

TBD

Nam H. Lee & c

March 25, 2011

1 Overview

For each (undirected) pair ij of n vertices and each $t \in [0, 1]$, we let $N_{ij}(t)$ be the number of (undirected) communication events between vertex i and vertex j , where each event is marked with one of the communication content topic classes $\{1, 2\}$. Let, for each $t \in [0, 1]$,

$$\mathcal{D}(t) = \{(\tau_\ell, ij_\ell, k_\ell) : \ell = 1, \dots, N(t)\},$$

where $N(t) = \sum_{i < j} N_{ij}(t)$ and for simplicity, let

$$\mathcal{D} = \mathcal{D}(1).$$

In this paper, we consider a problem of classifying the n vertices into two groups after seeing \mathcal{D} , where the members of each class have the similar degrees of interests in topics from two content topic classes. For this, we propose a classification algorithm derived from an EM-based parameter estimation procedure.

To formulate our EM-based algorithm, we introduce a model for \mathcal{D} through a data augmentation principle. The hidden variables augmenting the data are the collection of random variables $\{\Lambda_{i,k}(t)\}$, where $\Lambda_{i,k}(t)$ is the degree of vertex i 's interest in topics from content topic class k at time t . We assume that each $\Lambda_{i,k}(t)$ is a strictly positive random variable such that for some fixed constant $\lambda_i \in (0, \infty)$,

$$\lambda_i = \Lambda_{i,1}(t) + \Lambda_{i,2}(t). \tag{1}$$

Our model, yielding the observation \mathcal{D} , is then to be a doubly stochastic marked Poisson process whose latent intensity process is specified by the $n \times 2$ random matrix

$$\Lambda(t) = (\Lambda_{i,k}(t); i = 1, \dots, n, k = 1, 2). \quad (2)$$

For convenience, we denote the i -th row of $\Lambda(t)$ by $\Lambda_i(t)$.

To demonstrate the performance of our classification algorithm, we will also consider a so-called vertex nomination problem, in which the members of a particular class is already known and the objective is then to decide whether or not the class is to be expanded further and if to be expanded, which (single) vertex is to be included next. A solution to the vertex nomination problem is likely to be useful if some other information other than \mathcal{D} is available. On the other hand, we will also demonstrate the performance of our classification algorithm when the only available information is \mathcal{D} and the vertex nomination is then interpreted as a two-pass procedure.

2 Single-period model

For each vertex i and time $t \in [0, 1]$, let

$$X_i(t) = (X_{i,1}(t), X_{i,2}(t)), \quad (3)$$

where

$$X_{i,k}(t) = \frac{\Lambda_{i,k}(t)}{\Lambda_i(t)}, \quad (4)$$

and note that each $X_i(t)$ is a probability vector. For each (undirected) pair ij , we assume that the process $N_{ij} = (N_{ij}(t) : t \in [0, 1])$ is a doubly stochastic Poisson process that whose intensity process is

$$\Lambda_{ij}(t) = \lambda_i \lambda_j (X_{i,1}(t)X_{j,1}(t) + X_{i,2}(t)X_{j,2}(t)). \quad (5)$$

A particularly useful characterization is to describe each N_{ij} as a process obtained by thinning a homogeneous Poisson process C_{ij} whose constant intensity is $\lambda_i \lambda_j$. This affords an interpretation that each t such that $C_{ij}(t) = C_{ij}(t-) + 1$ is a potential communication opportunity between vertex i and vertex j , and at each such t , depending on values of $X_i(t)$ and $X_j(t)$, the

opportunity may or may not give rise to an actual communication event, i.e. t such that $N_{ij}(t) = N_{ij}(t-) + 1$. To be more precise, it is our assumption that given the filtration

$$\mathcal{F}_{ij}^\Lambda(t) = \sigma(\Lambda_i(s), \Lambda_j(t) : s \leq t),$$

if $C_{ij}(t) = C_{ij}(t-) + 1$, then $N_{ij}(t) = N_{ij}(t-) + 1$ with probability $\Lambda_{ij}(t)$ and the topic mark is of content class k with probability

$$\frac{X_{i,k}(t)X_{j,k}(t)}{X_{i,1}(t)X_{j,1}(t) + X_{i,2}(t)X_{j,2}(t)} = \frac{\Lambda_{i,k}(t)\Lambda_{j,k}(t)}{\Lambda_{i,1}(t)\Lambda_{j,1}(t) + \Lambda_{i,2}(t)\Lambda_{j,2}(t)}.$$

Now, we fix a stationary diffusion process $Y_i = (Y_i(t) : t \in [0, 1])$ such that

$$dY_i(t) = \mu(Y_i(t))dt + \sigma_i(Y_i(t))dB_i(t), \quad (6)$$

where each μ_i and σ_i are twice continuously and boundedly differentiable functions, and B_1, \dots, B_n are independent standard Brownian motions. In fact, in this paper, for concreteness, we assume that

$$dY_i(t) = \beta_i(\mu_i - Y_i(t))dt + \sigma_i dB_i(t), \quad (7)$$

for some fixed constants $\mu_1, \dots, \mu_n \in \mathbb{R}$, and $\sigma_1 = \dots = \sigma_n \in (0, \infty)$ and $\beta_1 = \dots = \beta_n \in (0, \infty)$. Then, we assume that

$$X_{i,1}(t) = \frac{\exp(-Y_i(t))}{1 + \exp(-Y_i(t))}. \quad (8)$$

While we could also take $\sigma_1, \dots, \sigma_n$ and β_1, \dots, β_n to be parameters, but for our vertex nomination problem or more generally classification problem, these potential parameters are (perhaps) nuisance parameters (?). In our algorithm, we propose to fix the nuisance parameters before proceeding to a classification/vertex nomination task. Finally, for future reference, we let

$$\psi = (\mu_1, \dots, \mu_n, \lambda_1, \dots, \lambda_n).$$

3 Greedy algorithm for iterative vertex nomination

Our main result of this paper concerns an algorithm for correctly assigning each vertex into one of two classes, where within each group, all the members

have the same parameter i.e. if vertex i and vertex j are in the same class, then $\psi_i = \psi_j$. In this section, we provide a greedy algorithm for arriving at a partition of the vertex set which is likely to be the true partition of the vertex set that have produced the data \mathcal{D} .

For the initial step $q = 1$, suppose that we have an arbitrary partition $(C_1(1), C_2(1))$ of $\{1, \dots, n\}$. Let ψ^1 be the output of our EM algorithm, where we assume that within each group, all vertices have the same parameter. Then, for each pair $i, j \in C_k(1)$, $\psi_i^1 = \psi_j^1$. Let

$$Q(1) := \mathbf{E}_{\psi(1)} [\log(f(\mathcal{D}, Y; \psi^1))] , \quad (9)$$

where $f(\cdot; \psi)$ denote the likelihood function of observing \mathcal{D} and Y with the given parameter ψ (see the next section for more details about this). Next, for the $(r+1)$ -st iteration, suppose that we have a partition $(C_1(r), C_2(r))$ from the r -th iteration. If $C_2(r) = \emptyset$, then our partitioning algorithm terminates, yielding the r -th partition $(C_1(r), C_2(r))$ as the final output. So, we assume that $C_2(r) \neq \emptyset$. Then, for some $\ell > 0$ and $m > 0$ with $\ell + m = n$, we have

$$C_1(r) = \{u_1, \dots, u_\ell\}, \quad (10)$$

$$C_2(r) = \{v_1, \dots, v_m\}. \quad (11)$$

For each $i \in C_2(r)$, let

$$C_1^i(r) = C_1(r) \cup \{i\}, \quad (12)$$

$$C_2^i(r) = C_2(r) \setminus \{i\}. \quad (13)$$

Now, for each i , let $\psi^{i,r}$ be the EM estimate of ψ with the partition $(C_1^i(r), C_2^i(r))$, and let

$$Q^i(r) = \mathbf{E}_{\psi^{i,r}} [\log(f(\mathcal{D}, Y; \psi^{i,r}))] . \quad (14)$$

If for all $i \in C_2(r)$, $Q^i(r) \leq Q(r)$, then the algorithm terminates, yielding $(C_1(r), C_2(r))$ as the final partitioning. Otherwise, we set

$$i^* = \arg \max_{i \in C_2(r)} Q^i(r), \quad (15)$$

and let

$$C_1(r+1) = C_1(r) \cup \{i^*\}, \quad (16)$$

$$C_2(r+1) = C_2(r) \setminus \{i^*\}. \quad (17)$$

4 Estimation by EM procedure

4.1 Overview

Let

- (i) M_{ij} is the (total) number of messages between vertex i and vertex j ,
- (2) $\tau_{ij}(\ell)$ is the time at which occurred the ℓ -th communication event between vertex i and vertex j .

In the expectation stage of our EM algorithm, for each ψ , we compute the expected value of the following random variable:

$$\prod_{i < j} f_{ij}(\mathcal{D}(ij); X_i^r, X_j^r, \psi) \prod_{i=1}^n f_i(Y_i^r; \psi), \quad (18)$$

where

- (i) Y_i^r is a diffusion process whose distribution is a conditional distribution of Y_i given the observation \mathcal{D} with the unconditional distribution of Y_i for the r -th estimated parameter ψ^r ,
- (ii) $f_i(\cdot; \psi)$ is the likelihood of the i -th vertex process' sample path Y_i with the parameter ψ ,
- (iii) $f_{ij}(\cdot; X_i^r, X_j^r, \psi)$ specifies the conditional likelihood given X_1^r, \dots, X_n^r of observing message at time

$$\tau_{ij}(1), \dots, \tau_{ij}(M_{ij}),$$

repsectively, about topic

$$\kappa_{ij}(1), \dots, \kappa_{ij}(M_{ij}).$$

Note that conditioning on the path of Y_i and Y_j (but not on \mathcal{D}), and given the value of ψ , the value of $f_{ij}(\cdot; X_i, X_j, \psi)$ can be seen to be

$$\prod_{i < j} \left(\prod_{\ell=1}^{M_{ij}} \Lambda_{i, \kappa_{ij}(\ell)}(\tau_{ij}(\ell)) \Lambda_{j, \kappa_{ij}(\ell)}(\tau_{ij}(\ell)) \right) \exp \left(- \int_0^T \Lambda_{ij}(s) ds \right). \quad (19)$$

Next, given $\sigma_i = 1$ and $\beta_i = 1$ for all $i = 1, \dots, n$, it can be shown that

$$\begin{aligned} & \log(f_i(Y_i^r(t); \psi)) \\ = & \mu_i(Y_i^r(1) - Y_i^r(0)) - \frac{1}{2}((Y_i^r(1))^2 - (Y_i^r(0))^2) - \frac{1}{2} \int_0^1 (\mu_i - Y_i^r(t))^2 - 1 dt. \end{aligned}$$

Now we remark here that while unconditionally, the diffusion processes Y_1, \dots, Y_n are independent, conditionally, Y_1, \dots, Y_n need not be independent. Computing the exact conditionally likelihood of Y_i^r analytically is difficult if not practically impossible even for the relatively simple diffusion process Y_1, \dots, Y_n that we are considering in this paper. Because of this, our algorithm relies instead on an EM algorithm in where the expected value in the E step is estimated by simulation and in the M step, the maximization is done numerically. An except to this arises if we set the nuisance parameter $\sigma_1 = \dots = \sigma_n = 0$. In this case, each $Y_i^r = \mu_i$ and it is not difficult to see that the MLE of ψ given \mathcal{D} satisfies the following equations:

$$\frac{1}{\lambda_i} \left(\sum_{j \neq i} M_{ij} \right) - \sum_{j \neq i} \lambda_j (2X_i X_j + 1 - X_i - X_j) = 0, \quad (20)$$

$$\frac{1}{X_{i,1}} \left(\sum_{j \neq i} M_{ij,1} \right) + \frac{1}{1 - X_{j,1}} \sum_{j \neq i} M_{ij,2} - \lambda_i \sum_{j \neq i} \lambda_j (2X_{i,j} - 1) = 0. \quad (21)$$

4.2 Conditional sampling algorithms

In the E step, one needs to know how to compute, for each ψ ,

$$Q_\ell(\psi) = \mathbf{E}_{\psi_\ell, \mathcal{D}} [\log f_{[0,T]}(\mathcal{D}, X^{(\ell)}; \psi)], \quad (22)$$

where the law of $X^{(\ell)}$ is the law of X conditioning on the data \mathcal{D} and the parameter ψ_ℓ . The general closed-form expression of $Q(\psi)$ is difficult to obtain, and we develop a way to obtain a close approximation \widehat{Q}_ℓ of Q . To do this, we use a Monte Carlo estimate based on the rejection technique. First, our simulation of each X_i (unconditionally) is achieved through discretization. For example, we can fix a small $\Delta t > 0$ and

$$Y_i(t + \Delta t) = Y_i(t) + \beta_i(\mu_i - Y_i(t))\Delta t + \sigma_i Z \Delta t, \quad (23)$$

where Z is a standard normal random variable. Next, note that

$$f(Y|\mathcal{D};\psi) \propto f(Y,\mathcal{D};\psi) \quad (24)$$

$$= f(\mathcal{D}|X,\psi)f(Y|\psi) \quad (25)$$

$$= \prod_{i<j} f_{ij}(\mathcal{D}(ij); X_i, X_j, \psi) \prod_{i=1}^n f_i(Y_i; \psi), \quad (26)$$

$$\leq \prod_{i<j} (\lambda_i \lambda_j)^{M_{ij}} \prod_{i=1}^n f_i(Y_i; \psi). \quad (27)$$

Therefore, (in theory), to simulate conitionally, it is enough to simulate X unconditionally as a proposal, and accept the proposed sample path with probability:

$$\prod_{i<j} \left(\exp \left(- \int_0^T \Lambda_{ij}(X_i(s), X_j(s)) ds \right) \prod_{\ell=1}^{M_{ij}} \bar{\Lambda}_{\kappa_{ij}(\ell)}(X_i(\tau_{ij}(\ell)), X_j(\tau_{ij}(\ell))) \right). \quad (28)$$

On the other hand, the rejection probability (as the algorithm is presented now) is not trivial and as $\sum_{i<j} M_{ij}$ gets larger, the second factor is likely to be a very small number. To remedy this, we use a sequential rejection sampling prodedure. First, we partition the interval $[0, T]$ into many pieces, say, with grid $0, t_1, \dots, t_L$, so that each interval $[t_\ell, t_{\ell+1})$ contains at most one messaging event (of any kind). Then, progressing from the left end point to the right end point recursively, starting from the $X(t_\ell)$ obtained from the preceding interval simulation, simulate unconditionally. Then, accept and append the proposed sample path segment with the probability:

$$\exp \left(- \int_{t_\ell}^{t_{\ell+1}} \Lambda(X_i(s), X_j(s)) ds \right). \quad (29)$$

if the new interval $[t_\ell, t_{\ell+1})$ has no messaging event but otherwise, accept and append the proposed sample path segment with the probability:

$$\exp \left(- \int_{t_\ell}^{t_{\ell+1}} \bar{\Lambda}(X_i(s), X_j(s)) ds \right). \quad (30)$$

In our experiment, this sequential simulation technique significantly improve the efficiency of algorithm but it is important to note that the size of each $t_{i+1} - t_i$ is chosen with care since making it too small creates another rare event type issue, namely, always accepting the proposed sample path segment when ocassionally rejected.

5 Estimation by Kalman-Bucy filtering