# Implement scalable stochastic block model of overlapping communities using Douban data

**Qiaohui Lin**
Department of Statistics and Data Sciences
University of Texas at Austin
Austin, TX 78713
qiaohui.lin@utexas.edu

**Yuguang Yue**
Department of Statistics and Data Sciences
University of Texas at Austin
Austin, TX 78713
yuguang@utexas.edu

## Abstract

We implement the assortive-Mixed Membership Stochastic Block Model using stochastic varational inference on a large real network dataset. The algorithm was proposed by Gopalan et.al (2012). We study the model, compare the algorithm with others and analyze one of the biggest the Chinese social network, Douban, using this strategy. We propose some further extensions at the end.

## 1 Introduction

Relational data, such as social network, protein-protein interaction, web page links, usually has the community structure of high concentration within groups and low concentration between groups [2]. Thus, in the resulting network analysis of these data, a fundamental problem is to detect the communities, i.e, groups of densely interconnected nodes [1,2,3].

In previous methods of community detection, such as clustering, mixture models, and the class of latent stochastic blockmodels, each node can only belong to one cluster. However, in many real networks, each node may belong to multiple clusters, known as overlapping communities. To allow this feature, mixed membership stochastic models (MMSB) associates each node with multiple clusters using a membership probability-like vector, and thus makes the nodes exhibit a mixture of communities [1,3].

In Airoldi et.al[2008], the original MMSB uses coordinate ascent to optimize the variational objective of the posterior inference, which makes it hard to scale to large datasets. In this project, we apply stochastic optimization algorithm [Gopalan et.al(2012)] to solve the variational inference problem in big data setting with subsampling method. We implent the model on a large real dataset: Douban network dataset, one of the largest Chinese social networking website, an undirected symmetric network with $154907$ nodes and $654188$ edges and recover the community strength of each community.

## 2 assortive-Mixed Membership Stochastic Block Model

Mixed membership models the overlapping communities as in each node $a$ associates with communities with a probability vector $\pi_a$. $\pi_{a,k}$ denotes the probability of node $a$ belonging to community $k$. We then introduce latent interaction indicator vector $z_{a\to b}$, $z_{a\to b}$ denotes the group membership of $a$ when he responds to the survey question about $b$ in group $k$. $z_{a\to b,k} = z_{b\to a,k} = 1$ or $z_{a\to b} = z_{b\to a} = k$ means group $k$ is active in determining whether there is a link between $a$ and $b$. A same latent community indicator of two nodes $z_{a\to b,k} = z_{b\to a,k} = 1$ means a higher chance of them being linked.

Preprint. Work in progress.

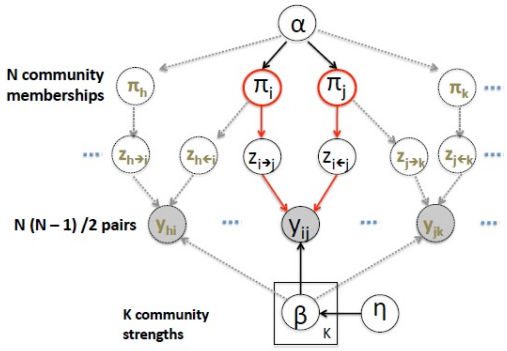


Figure 1: Sample figure caption.

As shown in Figure 1, whether the two nodes are linked (observed link $y_{ab}$), is both determined by the similarity of their membership ($z_{a\to b,k}, z_{b\to a,k}$) and the strength of the community they share($\beta_k$) when they are in the same group. When they are not in the same group, they can still be linked. We give another parameter $\epsilon$, indicating the possibility the nodes being linked when not in the same group.

The generative process is described below.

- For each community $k$,draw $\beta_k \sim Beta(\eta)$
- for each node $a$ ,draw $\pi_a \sim Dir(\alpha)$
- for each pair $a$ and $b$:
  - Draw community indicator $z_{a\to b,k} \sim \pi(a)$
  - Draw community indicator $z_{b\to a,k} \sim \pi(b)$
  - Draw link $y_{ab}$ from Bernoulli(r) where:

$$r = \begin{cases} \beta_k & z_{a\to b,k} = z_{b\to a,k} = 1 \\ \epsilon & z_{a\to b,k} \neq z_{b\to a,k}. \end{cases}$$

## 3 Stochastic Variational Inference

The goal is to compute the posterior distribution $p(\pi_{1:N}, \mathbf{z}, \beta_{1:K}|\mathbf{y}, \alpha, \eta)$. The posterior of $\pi_{1:N}$ gives us the community membership of nodes and the posterior of $\mathbf{z}$, the interaction indicator, identifies the links between each pair of nodes in each community, and the $\beta$ specifies the community strength.

We use a stochastic variational inference to compute the posterior. We define a mean-field family of latent variables $q(\beta, \pi, )$ with

$$q(z_{a\to b,k} = k) = \phi_{a\to b,k}; q(\pi_a) = Dir(\pi_a, \gamma_p); q(\beta_k, \lambda_k) = Beta(\beta_k; \lambda_k)$$

Minimizing the KL divergence between $q$ and true posterior $p$ is equivalent to maximizing the ELBO:

$$\begin{aligned}
L(y,\phi,\gamma,\lambda) =& E_q[\log p(y,\pi,z,\beta|\alpha,\eta)] - E_q[\log q(\beta,\pi,z)] \\
=& \sum_k E_q[\log p(\beta_k|\eta_k)] - \sum_k E_q[\log q(\beta_k|\lambda_k)] \\
&+ \sum_n E_q[\log p(\pi_n|\alpha)] - \sum_n E_q[\log q(\pi_n|\gamma_n)] \\
&+ \sum_{a,b} E_q[\log p(z_{a\to b}|\pi_a)] + \sum_{a,b} E_q[\log p(z_{b\to a})|\pi_b] \\
&- \sum_{a,b} E_q[\log q(z_{a\to b}|\phi_{a\to b})] - E_q[\log q(z_{b\to a}|\phi_{b\to a})] \\
&+ \sum_{a,b} E_q[\log p(y_{ab}|z_{a\to b}, z_{b\to a}, \beta)]
\end{aligned}$$

To derive the updates of each parameter, we first use the property for exponential family (Airoldoi,Blei,Fienberg& Xing(2008)[3], Hoffman,Blei & Bach (2010)[5]). If $\theta$ is Dirichlet with parameter $\alpha$, then

$$p(\theta|\alpha) = exp(\sum_{i=1}^{k}(\alpha_i - 1)\log\theta_i) + \log\Gamma(\sum_{i=1}^{k}\alpha_i) - \sum_{i=1}^{k}\log\Gamma(\alpha_i)$$

$$E[\log\theta_i|\alpha] = \Psi(\alpha_i) - \Psi(\sum_{j=1}^{k}\alpha_j)$$

Thus, in our model,

$$E_q(\log(\pi_i)) = \int q(\pi_i|\gamma_i)\log(\pi_i)d\pi = \Psi(\gamma_i) - \Psi(\sum_{j=1}^{k}\gamma_j)$$

$$E_q(\log(\beta_j)) = \int q(\beta_j|\lambda_j)\log(\beta_j)d\beta = \Psi(\lambda_j) - \Psi(\sum_{j=1}^{k}\lambda_j)$$

We arrive at the ELBO explicity in the form of

$$
\begin{aligned}
L(y,\phi,\gamma,\lambda) = &\sum_{k}^{K}\log(\Gamma(\eta_{k,1})+\Gamma(\eta_{k,2})-\Gamma(\eta_{k,1}+\eta_{k,2}))+\sum_{k}^{K}\sum_{j=1}^{2}(\eta_{k,j}-1)(\Psi(\lambda_{k,j}-\Psi(\sum_{j}^{2}\lambda_{k,j})))\\
&-\sum_{k}^{K}(\log(\Gamma(\lambda_{k,1})+\Gamma(\lambda_{k,2})-\Gamma(\lambda_{k,1}+\lambda_{k,2})))+\sum_{k}^{K}\sum_{j=1}^{2}(\lambda_{k,j}-1)(\Psi(\lambda_{k,j}-\Psi(\sum_{j}^{2}\lambda_{k,j})))\\
&+\sum_{i}^{N}\log(\Gamma(\sum_{k}^{K}\alpha_k))+\sum_{i,k}^{N,K}\log(\Gamma(\alpha_k))+\sum_{i,k}^{N,K}(\alpha_k-1)(\Psi(\gamma_{i,k}-\Psi(\sum_{k}\gamma_{i,k}))\\
&-\sum_{i}^{N}\log(\Gamma(\sum_{k}^{K}\gamma_k))+\sum_{i,k}^{N,K}\log(\Gamma(\gamma_k))+\sum_{i,k}^{N,K}(\gamma_{i,k}-1)(\Psi(\gamma_{i,k}-\Psi(\sum_{k}\gamma_{i,k}))\\
&+\sum_{a,b}^{N}\sum_{k}^{K}\phi_{a\to b,k}(\Psi(\gamma_k-\Psi(\sum_{k}\gamma_{a,k}))\\
&+\sum_{a,b}^{N}\sum_{k}^{K}\phi_{b\to a,k}(\Psi(\gamma_k-\Psi(\sum_{k}\gamma_{a,k}))\\
&-\sum_{a,b}^{N}\sum_{k}^{K}\phi_{a\to b,k}\log\phi_{a\to b,k}\\
&-\sum_{a,b}^{N}\sum_{k}^{K}\phi_{b\to a,k}\log\phi_{b\to a,k}\\
&+\sum_{a,b}\sum_{k}\phi_{a\to b,k}[\phi_{b\to a,k}(y_{ab}\log\beta_k+(1-y_{ab})\log(1-\beta_k))+\\
&\qquad(1-\phi_{b\to a,k})(y_{ab}\log\epsilon+(1-y_{ab})\log(1-\epsilon))]
\end{aligned}
$$

$program@ep$

The strategy is to optimize the ELBO by iteratively update the local parameter ($\phi_{a\to b}$,$\phi_{b\to a}$) and global parameter ($\gamma_a$,$\gamma_b$,$\lambda$). At each iteration, we subsample the data, then, given the global parameter, we optimize the local parameters with their natural gradients. Then, fixing the local parameter, we again update the global ones with stochastic natural gradients. Subsampling gives a noisy but still unbiased updates.

To update the global parameter $\lambda$,$\gamma$, we take derivative of $L$ with respect to $\lambda$ and $\gamma$

$$
\begin{aligned}
\gamma_k &= \alpha_k+\sum_{a,b=1}^{n}\phi_{a\to b,k}\\
\lambda_k &= \eta_{k,i}+\sum_{a,b=1}^{n}\phi_{a\to b,k}\phi_{b\to a,k}y_{ab,i}
\end{aligned}
$$

Following Hoffman,Blei & Bach (2010), we move from batch update to online update by setting

$$
\begin{aligned}
\lambda &= (1-\rho_t)\lambda+\rho_t\tilde{\lambda}\\
\gamma &= (1-\rho_t)\gamma+\rho_t\tilde{\gamma}
\end{aligned}
$$

where $\rho_t=(\tau_0+t)^{-\kappa}$ in $t$th iteration. $\tilde{\lambda}$,$\tilde{\gamma}$ are the natural gradients calculated above.

Considering the size of the dataset, we do a subsampling of the data in each iteration, thus the stochastic variational inference. We randomly pick node pair $(a,b)$ from distribution $g(a,b)$, and

treat the network as repeating $(a, b)$ by $\frac{1}{g(a,b)}$.

$$\tilde{\gamma} = \alpha_k + \frac{1}{g(a,b)}\phi^t_{a\to b,k}$$
$$\tilde{\lambda} = \eta_{k,i} + \frac{1}{g(a,b)}\phi^t_{a\to b,k}\phi^t_{b\to a,k}y_{a,b}$$

In this project, we implement the simplest method, to sample node pairs $(a, b)$ uniformly at random, resulting in an independent pair sampling.

For the local parameter $\phi_{a\to b,k}$,$\phi_{b\to a,k}$, we isolate the part in ELBO which contains $\phi$ and add in the Lagrangian multiplier of $\kappa(\sum_{k=1}^{K}\phi_{a\to b,k} - 1)$. Taking the derivative and set to 0, we have

$$\begin{aligned}\log\phi_{a\to b,k} \propto \sum_{a,b}\sum_{k}\phi_{a\to b,k}[&\phi_{b\to a,k}(y_{ab}\log\beta_k + (1-y_{ab})\log(1-\beta_k))\\ &+ (1-\phi_{b\to a,k})(y_{ab}\log\epsilon + (1-y_{ab})\log(1-\epsilon))]\\ &+ [\Psi(\gamma_{a\to b,k}) - \Psi(\sum_{k}\gamma_{a\to b,k})]\end{aligned}$$

which, by using $E_q(\log(\pi_i)) = \Psi(\gamma_i) - \Psi(\sum_{j=1}^{k}\gamma_j)$, can be simplified into

$$\begin{aligned}\phi_{a\to b,k} \propto \exp[E_q(\log(\pi_i)) + \sum_{a,b}\sum_{k}\phi_{a\to b,k}[&\phi_{b\to a,k}(y_{ab}\log\beta_k + (1-y_{ab})\log(1-\beta_k))\\ &+ (1-\phi_{b\to a,k})(y_{ab}\log\epsilon + (1-y_{ab})\log(1-\epsilon))]]\end{aligned}$$

Conditioning on $y = 0$ or $y = 1$, we have

$$\phi_{a\to b,k}\big|(y=0) \propto \exp\{E_q[\log\pi_{a,k}] + \phi_{b\to a}E_q[\log(1-\beta_k)] + (1-\phi_{b\to a,k})\log(1-\epsilon)\}$$
$$\phi_{a\to b,k}\big|(y=1) \propto \exp\{E_q[\log\pi_{a,k}] + \phi_{b\to a}E_q[\log(\beta_k)] + (1-\phi_{b\to a,k})\log(\epsilon)\}$$

## 4 Algorithm and Test-run on Toy Dataset

Following Gopalan, P.K, et.al (2012), the algorithm is:

---

Algorithm: Stochastic a-MMSB

---

Initialize $\gamma$, $\lambda$

**while** convergence is not met **do**:

- Subsampling S pair of nodes
- Local-step: Optimize $\phi_a, \phi_b$ $(a, b) \in S$
- Compute the natural gradients $\partial\gamma_n^t$, $\partial\lambda_k^t$, $\forall n, k$.
- Global-step: Update $(\gamma,)$
- set $\rho_t = (\tau + t)^{-k}$, $t = t + 1$

**end while**

---

The decreasing learning rate is set at $\rho_t = (\tau + t)^{-k}$, with $k = 1$, $\tau = 100$

We first test the algorithm on a widely used toy dataset: karate-club dataset built in library networkx in python. The toy dataset has 34 nodes and 78 edges shown in Figure 2(a).

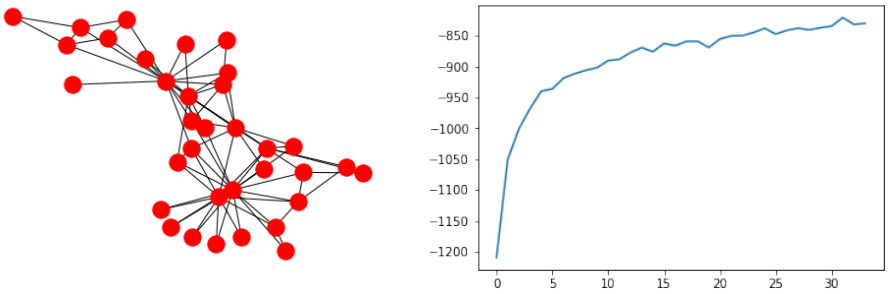


Figure 2: Test run on Karate-club dataset. Left: the entire karate-club dataset. Right: Traceplot of ELBO using our aMMSB model and stochastic variational inference algorithm.

Using the whole dataset without subsampling, setting $K = 2$, $\epsilon = 0.1$,the algorithm takes 5.42 seconds to achieve convergence and Figure 2(b) traces the ELBO.

The hyper-parameter $\lambda_k$ for community strength $\beta_k$ is:

Table 1: Community strength $\beta_k$ and its hyperparamter $\lambda_k$ in test-karate dataset

| Community | $\lambda$ | E($\beta$) |
|---|---|---|
| k=1 | (3.39,8.48) | 0.285 |
| k=2 | (3.14,8.25) | 0.276 |

For 3 random node $a_1, a_2, a_3$, if $a$ knows random $b$ in community $k$ is denoted by group interaction indicator $z_{a\to b,k}$, $z_{a\to b,k=1}$ with probability $\phi_{a\to b,k}$:

Table 2: Community interaction probability $\phi_{a\to b}$ for 3 random $a$ and random $b$ in test-karate dataset

| node | $\phi_{a\to b,k=1}$ | $\phi_{a\to b,k=2}$ |
|---|---|---|
| $a_1$ | 0.54 | 0.46 |
| $a_2$ | 0.47 | 0.53 |
| $a_3$ | 0.64 | 0.35 |

## 5 Implementation on Douban Dataset

We use the 2015 Douban Dataset,the largest Chinese social platform where members share communities of interest in music, movies, events. The Douban Dataset, from `http://socialcomputing.asu.edu/datasets/Douban`, is a symmetric, undirected network with $154907$ nodes and $654188$ edges.

We first randomly select 1000 nodes and 10000 edges from the dataset and visualize the network.

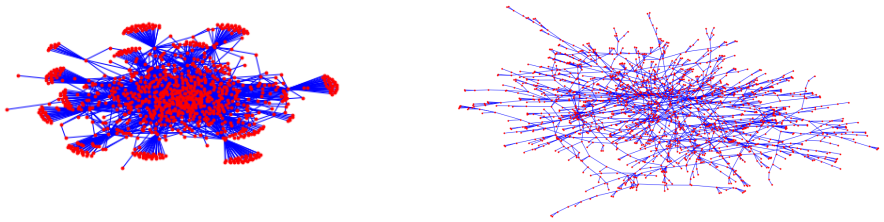

Figure 3: Top: A randomly-draw 1000 nodes network from 154907 nodes, Bottom: A randomly-draw 10000 edges network from 654188 edges from Douban Dataset

Visualization of only a small portion of dataset shows overlapping communities and noisy connectivity, suggesting a good reason to implement the a-MMSB model proposed.

We use the algorithm proposed and set K=100. In each iteration, we subsample the dataset by randomly picking 500 pair of nodes. We set the convergence criterion to be less than $0.1\%$ in Elbo changes between two iterations. This criterion is reached after $176$ iterations and we keep it running until $250$ iterations.

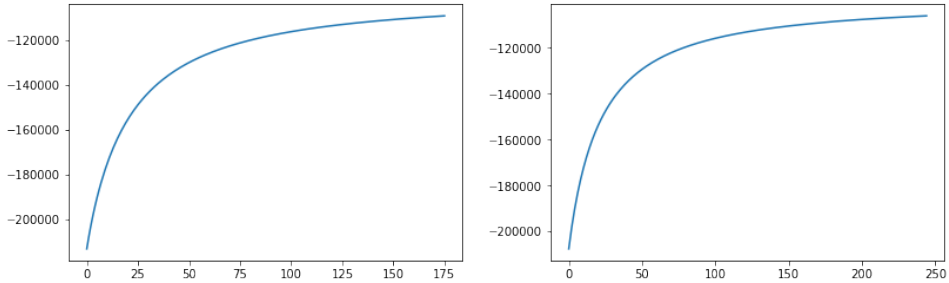


Figure 4: ELBO of implementing stochastic variational inference of a-MMSB on Douban Dataset. Left:convergence reached at iteration 176. Right: Keep running for 250 iterations

Recovering the truth in case mainly lies in retrieving the $\beta$ (community strength) for each community. We find the average commnity strength $E(\beta_k) = 0.13$ for our $100$ communities. This is relatively low considering our noise $\epsilon$ is set at $0.1$ in the first place. It could result from inaccurate number of communities specified or naive subsampling strategies.

## 6 Discussion of Douban implementation

A few things can be extended in future to improve the result of implementing a-MMSB on such a large dataset as Douban.

The first problem is how to choose the community number $k$. We specified $K = 100$ based on intuitive guess of $150000$ nodes, while actual $K$ could be $200$, $500$, etc. We do not have a labeled dataset, which means we do not know the ground truth about membership assignment. How to specify number of communities when not knowing ground truth can be our next exploration.

The second issue is the subsample strategy. Since large real-world networks are often sparse, how to subsample informative nodes makes a difference both in recovering the truth and improving the convergence speed. Any subsample should satisfy the criterion that the expectation of the stochastic gradient is equal to the true gradient. We use the simpliest method of randomly sampling nodes. There are many other possible methods to experiment on, such as stratified node sampling and neighborhood node pair search.

The third issue is runtime. We use python for simplicity of the code in this project while the original paper used C++. Though on the test toy dataset there is hardly any difference, the difference is significant when it comes to a large dataset as Douban. The digamma function and the for loop significantly slow down python. Our next step is to parallel the code in matlab.

a-MMSB opens a door to implement the traditional MMSB model large scale real network data by using stochastic variational inference. By tackling three technical issues mentioned above, it could be more useful in modeling overlapping communities on large datasets in future.

## References


[1] Gopalan, P.K. & Mimno, D. & Gerrish,S.M. Freedman, M.J. Blei.D.(2012) Scalable inference of overlapping communities. *Advances in Neural Information Processing Systems.*

[2] Fortunato.S. (2010) Community detection in graphs. *Physics Reports, 486(35):75–174*

[3] Airoldi, E. &Blei,D. &Fienberg,S. &Xing.E (2008) Mixed membership stochastic blockmodels. *Journal of Machine Learning Research, 9:1981–2014, 2008.*

[4] Hoffman, M. D. &Blei, D. &Wang, C. & Paisley, J. (2013). Stochastic variational inference. The Journal of Machine Learning Research, 14(1), 1303-1347.

[5] Hoffman,M. Blei,D. & Bach,F (2010) Online learning for latent Dirichlet allocation. In NIPS 2010.

[6] Nepusz,T. &Petrczi,A. & Ngyessy,L. &Bazs,F (2008) Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E, 77(1):016107*.