

Number 806



**UNIVERSITY OF
CAMBRIDGE**

Computer Laboratory

On joint diagonalisation for dynamic network analysis

Damien Fay, Jérôme Kunegis, Eiko Yoneki

October 2011

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500
<http://www.cl.cam.ac.uk/>

© 2011 Damien Fay, Jérôme Kunegis, Eiko Yoneki

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

<http://www.cl.cam.ac.uk/techreports/>

ISSN 1476-2986

On joint diagonalisation for dynamic network analysis

Damien Fay ^{#1}, Jérôme Kunegis ^{&2}, Eiko Yoneki ^{§3}

[#]University of Cork, Ireland

[&]University of Koblenz–Landau, Germany

[§]University of Cambridge, United Kingdom

¹damien.fay@cl.cam.ac.uk

²kunegis@uni-koblenz.de

³eiko.yoneki@cl.cam.ac.uk

Abstract—Joint diagonalisation (JD) is a technique used to estimate an average eigenspace of a set of matrices. Whilst it has been used successfully in many areas to track the evolution of systems via their eigenvectors; its application in network analysis is novel. The key focus in this paper is the use of JD on matrices of spanning trees of a network. This is especially useful in the case of real-world contact networks in which a single underlying static graph does not exist. The average eigenspace may be used to construct a graph which represents the ‘average spanning tree’ of the network or a representation of the most common propagation paths. We then examine the distribution of deviations from the average and find that this distribution in real-world contact networks is multi-modal; thus indicating several *modes* in the underlying network. These modes are identified and are found to correspond to particular times. Thus JD may be used to decompose the behaviour, in time, of contact networks and produce average static graphs for each time. This may be viewed as a mixture between a dynamic and static graph approach to contact network analysis.

Keywords. Social networks, joint diagonalisation, graph analysis, spanning tree, human contact networks

I. INTRODUCTION

Understanding the dynamic structure of contact networks is critical for designing dynamic routing algorithms [11], epidemic spreading [16] and message passing algorithms [10].

Time dependent networks are characterised by time dependant paths which are characterised by the order in which the paths occur. For example, a path between 3 nodes $A \rightarrow B \rightarrow C$ does not imply a reverse path exists; $A \rightarrow B \rightarrow C$ provides no information about how C may communicate with A . However, in many applications a static graph is constructed which represents typically the proportion of time a link was seen between two nodes. These static graphs often lose the time information which is critical in contact networks. However, at a *specific time* and from a *specific node* there is a single static network representing the paths between the root node and the rest of the network.

JD is used in this paper to look for commonalities in these specific static networks and to provide an *average* static network where appropriate. That is, we are looking for *modes* of operation in contact networks.

Joint Diagonalisation (JD) is a technique that is used to track the changes in eigenspace (i.e. eigenvectors and eigenvalues) of a system (see Section III for examples). Eigenvectors and eigenvalues play an important role in static network/graph analysis as they can be used to determine the centrality of nodes; communities and settling times among other things [1]. However, to the best of our knowledge tracking eigenspace evolution has not been applied to contact/time dependent networks previously. This paper examines the use of JD in network analysis.

II. RELATED WORK

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. [12] 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 [4] and blind source separation [20].

Sun et al. [19] use tensor analysis to examine time dependent networks. A tensor is multi-dimensional matrix (for example a set of adjacency matrices) which are essentially reduced using PCA to a *core* tensor. This technique is similar in spirit to that presented here, the difference is that we are looking to reduce a set of spanning trees representing propagation through a network; propagation information being preserved.

Scellato et al. [18] 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

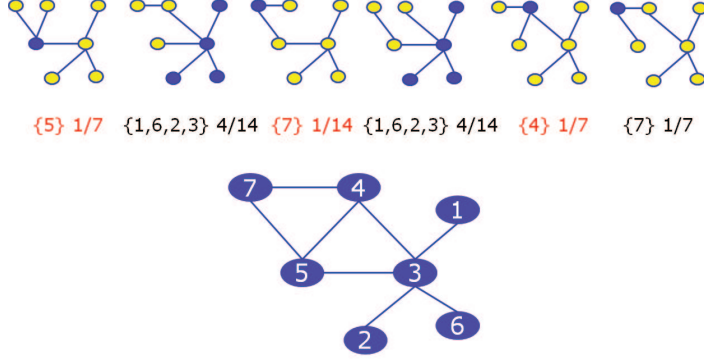


Fig. 1. A simple graph and its 6 spanning trees. (The numbers represent the root nodes and probability of observing the tree ex: 1,6,2,3 are the root nodes for the third tree and this tree is observed with probability 4/14)

fact are unordered. Graph measures (e.g. the clustering coefficient) are then measured from these graphs and a time series analysis of these follows. In contrast, here the amalgamation is dependent of the contact network itself.

Riolo et al. [16] 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. While the aim in this paper is similar, the methodology used is significantly different as they examine one time infections in real networks.

III. THEORETICAL BACKGROUND

We begin by defining snowball sampling which consists of selecting a root node randomly in the network with uniform probability and performing a *Breadth First Search* (BFS) from this node (i.e. determining a set of shortest paths from the source node to every other node in the network). 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 6 spanning trees shown in Figure 1. Note that the distribution of spanning trees in this network are not uniform but biased. That is, traffic generated uniformly from each node results in non-uniform percolation across the network.

Next we develop a centrality measure which is based on standard eigenvector centrality. Eigenvector centrality [14] is defined by letting the centrality of node i equal the average of the centrality of all nodes connected to it:

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

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:

$$X = \frac{1}{\lambda} AX \quad (2)$$

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

Given M samples of a network, $H_1 \dots H_M$, 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:

$$H_i = UC_iU^T \quad (3)$$

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:

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

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

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

where $C_i^{k,j}$ is the k^{th} row and j^{th} column of C_i . As shown in [21] and [3] Equation 4 may be minimised efficiently by a sequence of Givens rotations; convergence and stability properties are proven in [2].

Given the average eigenstructure of the sample matrices an average sampling matrix may be constructed from the eigenvector decomposition as:

$$\bar{H} = U\bar{C}U^T \quad (6)$$

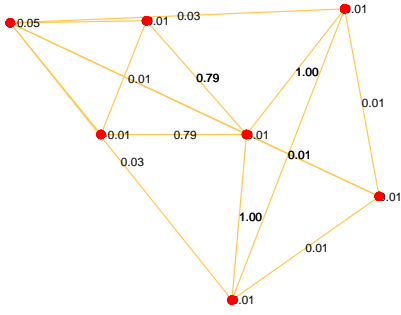


Fig. 2. The average sampling graph corresponding to \bar{H} .

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.

A. Simple examples.

Using the graph shown in Figure 1, 100 samples are taken by randomly choosing a root node and constructing a spanning tree from each. These are then jointly diagonalised producing the eigenvectors shown in Table I. The sampling centrality is the first eigenvector (column 1).

TABLE I

THE AVERAGE EIGENVECTOR, U FROM THE GRAPH IN FIGURE 1.

1	2	3	4	5	6	7
0.3095	0.74	-0.34	-0.29	0.13	-0.35	0.14
0.6780	-0.00	0.70	-0.19	-0.09	0.00	-0.09
0.3370	0.00	-0.26	0.47	-0.73	0.00	0.26
0.3357	0.00	-0.26	0.45	0.28	0.00	-0.73
0.3095	-0.07	-0.34	-0.29	0.13	0.81	0.14
0.1634	-0.00	0.14	0.54	0.57	0.00	0.58

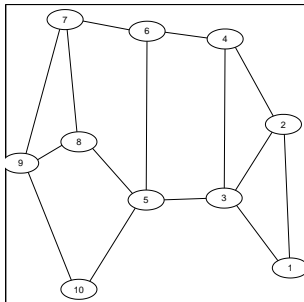


Fig. 3. The simple graph used in example 2.

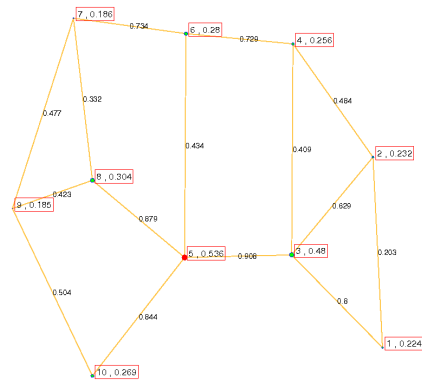


Fig. 4. \bar{H} for example 2 (Note: the red box contains the node number and the sampling centrality).

It is instructive to view the eigenvector reconstruction of \bar{H} which is formed from the eigenvectors in Table I via Equation 6. \bar{H} is drawn in Figure 2. Note that \bar{H} is a complete weighted graph; weights assigned to non-existent links are low and are a consequence of taking an average of many graphs. The edge between nodes 3 and 4 has a weight of $0.7857 \frac{11}{14}$, i.e. the proportion of trees that use that link (see Figure 1). \bar{H} represents the sample biased weight of this link; a nice result. There are two reasons why these numbers are not exactly the same; the first is that, as always, there is a slight error introduced when using empirical sampling. The second is that the \bar{H} is based on the average eigenspace of a set of sampling trees; this is not the same as simply taking the proportion of times a link has been observed.

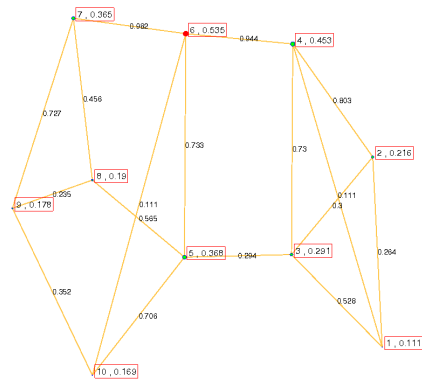


Fig. 5. \bar{H} for example 2 with a preference for routes $6 \leftrightarrow 4$.

The second example deals with a preferred route. Figure 3 shows a graph with a bridge formed from

nodes 6 4 and 5 3 ; these nodes are critical in joining the two parts of the graph.

Figure 4 shows the weighted graph created by sampling as above¹. The second stage involves creating a preferred route by removing 80% of trees that use the link between 5 3. The resultant average graph is shown in Figure 5. As can be seen the weight attached to link 5 3 is greatly reduced (from 0.9 to 0.2).

Thus far we have only dealt with samples taken on static graphs. However, joint diagonalisation is particularly well suited to contact networks. In these networks there is no underlying static graph as such, but rather a set of contacts that are time dependent. By flooding these networks spanning trees may be formed and combined by the use of JD. The next section details these real-world data sets.

IV. DATA SET DETAILS

In this paper, we use four experimental datasets gathered by the Huggle Project [7], referred to as *Cambridge*, *Infocom06*; one dataset from the MIT Reality Mining Project [6], referred to as *MIT*. Previously, the characteristics of these datasets such as inter-contact and contact distribution have been explored in several studies [9], 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*).

Experimental data set	Cambridge	Infocom06	MIT
Device	iMote	iMote	Phone
Network type	Bluetooth	Bluetooth	Bluetooth
Duration (days)	11	3	246
Granularity (seconds)	600	120	300
Number of Devices	36	78	97
Number of contacts	10,873	191,336	54,667
Average # Contacts/pair/day	0.345	6.7	0.024

TABLE II
CHARACTERISTICS OF EXPERIMENTAL DATA SETS

- In *Cambridge*, the iMotes were distributed mainly to two groups of students from University of Cambridge Computer Laboratory, specifically undergraduate year1 and year2 students, and also some PhD and Masters students. This dataset covers 11 days.
- In *Infocom06*, the trace contains 78 participants. Among 78 participants, 34 form 4 subgroups by academic affiliations.
- In *MIT*, 100 smart phones were deployed to students and staff at MIT over a period of 9 months. These phones were running software that logged contacts with other Bluetooth enabled devices

¹Links with low weights are removed for clarity.

by doing Bluetooth device discovery every five minutes. 1 month of data is used here to maintain the consistency of users.

The three experiments are summarised in Table II and the trace data can be downloaded at CRAWDAD database [5].

V. RESULTS

The results below examine the 3 data sets separately highlighting the features of each. Finally a synthetic contact network with known characteristics is constructed and JD used to extract these characteristics.

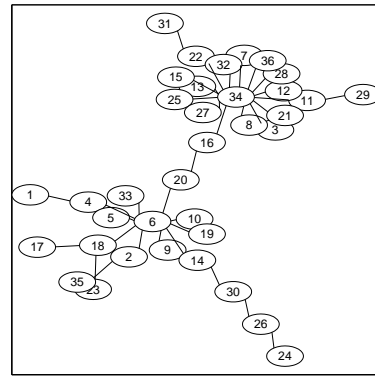


Fig. 6. A typical flooding tree; as seen from node 20.

Next the distribution of the sample start times is examined Figure 9. As can be seen the 5 modes correspond to different times in the data set. Modes 1 and 5 cover the first half of the data while modes 2 then 3 and then 4 become dominant in that succession.

A. Cambridge data

Figure 6 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

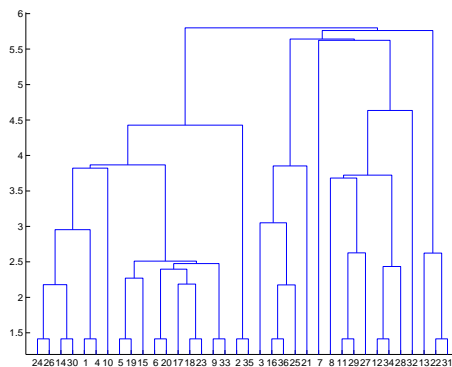


Fig. 7. Community based on Fiedler clustering (Cambridge data set).

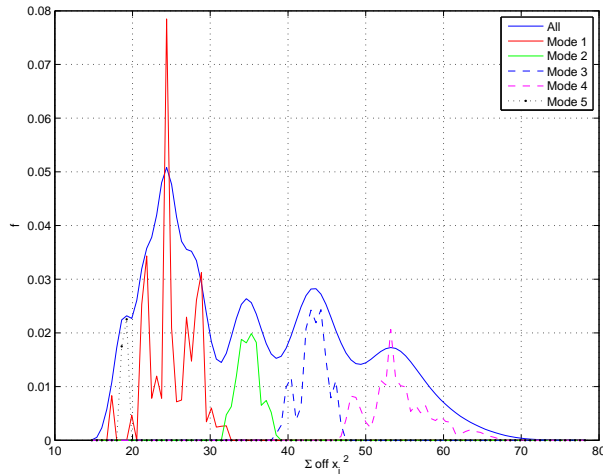


Fig. 8. Distribution of δ_i (Cambridge data set; kernel smoothing is employed for the overall average).

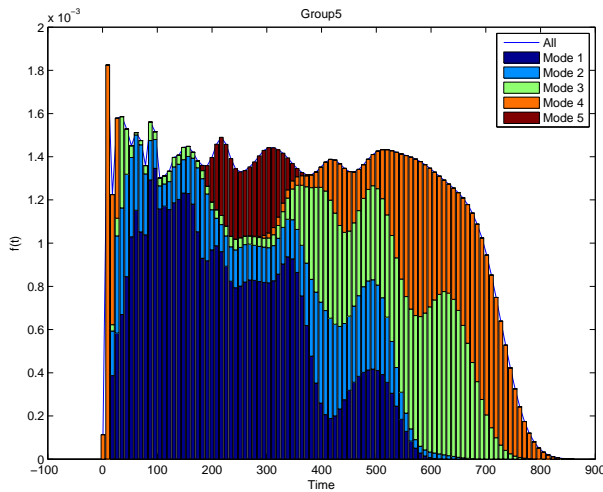


Fig. 9. Distribution of times by mode.

16 and 6 and from there to the rest of the network. For this experiment ten thousand such trees are generated² with the messages starting at a random times and from a random node (uniformly distributed). These are then combined using Joint Diagonalization to form \bar{H} .

The average graph, \bar{H} , for this data set is represented in Figure 10(a). This representation shows all links in the weighted shortest paths of \bar{H} ³. As can be seen the nodes split into two groups as expected. These groups may be represented by a standard dendrogram based on Fiedler vector clustering [8] as shown in Figure 7. The groups seen here correspond closely to those found on the same data set in [23].

The results so far have examined the average behaviour of the contact network which is interesting in

²A large sample size is used here to negate random effects. However, similar results are found for much smaller sample sizes.

³We found this to be the clearest means of representing a complete weighted graphs.

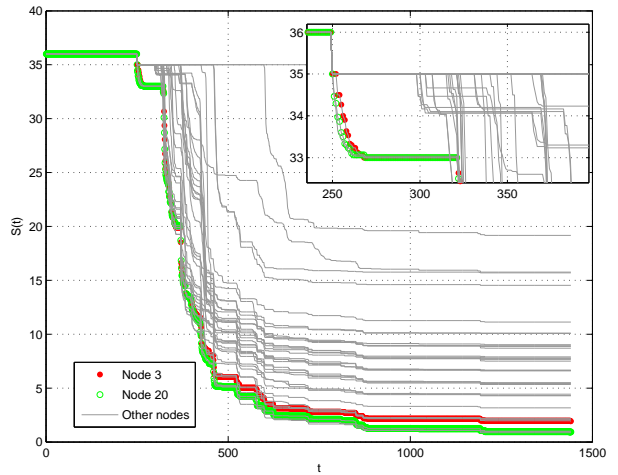


Fig. 11. Mean number of nodes susceptible to disease after time t (Root infection starts at time 250. Inset focuses on the start of the infection.).

itself. However, by examining the distribution of deviations, δ_i , from the average a more interesting behaviour may be observed. Figure 8 shows the distribution of δ_i , $i = 1 \dots 10,000$. As can be seen the distribution is multi-modal; i.e. the underlying process/contact network has different modes of operation. A Gaussian mixture model [13] is used to determine the different modes as shown in Figure 8. 5 different modes are identified. This is particularly useful as it allows the network to be characterized by different modes of behaviour at different times. \bar{H} for the overall data set and for each mode are shown in Figure 10. Mode one shows a highly structured network corresponding to the day when the groups are well defined by *class year* (i.e. year 1 and year 2). This structure then becomes less well defined as time moves on. Mode 5 is particularly interesting as there is an obvious bridge formed by nodes 3 and 20. This mode covers the night time and

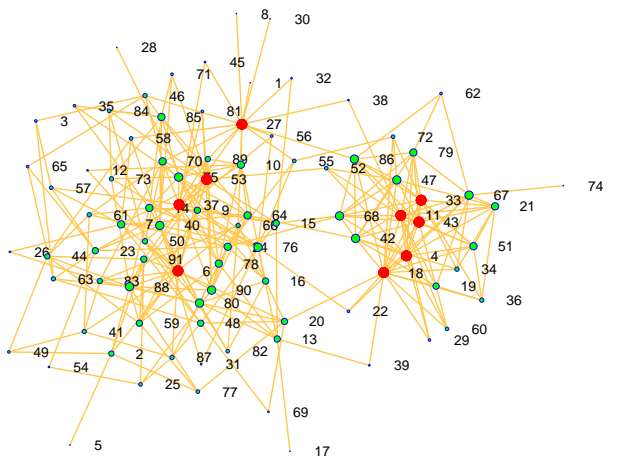


Fig. 12. Graph of shortest paths in \bar{H} ; overall. (MIT).

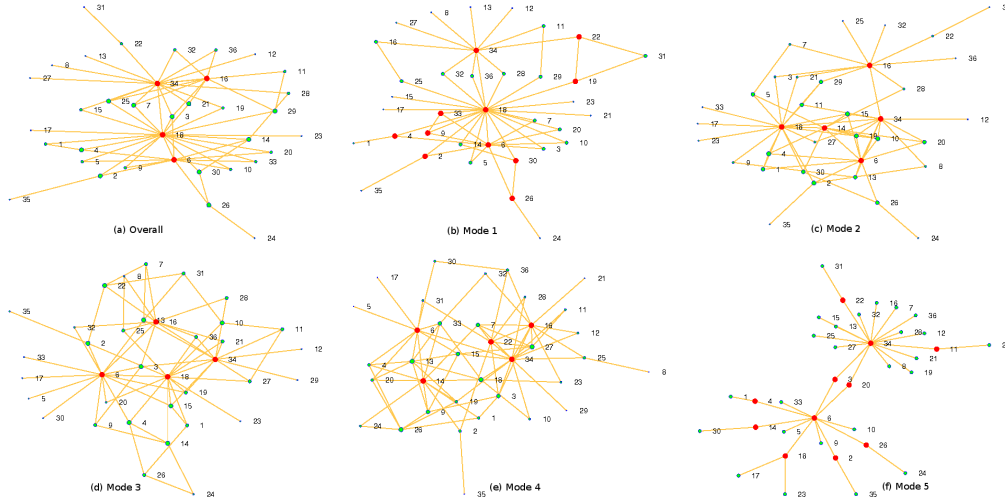


Fig. 10. Graph of shortest paths in \bar{H} for overall and 5 modes. (Cambridge data set; the size of a node is proportional to the sum of weights incident on that node)

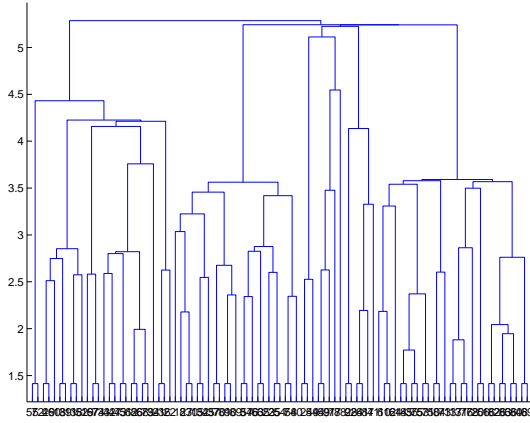


Fig. 13. Community based on Fiedler clustering (MIT data set).

nodes 3 and 20 are possibly staff who interact with the students in the morning⁴. Using \bar{H} as an indicator, this implies that a disease spread at this time from nodes 3 and 20 should have the fastest infection rate. Note that mode 1 is still dominant in this period; this mode is essentially being suspended overnight (due to few contacts) with the spanning trees being completed in the morning.

To test the infection rate, an SIR model is constructed⁵ and a disease is spread through the contact network starting at time index 250. The simulation is repeated 30 times for each node and the results bootstrapped to give estimates of the mean number of people susceptible (i.e. those that have not received the disease) at time, t , $S(t)$. Figure 11 shows the results of these simulations and as can be seen the number

⁴We cannot be sure as the data has been anonymised.

⁵Probability of infection 0.5; infection time Poisson distributed with mean 80 time steps, 800 mins.

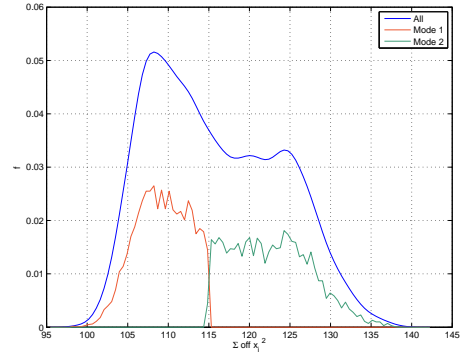


Fig. 14. Distribution of δ_i (MIT).

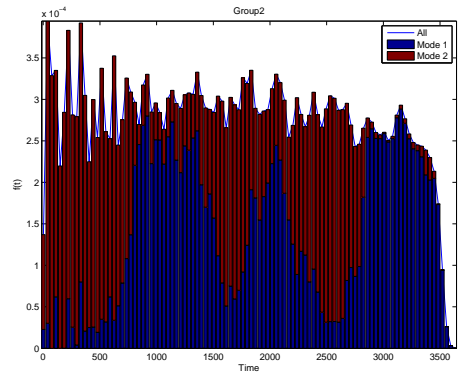


Fig. 15. Distribution of times by mode. (MIT)

of susceptible people falls most rapidly for infections started at nodes 3 and 20, as expected.

B. MIT data

The results from the MIT data set show a different type of behaviour. There are two main groups (Figure 12) the largest of which can be further subdivided

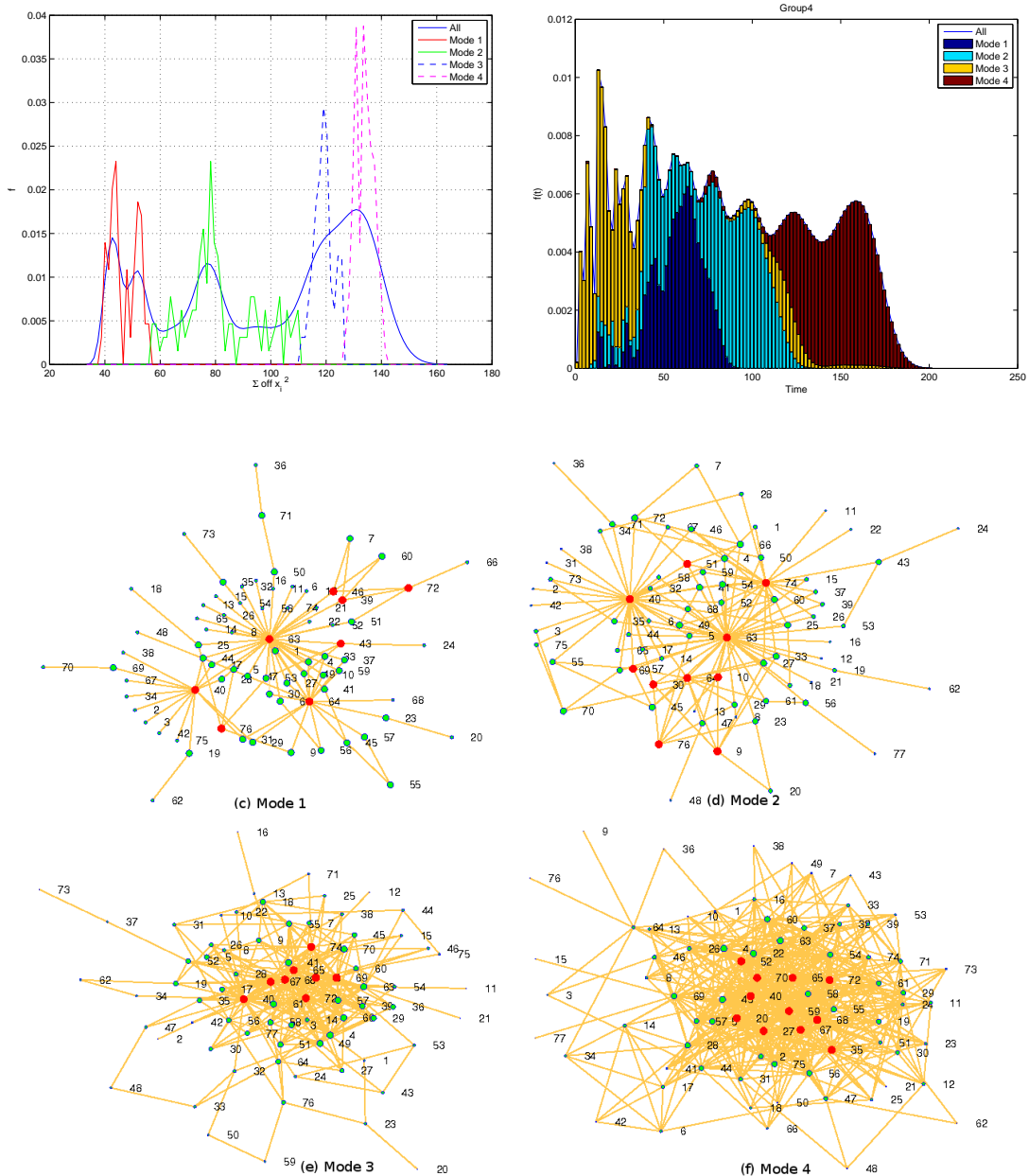


Fig. 16. Infocom06 results. (a) Distribution of δ_i . (b) Distribution of times by mode. (c-f) \bar{H} for each mode.

into three smaller groups (Figure 13).

The distribution of δ_i is shown in Figure 14 and two main modes are identified from this. The MIT data set spans a month of data and recurring patterns emerge from the data as shown in Figure 15. This is particularly interesting as it introduces the concept of being able to forecast the behaviour of a group at regular intervals and design strategies for those specific modes.

C. INFOCOM '06 data

The Infocom data is summarised in Figure 16 and follows the behaviour typically expected at a confer-

ence. Four modes are identified (Figure 16(a)) which correspond to four periods in time (Figure 16(b)). The first mode to occur is mode 3 showing much mixing between the delegates (Figure 16(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 16(c,d)) ending with a period of mixing (Figures 16(f)).

D. 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 18

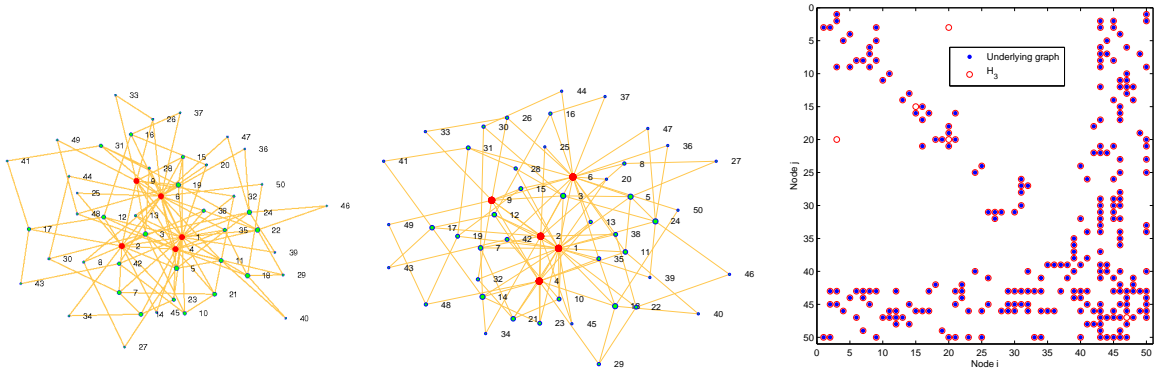


Fig. 17. (a) The generating graph for submode 2*. (b) \bar{H} shortest path graph for submode 2*. (c) A spy plot of the generating graph (.) and \bar{H} (o)**. (Synthetic data) * The size of a node is proportional to the sum of weights incident on that node. ** The rows and columns have been permuted 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 ($\bar{H} < 0.1 \mapsto 0$).

shows the distribution of δ_i for this network is unimodal as expected; there is only one underlying process. The distribution also follows a Γ distribution⁶.

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 [9] [15]); 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 20. Specifically, the first generator employs a Waxman topology [22] ($\alpha = 0.5, \beta = 0.3$)⁷. The second generator is also a

⁶ 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.

⁷Waxman 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 .

Waxman model with $\alpha = 0.7, \beta = 0.3$. The third and fourth generators are (GLP) generalised linear preferential topologies based on preferential attachment [17].

As can be seen 3 modes are detected in the data (Figure 19). These correspond with the generator times for 2 of the modes (Figure 20). However, mode 3 incorporates both generator 3 and generator 4. This occurs as generator 3 and 4 are quite similar (both based on GLP).

The samples in mode 3 may be examined separately using JD to produce the *submodes* seen in Figure 21 occurring at the times seen in Figure 21. 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 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.

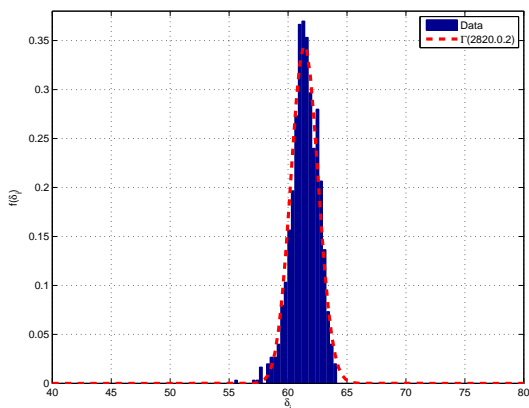


Fig. 18. Distribution of δ_i (Random network).

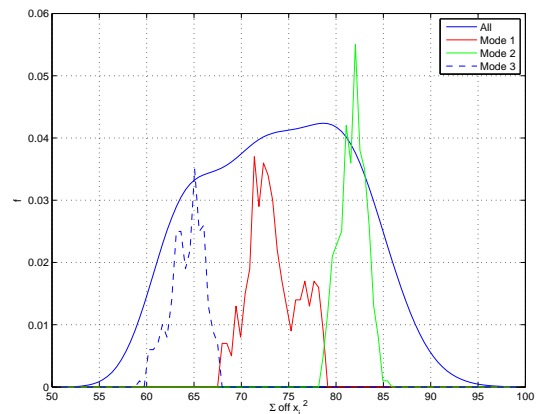


Fig. 19. Distribution of δ_i (Synthetic).

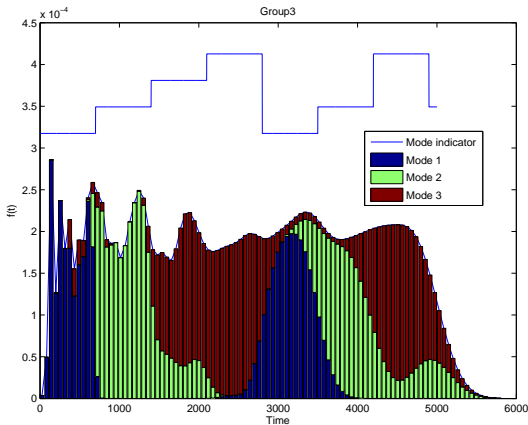


Fig. 20. Distribution of times by mode. (Synthetic)

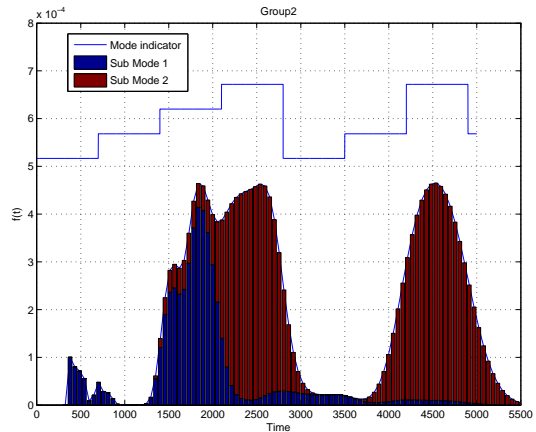


Fig. 22. Distribution of times by mode; submode. (Synthetic)

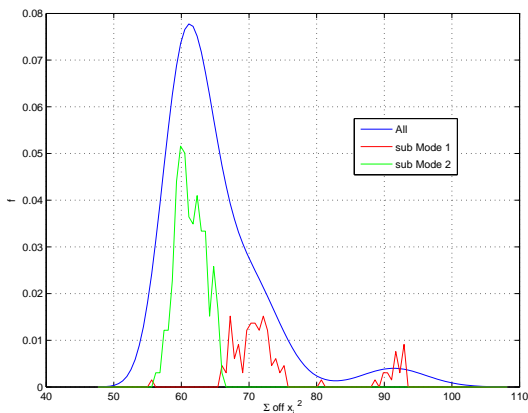


Fig. 21. Distribution of δ_i ; submode. (Synthetic).

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 were extracted including detection of a bridge in the Cambridge data set. The MIT data set in contrast, showed a repetitive behaviour which is useful for prediction of network behaviour; for example in advance of an infection. 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 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. 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. It is hoped that in future this technique will aid in the

design of time specific algorithms for time dependent networks.

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] U. Brandes and T. Erlebach, editors. *Network Analysis: Methodological Foundations*, volume 3418 of *Lecture Notes in Computer Science*. Springer, 2005.
- [2] A. Bunse-Gerstner, R. Byers, and V. Mehrmann. Numerical methods for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 14:927–949, 1993.
- [3] J.-F. Cardoso, J. Fran Cois Cardoso, and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM J. Mat. Anal. Appl.*, 17:161–164, 1996.
- [4] J.-F. Cardoso and A. Souloumiac. Blind beam-forming for non gaussian signals. *IEE Proceedings-F*, 140:362–370, 1993.
- [5] Dartmouth College. A community resource for archiving wireless data at dartmouth, <http://crawdad.cs.dartmouth.edu/index.php>, 2007.
- [6] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, V10(4):255–268, May 2006.
- [7] EU FP6 Huggle Project. <http://www.huggleproject.org>, 2010.
- [8] M. Fiedler. Algebraic Connectivity of Graphs. *Czechoslovak Mathematical Journal*, 23:298–305, 1973.
- [9] 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.
- [10] 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.
- [11] S. Ioannidis, A. Chaintreau, and L. Massoulié. Distributing content updates over a mobile social network. *SIGMOBILE Mob. Comput. Commun. Rev.*, 13:44–47, June 2009.
- [12] 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).
- [13] G. J. McLaughlan. *Finite Mixture Models*. Wiley, 2000.

- [14] T. Opsahl, F. Agneessens, and J. Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245 – 251, 2010.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] M. Wax and J. Sheinvald. A least-squares approach to joint diagonalization. *IEEE Signal Processing Letters*, 4(2):52–53, 1997.
- [22] B. Waxman. Routing of multipoint connections. *IEEE J. Select. Areas Commun.*, 6(9):1617–1622, 1988.
- [23] E. Yoneki. Visualizing communities and centralities from encounter traces. In *ACM MobiCom Workshop on Challenged Networks (CHANTS)*, 2008.