

Enhancing Communities of Interest using Bayesian Stochastic Blockmodels.

Deepak Agarwal*

Daryl Pregibon †

Abstract

Statistical inference for massive graphs is a challenging problem since it is hard to scale the existing statistical models to such settings. However, in many applications one is often interested in understanding the interaction of a single node with the rest of the graph. Recently, a data structure called Communities of Interest was introduced in the literature that represent a graph as a union of small subgraphs, each centered on a fixed node. In this paper, we describe statistical models that could be fitted to these subgraphs and are extensions of Stochastic Blockmodels that exist in the social networks literature. Our method can handle sparseness using shrinkage estimation, weights on edges and could be fitted to a graph consisting of around 100-200 nodes in less than 2 minutes on a 1 GHz PC. The methods are exemplified through a Community of Interest of a customer who has signed up for some service with AT&T.

1 Introduction

Transactional data consists of records of interactions between pairs of entities. Such data can be represented by a graph with nodes being transactors and edges representing the interactions. If the interaction is asymmetric (symmetric), the graph is directed (undirected). Examples include credit card purchases made by customers from merchants, hyperlink between websites on the world wide web, calls between telephone numbers, etc. In this article we are going to focus exclusively on one application, namely the directed graph induced by calls carried on a large telecommunications network involving a carrier like AT&T. (Henceforth nodes are telephone numbers, edges are calls between telephone numbers). Such a graph is massive (hundreds of millions of nodes and edges), sparse and heterogeneous in the sense that some nodes are relatively inactive while others are superactive. Also, our graph is not complete in the sense that we don't get to observe calls originating from carriers other than AT&T. We also have

attributes on nodes (e.g. is a node business or residence?) and edges (e.g. is a call from node i to node j a local call?) Here we assume that the presence/absence of edges between nodes is probabilistic in nature and discuss methods to do statistical inference using generalizations of Stochastic Blockmodels that have been used successfully to model relations in social networks (see, for example [8], [9]). However, the models we use have complexity $O(n^3)$ (n is the number of nodes in the graph) and impossible to scale to our graph. Moreover, the models when applied to the entire network are too simple to account for all the heterogeneity. Also, for most of our applications we are not interested in looking at all the nodes and their relationship with each other, we are more interested in a subset of nodes and how each one in the subset interacts with others in the network. For example, if we want to send a promotional offer to customers in a particular state to sign up for a new service, all we need to focus on are residential customers in that state. How does one study the interactions of a given node with the rest of the network? We use a data structure called Communities of Interest (COI) described in [4]. We will briefly describe the data structure in subsequent sections but loosely speaking a COI of a node X is a subgraph centered on X and consists of nodes connected to X . One could go further and expand the subgraph by recursively looking at nodes interacting with elements in the node set of X (after the first recursion) and so on. For our application we believe a two step recursion starting out at X adequately captures X 's interaction with the rest of the network. The entire graph is a union of COIs centered on nodes and statistical inference is done by fitting a probabilistic model to these COIs. This has several advantages: a) it is scalable (a typical COI in our application has like 100 nodes) b) the models fitted locally to each COI better account for heterogeneity in the data (in fact we could end up using different attributes for different COI's) and c) we only fit models to COIs' of interest (e.g. residential customers in area code 973).

*AT&T Shannon Laboratories, Florham Park, NJ 07932, dagarwal@research.att.com.

†AT&T Shannon Laboratories, Florham Park, NJ 07932, daryl@research.att.com.

2 Communities of Interest

In this section, we give an overview of Communities of Interest introduced by [4] as a way of maintaining signature of nodes in a dynamic graph consisting of hundred of millions of nodes and edges. We note that all notations and equations in this section are borrowed straight from that paper. Before we start, note that the graph in our application is dynamic in nature with millions of nodes and edges appearing everyday and millions of old ones disappearing. To proceed any further, one needs a definition of the graph \mathcal{G}_t at time point t . After discussing several possibilities, [4] settle with the one that allow for a smooth dynamic evolution of \mathcal{G}_t using exponential smoothing. To be more specific, we first define what is meant by linear combination of two graphs g and h

$$G = \alpha g \oplus \beta h$$

where α and β are real scalars. G is a graph whose nodes and edges are union of nodes and edges of g and h and the weight of an edge on G (weight could be an aggregation function like “total duration of calls” or the “number of calls” on an edge at time t) is $\alpha w(g) + \beta w(h)$ with w being zero if the edge is absent from the graph. Using this notation,

$$(2.1) \quad \mathcal{G}_t = \theta \mathcal{G}_{t-1} \oplus (1 - \theta)g_t$$

where g_t is the observed graph at time t and $\theta \in [0, 1]$ that controls the amount of influence history has on the current graph. From (2.1) it is clear that only g_t needs to be stored to update the graph at t resulting in huge storage gains. Expanding the recursion in (2.1) over t it is easy to see that \mathcal{G}_t is a linear combination of g_i $i = 1, \dots, t - 1$ with the weight on g_i being $\theta^{t-i}(1 - \theta)$. Thus, more recent data contributes more heavily to \mathcal{G}_t than older data. Also note that the definition in (2.1) would mean that we could only gain edges over time and loose none. To prevent this from happening, edges where the weight falls below some small threshold are dropped from the graph.

We now describe the actual process of splitting the graph into COIs’. Ideally, a COI should be a singly indexed list of nodes, each with an associated array of weighted directed edges. However for the present application it is neither feasible nor desirable. Instead the graph \mathcal{G}_t is approximated with a new graph $\hat{\mathcal{G}}_t$ whose atomic unit is a subgraph consisting of a node and its directed top- k edges to other nodes where top is relative to the aggregation function on edges. In addition, an overflow node called “other” is defined for aggregating traffic to/from nodes not contained in the

top- k slots. In practice, the edges may not always be retained in a symmetric fashion. For instance, a node like 800CALLATT receive calls from a lot of nodes and hence may find most of the callers absorbed in “other” while nodes calling 800CALLATT may not even exhaust their top- k slots and the edge to 800CALLATT will be in its top- k edge set. Finally, the approximate graph $\hat{\mathcal{G}}_t$ is obtained from $\hat{\mathcal{G}}_{t-1}$ and g_t by applying the top- k approximation to (2.1)

$$(2.2) \quad \hat{\mathcal{G}}_t = \text{top-}k\{\theta \hat{\mathcal{G}}_{t-1} \oplus (1 - \theta)g_t\}$$

2.1 Extended COI: Raw Data for Modeling All models we describe are going to be applied to a COI described above at a fixed time point. To capture the interactions better, the top- k recursion is done twice starting from the central node. However, this is going to miss edges between nodes introduced at the second stage of the recursion. To avoid this artifact, we apply the recursion one more time but only retain edges between nodes that are in the nodeset at the end of the second recursion. Note that the extra recursion also enables us to calculate the total traffic and proportion of total traffic in the “other” bin for every node. We call this new subgraph the “extended COI” and all our models would be applied to this unless otherwise mentioned.

3 Statistical Models

Let G be a directed graph consisting of n nodes and assume we observe a binary relation x_{ij} on the edges. The entire graph consists of $n(n - 1)/2$ pairs $\{(x_{ij}, x_{ji}) : i < j; i, j = 1, \dots, n\}$ called dyads. One can think of a dyad as a 2×2 table obtained by crossing the binary variables x_{ij} and x_{ji} with cell probabilities m_{ij} , a_{ij} , a_{ji} and n_{ij} denoting $Pr(x_{ij} = 1, x_{ji} = 1)$, $Pr(x_{ij} = 1, x_{ji} = 0)$, $Pr(x_{ij} = 0, x_{ji} = 1)$ and $Pr(x_{ij} = 0, x_{ji} = 0)$ respectively. Let $p_{ij} = m_{ij} + a_{ij}$ and $p_{ji} = m_{ij} + a_{ji}$ denote the marginal probabilities $Pr(x_{ij} = 1)$ and $Pr(x_{ji} = 1)$ with the cell probabilities satisfying the constraint $m_{ij} + a_{ij} + a_{ji} + n_{ij} = 1$. The probabilities could be re-parametrized in terms of the log-odds ratio $\rho_{ij} = \log((m_{ij}n_{ij})/(a_{ij}a_{ji}))$, conditional log-odds $\theta_{ij} = \log(a_{ij}/n_{ij})$ and $\theta_{ji} = \log(a_{ji}/n_{ij})$. The θ_{ij} ’s are called conditional log-odds because $a_{ij}/n_{ij} = Pr(X_{ij} = 1|X_{ji} = 0)/Pr(X_{ij} = 0|X_{ji} = 0)$, the odds of having an edge from i to j given there is no edge from j to i . Assuming the dyads to be independent, the likelihood of the graph is

$$(3.3) \quad \text{Kexp}\left(\sum_{i < j} \rho_{ij} x_{ij} x_{ji} + \sum_{i \neq j} \theta_{ij} x_{ij}\right)$$

where $K = \prod_{i < j} (1 + \exp(\theta_{ij}) + \exp(\theta_{ji}) + \exp(\theta_{ij} + \theta_{ji} + \rho_{ij}))^{-1}$

3.1 The p_1 Model In a seminal piece of work, [7] proposed the p_1 model obtained by plugging in $\rho_{ij} = \rho$ and $\theta_{ij} = \theta + \alpha_i + \beta_j$ in (3.3). The likelihood for the p_1 model is

$$(3.4) \quad K \exp(\rho \sum_{i < j} x_{ij} x_{ji} + \theta x_{++} + \sum_i \alpha_i x_{i+} + \sum_j \beta_j x_{+j})$$

which belongs to the exponential family of models with sufficient statistics $M = \sum_{i < j} x_{ij} x_{ji}$ (number of reciprocated edges in the graph), x_{++} (total number of edges in the graph), $\{x_{i+} : i = 1, \dots, n\}$ (out-degree of node i) and $\{x_{+j} : j = 1, \dots, n\}$ (in-degree of node j). The number of parameters in the model is $2n + 2$, much smaller than the number of data points, namely, $n(n - 1)$. Each of the parameters associated with the statistics have an interpretation in the context of our application which we shall explain below.

- ρ : known as “reciprocity”, it is the tendency of nodes in the COI to return calls. Here it’s assumed to be constant for every edge in the COI and hence called “uniform reciprocity” model.
- θ : the intercept, a measure of sparsity.
- α_i : known as “expansiveness” of node i , it determines the tendency of node i to call other nodes in the COI.
- β_j : known as “attractiveness” of node i , it determines the tendency of node i to receive calls from other nodes in the COI.

Note that one can enrich the reciprocity structure by introducing an additional parameter η_i for node i and writing $\rho_{ij} = \rho + \eta_i + \eta_j$. This is known as the “differential reciprocity” model which translated to our application would mean that the tendency to return calls for any dyad would depend on the nodes involved. We now get an additional set of sufficient statistics $\{M_{i+} = \sum_{i \neq j} x_{ij} x_{ji}, i = 1, \dots, n\}$.

3.2 Stochastic Blockmodels The p_1 described above does not incorporate attribute information we might have for the nodes or edges. For instance, let’s consider a binary node attribute which is 1 if node is an AT&T residential customer and 0 otherwise. This will give rise to a categorical edge attribute with 4 levels,

namely, calls from R to R, R to nonR, nonR to R and nonR to nonR. In an important piece of work, [9] referred to these as blocks. It might well be the case that the number of edges in these four blocks are radically different and if that’s so, this information has to be incorporated into the model. They modified the p_1 model to have a different intercept for each block. Based on domain knowledge, one might even reduce the number of parameters by merging blocks. For instance, we might have a single parameter for the blocks R to R and nonR to nonR. Although [9] just confined themselves to the block structure so that they can use a simple Iterative Proportional Fitting Algorithm to estimate parameters (see [5] for details on the algorithm), there is no reason why one cannot have a general regression structure that would incorporate any sort of edge attributes: continuous, ordinal and nominal. If \mathbf{z}_{ij} denotes the attribute vector associated with the directed edge from i to j , one can enrich the p_1 structure by adding the term $\mathbf{z}_{ij} \gamma$ to θ_{ij} , the vector γ to be estimated from data. In principle, we could even have non-linear functions here (e.g. linear combination of basis functions, generalized additive models etc.) but we have not explored these in this paper. One could also have a regression structure for ρ_{ij} subject to the symmetry restriction $\rho_{ij} = \rho_{ji}$. Again, we have not explored it here. Finally, one could also incorporate node attributes using “caller” and “callee” specific effects. If \mathbf{s}_i and \mathbf{r}_j denote the attribute vectors for i th caller and j th callee, these are incorporated by adding the term $\mathbf{s}_i \gamma_s + \mathbf{r}_j \gamma_r$ to θ_{ij} where γ ’s get estimated from data. The likelihood incorporating edge attributes, caller and callee specific effects is given by

$$(3.5) \quad K \exp(\rho M + \theta x_{++} + \sum_i \alpha_i x_{i+} + \sum_j \beta_j x_{+j} + \gamma'_s \sum_i x_{i+} s_i + \gamma'_r \sum_j x_{+j} r_j + \gamma' \sum_{i \neq j} z_{ij} x_{ij})$$

where K is defined by the expression below (3.3) with $\rho_{ij} = \rho$ and $\theta_{ij} = \theta + \alpha_i + \beta_j + \mathbf{z}_{ij} \gamma + \mathbf{s}_i \gamma_s + \mathbf{r}_j \gamma_r$. The differential reciprocity structure can be incorporated by modifying ρ_{ij} as discussed earlier in the context of p_1 model.

3.3 Shrinkage to combat sparseness in a COI

A typical COI in our application is very sparse with several nodes having zero out-degree or zero in-degree. The optimizer in such cases when applied to the function (3.5) will not converge. In order to counter this problem, we take recourse to shrinkage estimation. Specifically, we assume

$$\begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} \sim^{iid} N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_{2 \times 2}\right)$$

where

$$\Sigma_{2 \times 2} = \begin{pmatrix} \sigma_\alpha^2 & \sigma_\alpha \sigma_\beta \rho_{\alpha\beta} \\ \sigma_\alpha \sigma_\beta \rho_{\alpha\beta} & \sigma_\beta^2 \end{pmatrix}$$

and the variance components $\sigma_\alpha^2, \sigma_\beta^2$ and correlation $\rho_{\alpha\beta}$ are estimated from the data. For differential reciprocity, all three random effects $(\alpha_i, \beta_i, \eta_i)$ are shrunk toward zero and we need to estimate three variance components and three correlations. One can also shrink the regression coefficients γ 's toward zero just as in ridge regression if the corresponding design matrices consist of correlated attributes. We have implemented this but so far it has not led to any improvement for models fitted to our data although it could potentially improve things in other applications. Also note that although we start out assuming the dyads are independent of each other, marginally, edges sharing a common node are now correlated since they either share a common expansiveness or attractiveness parameter which are now random variables instead of fixed parameters.

3.4 Estimating missing edges One important feature of a typical COI is the presence of missing edges triggered by calls we don't see on our (AT&T) network. In principle, we would be able to estimate all parameters in our model as long we have some data on outbound and inbound edges for each node. If we can estimate all parameters, we can predict the probability of seeing a missing edge just like any other observed edge. The best situation is to have edges missing at random. Assuming that's the case, one then has to modify the likelihood in (3.5) to incorporate only the data we observe, i.e. the dyads we observe completely and the dyads we observe partially. While the contribution from completely observed dyads don't change, the ones from a partially observed dyad $\{x_{ij}, \text{missing}\}$ is the marginal likelihood $p_{ij}^{x_{ij}} (1-p_{ij})^{(1-x_{ij})}$ expressed as a function of parameters in (3.5). No change needs to be made to the shrinkage part. There are primarily two sources of missing edges in a COI.

- Local calls:- since there are regions in the United States where AT&T does not offer local service, we often miss these calls. These calls are carried by a local telephone company e.g. SNET in Connecticut. We have only been looking at missed local calls relative to the central node.
- Outbound calls from other carriers:- This is difficult to deal with since we won't see any outbound calls

from such a node. Currently we just assume the worst possible scenario, these edges are absent. We are working on multiple imputation techniques to deal with this in a better way.

3.5 Incorporating Edge weights So far we have assumed the data on the edges is binary but in reality we observe the intensity of calling (measured in minutes/week) between nodes. However, a typical COI is very sparse [e.g., 1] containing about 90-100 nodes with density (density = $(100 \times \# \text{edges}) / (\# \text{nodes}(\# \text{nodes} - 1))$) 2-5%. This means that more than 95% of what we observe are zeroes and fitting some continuous distribution to the weights (like gamma or log-normal) is not feasible due to a big spike at zero. One can imagine fitting a distribution which is a mixture of discrete and continuous components but again a bivariate generalization of such a mixture would be difficult to deal with. To simplify things, we binned the edge weights into L ($L=4$ in our case) bins based on histogram equalization obtained from a random sample of 500 COI's with the weights in the i th bin assigned a score i ($i = 1, \dots, L$). Every dyad now is a 2×2 table with a total of L as opposed to 1 in the binary case with the marginal totals given by the bin scores w_{ij} and w_{ji} on edges $i \rightarrow j$ and $j \rightarrow i$ respectively. Let t_{ij} denote the number of reciprocated edges for the dyad $\{i, j\}$. For the binary case, this is uniquely determined from the marginal totals of the 2×2 table. This is also the case if $\min(w_{ij}, w_{ji}) = 0$ ($t_{ij} = 0$) or $\max(w_{ij}, w_{ji}) = L$ ($t_{ij} = \min(w_{ij}, w_{ji})$). For all other configuration of w 's, t_{ij} is not unique. However, the statistical distribution of all possible values of t_{ij} is a generalized hypergeometric with log-odds ρ_{ij} with density

$$(3.6) \quad f(t_{ij}; w_{ij}, w_{ji}, \rho_{ij}) \propto \binom{w_{ij}}{t_{ij}} \binom{L - w_{ij}}{w_{ji} - t_{ij}} \rho_{ij}^{t_{ij}}$$

(see [3], pages 66-67). Hence, we could impute the average value of t_{ij} computed from the density in (3.6) using the current value of ρ_{ij} in the fitting algorithm. We chose $L=4$ since the cutpoints obtained were very close to 2 mins, 10 mins and 60 mins which, based on our experience are sensible.

4 Model Fitting using Empirical Bayes Method

We estimate all our parameters using Empirical Bayes method as described in [11]. We briefly describe the algorithm below. Let π be the vector of parameters involved in the likelihood (3.5) and Σ dispersion matrix of the random effects $\{(\alpha_i, \beta_i), i = 1, \dots, n\}$. The posterior distribution of π conditional on data and Σ

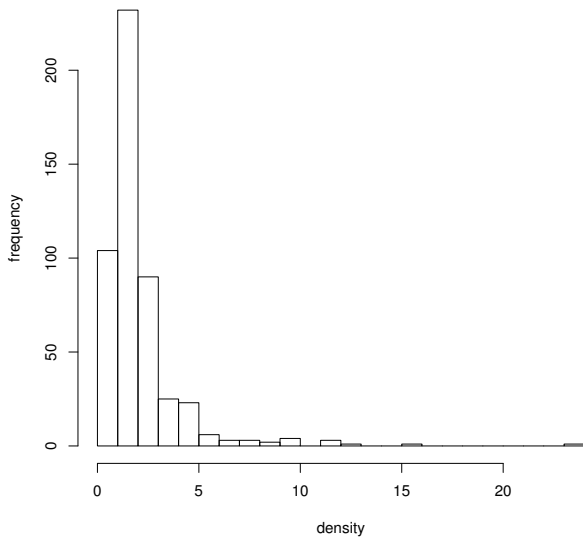


Figure 1: Distribution of density computed using a random sample of 500 COIs'

is proportional to the product of likelihood in (3.5) and the prior normal density. If $\tilde{\pi}$ and $I(\tilde{\pi})$ denote the mode and matrix of second derivatives of log-posterior (evaluated at the mode) respectively, by the central limit theorem the posterior distribution under consideration is approximately normal with first and second moments $\tilde{\pi}$ and $A = -I^{-1}(\tilde{\pi})$ respectively. Note that computing A requires inverting an $P \times P$ (P number of parameters in (3.5)) matrix which is the most expensive computational step and is a major impediment in scaling up this algorithm to networks consisting of thousands of nodes. The value of Σ gets updated using an EM algorithm using observed graph and π as complete data, observed graph as incomplete data (See [11] for details). For weighted data, we add an additional step that imputes t_{ij} 's using the current estimates of ρ_{ij} 's at each iteration. The main steps in the fitting algorithm are summarized below:

- Start with initial values π^0 and Σ^0 .
- At iteration t compute posterior mode $\tilde{\pi}^t$ and matrix $I(\tilde{\pi}^t)$ at Σ^{t-1} .
- Update Σ^{t-1} to Σ^t via EM using $\tilde{\pi}^t$ and $I(\tilde{\pi}^t)$.
- Impute the values of t_{ij} 's from generalized hypergeometric using current estimates of ρ_{ij} 's i.e. ρ_{ij}^t 's.

- Iterate till convergence.

Note that apart from getting point estimates for the parameters, we also get estimates of uncertainty from the information matrix once the algorithm converges. Since n (the number of nodes) in our application is large (like 100-200), the central limit approximation is quite good. The Empirical Bayes strategy we use takes about 1-2 minutes on a 1 GHz Pentium III pc and approximately 20-25 iterations to converge.

4.1 Selecting the starting points The values of all regression parameters (the γ 's) are set to zero. The values of θ and ρ are obtained by maximizing the p_1 model with binary response and setting α 's and β 's to zero. α 's (β 's) are obtained by maximizing the p_1 model with binary response and setting β 's (α 's) to zero. The variance covariance matrix of (α_i, β_i) 's viz. Σ is set to be the sample covariance matrix of the estimated α 's and β 's. For the differential reciprocity model, the η 's are estimated by maximizing the p_1 model with binary response and setting α 's and β 's to zero. To check the robustness of the algorithm to different starting points, we perturbed the values of the initial parameters and found little sensitivity to the choice of initial points. However, we note that poor starting points for Σ can lead to an increase in the number of iterations needed for convergence.

5 Illustration: Fitting models to a single COI

To exemplify the models discussed in the paper, we fit several of these to a single COI of an AT&T long distance residential customer (whom we call X) observed on a particular day. All phone numbers have been masked for reasons of confidentiality. Figure 2 show X's COI (centered on X) using the neato software([6]).

As mentioned earlier, our goal here is to make inference, impute missing edges and also remove some of the clutter that's present in the COI. We perform two recursions in building the COI even though the second step often creates clutter since using just a single recursion might loose important information about X's calling behaviour. For instance, X (who is located in New Jersey) might call his mother in Ohio and his sister in New Jersey who also calls up her mother in Ohio. However, we don't see X's call to his sister on AT&T's network since they are all local calls but we do see calls made by the siblings to their mother. If we don't do the second recursion here, X's sister would not appear in his COI at all.

There are 117 nodes and 172 edges in the COI. Of the 117 nodes, 15 were "local" to X i.e. calls between X and these numbers are local calls. AT&T has started

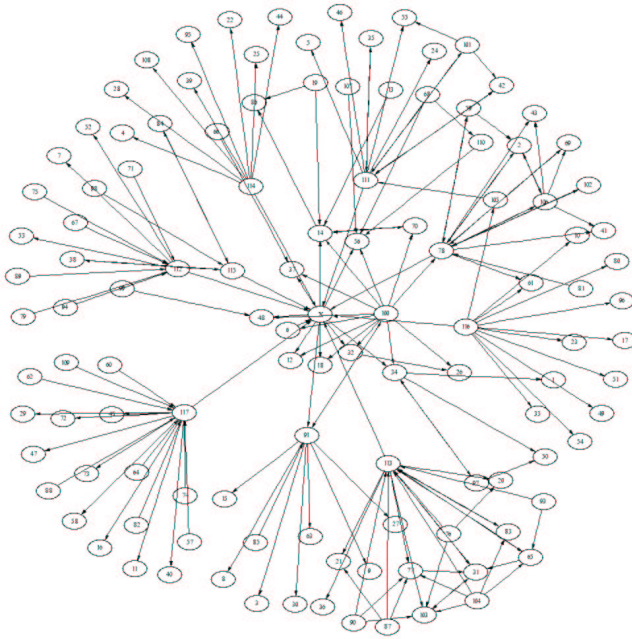


Figure 2: COI of X displayed using a neato layout

providing local service in the state of New Jersey and X has signed up for it. Apart from X, one of the 15 nodes local to X have also signed up for AT&T local. Edges from the 14 remaining nodes to X are local calls not seen on the AT&T network and are treated as missing in the fitting process. Other covariates which we considered for each node was if the node was a business, residential, cell phone or unknown number. For the COI under consideration, there were 59 residential nodes, 34 business nodes and 24 cell phone nodes. We used these as caller and callee specific effects but the effects were statistically insignificant in this case. We converted this to an edge covariate having 9 levels viz. calls from biz to biz, biz to cell, biz to res, cell to biz, cell to cell, cell to res, res to biz, res to cell and res to res respectively. There were no calls from cell to biz and cell to cell. Table 1 gives the total weight corresponding to each of these levels in the COI. We tried two different models:- a saturated one which included all 9 levels (actually 8 assuming res to res to be the baseline) and a second model with cell to biz, cell to cell and cell to res collapsed into a single category. Both these models were fitted assuming uniform reciprocity and the difference in $-2\log$ likelihood for the two models was 3.05 with 2 degrees of freedom which is statistically insignificant at the 5% level. Hence we favor the simpler model (which we call $M1$) in this case. Next, we augmented $M1$ by replacing the uniform reciprocity component with differential reciprocity (we call this $M2$) which reduced $-2\log$

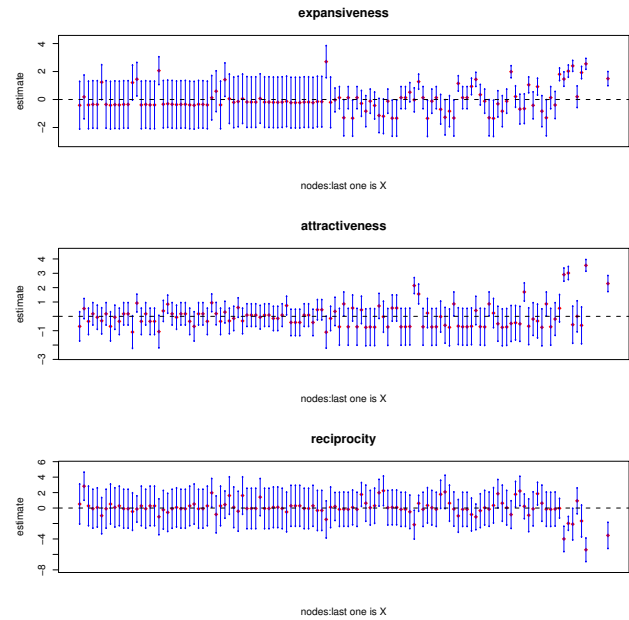


Figure 3: 95% confidence intervals for random effects in model $M2$

Table 1: Total weight assigned to 9 blocks defined by node attribute biz/cell/res

	Biz	Cell	Res
Biz	6	1	20
Cell	0	0	7
Res	120	88	187

likelihood by 206.64, statistically significant at 117 degrees of freedom. Table 2 provide estimates (standard errors within parantheses) for $M1$ and $M2$. The parameter estimates of γ 's are consistent with the weight counts in table 1. They don't have the exact ordering as the weights since the estimates are obtained by simultaneously adjusting for reciprocity, expansiveness, attractiveness and missing edges. Figure 3 shows the 95% confidence intervals for α 's, β 's and η 's for model $M2$ with the dots indicating point estimates. There are quite a few nodes whose reciprocity estimates are significantly different from zero giving more evidence of the appropriateness of the differential reciprocity model.

5.1 Forming blocks a-posteriori The models we fit have distinct random effects (α 's, β 's and/or η 's) for each node in the graph. We partition the nodes into blocks based on the values of estimated random effects. Intuitively, we want nodes belonging to the same block to have similar calling behaviour after adjusting for covariates. In fact, any two nodes belonging to

Table 2: Point and standard error(within parentheses) estimates for parameters in models M1 and M2

Parameter	M1	M2
θ	-5.62(.228)	-5.92(.240)
ρ	3.00(.263)	4.60(.535)
α (for X)	1.03(.326)	1.49(.325)
β (for X)	1.66(.352)	2.28(.357)
η (for X)	—	3.54(1.09)
γ (biz to biz)	-1.54(.547)	-1.33(.505)
γ (biz to cell)	-2.93(1.07)	-2.53(1.07)
γ (biz to res)	-1.85(.377)	-2.08(.381)
γ (merged cell)	-2.68(.513)	-2.77(.510)
γ (res to biz)	.802(.246)	.950(.256)
γ (res to cell)	.951(.271)	1.10(.285)
-2llik	3447.3	3240.7

the same block and having the same node covariates should be stochastically equivalent in the sense that their probabilities of interacting with any other node in the graph are exactly equal provided the edge covariates connecting the two nodes to the third node are same. Extending on the definition provided by [10] in the context of a p_1 model with unifrom reciprocity, we say two nodes are stochastically equivalent (or belong to the same block) if and only if vector (α, β, η) are equal for both. If the model has c blocks (where $c < n$), the number of random effects reduces from $3n$ to $3c$. We will call this model $M3$. The model is fitted using a standard non-linear optimization routine with the mean from each block as initial values for the $3c$ random effects and estimates from $M2$ as initial estimates for remaining parameters.

5.2 Determining blocks We partitioned the nodes into blocks by performing a principal components analysis on the three random effects obtained from $M2$ and splitting the space into $2^2 = 4$ blocks according to whether the first two principal components are positive or negative. The value of $-2\log$ likelihood for model $M3$ was 3745.1 compared to 3240.7 for model $M2$ (see table 2). The AIC(BIC) for $M3$ and $M2$ are 3785.1(4125.7) and 3958.7(10073.0) respectively, clearly showing the superiority of model $M3$. Note that the blocking strategy we use here is simple and could be improved further. For instance, one might start with $c = n$ blocks and combine blocks as in an agglomerative clustering algorithm. We are currently working on better blocking strategies and hope to report on them at SDM05.

Table 3: Size and parameter estimates for model $M3$

Block	Size	α	β	η
1 (-,-)	43	-.087	-1.03	.795
2 (+,-)	17	1.63	-.553	-.995
3 (-,+)	43	-1.84	.343	1.70
4 (+,+)	14	1.47	1.77	-1.96
Other parameters for $M3$				
θ	-5.44			
ρ	4.91			
γ (biz to biz)	-1.83			
γ (biz to cell)	-3.78			
γ (biz to res)	-1.77			
γ (merged cell)	-2.90			
γ (res to biz)	.464			
γ (res to cell)	.034			

5.3 Interpreting blocks Table 3 present the sample sizes, expansiveness, attractiveness, reciprocity estimates of blocks along with other parameter estimates. Sign of the first two principal components corresponding to each block id are indicated within parentheses with the center node X belonging to block 4. Estimates of other parameters do not change significantly except the coefficients to edges from res to biz and res to cell which reduce compared to those in $M2$.

Figure 4 shows the COI with nodes colored where blue, cyan, magenta and black representing blocks 1-4 respectively. The random effects for blocks roughly correspond to low/low, high/low, low/high and high/high on the (α, β) domain. However, nodes in blocks 1 and 3 (the largest of two blocks) are low on both expansiveness and attractiveness and correspond to the uninteresting ones on the periphery of the graph. The real interesting ones in block 4(which contains X) and block 2. The ones in block 4 send and receive a lot of calls but do not tend to reciprocate. Nodes in block 2 are heavy callers, and a few of their calls get reciprocated. The model seem to have gotten them right except a few ones (e.g. couple of blacks, a single cyan on the periphery). In a nutshell, the calling behaviour of X is for the most part captured by the $17 + 14 = 31$ nodes in blocks 2 and 4 and we can get rid of all other nodes in the graph except the block 1 and block 3 nodes that are directly connected to X. The subgraph induced by these nodes is shown in figure 5. We believe this pruned graph is a good compromise between looking at a diameter one COI (start from X and do just one recursion) which loses important information and depth two COI (two recursions starting from X) which introduces a lot of clutter. Interactive visualization software like ggobi, zoomgraph ([1],[2]) are extremely powerful tools to see

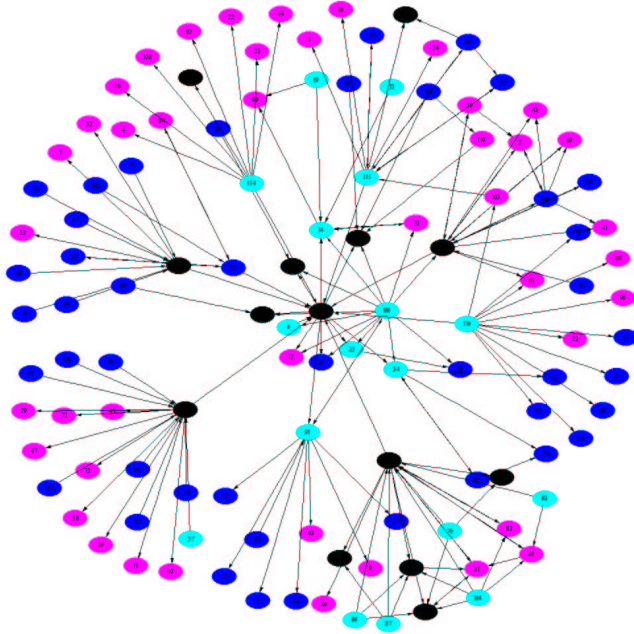


Figure 4: COI of X displayed with node colors from block model

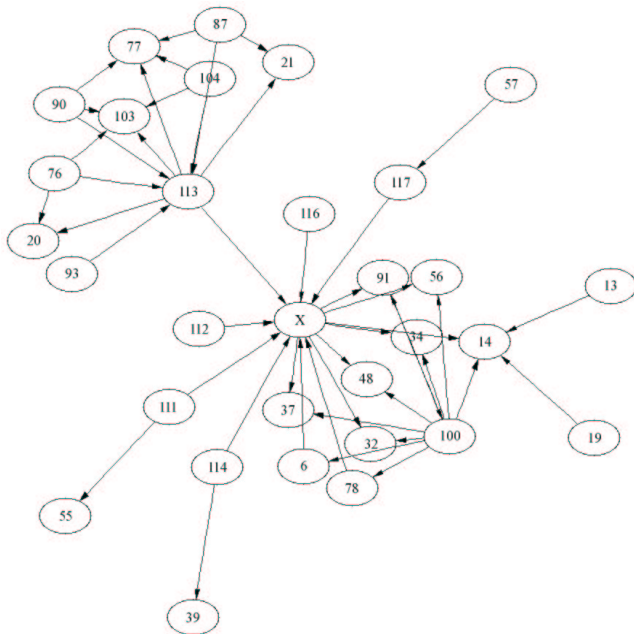


Figure 5: Pruned COI of X

patterns, clusters and various pruned COI's based on model output. One can also understand COI relationships among blocks by computing probabilities of having an edge between node i and j where i and j could belong to the same block or different blocks. In the absence of covariates, this only depends on the block id's of i and j . In our example above, this depends on the block id and whether the node is business, residence or cell.

6 Discussion and Ongoing Work

We have presented a framework for inferring interaction of a node with the rest of the graph by fitting statistical models to a data structure called COI introduced in [4]. The models we presented are extensions of those existing in the social networks literature.

We are currently working on methods to improve the blocking strategy used here. Using the 4 blocks we use here as our starting point, we can merge two blocks (6 possible cases) or split a block into two (4 possible cases) and move to a new configuration with improved BIC. One can repeat the process till the BIC improves no more.

The pruning strategy used here needs further improvement. Some measure of distance of each node from the center node X which is based on the fitted probabilities would be more appropriate than the current method. We are also working on shrinkage strategies to reduce variance when fitting these models to several COI's. Finally, we also need models to capture temporal variation in a COI.

References

- [1] www.ggobi.org.
- [2] www.hpl.hp.com/shl/projects/graphs.
- [3] A. Agresti. *Categorical Data Analysis*. Wiley, John and Sons, Incorporated, 1990.
- [4] C. Cortes, D. Pregibon, and C. Volinsky. Communities of interest. In *Proceedings of Intelligent Data Analysis*, 2001.
- [5] J. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43:1470–1480, 1972.
- [6] E. R. Gansner and S. C. North. An open graph visualization system and its applications to software. *Software - Practice and Experience*, 30:1203–1233, 2000.
- [7] P. Holland and S. Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76:33–61, 1981.
- [8] S. Wasserman and K. Faust. *Social Network Analysis: methods and applications*. New York: Cambridge University Press, 1994.
- [9] Y. Wang and G. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82:8–19, 1987.

- [10] S. Wasserman and C. Anderson. Stochastic a posteriori blockmodels construction and assessment. *Social Networks*, 9:1–36, 1987.
- [11] G. Wong. Bayesian models for directed graphs. *Journal of the American Statistical Association*, 82:140–149, 1987.