Context Based Identification of User Communities from Internet Chat

Ata Kabán
School of Computer Science
The University of Birmingham
Birmingham, B15 2TT, UK
E-mail: A.Kaban@cs.bham.ac.uk

Xin Wang
School of Computer Science
The University of Birmingham
Birmingham, B15 2TT, UK
E-mail: X.C.Wang@cs.bham.ac.uk

***Abstract*— We study the temporal connectivity structure of single-channel Internet-based chat participation streams. Somewhat similar to bibliometric analysis, and complementary to topic-analysis, we base our study solely on context information provided by the temporal order of participants' contributions. Experimental results obtained by employing both network-analysis indicators and an aggregate Markov modelling approach indicate the existence of distinguishable communities in the about one day worth real-world chat dynamics analysed.**

I. Introduction

With the increase of Internet-based on-line communication, such as Internet chat, the need for organising and structuring such processes has arisen. Previous work [10], [12], [11] has looked exclusively at analysing the text streams produced, in order to reveal the evolution of topics that underlie such discussion streams and possibly to provide a topographically organised visual summary of this process [12].

Here we address a different issue. Making abstraction from the actual text content of the contributions, we analyse the temporal connectivity structure produced in a single-channel Internet relay chat room and seek to investigate whether we can identify sub-communities or groupings amongst the participants. In a somewhat similar manner to web-based connectivity analysis studies, in this paper we base our analysis solely on context information that is provided by the order of activity of the participants.

Besides research purposes regarding the statistical analysis of web-based communication activity, finding connections between users may also be useful for practical purposes, such as splitting and organising participants in separate channels or providing personalised views or interfaces may be desirable with large numbers of participants with heterogeneous interests. Although Internet chat data is considered here, similar technologies could potentially be exploited in any computer-supported cooperative work, e.g. as a flexible educational resource. The possibility that computer-based conversational streams are automatically recorded and can further be analysed with the aid of various machine learning tools may provide valuable ways of further development of computer-based technology.

The remainder of the paper is organised as follows: Section II presents details about the data and problem setting. In Section III, the distribution of the participation frequencies in analysed. Section IV provides a simple network-analysis of the transition connectivity graph. A probabilistic clustering approach and results of community identification are presented in Section V and finally we conclude our study in the last section.

II. Chat recordings and transition connectivity

Internet based chat lines produce a temporal sequence of records of the form

$$< \text{username} > \text{ contributed text} \qquad (1)$$

generated by chat participants, including the chat moderator.

As mentioned, previous work has aimed at uncovering topics from the stream of text only, making abstraction from the user identity. Here in turn we make abstraction from the actual text content of the contributors and explore the context offered by temporal connectivity only. Temporal connectivity to some extent may also correspond to topical connectivity, as if participant i follows participant j then with high probability there is a topical connection between their contributed text, however, this doesn't imply a trivial correspondence between behaviour-based clusters and topical clusters.

Thus, the data that will be analysed here is a temporal sequence of symbols, where each symbol corresponds to a unique userID. A graph may easily be constructed based on the temporal order of these symbols. Nodes of the graph would correspond to userIDs whereas directed edges would indicate the strength of connections based on the frequency of a users following each other. Then, densely connected subgraphs would correspond to user communities. Although this scenario is a simplified one, as in reality a contribution may in fact come as a reply to an earlier contribution, we will adopt this first-order abstraction here, leaving more sophisticated possibilities for further research. From the results presented in the next sections, this setting seems adequate for a first study.

The results reported are based on real-world chat data collected[1] from Internet chat lines. It consists of a continuous

[1]The chat data has been collected and preprocessed by Ella Bingham, Helsinki University of Technology and first utilised in [11].

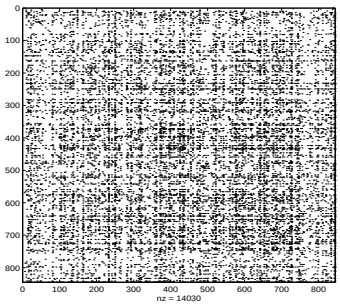

Fig. 1. The transition count matrix of the chat dynamics under investigation. UserIDs are listed in alphabetical order, no structure is apparent.

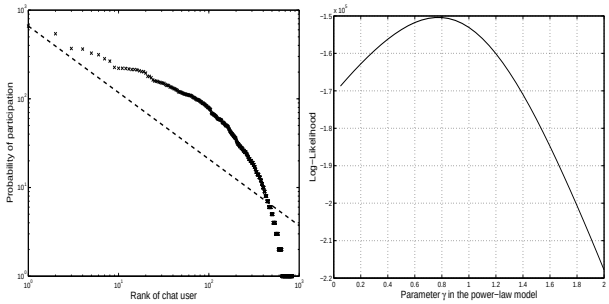

Fig. 2. The left-hand plot depicts on a log-log scale, the frequencies of participation for each user, in decreasing order of magnitude. The right-hand plot shows the data log-likelihood as a function of the unknown parameter γ, under a power model, used to find the ML estimate of he parameter γ. The power-law model, at $\gamma^{ML} \approx 0.75$ is then superimposed on the left figure (dashed line).

stream of about one day of discussions, totaling T=25,355 contributions from S=844 different chat participants.

Figure 1 shows the matrix of first order transition counts in our data. Participants are listed in alphabetical order on both rows and columns and each entry (i, j) represents the number of times urseID=i has followed userID=j. It is a quite sparse connectivity matrix, having only 14,030 non-zero transition entries.

The left plot of Figure 2 shows the participation frequencies for each user, ranked in descending order of magnitude, on a log-log scale. Before proceeding at analysing the transition structure, a brief analysis of the distribution of participation frequency counts is provided in the next section.

III. Analysis of participation frequencies

Figure 2 reveals an approximately linear relationship between $\log P(Rank)$ and $\log Rank$. That is, the ranks approximately follow a power-low distribution.

$$P(r|\gamma) \propto r^{-\gamma} \tag{2}$$

where $r = 1 : S$ are the possible values of the Rank variable, S is the number of chat participants and γ is the single parameter of this distribution. By making the standard iid. assumption, we can easily determine a Maximum Likelihood (ML) estimate of γ by plotting the log likelihood of the data under an iid. power-pow model against γ and reading its maximum argument from the plot. This is

$$\mathcal{L}(\gamma) = \sum_{r=1}^{S} \log P(r|\gamma)^{n_r} \tag{3}$$

$$= -\sum_{r=1}^{S} n_r \left\{ \gamma \log r + \log \sum_{r'=1}^{S} r'^{-\gamma} \right\} \tag{4}$$

where n_r are the participation frequency counts of the user ranked r. The power-model likelihood (4) is shown on the right plot of Figure 2 and we find that it is maximised at $\gamma^{ML} \approx 0.75$. The power law distribution corresponding to this value is then superimposed on the left plot of the same figure.

TABLE I

Clustering coefficient and average path length of the chat connectivity network compared to those of two extreme topologies.

Topology	C	L
regular lattice	0.75	17.0057
chat user topology	0.5811	4.2172
random graph	0.0294	2.0982

IV. Network-analysis of the first order temporal connectivity

Following up from the last section, an inspection from network analysis perspective will provide us useful insights regarding the inherent structure of the connectivity-data that we wish to analyse. In this section, we employ simple numerical descriptors developed in network-analysis studies [2] to show that, similarly to a variety of complex networks such as biological, technological and social networks — including the WWW [1] — the first order connectivity network of our data, (Figure 1) exhibits the so called small-world characteristics. That is, it exhibits a high local clustering coefficient compared to a random network and low average path length compared to a regular lattice — almost as low as a random network. These two coefficients are defined as follows:

- If a vertex v has k_v outgoing and incoming neighbors, then at most $k_v * (k_v - 1)$ directed edges can exist between those. Let C_v denote the existing fraction of edges out of these allowable edges. Then the local clustering coefficient C is defined as the average of C_v over all v.
- The average path length L is defined as the number of edges in the shortest path between two vertices, averaged over all pairs of vertices.

The values that we computed for the chat connectivity graph in relation with those found for a comparable regular lattice and a comparable random graph are shown in Table I. L for our graph has been computed using Dijkstra's algorithm and the full histogram of the values obtained in this computation are

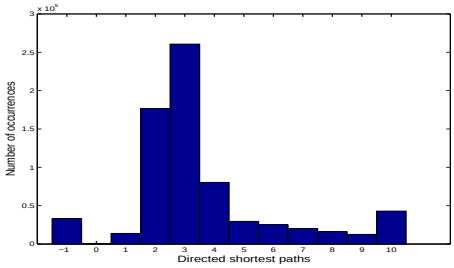

Fig. 3. Histogram of shortest path lengths (measured in number of links) between any two chat users. As can be seen, most of them contain 2–3 hops.

shown on Figure 3. Unrealisable paths have been discarded.

For computing C and L of the two comparable reference topologies (regular lattices and random graphs), we have used the following results given in [2]: Denoting by S the number of vertices and by E the average number of edges per vertex, then for a regular lattice we have $L \sim S/2E \gg 1$ and $C \sim 3/4$ (an empirically determined value). For a random graph, $L \approx L_{random} \sim \ln(S)/\ln(E)$ and $C \approx C_{random} \sim E/N \ll 1$. We use S and E derived from our data, that is $S = 844$ and $E \approx 24.81$ in these computations to obtain values for these two extreme topologies for comparison.

It is clear from the table that $C_{chat} >> C_{random}$ and $L_{chat} << L_{regular}$ and both C and L for our data fall between the corresponding C and L values of the two reference topologies. Thus, we can conclude that the chat connectivity network exhibits the small-world property: highly clustered like a regular graph, yet with small characteristic path length, like a random graph.

Although the clustering coefficient in this analysis is a local property only, the locally clustered topology corroborated with short average path lengths suggest that there may be densely intra-connected but weakly inter-connected subgraphs in this network. In other words, we may find groupings like communities amongst the nodes in spite that this is hardly visible from Figure 1. This hypothesis will be investigated in the next section.

V. Finding communities

In this section we employ a probabilistic model for clustering the states of our first order Markov transition matrix. This model has been formulated and utilised previously for both language modelling — as a class-based bigram model [6], [5] — known as aggregate Markov model and for bibliometric analysis [3], as a probabilistic version of the HITS algorithm, known under the name of probabilistic HITS (PHITS). In this model, clustering of states is achieved by estimating a first order discrete Markov probability transition matrix in a compressed form, invloving a 'bottleneck' latent variable. The latent variable may further be used for explanatory purposes, such as inferring groupings amongst states or amongst transitions. We prefer this approach to graph-based clustering approaches because it is easily extendable in our further work (eg. to clustering states of higher order temporal models).

Let us denote by $X = x_{1:T}$ a sequence of symbols of length T which for convenience will be modelled as a homogeneous first-order Markov chain. That is, for all $t \in \{1, ..T\}$, $P(x_t|x_{t-1}, x_{t-2}, ..., x_1) = P(x_t|x_{t-1})$ and this is independent of time. Although this is clearly an idealised assumption, it has often been found most useful by its simplicity.

The aggregate Markov transition probability model [5] that we employ here is then the following:

$$P(x_t|x_{t-1}) = \sum_{k=1}^{K} P(x_t|k)P(k|x_{t-1}) \tag{5}$$

where $k = 1 : K$ is the discrete domain of possible values of a hidden explanatory variable of the model. We can write the log likelihood of the whole sequence as follows.

$$\mathcal{L} = \sum_{t=2}^{T} \log \sum_{k=1}^{K} P(x_t|k)P(k|x_{t-1}) \tag{6}$$

$$= \sum_{i=1}^{S}\sum_{j=1}^{S} n_{ij} \log \sum_{k=1}^{K} P(i|k)P(k|j) \tag{7}$$

where S denotes a finite alphabet size for now (the number of participants) and n_{ij} is the number of times symbol (user) i has followed symbol j. We can see that maximising the log likelihood is equivalent to minimising the cross entropy, or, up to a constant, the Kullback-Leibler distance between the data and the model.

A ML estimate of the parameters of this model ($P(i|k)$ and $P(k|j)$, $i, j = 1 : S$) can be obtained by maximising (6) under the constraints that the parameter values must be probability values: $\sum_{i=1}^{S} P(i|k) = 1$ and $\sum_{k=1}^{K} P(k|j) = 1$. To enforce these constraints, we construct the Lagrangian, by adding terms with Lagrange multipliers to $\mathcal{L}$. This is then maximised by computing the partial derivatives with respect to the two parameters of the model, $P(i|k)$ and $P(k|j)$ and equating them to zero. These equations can be solved by fixed point iterations, as in [13] leading to the following alternating iterative algorithm:

$$P(i|k) \propto P(i|k) \sum_{j=1}^{S} \frac{n_{ij}}{\sum_{k'=1}^{K} P(i|k')P(k'|j)} P(k|j) \tag{8}$$

$$P(k|j) \propto P(k|j) \sum_{i=1}^{S} \frac{n_{ij}}{\sum_{k'=1}^{K} P(i|k')P(k'|j)} P(i|k) \tag{9}$$

The proportional notation is adopted here for the ease of exposition, both quantities need to be normalised after the update and the normalisation factor comes as the effect of the Lagrange multipliers.

Each fixed point iteration is guaranteed not to decrease the log likelihood and convergence to a local optimum can be achieved by alternating these two parameter updates, using the last parameter value obtained.

Although these multiplicative updates are simple and easy to implement, the popular solution that can be found in the literature for estimating aggregate Markov models [5], [3], [7] is the Expectation-Maximisation (EM). This is the following.

For obtaining an EM solution, an auxiliary function is first created for (6) in the standard way, by employing Jensen's inequality, which lower bounds the log-of-sums by a sum of logs as follows:

$$\mathcal{L} = \sum_{i=1}^{S}\sum_{j=1}^{S} n_{ij} \log \sum_{k=1}^{K} Q_{ij}(k) \frac{P(i|k)P(k|j)}{Q_{ij}(k)} \quad (10)$$

$$\geq \sum_{i=1}^{S}\sum_{j=1}^{S} n_{ij} \sum_{k=1}^{K} Q_{ij}(k) \log \frac{P(i|k)P(k|j)}{Q_{ij}(k)} \quad (11)$$

for any $Q_{ij}(k) \geq 0, \sum_k Q_{ij}(k) = 1$. Now solving the stationary equations with respect to all parameters $Q_{ij}(k)$, $P(i|k)$ and $P(k|j)$ and taking into account all constraints, we arrive at the EM algorithm given in [5], [3], [7].

- E step:

$$Q_{ij}(k) = P(k|i,j) \propto P(i|k)P(k|j) \quad (12)$$

- M step:

$$P(i|k) \propto \sum_j n_{ij} P(k|i,j) \quad (13)$$

$$P(k|j) \propto \sum_i n_{ij} P(k|i,j) \quad (14)$$

Because $Q_{ij}(k)$ is the exact posterior $P(k|i,j)$, than the lower bound is a so called an auxiliary function. That is, the bound can be made arbitrarily tight in each E step, so maximising the auxiliary function will always increase the data likelihood (6) and the maximum of the auxiliary function corresponds to a maximum of the data likelihood. Again, we have an alternating iterative solution, where any of the iterations can be shown not to decrease the likelihood. The E-step utilises the parameter values previously computed in the last iteration's M-step and the M-step uses the posterior previously computed in the E-step.

We can also observe a simple relation between the gradient-based fixed point (8)-(9) and the EM (12)-(14) solutions: By replacing the expressions for computing the posteriors from the E-step into the M-step of the EM procedure (note that now we need to take denominators of the posteriors also into account as these are not the same for all k or i), we recover (8)-(9). In other words, algorithmically, the only difference between the gradient-based fixed point and the EM solutions is that unless recomputing the posteriors after both parameter updates, the EM will use the old value of $P(i|k)$ when updating $P(k|j)$ — i.e. the one computed in the previous iteration rather than the newly updated value obtained in the current iteration. The gradient-based fixed point method in turn always uses the most recent parameter value. This small difference may also be a source of slower convergence of the EM, compared to the gradient-based alternating fixed point algorithm and we shall see that indeed this is the case in the next subsection.

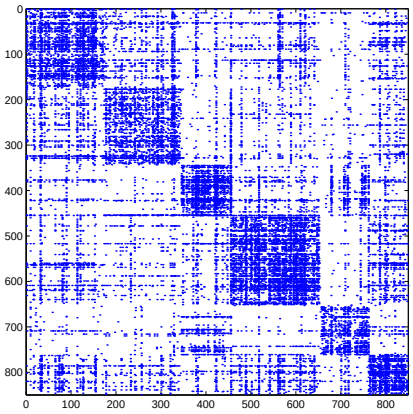

Fig. 4. Clustering result of the chat user participants.

It is also worth noticing that both algorithms scale linearly with the number of observed non-zero transitions, as sparsity of the data can be usefully taken into account in the implementation.

A. Results

Applying the aggregate Markov analysis to our data, we seek to rearrange the symbols (userIDs) such as to reveal connected subgraphs of the overall transition graph. To determine class memberships, we could either threshold the parameters $P(k|j)$ directly — this is the approach taken in [5] — or we can use Bayes theorem, which involves both parameters $P(i|k)$ and $P(k|j)$ — as in [3]. As can be expected, these two approaches provide slightly different results. We have opted for the latter for the following reason: Note that the parameter $P(k|j)$ is in fact concerned with compressing a recent history by representing it as a hidden explanatory state value k. In a more general model, the recent history taken into account may consist of several symbols at different time steps and it is not this what we seek to cluster. The parameter $P(i|k)$ is in turn concerned with the next symbol only, the one that follows a given history. So it is more logical to invert this and ask for the posterior probability $P(k|i)$. Thus, the cluster memberships will be computed as the following:

$$P(k|i) \propto P(k)P(i|k) \quad (15)$$

where we can compute $P(k) \propto \sum_{i,j} n_{ij} P(k|i,j)$. Then we define the label of symbol (user) i as $\max_k P(k|i)$ to obtain a hard partitioning for clustering.

Figure 4 shows the transition count matrix with the symbols reordered in accordance with the labels produced in the manner just described, on a run with $K = 6$. Clearly, the clustered structure is most apparent. Note that in general, the aggregate Markov representation allows users to belong to more than one group if this increases the overall likelihood. Likewise, disjoint classes will only be formed if this improves

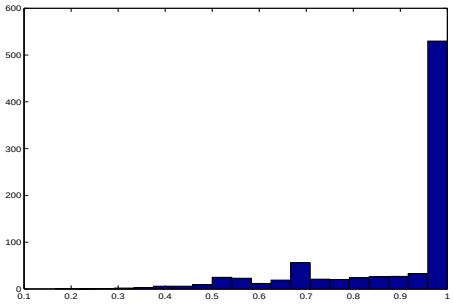

Fig. 5. Histogram of the 'winning assignment' probabilities, $max_k P(k|i)$ for all participants in the chat session analysed.

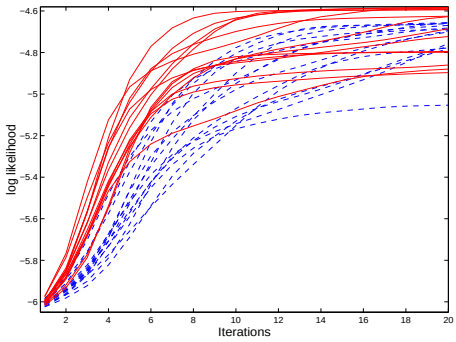

Fig. 7. Convergence speed of the gradient (solid line) versus EM (dotted line) over several random restarts of the iterative algorithm.

the overall likelihood. Figure 5 is meant to empirically assess the degree to which this partitioning is natural for this data by evaluating if any substantial information loss is incurred by the thresholding. The plot shows the histogram of the highest group-membership probability for each individual, $max_k P(k|i)$. Larger values towards one represent more certain community-membership values, i.e. bins of user-IDs for which the information loss when thresholding is lower. At smaller values in the histogram, the information loss is greater. It is apparent from this plot, that — unlike in language modelling applications of the aggregate Markov model [5] — for the chat usersID sequence considered here the histogram is significantly biased towards large values, thus indicating low information loss when thresholding.

Apart from cluster membership values, probabilistic quantities indicating analogous notions to hubs and authorities can also be computed from the model, similarly to [3]. Not surprisingly, the most authoritative user across all groups has turned out to be the chat moderator (userID='cic-cnn').

We also show the split-up of the whole time-course of the chat session on Figure 6, based on the cluster assignments obtained. There appears to be a spread of intensely active and less active time periods for all communities.

Finally, the plot on Figure 7 comparatively shows the convergence of both the gradient fixed point and the EM estimation algorithms over successive iterations, starting from several randomly (uniformly) initialised parameter values. The EM tends to be slightly slower in convergence than the fixed point gradient method. As we have discussed in the previous section, the gradient update equations always utilise the latest updated parameters values whereas both M-step updates of EM for this model use the posterior that has received parameter values from the last M-step only. This causes a slight delay in the speed of convergence.

B. Finding the optimal number of communities

Finally, the issue of model selection should now be addressed. That is, selecting the optimal number of clusters. It should be noted that in spite of the wide popularity of the Aggregate Markov and related models [3], [5], [9], [4], [7] in a number of areas, to our knowledge — except the work of [8] who employs marginal likelihood for model selection for a related but different model — issues of model selection have often been overlooked but and the number of clusters has been set a priori in a rather arbitrary manner. Indeed, the best number of clusters may depend on the application, and a model selection that reflects the objective of the modelling process should be adopted. Unlike in prediction problems, where a model selection criterion should be based on the quality of predictions, our aim here is data-explanation. As such, our motivation is rather driven by the aim of choosing a finite dimensional model that is closest to the possibly infinite dimensional true model in some sense. Following the arguments given in [15], [16], a procedure designed to achieve this goal is the Akaike Information Criterion (AIC) [14] and therefore we adopt AIC for selecting the optimal number of communities in this application. This has a very simple form as follows:

$$AIC = -2\mathcal{L} + 2P \tag{16}$$

where $\mathcal{L}$ is the log likelihood of the model (must not be divided by sequence length), P is the number of free parameters that need to be estimated and the factor of 2 has historical reasons only. For the case of an Aggregate Markov model, we have

$$P = (S-1) * K + (K-1) * S \tag{17}$$

where S is the number of participants, as before, and K is the assumed number of clusters. The optimal model order is then found by minimising (16) under K:

$$K_{opt} = \underset{K}{argmin}\ AIC(K) \tag{18}$$

The log likelihood and AIC-penalised log likelihood values can be comparatively seen on Figure 8 as obtained in 100 randomly (uniformly) initialised independent runs for each choice of K. The maximum log likelihoods values have been selected for each value of K ranging from 2 to 12 and these have then been used to create. the model selection curve shown in the figure. Naturally, the log likelihood continues to increase with increasing the model complexity — however, the AIC-penalised log likelihood peaks at $K = 6$ and thus

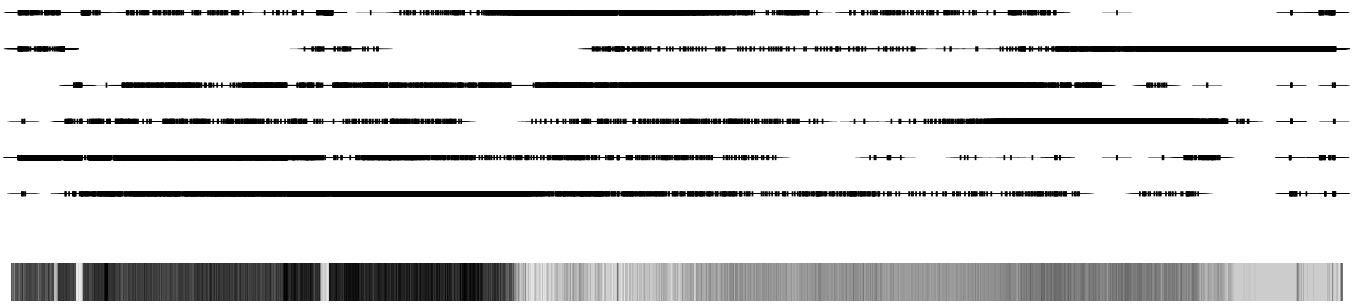

Fig. 6. Time flow of the cluster-contributions. Time is represented on the horizontal axis whereas the six different values on the vertical axis correspond to community label assignments as computed as $argmax_k P(k|i)$. The same information is also given in a color-coded form as it reflects more suggestively the extent of continuities / discontinuities created when contributions from different communities interleave. The six gray-levels (including white) represent community-labels and individual contributions are represented by equally thin lines.

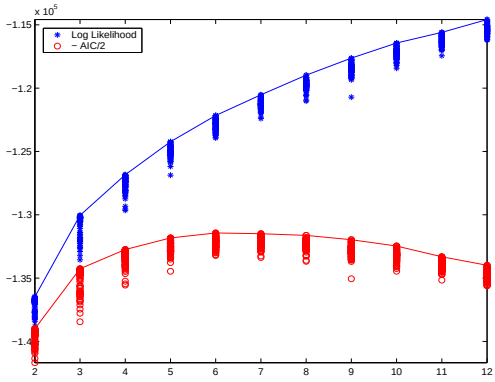

Fig. 8. Log likelihood and AIC-penalised log likelihood plotted against model order. AIC is optimised at $K = 6$ clusters.

this model order is chosen as optimal to explaining our data in terms of the explanatory cluster-variable. The transition matrix with states reordered into the optimal 6 clusters, as found by employing the described procedure, can be seen on Figure 4.

It should be noted that the formula for computing AIC assumes that the parameters have been estimated by Maximum Likelihood. This is most suited for our data-explanatory purposes.

VI. Conclusions and further work

We have presented a first study into the possibility of identifying communities of users from single-channel Internet-based communication streams as defined by dense temporal connectivity sub-structures. Results of analysis of up to first order transition dynamics based on both a network analysis and the probabilistic aggregated Markov modelling approach indicate the existence of communities in these terms.

An interesting question left for further research is how does the structure found purely from context information relate to the dynamics of the topical content of the contributions. We do not expect any significant correlations, as a topic may be discussed in several groups and groupings, however it will be interesting to investigate the extent of variations of topical characteristics accross the identified user communities. Further, maintaining the probabilistic formalism, various useful and flexible combinations of models can possibly be developed and used to better understand and potentially facilitate computer-mediated communication.

References

[1] Lada A. Adamic, and Adar, Eytan, Friends and Neighbors on the Web. Social Networks, 25(3): 211-230, 2003.

[2] Duncan J. Watts, Steven H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393:440–442.

[3] D. Cohn and H. Chang, Learning to Probabilistically Identify Authoritative Documents, Proc 17th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, pp. 167–174.

[4] J.K. Lin, Reduced Rank Approximations of Transition Matrices, Proc AI & Statistics 2003.

[5] L. Saul and F. Pereira, Aggregate Markov Models for statistical language processing, Proc of the Second Conference on Empirical Methods in Natural Language Processing, pp.81–89, 1997.

[6] P. Brown, V. Della Pietra, P. deSouza, J. Lai and R. Mercer, Class-based n-gram models of natural language. Computational Linguistics 18(4):467–479.

[7] T. Hofmann, Unsupervised Learning by Probabilistic Latent Semantic Analysis, Machine Learning, 42, 177–196, 2001.

[8] T. Griffiths, M. Steyvers, Finding scientific topics, submitted.

[9] R. Salakhutdinov, S. Roweis and Z. Ghahramani, Optimisation with EM and Expectation-Conjugate-Gradient, Proc International Conference of Machine Learning 2003, pp. 672–679.

[10] T. Kolenda, L.K Hansen and J. Larsen, Signal detection using ICA: application to chat room topic spotting. In: Lee and Jung and Makeig and Sejnowski (eds): Proc of the Third International Conference on Independent Component Analysis and Signal Separation (ICA2001), San Diego, CA: USA pp. 540–545, 2001.

[11] E. Bingham, A. Kaban and M. Girolami, Topic Identification in Dynamical Text by Complexity Pursuit, Neural Processing Letters, 17: 1–15, 2003, pp. 69–83.

[12] A. Kaban and M. Girolami, A Dynamic Probabilistic Model to Visualise Topic Evolution in Text Streams, Journal of Intelligent Information Systems, 18:2/3, 107–125, 2002.

[13] M. Girolami and A Kabán, Simplicial Mixtures of Markov Chains: Distributed Modelling of Dynamic User Profiles, Advances of Neural Information Processing Systems (NIPS'03), to appear.

[14] H. Akaike, Information Theory and an Extension of the Maximum Likelihod Principle. In Petrox, B. and Caski, F. eds., Second International Symposium on Information Theory, pp. 267.

[15] Joseph E. Cavanaugh, Unifying the Derivations for the Akaike and Corrected Akaike Information Criteria, Statistics and Probability Letters, 31, 201–208.

[16] Brian D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, 1996.