Centrality and mode detection in dynamic contact graphs; a joint diagonalisation approach.

Damien Fay
School of Design Engineering and Computing
Bournemouth University
United Kingdom
Email: dfay@bournemouth.ac.uk

Jérôme Kunegis
Centre for Web studies
University of Koblenz–Landau
Germany
kunegis@uni-koblenz.de

Eiko Yoneki
Computer Laboratory
University of Cambridge
United Kingdom
eiko.yoneki@cl.cam.ac.uk

***Abstract*—This paper presents a technique for analysis of dynamic contact networks aimed at extracting periods of time during which the network changes behaviour. The technique is based on tracking the eigenvectors of the contact network in time (efficiently) using a technique called Joint Diagonalisation (JD). Repeated application of JD then shows that real-world networks naturally break into several modes of operation which are time dependent and in one real-world case, even periodic. This shows that a view of real-world contact networks as realisations from a single underlying static graph is mistaken. However, the analysis also shows that a small finite set of underlying static graphs can approximate the dynamic contact graphs studied. We also provide the means by which these underlying approximate graphs can be constructed.**

Core to the approach is the analysis of spanning trees constructed on the contact network. These trees are the routes a broadcast would take given a random starting location and we find that these propagation paths (in terms of their eigenvector decompositions) cluster into a small subset of modes which surprisingly correspond to clusters in time. The net result is that a dynamic network may be approximated as a (small finite) set of static graphs. Most interestingly the MIT dataset shows a periodic behaviour which allows us to know in advance which mode the network will be in. This has obvious consequences as individuals in the network take differing roles in differing modes. Finally, we demonstrate the technique by constructing a synthetic network with an 4 underlying modes of operation; creating synthetics contacts and then used JD to extract the original underlying modes.

I. INTRODUCTION

Real world contact networks have attracted much attention in recent years with the rise of ubiquitous sensing devices and social networking in general. Of particular interest is the spread of viruses in the real world in time varying networks [13] [15] and targeted marketing of individuals or communities [7] based on their physical contacts. As noted in [13], often the time varying structure of these networks is considered too complex to model and instead a static underlying network is assumed. However, contact networks can change radically over a short space of time. The most obvious example being the diurnal pattern; many contact during the day and few at night. In addition to this, humans are generally members of several different communities and even within a community the connections can vary depending on the context; for example a work context versus a social occasion. While the different contexts may be inferred from graph measures or contextual information (for example location) the method presented here is data driven. Given a set of contacts we wish to infer the underlying *modes* of the network, when they occur, if they are repeated and to what degree the network can be classed as being in a certain mode at a given time. In addition, we present a means of creating a static graph which represents the interaction for a mode. The network as a whole may then be modeled as one which changes state between this (small) set of static graphs.

Another factor that makes time dependent networks difficult to model is path ordering; for a time dependent network, the order in which contacts occur is important. For example, a contact path between 3 nodes, $A \rightarrow B \rightarrow C$, does not imply the reverse; $A \rightarrow B \rightarrow C$. However, currently static graphs are constructed based on the total number of contacts between two individuals (for example in [7]), which ignores both the *time* and the *order* dependence in the network. However, consider the spread of a message from a single individual at a given time. This message will propagate through the network forming a (time and order dependent) spanning tree of the network. This spanning tree is a static graph. For many starting individuals and many starting times there will be many such spanning trees. The core of the algorithm proposed here is to look for common eigenspaces[1] in these spanning trees thus preserving ordering and time information.

In this paper we propose the application of Joint Diagonalisation (JD) to track the changes in the eigenspace of contact networks (Section III). The key contributions of our work are: 1. An algorithm for identifying the modes in a contact network and the times at which they appear, and 2. An approximation of a mode as a static network. Overall, therefore we are able to take a complex time-varying network and preserve order and time information while still providing a sparse representation of the information in terms of the modes of the network.

II. RELATED WORK

Scellato et al. [17] examine the different characteristics of contact networks as they evolve over time. However, the analysis there is based on forming static graphs by amalgamating all links seen in an interval of time. This may introduce connections which in fact are unordered. Graph measures (e.g. the clustering coefficient) are then measured from these graphs and a time series analysis of these follows. The technique presented here does not assume a fixed interval but rather

---

[1] i.e. eigenvectors and eigenvalues.

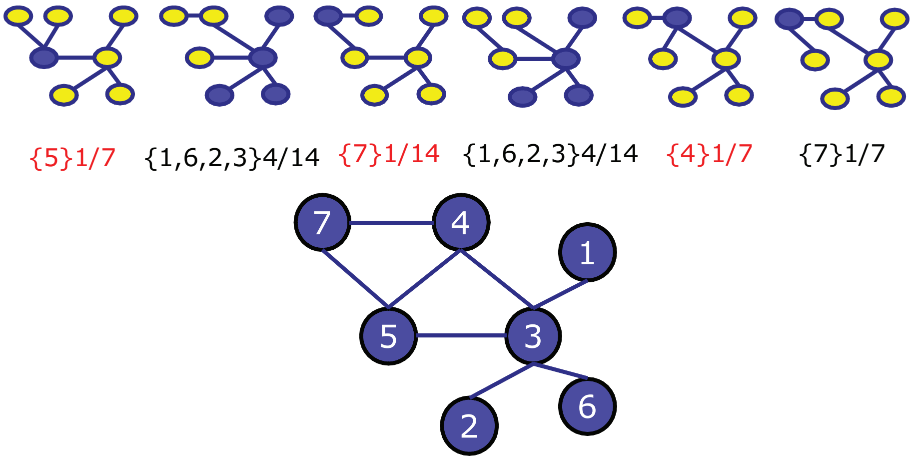

Fig. 1. A simple graph and its 6 spanning trees. (The numbers represent the root nodes and probability of observing the tree ex: 1,6,2,3 are the root nodes for the third tree and this tree is observed with probability 4/14)

determining the appropriate intervals is one of the research questions addressed.

Perhaps the closest papers in spirit to that presented here are those which use tensor decomposition [18][13][9]. A tensor is multi-dimensional matrix (for example a set of adjacency matrices) which are essentially reduced using PCA to a *core* tensor. Sun et al. [18] for example, use tensor analysis to examine time dependent networks. They identify core tensors which allow changes in the network to be tracked in time. Here however, (in their terminology) we uncover a small set of cores which are relevant at different times. In a similar paper, Lin et. al. [9] examine context driven networks such as Flikr and Digg with the focus being on community detection.

The spanning tree analysis carried out here is similar to that in Riolo et al. [15]. They investigate time dependent epidemic networks with a view to constructing *transmission graphs*, directed graphs which indicate the direction of transmission of a disease through a network.The methodology used is significantly different as they examine one time infections in real networks.

Joint diagonalisation has been used in many applications where the evolution of a system can be tracked smoothly via its eigenspace. For example, Macagnano et al. [10] present an algorithm for localisation of multiple objects given partial location information. As time evolves the location of the objects changes smoothly which may be seen through the evolution of the eigenvectors of a distance matrix. Other examples include blind beam forming [3] and blind source separation [19]. By applying joint diagonalisation to social networks we are not tracking position but rather centrality (as centrality is related to the entries of the largest eigenvectors). To the best of our knowledge decomposing a network via eigenspace evolution has not been applied to contact/time dependent networks previously.

III. THEORETICAL BACKGROUND

Snowball sampling consists of selecting a root node randomly in the network with uniform probability and performing a *Breath First Search* (BFS) from this node.[2] This produces a spanning tree, H, where H is a subset of the original graph $G(V, E)$, where V and E denote the vertex and edge sets respectively, and $|V| = N$ denotes the number of nodes. We call the starting node the *observer* or root and H, *the sample*. Figure 1 shows a simple graph which will be used for demonstration purposes. In the first sample node 5 is selected at random and a shortest path first search results in the first tree in Figure 1. In this simple graph there are a total of 6 spanning trees (Figure 1). If we assume that the root node is picked uniformly then it is interesting to note that all spanning trees are *not equally likely*; the first spanning tree occurs with probability 1/7 while the second has probability 4/14 ((Figure 1). We call this effect the sampling bias and the aim here is to develop a centrality which reflects this bias.

Eigenvector centrality [12] is defined such that the centrality of node i, x_i, equals the average of the centrality of all nodes connected to it:

$$x_i = \frac{1}{\lambda} \sum_{j=1}^{N} A_{i,j} x_j \tag{1}$$

where $A_{i,j}$ is element i, j of the (possibly weighted) adjacency matrix and λ is the largest eigenvalue of $A_{i,j}$. as can be seen by rewriting Equation 1 in matrix notation as:

$$X = \lambda^{-1} AX \tag{2}$$

The eigenvector corresponding to λ_{max} gives the eigenvector centrality of node i.

In the current context we have a set of adjacency matrices ordered in time, $\mathcal{A} = \{A_1 \dots A_T\}$, in which A_t is the adjacency matrix of contacts seen in a short interval of $t+\Delta t$ where Δt should be small and may include perhaps no interactions. The network is sampled by choosing a random node uniformly and a random start time uniformly. This node then propagates a message for a period appropriate to the data set (for those

---

[2] i.e. determining a set of shortest paths from the source node to every other node in the network.

used here the time-out is 12 hours) and a spanning tree of network is thus formed.[3]

The Given M samples of a network, $\mathcal{H} = \{H_1 \ldots H_M\}$, the question now arises; how can these be combined to give a matrix that reflects the sampling bias. We propose using a method known as joint diagonalisation which produces an average eigenspace of the samples. Specifically, we seek an orthogonal matrix such that:

$$H_i = UC_iU^T \ \forall i \tag{3}$$

If U corresponds to the eigenvectors of H_i then C_i is diagonal however no matrix U exists in which all C_i are diagonal (except for the trivial case in which all H_i are equal). Joint diagonalisation seeks average eigenvectors $\bar{U}$ such that the sum of squares of the off diagonal elements of C_i are minimised. Specifically:

$$\bar{U} = \underset{U}{\operatorname{argmin}} \operatorname{off}_2(\sum_{j=1}^{M} C_i) \tag{4}$$

where off_2 is the sum of the off diagonal elements squared, called the *deviation* of H_i from $\bar{H}$, δ_i :

$$\delta_i = \operatorname{off}_2(C_i) = \sum_{k \neq j} |C_i^{k,j}|^2 \tag{5}$$

where $C_i^{k,j}$ is the k^{th} row and j^{th} column of C_i. As shown in [20] and [2] Equation 4 may be minimised efficiently by a sequence of Givens rotations; convergence and stability properties are proven in [1].

Given the average eigenstructure of the sample matrices an weighted sampling graph may be constructed by jettisoning the information in the off diagonal elements in each C_i and taking the resulting average of the eigenvalues as:

$$\bar{H} = \bar{U}\bar{C}\bar{U}^T \tag{6}$$

Where $\bar{H}$ is a matrix in which the entries represent the average weight of the links as observed by the samples in the network (in a least squares sense) and $\bar{C}$ is the average of diagonals of H_i projected onto $\bar{U}$; i.e. the average eigenvalues. A sample based centrality may then be constructed from $\bar{H}$ using the standard eigenvector centrality; i.e. by using the eigenvector of $\bar{H}$ corresponding to the maximum eigenvalue. Note that the joint diagonalization of a set of symmetric matrices can be equivalently described as a tensor decomposition; specifically of a PARAFAC/CANDECOMP type [8].

A further decomposition can now be implemented but this time using the distribution of δ_i to perform clustering of the sample matrices, $\mathcal{H}$. The aim here is to find K sub-populations of $\mathcal{H}$ under the assumption that those H_i with similar values of δ_i are themselves similar. As will be seen in the example below, further rounds of clustering may be required to extract the modes from the network. The clustering is based on a Gaussian mixture mode (GMM) [11] to explain the distribution of the deviations; f_{δ_i}. Algorithm 1 summarises the steps above in the **Contact Network Mode Decomposition** (CNDM) algorithm.

```
Algorithm 1: CNMD
        Data: $\mathcal{A}$: a set of time ordered adjacency matrices
        Result: $\{C_1, ..., C_K\}$
        $\{\bar{H}, \bar{U}, \delta_i\}$
        $\{\bar{H}_{C1}, \bar{U}_{C1}, \delta_{i_{C1}}\}$...
        $\{\bar{H}_{CK}, \bar{U}_{CK}, \delta_{i_{CK}}\}$
        begin
            for $i = 1 \to M$ do
Sample:         $v \sim U[0, |V|]$
                $t \sim U[0, T]$
                $H_i \leftarrow$ BFS from $v$ at time $t$
Joint Diag: $\{\bar{U}, \bar{H}, \bar{C}, C_i, \delta_i\} \leftarrow JD(\mathcal{H})$
Cluster:    $\{K, C_1, ...C_K\} \leftarrow$ GMM($\delta_i$)
        for $j = C_1 \to C_K$ do
Joint Diag: $\{\bar{U}_j, \bar{H}_j, \bar{C}_j, C_{i_j}, \delta_{i_j}\} \leftarrow JD(\mathcal{H}_{C_j})$
```

A. Example: a Synthetic contact network.

The first network created in this section is a purely random contact network in which 5% of 50 nodes are connected at random in each time step. Figure 2 shows the distribution of δ_i for this network is uni-modal. The distribution also follows a Γ distribution[4]. This shows that a unimodal distribution is the default for purely random network with no structure.

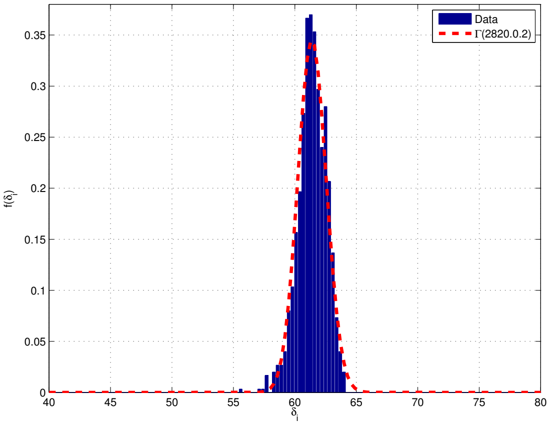

Fig. 2. Distribution of δ_i (Random network).

The second network is more complicated and involves generating four different behaviours for a contact network termed *generators*. A generator consists of a static underlying topology representing a set of possible contacts. These links are transformed into contacts by using a Lévy walk (as justified in [6] [14]); a set of times are generated from a power law distribution and used to demarcate when a contact takes place. The generator used is switched every 700 time units as shown by the mode indicator in Figure 4. Specifically, the first generator employs a Waxman topology [21] ($\alpha = 0.5$,$\beta = 0.3$)[5]. The second generator is also a Waxman model

---

[3] During the night many of the spanning trees are incomplete or empty; this has no effect on the JD algorithm as the algorithm performs a Givens rotation on the largest absolute off-diagonal element which is never a zero element.

[4] δ_i is a squared quantity which should follow a similar distribution to a sample variance; i.e. $\delta_i \sim \chi^2$. The Γ distribution is a generalisation of the χ^2 distribution and so is used.

[5] Waxman topology: $p(u \leftrightarrow v) = \alpha e^{-\beta d}$ where α and β are parameters of the model. The nodes are distributed randomly on a grid and the distance between them is d.

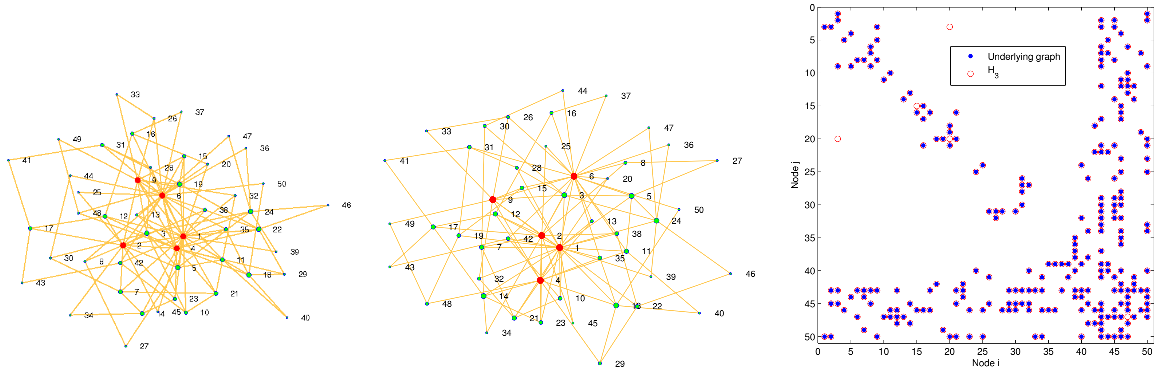

Fig. 7. (a) The generating graph for submode 2*. (b) $\bar{H}$ shortest path graph for submode 2*. (c) A spy plot of the generating graph (.) and $\bar{H}$ (o)**. (Synthetic data) * The size of a node is proportional to the sum of weights incident on that node. ** The rows and columns have been permuted using approximate minimum degree to highlight the preferential attachment community structure in the graph. $\bar{H}$ has been thresholded using a value of 0.1 ($\bar{H} < 0.1 \longmapsto 0$).

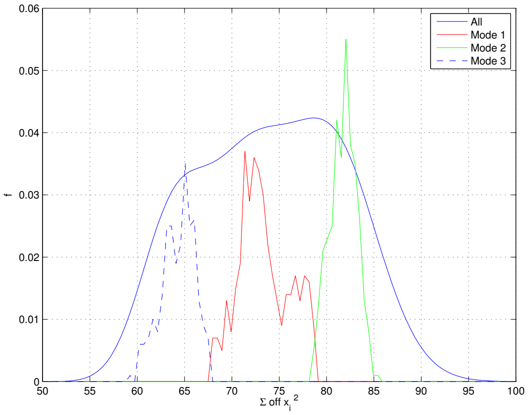

Fig. 3. Distribution of δ_i (Synthetic).

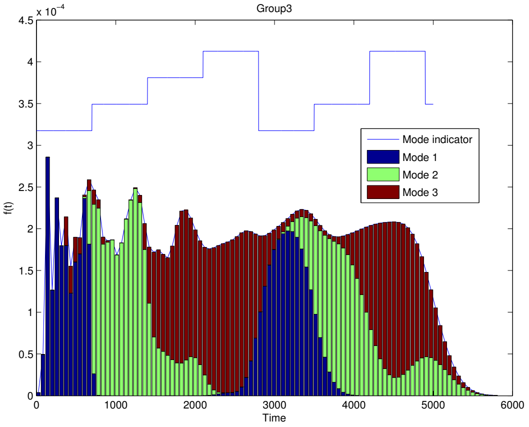

Fig. 4. Distribution of times by mode. (Synthetic)

with $\alpha = 0.7$,$\beta = 0.3$. The third and fourth generators are Generalised Linear Preferential (GLP) topologies based on preferential attachment [16]. As can be seen 3 modes are detected in the data (Figure 3). These correspond with the generator times for 2 of the modes (Figure 4). However, mode 3 incorporates both generator 3 and generator 4. This occurs as generator 3 and 4 are quite similar (both based on GLP).

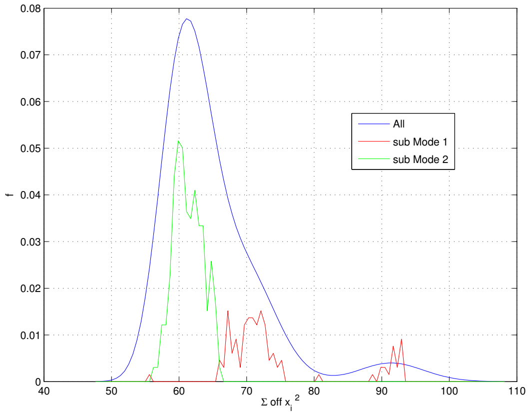

Fig. 5. Distribution of δ_i; submode. (Synthetic).

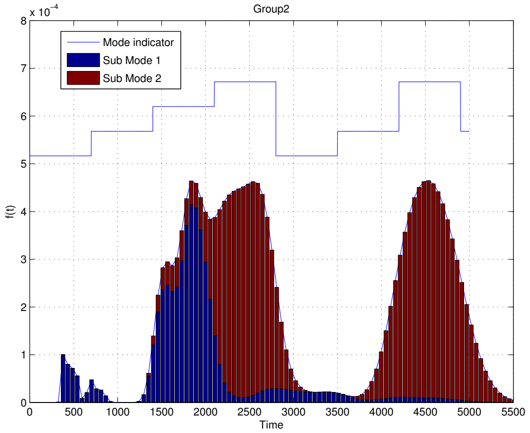

Fig. 6. Distribution of times by mode; submode. (Synthetic)

The samples in mode 3 may be examined separately using JD to produce the *submodes* seen in Figure 5 occurring at the times seen in Figure 5. As can be seen these submodes are generators 3 and 4. Thus the algorithm has successfully recovered the modes in the data. At this point we make a note on the transition between the modes. It is interesting that

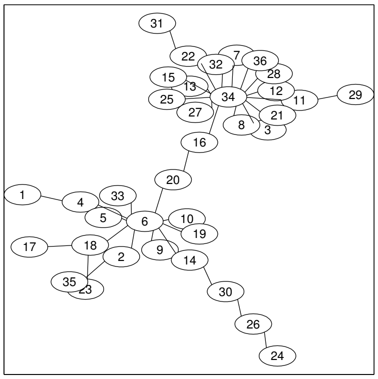

Fig. 8. A typical flooding tree; as seen from node 20.

this transition is not crisp even though the switching between modes is. This is because a spanning tree may begin in one mode but the message may end in the next mode.

IV. DATA SET DETAILS

In this paper, we use three experimental datasets gathered by the Haggle Project [5], referred to as *Cambridge*, *Infocom06*; one dataset from the MIT Reality Mining Project [4], referred to as *MIT*. Previously, the characteristics of these datasets such as inter-contact and contact distribution have been explored in several studies [6], to which we refer the reader for further background information. These three datasets cover a rich diversity of environments, ranging from a quiet university town (*Cambridge*), with an experimental period from a few days (*Infocom06*) to one month (*MIT*).

V. RESULTS

A. Cambridge data

Figure 8 shows a typical sampling tree for the Cambridge data set. Node 20 initiates a message and it is passed around the contact network; first to nodes 16 and 6 and from there to the rest of the network. For this experiment we use $M = 10,000$ [6]. $\bar{H}$ (Equation 6), for this data set is *represented* in Figure 11(a). This representation shows all links in the weighted shortest paths of $\bar{H}$[7]. As can be seen the nodes split into two groups which correspond with those reported in [22]. f_{δ_i} is shown in Figure 10. As can be seen there are $K = 5$ modes **which also correspond to different time periods**. This is particularly useful as it allows the network to be characterized by different modes of behaviour at different times.

B. INFOCOM '06 data

The Infocom data is summarised in Figure 12 and follows the behaviour typically expected at a conference. Four modes are identified (Figure 12(a)) which correspond to four periods in time (Figure 12(b)). The first mode to occur is mode 3

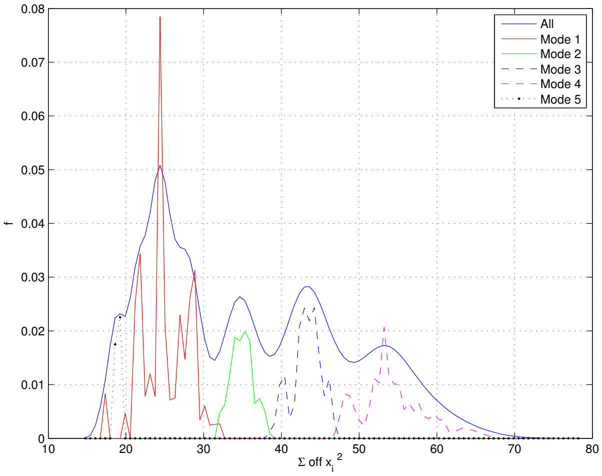

Fig. 9. Distribution of δ_i (Cambridge data set; kernel smoothing is employed for the overall average.).

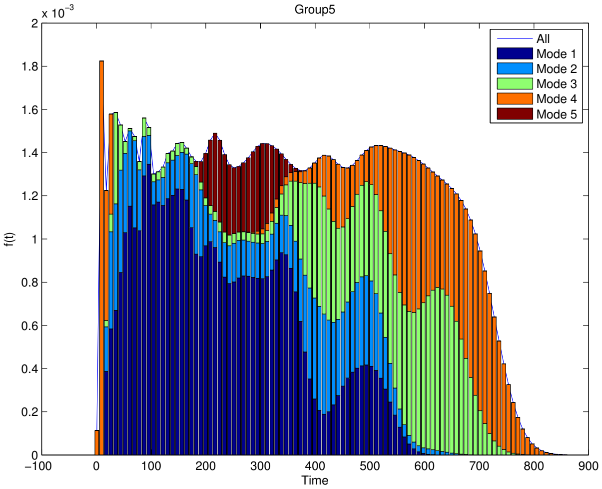

Fig. 10. Distribution of times by mode.

showing much mixing between the delegates (Figure 12(e)). This is probably the delegates meeting for coffee before the conference begins. This is then followed by two periods of structured graphs (i.e. presentations; Figures 12(c,d)) ending with a period of mixing (Figures 12(f)).

C. MIT data

The results from the MIT data set show a different type of behaviour. The graph of $\bar{H}$ clearly shows that there are two main communities in the network (Figure 13). f_{δ_i} is bi-modal (Figure 14) leading to two main modes. The MIT data set spans a month of data and again the modes correspond with periods in time but they are also recurring (Figure 15). This is particularly interesting as it implies that for this network we may be able to predict the mode in advance and act accordingly. To test this hypothesis we designed a marketing experiment which we run on the network in software.

The experiment consists of offering a discount to a group via an influential individual in the network for a period of 2 hours (one might imagine a lunch voucher for the individual

---

[6] A large sample size is used here to negate random effects. However, similar results are found for much smaller sample sizes.

[7] We found this to be the clearest means of representing these complete weighted graphs.

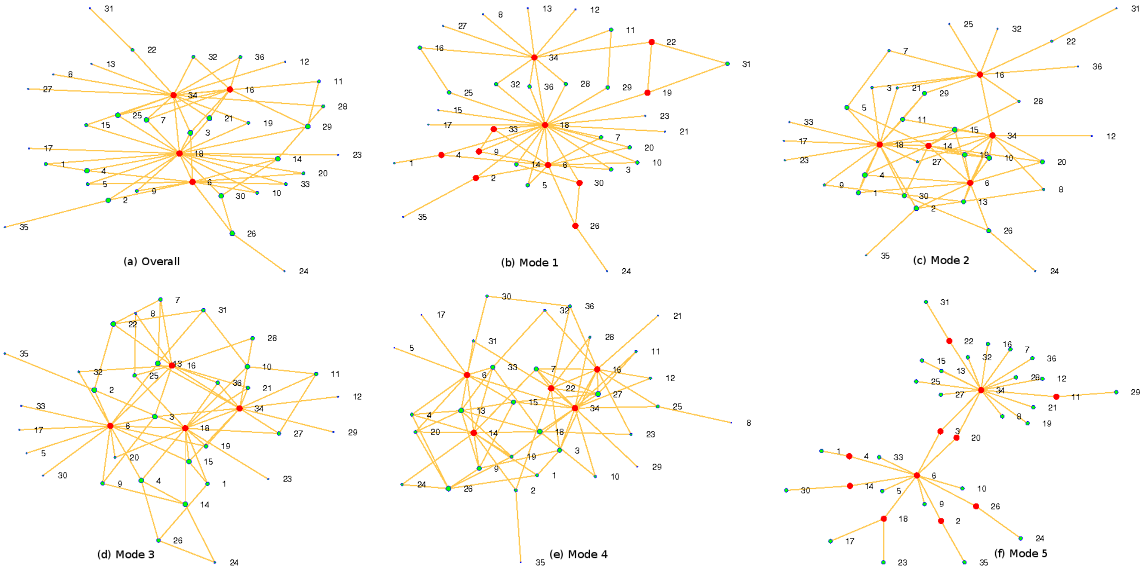
(a) Overall (b) Mode 1 (c) Mode 2
(d) Mode 3 (e) Mode 4 (f) Mode 5

Fig. 11. Graph of shortest paths in $\bar{H}$ for overall and 5 modes. (Cambridge data set; the size of a node is proportional to the sum of weights incident on that node)

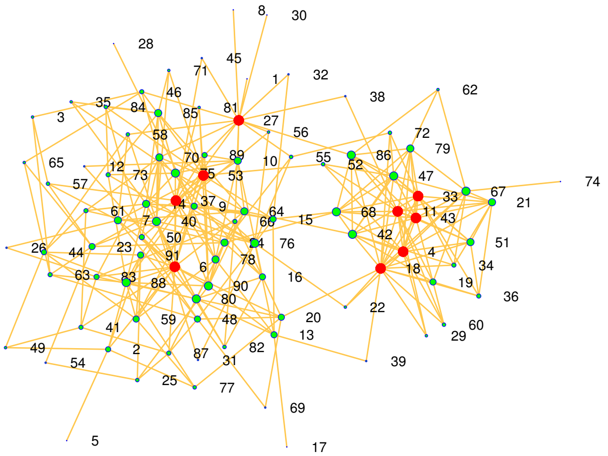

Fig. 13. Graph of shortest paths in $\bar{H}$; overall. (MIT).

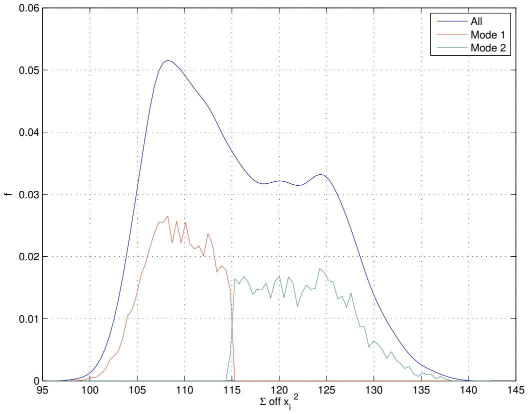

Fig. 14. Distribution of δ_i (MIT).

and his friends for example). A person coming into contact with the group is convinced to join with probability ρ. Thus, this experiment is similar to spreading a disease but with a fixed expiry time. Individuals are chosen to be the seed nodes with probability proportional to their centrality. The experiment is averaged over 50 runs and the average number of customers touched is recorded. First we note the difference between the centralities according to a static graph, those based on $barH$ and those based on the two modes; $\bar{H}_C1$ and $\bar{H}_C2$, Figure 16. The 45°line shows the static centrality versus itself. As can be seen the centralities according to $barH$, $\bar{H}_C1$, and $\bar{H}_C2$ are all similar but they deviate significantly from the static centralities especially for the most central nodes who's centrality is under estimated by the static representation. Figure 17 shows the number of customers touched in the simulation as a function of rho (summed over all time). As can be seen the centralities based on the sampling graphs are superior to those of the static graph. There is little difference between the graph based on $\bar{H}$ and that based on the two modes. For $rho = 1$, the centralities based on $\bar{H}$ give a 3.2 % improvement while those based on $\bar{H}_{C1}$ combined with $\bar{H}_{C2}$ improve on this figure to give a 3.8 % improvement. This result shows that the sampling centrality is a better measure of the centrality of a node with respect to the simulated scheme laid out above.

VI. CONCLUSIONS

This paper presented a method for extracting different modes of operation for contact networks. In the real-world contact networks examined, several interesting features where extracted including detection of a modes in time in the Cambridge data set. The MIT data set in contrast, showed a repetitive behaviour which is useful for prediction of network behaviour; a feature which simulations show may be useful in targetted advertising. The INFOCOM data set clearly showed the behaviour typical of a conference. In producing an average graph based on samples of a network, the order of contacts

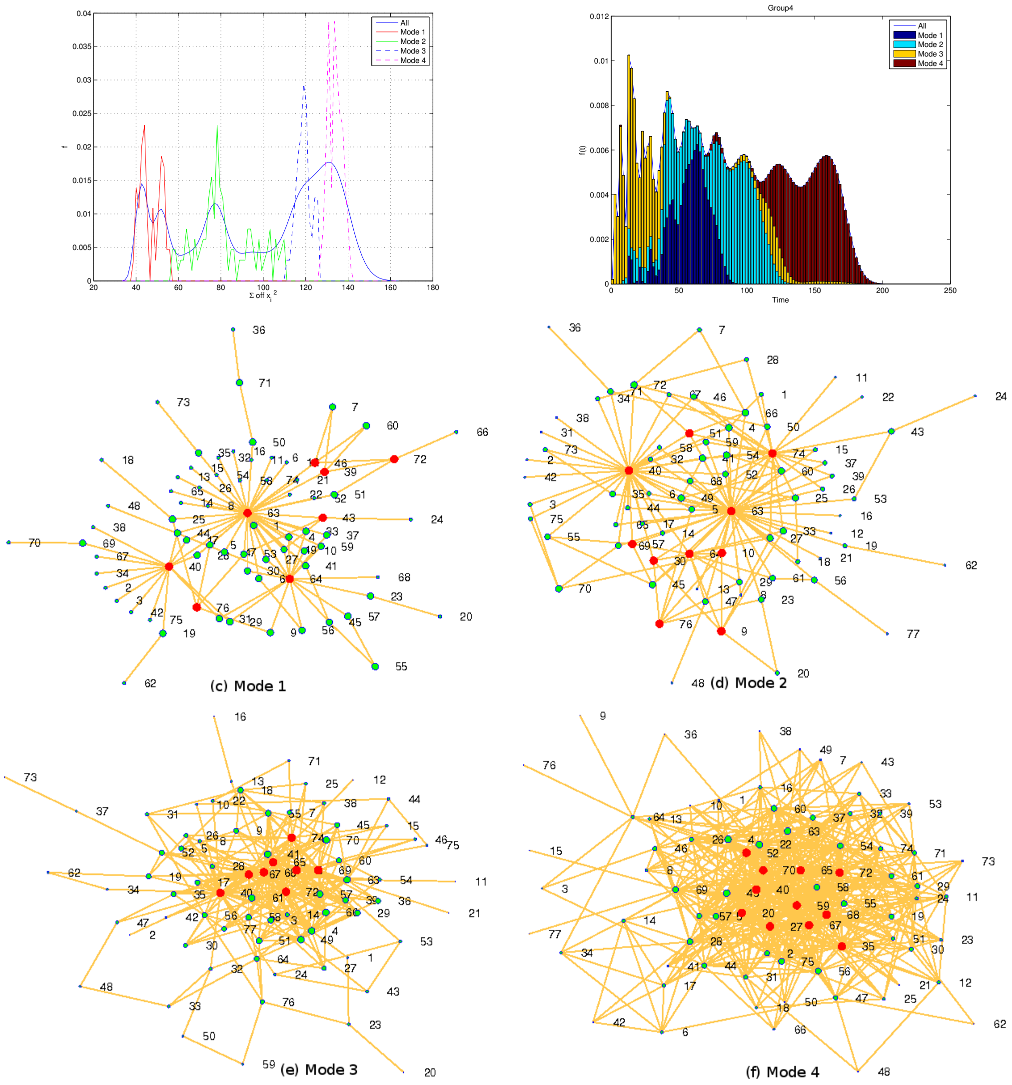

Fig. 12. Infocom06 results. (a) Distribution of δ_i. (b) Distribution of times by mode. (c-f) $\bar{H}$ for each mode.

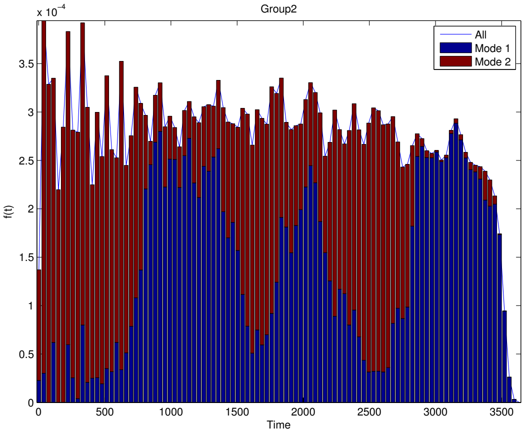

Fig. 15. Distribution of times by mode. (MIT)

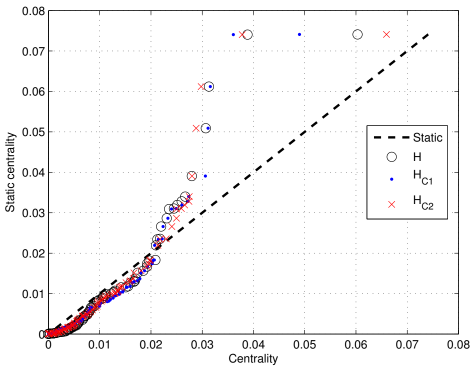

Fig. 16. The centrality of nodes according to the static representation, $\bar{H}$, $\bar{H}_{C_1}$ and $\bar{H}_{C_2}$ drawn against the static representation centrality. (MIT)

has been preserved and in addition the correlation between contacts has been preserved. For example aggregation based purely on counting the number of times a link is present does not take into account the fact that links may typically be present *together*; i.e. the time based correlation between links.

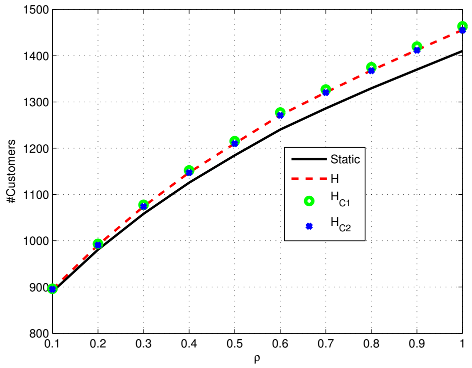

Fig. 17. The number of customers that are seen in the simulations. (MIT)

By using spanning trees the methodology takes advantage of a sampling mechanism present in many real-world networks; it might not be possible to record all contacts but it is often possible to flood a message in a network and record the paths taken. Here we concentrated on the eigenvector centrality and the adjacency matrix but extensions to other centralities and matrices is an avenue for future work.

ACKNOWLEDGMENT

The research is part funded by the EU grants for the Recognition project, FP7-ICT-257756 and the EPSRC DDEPI Project, EP/H003959. We would like to thank the members of Systems Research Group, University of Cambridge, for their comments and suggestions.

REFERENCES

[1] A. Bunse-Gerstner, R. Byers, and V. Mehrmann. Numerical methods for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 14:927–949, 1993.

[2] J.-F. Cardoso, J. fran Cois Cardoso, and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM J. Mat. Anal. Appl*, 17:161–164, 1996.

[3] J.-F. Cardoso and A. Souloumiac. Blind beam-forming for non gaussian signals. *IEE Proceedings-F*, 140:362–370, 1993.

[4] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, V10(4):255–268, May 2006.

[5] EU FP6 Haggle Project. http://www.haggleproject.org, 2010.

[6] M. Freeman, N. Watkins, E. Yoneki, and J. Crowcroft. Rhythm and randomness in human contact. In *Proc. International Conference on Advances in Social Networks Analysis and Mining*, 2010.

[7] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 241–250. ACM, 2008.

[8] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September 2009.

[9] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. Metafac: community discovery via relational hypergraph factorization. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 527–536, New York, NY, USA, 2009. ACM.

[10] D. Macagnano and G. T. F. de Abreu:. Gershgorin analysis of random gramian matrices with application to mds tracking. *IEEE Transactions on Signal Processing*, accepted for publication; available on IEEE early access (15/02/2011).

[11] G. J. McLaughlan. *Finite Mixture Models*. Wiley, 2000.

[12] T. Opsahl, F. Agneessens, and J. Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245 – 251, 2010.

[13] Prakash, H. Tong, N. Valler, M. Faloutsos, and C. Faloutsos. Virus Propagation on Time-Varying Networks: Theory and Immunization Algorithms. In J. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6323 of *Lecture Notes in Computer Science*, chapter 7, pages 99–114. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[14] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong. On the levy-walk nature of human mobility. In *Journal of Selected Areas in Communications*, volume 6(9), 2008.

[15] C. Riolo, J. Koopman, and S. Chick. Methods and measures for the description of epidemiologic contact networks. *Journal of Urban Health: Bulletin of the New York Academy of Medicine*, 78:446–457(12), 1 September 2001.

[16] C. Roth. Generalized preferential attachment: Towards realistic socio-semantic network models. In *ISWC 4th Intl Semantic Web Conference, Workshop on Semantic Network Analysis*, volume 171 of *CEUR-WS Series (ISSN 1613-0073)*, pages 29–42, Galway, Ireland, November 2005.

[17] S. Scellato, M. Musolesi, C. Mascolo, and V. Latora. On nonstationarity of human contact networks. In *Proceedings of 2nd Annual Workshop on Simplifying Complex Networks for Practitioners (SIMPLEX 2010). Co-located with ICDCS 2010.*, June 2010.

[18] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: Dynamic tensor analysis. In *Proc. Int. Conf. on Knowledge Discovery and Data Mining*, pages 374–383, 2006.

[19] W. Wang, S. Sanei, and J. Chambers. Penalty function-based joint diagonalization approach for convolutive blind separation of nonstationary sources. *IEEE Transactions on Signal Processing*, 53:1654 – 1669, May 2005.

[20] M. Wax and J. Sheinvald. A least-squares approach to joint diagonalization. *IEEE Signal Processing Letters*, 4(2):52–53, 1997.

[21] B. Waxman. Routing of multipoint connections. *IEEE J. Select. Areas Commun.*, 6(9):1617–1622, 1988.

[22] E. Yoneki. Visualizing communities and centralities from encounter traces. In *ACM MobiCom Workshop on Challenged Networks (CHANTS)*, 2008.