frequent patterns from patterns in [∅], and so on until no more frequent patterns is generated. The reader should refer to [5,6] for a detailed description.

4. Experimental results

We used real datasets provided by 'Caisse des Dépôts et Consignations' to have a preliminary evaluation of the approach. These datasets are composed by all data concerning a unique indices during Year 1999. Three kind of trends - market raise periods, market decline periods and market stability periods - have been identified by financial experts. Clearly, in this context, the data consitute a learning set (labelled sequences) and is not that large. Furthermore, this research is ongoing and further experiments will consider larger datasets since the scalability of the technique is effective [5].

We have three sets of market situations that are first represented by means of event sequences. Then, we have three sets of input sequences from which we extract characteristics patterns. These signatures are considered as a guide for the experts in market analysis. It provides information such as: 'for how long he/she has to focus his/her attention to find relevant characteristics inside a market trend?', 'Is there some interesting and characteristics facts inside the trend class it deals with?', etc. Beyond this feedback, signatures also provide a symbolic representation of these facts, giving him/her new arguments to support his/her analysis. Details about the classes are presented in Figure 5.

	Raises	Declines	Stabilities
Number of sequences	123	98	95
Average length	20	22	19
Average size	122	135	116

Figure 5: Description of the market trend classes

We used our implementation of cSPADE on each class to build the signature. Our execution time was quite acceptable, just a few minutes for each extraction. Extractions have been made using different frequency thresholds (25; 50; 75 and 100%) and a constraint on the length (fixed to 3 in this experiment). Details about the results of these extractions are presented in Figure 6.

The number of selected patterns corresponds to the number of patterns kept to constitute the signatures. Indeed, a lot of extracted patterns are common to the three classes, i.e., they can not be considered as discriminant for the market trend. These patterns have been removed from the signatures.

Furthermore, it appears that using a greater frequency threshold does not provide better characteristic patterns for a class. Indeed, the frequency of non frequent patterns of one class can be close to the user-defined threshold. In this case, these infrequent patterns will not be selected whereas their frequency is also important. Thus, it is impossible to assert whether or not

extracted patterns are relevant for a class. To do that, it is important to check beforehand that these patterns have a little frequency in the others classes. An alternative would be to look directly at discriminant patterns, i.e., patterns that are frequent in one dataset and infrequent in the others by adapting the technique in [13].

	Raises				Declines				Stabilities			
Frequency (%)	25	50	75	100	25	50	75	100	25	50	75	100
Maximum length	3	3	3	3	3	3	3	3	3	3	3	3
Number of patterns extracted	5643	79	3	0	7504	173	3	0	5203	93	5	0
Number of selected patterns	1575	7	0	0	2922	86	0	0	864	14	1	0

Figure 6: Descriptions of results

We have computed sets of patterns that occurred in more than 50% inside the sequence of a class and not in more than 20% in the others two classes. We got 4 characteristic patterns for the 'raise' class, 29 patterns for the 'decline' class and 16 patterns for the last class.

5. Conclusion and future works

We presented an approach for the computation of characteristic information for market situations represented by a set of event sequences. A preliminary experimentation has been done for market trend analysis. This kind of representation is useful during the search for similarities between market situations. Indeed, instead of directly search for similarities among a set of event sequences, the financial expert might use information provided by the signatures, i.e., clues about 'what is really specific of a trend?', 'where does experts have to focus their attention?' and, for example, 'why this period has been identified as a market raise period?'. A future application of this approach will be to classify current market situations to make 'predictions' that is, to be able to anticipate an important trend reversal.

Future work will concern each part of the process, that is the pre-processing of the market data to represent them by events sequences (including qualitative information that has not been integrated yet), the efficient extraction of sequential and important patterns , and the post-processing of the results. Pre-processing deals with questions such as "which relevant information will be considered to make the sequences?" and "which pertinent threshold choose for discretize the quantitative data. The extraction step also needs background knowledge to define the various constraints, that is, e.g., to decide that patterns which have a too long duration are no more relevant (window time constraint). Then, the pattern post-processing also needs relevant criteria for deciding to keep a pattern in the signature or not.

References

[1] Das, G., Lin K.I., Mannila, H., Renganathan, G., && Padhraic Smyth, P. Rule Discovery from Time Series. *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*, New York (USA), AAAI Press: pp.16-22, August 1998.

[2] Gusfield, D. *Algorithms on strings, trees and sequences*: *Computer Science and Computational Biology*, Cambridge University Press: New-York (USA), 1997.

[3] Agrawal, R., && Srikant, R. Mining sequential patterns. *Proc. of the Eleventh International Conference on Data Engineering (ICDE'95)*, Taipei (Taiwan), IEEE Computer Society: pp. 3-14, March 1995.

[4] Mannila, H., Toivonen, H., && Verkamo, A.I. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3), pp. 259 - 289, November 1997.

[5] Zaki, M.J. Sequence mining in categorical domains: incorporating constraints. *Proc. 9th International Conference on Information and Knowledge Management (CIKM'00)*, Washington DC (USA), pp 422-429, November 2000.

[6] Zaki, M.J. SPADE: an efficient algorithm for mining frequent sequences. *Machine Learning*, Special issue on Unsupervised Learning, Vol. 42 (1/2), pp 31-60, Jan/Feb 2001.

[7] Masseglia, F., Poncelet, P., && Teisseire, M. Incremental mining of sequential patterns in large databases. *Proc. of 16ièmes Journées Bases de Données Avancées (BDA'00)*, Blois, France, pp. 345-366, october 2000.

[8] Capelle, M., Masson, C., && Boulicaut, J.F. Mining frequent sequential patterns under a similarity constraint. *Proc. Third International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'02)*, Manchester, United Kingdom. August 2002. To appear as a Springer-Verlag LNCS Volume.

[9] Agrawal, R., && Srikant, R., Mining sequential patterns: generalizations and performance improvements. *Proc. of the 5th International Conference on Extending Database Technology (EDBT'96)*, Avignon, France, Springer-Verlag LNCS 1057: pp. 3-17, March 1996.

[10] Moen, P. *Attribute, event sequence, and event type similarity notions for data mining*. Ph.D. Thesis, Report A-2000-1, University of Helsinki, Department of Computer Science, February 2000.

[11] Garofalakis, M., Rastogi, R., && Shim K. SPIRIT: Sequential Pattern Mining with Regular Expression Constraints. *Proc. 25th International Conference on Very Large Databases (VLDB'99)*, Edinburgh, United Kingdom, Morgan Kaufmann, pp. 223-234, September 1999.

[12] Agrawal, R., Mannila, H., Srikant, R., Toivonen H., && Verkamo A.I. Fast Discovery of Association Rules. *Advances in Knowledge Discovery and Data Mining*, AAAI Press, pp. 307-328, 1996.

[13] De Raedt, L., && Kramer, S.. SPIRIT: Sequential Pattern Mining with Regular Expression Constraints. *Proc. 17th International Joint Conference on Artificial Intelligence (IJCAI'01)*, Seattle, United States of America, Morgan Kaufmann, pp. 853-862, August 2001.