Quantifying Social Influence in Social Networks with Stochastic Block Influence Model

YAN LENG, MIT Media Lab
CHUANQUAN SHU, Master of Business Analytics, ORC, MIT

We propose Stochastic Block Influence model (SBI) to study the effect of social influence, in which individuals' adoption decisions are based on their preferences and the social influence exerted by their others in the network. Inspired by stochastic block model, we assume that social influence varies across and within social groups. We, therefore, learn both the social groups as well as the strength of influence among the social groups. With two inference algorithms, namely MAP and NUTS, we infer the group-to-group influence and group-to-group connectivity matrices. We test our model on the micro-finance adoptions in five Indian villages with cross-validation. We then analyze the group-to-group influence and empirically show that group-to-group influence may be negative, which is empirical evidence against the assumption of most social influence models. Besides, we show that there is consistently one social group that exerts stronger social influence to all other groups across the five villages. Our findings have significant implications for viral marketing and political campaigns.

ACM Reference Format:
Yan Leng and Chuanquan Shu. 2018. Quantifying Social Influence in Social Networks with Stochastic Block Influence Model. 1, 1 (May 2018), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

We now live in an increasingly connected environment. Milgram puts forward the six degrees of separation theory [1]. Backstrom discovers that people on Facebook are linked to one another with 3.57 degrees of separation [2]. The inter-connections among people not only foster the information diffusion but also enables the inter-dependencies between neighbors' decision-making. Therefore, the understanding of how social influence changes people's decision-making is essential and critical for many practical applications, such as viral marketing, political campaign, large-scale health-related behavioral change, etc.

A considerable amount of work has investigated the effect of social influence. There are three main lines of work related to social influence to our knowledge. The most relevant line of research to our study is to model social influence with a probabilistic model. The second line of research is to distinguish social influence from homophily with mostly an econometric approach [3, 4]. The third line of research, though not close to what we study but worth mentioning, is influence maximization started by Kempe [5]. In the first research line, Gomez builds a generative model of information diffusion and infer the contagion network by maximizing the likelihood of a cascading tree [6]. Myers proposes a method to model contagion using the internal exposure (within the network) and external exposure (from an external event) to information with a probabilistic model and fits parameters by maximizing the log-likelihood [7]. Tang develops a linear influence model where the influence functions of individual nodes govern the overall rate of diffusion through the network and infer the parameters using simple least-square like formulation [8].

Authors' addresses: Yan LengMIT Media Lab, yleng@mit.edu; Chuanquan ShuMaster of Business Analytics, ORC, MIT, cqgz21@mit.edu.

XXXX-XXXX/2018/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Rather than modeling the mechanical process of social influence, we primarily investigate how social influence affects human decisions. Moreover, rather than assuming the social influence always increase individuals' likelihood of adoption as most studies do, we allow social influence to be negative and empirically analyze its effect. Not constraining social influence to be positive reflects reality better. The recent developments of stochastic block model have shown to capture the actual community structure [9] successfully, but also captures the underlying mechanism of network formation [10]. Therefore, it provides a fertile building block to study more complex social processes on the social network. In particular, we study the effect of social influence using a generative model by investigating the social group to social group influences. For example, the influence of leaders to non-leaders should be higher than that the reverse way of non-leaders to non-leaders. Building upon the stochastic block model, we infer the social groups each belongs to and analyze the within-group and inter-group social interactions simultaneously in our Stochastic Block Influence model.

The remaining sections are organized as follows. Section 2 describes the Stochastic Block Inference model (SBI) we propose in this study. The inference is covered in Section 3. Then, we describe our experiment setting and analyze the results. Lastly, we conclude and state future directions.

2 MODEL

We are interested in studying the effect of unobserved but significant mechanisms on individuals' decision-making process in social networks, namely the social influence. To do so, we borrow ideas from Mixed Membership Stochastic Blockmodel (MMSB) developed in [11], which provides a framework to learn the latent block structure in a community. Just like in MMSB, our model (Stochastic Block Influence; SBI) learns a mixed membership for an individual. In our SBI, we use a simpler method to learn the block structure, and in addition to learning group-to-group connectivity, our model also learns group-to-group influence.

The basic hypothesis of the SBI model is described as follows. There exist different social groups of people (blocks) in a network. Take an individual in the network, Smith. Smith's product adoption behavior is semi-determined by the group-to-group influence based on Smith's group membership, which si probabilistic, and the group memberships of everyone else in the network, weighted by their connectivity with Smith. In reality, outside influence accounts for half of the puzzle. Smith's characteristics [1] also contributes to his adoption status and hence are incorporated in SBI.

Assume there are K blocks in the given network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The latent variables to be learned are as follows:

- $\mathbf{b} \sim Norm(\boldsymbol{\mu}_b, \boldsymbol{\sigma}_b) \in \mathbb{R}^{1\times Dim}$: the linear model coefficients for a individual's attributes
- $\mathbf{m_i} \sim Dir(\mathbf{c}_i) \in \mathbb{R}^{1\times K}$: i^{th} individual's distribution across K groups; $\mathbf{c}_i$ is the concentration parameter.
- $\mathbf{B} \in \mathbb{R}^{K\times K}$: the group-to-group connectivity matrix; $\mathbf{B}_{ij} \sim Beta(\alpha_{ij}, \beta_{ij})$
- $\mathbf{F} \in \mathbb{R}^{K\times K}$: the group-to-group influence matrix; $\mathbf{F}_{ij} \sim Beta(\mu_{ij}, \sigma_{ij})$

The observations are as follows:

- $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$: the binary adjacency matrix representing the connectivity between two individual individuals
- $\mathbf{y} \in \mathbb{R}^{|\mathcal{V}|\times 1}$: the binary vector representing individual individuals' adoption status

The generative process of our model is as follows:

(1) For each node $i \in \mathcal{V}$, draw a K-dimensional mixed membership vector m_i. Together, they form the villagers matrix $M \in \mathbb{R}^{|\mathcal{V}|\times K}$
(2) Draw group-to-group connectivity matrix $\mathbf{B}$
(3) Draw group-to-group influence matrix $\mathbf{F}$
(4) Draw linear model coefficients $\mathbf{b}$

[1] $\mathbf{x}_{Smith} \in \mathbb{R}^{1\times Dim}$ where Dim is the number of features; together they form the feature matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times Dim}$ where $|\mathcal{V}|$ is the number of individuals

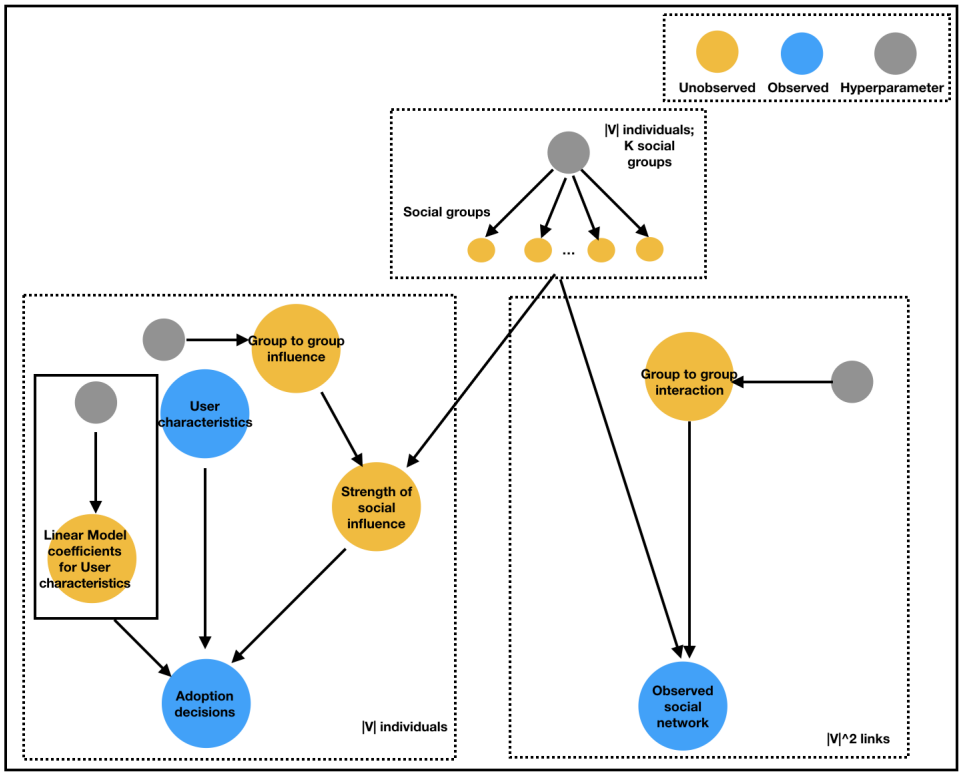

Fig. 1. Graphical representation of Influence Model. We use three nodes, yellow, blue and gray, to denote observed variables, unobserved variables, and hyper-parameters. We assume people's decision process is determined by their characteristics, the characteristics of the product, as well as the social influence exerted by their neighbors. Social influence is determined by the individual's social group, as well as his neighbor's social group. The social group is also related to the social network.

(5) The observations are generated by:
- $\mathbf{A}$ is then generated by MBM^T
- Denote $Mat = (MFM^T \circ A)$, where $\circ$ is the element-wise Matrix multiplication. $\mathbf{y}$ is then generated by $Xb + \sum_{j \in \{1,\ldots,|V|\}} Mat_{ij}$

For short-hand, We denote $Z = \{\mathbf{b}, \mathbf{m}, B, F\}$ and $\theta = \{\boldsymbol{\mu}_b, \boldsymbol{\sigma}_b, \mathbf{c}_i, \forall_{ij}\ \mu_{F_{ij}}, \sigma_{F_{ij}}, \alpha_{ij}, \beta_{ij}\}$
The graphical representation of our model is shown in Figure 1.

3 INFERENCE

3.1 Maximum a Posteriori

We use Maximum a Posteriori (MAP) Estimation to infer the posterior distribution of the latent variables Z. While there are many other inference techniques, such as Variational Inference or Variational Expectation Maximization, we only explore MAP primarily for its speed.
Essentially, MAP algorithm uses modes to approximate latent variables' posterior distributions as shown in the

following equation.

$$Z_{MAP} = \underset{z}{argmax} P(Z|\mathbf{y}_{train}, A; \theta) = \underset{z}{argmax} logP(Z|\mathbf{y}_{train}, A; \theta)$$

Log density is used to circumvent some numerical issue. Note that the above optimization is hard to solve because the posterior is intractable. Hence, Bayes Rule is applied such that

$$Z_{MAP} = \underset{z}{argmax} P(Z|\mathbf{y}_{train}, A; \theta) = \underset{z}{argmax} P(Z, \mathbf{y}_{train}, A; \theta) \tag{1}$$

This is valid because $logP(Z|\mathbf{y}_{train}, A; \theta) = logP(Z, \mathbf{y}_{train}, A; \theta) - logP(\mathbf{y}_{train}, A; \theta)$, where $logP(\mathbf{y}_{train}, A; \theta)$ is constant with respect to Z. The optimization part is done using gradient descent. This algorithm is implemented with the Edward Library in Python and it terminates with point estimates of all latent variables.

3.2 MCMC-based Sampling

We primarily use No-U-Turn-Sampling (NUTS) to estimate the posterior distributions of the latent variables Z. There are many others methods to do MCMC sampling; the notable ones are Gibbs, Metropolis, Hamiltonian Markov Chain (HMC). We stay away from Gibbs and Metropolis because they tend to be very slow as they explore the parameter space in a random walk fashion by design [12]. As we have modified a villager's block assignment from the binary vector (as in the standard SBM) to continuous vector, we can then use HMC or NUTS.

HMC is a lot faster than Gibbs and Metropolis because HMC can suppress the random walk behavior by adding auxiliary "momentum" variable. As a result, principles in Hamiltonian dynamics can be applied to simulate MCMC transition motion (rather than with a random walk) [13]. While HMC is promising, we still stay away from it because its performance and speed can be highly dependent on two key hyperparameters: the number of leapfrog steps and the leapfrog step size. Although HMC is relative fast, to finely tune those two hyperparameters, on top of all the other ones, is very demanding. This is why we only use NUTS, which is HMC that can dynamically adapt to the best numbers of steps and step sizes. We use the pymc3 package in Python.

The model setup in NUTS sampling is similar to that in the MAP. The joint log-likelihood is again

$$logP(Z, \mathbf{y}_{train}, A; \theta)$$

Some critical hyperparameters for NUTS are the number of burn-in samples, the number of samples after burn-in, the target acceptance probability and the number of chains. For all of our NUTS sampling runs, we burn 3,000 samples to ensure that MCMC mostly converge to the actual posterior distribution. The number of samples after burn-in is 500; usually, only less than ten samples (among the 500) are diverging. Next, we select the target acceptance probability to be 0.8. This selection might be considered high, given the theoretical optimal value for HMC is 0.65 [13], and likely restrain the adapted leapfrog step size. However, given the complex models in our analyses, we feel 0.8 is a suitable value. Finally, we only use one chain per simulation because we have a sizable number of simulations to run under a limited parallel computing environment.

At the end of each run, we average across the 500 samples to derive point estimates for all latent variables. Note that this is different from the MAP which uses mode.

3.3 Testing

With either MAP or NUTS, we can get the point estimates for all latent variables Z. We then rerun our model (as described in section 2) with all latent variables fixed to the estimates on the Test dataset. This returns the predicted adoption probability for each villager in the Test data. Performance is evaluated using the area under the ROC curve constructed using the predicted adoption probabilities and the actual adoption outcomes. The next section provides more details on the experiment setup.

4 EXPERIMENTS

The data we use is collected by JPAL in Indian villages [14]. Out of the 46 villages, We choose 5 with the most number of villagers, indexed by 3, 52, 59, 60 and 71. The researchers collected data on not only the relationships between households but also a variety of amenities of households, including roofing materials, type of latrine, quality of access to electric power, and so on. The amenities are used as the predictors ($\mathbf{X}$). The outcome variable is the adoption on micro-finance.

Model evaluation. To train our model and evaluate the performance for a K (number of social groups), we cross validate by randomly splitting the data into 75% training samples and 25% test samples. We repeat this process by ten times (with different seed) for each village (50 times in total). In our model, we first learn the social groups for all individuals in the training step. Moreover, then use the learned social group, the influence matrix among groups, as well as the learned coefficients for predictors to predict adoption. Note that the entire A (villager to villager connection) matrix is used during training (for uncovering the social groups). This is allowed because A or the derived group memberships does not contain the ground truth y. We evaluate our method using AUC since our data is imbalanced. AUC is the area under the Receiver-Operating-Characteristics curve, which is plotted by the false positive rate and correct positive rate for different thresholds.

Baselines. We run a logistic regression model with the researcher-collected households amenities data ($\mathbf{X}$) as the independent variables and the adoption outcomes as the dependent variables (Model attr_only).

SBI model learns group-wise connection (B), group-wise influence (F) and regression coefficients ($\mathbf{b}$) at once, all using Bayesian. Alternatively, this can be viewed in two stages:

(1) To learn the social groups (B,$\mathbf{m}_i$).
(2) With learned social groups, learn group-wise influence and predict adoption outcomes.

To study whether the learned social groups makes, sense, we also evaluate performances under this two-stage view. In particular, we first use SBM to learn the social groups (B,$\mathbf{m}_i$) and then use Logistic Regression to predict adoption outcomes with $\mathbf{m}_i$ as features (Model group_only) or both $\mathbf{m}_i$ and $\mathbf{X}$ as features (Model attr+group). See Figure 5 for performance.

Hyperparameter Tuning. We mainly tune three groups of hyper-parameters, $\boldsymbol{\sigma}_b$, μ_F and σ_F. Since in our baseline model, all the $\mathbf{b}$ are within the range of [-2, 2], we choose $\boldsymbol{\sigma}_b = 1.5$. Note that a large $\boldsymbol{\sigma}_b$ may overfit and a small $\boldsymbol{\sigma}_b$ makes it hard to fit the model. We use two ways to tune μ_F and σ_F; one is grid search, the other is imposing a Uniform uninformative before the two hyperparameters. Due to the slow converging speed with the sampling method, we use the MAP to tune such parameter and feed into our sampling method by fixing the number of social groups to be 10. The tuning result are shown in Figure 2.

Number of social groups. With the learned hyperparameters from the previous step, we tune the number of social groups, as shown in Figure 3. We see that as we increase the number of social groups, the performance for the training set increases, which is a sign of overfitting, while the performance of test set does not change much by comparing the median. However, we see that the standard deviations of the test are smallest when the number of the social group is 7. Therefore, we select seven as the number of social groups.

Performance. We compare the performance of our model with various baselines (Model attr_only, Model group_only, Model attr+group). We observe that except for MAP, all other methods shown in Figure 4 overfit to a large degree in the training set. On the test set, MAP slightly beats the Model attr_only and the Model attr+group. NUTS performs better than MAP perhaps because of the MAP, with its gradient descent optimization, may get stuck at local minima. However, the best-performing model is the baseline with the only learned group as the attribute, Model group_only.

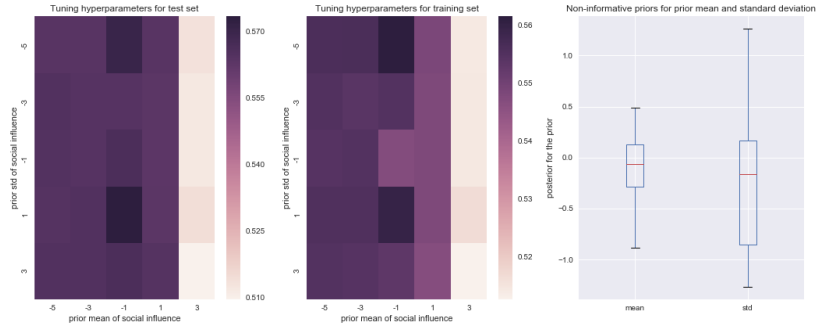

Fig. 2. Tuning hyper-parameters for the MAP. The x-axis and y-axis for the left-most and central plot are the priors for μ_F and σ_F. We do cross-validation on each village for ten times and aggregate the performance by the mean as shown in the color-code in these two plots. The rightmost plot is the posterior for the learned hyperparameter of μ_F and σ_F with Uniform uninformative priors from 50 random splits with the same random seed (10 times for each village).

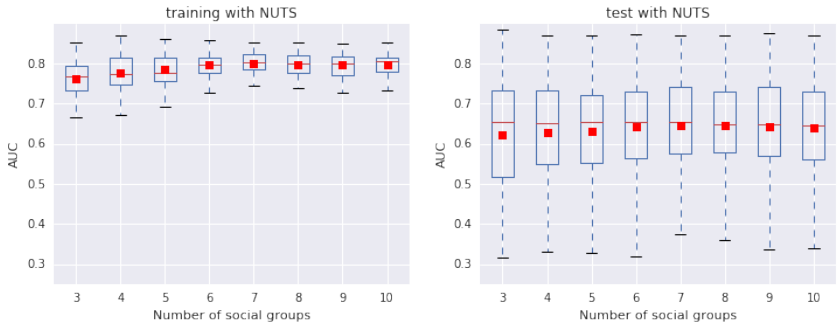

Fig. 3. Number of social groups vs. AUC in cross-validated training and test set. The left and right plots are the performance on the training and test set respectively with a different number of social groups. The red-filled square and red line denotes the mean and median. The upper and lower ends of the boxes show the 75% and 25% percentile

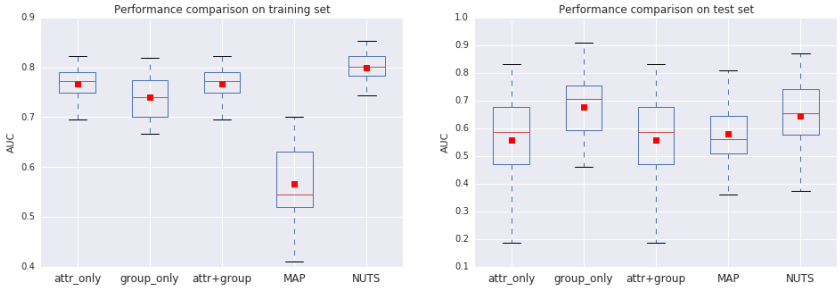

Fig. 4. Performance comparisons. The left and left plots are the performance on the training and test set respectively. The red-filled square denotes the mean; the red line denotes the median. Moreover, the upper and lower end of the box show the 75% and 25% percentile of performance.

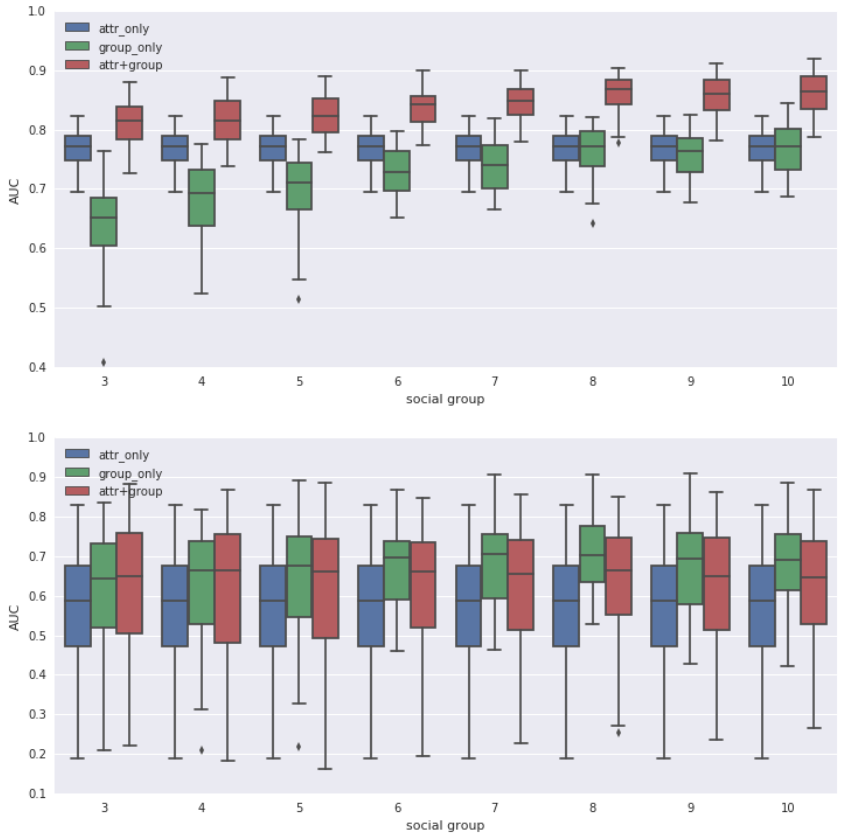

Fig. 5. Effectiveness of learned social groups. The blue bars are for the baseline Model attr_only; the green bars are for the baseline Model group_only; the red bars are for the baseline Model attr+group. Top panel for Train; Bottom panel for Test

5 DISCUSSION

Group assignment and node characteristics . Figure 5 displays the AUC metrics for both Train and Test scenarios. The blue boxplots are the same across a different number of social groups because it only uses the X as features which are independent of the number of social groups.

The key finding here is that in the Test scenario, Model group_only performs the best across the board. This indicates that the learned social groups (m_i for each villager i) are predictive.

Note that the in the test set, the model suffers from the high variance. Observations from the Model attr_only also exhibit high variance, which indicates that there is high variance in the data. Note that in each village, there are around 300 individuals, 25% test set is a small size, which may be the reason that contributes to the high variance. Furthermore, the variance for Model group_only's performance decreases as the number of social groups (K) increases to 8 and then slightly increases again as K increases towards 10. This indicates that classifying people into eight social groups might be the most reasonable.

Influence Matrix. We show the group-wise social influence and group-wise connection in Figure 6. As shown in the lower panel, the connections across groups, as shown in the lower panel, are very weak. There is consistently

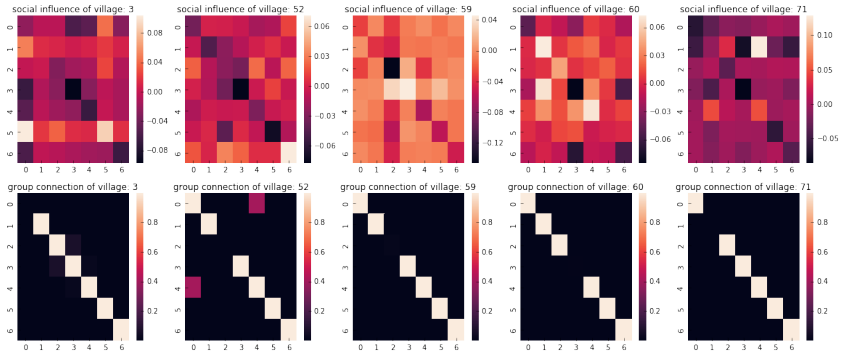

Fig. 6. Group-wise social influence (upper panel) and group-wise connection (lower panel). The indexes for the villages are 3, 52, 59, 60 and 71 from left to right.

one social group where the within-group connections are very low. This is because there are individuals in the network who are not very connected to other individuals. In village 3, 52, 59 and 60, there is one group whose influence on all other groups are stronger. Interestingly, many across-group social influences are negative after controlling for social influence.

Problems in the study. As we see in the previous discussion, a simple stochastic block model can successfully extract social group information for each villager. However, SBI, though beat other models, does not outperform group_only model. We think this is because the large range of hyperparameters, which may vary significantly across different villages. Thus, far more tuning is needed.

REFERENCES

[1] Jeffrey Travers and Stanley Milgram. The small world problem.
[2] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. Four degrees of separation. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 33–42. ACM, 2012.
[3] Sinan Aral, Lev Muchnik, and Arun Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, 106(51):21544–21549, 2009.
[4] Joshua D Angrist. The perils of peer effects. *Labour Economics*, 30:98–108, 2014.
[5] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.
[6] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):21, 2012.
[7] Seth A Myers, Chenguang Zhu, and Jure Leskovec. Information diffusion and external influence in networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 33–41. ACM, 2012.
[8] Jaewon Yang and Jure Leskovec. Modeling information diffusion in implicit networks. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 599–608. IEEE, 2010.
[9] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *arXiv preprint arXiv:1703.10146*, 2017.
[10] Matthew O Jackson. *Social and economic networks*. Princeton university press, 2010.
[11] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014, 2008.
[12] M. D. Hoffman and A. Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *ArXiv e-prints*, November 2011.
[13] R. M. Neal. MCMC using Hamiltonian dynamics. *ArXiv e-prints*, June 2012.
[14] Abhijit Banerjee, Arun G Chandrasekhar, Esther Duflo, and Matthew O Jackson. The diffusion of microfinance. *Science*, 341(6144):1236498, 2013.