Social Network Trend Analysis Using Frequent Pattern Mining and Self Organizing Maps

Puteri N. E. Nohuddin[1], Rob Christley[2], Frans Coenen[3], Yogesh Patel[1,3], Christian Setzkorn[2], Shane Williams[3]

Abstract A technique for identifying, grouping and analysing trends in social networks is described. The trends of interest are defined in terms of sequences of *support values* for specific patterns that appear across a given social network. The trends are grouped using a SOM technique so that similar trends are clustered together. A cluster analysis technique is then applied to identify "interesting" trends. The focus of the paper is the Cattle Tracing System (CTS) database in operation in Great Britain, and this is therefore the focus of the evaluation. However, to illustrate the wider applicability of the trend mining technique, experiments using a more standard, car insurance, temporal database are also described.

1 Introduction

Social network mining is a popular area of study. The aim is to extract knowledge from such networks. The networks themseves are conceptualised as graphs comprising nodes and links. Common examples of social networks are www applications such as Facebook, Bebo and Flicker. However, in the wider context social networks can include business communities, file sharing systems and co-authoring frameworks. In each case, the nodes represent individuals and the links communications. These communications often take the form of text (emails) but can be files (photopgraphs, movies, etc.). In this paper, we have abstracted out the concept of social networks even further to encompass the Cattle Tracing System (CTS) in operation in Greart Britain (GB). CTS incorporates a database that records cattle movements. The CTS database can be viewed as a large scale social network where the nodes

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

2 School of Veterinary Science, University of Liverpool and National Centre for Zoonosis Research, Leahurst, Neston, UK,
robc, c.setzkorn@liverpool.ac.uk

3 Deeside Insurance Ltd., Deeside, UK,
yogesh, shane@deesideinsurance.co.uk

represent cattle holding areas (farms, markets, abattoirs, etc) and the links are cattle movements between locations. This *cattle movement* social network can be mined, using relatively standard network mining techniques, to find (say) clusters of nodes. However, in this paper, the authors are interested in the dynamic mining of such social networks, as opposed to their static mining. The authors are particularly interested in mechanisms for identifying trends in social network in general, and the cattle movement social network in particular. The objective is to identify trends and variations in these trends. In the context of the cattle movement social network, the identification of trends and change points will provide knowledge of (say) the effect of the introduction of new legislation, or indicate changes in working practices, it will also give an insight into the way that cattle infections may spread through GB.

The trends we are interested in are defined as the changing frequencies with which common patterns occur across social network data. Trends are collected according to *epochs*, which can then be compared. The nature (duration) of an epoch is application dependent. However, for the cattle movement social network, it made sense to consider trends in terms of years (i.e. the duration of an epoch is 12 months) because this will serve to capture seasonal variations. Whatever the case, the epoch length is a user supplied variable that can be easily adjusted to fit alternative applications.

Using the proposed trend mining mechanism, a significant number of trends may be identified, too many to allow simple inspection by decision makers. Some mechanism was therefore required to allow the simple presentation of trend lines. The first technique advocated in this paper is to group (cluster) trends that display similar contours. To this end, Self Organising Map (SOM) technology has been adopted. Once the trends have been identified and grouped, we wish to determine how these trends change from epoch to epoch. The nature of the changes which we might be interested in will vary from application to application. For some applications, we may be interested in trends that remain constant, for others we may be interested in trends that change radically. To identify changes in trends, the advocated approach is to generate a sequence of SOM maps, one per epoch, and analyse how trends "move" (or do not move) from SOM to SOM (epoch to epoch).

The proposed approach to *trend mining*, using a sequence of SOMs has much wider application. Thus, although the focus of this paper is the cattle movement social network, for evaluation purposes, we also consider an alternative application, namely that of a customer database. More specifically a car insurance database containing requests for insurance quotes from potential customers. The "network" in this case is much simpler in that the nodes represent geographical locations which all communicate with a central "broker" node. The links in this case were labelled with the amount of traffic per time stamp (instead of the number of cattle moved per time stamp).

The contribution of this paper may thus be summarised as: (i) an unusual application of social network mining with respect to the CTS database, (ii) a mechanism for generating frequent pattern trends, (iii) a process for assisting the analysis of the identified trends using SOM technology and (iv) an approach to identify "interesting" changes in trends. The rest of this paper is organised as follows. Some previous work is described in Section 2. The proposed social network trend mining approach is described in Section 3. Sections 4 and 5 present an evaluation of the proposed technique, firstly using the cattle movement social network (the focus of this paper), and secondly a car insurance time stamped data set to illustrate the wider application of the proposed technique. Some conclusions are then presented in Section 6.

2 Previous Work

The general availability of advanced computer information systems have resulted in the rapid growth of temporal databases together with a corresponding desire to identify (mine) trends in these collections. For example, Google Trends, a public web facility that supports the identification of trends associated with keyword search volume [1]. Trend recognition processes can be applied to both qualitative and quantitative data, such as the forecasting of financial market trends based on numeric financial data, and usage of text corpi in business news [2]. Raza and Liyanage [4] proposed a trend analysis approach to mine and monitor data for abnormalities and faults in industrial production. There are many more examples, however, in this paper, we are interested in mining trends which are defined in terms of the changing frequency of individual patterns presented in the data.

A social network depicts the structure of some social entity, and normally comprises actors who are connected through one of more links [18]. To analyze this structure, techniques have been proposed which map and measure the relationships and flows between nodes. 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 a 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 wish to identify relationships between the nodes in the network by evaluating the spatio-temporal co-occurrences of events [6]. The latter is thus the focus of the work described in this paper.

As noted above, in this work, we define trends in terms of the changing frequency of patterns with time. A frequent pattern, as first defined by Agrawal et al. [7], is a subset of attributes that frequently co-occur according to some user specified support threshold. The frequent pattern idea has been extended in many directions. A number of authors have considered the nature of frequent patterns with respect to the temporal dimension, for example sequential patterns [8], frequent episodes [9], emerging patterns [10] and jumping and emerging patterns [3]. Many alternative frequent pattern mining algorithms, that seek to improve on Agrawal's original Apriori algorithm, have also been proposed. TFP (Total from Partial) [11] is one established algorithm that extends Apriori. For the work described here, TFP has been adapted for the purpose of trend mining.

Self Organising Maps (SOMs) were first introduced by Kohonen [13, 12]. Fundamentally, SOMs are a neural network based technique designed to reduce the number of data dimensions in some input space by projecting it onto a $n \times m$ "node map", which plots the similarities of the input data by grouping (clustering) similar data items together at nodes. The SOM learning process is unsupervised, in other words no predefined number of clusters is specified. Currently, there is no scientific method for determining the best values for $n \times m$, i.e. to identify how many clusters should be represented by a SOM. However, the $n \times m$ value does define a maximum number of clusters; although on completion some nodes may be empty [16]. Since SOM are based on competitive learning, the output nodes on the map compete among each other to be stimulated to represent the input data. With respect to the work described in this paper, we have adopted a SOM approach to group similar trends, and thus provide a mechanism for analysing social network trend mining results.

For many applications, such as those considered in this paper, we are interested in detecting changes in trends. This can be achieved by applying *cluster analysis* techniques to the SOM generated maps. Cluster analysis is concerned with the discovery of information about the relationship and/or similarity between clusters. When conducting cluster analysis practitioners are predominantly interested in cluster size enlargement and reduction and cluster membership migration. Several methods have been introduced to detect cluster changes and cluster membership migration. For example, Lingras et. al. [19] proposed the use of Temporal Cluster Migration Matrices (TCMMs) to visualize cluster changes in e-commerce sites usage that reflected changes in user spending patterns. A simple Euclidean distance measure is adopted in this paper.

3 The Trend Mining Mechanism

As noted in Section 1, a trend is defined as a sequence of *support* values, associated with a specific pattern, over a sequence of time stamps. The support of a pattern is the number of occurences of that pattern in the data set for some time stamp. The sequence of time stamps is referred to as an *epoch*. Thus, a trend t comprises a set of values $\{v_1, v_2, \ldots, v_n\}$ where n is the number of time stamps in the epoch. A trend associated with a particular pattern i is indicated by t_i. The jth value in a trend t_i is indicated by t_{ij}. We wish to identify changes in the trends associated with individual patterns and thus we wish to compare trends over two or more epochs. A sequence of trends T comprises a set of trends $\{t_1, t_2, \ldots, t_e\}$, where e is the number of epochs described by the sequence. The proposed approach (Figure 1) comprises three stages: (i) frequent pattern trend mining, (ii) trend clustering, and (iii) analysis of trend clusters.

3.1 Frequent Pattern Trend Mining

The input to the trend mining system comprises a binary valued, time stamped, data set $D = \{d_1, d_2, \ldots, d_{n \times e}\}$ (recall that n is the number of time stamps per epoch, and e

is the number of epochs under consideration). The records in each dataset in D comprise some subsets of a global set of binary valued attributes $A = \{a_1, a_2, \ldots, a_m\}$. The number of records in each data set need not be constant across the collection. The patterns we are interested in are thus also subsets of A. To limit the overall number of patterns a *support threshold* is used, in the same way as in Association Rule Mining (ARM). A pattern is not deemed to be "interesting" unless its number of occurrences in an individual dataset d is greater than this threshold. Some examples patterns are given in the Figure 1. Thus, the pattern {a,b,c,d} has a sequence of support values of {0, 0, 2500, 3311, 2718, 0, 0, 0, 2779} describing a nine time-stamp trend associated with a single epoch, similar sequences may be extracted for all e epochs. Note that a 0 support value indicates a support value below the support threshold.

To mine the trends, an extended version of the TFP algorithm [11] was used. TFP is an established frequent pattern mining algorithm distinguished by its use of two data structures: (i) a P-tree to encapsulate the input data and conduct a partial pattern count in the process, and (ii) a T-tree to store identified patterns. The T-tree is essentially a reverse *set enumeration tree* that allows fast *look up*. The TFP algorithm, in its original form, was not designed to address the temporal aspect of frequent pattern mining. The algorithm was therefore extended so that a sequence of $n \times e$ data sets could be processed and the frequent patterns stored in a way that would allow for differentiation between individual time stamps and epochs. The resulting algorithm was called TM-TFP (Trend Mining TFP) which incoporated a

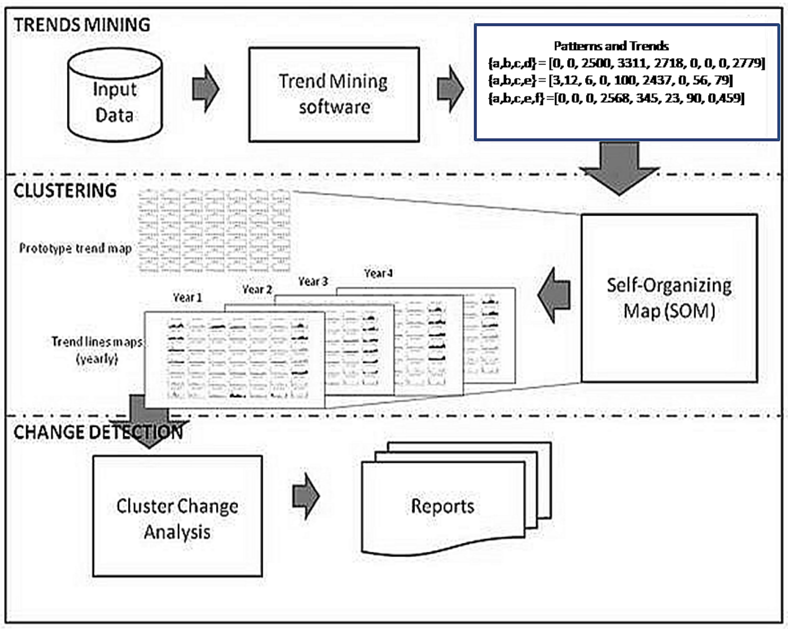
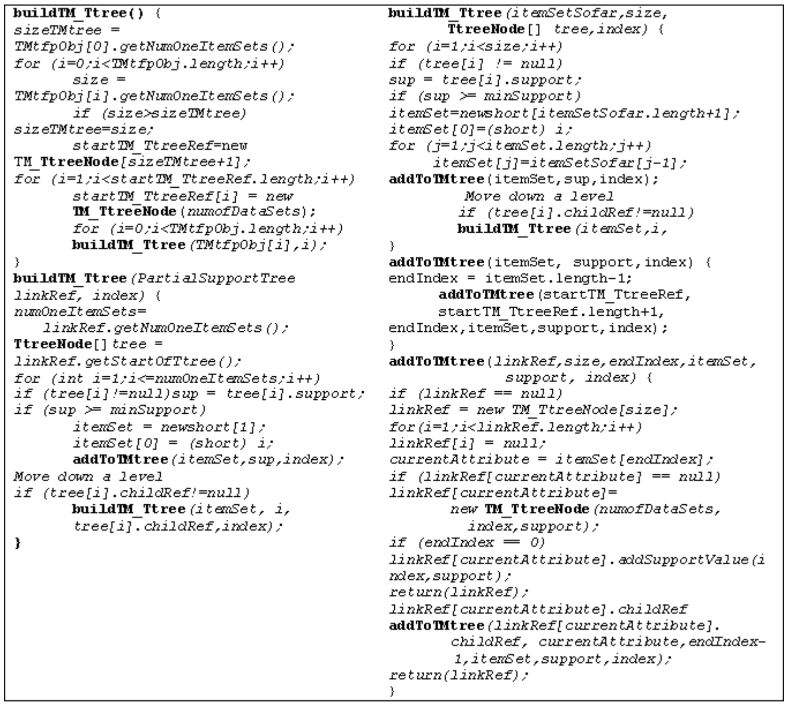

TRENDS MINING
Input Data
Trend Mining software
Patterns and Trends
{a,b,c,d} = [0, 0, 2500, 3311, 2718, 0, 0, 0, 2779]
{a,b,c,e} = [3,12, 6, 0, 100, 2437, 0, 56, 79]
{a,b,c,e,f} =[0, 0, 0, 2568, 345, 23, 90, 0,459]
CLUSTERING
Prototype trend map
Year 1
Year 2
Year 3
Year 4
Trend lines maps (yearly)
Self-Organizing Map (SOM)
CHANGE DETECTION
Cluster Change Analysis
Reports

Fig. 1 Trend Mining Framework

TM-T-tree to store the desired patterns. An overview of TM-TFP is given in Figure 2. The *buildTM_Tree* method processes the collection of T-trees built from the input data sets. The *addToTMtree* method adds an item set node to the TM T-tree with its support value. The resulting trends are the input data for the clustering process which is described in the following subsection.

3.2 Trend Clustering

The process described above for identifying trends operates successfully, but produces a great many trend lines when a low support threshold is used (the option to use a higher threshold does reduce the number of trends but entails the risk of missing potentially interesting trends). The large number of trends produced makes it difficult for decision makers to interpret the result. Some mechanism for assisting the desired interpretation was therefore desirable. The idea of clustering similar trends allows decision makers to focus on particular groups of trends. The concept of clustering is well established in the data mining community, however little work has been directed at clustering time series (trend lines). The approach advocated in this paper is to use Self Organising Maps (SOMs). Using the SOM concept one map was created per epoch. The SOM was initialized with $n \times m$ nodes such that each



```
buildTM_Ttree() {
sizeTMtree =
TMtfpObj[0].getNumOneItemSets();
for (i=0;i<TMtfpObj.length;i++)
      size =
TMtfpObj[i].getNumOneItemSets();
      if (size>sizeTMtree)
sizeTMtree=size;
      startTM_TtreeRef=new
TM_TtreeNode[sizeTMtree+1];
for (i=1;i<startTM_TtreeRef.length;i++)
      startTM_TtreeRef[i] = new
      TM_TtreeNode(numofDataSets);
      for (i=0;i<TMtfpObj.length;i++)
      buildTM_Ttree(TMtfpObj[i],i);
}
buildTM_Ttree(PartialSupportTree
linkRef, index) {
numOneItemSets=
   linkRef.getNumOneItemSets();
TtreeNode[] tree =
linkRef.getStartOfTtree();
for (int i=1;i<=numOneItemSets;i++)
if (tree[i]!=null)sup = tree[i].support;
if (sup >= minSupport)
      itemSet = newshort[1];
      itemSet[0] = (short) i;
      addToTMtree(itemSet,sup,index);
Move down a level
if (tree[i].childRef!=null)
      buildTM_Ttree(itemSet, i,
      tree[i].childRef,index);
}
```

```
buildTM_Ttree(itemSetSofar,size,
      TtreeNode[] tree,index) {
for (i=1;i<size;i++)
if (tree[i] != null)
sup = tree[i].support;
if (sup >= minSupport)
itemSet=newshort[itemSetSofar.length+1];
itemSet[0]=(short) i;
for (j=1;j<itemSet.length;j++)
     itemSet[j]=itemSetSofar[j-1];
addToTMtree(itemSet,sup,index);
        Move down a level
        if (tree[i].childRef!=null)
        buildTM_Ttree(itemSet,i,
}
addToTMtree(itemSet, support,index) {
endIndex = itemSet.length-1;
      addToTMtree(startTM_TtreeRef,
      startTM_TtreeRef.length+1,
endIndex,itemSet,support,index);
}
addToTMtree(linkRef,size,endIndex,itemSet,
                support, index) {
if (linkRef == null)
linkRef = new TM_TtreeNode[size];
for(i=1;i<linkRef.length;i++)
linkRef[i] = null;
currentAttribute = itemSet[endIndex];
if (linkRef[currentAttribute] == null)
linkRef[currentAttribute]=
       new TM_TtreeNode(numofDataSets,
         index,support);
if (endIndex == 0)
linkRef[currentAttribute].addSupportValue(i
ndex,support);
return(linkRef);
linkRef[currentAttribute].childRef
addToTMtree(linkRef[currentAttribute].
       childRef, currentAttribute,endIndex-
        1,itemSet,support,index);
return(linkRef);
}
```

Fig. 2 TM-TFP Algorithm

node represented a category of trend; the map was then trained and the remaining examples assigned to nodes using a distance function. 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 trends that are arithmetically possible 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 in which we might be interested. It was discovered that the second resulted in a map for which the majority of nodes were empty. The third option was therefore adopted, the SOM was trained using the trend lines associated with the first epoch. The resulting *prototype map* was then populated with data for all *e* epochs to produce a sequence of *e* maps. Figure 3 outlines the basic SOM algorithm.

3.3 Analysis of Trend Clusters

Change points in trend analysis can be interpreted in a number of ways. At its simplest, they may be interpreted as an abrupt change in direction of a trend line. A more complex interpretation may be the existence of changes in amplitude and/or frequency of fluctuating (seasonal) trends. Alternatively, an end user may be interested in an absence of change points. The interpretation applied to the cattle movement database is that we are interested in trends, associated with particular patterns, that change from epoch to epoch, i.e. are not consistent across the sampled temporal range. To this end, a simple cluster analysis technique was applied to identify trends that change location in the SOM associated with one epoch to the SOM associated with a subsequent epoch. The change can be measured by translating the trend line maps into a rectangular (D-plane) sets of coordinates and applying a Manhattan or Euclidean distance function to observe the similarities and differences of trends across the epochs. The greater the distance moved, the more significant the change.

```
t = current iteration
I = limit on time iteration
w = current weight vector
x = target data input
y = yearGenerate

Prototype vector map from data input of y
•Randomize the map's nodes' weight vectors, w.
•Grab an input vector, x.
•Traverse each node in the map
        •Use Euclidean distance formula to find similarity between the input
        vector and the map's node's weight vector
        •Track the node that produces the smallest distance (this node is the
        best matching unit, BMU)
•Adaptive process which updates the nodes in the neighbourhood of BMU by
pulling them closer to the input vector.
•Increment t and repeat from 2 while t < I

Generate Trend line maps
•Load labels (data input) of y
•For all data input, fit to BMU nodes on Prototype map
•Increment y and repeat from 1
```

Fig. 3 Basic SOM Algorithm

Thus, given a sequence of trend-line maps (SOMs) comparisons can be made to see how trends associated with individual frequent patterns change by analyzing the nodes in which they appear. The process may be described as follows:

1. Generate a matrix measuring $e \times k$ (k = number of frequent patterns).
2. Populate matrix with the node number for each pattern per *epoch*.
3. Calculate distance moved and store.
4. Identify movements above a given threshold.

4 Experimental Analysis Using The Cattle Movement Social Network

This, and the following, section presents an experimental analysis of the above described approach to trend mining. This section is directed at the cattle movement database which has provided the central focus of the work described. The following section considers a customer "database" so as to determine the potential benefits of the wider application of the proposed technique. This section commences with an overview of the cattle movement database and its transformation into a social network, followed by an analysis of the trend mining process as a applied to the generated network.

4.1 Cattle Movement Database

The CTS database records all the movements of cattle registered within or imported into GB. The database is maintained by the Department for Environment, Food and Rural Affairs (DEFRA). 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. The CTS was introduced in September 1998, and updated in 2001 to support disease control activities. Currently, the CTS database holds some 155 Gb of data.

The CTS database comprises a number of tables, the most significant of which are the animal, location and movement tables. For the analysis reported here the data from 2003 to 2006 was extracted to form 4 epochs 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 (epoch) and month (time stamp). The number of CTS records represented in each epoch 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 senders location in terms of easting and northing grid values, (v) the "type" of the sender's location, (vi) the receivers 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 between any pair of locations for a single time stamp was approximately 40 animals.

4.2 Cattle Movement Trend Mining

The TM-TFP algorithm was applied to the cattle movement social network and frequent pattern trends generated. For experimental purposes three support threshold values, 0.5%, 0.8% and 1.0% were used. Table 1 presents the number of frequent patterns trends discovered for each of the 4 epochs using the three support thresholds. As expected, the lower the support threshold used the greater the number of generated trends. Note also that the number of trends increases exponentially. An example of the nature of a frequent pattern, in the context of the cattle movement social network, is:

$$\{numberAnimalsMoved \leq 5, SenderPTI = 4, ReceiverArea = 54,\\ SenderLocationType = Agricultural\ Holding, SenderArea = 53,\\ AnimalAge \leq 1 year\ old\}$$

(the values for the *ReceiverArea* and *SenderArea* are Ordinance Survey grid square numbers). The associated sequence of support values (for 2003) representing the trend line for that year were:

$$[2391, 2609, 3218, 3009, 3890, 2759, 2298, 3124, 2911, 3331, 3791, 2417]$$

Table 1 Number of frequent pattern trends identified using TM-TFP for sequence of four cattle movement social network epochs and a range of support thresholds

Year	Support Threshold		
	0.5%	0.8%	1%
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

The generated trends were clustered using the SOM technique. The SOM was initializing with 7×7 nodes, and trained using the frequent pattern trends produced for the (earliest) 2003 year. The resulting prototype map is shown in Figure 4. Inspection of this map shows, for example, that node 1 (top-left) represents trend lines associated with patterns with higher support in spring (March to May) and autumn (September to November). Alternatively, node 43 (bottom-left) indicates trend lines with high support in spring only (March to April). Note that the distance between nodes indicates the dissimilarity between them; the greatest dissimilarity is thus between nodes at opposite ends of the diagonals. Once the initial prototype map has been generated, a sequence of trend line maps can be produced, one for each epoch.

Figure 5 gives the 2003 map. Note that in Figure 5, each node has been annotated with the number of trends in the "cluster" and that the "darker" trend lines indicate a greater number of trend lines within that cluster.

The cluster analysis mechanism highlighted interesting information beneficial to decision makers. Table 2 shows some example trends (representing frequent patterns) migrate from one cluster to another. Thus, the trend line representing the pattern $\{numberAnimalsMoved \leq 5, ReceiverPTI = NULL, ReceiverLocationType = Calf\,Collection\,Centre, SenderLocationType = Agricultural\,Holding, SenderArea = 14, AnimalAge \leq 1 year\,old, Gender = female\}$ was in node 49 (bottom right in Figure 5) in 2003 and 2004, but then migrated to node 48 in 2005 and disappeared in 2006.

Table 2 Examples of CTS Frequent Patterns migrating from one SOM node to another

Frequent Pattern Code	Node 2003	Dist	Node 2004	Dist	Node 2005	Dist	Node 2006
{441 436 329 301 213 4 3}	49	0	49	1	48	0	0
{441 436 329 301 213 196}	48	1	49	4.1	38	3.2	48
{378 301 263}	39	0	39	3.2	49	3.2	39
{441 329 214}	47	2	49	0	0	0	49

Using the above cluster analysis technique decision makers can "focus in" on particular types (clusters) of trends. In terms of further reducing the overall number of trend lines this can be achieved by considering only a subset of the detected frequent patterns according to particular attributes of interest. The term *meta-pattern* is introduced to represents a way of considering groups of patterns. In the context

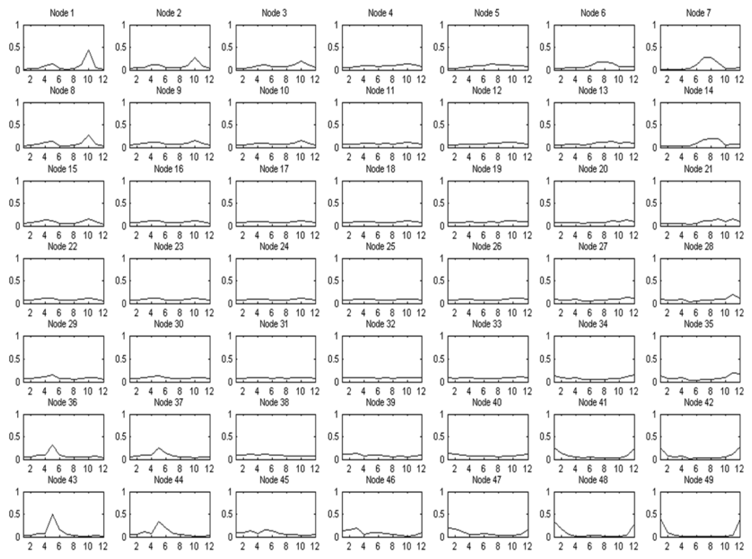


Fig. 4 CTS prototype map

of the cattle movement social network, we are interested in patterns that include spatial information (i.e. sender and receiver locations). Four categories of meta-pattern were therefore identified:

1. **Movement from start points**: patterns that include movement and sender attributes/columns.
2. **Movement to end points**: patterns that include movement and receiver attributes/columns.
3. **Movement from start to end points**: patterns that include movement and both sender and receiver attributes/columns.
4. **Movement for other non spatial attributes**: patterns which do not feature the above.

Meta-patterns form smaller groups of patterns for cluster and trend analysis thus simplifying the cluster analysis task.

5 Car Insurance Trend Mining

The above described technique has application with respect to alternative types of data. For example, if we consider a standard time stamped, tabular data set, we can identify trends in this data in the same manner as described for the cattle movement social network. This is illustrated in this section by considering a car insurance quote data set, the Deeside data set [1]. The data can be viewed as representing a "star" net-

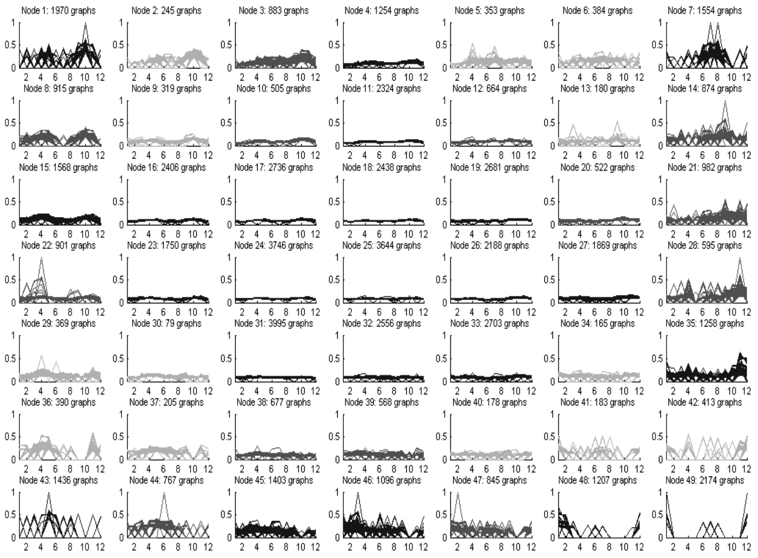


Fig. 5 CTS Map for 2003 frequent pattern trends

[1] The data set was provided by Deeside Insurance Ltd, Deeside, UK.

work with Deeside at the center as a *super node* and all other nodes radiating out from it. The outlying nodes represent geographical locations defined by the first characters of customer postcodes. The links are labelled with the number of interconnections between individual geographic locations and the center. The data set was partitioned into monthly time stamps and two epochs (2008 and 2009). Each month comprises some 1000 records. Each record consists of 13 attributes: (i) Aggregator [2], (ii) year of insurance contract, (iii) customer gender, (iv) make of car, (v) car engine size, (vi) year of manufacture, (vii) customer postcode, (viii) driver age (ix) conviction code, (x) conviction code number (xi) length of disqualification, (xii) fault and (xiii) penalty (note that the value for some of the attributes is *null*).

Table 3 Number of frequent pattern trends identified using TM-TFP for sequence of two Deeside Insurance epochs and a range of support thresholds

Year	Support Threshold		
	2%	3%	5%
2008	314,471	142,175	55,241
2009	284,871	122,371	49,983

Table 3 presents the number of trends generated by applying TM-TFP to the Deeside data set using a range of support thresholds of 2%, 3% and 5% respectively. Note that lower support thresholds were used than in the case of the CTS dataset because the Deeside data was smaller. The results presented in Table 3 corroborate those presented previously in Table 1. An example of a frequent pattern found in the Deeside data is:

$$\{Fault = NoBlame, LengthOfDisqualify \leq 5, Age \leq 50,$$
$$PostCodeArea = CH, CustomerGender = female\}$$

The associated sequence of trend line values (for 2008) were:

$$[23, 0, 31, 18, 0, 4, 0, 7, 25, 9, 16, 19]$$

A 7×7 SOM was again used and trained using the 2008 data. The prototype map is presented in Figure 6. From the figure it can be seen, for example, that node 1 indicates a trend line with high support mainly in February, whilst node 7 shows a trend line with high support mainly in March. It is interesting to note that there are more identified patterns in the first and last quarters of the year. The prototype map was then populated with the 2008 and 2009 data to produce a sequence of two maps that could be compared. Comparison of clusters allowed for the identification of changes in customer "quote request" habits. Table 4 presents some examples of trend migrations identified from within the Deeside Insurance data set. For example, the trend line representing the pattern $\{310, 286, 283, 145\}$ which translates to $\{Fine \leq 1000, ConvictCode = SP, 41 \leq DriverAge \leq 50, 1996 \leq CarYearManufacture \leq 2000\}$ which was in node 43 (bottom right in Figure 6) in

[2] An aggregator is a web application or search facility that allows users to obtain and compare a number of insurance quotes/prices.

2008 migrated to node 11 in 2009. This signifies that the pattern has changed from a trend with high support in September to the trend with high support in February and March.

Table 4 Example of Deeside Insurance Frequent Patterns that migrated to other clusters

Frequent Patterns	Node 2008	Dist	Node 2009
{310 286 283 145}	43	5.8	11
{310 286 283 145 1}	44	6.3	4
{310 286 283 146}	36	4.2	18
{310 286 283 146 1}	35	2.2	20

6 Conclusion

A social network trend mining mechanism has been described, founded on frequent pattern mining, SOM clustering and cluster analysis. The mechanisms were demonstrated using two applications: a social network derived from the CTS database, and a “star” network derived from the Deeside Insurance data. The analysis demonstrates that the mechanisms may be usefully employed to identify changes in trends discovered in the networks. TM-TFP is able to generate frequent time stamped patterns which can be sub-divided into epochs which may then be compared. By employing the SOM clustering technique, the large number of trend lines that are typically identified may be grouped to facilitate a better understanding of the nature of the trends. Using the proposed cluster comparison/analysis technique, trend migrations can be discovered. The research team is currently developing further methods in which change detection and visualization of the clustering result can be more effective with respect to the requirements of decision makers and stakeholders.

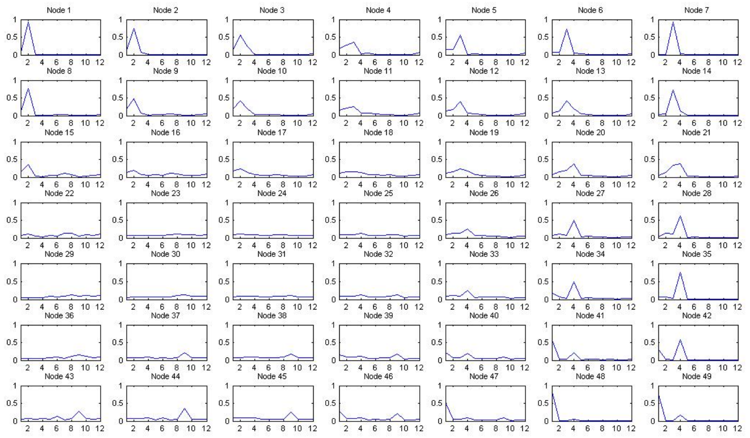


Fig. 6 Deeside Insurance prototype map

References

1. Google Trends. http://www.google.com/intl/en/trends/about.html
2. Streibel, O.: Trend Mining with Semantic-Based Learning. Proceedings of CAiSE-DC (2008)
3. Khan, M.S., Coenen, F., Reid, D., Tawfik, H., Patel, R., Lawson, A.: A Sliding Windows based Dual Support Framework for Discovering Emerging Trends from Temporal Data. Research and Development in Intelligent Systems XXVII, Springer London, pp 35-48 (2010)
4. Raza, J. and Liyanage, J. P.: An integrated qualitative trend analysis approach to identify process abnormalities: a case of oil export pumps in an offshore oil and gas production facility. Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering, Professional Engineering Publishing, vol 223 (4), pp 251-258 (2008)
5. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press (2006)
6. Lauw, H., Lim, E., Pang, H., Tan T.: Social Network Discovery by Mining Spatio-Temporal Events. Computational Mathematical Organization Theory, vol 11(2), pp. 97-118. Springer Netherlands (2005)
7. Agrawal, R., Imielinski, T., and Swami, A. Mining Association Rules between Sets of Items in Large Databases. In Proceedings of ACM SIGMOD Conference (1993)
8. Agrawal, R. andSrikant, R.: Mining sequential patterns. 11th International Conference on Data Engineering (1995)
9. Mannila, H., Toivonen, H., and Verkamo, A.: Discovery of Frequent Episodes in Event Sequences. Data Mining and Knowledge Discovery 1, pp 259289(1997)
10. Dong, G., and Li, J.: Efficient Mining of Emerging Patterns: Discovering Trends and Differences. In Proceeding of fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (1999)
11. Coenen, F.P., Goulbourne, G., Leng, P.: Computing Association Rules Using Partial Totals. Principles of Data Mining and Knowledge Discovery. LNCS, vol. 2168, pp. 54-66. Springer Berlin / Heidelberg (2001)
12. Kohonen, T.: The Self Organizing Maps. Neurocomputing Elsevier Science, vol. 21, pp. 1-6 (1998)
13. Kohonen, T.: The Self Organizing Maps. Series in Information Sciences, vol. 30. Springer, Heidelberg (1995)
14. Wang, J., Delabie, J., Aasheim, H.C., Smel, E., Myklebost, O.: Clustering of the SOM easily reveals distinct gene expression patterns: results of a reanalysis of lymphoma study. BMC Bioinformatics, vol 3(36) (2002)
15. Yan, S., Abidi, S.S.R, Artes, P.H.: Analyzing Sub-Classifications of Glaucoma via SOM Based Clustering of Optic Nerve Images. Studies in Health Technology and Informatics, vol 116 pp 483-488 (2005)
16. Cottrell, M., Rousset, P.: A powerful Tool for Analyzing and Representing Multidimensional Quantitative and Qualitative Data. In Proceedings of IWANN 97. LNCS, vol. 1240, pp. 861-871. Springer Berlin / Heidelberg (1997)
17. Kohonen, T., Oja, E., Simula, O., Visa, A., Kangas, J.: Engineering applications of the Self-Organizing Map. Proceedings of the IEEE, vol. 84(10), pp. 1358-1384 (1996)
18. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press (2006)
19. Lingras, P., Hogo, M. and Snorek, M.: Temporal Cluster Migration Matrices for Web Usage Mining. In Proceedings of IEEE/WIC/ACM InternationalConference on Web Intelligence (2004)
20. Denny, Williams, G.J and Christen, P.: ReDSOM: relative density visualization of temporal changes in cluster structures using self-organizing maps. IEEE International Conference on Data Mining (ICDM), IEEE Computer Society, pp 173-182 (2008)
21. Hido, S., Id T., Kashima, H., Kubo H. and Matsuzawa, H.: Unsupervised changes analysis using supervised learning. Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference. PAKDD. LNCS, vol. 5012, pp 148-159 (2008)