Sequential Pattern Mining of Price Interactions

Nuno C.Marques[1] and Luis Cavique[2]

[1] CITI and Departamento de Informática
Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa
[2] LabMAg e DCeT
Universidade Aberta
nmm@fct.unl.pt, lcavique@uab.pt

Abstract. The computational analysis of large quantities of data is an important asset for the economic study of interactions among social agents. However, most of available frequent pattern discovery techniques result in a huge number of rules and scalability problems that end up requiring unnecessary subjectivity in data interpretation. This work presents *Ramex-Forum*, a visualization technique that can highlight important relations often hidden in economic data. A case study using recent asset prices on global economic data confirm the usefulness of the approach for expressingeconomic influence cues as poly-trees.

Keywords: economic sequence pattern discovery, poly-trees

1 Introduction

This work proposes *Ramex Forum*[3], an extension of Ramex [4] for studying the tendencies detected in several historical prices. The goal is to mine influence cues (or possible market interest signals) that occur among several economic and social agents. Price correlations have been previously measured, namely by the Hurst exponent [9], a memory measure for time series dependency that is applicable to financial markets. The red line in figure 1 shows instantaneous Hurst values over price variations of the Dow Jones financial index. The observed Hurst value is always significantly above 0.65 (usually near 0.8). When Hurst values are bigger than 0.5, the global process is said to have fractal Brownian movement and the time series presents positive correlations. Sequence pattern mining will be applied for trying to detect and then analyze those possible correlations.

Ramex is an efficient sequential pattern mining algorithm capable of representing pattern interactions, where poly-trees are used as an alternative solution to the problems faced by most of the frequent pattern discovery techniques. Ramex algorithm has two valuable features that are needed to the present study, namely the production of a highly summarized and intuitive representation of economic influence cues and very good scalability:

[3] meaning *market* and *branch of a tree*, in latin.

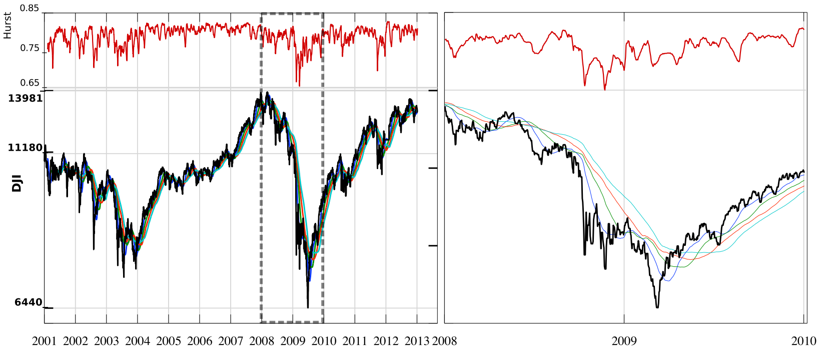


Fig. 1. The values for the Dow Jones Industrial Average index for the years 2001 to 2012 (heavy dark line). Top red line is the Hurst Index. Moving averages ranging from 40 to 160 trading days are respectively blue, green, light red and cyan lines. The graphic on left is a detailed zoom for the years 2008 − 2010.

1. Visualization: a poly-tree is used to represent most meaningful economic influence cues. Common association rules' systems that support the item set and sequence mining usually generate a huge number of rules, and therefore, it is difficult for the user to decide which rules to use and to represent graphically the solution.
2. Scalability: economists need interactive tools to study many concurrent economic measures. 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 an interactive study over very large databases.

Section 2 presents relevant work. A case study for economic data and consequent relevant financial measures are described in Section 3. Then Section 4, shows how to conjoin this relevant measures with Ramex sequential pattern algorithm (the joint algorithm is called *Ramex Forum*). Section 5 reposts a computational experience using *Ramex Forum* for the case study dataset. Finally, section 5, we draw some conclusions.

2 Related work

Sequence Mining on Financial data Financial data is available as a sequence of continuous values and both simple and multiple correlations can be applied [8]. However, in this work, the main concern is the type of relation and not its magnitude, so we are focusing the analysis of binary relations among financial products. In reality dependencies (and decisions) regarding buying or selling a given financial product are binary, so tendencies for buying or selling a given product may also be encoded as binary variables. Thus, market basket analysis — e.g., [1]— seems to be the most appropriate method to mine frequent sequences, namely for the analysis combinations of binary variables and the latency among detected tendence signals.

Most Frequent Sequence Mining algorithms use lattice structures in the search space. These algorithms include for breadth-first search, the Apriori-All algorithm and the GSP (Generalized Sequential Pattern) [1], [12] and for depth-first search the SPADE (Sequential PAttern Discovery using Equivalence classes) [13].

Markov chain is an acyclic graph with a set of states associated with a set of transitions between states. In the market basket each state corresponds to an item and in the user web navigation each state is a page. Markov models have been used to represent and analyze users web navigation data in [3]. At each time interval the system can change from a current state to the next state. The transition between states is quantified using probabilities. The first-order chain, is usually represented by a square matrix of the probabilities of the transitions from the current states to the following states. A second-order Markov chain takes into account the previous and the current state probability of the transition to the next state. The n^{th}-order matrix, with $n > 1$, grows into new dimensions ceasing to be square, as the first-order matrix. The higher-order chains consider sets of previous states leading to ordered sequences of states, while expanding the dimension of the original matrix and also the complexity of the problem.

As it was previously said, most of the frequent pattern discovery techniques present handicaps, namely by generating a huge number of rules that avoid a global visualization and by having scalability problems. Indeed, most of the tools can only represent the graph with all the arcs, which cause poor visualization when the graph is dense [2]. The poly-tree simplification is useful since it creates a highly readable graph.

Graph theory A short bibliographic overview of needed graph theory is presented.

Given a connected undirected graph, a spanning tree of the graph is a subgraph that connects all the vertices. 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 Kruskals algorithm, the edges are chosen without worrying about the connections to previous edges, but avoiding the cycles. In Prims algorithm the tree grows from an arbitrary root.

Maximum weight branching problem is the optimization problem that finds the largest branchpossible. It was proposed in [6] and is known as the Edmonds branching algorithm and is described in two steps: the condensation process, to remove the cycles, and the unraveling process where the branch is created.

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.

A poly-tree is a direct acyclic graph with a maximum of one path between any pair of nodes. The in-degree of any vertex of a tree is 0 (the root) or 1. On the other hand, the in-degree of a vertex in a poly-tree can be greater than 1. Both algorithms [6], [7] generate trees, with in-degree 0 or 1. To find the maximum weighted poly-tree, there is no optimal algorithm known.

3 Measuring Asset Price in Historical Data

Although a detailed economic analysis is outside of the scope of this paper, a case study is made using a set of high level common market indicators. Asset prices for financial products representing the world economy were selected during the recent economic crisis (years 2006 to 2012). The huge economic changes during this period are considered a challenging dataset for illustrating the usefulness of data mining algorithms over financial data. The present study used asset prices and trading volumes downloaded from Yahoo Finance for the following financial products ($\mathcal{P}$):

DJI - The Dow Jones Industrial Average, an adjusted market index for the 30 largest publicly owned companies based in the United States (figure 1).

DAX - Deutscher Aktien Index, an adjusted market index that follows the top 30 german stocks. During the analyzed period the German economy is considered a representative of the European economy.

HSI - Hang Seng Index is an adjusted market capitalization-weighted stock market index for the 48 largest companies in the Hong Kong stock market, and is considered as a representative of People Republic of China Economy.

NKY - Nikkei Stock Average is the most widely price-weighted index for Japanese equities. Japanese Economy is frequently related with China and it is still commonly considered a world leading Economy.

USO - United States Oil Fund, is a USA exchange-traded fund (ETF) designed to track the movements of light, sweet crude oil as traded on NYSE. Price variations in the oil price are a well known economic disturbing factor.

GLD - the largest physically backed gold exchange traded fund (ETF) in the world. Gold is a commodity that is traditionally used as a reserve currency for individuals, organizations and countries. During financial crisis, many have turned to Gold as a safe asset.

Asset Signal Indicator, I: Figure 1 illustrates four selected moving averages for DJI index, over the following n trading days periods: $n = 40$ (blue line), $n = 80$ (green line), $n = 120$ (light red line) and $n = 160$ (cyan line). Moving average indicators are commonly agreed as a good measure heuristic of the relative value given by investors to a given asset. From a engineering point of view, this measure is as a filter that removes high frequency data from the time series and gives the tendency for the price. Figure 1 clearly illustrates a desired behavior: the moving average is below the asset price during periods were the asset is increasing its value and below the asset price when the asset is decreasing its value. So, the measure for market indicator interest in a given financial product $\mathcal{P}$ is given by:

$$Ii(t,n) = \frac{I_{value}(t)}{MA(t,n)},$$

where, for a given instant t, when indicator is $I_{value}(t) = Price(t)$ the price of the asset $\mathcal{P}$ is used and when indicator is $I_{value}(t) = Volume(t)$ the total amount money traded for a given asset in that day is used. $MA(t, n)$ is the moving average of the values taken from time $t - n$ to the time t:

$$MA(t,n) = \frac{\sum_{i=t-n:t} I_{value}(i)}{n}.$$

Interest Type, T: For n consecutive trading days, a positive interest type (or signal) T in a given asset is marked as B (buy), and has the respective event counter increased for each occurrence, ie.:

$$Counter_B = Counter_B + 1 : Ii(t, n) >= 1.05,$$

and a negative interest is marked as S (sell), and has event counter also increased:

$$Counter_S = Counter_S + 1 : Ii(t, n) <= 0.95.$$

When no Interest is detected, event counters are reset. This is easily achieved by considering an initial signal, counted in the stock market temporal series filtering out small instabilities: an initial count with exponential decay is used (exponent 0.7) and a signal is only considered when this value is above a significance level of 0.2. In figure 2 uses a log scale where the dot line represents the initial count of buying signals for $n = 40$ resulting from the price time series (continuous black line). The evolution of two average intensity detectors $Count_{\mathcal{P}_{Tlm}}$ $(I \in \{S, B\})$ that count how many days have passed within a given continuous signal, are represented in the dash or on the dash-dot line.

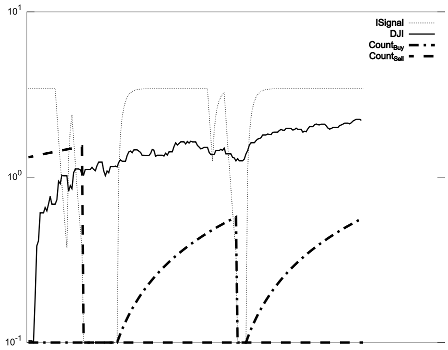


Fig. 2. Detail for the DJI Index in log scale, after a market recovery. Full line is the DJI price, the dotted line is a non-averaged counter and dashed or dashed dot lines are the $Count_{DJIBuy}$ and $Count_{DJISell}$ signal counters.

Stability of the Signal, m: The higher the number of trading days (n) observed, the more stable could the underlying a *market interest* be. Four values for stability m abstract moving average periods (illustrated in figure 1 for $\mathcal{P} = DJI$): respectively, $m = 1, m = 2, m = 3, m = 4$, for $n = 40, n = 80, n = 120$ and $n = 160$ trading days. $m = a$ will be used when a market interest is detectable in any of $1:4$ signals.

This way, interactive signals can be detected from the time series for asset $\mathcal{P}$, according to definition:

Definition 1. *A market signal ($\mathcal{P}_{TIm}$), measures interactions of type T ($T \in \{B, S\}$, with B for positive (or buy tendency) and S for negative interest (or sell tendency). Indicator I ($I \in \{P, V\}$, with P for Price and V for volume) on a period of type m ($m \in \{1, 2, 3, 4, a\}$).*

In the text when either the type of interaction T or the indicator I is omitted, all signals are jointly considered.

Market Influence is a co-occurence of any two market signals. The $Count_{\mathcal{P}_{Tlm}}$ (eg. figure 2) is calculated for each product. A directional co-occurrence of observed events can be detected with a several day lantency (δ) after the preceding event occurs. Based in previous empirical tests in [11] we have tested values for the δ parameter ranging from $n = 5$ trading days up to a fairly long interval of 160 trading days. This way, the present work assumes the following definition for observable market influence cues with a given latency δ:

Definition 2. $InfluenceCue(A, B, \delta)$*: For any δ trading period, an observable influence cue from market signal A to market signal B ($B \to A$) is considered present when:* $1 \leq Count_A - Count_B \leq \delta$.

Values of the Hurst Index for all studied asset prices were always in the $0.65 - 0.90$ range.

So, influence cues using daily trading prices and trading volumes width δ factor (set from 5 to 160 trading days) are considered as observable objective quantities for mining sequential correlations. The goal of this work is to provide a tool for the economist studying market interactions. However, measuring interactions only during large periods among several variables is no easy task.

4 The Poly-Tree Sequence Model

The Poly-tree Sequence model is generated by a two-phase algorithm: the transformation of the problem into a network and the search of the sequences.

```
Input: Financial dataset
Output: poly-tree of items
Proc 1: Calculate Network Transformation: build a state transition network;
Proc. 2: Find the most probable poly-tree sequence of items;
```

Algorithm 1: Two-phase procedure for the Poly-tree Sequence Model.

The original Ramex was applied to discrete sequences of purchases. In Ramex-Forum the Transformational phase is acquired from the financial data-set time-series set over daily price variations. For doing so, the financial dataset, in Table 1, each signal $\mathcal{P}_{TIm}$ (item), will be associated with $Count_{\mathcal{P}TIm}$ for the respective signal. According to definition 2, the type of signal is represented by a counter signal (B – buy as positive values and S – sell as negative values). Then, the difference between a pair of items, A and B, is between 1 and a constant δ, and $B < A$, Ramex will say that B influences A, ($B \rightarrow A$).

Table 1. example table taken out of time signals of two financial itens.

time instant	item A	item B	$\delta = 9$
2006-10-13	13	6	TRUE
2006-10-14	14	7	TRUE
2006-10-15	15	8	TRUE
2006-10-16	0	9	FALSE
2006-10-14	0	10	FALSE
2006-10-17	0	11	FALSE
2006-10-18	-1	12	FALSE
2006-10-19	-2	13	FALSE
2006-10-20	-3	14	FALSE

In our study four relevant transformation types were implemented: buy, sell, counter cycle (buy-sell), and all together (buy-buy, sell-sell or buy-sell). The general transformation procedure is the following.

```
Input: Financial dataset
Output: graph $G(vertice, edge)$
foreach time instant do
    foreach pair of items $(a, b)$ having a relevant transformation type do
        if $1 \leq a - b \leq \delta$ then
            $edge(a, b) = edge(a, b) + 1$;
        end
    end
end
```

Procedure 1: Transformation phase.

For each time instant in the dataset all pairs of items are compared. If a product B influences the product A, that is ($B \rightarrow A$), the $edge(B, A)$ is incremented in graph G. The graph, where cycles are allowed, condenses the information of the dataset by incorporating all influences. In graph G each state corresponds to an item and each transition represents the influence of one item over the subsequent item. The weight of each arc corresponds to the number of times that one item influences the next item. The transformation of a dataset into a network is identical in the Markov Chain approach.

In previous work, the Forward Heuristic, [4] a tree sequence was provided, given a graph G and a root vertex, by finding the highly probable branch sequence. Maximum Weight Rooted Branching algorithm was applied, as defined in Fulkerson [7]. Like Edmonds branching algorithm [6], Fulkersons algorithm has $O(N^2)$ in the worst case time complexity, where N is the number of vertexes.

Back-and-Forward Heuristic mode [4] is also based on the Prim algorithm. Different applications can use the Forward or Back-and-Forward heuristic. The Forward heuristic must be chosen if a starting node is given. For example, most of the web mining problems start in a root node, so we suggest this first mode. If there is no information about the starting node, the best edge should be chosen, and the Back-and-Forward heuristic can be applied.

To find the most probable Poly-Tree Sequence of financial items we are going to use the Back-and-Forward heuristic, since there is no information about a starting vertex. The internal representation of the problem is similar to Markov Chains. However, the poly-tree simplification in order to get a good visualization is a Ramex feature.

```
finds the most probable poly-tree sequence of items.
Input: graph $G$
Output: poly-tree $T$
Initialize $T$;
foreach $vertex \in G$ do
    foreach $edge \in G$ do
        Calculate $x$ = argmax(weighted forward-vertex not-visited in $G$
        and connected with $T$, weighted back-vertex not-visited in $G$ and
        connected with $T$) ;
    end
    Update solution $T$ with $x$;
end
```

Procedure 2: Back-and-Forward Heuristic.

5 Experimental Results

5.1 All Products and Signals

In a first experiment we have generated the graph for the full dataset. Since we were interested in any correlation, no distinction was made between selling and buying signals. A reasonable period of five trading days was used for studying short term dependencies. Results are shown in figure 3. We can easily notice a very systematic (and expectable) price dependency from short ($P1$) to long term ($P4$) strength signals (more than 200 observations, getting to 400 observations after $P2$ signal). A clear sequence of all signals is found in DJI, but DAX, GLD and NKY prices also show good correlations. However, some additional asset interactions are already detectable and disturb this basic and expectable relation.

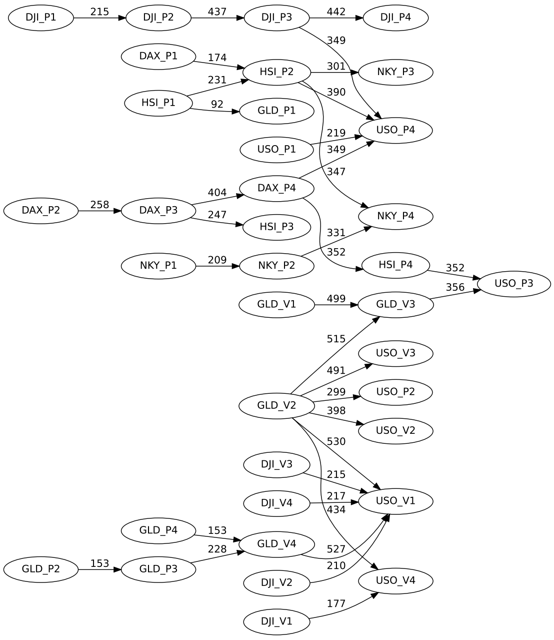


Fig. 3. Extracted correlation graph for all products and signal types with $\delta = 5$.

It is quite striking the influence between long term stock market for DJI, HSI and DAX and the oil price (USO): always above 300 observations for $P3 - P4$ signals. German DAX is also related with Chinese HSI in stable prices ($P4$) with more than 200 observations. It is also significative to notice the GLD dependency between price and volume in a period were sustainable growth was observed in gold market. Also high volumes in GLD_{V4} stable gold volume indicators are related with USO_{P3} price variations (356 observations). Short term price interactions ($P1$) were also selected between HSI and GLD (92 occurrences)

5.2 Joining price and volume interaction signals

Since stability interactions are quite trivial, it would be also interesting to join all interaction signals. So short and long term interaction stability (ie. $I1$ to $I4$, with $I = \{P, V\}$) were joined (defining Ia interaction). Ramex graph for five trading day interactions is shown in figure 4.

The acquire graph is now more simple and also confirms the previously made observations. Also, regarding volume dependencies (traditionally related with market tendencies), price variations in stock markets (DJI and NKY) are related with volume trends in GLD and USO (more than 150 observations). The main observed volume dependency is the GLD - USO volume dependency.

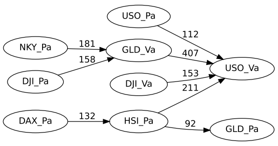
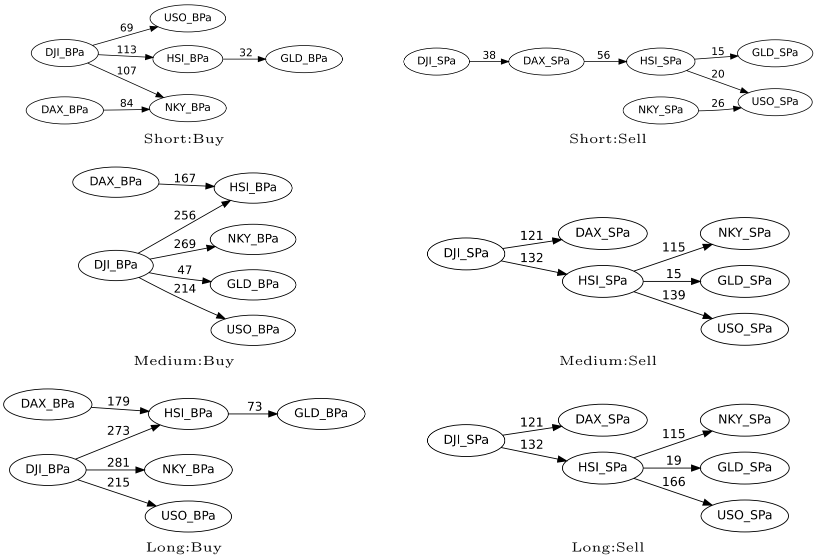


Fig. 4. Extracted correlation graph for all products with maximum signals for five trading days ($\delta = 5$).

5.3 Market Price Interactions

Main interest in markets is asset price. So, longer interactions are also shown considering longer number of trading days ($\delta = 40$ and $\delta = 160$). Figure 5 presents all these graphs.



Short:Buy

Short:Sell

Medium:Buy

Medium:Sell

Long:Buy

Long:Sell

Fig. 5. Extracted correlation graph for all selling and buying signals with *delta* in 5 (short), 40 (medium) and 160 (long) trading days.

Once again, previous dependencies were confirmed. Regarding Buy signals we should notice that USA DJI is influencing most major assets in the world. The only exception to this is European German DAX, that strongly co-influences Chinese HSI. A strong correlation is found between HSI and GLD tendencies in both short and long term dependencies. Regarding Sell signals, USA DJI reinforces its influence, now by way of Chinese and German indexes. Also sell signs on Chinese HSI keeps a strong influence on USO and Japanese NKY price and some influence on GLD prices.

6 Conclusions

In this paper we present a new application using the sequential pattern detection, extending the work [4]. This approach,presents a global view of the data, since all the items are taken into account, by condensing the information into a cyclic network. In our approach instead of using relative frequencies, we use absolute frequencies. A new Back-and-Forward heuristic is presented to find poly-trees inspired in the Prim algorithm.

Other ongoing work is trying to model quantitative short-time dependencies among distinct markets. The conjunction of the long term relations detected in this work with the magnitude of short time multivariate dependencies, is an interesting and challenging problem. Also, although the present study was applied to long term financial data-set time-series, other studies can be made with Ramex-Forum. The high efficiency of the method allows for much higher time resolutions than daily price variations, e.g., current financial data for high frequency trading already uses time resolutions in the millisecond range.

Achieved results show the interest of the proposed method. Long term stability dependencies in prices ($P1 \rightarrow P2 \rightarrow P3 \rightarrow P4$) were expectable, since we need unstable prices above/below the average ($P1$ signals) to have more stable ($P4$) signals. This both tested the validity of the method and shown some interesting instantaneous dependencies. Most of those decisions were still detectable in more high level analysis. Although only a high level Economic analysis was made, some interesting and expectable relations were found for the period under analysis. The US stock market DJI index is detected as the major influence in world Economy during the analyzed period. The analyzed period includes the recent 2008 financial crisis and major European crises. China is seen as a major player in world Economy. According to many analysts People's Republic of China was one of the major buyers of gold in the world. Indeed China still has fairly low reserves of gold when compared with traditionally wealthy countries such as the US or Germany. However China is also trying to make its currency a major player in world Economy, so its gold reserves are increasingly rapidly. Also China is one of the major buyers of debt. Probably because of that a high correlation was detected between German DAX and chinese HSI indexes. Almost all results seem to show that oil (USO) is still essential for Humanity's industrial energy needs, this is again another expectable result. But, as any other data-mining method, the proposed techniques can only measure observable correlations during a given historical period: the proposed approach is only extracting cues and can not fully determine market influences. Nevertheless, the proposed approach has the advantage of objectively counting possible economic dependencies.

References

1. Agrawal R., Srikant R.: Mining sequential patterns. In: Proceedings 11th International Conference Data Engineering, pp. 3–14. IEEE Press (1995)

2. Baragoin C. : Enhance Your Business Applications, Simple Integration of Advanced Data Mining Functions. IBM Information Management Red Books. (2002)
3. Borges J., Levene M.: Eval. Variable-Length Markov Chain Models of User Web Navigation Sessions. IEEE Trans. Knowl. Data Eng. 19(4), 441–452 (2007)
4. Cavique L.: Network Algorithm to Discover Sequential Patterns. In: J.Neves, M.Santos and J.Machado (eds.) Progress in Artificial Intelligence. LNAI, vol. 4874, pp. 406-414. Springer, Heidelberg (2007)
5. Cavique L., Coelho J. : Descoberta de Padrões Sequenciais utilizando Árvores Orientadas. Revista de Ciências da Computacão 3, 12–22 (2008)
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. Johnson R.A. and Wichern. D., Applied Multivariate Statistical Analysis, 6th Edition. Apr 2 (2007)
9. Mandelbrot B., Van Ness J. W. : Fractional Brownian motions, fractional noises and applications. SIAM review, 10(4), 422–437 (1968)
10. Marques, N.C., Gomes, C.: An Intelligent Moving Average. In: Proceedings of the 19th European Conference on Artificial Intelligence - ECAI 2010 (2010)
11. Marques N.C. and Gomes C. Maximus-AI: Using Elman Neural Networks for Implementing a SLMR Trading Strategy. In Williams M, Bi Y, (eds.), *Proc. 4th Int. Conf. on Knowledge Sc., Eng. and Mang.*, *LNCS* vol. 6291, 579–584. Springer (2010)
12. Srikant R and Agrawal R., Mining sequential patterns: Generalizations and performance improvements. Proceedings 5th International Conference Extending Database Technology, EDBT, 1057, 3–17 (1996)
13. Zaki M.J. : Spade: An efficient algorithm for mining frequent sequences. Machine Learning, 42, 3160 (2001)