Random Graphs Based on Self-Exciting Messaging Activities*

Nam H. Lee† Tim S.T. Leung‡ Carey E. Priebe§

October 31, 2011

Abstract

This paper studies a problem of identifying an inhomogeneous interaction structure amongst social agents. We construct a random graph based on the messaging activities which are modeled as observations from a self-exciting point process. We design a methodology that divides the agents into two disjoint groups so that members within each group are considered to be of similar attributes. Our methodology and algorithm are useful for investigating and detecting abnormal activities within a network. We provide numerical illustrations based on a large email dataset from Enron.

Keywords: social network; self-exciting point process; hypothesis testing; risk mitigation.

1 Introduction

In this paper, we propose a model to estimate and analyze the structure of messaging activities in a social network. This is motivated by the recent proliferation of mobile technology, along with spread of blogs, social networking site, and media-sharing technology. For classification, detection, tracking and other practical purposes, robust statistical analysis as well as a good understanding of the data structure are essential. In this paper, we consider a collection of messaging data, made public by the Federal Energy Regulatory Commission in 2003, that contains highly accurate information about the time at which each message was exchanged. We introduce a meaningful way to reduce messaging data to a random graph and explore its possible application to a community detection problem.

A simple and popular existing method to achieve this is to "pair-wise threshold", where for each pair of agents, an edge between vertex i and vertex j is formed if the number of messaging events between them exceeds a certain threshold. Such graphs are often thought to reveal a structure of an underlying social dynamic, motivating several successful models for a social network with sub-communities. These models are based on parameterizing the distribution of a random graphs, and there have been many tools for detecting a community with a particular graph theoretic and statistical properties. For a survey of such methodologies, see [7] and [9].

On the other hand, some recent research, e.g. [4], has documented that changing the thresholds in the reduction procedure can produce dramatically different graphs, resulting in dissimilar communities. This issue has motivated the work such as [8, 12]. In both studies, as a remedy, the messaging events are modeled by way of point processes. In [8], a piecewise-constant interaction rate is considered, while in [12] a Cox multiplicative intensity model is used with covariates that depend on the history of the process.

In the current paper, we analyze a multivariate self-exciting point process model for occurrence of high risk messages. In addition, we propose a method to construct the associated graph, whose distribution under correct estimation will be close to the distribution of an Erdos-Renyi random graph. This feature distinguishes our methodology favorably from the "pair-wise thresholding" graph whose distribution can be far more complex.

Based on our methodology, we can produce a random graph with a particular structure, and identify a graph partitioning algorithm with good performance (e.g. with high success probability). This is very useful for the purpose of community detection.

*The authors would like to thank Dr. Mihn Tang for his helpful comments. This work is partially supported by NSF Grant DMS-0908295, National Security Science and Engineering Faculty Fellowship (NSSEFF), Air Force Office of Scientific Research (AFOSR), Johns Hopkins University Human Language Technology Center of Excellence (JHU HLT COE).

†Department of Applied Mathematics & Statistics, Johns Hopkins University, Baltimore, MD 21218

‡Department of Industrial Engineering & Operations Research, Columbia University, New York, NY 10027

§Department of Applied Mathematics & Statistics, Johns Hopkins University, Baltimore, MD 21218

The rest of this paper is organized as follows. In Section 2, we formulate a community discovery problem named "vertex nomination" with a focus on a population consisting of at most two sub-communities. In Section 3, we offer a simple approximation argument that yields the first-order solution to the vertex nomination problem. Section 4 provides several numerical illustrations.

2 Vertex nomination based on self-exciting processes

We fix a probability space $(\Omega, \mathcal{F}, \mathbf{P})$ with filtration $\mathbb{F} = (\mathcal{F}_t)_{0\le t\le T}$, satisfying the usual conditions of completeness and right-continuity. We consider a network of n vertices, and denote $\mathbb{V} = \{1, \dots, n\}$.

2.1 Messaging Events and Memberships

Assume that $\mathcal{C}_1$ and $\mathcal{C}_2$ form a partition of $\mathbb{V}$, that is,

$$\mathcal{C}_1 \cap \mathcal{C}_2 = \varnothing, \text{ and } \mathcal{C}_1 \cup \mathcal{C}_2 = \mathbb{V}.$$

As a convention without loss of generality, we will allow that $\mathcal{C}_2$ can be an empty set but not $\mathcal{C}_1$. Our objective is to determine two disjoint groups of vertices so that the members within each group are said to have common risk characteristics reflected by their messaging patterns.

Definition 1. *A vertex $v \in \mathbb{V}$ is said to have* membership $m \in \mathbb{M}$, *denoted by $\lfloor v \rfloor = m$, if $v \in \mathcal{C}_m$.*

In this paper, we need not distinguish the pairs $(\lfloor i \rfloor, \lfloor j \rfloor)$ and $(\lfloor j \rfloor, \lfloor i \rfloor)$. To simplify notation, we will write $\lfloor i, j \rfloor$ for $(\lfloor i \rfloor, \lfloor j \rfloor)$. This allows us to keep track of only two distinct values among the four possible combinations of $(\lfloor i \rfloor, \lfloor j \rfloor)$. For all possible membership combinations, we will use and assume the following relation: $(1, 2) = (2, 1) = (1, 1) \neq (2, 2)$.

For $\ell \in \mathbb{N}$, we denote by τ_ℓ, $\{i_\ell, j_\ell\}$ and k_ℓ, respectively, the occurrence time, the messaging pairs and the risk level of the ℓ-th message. Collectively, $(\tau_\ell, \{i_\ell, j_\ell\}, k_\ell)$ represents the ℓ-th messaging event. Topic k can either be a low risk topic $k = 0$ or a high risk topic $k = 1$. In addition, we require that

$$\tau_\ell < \tau_{\ell+1} \quad \text{and} \quad 1 \le i_\ell \neq j_\ell \le n.$$

Henceforth, we will focus our analysis only on the *high risk* messages (i.e. $k = 1$). For each (undirected) pair ij of the vertices and $t \in [0, T]$, we denote by $\mathbf{N}_{ij}(t)$ the number of (undirected) messaging events on the high risk topic, i.e., $k = 1$ between vertex i and vertex j during $[0, t]$. For each $t \in [0, T]$, let $\mathcal{D}(t)$ be the collection of all communication messaging events by time t on the high risk topic, i.e.,

$$\mathcal{D}(t) = \left\{ (\tau_{1,\ell}, \{i_{1,\ell}, j_{1,\ell}\}, 1) : \ell = 1, \dots, \sum_{i<j} \mathbf{N}_{ij}(t) \right\}.$$

Figure 1 provides an illustration of the messaging records.

2.2 Vertex nomination

We assume that the σ-algebra $\mathcal{F}_0$ does not contain the full information about which vertices are the members of $\mathcal{C}_1$. However, it is assumed that for at least one vertex $v \in \mathbb{V}$, the membership information may be known.

In order to formulate our vertex nomination problem, we fix $C_1 \subset \mathcal{C}_1$ and $C_2 \subset \mathcal{C}_2$ as two disjoint subsets. For each $V \subset \mathbb{V}$, we denote

$$C_1(V) = C_1 \cup V, \quad \text{and} \quad C_2(V) = \mathbb{V} \setminus C_1(V).$$

The purpose of vertex nomination is to determine which vertices are members of $\mathcal{C}_1$ and $\mathcal{C}_2$.

Definition 2. *For any fixed $c \in \{1, \dots, |\mathbb{V} \setminus (C_1 \cup C_2)|\}$, the* candidate set *is defined by*

$$\mathcal{V}_c := \{V \subseteq \mathbb{V} \setminus (C_1 \cup C_2) : |V| \le c\}. \tag{1}$$

Each $V \in \mathcal{V}_c$ is called a vertex nomination, for which the vertex nomination solution *is the partition $(C_1(V), C_2(V))$.*

In other words, the subset $V \in \mathcal{V}_c$ is nominated to class $\mathcal{C}_1$, allowing us to partition the graph into two. Our objective is to employ the messaging events $\mathcal{D}(T)$ to form a basis for our vertex nomination. In addition, a sensible nomination procedure should also use the information on the vertices whose classes are already known in deciding where other vertices belong.

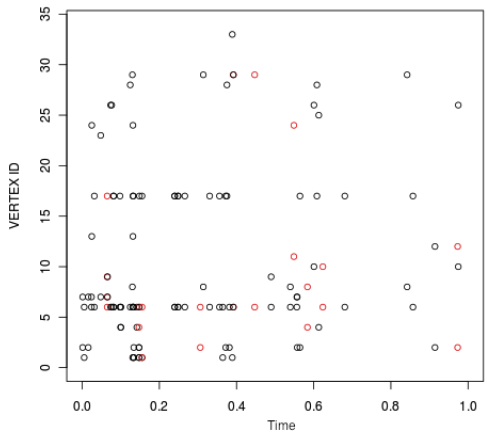


Figure 1: The (pair-wise) messaging records amongst 34 vertices extracted from the Enron Email corpus. Any vertically drawn line starting from the horizontal axis will either cross nothing or cross two circles of the same color. When the vertical line crosses two circles of the same color, for some ℓ, the horizontal axis coordinate represents τ_ℓ, the pair of vertical coordinates of two circles correspond to $\{i_\ell, j_\ell\}$ and the color of the circles corresponds k_ℓ. In this particular example, the red circles represents the high risk messages.

2.3 Multi-variate self-exciting point processes

2.3.1 Data generating process

We observe that people tend to communicate more often during the day than, say, midnight. Hence, it is important to account for the periodic patterns in the intensity function, possibly at different scales, such as daily, weekly, month, and yearly. On top of this, the intensity function is also subject to random fluctuation.

In order to formulate a model that relates the memberships of the vertices to the observation $\mathcal{D}(T)$, we shall consider a reduced-form approach to describe the generation of messaging data. To this end, we utilize the theory of point processes; see e.g. [13, 3].

Let $\mathbf{A}$ and $\mathbf{B}$ be 2×2 constant matrices. We also fix $\omega \in \mathbb{R}_+$, $\mu \in \mathbb{R}^2$, as well as $\Lambda_1, \Lambda_2 > 0$. We assume that $\mathbf{N}$ is a multi-channel self-exciting point process such that its intensity function is given by

$$\lambda_{ij}(t, \mathbf{x}) = \Lambda_{\lfloor i,j \rfloor}(1 + \cos(\omega t + \mathbf{x}_{\lfloor i,j \rfloor})).$$

We assume that the intensity of $\mathbf{N}$ is modulated by "phase-shift" process $\mathbf{X}$ that satisfies the system of stochastic differential equation (SDE):

$$d\mathbf{X}(t) = \mathbf{A}(\mathbf{X}(t) - \mu)dt + \ \mathbf{C}(t)\partial\lambda(t, \mathbf{X}(t))\mathtt{diag}^{-1}(\lambda(t, \mathbf{X}(t)))d\mathbf{Z}(t), \tag{2}$$

where

$$d\mathbf{Z}(t) \ = \ d\mathbf{N}(t) - \lambda(t, \mathbf{X}(t))dt, \tag{3}$$

$$d\mathbf{C}(t) \ = \ \left(\mathbf{AC}(t) + \mathbf{C}(t)\mathbf{A}' + \mathbf{BB}'\right)dt - \ \mathbf{C}(t)\partial\{\partial\lambda(t, \mathbf{X}(t))\mathtt{diag}^{-1}[\lambda(t, \mathbf{X}(t))]d\mathbf{Z}(t)\}\mathbf{C}(t). \tag{4}$$

Note that we choose to suppress the dependence of λ on t for simplicity in our notation, and the partial derivatives are taken with respect to $\mathbf{x}$. Also, note that $\mathbf{Z}$ is the compensated martingale associated with $\mathbf{N}$. We will write $\mathbf{X}(0) = \mathbf{x}_0$ and $\mathbf{C}(0) = \mathbf{C}_0$.

Remark. The preceding system (2)-(4), sometimes called the processor equations, is motivated by the idea of filtering for doubly stochastic processes [13, Chapter 6]. In analyzing a certain doubly stochastic point process whose intensity function is λ and the underlying information process is a diffusion process (with no jump terms), the processor equation can be used as an tractable alternative to the original doubly stochastic process. A key appeal for using processor equations (of the original doubly stochastic process) is the fact that the value of

$\mathbf{X}(t)$ is deducible exactly from the values $\mathcal{D}(T)$ provided that the model parameter is known. □

2.3.2 Auxiliary processes

Next, we introduce a family $\{\mathbf{X}^V\}$ of processes that are computed with $\mathbf{N}$, where V runs over all possible vertex nominations. The family of these processes (conditioning on the known model parameters) is to capture all the possible sample paths that could have yielded the observed messaging data $\mathcal{D}(T)$.

In particular, conditioning on all the true model parameters except the full identification of the vertices' classes, when vertex nomination V is incorrect, the trajectory of $\mathbf{X}^V$ is likely to be different from true path of $\mathbf{X}$ but when vertex nomination V is correct, the path of $\mathbf{X}^V$ coincides with the path of $\mathbf{X}$.

For each i and j, $\lfloor i \rfloor$ and $\lfloor j \rfloor$ denote the "true" membership of vertex i and vertex j. On the other hand, when we talk about the membership of vertex i proposed by some vertex nomination V, we will instead write $\lfloor i \rfloor_V$ for the proposed membership of i. However, when $i \in C_1 \cup C_2$, we will write $\lfloor i \rfloor_V = \lfloor i \rfloor$ regardless of the underlying vertex nomination V.

Definition 3. *Let V be a vertex nomination. The auxiliary process $\mathbf{X}^V$ is the solution to the following system of SDEs:*

$$d\mathbf{Z}^V(t) = d\mathbf{N}(t) - \lambda^V(t, \mathbf{X}^V(t))dt,$$
$$d\mathbf{X}^V(t) = \mathbf{A}(\mathbf{X}^V(t) - \mu)dt + \mathbf{C}^V(t)\partial\lambda^V(t, \mathbf{X}^V(t))\,\mathtt{diag}^{-1}(\lambda^V(t, \mathbf{X}^V(t)))d\mathbf{Z}^V(t),$$
$$d\mathbf{C}^V(t) = \left(\mathbf{A}\mathbf{C}^V(t) + \mathbf{C}^V(t)\mathbf{A}' + \mathbf{B}\mathbf{B}'\right)dt - \mathbf{C}^V(t)\partial\{\partial\lambda^V(t, \mathbf{X}^V(t))\,\mathtt{diag}^{-1}[\lambda^V(t, \mathbf{X}^V(t))]d\mathbf{Z}^V(t)\}\mathbf{C}^V(t),$$

where $\mathbf{X}^V(0) = \mathbf{x}_0$, $\mathbf{C}^V(0) = \mathbf{C}_0$ and

$$\lambda_{ij}^V(t, \mathbf{x}) = \Lambda_{\lfloor i,j \rfloor_V}(1 + \cos(\omega t + \mathbf{x}_{\lfloor i,j \rfloor_V})).$$

For simplicity, whenever it is convenient, we will also write $\lambda_{ij,k}(t, \mathbf{x}|V)$ for $\lambda_{ij,k}^V(t, \mathbf{x})$, and we apply a similar convention for other cases.

2.4 From pair-wise time scaling to random graphs

In this section, we outline how to associate a graph with each of the auxiliary processes. To this end, we employ the random time-change described by the mapping:

$$t \to \int_0^t \lambda_{ij}(s, \mathbf{X}(s)|V)ds. \tag{5}$$

For each vertex nomination V and $\ell = 1, \ldots, \mathbf{N}_{ij}(T)$, we define

$$\xi_{ij}(\ell|V) = \int_{\tau_{ij}(\ell-1)}^{\tau_{ij}(\ell)} \lambda_{ij}(s, \mathbf{X}(s)|V)ds. \tag{6}$$

Then, the theory of point processes (see e.g. Theorem 7.4.V in [3]) yields that, given the vertex nomination V is correct, the sequences

$$\{\xi_{ij}(\ell|V) : i \neq j \in \mathbb{V} \text{ and } 1 \leq \ell \leq \mathbf{N}_{ij}(T)\}$$

form a collection of independent exponential random variables with a common mean one. We emphasize that the model uncertainty is only in the identification of the vertices' classes.

Given a nomination V and the messaging events $\mathcal{D}(T)$, we construct a graph as follows. First, for each ij, we test if $\{\xi_{ij}(\ell|V) : \ell = 1, \ldots, \mathbf{N}_{ij}(T)\}$ is indeed a sequence of independent unit rate exponential random variables by, say, the test statistic $Q_{ij}(T|V)$ (cf. Algorithm 7.4.V in [3]). For example, $Q_{ij}(T|V)$ could be the sample mean of $\{\xi_{ij}(\ell|V) : \ell = 1, \ldots, \mathbf{N}_{ij}(T)\}$. Then, let $G(T|V)$ be the unweighted simple graph where for each pair of vertices i and j, there is an edge between i and j if and only if the p-value for the test statistic $Q_{ij}(T|V)$ is below the level of significance. Henceforth, by the term "goodness-of-fit" graph, we shall mean the graph obtained via the procedure we have just described.

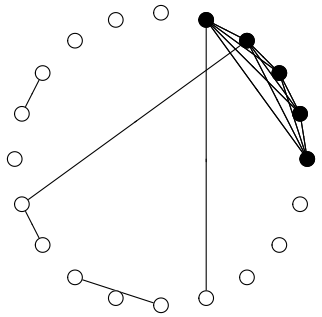


Figure 2: An idealized goodness-of-fit graph $G_0(T|V_0')$. The dark circles represent the $\mathcal{C}_2$ vertices that are incorrectly classified as $\mathcal{C}_1$ by vertex nomination V_0'. The white circles represents $\mathcal{C}_1$ vertices. The edges between the dark and white circles are purely due to randomness, and the same for the edges among white circles.

3 Approximation analysis

To illustrate our methodology, we proceed to describe a simple algorithm for obtaining a vertex nomination solution.

3.1 Algorithm

Our algorithm is as follows. First, let $V_0 = \varnothing$ and then construct "the goodness-of-fit graph" $G_0(T|V_0)$ based on the vertex nomination V_0. Next, compute the subgraph $G_0'(T|V_0)$ induced by $\mathbb{V} \setminus (C_1 \cup C_2)$. Then, let $V_1(T)$ be the community of $G_0(T|V)$ identified by the underlying graph partitioning algorithm as the subgraph "with the highest self connectivity." Using the vertex nomination $V_0' = \mathbb{V} \setminus (C_1 \cup C_2)$ instead, we repeat the process to obtain another "goodness-of-fit graph" $G_0(T|V_0')$ and subsequently the vertex nomination $V_1'(T)$.

Hence, if either V_0 or V_0' is a correct vertex nomination, then we expect to see evidence for both graphs to being simple, i.e., Erdos-Renyi random graphs, one with small edge probability and another with hight edge probability. On the other hand, the behavior of $G(T|V)$ when V is incorrect can be more complex. This is analogous to the hypothesis testing situations in which the alternative hypothesis is composite.

However, we can make some general comments about what to expect when T is sufficiently large. First, ideally, the graph $G_0(T|V_0)$ should have the following properties: (1) vertices incorrectly placed in $\mathcal{C}_2$ are likely to be fully connected, (2) vertex pairs with one incorrectly placed in $\mathcal{C}_2$ and the other correctly placed in $\mathcal{C}_2$ are likely to be connected, (3) any other vertex pair is unlikely to be connected. In contrast, ideally, the graph of $G_0(T|V_0')$ should have the following properties: (1) vertices classified incorrectly as $\mathcal{C}_1$ are likely to be fully connected, (2) vertex pairs with one incorrectly placed in $\mathcal{C}_1$ and the other correctly placed in $\mathcal{C}_1$ are likely to be connected, and (3) any other vertex pair is unlikely to be connected. Hence, ideally, it would be that

$$\mathcal{C}_1 = C_1 \cup V_1(T) \quad \text{and} \quad \mathcal{C}_2 = C_2 \cup V_1'(T).$$

Definition 4. *The performance of a graph partitioning algorithm is the likelihood that the graph partitioning algorithm identifies $V_1(T)$ as $\mathcal{C}_1 \setminus C_1$ and $V_1'(T)$ as $\mathcal{C}_2 \setminus C_2$ respectively given the idealized $G_0(T|V_0)$ and $G_0(T|V_0')$ as its input.*

Note that in theory, the performance of a graph partitioning algorithm may be either zero or one provided that the algorithm is run completely. In practice, due to the running time consideration, the algorithm may have to be terminated prematurely. Our definition above considers this practical issue.

3.2 First-order approximation

We now motivate our algorithm with a simple example from our general model by setting $\omega = 0$. With this specification, the high risk bearing class has the intensity process that is a vertical shift of the base-line class, and the multivariate counting process $\mathbf{N}$ is a time-homogeneous multivariate counting process. To further simplify our derivation, we will focus on the expected behavior of $\mathcal{D}(T)$ as $T \to \infty$.

For each pair ij, the stream of messages between vertex i and vertex j is assumed to be generated by a homogeneous Poisson process, Also, the streams are assumed to be mutually independent from each others.

By standard calculations (cf. [13]), the likelihood that $\mathcal{D}(T)$ is generated by the model specified by V is given by

$$\log f(V|\mathcal{D}(T)) = -\sum_{i<j} \Lambda_{\lfloor i,j \rfloor_V} T + \sum_{i<j} \log(\Lambda_{\lfloor i,j \rfloor_V}) \mathbf{N}_{ij}(T).$$

The next result states that when the only unknown in the model is the membership of the vertices, maximizing the log-likelihood function asymptotically yields the true vertex nomination.

Theorem 1. *For any arbitrary vertex nomination V, we have the following inequality:*

$$\lim_{T\to\infty} \frac{1}{T} \left(\log \frac{f(V|\mathcal{D}(T))}{f(V_*|\mathcal{D}(T))} \right) \leq 0,$$

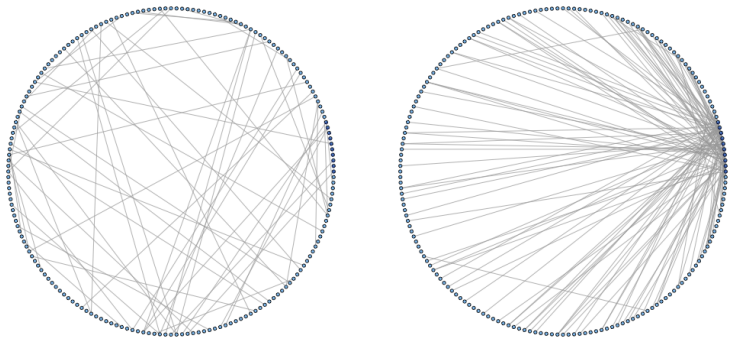
where V_* denotes the correct vertex nomination.

The preceding theorem says that asymptotically the maximum likelihood estimate of V_* is consistent. However, as n gets large, the brute force search of the maximum likelihood estimate quickly becomes intractable. To circumvent this difficulty, one can show that our vertex nomination algorithm is capable of finding the correct vertex nomination provided that the underlying graph partitioning algorithm performs well.

Theorem 2. *Fix a graph partitioning algorithm. For each $\varepsilon > 0$, there exists $T_0 > 0$ such that if $T \geq T_0$, then*

$$\mathbf{P}(V_1'(T) = \mathcal{C}_1 \setminus C_1) \geq (1 - \varepsilon) p_0, \tag{7}$$

where $p_0 \in [0, 1]$ is the performance of the underlying graph partitioning algorithm.



(a) The correct nomination (b) An incorrect nomination

Figure 3: Goodness-of-fit graphs based on a simulation record. The correct nomination graph shows neither apparent concentration pattern nor connectedness. On the other hand, the incorrect nomination graph shows both the apparent concentration around the incorrectly classified vertices and the connectedness.

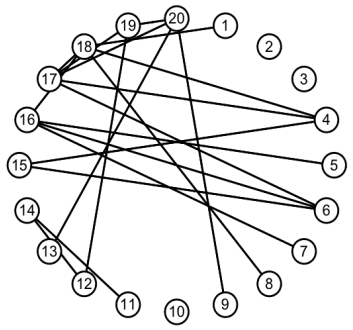


Figure 4: The goodness of fit graph with the vertex nomination that all vertices are in $\mathcal{C}_1$ when $\mathcal{C}_1 = \{1, \ldots, 10\}$. The most apparent signals are from the bipartite relation between two classes.

4 Numerical result

Our first numerical experiment in this section shows that our first-order algorithm work as expected in a situation for which the algorithm is designed. In our second numerical experiment, we show that our first-order algorithm may not maintain its original performance across all possible underlying subcommunity structures which are drastically different from the original structure but may keep some of its original performance to a lesser degree.

4.1 Performance under a vertical shift with no phase shift

In Figure 3a, each gray edge represents an event that the test statistic for the pair has the p-value lower than the pre-specified level of significance 0.0283. In Figure 3b, ten vertices with the most edges are also the vertices whose model specification happens to be the most corrupt.

The particular choice of 0.0283 is motivated by the the connectivity threshold function for an Erdos-Renyi random graph process. See [1] for reference. For a Erdos-Renyi random graph $\mathcal{G}(n, p)$ of n vertices and edge probability p, the threshold function is given by $\log(n)/n$. That is, if $\log(n)/n < p$, then almost surely as $n \to \infty$, the random graph is connected, but if $\log(n)/n \geq p$, then almost surely, the graph is not connected.

Indeed, Figure 3a is not fully connected providing an evidence for the underlying vertex nomination being correct. In comparison, Figure 3b is fully connected providing an evidence for the underlying vertex nomination being incorrect. Moreover, one can observe that the factor contributing to the connectedness in Figure 3b is the fact that at least one of the incorrectly classified vertices is highly probable to be connected to all vertices in the graph.

4.2 Performance under a random phase shift with no vertical shift

For numerical experiments, we simulate $\mathcal{D}(T)$, where $\mathbf{N}_{ij}$ is assumed to be a self-exciting process whose intensity process we now describe. For each $\mathbf{x} \in \mathbb{R}^2$, let

$$\lambda_{ij}(t, \mathbf{x}) = \Lambda_{\lfloor i,j \rfloor}(1 + \cos(2\pi t/24 + \mathbf{x}_{\lfloor i,j \rfloor})),$$

where $\Lambda_1 = 20$ and $\Lambda_2 = 8$, and let $(1,1) = (1,2) = (2,1) = 1,$ and $(2,2) = 2$.

Then, fix $\mu = (0, 3)^T \in \mathbb{R}^2$ and for each membership group $k = 1, 2$, we set

$$\begin{aligned} d\mathbf{X}_k(t) = &-(\mathbf{X}_k(t) - \mu_k)dt \\ &+ \mathbf{C}_{kk}(t) \sin(2\pi t/24 + \mathbf{X}_k(t)) \sum_{i<j} \delta_k(\lfloor i, j \rfloor)\Lambda_{\lfloor i,j \rfloor} dt \\ &+ \mathbf{C}_{kk}(t) \frac{\sin(2\pi t/24 + \mathbf{X}_k(t))}{1 + \cos(2\pi t/24 + \mathbf{X}_k(t))} \sum_{i<j} \delta_k(\lfloor i, j \rfloor) d\mathbf{N}_{ij}(t), \end{aligned}$$

where δ_x denotes the Dirac delta function. Then, for simplicity, we use an approximation that

$$\mathbf{C}(t) = \mathbf{C}(0) \quad \text{for } t \geq 0,$$

where $\mathbf{C}(0)$ is a constant diagonal matrix with positive diagonal entries. The motivation is from a steady state approximation that can be found in [13]. Then, we have considered a network with 20 vertices, where

$$\mathcal{C}_1 = \{1, \ldots, 10\}, \quad \text{and} \quad \mathcal{C}_2 = \{11, \ldots, 20\},$$

and we assume that $C_1 = C_2 = \varnothing$.

The numerical experiment result is illustrated in Figure 4. In this case, the vertex nomination solution considered states that all vertices are in $\mathcal{C}_1$. In particular, as expected, the subgraph restricted to $\{1, \ldots, 10\}$ is "sparse." On the other hand, the subgraph restricted to $\{11, \ldots, 20\}$ is somewhat "dense" but not as "dense" as expected. Nevertheless, the bipartite graph between $\mathcal{C}_1$ and $\mathcal{C}_2$ is noticeably "dense."

References

[1] Bollobas, B. 1998. *Modern Graph Theory*. Springer, New York.

[2] Coppersmith, G., Priebe, C. 2011. Vertex nomination via content and context. *Submitted for publication*.

[3] Daley, D. J., D. Vere-Jones. 2008. *An introduction to the theory of point processes: Volume 1*. Springer-Verlag.

[4] De Choudhury, M., Mason, W., Hofman, J., and Watts, D. (2010) Inferring relevant social networks from interpersonal communication.. *In Proc. 19th Intl Conf. World Wide Web, pp. 301-310. New York: Association for Computing Machinery*.

[5] Escanociano, J. and Velasco, C. 2006. Generalized spectral tests for the martingale difference hypothesis. *Journal of Econometrics* **134** 151–185.

[6] Giesecke K. and Kim B. 2011. Risk analysis of collateralized debt obligations. *Operations Research*.

[7] Goldenberg A., Zheng A., Fienberg S. and Airoldi E. 2010 A survey of statistical network models. *Found Trends Mach Learn* 2:129-233.

[8] Heard, N., Weston, D., Platanioti, K., and Hand, D. (2010). Bayesian anomaly detection methods for social networks. *Ann. Appl. Statist.* 4, 645-662.

[9] Kolacyzk, E. 2009. *Statistical Analysis of Network Data.* Springer, New York.

[10] Lo, A. and MacKinlay, C. 1989. The size and power of the variance ratio test in finite samples. *Journal of Econometrics* 203–238.

[11] Malmgren, R., Hofman, J., Amaral, L. and Watts, D. 2009. Characterizing Individual Communication Patterns *In Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery Data Mining, pp. 607-616. New York: Association for Computing Machinery*

[12] Perry, P. and Wolfe, P. 2010 Point process modeling for directed interaction networks *Submitted for publication.*

[13] Snyder, D. 1975. *Random point processes*. John Wiley & Sons Inc.

[14] Song, Z. 2011. A martingale approach for testing diffusion models based on infinitesimal operator. *Journals of Econometrics* **162** 189–212.