Trend Mining and Visualisation in Social Networks

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

Abstract A framework, the IGCV (Identification, Grouping, Clustering and Visualisation) framework, is described to support the temporal analysis of social network data. More specifically the identification and visualisation of "traffic movement" of patterns in such networks, and how such patterns change over time. A full description of the operation of IGCV is presented, together with an evaluation of its operation using a cattle movement network.

1 Introduction

Trend mining is concerned with the application of data mining techniques to extract trends from time stamped data collections [11, 12]. Likewise, social network mining is normally directed at finding communities but in this work we are interested in the "traffic" flow in such networks. The work described in this paper is directed at trend mining of social networks. The trends of interest are "traffic movement" patterns in which trends are defined in terms of the fluctuations of traffic between nodes in the networks. The main issues associated with social network trend mining 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

1 Department of Computer Science, University of Liverpool, UK,
puteri, frans@liverpool.ac.uk

2 Graduate School of Information Sciences, Hiroshima City University, Japan,
sunayama@sys.info.hiroshima-cu.ac.jp

3 Dept. of Epidemiology & Population Health, University of Liverpool and National
Centre for Zoonosis Research, Leahurst, Neston, UK,
robc, c.setzkorn@liverpool.ac.uk

given social network; and, as an end result, produces a visualisation of the most significant trends. The process is predicated on the assumption that end users are interested in the manner in which frequent patterns and trends change over time. We refer to this framework as the IGCV (Identification, Grouping, Clustering and Visualisation) framework. IGCV comprises four stages (Figure 1):

1. **Trend Identification**: The application of frequent itemset mining techniques to define and identify frequent patterns and 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. **Pattern Migration Clustering**: Identification of "communities" of pattern migrations, within the SOM groupings, using a hierarchical clustering mechanism based on the Newman method.
4. **Pattern Migration Visualisation**: Visualisation of the pattern migrations using a *spring model* to display, what are considered to be the most significant, pattern migrations.

To evaluate the operation of IGCV a social network extracted from the Cattle Tracing System (CTS) in operation in Great Britain has been used. CTS includes a database that records cattle movements through out Great Britain. By considering the *holding areas* (farms, markets, abattoirs, etc.) as nodes, and the cattle movement between holding areas as the traffic, a large scale social network can be derived.

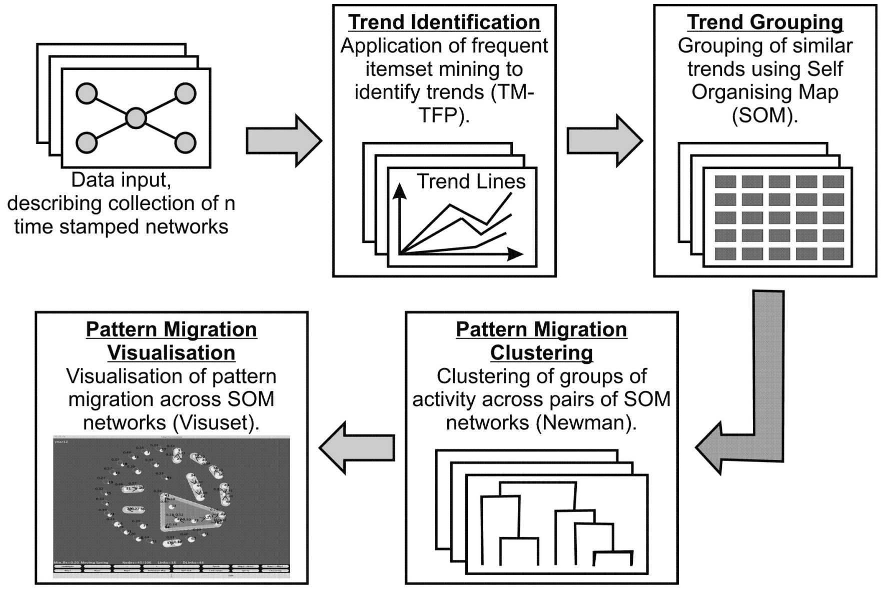

Fig. 1 Schematic for The IGCV Framework

2 Related Work

Social networks are communities of interacting entities. Well known examples include web-based applications such as Facebook, Bebo and Flickr. Social network mining is typically directed at identifying patterns and sub-communities (clusters) within network data [21, 23]. 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. There has been some previous work on social networks trend analysis. For example, Gloor *et al.* introduced a trend analysis algorithm to generate trends from Web resources [8]. There has also been some work on the identification of trends in social networks in the context of online viral marketing [19] and stock market activities [2]. However these systems do not define trends in terms of traffic movement frequency, and do not include any visualisation.

The IGCV framework provides for the visualisation of pattern and trend changes using Visuset, software specifically developed for this purpose. There has been some previous work on network trend visualisation. For example Havre *et al.* [9] described a technique for displaying thematic changes as *river flows* so that changes of topics can be observed. Chen [6] describes a system to visualise 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 can display trend transitions as an animation so as to demonstrate how trends change over a given period. Robertson et al. [20] also introduced a system to illustrate trends by animation in the form of *traces*, however in this system trend 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 Definitions

The input to IGCV comprises a sequence of n time stamped data sets, $D = \{d_1, d_2, \ldots, d_n\}$. 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. In the case of the CTS application a number of node categories are identified (farms, markets, abattoirs, etc.). The 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, comprise a subset of a global set of binary valued attributes $A = \{a_1, a_2, \dots, a_m\}$.

A pattern trend t is then defined in terms of the frequency of occurrence, over time, of discovered patterns within the input data. The trends are conceptualised as *trend lines*, one per pattern, representing a mapping of frequency of occurrence against time. Similar trends (time series), which are mapped by the trained SOM, can still belong to very different patterns. To identify changes in frequent patterns (or lack of them), the number of time stamps is subdivided into *e episodes*, 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. Thus, a trend t comprises a set of values $\{v_1, v_2, \dots, v_n\}$ where each value represents an occurrence count. The collection of trends, T, that we wish to analyse thus comprise a sequence of sub-collections $\{T_1, T_2, \dots, T_e\}$ (where e is the number of episodes).

4 Pattern and Trend Identification

The first stage in the IGCV framework is to identify the traffic movement trends of interest. In the work described in this paper, we define trends in terms of the changing frequency of traffic movement patterns found in social network data. Frequent patterns are sets of attributes that "frequently" co-occur within data according to some user specified *frequency threshold* [1]. To mine pattern trends an extended version of the TFP (Total From Partial) algorithm [3, 4] was used, TM-TFP (Trend Mining TFP). 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, founded on the well documented *support framework*, whereby a frequency count threshold (the *support threshold*) defines a frequent pattern. TM-TFP incorporates a TM-T-tree to store the desired patterns. Further details of the TM-TFP can be found in [16] and [17]. The output from the TM-TFP algorithm is a collection of trends $T = \{T_1, T_2, \dots, T_e\}$. Typically, a large number of trends are identified. In the case of the CTS network, Table 1 presents the number of patterns discovered using three different support thresholds (the first column gives the episode identifier). The large number of discovered trends, and the consequent difficulty in analysing the trends, was one of the main motivations for the IGCV framework.

Episode	Support Threshold		
(year)	0.5%	0.8%	1.00%
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

Table 1. Number of patterns identified using TM-TFP for a sequence of four CTS network episodes and a range of support threshold values.

5 Trend Grouping

The second stage in the IGCV process is to group the discovered trends. The intuition here is that end users are 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 [10]. 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 the record. For each record, once it has been assigned to a node, the network's weightings are adjusted to reflect the new position. A feature of the network is that adjacent nodes hold similar records, the greatest dissimilarity is thus 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. The most effective was found to be training the SOM using trends 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 $M = \{M_1, M_2, \ldots, M_e\}$.

SOMs are often described as a visualisation technique. However, given a large and/or complex data set 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 potential solution was to increase the size of the grid, however this would have resulted in an undesirable computational overhead and was found not 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. There is no certain scientific method [5] to specify the optimum value for $n \times m$ for how many clusters should be presented in SOM. A 10×10 node SOM was chosen as a result of earlier experiments repeated in [16, 18].

6 Pattern 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 trends associated with individual patterns changed, or did not change, with time. Thus whether patterns associated with a trend in an a given episode remained associated with that trend in following episodes, or whether patterns move from one trend to another. Thus we are also interested in how patterns *migrate* 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 represented using a second network, a "SOM network" (containing potentially $i \times j$ nodes and $(i \times j)^2$ links, including "self links"). The nodes in the SOM network represent nodes in the SOM maps, and the links the migration of trends from M_{e_k} to $M_{e_{k+1}}$. It was also considered desirable to display "communities" within these networks, i.e. clusters of SOM network nodes which were "strongly" connected, i.e. where substantial migration had taken place. To this end a hierarchical clustering mechanism, founded on the Newman method [13] 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. Best similarity is defined in terms of a *Q-value* (a "modularity" value). Experimentation conducted by Newman and Girvan [14] indicates that if the Q-value is above 0.3 then communities can be said to exist within the target network; a value of 0.3 was thus adopted with respect to IGCV. Note that if all nodes are placed in one group the Q-value will be 0.0 (i.e. a very poor clustering).

7 Pattern Migration Visualisation and Animation using Visuset

IGCV provides two forms of visualisation (integrated into a single software system called Visuset):

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

In each case the visualisation (animation) includes the pattern migration clusters discovered, using Newman, as described above. The clusters 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.

7.1 Visualisation of Trend Migration

For the visualisation, IGCV locates nodes in a 2-D "drawing area" using the *Spring Model* [22]. 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 objects connected by the 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 a set of (random) default locations (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):

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

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 trends at node i (j) in SOM M_{e_k} ($M_{e_{k+1}}$). 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:

1. Set drawing area size constants, $SIZEX$ and $SIZEY$.
2. 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.
3. 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.
4. For all node pairs determine the actual pixel distance $RDIST_{ij}$ (where i and j are node numbers).
5. For all nodes, recalculate the coordinates using equations 2 and 3 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 $Node_{i_x} - Node_{j_x}$ ($Node_{i_y} - Node_{j_y}$).
6. 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.

7. Go to Step 4.

$$node_{i_x} = node_{i_x} + \sum_{j=1}^{j=n} \left(dx_{i_j} \times K \times \left(1 - \frac{IDIST_{i_j}}{RDIST_{i_j}} \right) \right) \tag{2}$$

$$node_{i_y} = node_{i_y} + \sum_{j=1}^{j=n} \left(dy_{i_j} \times K \times \left(1 - \frac{IDIST_{i_j}}{RDIST_{i_j}} \right) \right) \tag{3}$$

For the current version of Visuset $SIZEX = 1280\ pixels$ and $SIZEY = 880\ pixels$, and the spring constant was set to 0.2. These values were chosen, as result of sequence of experiments, because they tended to give the most desirable end result. The stopping threshold can be set to 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 [15].

In the current implementation of Visuset nodes are depicted as: single nodes (i.e. self links where the "migration" is to the same node), node pairs linked by an edge, chains of nodes linked by a sequence of edges, or node clusters (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 Pattern Migration

The animation mechanism, provided by Visuset, can be applied to pairs of visualisations (as described above) to illustrate the migration of patterns 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. 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). The example presented here focuses on three subsequent maps, however the same technique can be applied to a series of *N* maps.

8 Demonstration

In this section the operation of the ICGV framework is presented in terms of 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 above.

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) [7]. Cattle movements can be "one-of" 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. 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. The maximum number of cattle moved (traffic value) between any pair of locations for a single time stamp was approximately 40 animals.

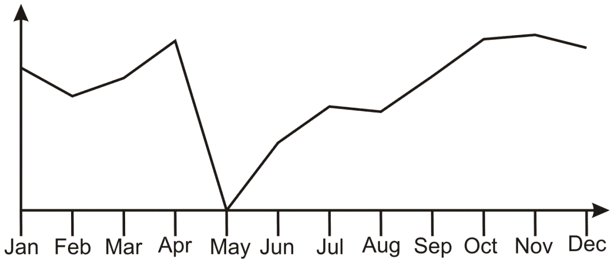

Fig. 2 Example frequent pattern trends associated with a single episode

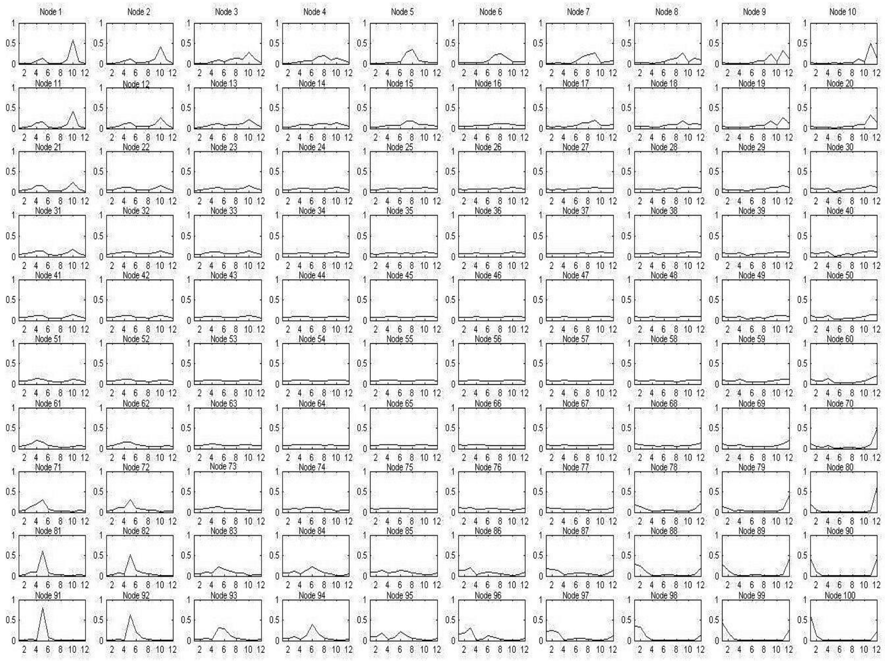

Fig. 3 CTS prototype map generated using 2003 episode

8.2 Cattle Movement Trend Mining

IGCV commences with the identification of trends using the TM-TFP algorithm. For the experiments reported here, a support threshold values of 0.5% was used. The number of identified frequent patterns discovered over the four episodes (2003, 2004, 2005 and 2006) were 63117, 66870, 65154 and 62713 respectively. Each pattern is attached with a trend line, the frequencies of occurrences in an episode. An example trend line is shown in Figure 2 for the the frequent pattern $\{AnimalAge = 2\_to\_5\_years,\ Breed = Friesian,\ BreedType = dairy,\ ReceiverLocationType = SlaughterHouse(RedMeat)\}$.

8.3 Cattle Movement Trend Grouping

To identify groupings within the collection of trends identified using TM-TFP the SOM software was initialising with a 10×10 node map, and trained using the frequent pattern trends produced for the (earliest) 2003 episode. The resulting prototype map is shown in Figure 3. The prototype map groups similar trends occurring in the 2003 episode so that (say) seasonal variations can be identified. For example: node 3, node 4, node 13 and node 14 describe trends where the number of cattle movements increases slightly in March, June and October; nodes 95 and 96 both describe trends where the number of cattle movement is considerably higher in spring and summer; and so on. Once the initial prototype map had been generated a sequence of trend line maps was produced, one for each episode.

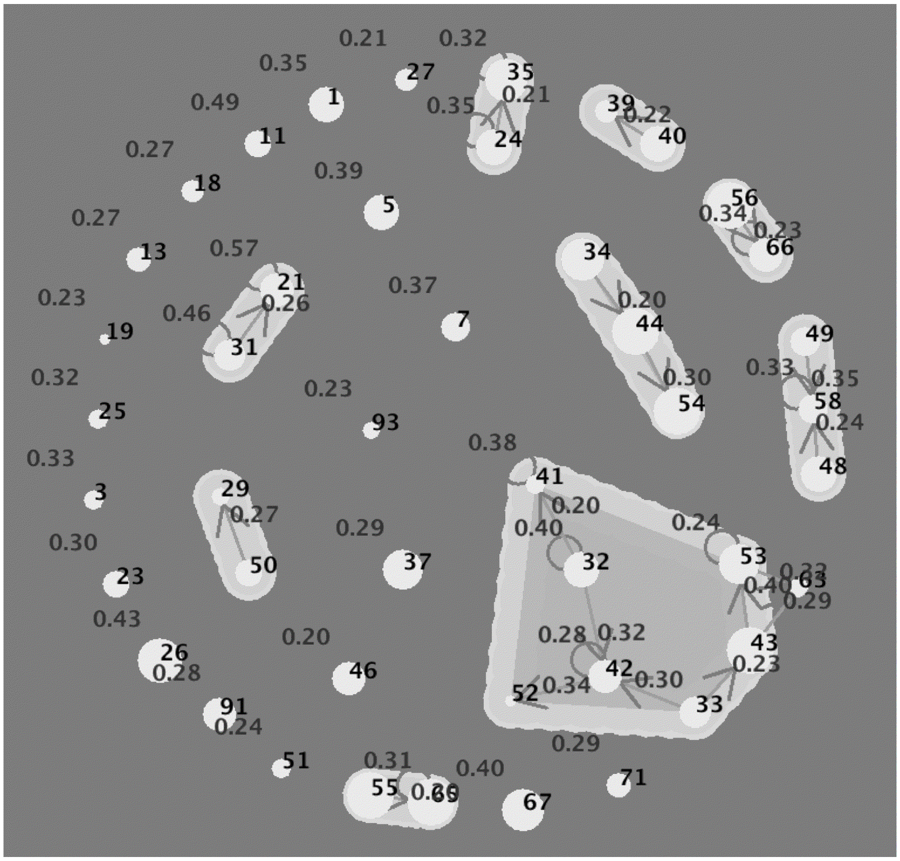

Fig. 4 Visuset visualisation (map) indicating movement of trends from episode 2003 to episode 2004

8.4 Cattle Movement Pattern Migration Visualisation and Animation

Using the IGCV framework, once we have generated a sequence of SOM maps, we can perform some further analysis. With respect to the CTS application we were particularly interested in how patterns associated with trends change with time (from one episode to the next), i.e. we wish to determine how patterns move from one map to another. As noted above, we are also interested in identifying clusters of migrating patterns. Using Visuset we can generate "plots" of the form shown in Figures 4, which shows the migration of patterns from episode 2003 to episode 2004 (A Min-Rel threshold of 2.0 was used).

Inspection of Figure 4 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 2.0 (and are therefore deemed interesting). Several islands are displayed, determined using the Newman method described above, including a large island comprising eight nodes. The islands indicate communities of pattern migrations. 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 patterns are fairly constant. However, we can deduce that (for example) patterns are migrating from node 34 to node 44, and from node 44 to 54 (thus indicating a trend change). From Figure 3, 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 and thus migration between these three trends is understandable.

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, patterns and 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 patterns is considered and "communities" of pattern migrations identified. These pattern migrations are then presented, using visualisation software (Visuset), in the final visualisation stage. Detail concerning each of these four stages has been presented. The operation of the framework was illustrated using a sequence of networks extracted form the Cattle Tracing System (CTS) in operation in Great Britain.

References

1. Agrawal, R., Imielinski, T., and Swami, A. 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 (1993)
2. Choudhury, M.D., Sundaram, H., John, A. and Seligmann D.D. Can blog communication dynamics be correlated with stock market activity? Proc of the 19th ACM Conference on Hypertext and hypermedia, ACM, pp 55–60 (2008)
3. Coenen, F.P., Goulbourne, G. and Leng, P. Computing Association Rules Using Partial Totals. Proc. PKDD, LNCS 2168, Springer, pp 54–66 (2001)
4. Coenen, F., Leng, P. and Ahmed, S. Data Structures for Association Rule Mining: T-trees and P- trees. IEEE Transactions on Data and Knowledge Engineering, 16(6), pp 774–778 (2004)
5. Cottrell, M., Rousset, P. A powerful Tool for Analyzing and Representing Multidimensional Quantitative and Qualitative Data. In Proceedings of IWANN 97. LNCS, Springer Berlin / Heidelberg, vol. 1240, pp 861-871 (2006)
6. Chen, C. CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. Journal of the American Society for Information Science and Technology, 57(3), pp 359–377 (2006)
7. Defra. Livestock movements, identification and tracing: Cattle Tracing System. http://www.defra.gov.uk/foodfarm/farmanimal/movements/cattle/cts.htm
8. Gloor, P.A., Krauss, J.S., Nann, S., Fischbach, K. and Schoder, D. Web Science 2.0: Identifying Trends Through Semantic Social Network Analysis. Social Science Research Network. (2008)
9. Havre, S., Hetzler, E., Whitney, P. and Nowell, L. ThemeRiver: Visualizing Thematic Changes in Large Document Collections. IEEE Transactions on Visualization and Computer Graphics, 8(1), pp 9–20 (2002)
10. Kohonen, T. The Self Organizing Maps. Series in Information Sciences, vol. 30. Springer, Heidelberg. (1995)
11. Kohavi, R., Rothleder, N.J. and Simoudis, E. Emerging trends in business analytics, Commun. ACM, 45(8), pp 45–48 (2002)
12. Lent, B., Agrawal, R. and Srikant, R. Discovering Trends in Text Databases Proc ACM SIGMOD International Conference on Knowledge Discovery and Data Mining (KDD'93), ACM, pp 227–230 (1997)
13. Newman, M.E.J. Fast Algorithms for Detecting Community Structure in Networks. Phys. Rev. E 69, 066113, pp 1–5 (2004)
14. Newman, M.E.J. and Girvan, M. Finding and evaluating community structure in networks. Phys. Rev. E 69, 026113, pp 1–15 (2004)
15. Nishikido, T., Sunayama W. and Nishihara, Y. Valuable Change Detection in Keyword Map Animation. Proc. 22nd Canadian Conference on Artificial Intelligence, Springer-Verlag, LNCS 5549, pp 233–236 (2009)
16. Nohuddin, P.N.E., Coenen, F., Christley, R. and Setzkorn, C. 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 (2010)
17. Nohuddin, P.N.E., Christley, B., Coenen, F. and Setzkorn, C. 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 (2010)
18. Nohuddin, P.N.E., Christley, R., Coenen, F., Patel, Y., Setzkorn, C. and Williams, S. Social Network Trend Analysis Using Frequent Pattern Mining and Self Organizing Maps. Research and Development in Intelligent Systems XXVII, Springer-Verlag London Limited, pp 311 (2011)
19. Richardson, M. and Domingos, P. Mining Knowledge Sharing Sites for Viral Marketing, Proc ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02), ACM, pp 61–70 (2002)

20. Robertson, G., Fernandez, R., Fisher, D., Lee, B. and Stasko, J. Effectiveness of Animation in Trend Visualization. Transactions on Visualization and Computer Graphics, 14(6), pp 1325–1332 (2008)
21. Safaei, M., Sahan, M. and Ilkan, M. Social Graph Generation and Forecasting Using Social Network Mining. 33rd Annual IEEE International Computer Software and Applications Conference, Compsac, vol. 2, pp 31–35 (2009)
22. Sugiyama K. and Misue, K. Graph Drawing by the Magnetic Spring Model, Journal of Visual Languages and Computing, Vol. 6, No. 3, pp 217–231 (1995)
23. Xu Z., Tresp, V., Achim, R. and Kersting, K. 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 (2008)