Trend Mining in Social Networks: From Trend Identification to Visualisation

Puteri N.E. Nohuddin[1], Wataru Sunayama[2], Rob Christley[3], Frans Coenen[1] and Christian Setzkorn[3]

(1) Department of Computer Science, University of Liverpool, Liverpool, UK
(2) Graduate School of Information Sciences, Hiroshima City University, Japan
(3) Department of Epidemiology & Population Health, University of Liverpool and National Centre for Zoonosis Research, Leahurst, Neston, UK

Abstract: *A four stage social network trend mining framework, the IGCV (Identification, Grouping, Clustering and Visualisation) framework, is described. The framework extracts trends from social network data and then applies a sequence of techniques ("tools") to this data to facilitate interpretation of the identified trends. Of particular note is the visualisation of trend migrations (changes) that feature within time stamped network data. The framework is illustrated using a sequence of four social networks extracted from the Cattle Tracing System (CTS) in operation in Great Britain, although it could equally well be applied to other forms of temporal data. The presented analysis of the IGCV framework indicates advantages, with respect to network trend mining, that can be gained; especially when the framework is applied to large real-world datasets.*

Keywords: Trend Mining, Social Network Data, Visualisation

1. Introduction

The identification of trends has been an important activity in many application domains such as business intelligence, demography and epidemiology. Trend mining is concerned with the application of data mining techniques to extract trends from time stamped data collections (Kohavi *et al*., 2002; Lent *et al*., 1997). The work described in this paper is directed at trend mining within the context of social networks. Social networks are communities of interacting entities. Well known examples include web-based applications such as Facebook, Bebo and Flickr. However, other examples include business communities, file sharing systems and co-authoring frameworks. Social network mining is typically directed at identifying patterns and sub-communities (clusters) within the network data (Safaei *et al*., 2009; Xu *et al*., 2008). The mining of social networks is usually conducted in the static context whereby data mining techniques are applied to a "snap shot" of the network of interest. Little work has been directed at applying data mining techniques to social network data in the dynamic context so as to discover, for example, trends in the network data. The problem domain, which is the focus of the work described is this paper, is therefore the identification of trends in dynamic social networks. We define trends in social networks in terms of the fluctuations of traffic between nodes, or groups of nodes, in such networks. The main issues associated with this form of trend mining, when applied to social network data, are: (i) the large amount of data that has to be processed, social network datasets tend to be substantial; and (ii) trend mining techniques typically generate large numbers of trends which are consequently difficult to analyse.

To address these two issues we present an end-to-end social network trend mining framework that takes as input a time stamped data set, describing the activity in a specific social network; and, as an end result, provides a visualisation of the most significant trends. The process is predicated on the assumption that end users are interested in the progress of trends, thus the manner in which trends change over time (*migrate*) or remain unchanged. We refer to this framework as the IGCV (Identification, Grouping, Clustering and Visualisation) framework. IGCV comprises four stages:

1. **Trend Identification:** The application of frequent item set mining techniques to define and identify trends within social network data.
2. **Trend Grouping:** The grouping, using a Self Organising Map (SOM) approach, of the large number of trends that are typically identified.
3. **Trend Clustering:** Identification of "communities" of trend migrations, within the SOM groupings, using a hierarchical clustering mechanism based on the Newman method.
4. **Trend Visualisation:** Visualisation of the trend migrations using a *spring model* to display, what are considered to be the most significant, trend migrations.

The IGCV process is illustrated in Figure 1. Each of the four stages making up the framework is considered in further detail later in Sections 4, 5, 6 and 7 respectively.

To illustrate, and evaluate, the above process we have used a social network extracted from the Cattle Tracing System (CTS) in operation in Great Britain. CTS includes a database that records cattle movements throughout Great Britain. By considering the *holding areas* (farms, markets, abattoirs, etc.) recorded in the CTS database as nodes, and the cattle movement between holding areas as the traffic (links) between nodes, a large scale social network may be derived. The derivation of this social network is discussed in further detail in Section 8 together with a discussion and evaluation of the operation of the IGCV framework with respect to this network.

2. Related Work

Trend mining has becoming a popular approach for the study of time series data so as to identify changes and relationships within the temporal patterns contained in the data. There are many examples of trend identification applications and tools in the literature. For example, Streibel (2008) used quantitative numeric financial data, and qualitative text corpi data extracted from business news articles, to forecast financial market trends. Google provides Google Trends[1], a public web facility that supports the identification of trends associated with keyword search volume. Raza and Liyanage (2008) introduced a trend analysis approach to mine and monitor data for abnormalities and faults in industrial production processes. Somaraki *et al*. (2010) describe an application of trend mining in the field of diabetic retinopathy.

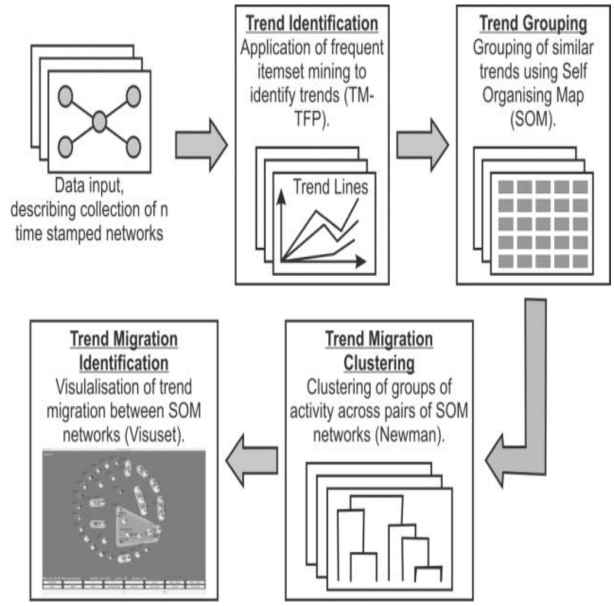

Figure 1: Schematic for The IGCV Framework

In the work described in this paper, we define trends in terms of the changing frequency of temporal patterns found in social network data which has been normalised into a set of binary valued attributes. Frequent patterns are sets of attributes that "frequently" co-occur within data according to some user specified *frequency threshold* (Agrawal *et al*., 1993). Several researchers have proposed techniques for the mining of patterns in temporal data mining; this work includes the identification of sequential patterns (Agrawal and Srikant, 1995), frequent episodes (Mannila et al., 1997), emerging patterns (Dong and Li, 1999) and jumping and emerging patterns (Khan et al., 2010). There are also many established frequent pattern mining techniques; one of these, TFP, has

[1] http://www.google.com/intl/en/trends/about.html

been extended with respect to the IGCV framework, so as to permit the identification of temporal frequent pattern trends.

A social network is a representation of the link structure described by some social entity, and normally comprises nodes (*actors*) connected by one of more links (Wasserman and Faust, 2006). To analyze this structure, techniques have been proposed which map and measure the relationships and flows between nodes. In general social network mining can be applied in a static context, which ignores the temporal aspects of the network; or in a dynamic context, which takes temporal aspects into consideration. In the static context, we typically wish to: (i) find patterns that exist across the network, (ii) cluster (group) subsets of the networks, or (iii) build classifiers to categorize nodes and links. In the dynamic context, we typically wish to identify relationships between the nodes in the network by evaluating the spatio-temporal co-occurrences of events (Lauw *et al*., 2005). The latter is thus the focus of the work described in this paper. There has been some related work, to that described in this paper, on social networks trend analysis. For example, Gloor *et al*. (2008) introduced a trend analysis algorithm to generate trends from Web resources. The algorithm calculated the values of temporal *betweeness* of online social network node and link structures to observe and predict trends concerning the popularity of concepts and topics such as brands, movies and politicians. There has been some work on the identification of trends in social networks in the context of online viral marketing (Richardson and Domingos, 2002) and stock market activities (Choudhury *et al*., 2008). Nevertheless, these systems tended to be directed at the online social network domain and generated trends in the static context. Conversely, the IGCV framework generates frequent patterns from unusual tabular social network data and collects trends from a sequence of time periods to identify dynamic changes in the data.

The IGCV framework provides for the visualisation of trend changes in social network data using Visuset software (Nishikido et al., 2009) specifically developed for this purpose. A brief review of some related work on network visualisation is therefore also presented in this section. Kandogan (2001) developed a system to display multi-dimensional data on a two dimensional surface as a scatter plot. However, no indication is given of the inter relationships between data points. Visuset groups data into "islands", data within an island is closely linked according to co-relationship values. Visuset thus highlights the nature of the groupings that exist and how the data is correlated. Havre *et al*. (2002) described a technique for displaying thematic changes as *river flows*, so that changes of topics can be observed. However, unlike Visuset, the relationships between topics are not considered. Chen (2006) described a system to visualize a network so as to identify *emerging trends*. However, the network is displayed with respect to a specific time stamp, therefore changes in trends cannot be easily observed. Visuset displays trend transitions as an animation so as to demonstrate how trends change over a given period. Robertson *et al*. (2008) introduced a system to also show trends by animation. This method illustrated changes in the data in the form of *traces*, but changes are considered independently. In Visuset trends are correlated against one another so that observers can see how groups of trends change with time.

3. Formalism and Definition

The input to IGCV comprises a sequence of n time stamped data sets, $D = \{d_1, d_2, ..., d_n\}$. Each data set comprises a binary valued table such that each record represents the traffic between a node pair in the social network of interest. The level of detail provided may vary between applications, nodes may be described in terms of a single attribute or a number of attributes. For example nodes may include information about the entity they represent, such as geographical location (for example post code, or easting and northing) and the nature of the attribute. In the case of the CTS application, described in more detail in Sub-section 8.1, a number of node categories are identified (farms, markets, abattoirs, etc.). The quantity of traffic is defined in terms of a sequence of ranges. Additional traffic information may also be provided, for example in the case of the CTS application information concerning the nature of the cattle moved is included (breed type, gender, etc.). Thus, each record, in each dataset d_1 to d_n, comprises a subset of a global set of binary valued attributes $A = \{a_1, a_2, ..., a_m\}$. Note that the number of records in each dataset need not be constant across the collection.

A pattern trend t is then defined in terms of the frequency of occurrence, over time, of the patterns within the input data. The trends are conceptualised as *trend lines*, one per pattern, representing a mapping of frequency of occurrence against time.

To identify changes in trends (or lack of them) the number of time stamps is subdivided in e *episodes*[2], each of equal length m, thus $n=e \times m$. The size of m, and hence the number of episodes e, will be application dependent. However, with respect to the CTS application a granularity of one month was used and hence m was set at 12; consequently each episode represented a year (four experimental purposes CTS data for four episodes was obtained: 2003, 2004, 2005 and 2006). Thus, a trend t comprises a set of values $\{v_1, v_2, ..., v_n\}$ where each value represents an occurrence count. The collection of trends, T, that we wish to analyse therefore comprises a sequence of sub-collections $\{T_1, T_2, ..., T_e\}$ (where e is the number of episodes).

4. Trend Identification

As noted above, a trend is defined in terms of a sequence of occurrence counts for a given pattern in the input data. The patterns in this context are frequent item sets as popularised in association rule mining (Agrawal and Srikant, 1994). More specific parallels can also be drawn with temporal association rule mining (Harms and Deogun, 2004; Mannila *et al*., 1997). To mine pattern trends an extended version of the TFP (Total From Partial) algorithm (Coenen *et al*., 2001; Coenen *et al*., 2004) was used. TFP is an established frequent pattern mining algorithm distinguished by its use of two data structures: (i) a P-tree used to both encapsulate the input data and record a *partial* frequency count for each pattern, and (ii) a T-tree to store the identified patterns together with their *total* frequency counts. The T-tree is essentially a reverse *set enumeration tree* that allows fast *look up*. TFP follows an *apriori* style of operation to generate frequent items sets where by the *antimonotone property* of item sets is used to limit the search space. The well documented *support framework* is used, whereby a frequency count threshold (the *support threshold*) defines "interesting" patterns; typically the lower the support threshold the more patterns that are discovered.

The TFP algorithm, in its original form, was not designed to address the temporal aspect of frequent pattern mining. For the purpose of the IGCV framework the TFP algorithm was therefore extended so that sequences of datasets could be processed, and the discovered frequent patterns stored, in a way that would allow for differentiation between individual time stamps and episodes. The resulting algorithm was called TM-TFP (Trend Mining TFP) which incorporated a TM-T-tree to store the desired patterns. Further details of the TM-TFP algorithm can be found in (Nohuddin *et al*., 2010a) and (Nohuddin *et al*., 2010b). The output from the TM-TFP algorithm is thus the collection of trends $T = \{T_1, T_2, ..., T_e\}$. Experiments using a variety of network datasets (reported in (Nohuddin *et al*., 2011)) have indicated that a large number of trends are often identified. Of course, the number of patterns to be considered can be reduced by using a higher support threshold, but the established argument against this expedient is that potential interesting patterns may be missed. In the case of the CTS network, Table 1 presents the number of patterns discovered

[2] Some authors use the term *epoch*.

using three different support thresholds (the first column gives the episode identifier). The large number of discovered trends was one of the main motivations for the IGCV framework, which incorporates a number of mechanisms to support the analysis of the discovered trends. These analysis mechanisms are discussed further in the following sections.

Table 1: *Number of trends identified using TM-TFP for a sequence of four CTS network episodes and a range of support thresholds*.

Episode (year)	Support Threshold		
	0.5%	**0.8%**	**1.0%**
2003	63,117	34,858	25,738
2004	66,870	36,489	27,055
2005	65,154	35,626	25,954
2006	62,713	33,795	24,740

5. Trend Grouping

As noted in the previous section, a large number of trends are typically identified using TM-TFP. One mechanism, to support the desired trend analysis, incorporated into the IGCV framework was to group the discovered trends according to their distinguishing features. The intuition here was that end users were expected to be interested in particular types of trends, for example increasing or decreasing trends. To perform the grouping Self Organising Map (SOM) technology was adopted.

SOMs, as first proposed by Kohonen (1995), provide a useful unsupervised technique whereby data can be grouped into a predefined $i \times j$ grid so as to aid the interpretation of the data. SOMs have been utilised with respect to many applications, examples include: Geographic Information Systems (GIS) (Agrawal and Skupin, 2008), the exploration of document collections (Kohenen, 1997) and trajectory analysis (Schreck *et al.*, 2009). SOMs may be viewed as a type of feed-forward, back propagation, neural network that comprises an input layer and an output layer (the $i \times j$ grid). Each output node is connected to every input node. The SOM is "trained" using a training set. Each record in the training set is presented to the SOM in turn and the output nodes compete for each record. Once a record has been assigned to the "winning" node the network's weightings are adjusted to reflect the new position. At first the adjustments are relatively large, but as the training continues the adjustments become smaller. A feature of the adjustment is that adjacent nodes hold similar records, the greatest dissimilarity is between nodes at opposite corners of the grid.

In the case of the CTS network the authors experimented with different mechanisms for training the SOM, including: (i) devising specific trends to be represented by individual nodes, (ii) generating a collection of all the mathematically possible trends and training the SOM using this set, and (iii) using some or all of the trends in the first epoch to be considered. The first required prior knowledge of the trend configurations of interest; which, it was conjectured, tended to defeat the objective of the trend mining process. The second mechanism, it was discovered, resulted in maps for which the majority of nodes were empty. The third option was therefore adopted; the SOM was trained using the trend lines associated with one of the episodes. The resulting *proto-type map* was then populated with data from the remaining *e-1* episodes, to produce a sequence of *e* maps $M = \{M_1, M_2, ..., M_e\}$.

SOMs are often described as a visualisation technique. However, given a large and/or complex dataset, the number of items within each group (map node) may still be large. This was found to be the case with respect to the CTS application. One obvious solution is to increase the size of the grid, however this may result in an undesirable computational overhead and in many cases does not serve to resolve the situation as many of the map *nodes* remain empty (i.e. the items are consistently held in a small number of map nodes such that increasing the size of *i* and *j* has little or no effect). In the case of the CTS network a 10×10 node SOM was found to be the most effective as this gave a good decomposition while still ensuring computational tractability.

6. Trend Migration Clustering

The next stage in the IGCV process provides for further analysis of the trend data contained in the generated SOMs (one per episode). The motivation here was that, at least in the context of the CTS network, consultation with end users indicated that it would be of interest to know how particular trends (i.e. trends associated with a specific pattern) *migrated* across the collection of SOMs from a SOM (map) M_{e_k} to a SOM $M_{e_{k+1}}$ (where e_k and e_{k+1} are "episode stamps"). For this purpose, pairs of SOMs were viewed in terms of a second network containing potentially $i \times j$ nodes and $(i \times j)^2$ links (including "self links"). The nodes in this second network represent groupings of trends that display similar characteristics, as identified using the SOM analysis technique described above; nodes were labelled with the number of trends at the node in map M_{e_k} (i.e. the "from" map). The links then represented the migration of trends from M_{e_k} to $M_{e_{k+1}}$ and were labelled with the number of migrating trends (i.e. a "traffic" value). The process of visualising such networks is discussed in the following section. It was also considered desirable to display "communities" within these networks, i.e. clusters of nodes which were "strongly" connected. A hierarchical clustering mechanism, founded on the Newman method (Newman, 2004) for identifying clusters in network data, was applied. Newman proceeds in the standard iterative manner on which hierarchical clustering algorithms are founded. The process starts with a number of clusters equivalent to the number of nodes. The two clusters (nodes) with the greatest "similarity" are then combined to form a merged cluster. The process continues until a "best" cluster configuration is arrived at or all nodes are merged into a single cluster. The overall process is typically conceptualised in the form of a *dendrogram*. Best similarity is defined in terms of the *Q-value*, this is a "modularity" value which is calculated as follows:

$$Q_i = \sum_{i=1}^{i=n}(c_{ii} - a_i^2) \qquad (1)$$

where Q_i is the Q-value associated with the *current* cluster i, n is the total number of nodes in the network, c_{ii} is the fraction of intra-cluster (within cluster) links in cluster i over the total number of links in the network, and a_i^2 is the fraction of links that end in the nodes in cluster i if the edges were attached at random. The value a_i is calculated as follows:

$$a_i = \sum_{i=1}^{i=n} c_{ij} \qquad (2)$$

where c_{ij} is the fraction of inter-cluster links, between the current cluster i and the cluster j, over the total number of links in the network.

Thus, at each iteration, the Q-values for all possible cluster pairings are calculated and the pairing with the highest Q-value selected for merging. The process proceeds until a best cluster configuration is achieved. This is defined as the configuration with the highest overall Q-value. Generally speaking, if the Q-value is above 0.3 then communities can be said to exist within the target network; the value of 0.3 was derived experimentally by Newman and Girvan (2004). Note that if all nodes are placed in one group the Q-value will be 0.0 (i.e. a very poor clustering).

6.1 Worked Example of Hierarchical Clustering Using Newman

Considering the example network presented in Figure 2, the Q value for this network at the start of the process, when each vertex is considered to represent a group, is (using data from Table 2):

Q = -0.01- 0.01- 0.04 - 0.01 = -0.07

Table 2: Start Condition

i	c_{ii}	a_i	a_i^2	Q
A	0	0.1	0.01	-0.01
B	0	0.1	0.01	-0.01
C	0	0.2	0.04	-0.04
D	0	0.1	0.01	-0.01

We then have six potential joins *AB*, *AC*, *AD*, *BC*, *BD* and *CD*; giving rise to six potential configurations. Calculating the Q-value for each configuration (Table 3) gives a best Q-value of 0.04, this therefore represents the first join and we have the configuration {*AB, C, D*}.

For the next join, there are three possible configurations: {*ABC, D*}, {*ABD, C*} and {*AB, CD*}. Calculating the Q-value for each of these configurations (Table 3) gives a best Q-value of 0.28, so this is the second join and we have the configuration {*AB, CD*}.

For the third iteration, we combine all the vertices and get a Q-value of 0.0. The discovered maximal value for Q is then 0.28 and hence the configuration associated with this value, {*AB, CD*}, is selected as the best grouping (clustering). The dendrogram for the example is given in Figure 3. The identified clustering (communities) are then displayed as "islands" in the following stage in the IGCV framework. This will be described in the following section.

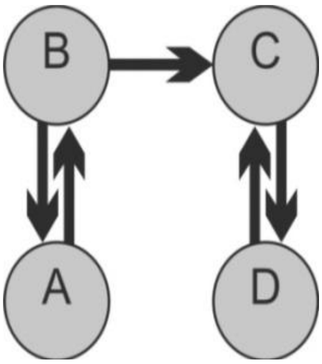

Figure 2: Four Node Example Network

Figure 3: Dendrogram for Hierarchical Clustering Example (Note: the heights of the dendrogram "branches" are not significant)

7. Trend Visualisation and Animation using Visuset

IGCV provides two forms of visualisation which are integrated into a single software system called Visuset:

1. Visualisation of trend migration between two successive SOMs.
2. Animation of the trend migration between three successive SOMs.

In each case the visualisation (animation) includes the trend migration communities discovered, using Newman, as described above. The communities are depicted as "islands" demarcated by a "shoreline" (for aesthetic purposes the islands are also contoured, although no meaning should be attached to these contours). The visualisation process is described in Sub-section 7.1, and the animation in Sub-section 7.2, below.

Table 3: First Iteration

Groups			Internal Links									
1	2	3	c_{11}	c_{22}	c_{33}	a_1	a_2	a_3	a_1^2	a_2^2	a_3^2	Q
AB	C	D	0.4	0	0	0.4	0.4	0.2	0.16	0.16	0.04	0.04
AC	B	D	0	0	0	0.6	0.2	0.2	0.36	0.04	0.04	-0.44
AD	B	C	0	0	0	0.4	0.2	0.4	0.16	0.04	0.16	-0.36
BC	A	D	0	0.2	0	0.6	0.2	0.2	0.36	0.04	0.04	-0.24
BD	A	C	0	0	0	0.4	0.2	0.4	0.16	0.04	0.16	-0.36
CD	A	B	0	0.4	0	0.6	0.2	0.2	0.36	0.04	0.04	-0.04

Table 4: Second Iteration

Groups		Internal Links						
1	2	C_{11}	C_{22}	a_1	a_2	a_1^2	a_2^2	Q
ABC	D	0.6	0	0.8	0.2	0.64	0.04	-0.08
ABD	C	0.4	0	0.6	0.4	0.36	0.16	-0.12
AB	CD	0.4	0.4	0.4	0.6	0.16	0.36	0.28

7.1 Visualisation of Trend Migration

For the visualisation, Visuset locates nodes in a 2-D "drawing area" using the *Spring Model* (Sugiyama and Misue, 1995). The spring model for drawing graphs in 2-D space is designed to locate nodes in the space in a manner that is both aesthetically pleasing and limits the number of edges that cross over one another. The graph to be depicted is conceptualised in terms of a physical system where the edges represent springs and the nodes inanimate objects connected by springs. Nodes connected by "strong springs" therefore attract one another while nodes connected by "weak springs" repulse one another. The graphs are drawn following an iterative process. Nodes are initially located within the 2D space using some set of (random) default locations (usually defined in terms of an x and y coordinate system) and, as the process proceeds, pairs of nodes connected by strong springs are "pulled" together. In the context of IGCV the spring value was defined in terms of a *correlation coefficient* (C):

$$C_{ij} = \frac{X}{\sqrt{(|M_{e_k i}| \times |M_{e_{k+1} j}|)}} \qquad (3)$$

where C_{ij} is the correlation coefficient between a node i in SOM M_{e_k} and a node j in SOM $M_{e_{k+1}}$ (note that i and j can represent the same node but in two different maps), X is the number of trends that have moved from node i to j and $|M_{e_k i}|$ ($|M_{e_{k+1} j}|$) is the number of trends at node i (j) in SOM $M_{e_k i}$ ($M_{e_{k+1} j}$). A migration is considered "interesting", and thus highlighted by Visuset, if C is above a specified minimum relationship threshold (Min-Rel). With respect to the CTS network we have discovered that a threshold of 0.2 is a good working Min-Rel value; although Visuset does allow users to specify, and experiment with, whatever Min-Rel value they like. The Min-Rel value is also used to prune links and nodes; any link whose C-value is below the Min-Rel value is not depicted in the visualisation, similarly any node that has no links with a C-value above Min-Rel is not depicted.

The Visuset spring model algorithm (a simplified version) proceeds as follows:
Set drawing area size constants, *SIZEX* and *SIZEY*.

1. For all pair of nodes, allocate an *ideal distance*, $IDIST_{ij}$, where i and j are node numbers. In the current implementation: if a pair has a link, the distance is set as 200 pixels; otherwise it is set to 500 pixels.
2. Set initial coordinates for all nodes. All nodes are "queued" in sequence, according to their node number, from the top-left of the drawing area to the bottom-right.
3. For all node pairs determine the actual pixel distance $RDIST_{ij}$ (where i and j are node numbers).
4. For all nodes, recalculate the coordinates using equations 4 and 5 where: $node_{i_x}$ ($node_{i_y}$) is the x (y) coordinate of $Node_i$, n is the number of nodes to be depicted, K is the *spring constant*, and dx_{ij} (dy_{ij}) is the absolute value of $node_{i_x}$ - $node_{j_x}$ ($node_{i_y}$ - $node_{j_y}$).
5. If dx_{ij} + dy_{ij} is below a specified threshold (in terms of a number of pixels), or if some maximal number of iterations is reached, exit.
6. Go to Step 4.

$$node_{i_x} = node_{i_x} + \sum_{j=1}^{j=n} (dx_{ij} \times K \times (1 - \frac{IDIST_{ij}}{RDIST_{ij}})) \qquad (4)$$

$$node_{i_y} = node_{i_y} + \sum_{j=1}^{j=n} (dy_{ij} \times K \times (1 - \frac{IDIST_{ij}}{RDIST_{ij}})) \qquad (5)$$

For the current version of Visuset *SIZEX = 1280 pixels* and *SIZEY = 880 pixels*, and the spring constant was set to 0.2. It should also be noted that the selected values for the ideal distances, spring constant K, are related to the values chosen for *SIZEX* and *SIZEY* and the number of nodes and links in the system to be visualised. The stopping threshold can be set at any value, but from experimentation we have found that the number of nodes (as a pixel value) provides good operational results. Using Visuset it is also possible to disable the spring model so that the user can manually position nodes (and, if applicable, also change the size of individual islands at the same time). Further details concerning the background and development of Visuset can be found in (Nishikido et al., 2009).

In the current implementation of Visuset nodes are depicted as: single nodes (i.e. self links where the "migration" is from and to the same node), node pairs linked by an edge, chains of nodes linked by a sequence of edges, or more complex sub-graphs (islands). The size (diameter) of the nodes indicates the number of elements represented by that node in M_{e_k} (the size of nodes at $M_{e_{k+1}}$ could equally well have been used, or some interpolation between M_{e_k} and $M_{e_{k+1}}$).

7.2 Animation of Trend Migration

The animation mechanism, provided by Visuset, can be applied to pairs of visualisations (as described above) to illustrate the migration of trends over three episodes (SOMs). We refer to each visualisation as a mapping of the nodes in a SOM M_{e_i} to a SOM M_{e_j}. At the start of an animation the display will be identical to the first visualisation (Map 1) and will move to a configuration similar to the second visualisation (Map 2), although nodes will not necessarily be in the same display location. Thus the animations show how subsequent mappings change and consequently how the trend "communities" change. As the animation progresses the correlation coefficient (C-values) are linearly incremented or decremented from the value for the first map to that of the second map. Thus, as the animation progresses, the links, nature of the islands, and overall number of nodes will change. For example if the correlation coefficient for a node in Map 1 is 0.3 and in Map 2 is 0.1 (assuming a threshold of 0.2) the node will "disappear" half way through the animation. Alternatively, if the correlation coefficient for a node in Map 1 is 0.1 and in Map 2 is 0.5 (again assuming a threshold of 0.2) the node will "appear" a quarter of the way through the animation. Nodes that disappear and appear are highlighted in white and pink respectively (nodes that persist are coloured yellow).

7.3 Worked Example of C-value Calculation

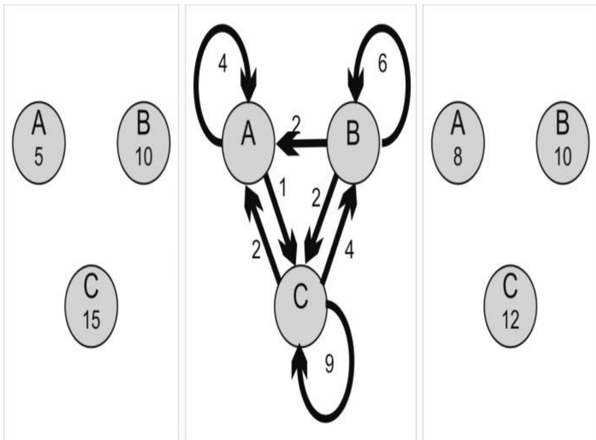

Figure 4: Three Node Example network showing Trend Migrations from *T1* to *T2*

Figure 4 shows the migration of trends through a three node network. The left hand network shows the state at time one (T1) and the right hand network at time two (T2). The nodes in each case are labelled with the number of trends held at the node at these times. The middle network (in Figure 4) shows the number of trends that have migrated to and from the nodes in the network from time T1 to time T2. Table 5 summarises this migration. The calculation of the C-values (correlation coefficients) for this network is given in Table 6. If we use a Min-Rel threshold of 0.2 (as advocated by our experiments) five of the migrations remain, as illustrated in Figure 5 (in the figure the arcs are labeled with the relevant C-values).

Table 5: Trend Migration Summary for Example Network Given in Figure 4

T2 Node ID	T1 Node ID			Total
	1	2	3	
1	4	2	2	8
2	0	6	4	10
3	1	2	9	12
Total	5	10	15	30

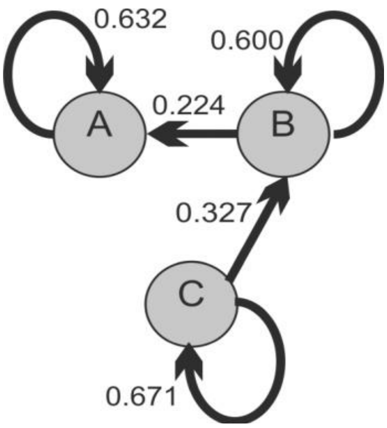

Figure 5: Three Node Example Network with Irrelevant links removed

Table 6: C-Value calculation for Example Network given in Figure 4

T2 Node ID	T Node ID	Trends at T1 (P)	Trends at T2 (Q)	Trends Moved (X)	$P \times Q$	$\sqrt{P \times Q}$	X $\div$ $\sqrt{P \times Q}$
1	1	5	8	4	40	6.32456	0.63246
1	2	5	10	0	50	7.07107	0
1	3	5	12	1	60	7.74597	0.1291
2	1	10	8	2	80	8.94427	0.22361
2	2	10	10	6	100	10.00000	0.60000
2	3	10	12	2	120	10.95445	0.18257
3	1	15	8	2	120	10.95445	0.18257
3	2	15	10	4	150	12.24745	0.3266
3	3	15	12	9	180	13.41641	0.67082

8. Demonstration

Although the ICGV framework can be applied to social network data in general this section will demonstrate the operation of IGCV using the CTS network introduced earlier. Some further detail concerning the CTS network is first presented in Sub-section 8.1. Then, in the following sections, the operation of IGCV is illustrated in terms of its four component stages as described in the foregoing.

8.1 Cattle Movement Database

The Cattle Tracing System (CTS) in operation in Great Britain records all the movements of cattle registered within or imported into Great Britain. The database is maintained by the Department for Environment, Food and Rural Affairs (DEFRA). Cattle movements can be "one-off" movements to final destinations, or movements between intermediate locations. Movement types include: (i) cattle imports, (ii) movements between locations, (iii) movements in terms of births and (iv) movements in terms of deaths. The CTS was introduced in September 1998, and updated in 2001 to support disease control activities. Currently the CTS database holds some 155 Gbytes of data.

The CTS database comprises a number of tables, the most significant of which are the animal, location and movement tables. For the demonstration reported in this section the data from 2003 to 2006 was extracted to make up 4 episodes (2003, 2004, 2005 and 2006) each comprising 12 (one month) time stamps. The data was stored in a single data warehouse such that each record represented a single cattle movement instance associated with a particular year (episode) and month (time stamp). The number of CTS records represented in each data episode was about 400,000. Each record in the warehouse comprised: (i) a time stamp (month and year), (ii) the number of cattle moved, (iii) the breed, (iv) the sender's location in terms of easting and northing grid values, (v) the “type” of the sender's location, (vi) the receiver's location in terms of easting and northing grid values, and (vii) the “type” of the receiver's location. If two different breeds of cattle were moved at the same time from the same sender location to the same receiver location this would generate two records in the warehouse. The maximum number of cattle moved (link value) between any pair of locations for a single time stamp was approximately 40 animals. Sender location eastings and northings were grouped into grid squares measuring 100km per location area. The sequence of cattle movement networks extracted from the CTS data thus comprised, on average, some 150,000 nodes and 300,000 links per network.

8.2 Cattle Movement Trend Mining

IGCV commences with the identification of trends using the TM-TFP algorithm. For experimental purposes three support threshold values of 0.5%, 0.8% and 1% were used. Some examples of the nature of the frequent patterns discovered, in the context of the CTS social network, are presented in Table 7. Using a support threshold of 0.5%, the number of identified trends discovered over the four episodes (2003, 2004, 2005 and 2006) were 63117, 66870, 65154 and 62713. For example: node 34 describes trends where the number of cattle movements increases slightly in March, June and October; nodes 44 and 54 both describe trends where the number of cattle movements is considerably higher in spring and autumn; and so on.

The analysis of the prototype map indicates, as might be expected, that hierarchies of patterns, comprising collections of sub-sets of a “parent” pattern, tend to appear in the same clusters. Recall also that the proximity between SOM nodes indicates the similarity between them; the greatest dissimilarity is thus between nodes at opposite ends of the diagonals. Once the initial prototype map had been generated a sequence of trend line maps was produced, one for each episode. Figure 7 gives the map for the 2003 trend lines. Note that in Figures 7 and 8 each node has been annotated with the number of trends in the “cluster”, and that nodes with “darker” trend lines indicate a greater number of lines within that cluster.

Table 7: Example trend patterns obtained from the 2003 CTS data episode

Pattern	Trends
{2 year old ≤ Animal Age ≤ 5 year old, Breed = Friesian, Breed Type = dairy, Receiver Location Type = Slaughter House (Red Meat)}	{2765, 2211, 2562, 3279, 0, 1307, 2004, 1906, 2593, 3315, 3391, 3152}
{Gender = female, 2 year old ≤ Animal Age ≤ 5 year old, Breed = Friesian, Breed Type = dairy, Receiver Location Type = Slaughter House (Red Meat)}	{2741, 2193, 2541, 3251, 0, 1295, 1995, 1896, 2581, 3299, 3384, 3145}
{Gender = female, Breed = Simmental Cross, Breed Type = beef and dairy, Receiver Location Type = Slaughter House (Red Meat)}	{4050, 3322, 3175, 3690, 2777, 2722, 2972, 2494, 3082, 3823, 3951, 3717}
{Breed Type = beef, Sender Area = 13, easting (200001-300000) and northing (100001-200000), Receiver Location Type = Slaughter House (Red Meat)}	{1786, 1593, 1553, 1736, 1410, 1291, 1541, 1369, 1839, 2000, 1772, 1694}
{Animal Age ≤ 1 year old, Breed Type = beef, Sender Area = 14, easting (300001-400000) and northing (100001-200000), Receiver Location Type = Agricultural Holding, Number Cattle Moved ≤ 5}	{2098, 1925, 2854, 3051, 3364, 2705, 2793, 2469, 3018, 3189, 3031, 2336}

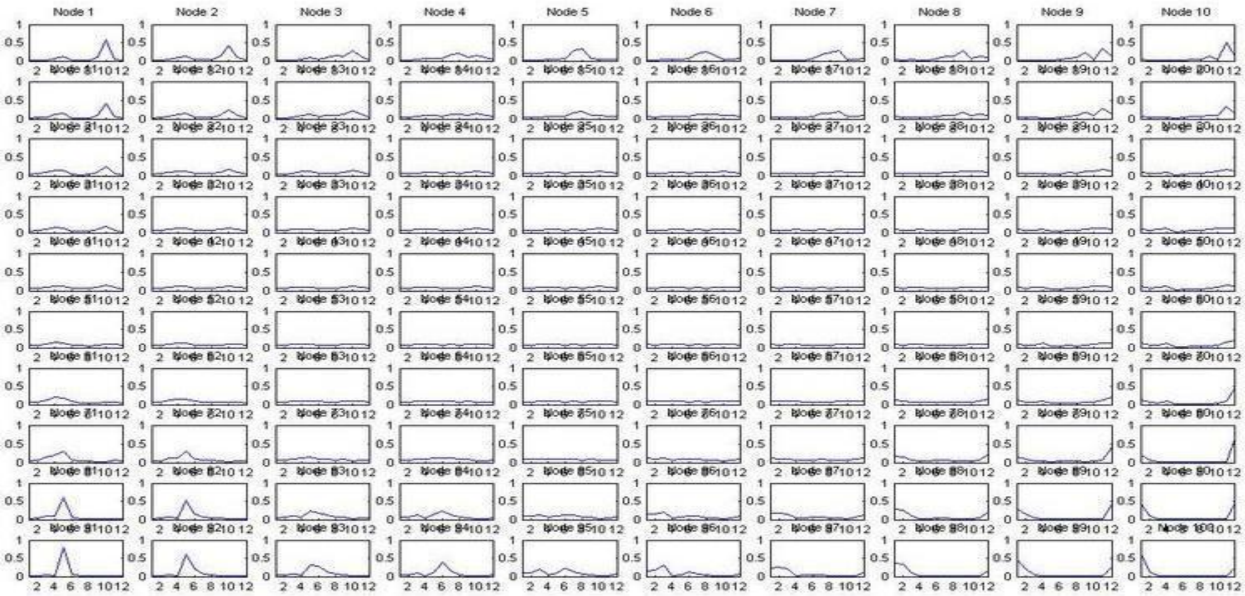

Figure 6: CTS prototype map generated using 2003 episode}

8.4 Cattle Movement Trend Migration Visualisation and Animation

Using the IGCV framework, once we have generated a sequence of SOM maps, we can perform some analysis. With respect to the CTS application we were particularly interested in how trends change with time (from one episode to the next). If we consider the maps for episode 2003 and 2004, presented in Figures 7 and 8 respectively, we wish to determine how trends move from one map to another; we are also interested in identifying "communities" of migrating trends. Using Visuset we can generate "plots" of the form shown in Figures 9 and 10. Figure 9 shows the migration of trends from episode 2003 to episode 2004, while Figure 10 shows the migration of trends from 2004 to 2005. In both cases the Min-Rel threshold was set to 0.2

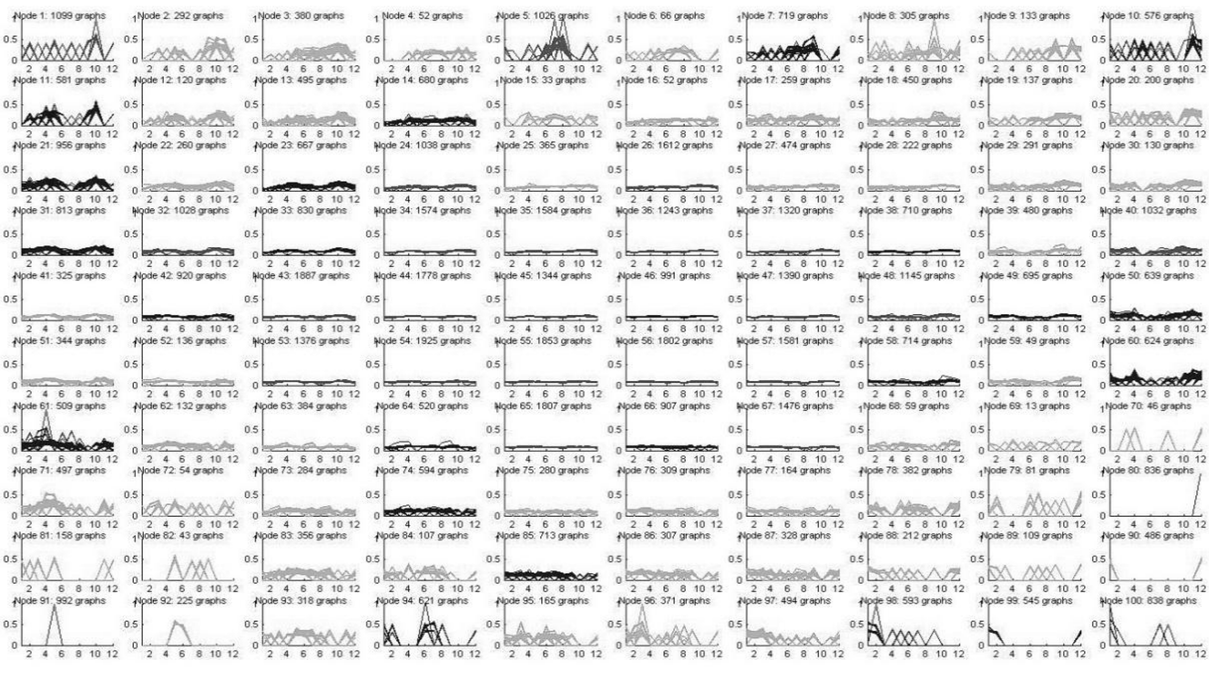

Figure 7: CTS Map for 2003 episode

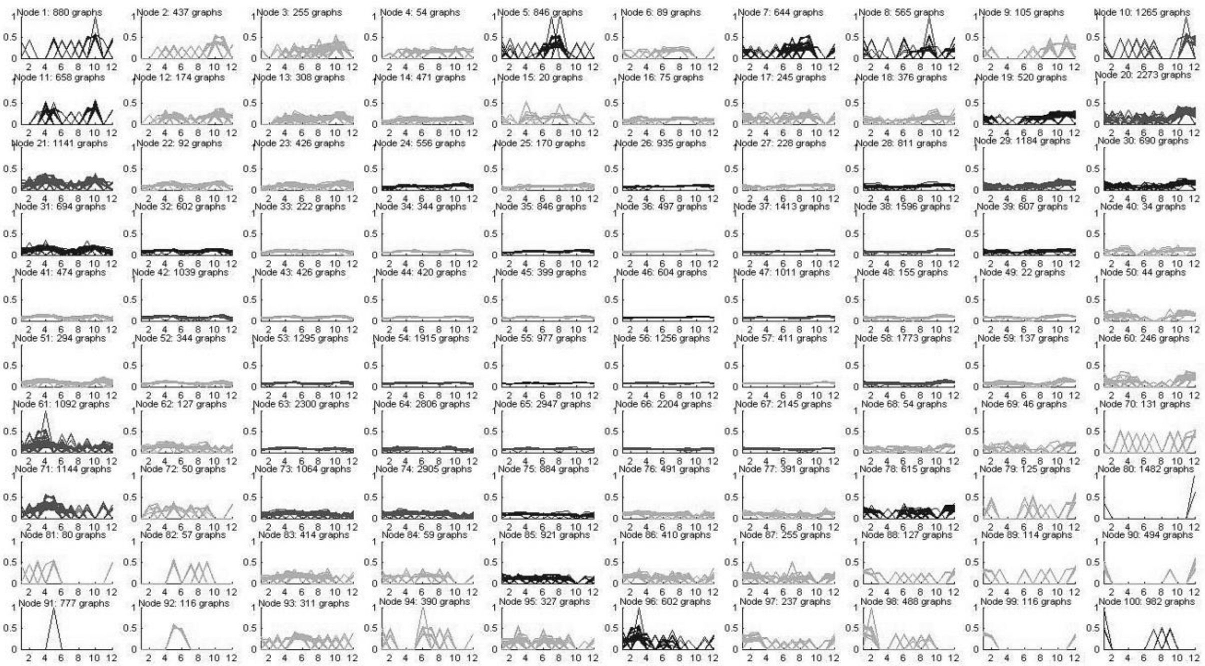

Figure 8: CTS Map for 2004 episode

Inspection of Figure 9 shows that the plot displays 45 nodes out of a total of 100, thus only 45 nodes included links with a C-value greater than 0.2 (and are therefore deemed interesting). The circular pattern in which the nodes are arranged on completion of the spring model algorithm is typical of the display produced (initially all nodes are placed along a diagonal). Several islands are displayed, determined using the Newman method described above, including a large island comprising eight nodes. The nodes are annotated with an identifier (the “from” SOM node number) and the arcs with their C-value number. From the map we can see that there are a relatively large number, 30 in all, of self-links; excluding self-links there are only 18 links indicating that, with respect to the 2003 and 2004 episodes, the trends are fairly constant. However, we can deduce that (for example) trends are migrating from node 34 to node 44, and from node 44 to 54. From Figure 6, we can observe that the nodes hold a fairly similar shape of trend line which has consistent numbers of cattle movement throughout the 12 month time stamps.

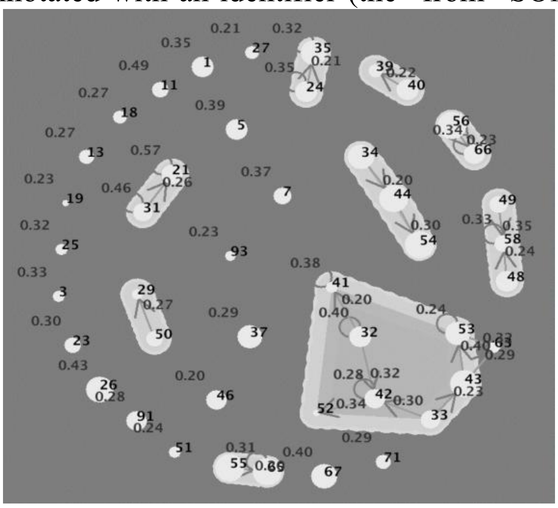

Figure 9: Visuset visualisation (map) indicating movement of trends from episode 2003 to episode 2004

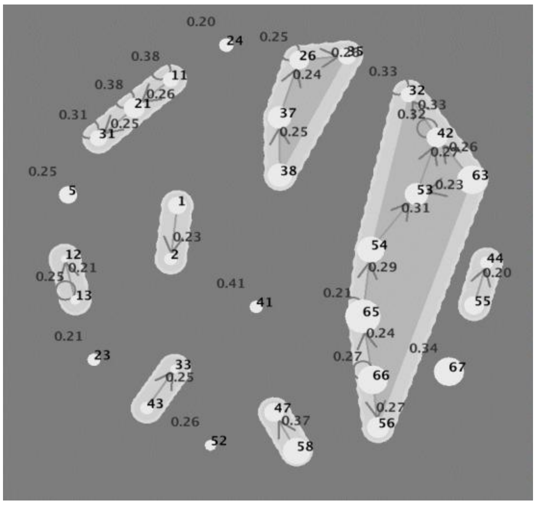

Figure 10: Visuset visualisatipon (map) indicating movement of trends from episode 2004 to episode 2005

Figure 10 shows the migration of trends from episode 2004 to episode 2005. Comparing this map with the previous, 2003-2004, map we can see that more “islands” have appeared indicating more trend migration communities. We can, for example, notice that whereas between 2003 and 2004 trends were migrating from node 44 to 54, in 2004 to 2005 there was no such migration. To give one more example, in 2003 and 2004 trends migrated from node 31 to 21, and then in 2004 to 2005 they moved back from node 21 to 31. We can also note that node 34 is not displayed in the 2004-2005 map because the C-values for its associated links are all below the Min-Rel threshold value of 0.2 (in the 2003-2004 map the C-value displayed for node 34 was only 0.2 so this is not surprising). When the animation provided with Visuset is run (although this cannot be illustrated here) we can see that node 34 disappears half way through the animation, thus indicating that the C-value is about 1.9.

9. Conclusion

The IGCV trend mining framework has been described. The framework comprises four distinct stages: Identification, Grouping, Clustering and Visualisation. During the identification stage trends are identified and extracted. To facilitate interpretation, during the grouping stage trends that display similar features are collected together. To further facilitate interpretation, during the clustering stage, the migration of trends is considered and “communities” of trend migrations identified. These trend migrations are then presented, using visualisation software (Visuset), in the final visualisation stage. Detail concerning each of these four stages has been presented. The single most significant contribution of the paper is the visualisation mechanism and its associate techniques. The operation of the framework was illustrated using a sequence of networks extracted from the Cattle Tracking System (CTS) in operation in Great Britain. However, although the framework is directed at the identification, extraction and analysis of trends in social networks, it could equally well be applied to other forms of temporal data such as temporally stamped graph data or longitudinal data.

References

AGARWAL, P. AND SKUPIN, A. (2008) Self-organising Maps: Applications in Geographic Information Science. John Wiley and Sons Ltd.

AGRAWAL, R., IMIELINSKI, T., AND SWAMI, A. (1993) Mining Association Rules between Sets of Items in Large Databases. Proc ACM SIGMOD International Conference on Knowledge Discovery and Data Mining (KDD'93), ACM, pp 207-216.

AGRAWAL, R. AND SRIKANT, R. (1994) Fast Algorithms for mining Association Rules. Proc. 20th Very Large Data Bases Conference (VLDB'94), pp 487-449.

AGRAWAL, R. AND SRIKANT, R. (1995) Mining sequential patterns. Proc 11th International Conference on Data Engineering, ICDE '95, pp 3-14.

CHOUDHURY, M.D., SUNDARAM, H., JOHN, A. AND SELIGMANN D.D. (2008) Can blog communication dynamics be correlated with stock market activity? Proc of the 19th ACM Conference on Hypertext and hypermedia, ACM, pp 55-60.

COENEN, F.P., GOULBOURNE, G. AND LENG, P. (2001) Computing Association Rules Using Partial Totals. Proc. PKDD, LNCS 2168, Springer, pp 54-66.

COENEN, F., LENG, P. AND AHMED, S. (2004) Data Structures for Association Rule Mining: T-trees and P- trees. IEEE Transactions on Data and Knowledge Engineering, vol 16(6), pp 774-778.

CHEN, C. (2006) CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. Journal of the American Society for Information Science and Technology, vol 57(3), pp 359-377.

DONG, G. AND LI, J. (1999) Efficient Mining of Emerging Patterns: Discovering Trends and Differences. Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99), ACM, pp 43-52.

GLOOR, P.A., KRAUSS, J.S., NANN, S., FISCHBACH, K. AND SCHODER, D. (2008) Web Science 2.0: Identifying Trends Through Semantic Social Network Analysis. Social Science Research Network.

HARMS, S.K. AND DEOGUN, J.S. (2004) Sequential Association Rule Mining with Time Lags. Journal of Intelligent Information Systems, vol 22(1), pp 7-22.

HAVRE, S., HETZLER, E., WHITNEY, P. AND NOWELL, L. (2002) Theme River: Visualizing Thematic Changes in Large Document Collections. IEEE Transactions on Visualization and Computer Graphics, vol 8(1), pp 9-20.

KANDOGAN, E. (2001) Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 107-116.

KHAN, M.S., COENEN, F., REID, D., TAWFIK, H., PATEL, R. AND LAWSON, A. (2010) A Sliding Windows based Dual Support Framework for Discovering Emerging Trends from Temporal Data. Knowledge Based Systems, vol 23(4), pp 316-322.

KOHONEN, T. (1995) The Self Organizing Maps. Series in Information Sciences, vol. 30. Springer, Heidelberg.

KOHONEN, T. (1997) Exploration of Large Document Collections by Self-Organizing Maps. Proc. of 6th Scandinavian Conference on Artificial Intelligence, IOS Press, Amsterdam, Netherlands, pp 5-7.

KOHAVI, R., ROTHLEDER, N.J. AND SIMOUDIS, E. (2002) Emerging trends in business analytics, Commun. ACM, vol 45(8), pp 45-48.

LAUW, H., LIM, E., PANG, H. AND TAN T. (2005) Social Network Discovery by Mining Spatio-Temporal Events. Computational and Mathematical Organization Theory, vol 11(2), Springer, pp 97-118.

LENT, B., AGRAWAL, R. AND SRIKANT, R. (1997) Discovering Trends in Text Databases. Proc ACM SIGMOD International Conference on Knowledge Discovery and Data Mining (KDD'93), ACM, pp 227-230.

MANNILA, H., TOIVONEN, H. AND VERKAMO, A. (1997) Discovery of Frequent Episodes in Event Sequences. Data Mining and Knowledge Discovery vol 1, pp 259-289.

NEWMAN, M.E.J. (2004) Fast Algorithms for Detecting Community Structure in Networks. Phys. Rev. E 69, 066113, pp 1-5.

NEWMAN, M.E.J. AND GIRVAN, M. (2004) Finding and evaluating community structure in networks. Phys. Rev. E 69, 026113, pp 1-15.

NISHIKIDO, T., SUNAYAMA W. AND NISHIHARA, Y. (2009) Valuable Change Detection in Keyword Map Animation. Proc. 22nd Canadian Conference on Artificial Intelligence, Springer-Verlag, LNCS 5549, pp 233-236.

NOHUDDIN, P.N.E., COENEN, F., CHRISTLEY, R. AND SETZKORN, C. (2010a) Trend Mining in Social Networks: A Study Using A Large Cattle Movement Database. Proc. 10th Ind. Conf. on Data Mining, Springer LNAI 6171, pp 464-475.

NOHUDDIN, P.N.E., CHRISTLEY, B., COENEN, F. AND SETZKORN, C. (2010b) Detecting Temporal Pattern and Cluster Changes in Social Networks: A study focusing UK Cattle Movement Database. Proc. 6th Int. Conf. on Intelligent Information Processing (IIP'10), IFIP, pp 163-172.

NOHUDDIN, P.N.E., CHRISTLEY, R., COENEN, F., PATEL, Y., SETZKORN, C. AND WILLIAMS, S. (2012) Finding “Interesting” Trends in Social Networks Using Frequent Pattern Mining and Self Organizing Maps. Knowledge Based Systems, vol 29, pp 104–113.

RAZA, J. AND LIYANAGE, J. P. (2008) An integrated qualitative trend analysis approach to identify process abnormalities: a case of oil export pumps in an offshore oil and gas production facility. Proc. of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering, Professional Engineering Publishing, vol 223 (4), pp 251-258.

RICHARDSON, M. AND DOMINGOS, P. (2002) Mining Knowledge Sharing Sites for Viral Marketing, Proc ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02), ACM, pp 61-70.

ROBERTSON, G., FERNANDEZ, R., FISHER, D., LEE, B. AND STASKO, J. (2008) Effectiveness of Animation in Trend Visualization. Transactions on Visualization and Computer Graphics, vol 14(6), pp 1325-1332.

SAFAEI, M., SAHAN, M. AND ILKAN, M. (2009) Social Graph Generation and Forecasting Using Social Network Mining. 33rd Annual IEEE International Computer Software and Applications Conference, Compsac, vol. 2, pp 31-35.

SCHRECK, T., BERNARD, J., LANDESBERGER T. AND KOHLHAMMER, J. (2009)Visual Cluster Analysis of Trajectory Data with Interactive Kohonen Maps. Information Visualization (2009) vol 8, pp 14-29.

SOMARAKI, V., BROADBENT, D., COENEN, F. AND HARDING, S. (2010). Finding Temporal Patterns in Noisy Longitudinal Data: A Study in Diabetic Retinopathy. Proc. 10th Ind. Conf. on Data Mining, Springer LNAI 6171, pp 418-431.

STREIBEL, O. (2008) Trend Mining with Semantic-Based Learning. Proc. of CAiSE-DC (2008), Free University Berlin publication, vol. 358.

SUGIYAMA K. AND MISUE, K. (1995) Graph Drawing by the Magnetic Spring Model, Journal of Visual Languages and Computing, vol. 6, No. 3, pp 217-231.

WASSERMAN, S. AND FAUST, K. (2006) Social Network Analysis: Methods and Applications. Cambridge University Press.

XU Z., TRESP, V., ACHIM, R. AND KERSTING, K. (2008) Social Network Mining with Nonparametric Relational Models. Advances in Social Network Mining and Analysis - the Second SNA-KDD Workshop at KDD 2008, LNCS vol. 5498 (2010), pp 77-96.

The authors

Puteri N. E. Nohuddin

Puteri N. E. Nohuddin received a BSc in Computer Science from the University of Missouri - Columbia, USA in 1995 and an MSc in Information Technology from the University of Technology MARA, Malaysia in 2003. Currently, she is a Ph.D. student at the Department of Computer Science, University of Liverpool. Her research interests include data mining, social network mining and trend analysis.

Wataru Sunayama

Wataru Sunayama received his BSc from the Department of Control Engineering, Osaka University, in 1995, completed the first half of the doctoral program in 1997, and withdrew from the second half in 1999 to become a research associate in the Graduate School. He received a Doctor of Engineering degree from Osaka University in 2000. He is currently working at Hiroshima City University as an associate professor.

Robert Christley

Robert Christley is a Senior Lecturer in Epidemiology and deputy director of the Department of Epidemiology & Population Health at the University of Liverpool. He is also a Co-director of the national Centre for Zoonosis Research. He has extensive experience in Social Network Analysis, particularly of animal-trade networks, where his particular interests lie in the impact of network structure on the potential for spread of disease, and in the impact of legislative and other change on network structure.

Frans Coenen

Frans Coenen has a general background in AI, and has been working in the field of data mining and Knowledge Discovery in Data (KDD) for the last twelve years. His research interest includes in: Social Network Mining; Trend Mining; the mining of non-standard data sets such as Graph, Image and document collections. Current applications include the classification of retina image and MRI scan data, and the discovery of trends in cattle movement data. He is currently a senior lecturer within the Department of Computer Science at the University of Liverpool where he is the director of studies for the department's online MSc programmes.

Christian Setzkorn

Christian Setzkorn holds a PhD in data mining using multi-objective evolutionary algorithms (MOEAs). Christian is currently working for the National Centre for Zoonosis Research, UK as a research assistant. He develops tailor made software solutions for the collection of data and their analysis. His research focuses on the application of novel data mining techniques to real world data to perform, for example, text mining, classification, clustering and social network analysis.