Particle Filter Inference based on Activities for Overlapping Community Models

Shohei Sakamoto
Graduate School of System Informatics Kobe University
1-1 Rokkodaicho, Nada
Kobe 657–8501, Japan
shohei@cs25.scitec.kobe-u.ac.jp

Koji Eguchi
Graduate School of System Informatics Kobe University
1-1 Rokkodaicho, Nada
Kobe 657–8501, Japan
eguchi@port.kobe-u.ac.jp

ABSTRACT

Various kinds of data such as social media can be represented as a network or graph. Latent variable models using Bayesian statistical inference are powerful tools to represent such networks. One such latent variable network model is a Mixed Membership Stochastic Blockmodel (MMSB), which can discover overlapping communities in a network and has high predictive power. Previous inference methods estimate the latent variables and unknown parameters of the MMSB on the basis of the whole observed network. Therefore, dynamic changes in network structure over time are hard to track. Thus, we present a particle filter based on node activities with various term lengths for online sequential estimation of the MMSB. For instance, in an e-mail communication network, each particle only considers e-mail accounts that sent or received a message within a specific term length, where the length may be different from those of other particles. We show through experiments that our proposed methods achieve both high prediction performance and computational efficiency.

Keywords

Network analysis, latent variable models, sequential inference, and particle filter.

1. INTRODUCTION

Many kinds of data can be represented as a network or a graph, which is sometimes dynamic and large in scale. Typical examples of such dynamic, large-scale networks are social networks. By modeling such networks, we can discover communities that have a shared property, so as to avoid high-dimensional difficulties and to visualize complex networks, and can also uncover temporal dynamics in such communities. Moreover, we can predict links or relationships that do exist but have not been observed or do not exist but may appear in the near future. Latent variable models using Bayesian statistical inference are a powerful tool to analyze such networks [4].

In latent variable models for networks, latent random variables are used to represent communities or groups underlying a network, and observed random variables are used to represent the nodes. In this paper, we focus on a Mixed Membership Stochastic Blockmodel (MMSB) [1] as a typical latent variable model that provides overlapping communities. In the MMSB, each node is represented by a mixture of latent groups, where each group is represented by a multinomial distribution over nodes. The MMSB is effective for community discovery and link prediction.

The latent variables and unknown parameters of MMSB can be estimated by using variational Bayesian inference [1] or collapsed Gibbs sampling [9]. The MMSB is usually estimated on the basis of a whole observed network. This is called the batch estimation method. However, this method is not suitable in realistic situations, such as when the observations of links are given sequentially. Online estimation methods are promising for addressing these problems; however, previous online estimation methods [9] have room for improvement. The structure of a real-world complex network often changes over time, so old observations of links do not help and can even harm the estimation accuracy. In this paper, we address online estimation problems in such dynamic settings.

We present a particle filter based on node activities with various term lengths. In an e-mail communication network, each particle only considers e-mail accounts that sent or received a message in a specific term length, where the length may be different from those of other particles. Through experiments with a university community site dataset and an e-mail communication dataset, we show that our proposed particle filter can achieve both high prediction performance and computational efficiency.

2. RELATED WORK

A number of statistical network models were explored in previous studies, for example, to discover social roles in social network data and predict missing links in biological networks [4].Latent variable models with Bayesian statistical inference are a powerful tool to analyze such networks [13, 11, 7, 1, 10]. Nowicki and Snijders [13, 11] developed a stochastic blockmodel where each node is assigned to a cluster drawn from a multinomial over a fixed, finite number of clusters. Kemp et al. [7] extended stochastic blockmodels to an Infinite Relational Model (IRM) that assumes

WWW'17 Companion, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3053905

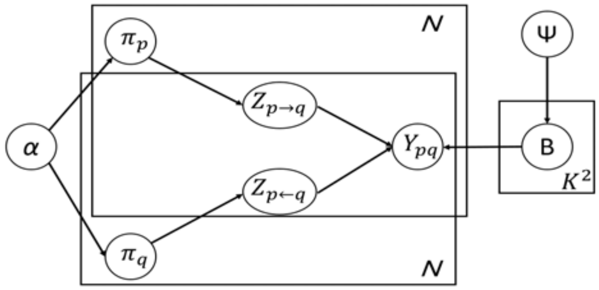


Figure 1: Graphical model of MMSB.

an infinite number of clusters. These models are based on the assumption that every node is assigned to a single cluster. In contrast, another line of research is based on the assumption that every node is assigned to multiple groups or communities, resulting in overlapping communities [1, 10]. The Mixed Membership Stochastic Blockmodel (MMSB) [1] is a typical approach for such overlapping community models, where each node is represented by a mixture of latent groups, and each latent group is represented by a multinomial distribution over nodes.

As for the previous work on sequential inference, Gopalan et al. [5] developed an efficient estimation for overlapping community models, where only random samples of observed links are used for the inference. They focused on efficiency in estimating the latent variables and unknown parameters of overlapping community models, not temporal dynamics in network structure. Kobayashi and Eguchi [9] explored sequential inference with overlapping community models in online settings. However, they did not consider dynamic phenomena when the structure of a real-world complex network often changes over time and therefore old observations of links do not help and can even harm the inference. This paper presents sequential inference based on a particle filter that uses node activities for capturing temporal dynamics in modeling overlapping communities, which has not been explored in previous studies.

3. MIXED MEMBERSHIP STOCHASTIC BLOCKMODEL

A Mixed Membership Stochastic Blockmodel (MMSB) [1] is an overlapping community model for network data. In this section, we first outline the modeling of the MMSB and then review the inference methods in both batch and online settings [9].

3.1 Modeling

First, we give the definitions used in this paper. We represent a graph as $\mathbf{G} = (\mathbf{N}, \mathbf{Y})$[1], where $\mathbf{N}$ is a set of nodes or vertices, and (p, q) element in adjacency matrix $\mathbf{Y}$ indicates whether a link or arc is absent or present from node p to node q as $Y(p, q) \in \{0, 1\}$. Each node is associated with a multinomial $\mathbf{Mult}(\boldsymbol{\pi}_p)$ over latent groups or communities (hereinafter, just "groups"), assuming a Dirichlet prior $\mathbf{Dir}(\boldsymbol{\alpha})$ over multinomial parameters $\boldsymbol{\pi}_p = \{\pi_{p,g} : g \in \{1, \cdots, K\}\}$. Here, $\pi_{p,g}$ indicates node p's multinomial parameter for any group $g \in \{1, \cdots, K\}$, representing the probability that node p falls into group g. Relationships between each pair of groups are defined by matrix $B_{K\times K}$ where each element represents a Bernoulli parameter with a Beta prior $\mathbf{Beta}(\boldsymbol{\psi})$. Here, $B(g, h)$ indicates the probability of generating a link from an arbitrary node in group g to another arbitrary node in group h. Given a link from p to q, indicator vector $\mathbf{z}_{p\rightarrow q}$ represents a group assigned to p, and $\mathbf{z}_{p\leftarrow q}$ represents a group assigned to q. These indicator vectors are denoted by $\mathbf{Z}_{\rightarrow} = \{\mathbf{z}_{p\rightarrow q} : p, q \in \mathbf{N}\}$ and $\mathbf{Z}_{\leftarrow} = \{\mathbf{z}_{p\leftarrow q} : p, q \in \mathbf{N}\}$. In accordance with the definitions above, the generative process of MMSB can be described as follows.

1. For each node p:
   - Draw a K-dimensional vector of multinomial parameters, $\boldsymbol{\pi}_p \sim \mathbf{Dir}(\boldsymbol{\alpha})$
2. For each pair of groups (g, h):
   - Draw a Bernoulli parameter, $B(g, h) \sim \mathbf{Beta}(\boldsymbol{\psi}(g, h))$
3. For each pair of nodes (p, q):
   - Draw an indicator vector for the initiator's group assignment, $\mathbf{z}_{p\rightarrow q} \sim \mathbf{Mult}(\boldsymbol{\pi}_p)$
   - Draw an indicator vector for the receiver's group assignment, $\mathbf{z}_{p\leftarrow q} \sim \mathbf{Mult}(\boldsymbol{\pi}_q)$
   - Sample a binary value that represents the presence or absence of a link, $Y(p, q) \sim \mathbf{Bern}(\mathbf{z}_{p\rightarrow q}^{\mathrm{T}} \mathbf{B} \mathbf{z}_{p\leftarrow q})$

The joint distribution with all the random variables (full joint distribution) is given as follows:

$$
\begin{aligned}
& P(\mathbf{Y}, \boldsymbol{\pi}_{1:N}, \mathbf{Z}_{\rightarrow}, \mathbf{Z}_{\leftarrow}, \mathbf{B} | \boldsymbol{\alpha}, \boldsymbol{\Psi}) \\
&= P(\mathbf{B}|\boldsymbol{\Psi}) \prod_{p,q:p\neq q} P(Y(p,q)|\mathbf{z}_{p\rightarrow q}, \mathbf{z}_{p\leftarrow q}, \mathbf{B}) P(\mathbf{z}_{p\rightarrow q}|\boldsymbol{\pi}_p) \\
& \quad P(\mathbf{z}_{p\leftarrow q}|\boldsymbol{\pi}_q) \prod_{p} P(\boldsymbol{\pi}_p|\boldsymbol{\alpha}) \qquad (1)
\end{aligned}
$$

A graphical model representation of the MMSB is shown in Fig. 1.

3.2 Batch Gibbs Sampler

Next, we describe a batch Gibbs sampler for estimating the latent variables and unknown parameters of an MMSB. For an observed link from node p to node q, the full conditional probability of assigning groups g and h to p and q, respectively, is given by:

$$
\begin{aligned}
& P(z_{p\rightarrow q} = g, z_{p\leftarrow q} = h | \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p,q)}, \mathbf{Z}_{\leftarrow}^{\neg(p,q)}, \boldsymbol{\alpha}, \boldsymbol{\psi}) \\
& \propto \; (n(p,g) - 1 + \Delta(g' \neq g) + \alpha_g)(n(q,h) - 1 + \Delta(h' \neq h) \\
& \quad + \alpha_h) \\
& \quad \frac{n(g,h,\delta) - 1 + \Delta(g' \neq g \wedge h' \neq h) + \psi_\delta}{n(g,h,0) + n(g,h,1) - 1 + \Delta(g' \neq g \wedge h' \neq h) + \psi_0 + \psi_1} \\
& = \begin{cases} \pi_{p,g}^{\neg(p,q)} \pi_{q,h}^{\neg(p,q)} B(g,h)^{\neg(p,q)} & (\mathit{if}\ \delta = 1) \\ \pi_{p,g}^{\neg(p,q)} \pi_{q,h}^{\neg(p,q)} (1 - B(g,h)^{\neg(p,q)}) & (\mathit{if}\ \delta = 0) \end{cases} \qquad (2)
\end{aligned}
$$

where $z_{p\rightarrow q}$ and $z_{p\leftarrow q}$ are the latent random variable[2] representing group assignments to initiator node p and receiver

[1]In this paper, we assume a directed graph for representing the network structure, but the network structure can also be easily applied to an undirected graph.

[2]In Section 3.1, we used an indicator vector $\mathbf{z}_{p\rightarrow q}$ where a specific component is one, corresponding to the group indicated by $z_{p\rightarrow q}$, and all the others are zero, for convenience.

node q, respectively, as mentioned in the previous section. $n(p,g)$ indicates the count of g assigned to p. $n(g,h,\delta)(\delta \in \{0,1\})$ indicates the count of presence ($\delta = 1$) or absence ($\delta = 0$) of links, where any initiator node is assigned to g and any receiver node is assigned to h. Moreover, α_g indicates g-th component of K-dimensional vector of Dirichlet hyperparameter α. ψ_1 and ψ_0 indicate Beta hyperparameters corresponding to the presence and absence of links, respectively. "$\neg(p,q)$" indicates ignoring the current group assignment to the link from p to q. The indicator function $\Delta(\cdot)$ takes one when the designated event occurs and zero if otherwise. g' and h' indicate the groups currently assigned to p and q, respectively.

In general, many kinds of real-world networks are sparse, as can often be seen in social networks. Therefore, there are often zero elements in the adjacency matrix $\mathbf{Y}$. To avoid bias due to this nature of networks, a sparsity parameter ρ is sometimes introduced into Eq. (2), as in the work of [1]:

$$P(z_{p\to q} = g, z_{p\leftarrow q} = h|\mathbf{Y}, \mathbf{Z}_{\to}^{\neg(p,q)}, \mathbf{Z}_{\leftarrow}^{\neg(p,q)}, \boldsymbol{\alpha}, \boldsymbol{\psi}) \propto \begin{cases} (1-\rho)\pi_{p,g}^{\neg(p,q)}\pi_{q,h}^{\neg(p,q)}B(g,h)^{\neg(p,q)} & (if\ \delta = 1) \\ (1-\rho)\pi_{p,g}^{\neg(p,q)}\pi_{q,h}^{\neg(p,q)}(1 - B(g,h)^{\neg(p,q)}) + \rho & (\text{if } \delta = 0) \end{cases} \quad (3)$$

where ρ is given by:

$$\rho = 1 - \sum_{p,q} \frac{Y(p,q)}{N(N-1)} \quad (4)$$

By using the full conditional probability in Eq. (3), a collapsed Gibbs sampler [6, 9] estimates latent variables and unknown parameters of MMSB.

3.3 Particle Filter

Particle filters, also known as sequential Monte Carlo methods, are based on the weighted average of multiple particles [3, 2, 9]. Each particle estimates group assignments for observed node pairs differently from the other particles in accordance with the steps of the incremental Gibbs sampler described in [9].The weight of each particle represents its importance, which is updated by using the likelihood of generating the observed link. When the variance of the weight is larger than a predefined threshold referred to as effective sample size (ESS) threshold, resampling is performed to make a new set of particles where the particles with negligibly low weights are replaced by new particles copied from those with higher weights. The simplest resampling scheme draws particles from the multinomial specified by the normalized weights [3]. After the resampling, the weights are reset to P^{-1}, where P indicates the number of particles.

The particle filter gives a posterior as:

$$P_{particle} = \sum_k (P^{(k)} \times \omega^{(k)}) \quad (5)$$

where $P^{(k)}$ indicates the posterior given by k-th particle in accordance with Eq. (3). $\omega^{(k)}$ indicates the weight of the k-th particle, which is proportional to the likelihood of generating observed links by using the particle. The incremental Gibbs sampler [9] can be considered as a special case when the number of particles is one.

4. ONLINE INFERENCE USING NODE ACTIVITIES

The structure of a real-world complex network often changes over time, so old observations of links do not help and can even harm the estimation accuracy. The previous online estimation methods in Section 3 have some drawbacks in such dynamic settings. When a link from/to an unseen node is observed, the previous online estimation methods assume the absence of links between the unseen node and the other nodes that were observed previously, except for the node that is currently linked to/from the unseen node. Thus, they assign groups to not only the pair of nodes that form the observed link but also all the other nodes that were observed previously. However, groups are sometimes inappropriately assigned on the basis of this assumption since some of the node pairs *potentially* have links that have no chance to be observed. We take into account *node activities* to address this problem, as detailed below.

4.1 Definition of Node Activities

We first define *active* nodes as those that were observed to be linked to/from others during a period of observing the most recent ℓ links, while all the other already observed nodes are defined as *inactive*. We then assign groups on the assumption that, when a link from/to an unseen node is observed, the absence of links is also observed only to/from the active nodes. Fig. 2 illustrates the flow of estimating the group assignments by using the previous and proposed methods. In this figure, $a, b, \cdots, d$ represents the links that are observed in alphabetical order. When the presence of a link is observed between a new node D and an already observed node B at time $t = 1$, the previous methods assign groups to the pair of nodes (D, B) and also to (D, A) and (D, C), assuming that A and C had already been observed by that time and that the absence of links is observed for (D, A) and (D, C) at that time. On the other hand, in the same situation, the proposed method assign groups to the pair of nodes (D, B) and (D, C) but not (D, A) when C is active (observed to have a link closely by that time) but A is inactive (not observed to have had a link for a while). When $t = 2$, groups are assigned on the same assumption. In this way, the proposed methods assign groups only to active nodes when a new node is observed, while the previous methods assign groups to all the already observed nodes in the same situation, regardless of whether the new node is linked to/from the already observed nodes. Thus, the proposed methods may work to avoid inappropriate group assignments that are caused by the frequent observations of absence of links, making model estimation more accurate. In addition, since estimation of group assignments for non-active nodes (treated as missing values) are not performed, the computational cost can be expected to be reduced.

4.2 Particle Filter using node activities

By simply applying the node activities discussed in the previous section, we can modify the particle filter described in Section 3.3.

The node activities rely on a term length since the node is deemed to be active if some activities are observed within the term but inactive if otherwise. From this consideration, we propose using a particle filter based on node activities with various term lengths by setting a different term length for

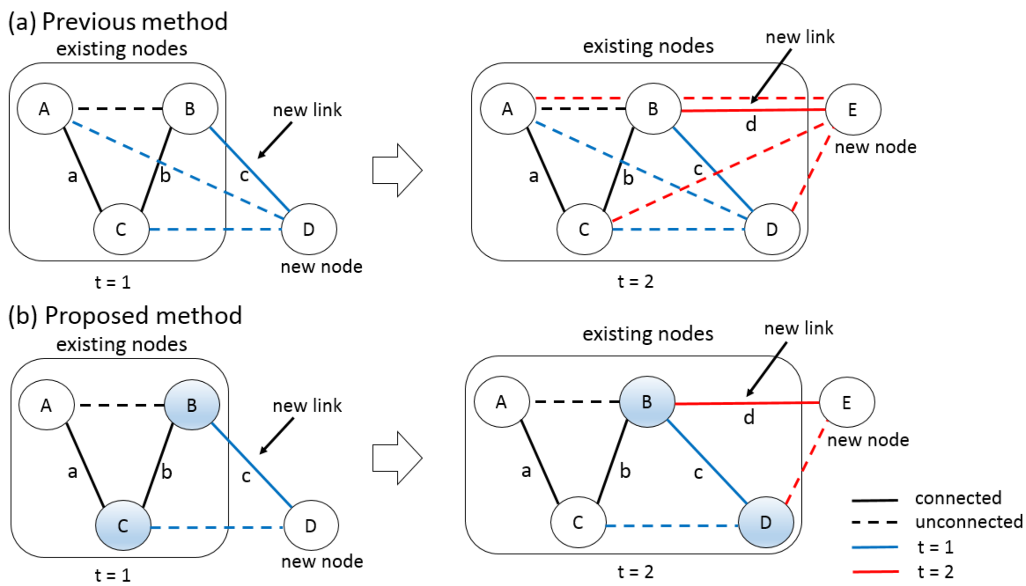


Figure 2: Illustration of how to estimate group assignments using previous non-activity-based methods (top) and proposed activity-based methods (bottom) when $\ell = 1$. Active nodes are shaded in this figure.

each particle. To calculate the final posterior for this particle filter, we can use Eq. (5) in the same way of the particle filter that is previously described in Section 3.3. Note that each particle is now assumed to consider a different term length to detect the node activities.

For instance, in an e-mail communication network, each particle only considers e-mail accounts that sent or received a message within a specific term length, where the length may be different from those of other particles. This particle filter inference may be made more robust by considering multiple terms for the node activities instead of uniformly setting a fixed term length for every particle. We believe that considering multiple terms for the node activities is especially effective to track dynamical changes in network structure over time. The node activities based on a long term should work effectively when the network structure is stable over time. When the network structure changes drastically, the long-term node activities cause inappropriate estimation, so the term should be shorter. Assuming a realistic situation where we do not know the network dynamics in advance, this kind of diversity of particles should work effectively for sequential (online) estimation. We assume that a term length for each particle is sampled from a Poisson distribution, as below:

$$\ell \sim Poisson(\lambda)$$

where λ is the Poisson mean parameter. A Gaussian can also be used instead, but a Poisson is more appropriate since it generates positive integers.

5. EXPERIMENTS

In this section, we evaluate our methods for the online sequential estimation of a MMSB through experiments with time-series network data. We then discuss both the prediction performance and computational efficiency in time.

5.1 Settings

5.1.1 Datasets

We used two online communication datasets in our experiments, since our ideas on node activities are expected to work more effectively with such datasets.

Dataset A.

This dataset is extracted from an online community site for students at the University of California, Irvine from April to October 2004 [12]. A link is assumed to occur when a message is sent from one user to another. The number of nodes is 1,899.

Dataset B.

This dataset is extracted from the Enron e-mail communication archive [8] from December 1999 to March 2002 and further cleaned so that only the users (i.e., e-mail addresses) who sent and received at least seven e-mails are included [9]. Each node represents an e-mail address, and each link represents an e-mail communication from a sender to a receiver with a time stamp. The number of nodes is 2,356.

Cross-validation settings.

To use five-fold cross-validation, we divided a set of observations (i.e., presence of links) in each dataset evenly into five sets. We further divided each set into a test set and a validation set and used the remaining four sets as training sets Since we estimated a model in an online setting, we used the observations sequentially within the training set. We determined the number of groups and hyperparameters by a grid search using the training and validation sets. We then evaluated the estimation methods using the test set.

Table 1: Average increase rate of test-set log-likelihood and its sample standard deviation for particle filters with Datasets A and B.

Dataset	Dataset A	Dataset B
Activity-based (fixed-term)	0.0373 ± 0.0046	0.0321 ± 0.0055
Activity-based (Poisson)	0.0420 ± 0.0043	0.0326 ± 0.0037
Non-activity-based (baseline)	(0.0000 ± 0.0000)	(0.0000 ± 0.0000)

5.1.2 Number of groups and hyperparameters

Before detailing the online experiments, we describe how we set the number of groups K and hyperparameters α, ψ_0 and ψ_1. We determined the number of groups K by using a grid search over $\{5, 10, 15, \cdots, 40\}$ in a batch setting, resulting in $K = 10$ for Dataset A and $K = 30$ for Dataset B. We also determined ψ_0 and ψ_1 in the manner above. For α, we used the symmetric Dirichlet hyperparameter fixed at 0.1. For a fair comparison, these settings are determined in the same manner used by [9].

For our proposed activity-based method, we determined the term $\ell = 60$ for Dataset A and $\ell = 30$ for Dataset B and the Poisson mean parameter $\lambda = 70$ for Dataset A and $\lambda = 60$ for Dataset B by doing a grid search, but the details are omitted due to limitations of space. In these experiments with particle filters, we determined the ESS threshold by doing a grid search over $\{4, 8, 12, 16, 20\}$, fixing the number of particles to 24.

5.1.3 Inference Methods

First, for estimating MMSB in online settings, we compare our activity-based method and non-activity-based method as the number of particles is one. Then, we compare two versions of the proposed activity-based particle filter.

5.1.4 Evaluation metrics

We evaluate the prediction performance using the average value of the rate of change of the test-set log-likelihood. The likelihood of the test set $\mathbf{s}^{test}$ indicates how effectively the model predicts unseen data at time interval t using the model estimated with observed data by $t-1$ and is given by:

$$p(\mathbf{s}^{(t)}_{test}) = \prod_{(p,q)\in\mathbf{s}^{(t)}_{test}} \sum_{g,h} \Big[(1-\rho^{(t-1)})\pi^{(t-1)}_{p,g}\pi^{(t-1)}_{q,h}B(g,h)^{t-1}\Big]^{\delta(p,q)} \Big[(1-\rho^{(t-1)})\pi^{(t-1)}_{p,g}\pi^{(t-1)}_{q,h}(1-B(g,h)^{t-1}+\rho^{(t-1)}\Big]^{1-\delta(p,q)} \tag{6}$$

where $\delta(p, q) \in \{1, 0\}$ represents the presence or absence of a link from node p to node q. ρ is the sparsity parameter, as defined in Eq. (4). Multinomial parameters $\pi_{p,g}$ and $\pi_{q,h}$ and Bernoulli parameter $\mathbf{B}(g, h)$ are estimated using Eq. (3) using the observations by time $t-1$. Given a discrete-time series network at $t \in \{1, ..., T\}$, the average increase rate of test-set log-likelihood is

$$\frac{1}{T}\sum_{t=1}^{T}\frac{X(t)-I_0(t)}{|I_0(t)|} \tag{7}$$

where $X(t)$ represents test-set log-likelihood that has the target inference algorithm at time interval t. $I_0(t)$ represents test-set log-likelihood that is the baseline at t. Here, the baseline is set to the particle filter with $P = 1$ at t in our experiments. The larger the average increase rate of test-set log-likelihood, the better the target method's prediction performance compared with the baseline.

5.1.5 Experimental environment

All the experiments in this paper were performed on a server with 48 gigabytes of memory and 12 CPU cores(24 threads). Also, we used C++ for implementation.

5.2 Results

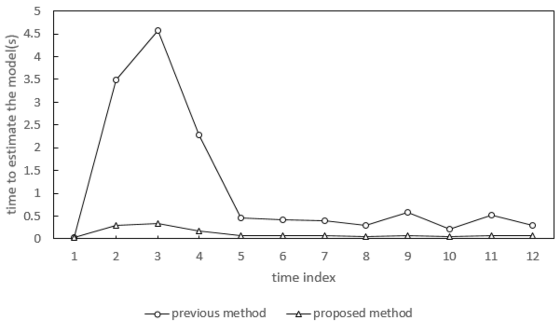
In this section, we show the experimental results using the datasets described previously in terms of prediction performance and computational efficiency in time.

5.2.1 Results of the estimation time using node activities

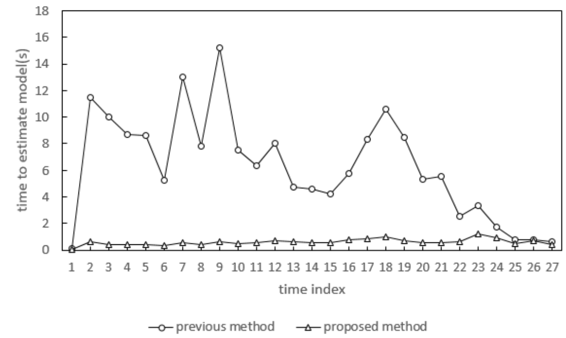
We first evaluate the time required for the model estimation with particle filter when $P = 1$. Fig. 3 demonstrates the time for estimation with each method in time-series plots for Datasets A and B. As shown in these graphs, the proposed activity-based method needed less time for estimation than the previous non-activity-based one. This indicates that the proposed method improves computational efficiency in time by considering the node activities. Furthermore, the proposed method brought about a larger improvement for Dataset B than for Dataset A. This is because the proposed method works more efficiently when the number of observed links is larger and so the number of newly observed nodes is also larger, as in Dataset B. This indicates that the proposed method can reduce the time for model estimation for a larger-scale network.

5.2.2 Results of particle filter using node activities

Next, we evaluated the particle filter based on the node activities with a fixed term length and that with various term lengths. Table 1 shows the prediction performance results of the proposed activity-based particle filters for Datasets A and B. In this table, 'Activity-based (Poisson)' indicates the particle filter described in Section 4.2 and 'Activity-based (fixed-term)' indicates the particle filter using the fixed-term node activities. The baseline is the previous non-activity-based particle filter with $P = 1$ described in Section 3.3, as mentioned previously. As can be seen in Tables 1, the node activities are helpful for the particle filter inference. Moreover, the particle filter using the node activities with various term lengths works more effectively than that using the fixed-term node activities. From these results, such diverse particles achieve robust sequential (online) estimation, especially for dynamic networks.



(a) Dataset A



(b) Dataset B

Figure 3: Evaluation results on the time for estimation in time series plots for Datasets A and B.

From the two evaluations above, the proposed methods are better than the previous methods in terms of both the prediction performance and the estimation time.

6. CONCLUSIONS

In this paper, we proposed using a particle filter for an online sequential estimation of MMSB by considering node activities to track dynamical changes in network structure over time. We used two datasets in our experiments and evaluated the proposed activity-based methods in terms of prediction performance and computational efficiency. In terms of prediction performance, the proposed methods were better than the previous non-activity-based methods. In terms of computational efficiency, the proposed methods reduced the cost in estimation time compared with the baselines.

Evaluation under more practical situations are left for future work. Another direction for future work is to apply our ideas to various types of statistical network models. We are especially interested in online sequential estimation of nonparametric relational models, such as a Latent Feature Relational Model (LFRM) [10].

Acknowledgements

This work was supported in part by the Grant-in-Aid for Scientific Research (#15H02703) from JSPS, Japan.

References

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

[2] K. R. Canini, L. Shi, and T. L. Griffiths. Online inference of topics with latent Dirichlet allocation. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, pages 65–72, Clearwater Beach, Florida, USA, 2009.

[3] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer New York, 2001.

[4] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, 2010.

[5] P. K. Gopalan, S. Gerrish, M. Freedman, D. M. Blei, and D. M. Mimno. Scalable inference of overlapping communities. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

[6] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101:5228–5235, 2004.

[7] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence*, volume 1, pages 381–388, Boston, Massachusetts, USA, 2006.

[8] B. Klimt and Y. Yang. Introducing the Enron corpus. In *First Conference on Email and Anti-Spam CEAS*, Mountain View, California, USA, 2004.

[9] T. Kobayashi and K. Eguchi. Online inference of mixed membership stochastic blockmodels for network data streams. *IEICE Transactions on Information and Systems*, E97-D(4):752–761, 2014.

[10] K. T. Miller, M. I. Jordan, and T. L. Griffiths. Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems*, volume 22, pages 1276–1284, 2009.

[11] K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

[12] T. Opsahl and P. Panzarasa. Clustering in weighted networks. *Social Networks*, 31(2):155–163, 2009.

[13] T. A. B. Snijders and K. Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, (14):75–100, 1997.