Finding Frequent Subgraphs in Longitudinal Social Network Data Using a Weighted Graph Mining Approach

Chuntao Jiang, Frans Coenen, and Michele Zito

Department of Computer Science,
University of Liverpool, Ashton Building,
Ashton Street, L69 3BX Liverpool, UK
{c.jiang,coenen,zito}@liverpool.ac.uk

Abstract. The mining of social networks entails a high degree of computational complexity. This complexity is exacerbate when considering longitudinal social network data. To address this complexity issue three weighting schemes are proposed in this paper. The fundamental idea is to reduce the complexity by considering only the most significant nodes and links. The proposed weighting schemes have been incorporated into the weighted variations and extensions of the well established gSpan frequent subgraph mining algorithm. The focus of the work is the cattle movement network found in Great Britain. A complete evaluation of the proposed approaches is presented using this network. In addition, the utility of the discovered patterns is illustrated by constructing a sequential data set to which a sequential mining algorithm can be applied to capturing the changes in "behavior" represented by a network.

Keywords: Frequent subgraph mining, Weighted graph mining, Social network mining, Longitudinal data.

1 Introduction

Social Network Mining (SNM) [1] is a sub-field of research within the context of Knowledge Discovery in Data (KDD). A social network comprises a set of nodes that represent entities and links that represent some form of communication between such entities. The entire network is not necessarily connected and usually includes some *super-nodes* to which many other nodes are connected. Popular examples of social networks include social networking sites such as FaceBook[1] and Flickr[2]. However, there are many other forms of social networks such as information sharing networks and co-authoring networks. Many of these networks, such as co-authoring networks, are derived from tabular data. In the case of co-authoring networks [2] these can be derived from web applications such as DBLP[3] and CiteSeer[4]. The focus of the work described in this paper is the

[1] http://www.facebook.com
[2] http://www.flickr.com
[3] http://www.informatik.uni-trier.de/ ley/db/
[4] http://citeseerx.ist.psu.edu/

L. Cao, J. Zhong, and Y. Feng (Eds.): ADMA 2010, Part I, LNCS 6440, pp. 405–416, 2010.

Cattle Tracking System (CTS) database, in operation in Great Britain, from which a cattle movement network can be extracted. However, the techniques described are generally applicable.

SNM is directed at the identification of patterns within social networks networks. The nature of the patterns can take many forms, a common type of SNM is the identification of clusters of frequently "corresponding" nodes. SNM is usually applied in a static context, i.e. a "snap shot" of the network is taken to which SNM is then applied. In this paper we are interested in applying SNM techniques to sequences of such snap shots where each snap shot is time stamped in someway. We refer to these sequences as *longitudinal social networks*, in that the sequences may be compared to longitudinal data collections such as those found in medical applications [3]. More specifically we are interested in finding frequently occurring subgraphs in longitudinal social network data.

Social network data is often substantial, usually comprising many nodes and links. Consequently, longitudinal social network data tends to be even more substantial, typically an order of magnitude relative to the number of time stamps to be considered. Standard frequent subgraph mining algorithms, such as gSpan [4], are thus unable to process such large longitudinal networks in a realistic manner. In recognition that some links (and/or nodes) may be considered to be more important than others, in this paper weighted longitudinal social network mining is proposed. Weightings can be applied in a number of manners, three weighting schemes are proposed and evaluated in this paper.

The rest of this paper is structured as follows. Some previous work and a problem definition are presented in Sects 2 and 3 respectively. Three proposed weighting schemes to achieve longitudinal SNM are presented in Sect 4. The suggested weighting schemes are evaluated and compared in Sect 5. Some conclusions and main findings are presented in Sect 6.

2 Previous Work

Examples of research work on applying frequent subgraphs to analyzing social networks include: community pattern mining [5], targeted advertising [6], and structural prediction [7]. Frequent subgraph mining [8,9,4] entails two significant overheads: candidate set generation and (sub)graph isomorphism checking. However, these overheads are exacerbated when the size of the graph data is substantial and the support threshold is low. Weighted frequent subgraph mining [10] advocates the use of *weighted support counts* to identify weighted frequent subgraphs. Hence, the "computational burden" of subgraph mining can be considerably alleviated by generating a set of weighted frequent subgraphs.

3 Problem Definition

This section provides the necessary longitudinal social network representation and mining definitions

Definition 1. *A longitudinal social network is comprised of a sequence of graphs* $N_G = \{G_1, G_2, \cdots, G_T\}$*, where* G_t *is a graph corresponding to the social network representation at time period of* $t \in [1, T]$*. Let* $V = \bigcup_{t=1}^{T} V_t$ *denote the whole set of entities for the network. Each entity* $v \in V$ *is uniquely labeled, and each* v *can appear only once at each time-step.*

Definition 2. *For any arbitrary subgraph* $g = (V', E')$ *such that* $V' \subseteq V$*, the support set of* g *is defined as* $\delta(g) = \{t | g \subseteq G_t\}$*, e.g., the set of time-steps for which* g *is a subgraph of* G_t*. The support of* g *with respect to* N_G*,* $sup(g)$*, is defined to be the cardinality of the support set,* $|\delta(g)|$*.*

Definition 3. *A subgraph* g *is frequent with respect to* N_G *if* $|\delta(g)| \geq \tau \times T$*, where* $0 < \tau \leq 1$ *is a minimum support threshold. The frequent subgraph mining problem is thus to find all frequent subgraphs in* N_G*.*

4 Weighted Frequent Subgraph Mining

Most research work in frequent subgraph mining [8,9,4], assumes each discovered frequent subgraph is equally important. Because of this, a lot of redundant and repetitive frequent patterns exist in the resultant set. In addition, if the size of the graphs is substantial and the minimum support is low, a typical frequent subgraph mining task will not terminate within a reasonable period of time due to the exponential computation incurred by subgraph isomorphism testing. If we put emphasis on differentiating each discovered frequent subgraph by their importance as defined by the user, or as derived from the application domain, a reduced computational complexity can be achieved without compromising the effectiveness of the discovery process.

Therefore, the graphs in N_G are assumed to have weights associated with their nodes or links. Let $W(g)$ be a weighting function that assigns a weight to each discovered subgraph g. The *weighted support* of g with respect to N_G, is then defined as $wsup(g) = W(g) \times |\delta(g)|$.

When a weighting function is integrated into the process of mining weighted frequent subgraphs, the well-known *anti-monotone property*[5], which is often used to prune the search space of the patterns, is not satisfied anymore. There are two general solutions to this dilemma: (i) design a weighting function that keeps the property; (ii) utilize some heuristics to reduce the computation incurred by not maintaining the property.

In the context of weighted frequent subgraph mining, the weighting function associated with a subgraph pattern g can be defined in various manners. Three example approaches are proposed in this paper: (i) Average Mutual Information Based Weighting (AMW), (ii) Affinity Weighting (AW), and (iii) Utility Based Weighting (UBW). The first two approaches satisfy the anti-monotone property while the last one adopts an alternative pruning heuristic. The last two approaches employ two parameters to control the mining result while the first one uses one parameter only. Each approach is discussed in further detail in the following three sub-sections.

[5] If a graph is infrequent, then all its supergraphs are infrequent.

4.1 Average Mutual Information Based Weighting (AMW)

In the AMW approach, the weight for a subgraph g is calculated by dividing the sum of the average weights in graphs that contain g with the sum of the average weights across the entire data set T. Thus:

Definition 4. *Given a node weighted graph g with node weights* $\{w_1,w_2,\cdots,w_k\}$, *the average weight associated with g is defined as* $W_{avg}(g) = \frac{\sum_{i=1}^{k} w_i}{k}$.

Where w_i can be user defined or calculated by some weighting methods. In this paper, a weighting method to generate the node weight, is introduced as follows:

Definition 5. *Given a subgraph* $g = \{e_1, e_2, \cdots, e_k\}$, *for each link* e_i *with two connecting nodes* v_a *and* v_b, *their corresponding support values are* $sup(e_i)$, $sup(v_a)$, *and* $sup(v_b)$. *The mutual information between two nodes,* $PMI(e_i)$, *is then defined as* $PMI(e_i) = \log\left(\frac{sup(e_i)}{sup(v_a)\times sup(v_b)}\right)$. $PMI(e_i) = 0$, *when* $sup(v_a) = 0$ *or* $sup(v_b) = 0$.

Definition 6. *Given a subgraph g, with* $V(g) = \{v_1, v_2, \cdots, v_m\}$ *and* $E(g) = \{e_1, e_2, \cdots, e_k\}$, *the weight for node* v_i, w_i *is defined as:*

$$w_i = \frac{\sum_{i=1}^{k} PMI(e_i)}{deg(v_i) - 1} \quad . \tag{1}$$

Where $deg(v_i)$ denotes the number of links incident to v_i, if $deg(v_i) = 1$ or 0, $w_i = 0$.

Definition 7. *Given a set of graphs* $N_G = \{G_1, G_2, \cdots, G_T\}$, *the total weight of this set of graphs is defined as* $W_{sum}(N_G) = \sum_{i=1}^{T} W_{avg}(G_i)$.

Definition 8. *Given an arbitrary subgraph g with its support set* $\delta(g)$, *the weighting function of g with respect to* N_G, $W_{N_G}(g)$, *is defined as:*

$$W(g) = \frac{\sum_{G_i \in \delta(g)} W_{avg}(G_i)}{W_{sum}(N_G)} \quad . \tag{2}$$

Definition 9. *A subgraph g is weighted frequent with respect to* G_N, *if* $|\delta(g)| \times W(g) \geq \tau \times T$, *where* $0 < \tau \leq 1$ *is a minimum support threshold.*

From the above it can be easily inferred that the function $W(g)$, as defined by (2), satisfies the anti-monotone property. Therefore, if a k-subgraph candidate is not frequent, then any of its $(k+1)$-supergraph candidates can be safely pruned from this branch in the search space lattice during the $k+1$ candidate generation process. It should be noted, however, that the approach will tend to bias large transaction graphs over smaller transaction graphs, thus is best applied to graph sets where the individual graphs are of a similar size.

4.2 Affinity Weighting (AW)

The AW approach is founded on two elements to restrict the growth of the search space: (i) a graph distance measure, and (ii) a weighting ratio. For a subgraph g to be frequent both elements must be greater than specified user thresholds. The graph distance measure is calculated using an appropriately defined support weighting function, $W(g)$. This is defined as follows. Let g be a candidate pattern for a database $N_G = \{G_1, G_2, \cdots, G_T\}$. In the context of AW we define:

$$W(g) = \frac{1}{|V(g)|} \sum_{G_i \in \delta(g)} \frac{|V(G_i)| - |V(g)|}{|V(G_i)|} \quad . \tag{3}$$

Where $V(G_i)$ is the set of nodes in transaction graph G_i and $V(g)$ is the set of nodes in the subgraph g. Observe that $W_T(g)$ satisfies:

$$W(g) = \frac{|\delta(g)|}{|V(g)|} - \sum_{G_i \in \delta(g)} \frac{1}{|V(G_i)|} \quad . \tag{4}$$

It should be noted that adding nodes to g can only reduce the value of the above expression because the support($|\delta(g)|$) cannot be increased; the sum contains as many terms as $|\delta(g)|$ and each of these cannot be larger than $1/|V(g)|$. Thus $W(g)$ as defined above, insures that the weighted support of g is non-increasing (i.e. anti-monotone) in $|V(g)|$.

The graph distance measure is directed at the number of nodes contained in a graph, the weighting ratio concerned with the link weights. The weighting ratio of an link-weighted graph g is a function $c(g)$ returning a value between zero and one which is decreasing in the number of links of g. Given an link weighted subgraph g with link weights $W = \{w_1, w_2, \cdots, w_k\}$ the weighting ratio function, $c(g)$, is defined as follows:

$$c(g) = \frac{MIN_{w_i \in W}\{w_i\}}{MAX_{w_j \in W}\{w_j\}} \quad . \tag{5}$$

Definition 10. *An link-weighted graph g is a weighted frequent (i.e. weighted affinity) pattern within a data set $N_G = \{G_1, G_2, \cdots, G_T\}$, with respect to a support threshold $\tau > 0$ and weighting ratio threshold $\gamma \in [0, 1]$, if the following two conditions (C1 and C2) are satisfied:*

$$\textbf{(C1)}\ wsup(g) \geq \tau \times T, \qquad \textit{and} \qquad \textbf{(C2)}\ c(g) \geq \gamma \quad .$$

Definition 10 leads to an alternative pruning strategy which, may be used as part of any frequent subgraph mining algorithms. During the candidate selection phase, the mining will keep track of the weighted support and weighting ratio of all candidates and discard all those candidates that do not satisfy at least one of **(C1)** and **(C2)**.

4.3 Utility Based Weighting (UBW)

The previous two approaches both satisfy the anti-monotone property. In this section an alternative weighting scheme which does not feature the property is proposed, instead an alternative mechanism for limiting growth is adopted. The UBW scheme is influenced by ideas suggested in [11]. As in the case of the AW scheme, the UBW scheme is founded on two elements: (i) weighted support and (ii) the share (SH) of a subgraph. Thus:

Definition 11. *Given a subgraph g with links $E(g) = \{e_1, e_2, \cdots, e_k\}$. For each $e_i \in E(g)$, two nodes connecting e_i are v_1 and v_2. Their associated support sets are given as $\delta(v_1)$ and $\delta(v_2)$. The Jaccard similarity coefficient between the two nodes is defined as $jC(e_i) = |\delta(v_1) \cap \delta(v_2)| / |\delta(v_1) \cup \delta(v_2)|$. The weighting function of g, $W(g)$, is then defined as*

$$W(g) = \frac{1}{\sum_{e_i \in E(g)} jC(e_i)} \quad . \tag{6}$$

From the above it is clear that $W(g)$ satisfies the anti-monotone property.

Definition 12. *Given an link weighted graph set $N_G = (G_1, \ldots, G_T)$ with link weights $\{w_1, w_2, \cdots, w_k\}$ for each transaction graph G_t and a subgraph g. Let $g \subseteq G_t$, the weight of g denoted as $W(g, G_t)$, is the sum of the weights of the links which occurred in G_t. That is, $W(g, G_t) = \sum_{e_i \in g, g \subseteq G_t} w_i$. The total weight of N_G, denoted as $TW(N_G)$, represents the sum of link weights in N_G, where $TW(N_G) = \sum_{G_t \in N_G} \sum_{e_i \in G_t} w_i$. The total weight of $\delta(g)$, is defined as $TW(\delta(g)) = \sum_{G_t \in \delta(g)} \sum_{e_i \in G_t} w_i$.*

Definition 13. *The graph weight of g with respect to N_G, denoted as $GW(g)$, is the sum of the weight of the g in each transaction graph $G_t \in \delta(g)$. That is, $GW(g) = \sum_{G_t \in \delta(g)} W(g, G_t)$.*

Definition 14. *The share of a subgraph g, denoted as $SH(g)$, is the ratio of the graph weight of g with respect to N_G to the total weight of N_G. Thus:*

$$SH(g) = \frac{GW(g)}{TW(N_G)} \quad . \tag{7}$$

Given a share threshold λ, a subgraph g is SH-frequent if $SH(g) \geq \lambda$; otherwise, g is SH-infrequent.

Theorem 1. *Given a $N_G = (G_1, \ldots, G_T)$, a subgraph g, and a threshold λ, if $TW(\delta(g)) < \lambda \times TW(N_G)$, all supergraphs of g are SH-infrequent.*

Proof. Let h be an arbitrary supergraph of g. Clearly, $GW(h) \leq TW(\delta(h)) \leq TW(\delta(g))$. If $TW(\delta(g)) < \lambda \times TW(N_G)$ holds, $GW(h) < \lambda \times TW(N_G)$. That is, $SH(h) = GW(h)/TW(N_G) < \lambda$. Therefore, h is SH-infrequent. □

By Theorem 1, if $TW(\delta(g)) < \lambda \times TW(N_G)$, all supergraphs of g and g are SH-infrequent and can be pruned; otherwise, g is a candidate subgraph.

Definition 15. *An link-weighted graph g is a weighted frequent pattern for a graph set $N_G = (G_1, \ldots, G_T)$ with respect to a support threshold $\tau > 0$ and share threshold $\lambda \in (0, 1]$ if the following two conditions are satisfied.*

$$\textbf{(D1)}\ wsup(g) \geq \tau \times T, \qquad \textit{and} \qquad \textbf{(D2)}\ SH(g) \geq \lambda \quad .$$

5 Evaluation

To evaluate the proposed weighting schemes experiments were conducted using a projection of the Cattle Tracking System (CTS) database in operation in Great Britain (GB). The proposed weighting schemes were incorporated into weighted variations and extensions of the well established gSpan frequent subgraph mining algorithm [4]. For comparison purposes we also derived variation of gSpan, *extGspan*; that did not feature weightings, but could handle directed graphs, self-cycles and multiple links. Results from the experiments are presented in the following sub-sections.

Table 1. The statistics of graph data generated by the projection of the CTS database

	Derbyshire	Lancashire	*GB*
# graphs	53	53	53
Max # links	179	394	30107
Average # links	137	297	23055
Max # nodes	227	396	23660
Average # nodes	172	318	18749

5.1 The Cattle Tracking System Database

The Cattle Tracking System (CTS) database in operation in Great Britain (GB), which forms the focus of the research described in this paper, is maintained by the Department for Environment, Food and Rural Affairs (DEFRA) as part of the Rapid Analysis and Detection of Animal-related Risks (RADAR) initiative[6]. The CTS database records all cattle movements in GB, each record describes the movement of a single animal, identified by a unique ID number, between two *holding locations* (e.g. agriculture holdings, markets, etc). Social network can be extracted from this database such that each node represents a geographical location and the links the number of animals moved between locations. By considering the time stamps associated with movements, temporal sequences of networks can be extracted (i.e. longitudinal social networks).

For the experimental analysis three distinct longitudinal social network data sets were extracted from the CTS database using data from 1 January 2005 to 31 December 2005. The first two data sets, Derbyshire[7] and Lancashire[8].

[6] http://www.defra.gov.uk/foodfarm/farmanimal/diseases/vetsurveillance/radar

[7] *Derbyshire* is a county in the East Midlands of England.

[8] *Lancashire* is a county in the North West of England.

The third data set represented GB in its entirety. The data divided into 7-day "episodes" (there is a 6-day movement restriction that applies to agriculture holdings in GB), giving longitudinal sequences of 52 episodes. The links were annotated with a weight, indicating the number of animals that were moved, and a label, indicating the type of movement (e.g. farmToFarm, farmToMarket, etc). Some statistics for each of the data sets is presented in Table 1. Note that the *GB* database is significantly larger than the *Lancashire* data set, which in turn was larger than the *Derbyshire* data set. Note that all the graphs featured "self-cycles" and "multiple links".

5.2 Comparison between Weighted and Non-weighted Approaches

In this sub-section the proposed weighting schemes (AMW-gSpan, AW-gSpan, and UBW-gSpan) are compared with the extended gSpan algorithm in terms of efficiency (runtime and the number of frequent subgraphs generated). For AW-gSpan, $\gamma = 0.6$ was chosen as the weighgting ratio threshold, and $\lambda = 8\%$ was used as the share threshold for UBW-gSpan. The justification for these γ and λ values is given in Sect 5.3 below.

Figure 1 shows the performance of the weighting schemes and extGspan on the *Derbyshire* and *Lancashire* data sets. It can be clearly seen from the figure that all four algorithms display a similar behavior when the support value is between 12% to 20%, however the number of patterns generated by the extGspan algorithm increases abruptly when the support value is decreased to below 12%. In the figure, it can be observed that: (i) significantly more frequent subgraphs are found using UBW algorithm than using any of the others, indicating that UBW algorithm can not show its advantage against the non-weighted extGspan

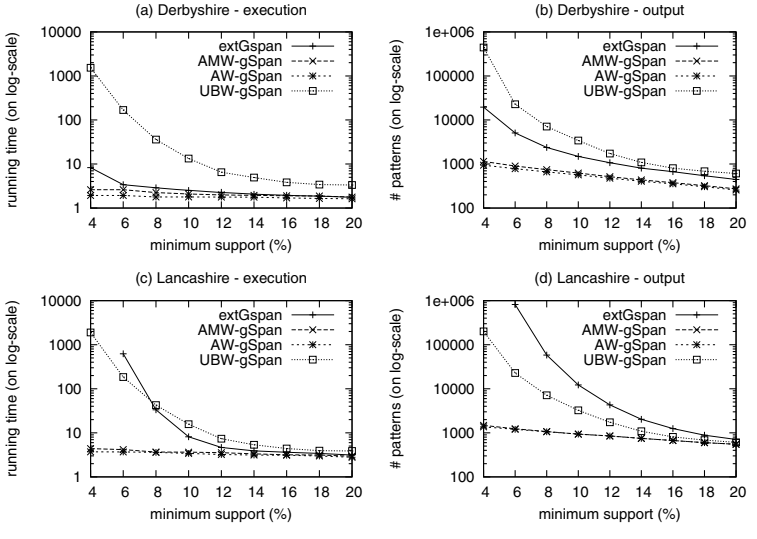

Fig. 1. Performance comparison of weighting schemes vs. extGspan on two graph sets

for a smaller data set, (ii) Two weighting schemes: AW and AMW algorithms achieves better performance than extGspan and UBW algorithms. The reason for the better performance of AMW and AW schemes is that the pruning technique adopted by UBW scheme is not sufficiently effective in reducing the search space when compared to the anti-monotone method used by the AMW and AW schemes.

Experiments (not shown) using extGspan and the *GB* data set failed to produce any results (because of memory errors) unless the support threshold was set to 30% or above, a threshold at which only one node size subgraphs are discovered. Thus it was not possible to conduct any meaningful comparison between the weighted frequent subgraph mining algorithms and a non-weighted approach using the *GB* data set.

5.3 Comparison of Weighting Schemes

In this sub-section the three proposed weighting schemes are compared with one another using the large *GB* data set. As above, γ was initially set to 0.6 and λ to 8% for use with AW-gSpan and UBW-gSpan algorithms. Figure 2 shows the performance of the weighting schemes on the *GB* data set. In Fig. 2 (a), each curve depicts the number of patterns generated against the minimum support value used. From the figure it can be seen that UBW-gSpan produces the least number of patterns while AW-gSpan produces the most. Figure 2 (b) indicates the "run time" for the approaches using the same sequence of support threshold values. From the figure it can be seen that UBW-gSpan is the most "expensive", indicating that the cost of finding a minimum number of patterns is higher compared to the other two mechanisms. AMW-gSpan is the most economical. It is interesting to note in Fig. 2 (b) that as the support threshold is reduced the effect on run-time is much smaller for AMW-gSpan than the other two weighting schemes.

Figure 3 displays the effect on performance of different values for the weighting ratio threshold (γ) used in conjunction with AW-gSpan, and the share threshold (λ) used with UBW-gSpan, for a range of support threshold values from 4% to 12%. From Fig. 3 (a) and (c) it can be seen that the run time increased as the γ value is decreased, while a marginal increase in the number of patterns is

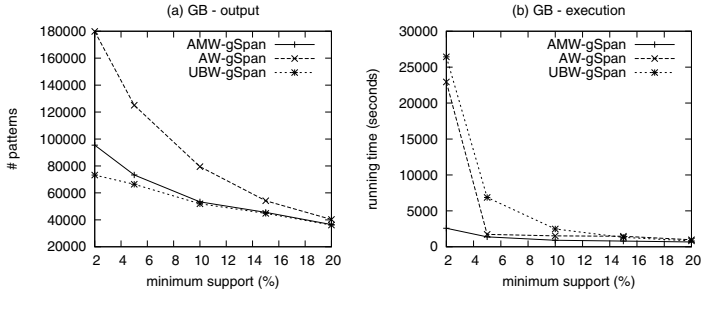

Fig. 2. Performance comparison of using three weighting schemes on the *GB* data set

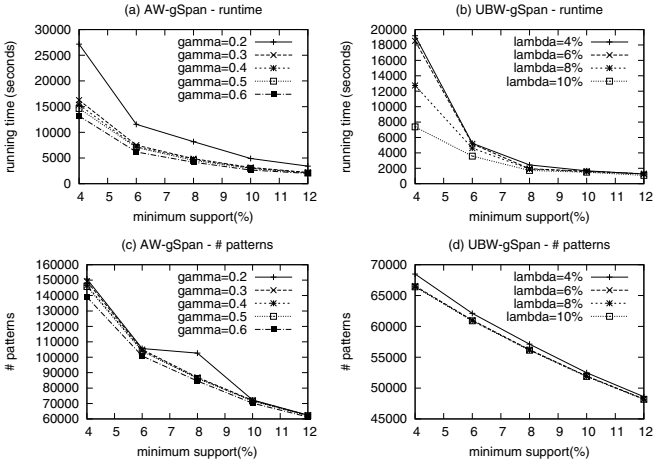

Fig. 3. Analysis of the performance of AW-gSpan and UBW-gSpan mining algorithms

witnessed. With respect to Fig. 3 (b) and (d) it can be seen that the run time increases as the λ value is decreased, while a small corresponding increase in the number of identified patterns is witnessed. However, increasing the λ value beyond 8% seems to have very little effect on the number of patterns. Overall it was found that a γ value of 0.6 and a λ value of 0.8% was the most appropriate.

5.4 Subgraph Pattern Analysis

To demonstrate that the the utility of the subgraphs that have been discovered this sub-section briefly discusses an application of the approach. The frequent subgraphs identified by the above mining algorithms can be further used to construct a sequential database where each item of the sequence is a frequent subgraph. Formally, each sequential transaction is extracted using the following identity (8).

$$S_{t,t+ts} = \{f_i | f_i \in (FS(G_t) \cup FS(G_{t+ts}) - FS(G_t) \cap FS(G_{t+ts}))\} \quad . \qquad (8)$$

where $S_{t,t+ts}$ represents the sequence of frequent subgraphs in a ts time period, f_i represents a frequent subgraph and $FS(G_t)$ represents all frequent subgraphs of the graph G_t, where G_t represents the graph at a time instance t.

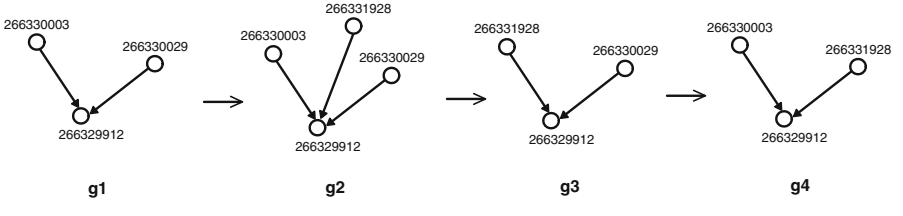

Fig. 4. An example of the sequential patterns extracted using AMW-gSpan algorithm

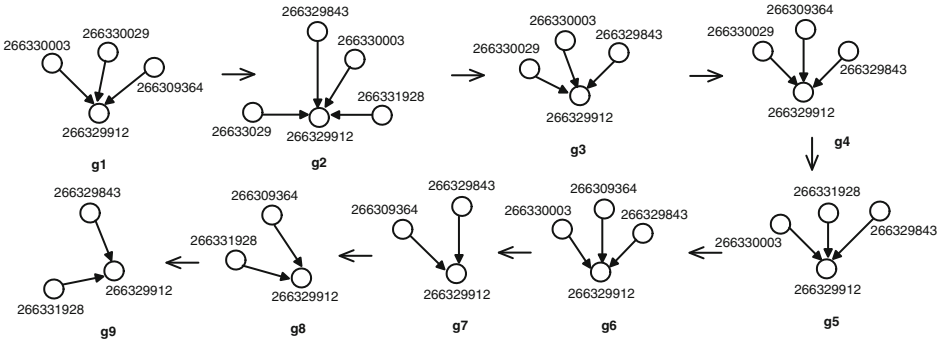

Fig. 5. Another example of a longer sequential pattern extracted using AMW-gSpan

Using the AMW-gSpan algorithm as an example; the output when applied to the *GB* data set, with a support of 10%, was utilized to create a sequential data collection with a time-step value of 1. A sequential mining algorithm, PrefixSpan [12], was then applied to this database with a support threshold of 60%. One example of the maximal-size sequential patterns is displayed in Fig. 4. The patterns in the figure indicate these four subgraphs occurred together on 32 occasions out of 52 in the order showed in the figure. In the figure, each node denotes the location of the agriculture holding, and the number next to the node denotes the unique identification number. It can be seen that $g1$, $g3$, and $g4$ are all subgraphs of $g2$. All the movements are pointed to the location "266329912", and $g1$ always occurs before $g2$, while $g3$ and $g4$ always occur after $g2$. If a smaller support threshold of 30% was used, a longer sequence consisting of 9 patterns were extracted as illustrated in Fig. 5. Figure 5 features additional movements to those given in Fig. 4, and includes new locations. In the figure, the movement was still centered on the location "266329912", however two new locations "266329843", and "266309364" were added into the sequence, and each pattern in the figure contains either one of them or both.

6 Conclusions

A weighted approach to longitudinal social network mining is described. The approach allows large longitudinal networks, such as the GB network used to illustrate this paper, to be mined where this was not possible using more conventional approaches. Three weighting mechanisms were proposed to reduce the overall computational complexity. Reported experiments comparing the operation of the weighting schemes with each other and a non-weighted version of gSpan demonstrated that many fewer patterns are derived. The reported experiments also indicated that UBW-gSpan finds the least number of patterns while requiring the largest amount of run-time. AMW-gSpan provided the best compromise, a limited number of patterns found in reasonable time (especially at low support threshold values). To illustrate that the utility of the subgraphs

that were discovered; further analysis was conducted to capturing changes in "behavior" within the network structures.

References

1. Wasserman, S., Faust, K.: Social Network Analysis, Method and Applications. Cambridge University Press, New York (1994)
2. Barabsi, A.L., Jeong, H., Nda, Z., Ravasz, E., Schubert, A., Vicsek, T.: Evolution of the Social Network of Scientific Collaborations. Physica A: Statistical Mechanics and Its Applications 311, 590–614 (2002)
3. Somaraki, V., Broadbent, D., Coenen, F., Harding, S.: Finding Temporal Patterns in Noisy Longitudinal Data: A Study in Diabetic Retinopathy. In: Proceedings of the 10th Industrial Conference on Data Mining, Berlin, pp. 418–431 (2010)
4. Yan, X., Han, J.: gSpan:Graph-based Substructure Pattern Mining. In: Proceedings of 2002 International Conference on Data Mining (2002)
5. Mukherjee, M., Holder, L.B.: Graph-based Data Mining on Social Networks. In: Proceedings of the ACM KDD Workshop on Link Analysis and Group Detection (2004)
6. Yang, W., Dia, J., Cheng, H., Lin, H.: Mining Social Networks for Targeted Advertising. In: Proceedings of the 39th Annual Hawaii International Conference on System Science (2006)
7. Lahiri, M., Berger-Wolf, T.Y.: Structure Prediction in Temporal Networks using Frequent Subgraphs. In: Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining, Hawaii, pp. 35–42 (2007)
8. Inokuchi, A., Washio, T., Motoda, H.: An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. In: Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (2000)
9. Kuramochi, M., Karypis, G.: Frequent Subgraph Discovery. In: Proceedings of IEEE International Conference on Data Mining (2001)
10. Jiang, C., Coenen, F., Zito, M.: Frequent Subgraph Mining on Edge Weighted Graphs. In: Proceedings of the 12th International Conference on Data Warehousing and Knowledge Discovery (2010)
11. Carter, C.L., Hamilton, H.J., Cercone, N.: Share based Measures for Itemsets. In: Komorowski, J., Żytkow, J.M. (eds.) PKDD 1997. LNCS, vol. 1263, pp. 14–24. Springer, Heidelberg (1997)
12. Pei, J., Han, J., Asl, M.B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Patterns Growth. In: Proceedings of the 17th International Conference on Data Engineering (2001)