Correlating Time-Related Data Sources with Co-clustering

Vassiliki Koutsonikola[1], Sophia Petridou[1], Athena Vakali[1,⋆], Hakim Hacid[2,⋆], and Boualem Benatallah[2,⋆]

[1] Aristotle University of Thessaloniki
[2] University of New South Wales

Abstract. A huge amount of data is circulated and collected every day on a regular time basis. Given a pair of such datasets, it might be possible to reveal hidden dependencies between them since the presence of the one dataset elements may influence the elements of the other dataset and vice versa. Furthermore, the impact of these relations may last during a period instead of the time point of their co-occurrence. Mining such relations under those assumptions is a challenging problem. In this paper, we study two time-related datasets whose elements are bilaterally affected over time. We employ a co-clustering approach to identify groups of similar elements on the basis of two distinct criteria: the direction and duration of their impact. The proposed approach is evaluated using time-related news and stock's market real datasets.

1 Introduction

A huge amount of data is circulated and collected every day on a regular time basis in order to understand and capture various physical and commercial phenomena. Such data involve the recording of electricity demand, temperature and percentages of humidity, fluctuations in a company's sales and a stock's price etc over a specific time period. It is also true that there is a high level of dependencies between such different datasets since, for example, weather changes affect electricity consumption, news reporting influences stock prices, commercial advertisements have impact on consumers acts etc.

A common choice is to model such datasets in the form of event time series which capture measurements describing the raw data at successive (often uniform) time intervals. Analyzing different datasets measurements considering time as their common feature may reveal dependencies and relevance between datasets which otherwise might not be obvious. Considering for example a time series that records the temperatures measured on a daily basis over the summer period and another one recording the values of energy demands for the same period, dependencies between temperatures and quantities of energy demands

[⋆] This work was supported by the DEST-ISL Competitive Grants Programme on "Efficient Management of Information Resources Over Ad-Hoc Data Grids", 2007-2008, UNSW, Australia.

J. Bailey et al. (Eds.): WISE 2008, LNCS 5175, pp. 264–279, 2008.

can be found. Such interactions may be further used in forecasting in order to predict future outcomes. News and market datasets are a quite representative such scenario, since, for example, news announcements would be an indication of either the increase or decrease in specific stocks' values. These datasets exist parallel in time and finding relationships and/or dependencies between them will largely impact investors, traders, journalists and news agencies. Relevance between these datasets seems to be of high interest since currently almost all major news agencies offer market information (such as stocks rates, flow) on their portals' index page.

Table 1. Related Work

Datasets	Representation		Methodology
	vectors	time series	
Financial news articles, stock market data [13]	✓		k-NN learning algorithm, regression analysis, neural networks
Newsgroups articles, historical stock market data [12]	✓		Natural Language Processing, Time Delay Neural Network
Archives of news articles and stock data [6]	✓	✓	Segmentation, Clustering, Support Vector Machines
News headlines, quoted exchange rate data [11]	✓		Classification rules
Historical data of electricity consumption and weather [10]		✓	Regression models

Existing approaches that attempt to find dependencies between time-related datasets combine ideas from various research areas such as data mining, engineering and mathematical programming. A summarization of these approaches is given in Table 1. According to the authors knowledge, all earlier work considers that there exists a "one-way" impact between the involved datasets. For example, works in [13], [12], [6] and [11] study the impact of news on financial data while in [10] electricity demands are forecasted in terms of meteorological parameters. However, these dependencies are often bidirectional as the authors realized from their experience in the ADAGE project [1]. In financial world, for example, a banks' corporation announcement could trigger significant stocks' fluctuation while a stock-market's crash would certainly raise political statements. Besides the direction, the duration of these interactions is also of high interest since the impact of one event to another may last for a period instead of the time point of their co-occurrence.

In this paper, we use co-clustering to simultaneously yield clusters of elements from two different but related over time datasets. Co-clustering is proposed as a more suitable approach, which has been used in grouping together elements from different datasets [2], [3], [7]. We apply the proposed model on coinstantaneous news and financial datasets to reveal dependencies between them. It was quite

[1] ADAGE project: http://cgi.cse.unsw.edu.au/ soc/adage/

challenging to deal with such distinct datasets due to their different formats and scales and their preprocessing was carried out by proposing data structures tailored to each dataset and resulting in formats that could be commonly processed. The experimentation results show that the proposed framework manages to reveal relations between news and stocks with respect to their short and long term interactions. Our main contribution is summarized in proposing a framework for clustering mutually affected datasets, by considering the following criteria:

- *duration of impact* is taken into account to construct the data sources representation structures (vectors). A time window parameter w is used to extend original data sources vectors in a common time-aware format which will involve duration in the co-clustering process.
- *direction of influence* is "embedded" in the calculation of similarities between data sources. A weight factor $\alpha \in [0 \ldots 1]$, which characterizes the role of a data source ranging from a triggering to a triggered status, is employed.

The remainder of the paper is organized as follows. Section 2 defines our problem whereas Section 3 analyzes the proposed co-clustering approach. Section 4 provides the experimentation on both synthetic and time-related news and stocks market data. The conclusions are presented in Section 5.

2 Problem Formulation

Consider two time-related datasets where elements of the first dataset trigger elements belonging to the other and vice versa. We define $A = \{a_1, \ldots, a_n\}$ to be the first dataset containing n elements, $B = \{b_1, \ldots, b_m\}$ to be the second dataset consisting of m elements and $T = \{1, \ldots, t\}$ to be the set of t time points. Then, for each a_i element, $i = 1, \ldots, n$, we define the vector $VA(i, :)$ to track its t measurements:

$$VA(i, :) = (\,VA(i, 1), \ldots, \,VA(i, t)) \tag{1}$$

where $VA(i, l)$, $l = 1, \ldots, t$, indicates the value of the element a_i during the time point l. All the $VA(i, :)$ vectors are organized in the n x t table VA. For

Table 2. Basic symbols notation

Symbol	Description
n, m	Number of elements in the two datasets
t	Number of time points
$A = \{a_1, \ldots, a_n\}$	Set of the first dataset elements
VA	nxt table of the A dataset measurements
$B = \{b_1, \ldots, b_m\}$	Set of the second dataset elements
VB	mxt table of the B dataset measurements
$T = \{1, \ldots, t\}$	Set of time points
w	Time window

the second dataset B, we similarly define the VB two dimensional m x t table which consists of m $VB(j,:)$ multidimensional vectors that provide the values of element b_j, $j = 1, \dots, m$, over time:

$$VB(j,:) = (\,VB(j,1), \dots, \,VB(j,t)) \tag{2}$$

The $VB(j,l)$ element indicates the value of b_j during the time point l.

Based on the above, we consider that a measurement of element a_i (b_j) on time point l i.e. $VA(i,l)$ ($VB(j,l)$) affects measurements on elements b_j (a_i) for a period starting from l and lasting up to $l + w - 1$ (w is the so-called time window), $w = 1, \dots, t$. In practice, for $1 \leq l \leq t - w + 1$ the time duration of impact is w, whereas for $t - w + 2 \leq l \leq t$ the corresponding impact time period may last from $w - 1$ down to 1.

In the proposed approach, it is important to identify the problem to be solved, since as mentioned earlier we are dealing with two distinct criteria: the bilateral relation between our datasets and the time period of their interaction (expressed by w). More specifically, given the set A of n elements, the set B of m elements, the set T of t time points, the time window w and the desired number of clusters k we are looking for k subsets such that each subset contains both a_i and b_j elements. Members of each subset should be strongly related in terms of both the direction and duration of their impact. Thus, based on the above, we can define the Time-Related Data Co-Clustering problem as follows:

Problem 1 (Time-Related Data Co-Clustering). Given two time-related datasets A and B of n and m elements respectively, a set T of t time points, the integers w and k and a *Similarity* function, find a set C of k subsets $C = \{C_1, \dots, C_k\}$ such that $\sum_{x=1}^{k} \sum_{a_i, b_j \in C_x} \mathit{Similarity}(a_i, b_j)$, $i = 1, \dots, n$ and $j = 1, \dots, m$, is maximized.

The *Similarity* function should capture our two main criteria, namely the bilateral influence between data elements and its duration.

3 Capturing Impact between Datasets

3.1 Measuring Similarities

A common measure used to capture similarity between two (same dimension) vectors is the *Cosine Coefficient* [14] which calculates the cosine of the angle between them. The cosine coefficient $CC(i,j)$ between two vectors $VA(i,:)$, where $i = 1, \dots, n$, and $VB(j,:)$, where $j = 1, \dots, m$, of the same length t is defined as follows:

$$CC(i,j) = \frac{VA(i,:) \cdot VB(j,:)}{|\,VA(i,:)| \cdot |\,VB(j,:)|} = \frac{\sum_{l=1}^{t} VA(i,l) \cdot VB(j,l)}{\sqrt{\sum_{l=1}^{t} VA(i,l)^2 \cdot \sum_{l=1}^{t} VB(j,l)^2}} \tag{3}$$

Using cosine coefficient to measure similarities between the vectors $VA(i,:)$ and $VB(j,:)$, we manage to capture their relation under the assumption that

$VA(i,l)$ and $VB(j,l)$ values can be related on the basis of time l. In our framework, a_i elements may trigger b_j elements and at the same time, b_j elements may trigger a_i elements for up to w successive time points after the triggering element has occurred. However, the correlation coefficient applied on $VA(i,:)$ and $VB(j,:)$ captures neither the bilateral relations nor the impact over the time window w. Thus, we firstly extend the $VA(i,:)$ and $VB(j,:)$ vectors in order to include time window w. Then, we use the correlation coefficient in conjunction with the extended vectors and we create two n x m tables, namely the AB and BA to represent the impact of a_i elements on b_j and vice versa.

Definition 1 (The extended triggering vector)**.** *Given a t-dimensional triggering vector $VA(i,:)$ with measurements of a period of t time points, and a time window w ($w = 1, \ldots, t$), we define its extended triggering t_1-dimensional vector $VAE(i,:)$, $t_1 \geq t$, which is constructed by putting up to w repetitive $VA(i,l)$ values, between each pair of the $VA(i,l)$ and $VA(i,l+1)$ measurements.*

Lemma 1. *Let $VAE(i,:)$ be the extended triggering vector of $VA(i,:)$. Given that the $VA(i,:)$ consists of t measurements which will be repeated up to w successive times and t_1 represents the number of values of the $VAE(i,:)$ vector, it holds that:*

$$t_1 = t * w - \frac{w(w-1)}{2} \tag{4}$$

Proof. We split the t values of the $VA(i,:)$ vector into two groups. The first contains the values l, where $1 \leq l \leq t-w+1$, which will be repeated w successive times. The second group contains the values l, where $t-w+2 \leq l \leq t$, which will be repeated from $w-1$ down to 1 times (i.e. the $t-w+2$ value will be repeated $w-1$ times, the $t-w+3$ value will be repeated $w-2$ times and that will continue until the last t value that will be presented 1 time). This second group contains a number of elements equal to an arithmetic sequence from 1 to $w-1$. As a result:

$$t_1 = (t-w+1) * w + \frac{w-1}{2} * w = t * w - \frac{w * (w-1)}{2}$$

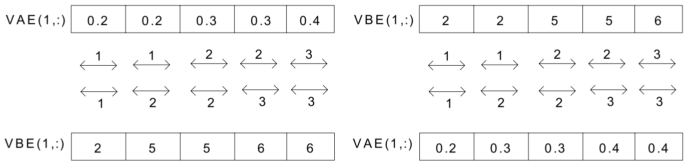
Definition 2 (The extended triggered vector)**.** *Given a t-dimensional triggered vector $VB(j,:)$ of t measurements of a period of t time points, and a time window w ($w = 1, \ldots, t$), we define its extended triggered t_2-dimensional vector $VBE(j,:)$, $t_2 \geq t$, which is constructed by putting a group of up to w values $VB(j,l) \ldots VB(j,l+w-1)$ between each pair of the $VB(i,l)$ and $VB(i,l+1)$ measurements.*

Lemma 2. *Let $VBE(j,:)$ be the extended triggered vector of $VB(j,:)$. Given that the $VB(j,:)$ consists of t measurements which will be repeated as groups of maximum w successive values and t_2 represents the number of values of the $VBE(i,:)$ vector, it holds that:*

$$t_2 = t * w - \frac{w(w-1)}{2} \tag{5}$$

Proof. The proof is similar to that of Lemma 1.

Example 1. Let *VA(1,:)=[0.2 0.3 0.4]* denote the vector with the measurements describing data element a_1 for 3 time points and *VB(1,:)=[2 5 6]* the vector of element b_1 for the same time period. We assume that $w = 2$. Considering that a_1 triggers b_1, the value 0.2 of a_1 that was recorded on time point 1 affects the values 2 and 5 of b_1 on time points 1 and 2. Similarly, the value 0.3 of a_1 will affect values 5 and 6 of b_1 while the last value 0.4 of a_1 will affect only the last value 6 of b_1. Figure 1(a) shows the extended vectors and their relation. On the other hand, considering that b_1 triggers a_1 the extended vectors and their relation are depicted in Figure 1(b). □



(a) a_i measurements trigger b_j measurements (b) b_j measurements trigger a_i measurements

Fig. 1. The extended vectors for $w = 2$

From Lemmas 1 and 2, it holds that $t_1 = t_2$ and thus, we can use the $VAE(i,:)$ and $VBE(j,:)$ vectors, instead of $VA(i,:)$ and $VB(j,:)$, in Equation 3 to calculate their similarity. Since the impact of time window w is "embedded" into the extended vectors the cosine coefficient will capture the duration of impact between elements. Equations 4 and 5 indicate that the cost in terms of space complexity shifts from $O(t)$ to $O(tw)$ i.e. the space burden posed by the extended vectors depends on w value chosen.

Based on the above, we create the n x m AB and BA tables in order to proceed to the impacts' direction criterion. The $AB(i,j)$ element indicates the similarity between $VAE(i,:)$ and $VBE(j,:)$ (evaluated by Equation 3) in case that $VAE(i,:)$ acts as the triggering and $VBE(j,:)$ as the triggered vector, while the $BA(i,j)$ element indicates the similarity between the former vectors in case that $VBE(j,:)$ triggers the $VAE(i,:)$ vector. We notice that the values of elements of the AB and BA tables fall in the interval $[-1..1]$ as $CC(i,j)$ expresses the cosine of the angle that these vectors define.

Next, we combine the information of tables AB and BA in the formula of the $Similarity(a_i, b_j)$ function via a weight factor α:

$$Similarity(a_i, b_j) = \alpha * AB(i,j) + (1 - \alpha) * BA(i,j) \tag{6}$$

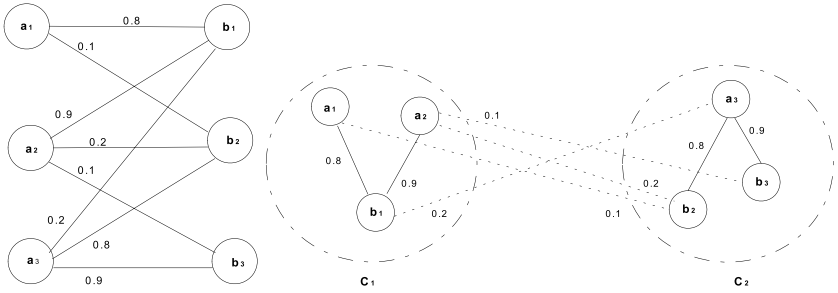
The values of α fall in the interval $[0..1]$ differentiating the significance of the bilateral relation between a_i and b_j data elements. More specifically, when $\alpha = 1$, $Similarity(a_i, b_j) = AB(i,j)$, we consider solely the impact of a_i to b_j, while for $\alpha = 0$, $Similarity(a_i, b_j) = BA(i,j)$, we consider that only b_j affects a_i. For any other value of α, we consider a mutual impact between a_i and b_j balanced by the choice of α. Thus, the *Similarity* function (Equation 6) captures our two distinct criteria, since α balances the direction of the bilateral relations between a_i and b_j and duration period w is "embedded" in the construction process of the AB and BA tables (since they are created by the extended triggering and triggered vectors).

3.2 Dataset Representation

Applying a typical clustering algorithm on the table structured defined in previous subsection would yield clusters of elements from only one dataset. Since our problem deals with a simultaneous clustering we need a convenient data structure which, on one hand, will keep the information stored in vectors and tables' and, on the other hand, will enable the datasets co-clustering. A graph is such a convenient structure, since it can represent relations between two sets of elements [1] and it has been already used in co-clustering approaches [2], [7]. In our case, we use a bipartite graph with its vertices and edges indicating the datasets' elements and their similarities respectively. Edges relate data elements on the basis of the direction and duration of their relations, the two criteria of the *Similarity* function.

We assume an undirected bipartite graph $G = (A, B; E)$ where $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3\}$ are two sets of vertices corresponding to the time-related datasets and E is the set of edges $\{\{a_i, b_j\} : a_i \in A, b_j \in B\}$ connecting nodes in A and B as depicted in Figure 2(a). In this bipartite model, the $\{a_i, b_j\}$ edges are undirected since the relation between a_i and b_j is bidirectional and is expressed by the *Similarity* function (Equation 6), while there are no edges between elements in A or between elements in B.

In practice, the *Similarity*(a_i, b_j) serves as an edge weighting function which captures both the direction and duration of impact between a_i and b_j. Given the



(a) The weighted undirected bipartite graph G

(b) Cut of the bipartite graph G

Fig. 2. Data representation and partitioning

similarity values between each pair of related a_i and b_j vertices, we can define the n x m symmetric table *WE* to express the weights of graph's edges:

$$WE(a_i, b_j) = Similarity(a_i, b_j) \tag{7}$$

Example 2. In Figure 2(a), $WE(1,1) = 0.8$ while $WE(1,2) = 0.1$ indicating that the element a_1 is more related with the element b_1 than with the element b_2. □

According to Problem 1 each of the k subsets $C_1, C_2, \ldots, C_k$ includes elements from both datasets such that a_i and b_j will belong to the same C_x, where $x = 1, \ldots, k$, once their $WE(a_i, b_j)$ contributes to the maximization of $\sum_{x=1}^{k} \sum_{a_i, b_j \in C_x} Similarity(a_i, b_j)$. Let us consider the graph G depicted in Figure 2(a) and assume that we want to create $k = 2$ clusters. It is obvious that the 2-partitioning of Figure 2(b) should maximize the sum of similarities between elements of the same clusters (intra-clusters sum) and minimize the sum of similarities between elements of different clusters (inter-clusters sum). The last, inter-clusters sum is expressed by:

$$\sum_{a_i \in C_x} \sum_{b_j \in C_y} WE(a_i, b_j) \tag{8}$$

where $x, y = 1, \ldots, k$ and $x \neq y$, and corresponds to the *cut* of the graph G. Its minimization provides a solution to the graph partitioning problem [4]. Therefore, Problem 1 is transformed into a graph k-partitioning problem.

3.3 The Co-clustering Approach

The k-partitioning of the graph G implies the creation of k groups with elements originating from both A and B datasets which is our co-clustering approach's goal. Moreover, given that the edges of the graph G carry information about the direction and duration of impact between elements (Equation 7), our co-clustering approach considers these two criteria.

Given that our problem has been transformed into a cut minimization problem on a weighted undirected graph, we define structures that will provide meaningful information about the underlying data model and will contribute in the co-clustering of a_i and b_j elements. Since the importance of a graph's vertex is characterized by its degree we define two degree tables, namely the n x n diagonal table D_A and the m x m diagonal table D_B. In the weighted graph G the degree of a vertex is the sum of the edges' weights adjacent to it. Thus, the D_A contains the degree of the a_i elements while the D_B consists of the degrees of the b_j elements. Based on *WE* (Equation 7) we define D_A and D_B as follows:

$$D_A(i,i) = \sum_{j=1}^{m} WE(a_i, b_j), i = 1, \ldots, n$$

$$D_B(j,j) = \sum_{i=1}^{n} WE(a_i, b_j), j = 1, \ldots, m$$

Inspired from spectral clustering approaches in [2] and [7] and given the D_A, D_B and WE tables, we create the n x m two dimensional table WAB:

$$WAB = D_A^{-1/2} WE D_B^{-1/2}$$

The $D_A^{-1/2}, D_B^{-1/2}$ tables originate from the diagonal tables D_A, D_B respectively and are used in order to result in the normalized table WAB.

As it has been proved [2], the k left and right singular vectors of WAB will give us a k-partitioning of a_i and b_j elements respectively. We denote as L_A the n x k table of the left singular vectors and R_B the m x k table of the right singular vectors. In order to perform a simultaneous clustering of a_i and b_j elements, we create the $(n + m)$ x k two dimensional table SV defined as:

$$SV = \begin{bmatrix} D_A^{-1/2} L_A \\ D_B^{-1/2} R_B \end{bmatrix}$$

Then, we obtain the desired k clusters by running a typical clustering algorithm such as k-means [9] or fuzzy c-means [5] on SV.

Algorithm 1. The Co-clustering algorithm.

Input: The A and B datasets containing n and m elements respectively over the t time points, the integers w and k and the factor a, where $\alpha \in [0 \ldots 1]$.

Ouput: A set $C = \{C_1, \ldots, C_k\}$ of k subsets consisting of elements from both A and B such that the sum of inter-clusters similarities defined by (8) is minimized.

```
/*Preprocessing*/
$(VA, VB) = Preprocess(A, B)$
/*Capturing duration of impact via w*/
$(VAE, VBE) = ExtendedVectors(VA, VB, w)$
/*Capturing direction of impact*/
$AB = CosineCoefficient(TriggeringVAE, TriggeredVBE)$
$BA = CosineCoefficient(TriggeringVBE, TriggeredVAE)$
$WE = a * AB + (1 - a) * BA$
/*Co-clustering*/
$(D_A, D_B) = DegreeTables(WE)$
$WAB = D_A^{-1/2} WE D_B^{-1/2}$
$(L_A, R_B) = SingularVectors(WAB)$
$SV = IntegratedTable(D_A, D_B, L_A, R_B)$
$C = FuzzyCmeans(SV, k)$
```

The Algorithm 1 solves the Time-Related Data Co-Clustering problem taking into account the two distinct criteria, namely the direction and duration of datasets' relations. More specifically, during the preprocessing step we build the VA and VB tables according to the Equations 1 and 2. Considering that each row of these tables constitutes a vector, we incorporate time window w in their corresponding extended vectors VAE and VBE. Then, based on these extended vectors we form the AB and BA tables indicating the direction of relations as described in Subsection 3.1. These relations are quantified by the similarities recorded in the table WE (Equation 6). The factor α weights the significance of the bilateral relations. Once the WE is created, we proceed to the co-clustering step. We create the D_A and D_B degree tables and then the WAB on which we apply a singular value decomposition in order to obtain the k left and right singular vectors organized in tables L_A and R_B respectively. We integrate D_A, D_B and L_A, R_B into SV on which we run the fuzzy c-means clustering algorithm. The algorithm finalizes the k clusters which contain elements from both A and B datasets. In fuzzy c-means the data elements can be assigned to all clusters with different probability. This is preferable compared to a hard clustering approach (e.g. k-means), since we are aiming at a flexible clustering scheme.

4 Experimentation

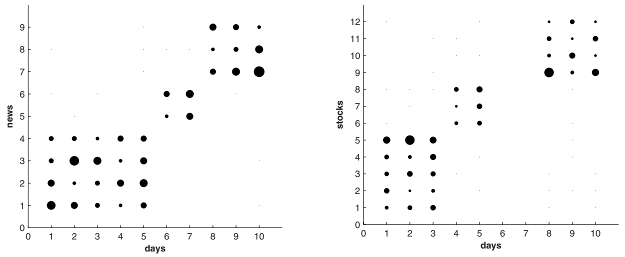
To evaluate the proposed approach we carried out experiments on both synthetic and real datasets. Both types of datasets were needed in order to capture the perspective of the proposed algorithm and its actual behavior. In case of the synthetic data we choose not to experiment with a large-scale dataset in order to be able to drill down on the results [7]. However, our algorithm is scalable and applicable in larger-scale datasets as indicated by the real data experimentation. The algorithm's results are discussed in order to give an insight of its applicability and importance.

4.1 Synthetic Data

We consider the datasets A and B of sizes 9 and 12 respectively over 10 time points. To facilitate the results' discussion we assume that A represents news articles, B corresponds to stock market data and the time period refers to 10 days starting from Monday. Then, the news frequency and stocks' fluctuation, recorded on tables VA and VB respectively, are visualized on Figures 3(a) and 3(b). The size of circles ranges to denote the values of VA and VB tables. Thus, all elements of VA exhibit some value on different days while in VB there are no values during the days 6 and 7 because these days refer to a weekend during which there is no "traffic" in stocks market.

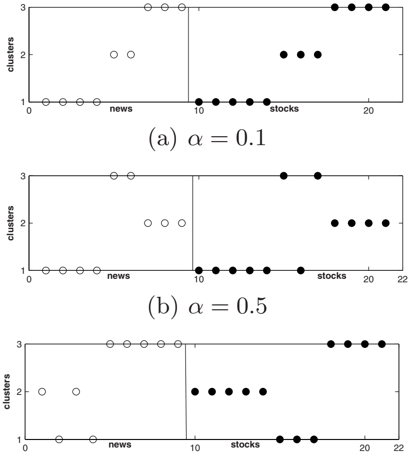
As depicted in Figure 3 we consider $k = 3$ groups of elements in each dataset which are clearly separated over time. In particular, in Figure 3(a) we have the groups $\{a_1, a_2, a_3, a_4\}$, $\{a_5, a_6\}$ and $\{a_7, a_8, a_9\}$ while in Figure 3(b) the elements are arranged as $\{b_1, b_2, b_3, b_4, b_5\}$, $\{b_6, b_7, b_8\}$ and $\{b_9, b_{10}, b_{11}, b_{12}\}$. We experimented tuning the factor α to the values $0.1, 0.5, 0.9$ and fixing the time window to the indicative value $w = 3$. Certainly, in practice, there are events that their impact lasts only one to two days or, to the other extent, whole months but these are the exception.

Due to the lack of space we present a small portion of the results in Figure 4. According to Figures 4(a), 4(b) and 4(c), it is apparent that our co-clustering succeeds in grouping together elements from both datasets. Moreover, we highlight results of different α which show the behavior of our approach in terms of



(a) The 9 x 10 news articles table *VA* (b) The 12 x 10 stock's market data table *VB*

Fig. 3. Synthetic data: the input tables



(a) $\alpha = 0.1$

(b) $\alpha = 0.5$

(c) $\alpha = 0.9$

Fig. 4. Co-clustering over synthetic data for $k = 3$ and $w = 3$

the direction and duration of data interactions. More specifically, from Equation 6 it holds that when $\alpha = 0.1$ our approach considers mainly the impact of stocks fluctuation on news articles. Based on this we can explain the result of Figure 4(a) where, for example, the news $\{a_5, a_6\}$ and stocks $\{b_6, b_7, b_8\}$ belong to the same cluster (C_2) indicating that stocks fluctuation at the end of a week affects news during weekend. For $\alpha = 0.9$ (Figure 4(c)) it seems that a different co-clustering was produced where, for example, the news elements $\{a_5, a_6, a_7, a_8, a_9\}$ were grouped together with stocks $\{b_9, b_{10}, b_{11}, b_{12}\}$ (C_3) denoting that news announcements during weekend have impact on the week stock market's opening. In case that $\alpha = 0.5$ (Figure 4(b)), where there is a mutual interaction between news and stocks, the co-clustering is similar to that of the Figure 4(a) with the exception that the group $\{b_6, b_7, b_8\}$ were split in two clusters. The $\{b_6, b_8\}$ and $\{a_5, a_6\}$ belong to the same cluster (C_3) since the value

$w = 3$ captures the duration of their impact. Specifically, $\{b_6, b_8\}$ occur on Thursday and Friday and influence $\{a_5, a_6\}$ which occur on Saturday and Sunday and vice versa. On the other hand, cluster C_1 consists of $\{b_1, b_2, b_3, b_4, b_5, b_7\}$ and $\{a_1, a_2, a_3, a_4\}$ which all present values during the first 5 days.

4.2 The News and Market Paradigm

Data Workload. The proposed approach has been applied on real news and stock market datasets in order to find groups of related news topics and stocks. Our news dataset was retrieved by the Reuters [2] news archives for a time period of 6 months (June 2004 - November 2004). Codes and especially topic codes are normally used to identify or categorize a news story. Thus, we experimented with the news data based on their topic codes. More specifically, we handled news topics by calculating their daily frequency, since several topic codes may be assigned to a news story, and the idea was to recover the most interesting topics and discard not frequent topics or outliers. The retrieved news data was categorized in 290 topics. However, calculating the daily frequency of each topic for the 6 months i.e. $t = 183$ days we conclude in keeping the 49 most popular topics i.e. $n = 49$ topics (the rest refer to topics with either low i.e. 10 or high i.e. 40000 values of frequency). Thus, the first dataset input to our algorithm is the 49 x 183 news topics' frequency table *VA*.

Our stock-market data was retrieved by the Standard & Poor 500 index (S&P) historical data [3]. The S&P 500 carries information of about 500 companies and their stocks which fall into 119 categories according to the company's operation field (e.g. the *AAPL* stock of the Apple Computer's company belongs to the Computer Hardware category). The retrieved stock market data refers to the same time period i.e. June 2004 - November 2004. The source file (raw data) contains one record per stock and per day. The total number of stocks is $m = 410$ while the 6 month period consists of $t = 183$ days. Given the stock's daily open and close values (no data are given for weekends and holidays since stock markets are closed) we can define its fluctuation which describes the stock's behavior adequately. We calculate absolute values of stocks daily fluctuations since our aim was to find groups of related stocks that are strongly (either positive or negative) affected by news topics and vice versa. Thus, we create the 410 x 183 stocks' fluctuation *VB* table which is our algorithm's second dataset input.

Real Data Experiments. We run the Co-clustering algorithm on the news and stocks datasets tuning the factor α to the values 0.1, 0.5, and 0.9 in order to consider three distinct scenarios where news and stocks are mutually affected, while we initially fixed the time window to $w = 3$. This value of w enables us to observe short term influences which would probably be of the interest of small scale or occasional investors. Furthermore, we experimented with a higher value of time window i.e. $w = 10$ in order to reveal long term interactions which

[2] Reuters: http://www.reuters.com/

[3] S&P500 historical data : http://kumo.swcp.com/stocks/

concern long term traders, financiers or bankers. In our implementation, any value of w in the interval $w = 1, \ldots, t$ can be given, since w is a parameter. However, we select the above values of w, since $t = 183$ and, in practice, the dependencies between news and stock markets data rarely approximate the value of t. Finally, due to the lack of space, results are presented for $k = 4$ and $k = 5$ clusters.

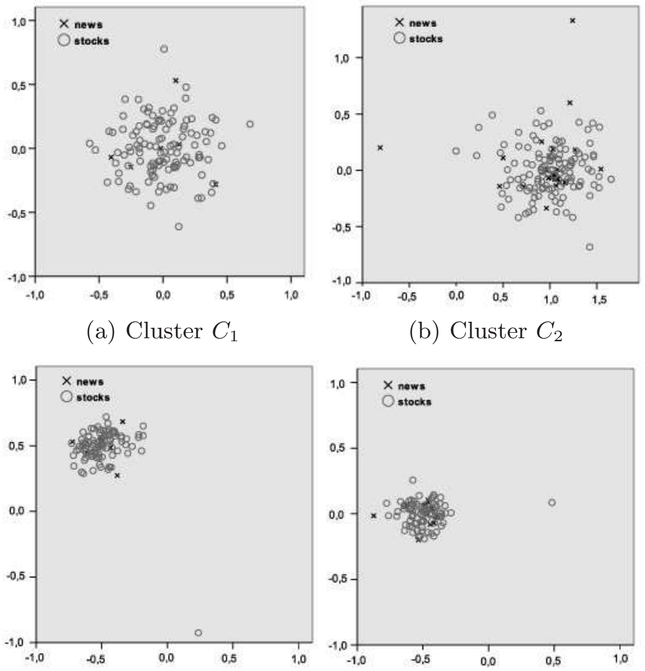
Table 3 presents co-clustering results in terms of the number of members in each cluster for $k = 5$ and for different values of w and α. We observe that the obtained clusters relate elements from both datasets, since there is no cluster consisting solely of news or stocks. Moreover, both news topics and stocks are distributed in a balanced way to the $k = 5$ clusters for $w = 3$ and $\alpha = 0.1, 0.5, 0.9$. Increasing the time window to $w = 10$ the clusters' membership changes, while the clusters remain balanced for the same values of α. Having a greater time window contributes to understanding the duration of impact of news to stocks (and vice versa). This is explained by the fact that an increased (decreased) number of stocks/news topics related to particular news topics/stocks in a cluster shows a long (short) term impact.

Table 3. Cluster members for different values of w and a

	$w = 3$						$w = 10$					
	$a = 0.1$		$a = 0.5$		$a = 0.9$		$a = 0.1$		$a = 0.5$		$a = 0.9$	
	news	stocks	news	stocks	news	stocks	news	stocks	news	stocks	news	stocks
C_1	7	78	22	90	16	95	5	69	9	82	6	90
C_2	11	98	8	70	5	86	4	74	10	93	17	79
C_3	6	97	6	103	6	60	20	91	6	75	8	82
C_4	5	73	6	77	4	89	9	91	9	87	7	70
C_5	20	64	7	70	18	80	11	85	15	73	11	89

Apart from the denoted clusters' balance in terms of their membership it is important to study their quality too. Thus we have proceeded to a correspondence analysis [8] for visualizing the associations between news and stocks in each of the obtained clusters. We indicatively present the algorithm's results for $k = 4$, $w = 3$ and $a = 0.5$ in Figure 5. In each subfigure, which depicts one of the obtained clusters, the "x" marker corresponds to news while the circle marker to stocks. It is apparent that the coclustering algorithm creates groups of closely related news and stocks as indicated by the clusters compactness. Moreover, the obtained clusters are well separated, while there is no overlap between them.

In addition, it would be interesting to attempt a more "conceptual" analysis of the algorithm's results. The algorithm succeeds in grouping news topics and stocks that according to the their "nature" seem to be related. For example, topics that describe news about central banks and interest rates (e.g. CENT and INT) were assigned to the same cluster with stocks referring to regional banks and insurance (e.g. BK, CMA and MI). Similarly, the topic DRU related to pharmaceutical and health care and the stocks that refer to health care distributors, pharmaceutical and health care equipment (e.g. ABC, AET and CAH)



(a) Cluster C_1 (b) Cluster C_2

(c) Cluster C_3 (d) Cluster C_4

Fig. 5. Correspondence analysis results for $k = 4$

were grouped together. In the last example, we observed that fixing time window to $w = 3$ and tuning α to the values $0.1, 0.5, 0.9$ there are specific stocks such as ABC, AET and CAH that remain related to the topic DRU. Independently whether the news "prevail" over stocks ($\alpha = 0.9$) or stocks over news ($\alpha = 0.1$), it seems that there exists a "core" of elements that remain related. Indicatively, for $\alpha = 0.1, 0.5, 0.9$ and $w = 3$ there are $18, 14, 19$ stocks related to the topic DRU respectively. For the corresponding values of α and $w = 10$ the topic DRU was related with $17, 15, 16$ stocks. Some of these stocks are the same with those in case that $w = 3$. Thus, the short term interactions of topic DRU are different compared to the long term ones but there are also interactions which remain stable in spite of the time period.

Studying the above results is useful to parties or individuals of different interests and perspectives, since many people are involved in recommending, trading, publicizing and circulating of both news and market data. For example, the proposed co-clustering would be beneficial for traders who could proceed to recommending certain stock trading acts to their clients, based on same cluster topic codes and stocks for a preferable time interval. Also, news agencies representatives may proceed to certain news topics announcements, based on intense stocks traffic which under co-clustering is shown to influence news.

5 Conclusions

This paper introduces a co-clustering approach which yields clusters consisting of elements belonging to two different but related over time datasets under two main criteria: the direction and duration of data interactions. The Co-Clustering algorithm differentiates data similarity via α and w parameters. The factor α balances the direction of the bilateral relations, while the time window w shifts the duration of data impact. The proposed algorithm has been evaluated using real workloads which correspond to news articles and stock market data. The results showed that our approach succeeds in creating balanced clusters whose membership varies according to α, indicating the news that affect specific stocks and vice versa, as well as according to w revealing different relations with reference to short and long term periods of interactions. Understanding the resulted clusters would facilitate acts of investors, traders, bankers, journalists, news agencies etc.

References

1. Afrati, F., Das, G., Gionis, A., Mannila, H., Mielikainen, T., Tsaparas, P.: Mining Chains of Relations. In: Proc. of the 5th IEEE Int. Conf. on Data Mining, ICDM, pp. 553–556 (2005)
2. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Proc. of the 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data mining, KDD, pp. 269–274 (2001)
3. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-Theoretic Co-clustering. In: Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD, pp. 89–98 (2003)
4. Ding, C., He, X., Zha, H., Gu, M., Simon, H.: A Min-max Cut Algorithm for Graph Partitioning and Data Clustering, pp. 107–114 (2001)
5. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics 3, 32–57 (1973)
6. Fung, G., Xu Yu, J., Lam, W.: News Sensitive Stock Trend Prediction. In: Proc. of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, PAKDD, pp. 481–493 (2002)
7. Gao, B., Liu, T., Zheng, X., Cheng, Q., Ma, W.: Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In: Proc. of the 11th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data mining, KDD, pp. 41–50 (2005)
8. Greenacre, M.J.: Correspondence Analysis in Practice. Academic Press, London (1993)
9. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Heidelberg (2001)
10. Mirasgedisa, S., Sarafidis, Y., Georgopoulou, E., Lalas, D.P., Moschovits, M., Karagiannis, F., Papakonstantinou, D.: Models for mid-term electricity demand forecasting incorporating weather influences. Energy 31, 208–227 (2006)
11. Peramunetilleke, D., Wong, R.: Currency exchange rate forecasting from news headlines. In: Proc. of the 13th Australasian database conference, ADC, vol. 24, pp. 131–139 (2002)

12. Sagar, V.K., Kiat, L.C.: A neural stock price predictor using qualitative and quantitative data. In: Proc. of 6th Int. Conf. on Neural Information Processing, ICONIP, vol. 2, pp. 831–835 (1999)
13. Wuthrich, B., Cho, V., Leung, S., Permunetilleke, D., Sankaran, K., Zhang, J., Lam, W.: Daily Stock Market Forecast from Textual Web Data. In: Proc. of IEEE Int. Conf. On System, Man and Cybernetics, vol. 3, pp. 2720–2725 (1998)
14. Zhang, J., Korfhage, R.: A Distance and Angle Similarity Measure Method. Journal of the American Society for Information Science 50, 772–778 (1999)