Supporting Information

Gopalan and Blei 10.1073/pnas.1221839110

SI Text

Introduction

This document is organized as follows. We first develop the stochastic variational inference (SVI) algorithm (1) for the mixed-membership stochastic block model (MMSB) (2) described in the article. Each iteration of the algorithm subsamples the network and updates its estimate of the community structure. We extend the algorithm to allow for nonuniform sampling from the network, and study a number of sampling strategies. We then develop SVI with link sampling, an algorithm whose per-iteration complexity scales linearly in the number of links. Finally, we present supporting material for the empirical study on real and synthetic networks.

SVI

The article describes a subclass of the MMSB (2, 3) that is appropriate for community detection in assortative undirected networks. In this section, we present SVI for the MMSB (3).

In variational inference, we define a family of distributions over the hidden variables $q(\boldsymbol{\beta},\boldsymbol{\theta},\boldsymbol{z})$ and find the member of that family that is closest to the true posterior. Closeness is measured with Kullback–Leibler (KL) divergence (4). We use the mean-field family, under which each variable is endowed with its own distribution and its own variational parameter. This allows us to tractably optimize the parameters to find a local minimum of the KL divergence. The mean-field variational family for the MMSB, with N nodes and K communities and with Beta priors placed over the community strengths $\boldsymbol{\beta}$, is as follows:

$$q(\boldsymbol{\beta},\boldsymbol{\theta},\boldsymbol{z}) = \prod_{k=1}^{K} q(\beta_k|\lambda_k) \prod_{n=1}^{N} q(\theta_n|\gamma_n) \prod_{i<j} q\left(z_{i\to j}|\phi_{i\to j}\right) q\left(z_{i\leftarrow j}|\phi_{i\leftarrow j}\right). \quad [S1]$$

Here, the variational distributions are $q(z_{i\to j}=k)=\phi_{i\to j,k}$, $q(\theta_n)=\text{Dirichlet}(\theta_n;\gamma_n)$, and $q(\beta_k)=\text{Beta}(\beta_k;\lambda_k)$. The posterior over $\mathbf{z}$ is parameterized by the interaction parameters $\boldsymbol{\phi}$, the posterior over $\boldsymbol{\theta}$ is parameterized by the community memberships $\boldsymbol{\gamma}$, and the posterior over $\boldsymbol{\beta}$ is parameterized by the community strengths $\boldsymbol{\lambda}$.

Minimizing the KL divergence between q and the true posterior is equivalent to optimizing an "evidence lower bound" (ELBO) $\mathcal{L}$, a bound on the log likelihood of the observations (5, 6). The ELBO is as follows:

$$\begin{aligned}\mathcal{L} = &\sum_k \mathrm{E}_q\left[\log p(\beta_k|\eta)\right] - \sum_k \mathrm{E}_q\left[\log q(\beta_k|\lambda_k)\right] \\ &+ \sum_n \mathrm{E}_q\left[\log p(\theta_n|\alpha)\right] - \sum_n \mathrm{E}_q\left[\log q(\theta_n|\gamma_n)\right] \\ &+ \sum_{a,b} \mathrm{E}_q\left[\log p(z_{a\to b}|\theta_a)\right] + \mathrm{E}_q\left[\log p(z_{a\leftarrow b}|\theta_b)\right] \\ &- \sum_{a,b} \mathrm{E}_q\left[\log q(z_{a\to b}|\phi_{a\to b})\right] - \mathrm{E}_q\left[\log q(z_{a\leftarrow b}|\phi_{a\leftarrow b})\right] \\ &+ \sum_{a,b} \mathrm{E}_q\left[\log p(y_{ab}|z_{a\to b}, z_{a\leftarrow b}, \boldsymbol{\beta})\right]. \end{aligned} \quad [S2]$$

The expectations in Eq. **S2** are taken with respect to the variational distribution q. Notice the first two lines in Eq. **S2** contain summations over communities and nodes; we call these "global terms." They relate to the "global variables," which are the community strengths and community memberships. The remaining lines contain summations over all node pairs, which we call "local terms." They depend on both the global and "local variables," the latter being the interaction memberships.

SVI optimizes the ELBO using stochastic gradient ascent. Stochastic gradient algorithms follow noisy estimates of the gradient with a decreasing step size. These algorithms are guaranteed to converge to a local optimum if the expectation of the noisy gradient is equal to the gradient and if the step-size decreases according to a certain schedule (7). In SVI, we form noisy gradients by subsampling the network. This leads to a scalable algorithm because it avoids the expensive all-pairs sums in the ELBO.

Existing SVI methods require the data be sampled uniformly to form noisy gradients (1). We now develop an SVI algorithm that allows for nonuniform samples of links and nonlinks at each iteration. We then present SVI with link sampling, an algorithm that samples only the links of a network.

SVI with Nonuniform Sampling. SVI iteratively updates the local and global parameters. At each iteration, it first subsamples the network. It then computes the optimal local parameters of the subset—the $(\phi_{i\to j}, \phi_{i\leftarrow j})$ for each sampled node pair (i,j)—given the current settings of the global parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\lambda}$. Finally, it updates the global parameters using a noisy natural gradient (8) computed from the subsampled data and the optimized local parameters. The first phase is the local step; the second phase is the global step (1).

The pseudocode of SVI for the MMSB is as follows:

1. Initialize global parameters $\boldsymbol{\gamma}=(\gamma_n)_{n=1}^N$, $\boldsymbol{\lambda}=(\lambda_k)_{k=1}^K$.
2. Subsample a set $\mathcal{S}$ of node pairs.
3. Local step. For each pair $(i,j)\in\mathcal{S}$, compute the optimal interaction parameters $\phi_{i\to j}$ and $\phi_{i\leftarrow j}$ as a function of the global parameters.
4. Global step.
   - For each node a, compute the community membership natural gradients $\partial\gamma_a^t$ and update γ_a.
   - For each community k, compute the community strength natural gradients $\partial\lambda_k^t$ and update λ_k.
5. Repeat.

The subsampling of the network in each iteration provides a way to plug in a variety of network sampling algorithms into the estimation procedure. However, to maintain a correct stochastic optimization algorithm of the variational objective, the subsampling method must be valid. That is, the noisy gradients estimated from the subsample must be unbiased estimates of the true gradients.

The Global Step. The global step updates the global community strengths $\boldsymbol{\lambda}$ and community memberships $\boldsymbol{\gamma}$ with a stochastic gradient of the ELBO in Eq. **S2**. The ELBO contains summations over all $O(N^2)$ node pairs. Consider drawing a node pair (a,b) at random from a distribution $g(a,b)$ over the $M=N(N-1)/2$ node pairs. We can rewrite the ELBO as a random function of the variational parameters that includes the global terms and the local terms associated only with (a,b). The expectation of this random function is equal in objective to Eq. **S2**.

For example, the term in the fifth line in Eq. **S2** is rewritten as follows:

$$\sum_{a,b} \mathrm{E}_q\left[\log p(y_{ab}|z_{a\to b}, z_{a\leftarrow b}, \boldsymbol{\beta})\right] = \mathrm{E}_g\left[\frac{1}{g(a,b)} \mathrm{E}_q\left[\log p(y_{ab}|z_{a\to b}, z_{a\leftarrow b}, \boldsymbol{\beta})\right]\right]. \quad \text{[S3]}$$

Evaluating the rewritten Eq. **S2** for a node pair sampled from g gives a noisy but unbiased estimate of the ELBO. Following (1), the stochastic natural gradients computed from a sample pair (a,b), at iteration t, are as follows:

$$\partial\gamma^t_{a,k} = \alpha_k + \frac{1}{g(a,b)}\phi_{a\to b,k} - \gamma^{t-1}_{a,k}, \quad \text{[S4]}$$

$$\partial\lambda^t_{k,i} = \eta_i + \frac{1}{g(a,b)}\phi_{a\to b,k}\cdot\phi_{a\leftarrow b,k}\cdot y_{ab,i} - \lambda^{t-1}_{k,i}, \quad \text{[S5]}$$

where $y_{ab,0} = y_{ab}$, and $y_{ab,1} = 1 - y_{ab}$. In practice, we sample a "mini-batch" S of pairs per update, to reduce noise.

The update in Eq. **S4** can be interpreted as follows. When a single pair (a,b) is sampled, we find a noisy natural gradient in Eq. **S4** by computing the community memberships $\boldsymbol{\gamma}$ that would be optimal (given interaction parameters $\boldsymbol{\phi}$) if our entire network were a multigraph containing the interaction y_{ab} repeated $1/g(a,b)$ times.

Once the noisy gradients are computed, the global step follows it with an appropriate step size,

$$\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} + \rho_t\partial\boldsymbol{\gamma}^t;\ \boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \rho_t\partial\boldsymbol{\gamma}^t. \quad \text{[S6]}$$

We require that $\sum_t \rho_t^2 < \infty$ and $\sum_t \rho_t = \infty$ for convergence to a local optimum (7). We set $\rho_t \triangleq (\tau_0 + t)^{-\kappa}$, where $\kappa \in (0.5, 1]$ is the forgetting rate and the delay $\tau_0 \geq 0$ downweights iterations (1).

Set-Based Sampling. Our algorithm has assumed that the subset of node pairs S are sampled independently. We can relax this assumption by defining a distribution over predefined sets of pairs. These sets can be defined using information about the pairs, such as network topology, which lets us take advantage of more sophisticated sampling strategies. For example, we can define a set for each node that contains the node's adjacent links and nonlinks. At each iteration, we sample one of these sets at random.

We set two constraints to ensure that set-based sampling results in unbiased gradients. First, the union of the sets s must be the total set of all node pairs, $U : U = \cup_i s_i$. Second, every pair (a,b) must occur in some constant number of sets $c \geq 1$. With these conditions satisfied, we can again rewrite Eq. **S2** as the sum over its global terms and an expectation over the local terms. Let $h(t)$ be a distribution over the sets. For example, the term in the fifth line in Eq. **S2** can be written as follows:

$$\sum_{a,b}\mathrm{E}_q[\log p(y_{ab}|z_{a\to b}, z_{a\leftarrow b}, \boldsymbol{\beta})] = \mathrm{E}_h\left[\frac{1}{c}\frac{1}{h(t)}\sum_{(a,b)\in s_t}\mathrm{E}_q[\log p(y_{ab}|z_{a\to b}, z_{a\leftarrow b}, \boldsymbol{\beta})]\right]. \quad \text{[S7]}$$

Under set-based sampling, the stochastic gradients of the ELBO are as follows:

$$\partial\gamma^t_{a,k} = \alpha_k + \frac{1}{c}\frac{1}{h(t)}\sum_{(a,b)\in s_t}\phi_{a\to b,k} - \gamma^{t-1}_{a,k}, \quad \text{[S8]}$$

$$\partial\lambda^t_{k,i} = \eta_i + \frac{1}{c}\frac{1}{h(t)}\sum_{(a,b)\in s_t}\phi_{a\to b,k}\cdot\phi_{a\leftarrow b,k}\cdot y_{ab,i} - \lambda^{t-1}_{k,i}, \quad \text{[S9]}$$

where $y_{ab,0} = y_{ab}$, and $y_{ab,1} = 1 - y_{ab}$. The global steps are the same as in Eq. **S6**.

The Local Step. The local step optimizes the interaction parameters $\boldsymbol{\phi}$ with respect to a subsample of the network. Recall that there is an interaction membership parameter for each node pair—$\phi_{a\to b}$ and $\phi_{a\leftarrow b}$—representing the posterior approximation of which communities are active in determining whether there is a link. We optimize these parameters in parallel. (We will discuss an alternative local step optimization for the interaction parameters in *Link Sampling*.) The update for $\phi_{a\to b}$ given y_{ab} is as follows:

$$\begin{aligned}\phi^t_{a\to b,k}|y_{ab}=0 &\propto \exp\Big\{\mathrm{E}_q\left[\log\theta_{a,k}\right] + \phi^{t-1}_{a\leftarrow b,k}\mathrm{E}_q[\log(1-\beta_k)] \\ &\quad + (1-\phi^{t-1}_{a\leftarrow b,k})\log(1-\epsilon)\Big\} \\ \phi^t_{a\to b,k}|y_{ab}=1 &\propto \exp\Big\{\mathrm{E}_q\left[\log\theta_{a,k}\right] + \phi^{t-1}_{a\leftarrow b,k}\mathrm{E}_q[\log\beta_k] \\ &\quad + (1-\phi^{t-1}_{a\leftarrow b,k})\log\epsilon\Big\}.\end{aligned} \quad \text{[S10]}$$

The updates for $\phi_{a\leftarrow b}$ are symmetric. Thus, we iteratively update $\phi^t_{a\to b,k}$ using $\phi^{t-1}_{a\leftarrow b,k}$ and $\phi^t_{a\leftarrow b,k}$ using $\phi^{t-1}_{a\to b,k}$ until convergence. In Eq. **S10**, t counts the iterations within the local step. This is natural gradient ascent with a step size of 1.

Sampling Strategies. Our algorithm is flexible about how the subset of pairs is sampled, as long as the expectation of the stochastic gradient is equal to the true gradient. We can choose the distribution over pairs to sample from independently or choose the distribution over sets. There are several options.

Random pair sampling. The simplest method is to sample node pairs uniformly at random. This method is an instance of independent pair sampling, with $g(a,b)$ (used in Eq. **S3**) equal to $\frac{1}{N(N-1)/2}$.

Random node sampling. This method focuses on local neighborhoods of the network. A set consists of all of the pairs that involve one of the N nodes. At each iteration, we sample a set uniformly at random from the N sets, so $h(t) = 1/N$. Because each pair involves two nodes, each link or nonlink appears in two sets and so $c = 2$. Following Eq. **S8** and Eq. **S9**, we compute the stochastic gradients from a sampled node a as follows:

$$\partial\gamma^t_{a,k} = \alpha_k + \frac{N}{2}\sum_{(a,b)}\phi_{a\to b,k} - \gamma^{t-1}_{a,k}, \quad \text{[S11]}$$

$$\partial\lambda^t_{k,i} = \eta_i + \frac{N}{2}\sum_{(a,b)}\phi_{a\to b,k}\cdot\phi_{a\leftarrow b,k}\cdot y_{ab,i} - \lambda^{t-1}_{k,i}, \quad \text{[S12]}$$

where $y_{ab,0} = y_{ab}$, and $y_{ab,1} = 1 - y_{ab}$. In practice, we sample a "mini-batch" of nodes per update, to reduce noise.

Informative set sampling. The idea behind this method is to sample a set of pairs around each node with a bias toward pairs that help estimation. This is a type of set-based sampling.

For each node a, we define an "informative set" consisting of all of its links and a small number of nonlinks. In our experiments, we chose nonlinks to nodes that are at most h hops from the node a. (We set $h = 2$.) Such nodes may be more relevant to estimating the communities of node a (9). For each node, we also define m "noninformative sets" that partition its remaining nonlinks. Because the number of nonlinks associated with each node is large, dividing them into many sets allows the computation in each iteration to be fast. At each iteration, we select a node uniformly at random from the N nodes and choose the informative set with high probability by flipping a biased coin. Otherwise, with low probability, we select one of the m noninformative sets of the selected node. To compute Eq. **S7**, we note that $c = 2$ and the distribution over sets is

$$h(t) \propto \begin{cases} (1-\xi)\dfrac{1}{N} & \text{if the set is informative} \\ \xi\dfrac{1}{Nm} & \text{if the set is non-informative.} \end{cases} \quad \text{[S13]}$$

Note that we set $\xi \ll 1$. We describe additional sampling methods in ref 3.

Link Sampling. The above subsampling methods include the network nonlinks. Many real networks are sparse and only a small fraction of their node pairs are links (Table S2). As the number of nodes increases, subsampling nonlinks becomes increasingly inefficient. Here, we consider "link-based variational inference" and "link sampling," a subsampling approach that involves only the links in the network. We develop this algorithm by assuming that a node's nonlinks are explained by the same communities that a node exhibits while generating links.

We specify the variational family in a particular way to focus on the links. It differs from the family in Eq. **S1** in the variational interaction parameters for the links. The new family specifies an interaction parameter for the joint distribution over the pair of node community indicators of each link. The interaction parameters for the nonlinks remain the same as in Eq. **S1**. We then constrain the nonlink interaction parameters of each node to equal the mean of the link interaction parameters of that node. In the following discussion, *links(a)* is the set of links of node a in the training set, and *links* are the set of all links in the training set. We define the sets for nonlinks similarly.

In particular, we use the following family in link-based variational inference,

$$\begin{aligned} q(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{\beta}) = &\prod_{n=1}^{N} q(\theta_n|\gamma_n) \prod_{(i,j)\in\text{links}} q\left(z_{i\to j}, z_{i\leftarrow j}|\phi_{ij}\right) \\ &\prod_{(i,j)\in\text{nonlinks}} q\left(z_{i\to j}|\phi_{i\to j}\right) q\left(z_{i\leftarrow j}|\phi_{i\leftarrow j}\right) \\ &\prod_{k=1}^{K} q(\beta_k|\lambda_k). \end{aligned} \quad \text{[S14]}$$

We constrain the interaction parameters of each nonlink (i, m) of a node i,

$$\phi_{i\to m,k} = \frac{\sum_{(i,j)\in\text{links}(i)}\sum_{l=1}^{K}\phi_{ij}^{kl}}{d_i} = \frac{\sum_{(i,j)\in\text{links}(i)}\phi_{ij}^{kk}}{d_i} = \overline{\phi}_{i,k}, \quad \text{[S15]}$$

where d_i is the degree of node i in the training set.

The simplification in Eq. **S15** arises because $\sum_{k\neq l}\phi_{ij}^{kl} = 0$. When $k \neq l$, the community strength parameters are the nondiagonal entries of the block model, each set to ε. Because $\epsilon \to 0$ by our modeling assumption of assortativity, when $k \neq l$, $\phi_{ij}^{kl} \propto \exp\{-\infty\}$. Notice that because $\sum_{k=1}^{K}\phi_{ij}^{kk} = 1$, $\overline{\phi}_i$ is normalized.

The ELBO in the link-based variational inference is a function of the variational parameters $(\phi_{\text{links}}, \overline{\phi}, \boldsymbol{\gamma}, \boldsymbol{\lambda})$. The ϕ_{links} are the $M \times K$ matrix of interaction parameters defined over the links, where M is the number of links in the training set. The $\overline{\phi}$ are the $N \times K$ matrix of the mean interaction parameters. We can compute the optimal ϕ_{ab}, given a link $y_{ab} = 1$, while fixing the other parameters:

$$\phi_{ab}^{kk} \propto \exp\{E_q \log \theta_{ak} + E_q \log \theta_{bk} + E_q \log \beta_k\}. \quad \text{[S16]}$$

The natural gradient of the ELBO with respect to the node's community memberships is as follows:

$$\begin{aligned} \partial\gamma_{a,k}^{t} &= \alpha_k + \sum_{(a,b)\in\text{links}(a)} \phi_{ab}^{kk} + \sum_{(a,b)\in\text{nonlinks}(a)} \phi_{a\to b,k} - \gamma_{a,k}^{t-1} \\ &= \alpha_k + \sum_{(a,b)\in\text{links}(a)} \phi_{ab}^{kk} + c_a\overline{\phi}_{a,k} - \gamma_{a,k}^{t-1}, \end{aligned} \quad \text{[S17]}$$

where c_a is the number of nonlinks of node a in the training set. The natural gradient of the ELBO with respect to the community strengths is as follows:

$$\begin{aligned} \partial\lambda_{k,0}^{t} &= \eta_0 + \sum_{(a,b)\in\text{links}} \phi_{ab}^{kk} - \lambda_{k,0}^{t-1} \\ \partial\lambda_{k,1}^{t} &= \eta_1 + \sum_{(a,b)\in\text{nonlinks}} \phi_{a\to b,k}\phi_{a\leftarrow b,k} - \lambda_{k,1}^{t-1}. \end{aligned} \quad \text{[S18]}$$

We can rewrite Eq. **S18** as a function of only the link interaction parameters using the following:

$$\begin{aligned} \sum_{(a,b)\in\text{nonlinks}} \phi_{a\to b,k}\phi_{a\leftarrow b,k} &= \sum_{(a,b)\in\text{nonlinks}} \overline{\phi}_{a,k}\overline{\phi}_{b,k} \\ &= \left(\sum_n \overline{\phi}_{n,k}\sum_n \overline{\phi}_{n,k} - \sum_n \left(\overline{\phi}_{n,k}\right)^2\right)\Big/2 \\ &\quad - \sum_{a,b\in\text{links}} \overline{\phi}_{a,k}\overline{\phi}_{b,k}. \end{aligned} \quad \text{[S19]}$$

We have expressed the natural gradients of the community memberships and community strengths as a function of ϕ_{links} and $\overline{\phi}$. We now describe a SVI algorithm that iterates only over the links.

Our link subsampling method extends random node sampling. The structure of the algorithm is similar to the general SVI algorithm, with a subsampling step, local steps, and global steps. At each iteration, we sample a node uniformly at random and observe all of its training links. In practice, we sample a minibatch of nodes. In the local step, we iterate over the links and compute the optimal link interaction parameters using Eq. **S16**. We then compute the mean interaction parameters of the sampled nodes using Eq. **S15**.

As with the previous sampling methods, we consider the stochastic optimization of the global community strengths $\boldsymbol{\lambda}$ and the global community memberships $\boldsymbol{\gamma}$. Previously, we obtained community membership gradients with respect to the entire vector $\boldsymbol{\gamma}$ of dimension $N \times K$.

In link sampling, we optimize the community memberships of each node separately, using distinct learning rates. Furthermore, when we sample a node, we observe all links of a sampled node in the training set. We can therefore update the community memberships of the sampled node using the complete natural gradients in Eq. **S17**.

Because many networks are sparse, including the real datasets and the synthetic networks analyzed in the article, the link sampling algorithm scales to such networks even when the minibatch in each iteration is the set of all links in the training set. In this case, the natural gradients in Eq. **S17** and Eq. **S18** are used in the global step.

In our study on synthetic networks, we set the minibatch to the entire set of links and used a learning rate of 1. This leads to good convergence of the variational objective (Fig. S1). Furthermore, we rescaled γ during an initial phase as follows:

$$\gamma_{a,k} = \gamma_{a,k} * \frac{\sum_{(i,j)\in links} 1}{\sum_{(i,j)\in links}\phi_{ij}^{kk}}. \quad \text{[S20]}$$

The rescaling of γ in Eq. **S20** ensures that each community makes an equal contribution to the observations. This avoids small communities with high community strengths and unused communities during the early iterations. The initial phase is run until the expected log likelihood on a held-out set of node pairs no

longer improves. At this point, the inference continues without the scaling in Eq. **S20** until the algorithm converges. (This can be interpreted as a form of annealing, a technique that is sometimes used in variational inference.)

As we demonstrate in the empirical study on synthetic networks, the SVI algorithm with link sampling recovers true communities with high accuracy, and scales to networks with millions of nodes.

Further subsampling can be applied to improve the efficiency of the SVI algorithm with link sampling. For instance, we can apply informative set sampling to the links. We maintain two dynamic sets of links: links whose corresponding interaction parameters have "converged" and links that have not converged. Each iteration, we sample links with a bias toward links that have not converged. (See Eq. **S13**.)

Setting the Number of Communities and Initializing Parameters. SVI requires initial settings of the global variational parameters. There are many ways to initialize these parameters. We set the community strength parameters λ from "false observations" by dividing the links and nodes equally among the communities and adding a small random offset drawn from a Gamma distribution with mean 1. We initialize the community memberships $\boldsymbol{\gamma}$ randomly in our empirical study on real and synthetic networks. Alternatively, we can initialize the γ using an "initialization algorithm" that we describe below.

The initialization algorithm provides a decomposition of the network into overlapping communities that can be used to initialize the community memberships $\boldsymbol{\gamma}$ and set the number of communities in the SVI algorithm. These communities are a good start, but it significantly improves as we run the SVI algorithm.

The pseudocode of our fast, scalable initialization algorithm for estimating the number of communities K is enumerated below.

1. Initialize variational parameters of MMSB model M.
   - M has N nodes and N communities.
   - Initialize $\boldsymbol{\gamma} = (\gamma_n)_{n=1}^{N}$ randomly.
   - Assign each node n to its own community n by adding a small positive weight to $\gamma_{n,n}$.
   - Keep only the top r communities of each node.
2. For each link (a, b) in the training set,
   - Let t_a and t_b be the top communities of nodes a and b.
   - Set $\gamma_{a,t_b} \leftarrow \gamma_{a,t_b} + 1$; $\gamma_{b,t_a} \leftarrow \gamma_{b,t_a} + 1$.
3. Recompute the top r communities of each node.
4. Repeat steps 2, 3 for $\log N$ iterations.
5. For each link (a, b) in the training set,
   - Assign nodes a and b to community k if the approximate posterior probability $p(z_{a\to b} = z_{a\leftarrow b} = k|\mathbf{y}, M) > 0.5$.
6. Return the overlapping communities and their cardinality.

The initialization algorithm approximates a batch variational inference algorithm for a subclass of the MMSB where the community strengths $\boldsymbol{\beta}$ are set to 1. This algorithm is fast: It completes in minutes on networks with millions of nodes and thousands of communities. In simulations, the algorithm frequently finds the number of communities reasonably close to the ground truth number. (See the empirical study on synthetic networks.)

The algorithm begins by assigning each node to its own community. It then computes the community memberships γ for all nodes while maintaining only the top r communities with each node. (We set $r = 5$ in all our experiments.)

Under the restricted model, with community strengths set to 1 and nodes initialized to their own community, the local step for a link in Eq. **S10** dictates that the optimal community indicator for node a is the dominant community of node b, and vice versa. This amounts to an exchange of the dominant community memberships of the nodes and is computed in $O(1)$ time by maintaining the peak communities of nodes. Such exchanges bear similarities to the label propagation steps in ref 10.

The algorithm terminates after exactly $\log N$ iterations, where N is the number of nodes. This stopping criteria is reasonable under the assumption of "small-world" behavior, where the average path length in the network grows proportional to $\log N$ (11).

When the initialization algorithm terminates, it writes a list of communities. Each community consists of a list of nodes, and nodes can appear in multiple communities. A node is added to community k if it is adjacent to at least one link whose approximate posterior probability of belonging to community k is greater than a high threshold. We can use the list of communities to initialize the γ for the SVI algorithm. For example, we can initialize the memberships of a node randomly but with a greater weight on the community assignments from the initialization algorithm. The number of communities, i.e., the number of ways in which links are colored, gives us the input K for the SVI algorithm.

We note again that the initialization algorithm provides us with an optional starting point for the SVI algorithm and an estimate of the number of communities in the observed data.

Computational Complexity. The local step of the SVI algorithm can be computed in $O(SK)$ operations, where S is the number of node pairs sampled in each iteration and K is the number of communities. Due to the assortativity assumptions in our model, the local step is not quadratic in K as is typical for the MMSB (2, 3). The time for the global step of the SVI algorithm is $O(NK)$ operations per iteration, where N is the number of nodes. To avoid updating all nodes in the network, we can maintain a distinct learning rate for each node. In a given iteration, we skip updating the community membership parameters and learning rates of nodes not in the minibatch. The sequence of positive step sizes used in updating a node's membership parameter continue to satisfy the Robbins–Monro conditions (7). This improves the time for the global step to O(nK) operations per iteration, where n is the number of nodes in the minibatch.

In the SVI algorithm with link sampling, with the minibatch set to all links, the computational complexity is $O(\mathrm{MK} + \mathrm{NK})$ operations per iteration, where M is the number of links. The SVI algorithm with link sampling does not require subsampling nonlinks and converges much faster than other subsampling methods.

Open-Source Software. We implemented the SVI algorithm and the various subsampling variants in C++. (Our software is available at https://github.com/premgopalan/svinet.) The software takes as input a text file of undirected links, the number of nodes, the type of subsampling method, and optionally, the hyperparameter values and the number of communities. The software generates as output the list of discovered overlapping communities, the fitted model, the computed log likelihood on various held-out sets, and Graph Modeling Language (GML) format files for visualizing the communities.

Empirical Study on Real-World Networks

In this section, we describe the details of the empirical study on real-world networks. We ran the SVI algorithm with informative set sampling on the real networks in Table S2, The input to the SVI algorithm is a list of links and the number of communities. We preprocessed the networks to associate each node with "informative" and "noninformative" sets of node pairs.

Assessing Convergence on the Training Set. We measure convergence of the SVI algorithm by computing the link prediction accuracy on a held-out set of node pairs. In our experiments on real networks, we set aside two validation sets and a test set, each

having $h\%$ of the network links and an equal number of nonlinks. In the experiments on real networks, we set $h = 5\%$. The links and nonlinks are chosen from the network uniformly at random. We use the validation sets to assess convergence, choose learning parameters, and study the sensitivity to the number of communities.

A "50%-links" validation set poorly represents the severe class imbalance between links and nonlinks in real-world networks. For example, links form only 0.0039% of the node pairs in the arXiv network (12) listed in Table S2. On the other hand, a validation set matching the network sparsity would have too few links. We address the class imbalance by computing the "validation log likelihood at network sparsity". This quantity is computed by reweighting the average link and nonlink log likelihood (estimated from the 50% links validation set) by their respective proportions in the network.

We stop training when the average change in expected validation log likelihood at network sparsity is less than 0.001% or if the expected validation log likelihood at network sparsity no longer increases.

Under the MMSB, we approximate the predictive distribution using point estimates of the posterior community memberships of nodes $\hat{\boldsymbol{\theta}}$ and the posterior community strengths $\hat{\boldsymbol{\beta}}$; these point estimates are computed as the mean of the variational posterior parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\lambda}$, respectively. The estimated predictive distribution of a held-out node pair y_{ab} is as follows:

$$p(y_{ab}|y_{\text{observed}}) \approx \sum_{z_{a\to b}} \sum_{z_{a\leftarrow b}} p\left(y_{ab}|z_{a\to b}, z_{a\leftarrow b}, \hat{\boldsymbol{\beta}}\right) p\left(z_{a\to b}|\hat{\theta}_a\right) p\left(z_{a\leftarrow b}|\hat{\theta}_b\right). \quad \textbf{[S21]}$$

It is straightforward to show that Eq. **S21** is a valid approximation. We then evaluate the log probability of the node pairs in the held-out set under this distribution. Fig. S2 shows the convergence of the "perplexity" at network sparsity on a validation set. Results are shown for the arXiv network (12) and the Google network (13). Perplexity is defined using the average predictive log likelihood of a held-out set of node pairs H,

$$\text{perplexity}(H) = \exp\left\{-\frac{\sum_{a,b\in H} \log p(y_{ab}|y_{\text{observed}})}{|H|}\right\}. \quad \textbf{[S22]}$$

Perplexity is a measure of model fitness (lower numbers are better). Fig. S2 shows that the SVI algorithm with informative set sampling can approximate the posterior distribution on large networks in several hours. Notice that the perplexity values are small in magnitude. This is because we compute the validation log likelihood at network sparsity. The model can predict a large fraction of the nonlinks with high accuracy, and is not "surprised" by them.

Model Selection. As with many probabilistic models of community detection, the MMSB requires setting the number of communities. In our empirical study, we addressed this model selection problem in two ways. One was by evaluating the predictive performance of the model for varying numbers of communities. We held out a set of node pairs and computed the average predictive log likelihood, as described above. A better model will assign a higher probability to the held-out set. This reflects a predictive approach to model selection, and has good statistical properties (14). (We note that nonprobabilistic methods for detecting overlapping communities usually cannot provide a mechanism for predicting unseen pieces of the network.) Fig. S3 shows the sensitivity of the MMSB to the number of communities on the arXiv network (12) and the Google network (13). A second way was to set the number of communities to the estimate from our initialization algorithm. We used this second approach in our empirical study on synthetic networks.

Comparison with the Stochastic Blockmodel. We compared the model fitness of the MMSB to the stochastic blockmodel (15) on real-world networks. The stochastic block model attaches a single community to each node. We consider the following constrained stochastic blockmodel, with K communities, in a full Bayesian setting (16, 17). The modeling assumptions are captured in the following generative process:

1. For each community k,
   (a) Draw intracommunity strengths $\beta_k \sim \text{Beta}(\eta)$.
2. Draw the intercommunity strength $\beta' \sim \text{Beta}(\eta')$
3. Draw the node memberships $\theta \sim \text{Dirichlet}(\alpha)$
4. For each node i:
   (a) Draw a community indicator $z_i \sim \theta$
5. For each pair of nodes i and j, where $i < j$:
   (a) Draw the connection between them from

$$p\left(y_{ij} = 1|z_i, z_j, \boldsymbol{\beta}\right) = \begin{cases} \beta_{z_{ij}} & \text{if} \quad z_i = z_j \\ \beta' & \text{if} \quad z_i \neq z_j \end{cases}. \quad \textbf{[S23]}$$

Unlike the MMSB of ref. 3, the single-membership model of Eq. **S23** cannot explain all links as arising from shared memberships; hence, it must learn the intercommunity strength β' from the data. Our model is a generalization of ref 16.

We derived a scalable SVI algorithm for the model in Eq. **S23** by treating all hidden variables, including the community indicators, as global. This is necessary because the community indicators associated with each node are not local to an observation. Therefore, the variational parameters, including those for the community indicators, were updated using noisy natural gradients in the global step. We note that the subsampling methods discussed earlier, with the exception of link sampling, apply to the single-membership model.

In Fig. S3, we compared the predictive performance of the MMSB to the stochastic blockmodel on the arXiv network and the Google network. We fit both models using the SVI algorithm with informative set sampling. Fig. S3 shows that the mixed-membership model demonstrates better predictive performance than the analogous single-membership model of Eq. **S23** over a range of the number of communities.

Hyperparameters and Learning Parameters. SVI requires setting hyperparameters of the model and learning rates of the algorithm. We set the node membership forgetting rate (κ) to 0.5 and the community strength forgetting rate to 0.9. We set the delay $\tau_0 = 1024$. We set Dirichlet hyperparameters $\alpha = \frac{1}{K}$, where K is the number of communities. On real networks, the prior on the community strengths was set using a uniform assignment of links and nodes to communities. We set the probability of a link when nodes assume different communities, ε to a low value of 10^{-30}. This reflects our assortativity assumption that links arise from similarity in communities between a pair of nodes.

Empirical Study on Synthetic Networks

The goal of the study on synthetic networks is to assess the accuracy of the SVI algorithm and compare with other scalable methods in the research literature. We want the synthetic networks to match the properties of real networks—skewed community and node degree distributions (18), significant community overlap (19, 20), and a large fraction of nodes with multiple memberships (20).

We ran experiments to evaluate the performance of the algorithms on benchmark networks with and without "noisy" links. Notice that our significant community overlap requirement naturally avoids well-separated clusters. The inclusion of noisy links tests the algorithm's ability to identify overlapping com-

munities even when a significant fraction of a node's links are to nodes sharing no communities.

For the experiments on networks without noise, we generated 20 Lancichinetti–Fortunato–Radicchi (LFR) benchmark networks (21) varying in size from $N = 1000$ to $N = 1{,}000{,}000$ nodes. One-half of the nodes in each network have memberships in $m = 4$ communities. We set the average degree of nodes as $15 \times m$, similar to the experiments in ref. 22. The LFR benchmarks give the node degrees and community sizes power laws; the degree distribution and community size distribution exponents were set to the default values of 2.0 and 1.0, respectively. Research on scale-free networks (23) have assumed the maximum degree to vary as $k_{\max} \sim N^{\frac{1}{\alpha}}$, where α is the power law exponent for node degrees. We set $k_{\max} = \sqrt{N}$. We varied the minimum and maximum community sizes as $\left(20\frac{N}{1000}, 50\frac{N}{1000}\right)$. However, because community sizes are typically small (13), we set the minimum and maximum community sizes when $N = 1{,}000{,}000$ nodes to (2000, 5000). These settings result in about ~750 ground truth communities when $N = 1{,}000{,}000$ and ~ 30 communities when $N = 1{,}000$. Finally, we set the "mixing parameter" μ (21) to 0 in our experiments on networks without noise. The mixing parameter is the fraction of a node's links that connect to nodes sharing no communities.

On each network, we ran the following algorithms:

1. The COPRA label propagation algorithm (10).
2. The INFOMAP algorithm based on flow compression (24).
3. The MOSES seed expansion algorithm (22).
4. The Poisson expectation-maximization (EM) algorithm (the Poisson community model, fit with EM) (25).
5. The OSLOM algorithm for finding statistically significant communities (26).
6. The Clique percolation algorithm (19).
7. The Link clustering algorithm (20).

For the experiments on networks with noisy links, we varied μ in steps of 0.2 from 0 to 0.8. We fixed the number of nodes at 10,000, and kept the other settings the same as the preceding experiment. We generated 25 LFR benchmark networks and included only the candidate algorithms that successfully scaled to 1,000,000 nodes in the preceding experiment.

We used the author's source code for all algorithms. We ran the SVI algorithm with the link sampling method. For each run, we measured the normalized mutual information (NMI) (21) between the inferred community structure and the true community structure. For the algorithms that find communities at various resolutions—Clique percolation, Link clustering, and COPRA—we varied the parameters as described below, and kept the best NMI score.

For the SVI and the Poisson EM algorithm, we ran the algorithms until convergence on networks with up to 100,000 nodes. We measured convergence of the SVI algorithm with link sampling using the average change in average validation log likelihood at network sparsity, as we did with the experiments on real networks. However, since our goal is to assess the accuracy in recovering ground truth communities, we set aside only a single validation set of node pairs, having 1% of network links and an equal number of nonlinks. We gave all algorithms, except the SVI algorithm, the complete synthetic network as input.

On the million node networks, the SVI and the Poisson EM algorithm can take a long time for convergence in likelihood, whereas their NMI scores have typically "converged" quickly. One reason for this is the large number of links (~54 million links) in these synthetic networks. We instrumented the author's source code for the Poisson EM algorithm and the SVI algorithm to periodically report the accuracy scores when provided with ground truth communities. We gave both algorithms a computational budget of 24 h and recorded the NMI scores attained by them. The Poisson EM algorithm's NMI score had typically "converged" at this point, even if the likelihood did not. (We note that in other applications of EM, such as probabilistic latent semantic indexing, "early stopping" is an effective form of regularization.)

Table S3 shows the NMI results on the networks without noise. Some algorithms could not scale to one million node networks. The four that did were the Poisson EM, the SVI, the COPRA, and the INFOMAP algorithms. The SVI algorithm performs better than the COPRA and the INFOMAP algorithms and is as accurate as the Poisson EM algorithm on the one million node network. On smaller networks, the SVI algorithm performs as well as the Poisson EM algorithm; it performs second to Clique percolation on the one thousand node networks. However, the Clique percolation algorithm does not scale beyond the 10,000 node networks.

Fig. S4 shows the NMI results on the networks with noisy links. We find that the SVI algorithm performs better than two of the three other scalable alternatives—the COPRA and the INFOMAP algorithms—and is as accurate as the Poisson EM algorithm.

Hyperparameters and Learning Parameters. We set the number of communities K of the SVI algorithm with link sampling to the value chosen by the initialization algorithm. We provided the same K to the Poisson EM algorithm (25). We set the minibatch for the link sampling method to set of all training links, and set the learning rate to 1. Other hyperparameters of the SVI algorithm were set to the same values as our experiments on real networks.

Algorithm Settings for the LFR Experiments. The Clique percolation algorithm identifies communities from a series of adjacent k-cliques (19). In our experiments, we varied k from 4 to 7, a typical range for LFR experiments (10, 22). The Link clustering algorithm defines a similarity function over nodes sharing a link, and uses hierarchical clustering to find hierarchical community structures (20). Because the dendrogram can be partitioned in multiple ways, the algorithm uses a measure of the quality of a link partition, called the partition density D. We varied D from 0.1 to 0.4—the range we found to be best—in steps of 0.1. The COPRA algorithm is a fast heuristic based on label propagation and includes a overlap parameter that we varied from 2 to 10, in steps of 2. This is a typical range (10). The OSLOM (26), MOSES (22), and the INFOMAP (24) algorithms were run with parameters set to default values.

The author's software for most of the algorithms we compare with generate "community assignments," the discovered mapping between nodes and communities. The mapping is used to compute the accuracy when given the ground truth communities.

We extended both the SVI and the Poisson EM algorithm to generate the community assignment files. In both algorithms, we assigned a link to a community if the approximate posterior probability of link assignment to a community exceeded a threshold t. We took the best NMI values obtained from thresholds $t = 0.5$ and $t = 0.9$. For the experiments on networks without noise, we assigned each node associated with a link to the same community as the link. For the experiments with noisy links, we required at least three links of a node to be assigned to a community before assigning the node to that community. We added this setting to both algorithms to control sensitivity to noise. [Notice in Fig. S4 that both algorithms continue to show a high accuracy on networks without noise $(\mu = 0)$ with the threshold set to three links.]

1. Hoffman M, Blei DM, Wang C, Paisley J (2013) Stochastic variational inference. *J Mach Learn Res* 14:1303–1307.
2. Airoldi EM, Blei DM, Fienberg SE, Xing EP (2008) Mixed membership stochastic blockmodels. *J Mach Learn Res* 9:1981–2014.
3. Gopalan P, Mimno D, Gerrish SM, Freedman MJ, Blei DM (2012) Scalable inference of overlapping communities. *Adv Neural Inf Process Syst* 25.
4. Kullback S, Leibler RA (1951) On information and sufficiency. *Ann Math Stat* 22:79–86.
5. Wainwright MJ, Jordan MI (2007) Graphical models, exponential families, and variational inference. *Found Trends Mach Learn* 1:1–305.
6. Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK (1999) An introduction to variational methods for graphical models. *Mach Learn* 37:183–233.
7. Robbins H, Monro S (1951) A stochastic approximation method. *Ann Math Stat* 22: 400–407.
8. Amari S (1982) Differential geometry of curved exponential families–curvatures and information loss. *Ann Stat* 10:357–385.
9. Raftery AE, Niu X, Hoff PD, Yeung KY (2012) Fast inference for the latent space network model using a case-control approximate likelihood. *Journal of Computational and Graphical Statistics* 21:909–919.
10. Gregory S (2010) Finding overlapping communities in networks by label propagation. *New J Phys* 12:103018.
11. Newman MEJ (2003) The structure and function of complex networks. *SIAM Rev Soc Ind Appl Math* 45:167–256.
12. Ginsparg P (2011) ArXiv at 20. *Nature* 476(7359):145–147.
13. Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2009) Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Math* 6:29–123.
14. Geisser S, Eddy WF (2011) A predictive approach to model selection. *J Am Stat Assoc* 74:153–160.
15. Nowicki K, Snijders TAB (2001) Estimation and prediction for stochastic blockstructures. *J Am Stat Assoc* 96:1077–1087.
16. Hofman JM, Wiggins CH (2008) Bayesian approach to network modularity. *Phys Rev Lett* 100(25):258701.
17. Latouche P, Birmel E, Ambroise C (2012) Variational Bayesian inference and complexity control for stochastic block models. *Stat Model* 12(1):93–115.
18. Karrer B, Newman MEJ (2011) Stochastic blockmodels and community structure in networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 83(1 Pt 2):016107.
19. Derényi I, Palla G, Vicsek T (2005) Clique percolation in random networks. *Phys Rev Lett* 94(16):160202.
20. Ahn YY, Bagrow JP, Lehmann S (2010) Link communities reveal multiscale complexity in networks. *Nature* 466(7307):761–764.
21. Lancichinetti A, Fortunato S (2009) Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys Rev E Stat Nonlin Soft Matter Phys* 80(1 Pt 2):016118.
22. McDaid AF, Hurley NJ (2010) Detecting highly overlapping communities with model-based overlapping seed expansion. (*Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*) (IEEE Computer Society, Washington, DC), pp 112–119.
23. Aiello W, Chung F, Lu L (2000) A random graph model for massive graphs. *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing* (ACM, New York), pp 171–180.
24. Esquivel AV, Rosvall M (2011) Compression of flow can reveal overlapping-module organization in networks. *Phys Rev X* 1:021025.
25. Ball B, Karrer B, Newman MEJ (2011) Efficient and principled method for detecting communities in networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 84(3 Pt 2):036103.
26. Lancichinetti A, Radicchi F, Ramasco JJ, Fortunato S (2011) Finding statistically significant communities in networks. *PLoS One* 6(4):e18961.

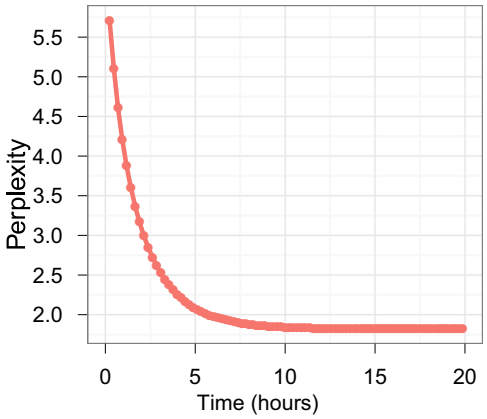

Fig. S1. The convergence of held-out perplexity on an LFR synthetic network (1) with 1,000,000 nodes and ~54,000,000 links. The SVI algorithm was run with link sampling. The network has 750 ground truth overlapping communities, and 50% of the nodes have memberships in four communities. The degree distribution and the community size distribution follow power laws.

1. Lancichinetti A, Fortunato S (2009) Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys Rev E Stat Nonlin Soft Matter Phys* 80(1 Pt 2):016118.

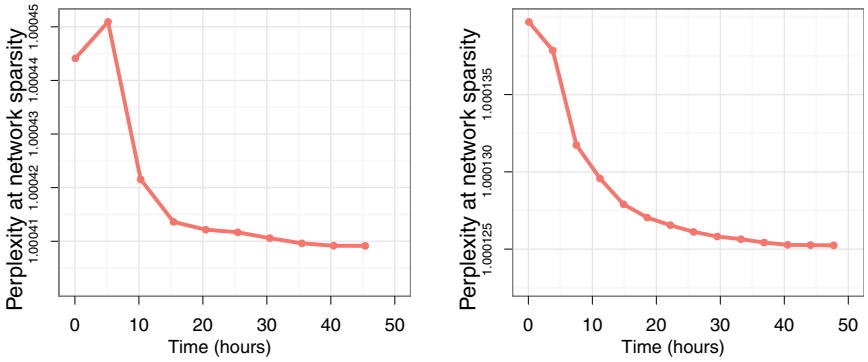

Fig. S2. The convergence of the held-out perplexity at network sparsity on the arXiv network (1) (*Left*) and the Google network (2) (*Right*). The SVI algorithm was run with informative set sampling. The number of communities K was set to 1,000.

1. Ginsparg P (2011) ArXiv at 20. *Nature* 476(7359):145–147.
2. Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2009) Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Math* 6:29–123.

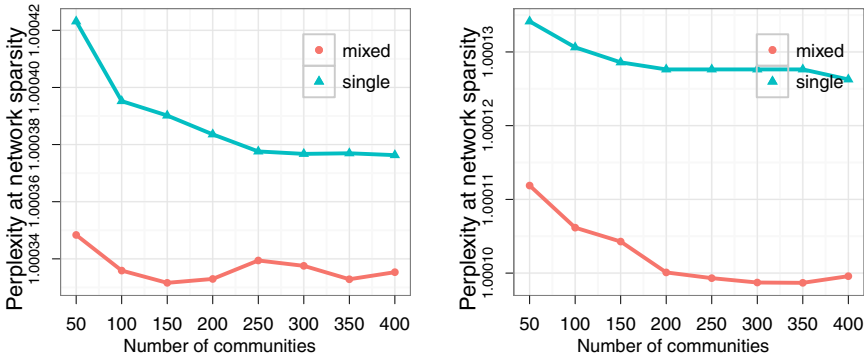

Fig. S3. Perplexity results on the arXiv network (1) (*Left*) and the Google network (2) (*Right*). The mixed-membership model outperforms the single-membership model. We fit both models using the SVI algorithm with informative set sampling.

1. Ginsparg P (2011) ArXiv at 20. *Nature* 476(7359):145–147.
2. Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2009) Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Math* 6: 29–123.

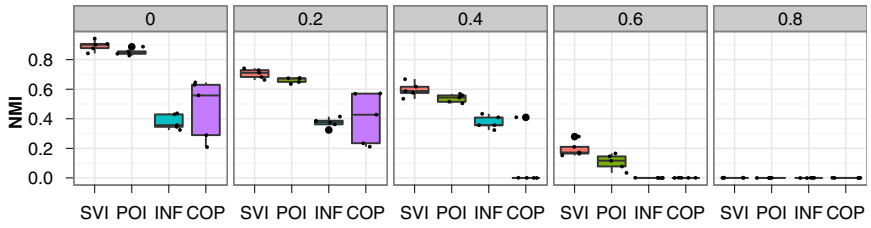

Fig. S4. The stochastic inference algorithm (SVI) with link sampling outperforms COPRA (COP) (1) and INFOMAP (INF) (2) and is as accurate as the Poisson EM algorithm (POI) (3) in discovering ground truth communities in 25 LFR benchmark networks with noisy links. Each panel shows the performance of the algorithms on five replications of the random network generated with 10,000 nodes and a fixed mixing parameter (4). The mixing parameter is the fraction of a node's links that connect to nodes sharing no communities. From *Left* to *Right*, the panels correspond to increasing noise.

1. Gregory S (2010) Finding overlapping communities in networks by label propagation. *New J Phys* 12:103018.
2. Esquivel AV, Rosvall M (2011) Compression of flow can reveal overlapping-module organization in networks. *Phys Rev X* 1:021025.
3. Ball B, Karrer B, Newman MEJ (2011) Efficient and principled method for detecting communities in networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 84(3 Pt 2):036103.
4. Lancichinetti A, Radicchi F, Ramasco JJ, Fortunato S (2011) Finding statistically significant communities in networks. *PLoS One* 6(4):e18961.

Table S1. Top 10 articles in the arXiv network (1) by estimated bridgeness (2)

Title	No. citations	Estimated bridgeness
Maps of dust infrared emission for use in estimation of reddening and cosmic microwave background radiation foregrounds (3)	5,946	2,893.7
First-year Wilkinson microwave anisotropy probe (WMAP) observations: Determination of cosmological parameters (4)	5,707	2,270.7
Three-year Wilkinson microwave anisotropy probe (WMAP) observations: Implications for cosmology (5)	4,488	1,907.1
Big bang nucleosynthesis (6)	2,896	1,882.9
The cosmological parameters 2006 (7)	2,472	1,703.9
Five-year Wilkinson microwave anisotropy probe (WMAP) observations: Cosmological interpretation (8)	2,804	1,485.9
A large mass hierarchy from a small extra dimension (9)	3,644	1,426.7
The large *N* limit of superconformal field theories and supergravity (10)	3,914	1,378.4
An alternative to compactification (11)	2,803	1,275.8

Notice that some articles have higher bridgeness but a smaller citation count than others.

1. Ginsparg P (2011) ArXiv at 20. *Nature* 476(7359):145–147.
2. Nepusz T, Petróczi A, Négyessy L, Bazsó F (2008) Fuzzy communities and the concept of bridgeness in complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 77(1 Pt 2)(1):016107.
3. Schlegel DJ, Finkbeiner DP, Davis M (1998) Maps of dust infrared emission for use in estimation of reddening and cosmic microwave background radiation foregrounds. *Astrophys J* 500:525–553.
4. Spergel DN, et al. (2003) First-year Wilkinson microwave anisotropy probe (WMAP) observations: Determination of cosmological parameters. *The Astrophysical Journal Supplement Series* 148(1). arXiv:astro-ph/0302209.
5. Spergel DN, et al. (2007) Three-year Wilkinson microwave anisotropy probe (WMAP) observations: Implications for cosmology. *The Astrophysical Journal Supplement Series* 170(2). arXiv:astro-ph/0603449.
6. Fields B, Sarkar S (2004) Big bang nucleosynthesis. arXiv:astro-ph/0406663.
7. Yao W-M, et al. (2006) Review of particle physics. *Journal of Physics G: Nuclear and Particle Physics* 33(1). arXiv:astro-ph/0601168.
8. Komatsu E, et al. (2009) Five-year Wilkinson microwave anisotropy probe observations: Cosmological interpretation. *The Astrophysical Journal Supplement Series* 180(2). arXiv:0803.0547.
9. Randall L, Sundrum R (1999) Large mass hierarchy from a small extra dimension. *Phys Rev Lett* 83(17):3370–3373.
10. Maldacena JM (1998) The large *N* limit of superconformal field theories and supergravity. *Advances in Theoretical and Mathematical Physics* 2:231–252.
11. Randall L, Sundrum R (1999) An alternative to compactification. *Phys Rev Lett* 83:4690–4693.

Table S2. Real-world networks analyzed in the article and *SI Text*

Dataset	No. of nodes	No. of links	% links	Type	Source
arXiv	576,000	6,640,000	0.0039%	Citation	Ref. 1
Google	875,000	4,320,000	0.0011%	Hyperlink	Ref. 2
US patents	3,700,000	16,500,000	0.00023%	Citation	Ref. 3

1. Ginsparg P (2011) ArXiv at 20. *Nature* 476(7359):145–147.
2. Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2009) Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Math* 6: 29–123.
3. Leskovec J, Kleinberg J, Faloutsos C (2005) Graphs over time: Densification laws, shrinking diameters and possible explanations. Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM, New York), pp 177–187..

Table S3. Accuracy results on 20 LFR benchmark networks (1) measured using normalized mutual information (1)

Nodes	Replication	SVI	COPRA (2)	INFOMAP (3)	MOSES (4)	POISSON (5)	OSLOM (6)	CLIQUE (7)	LC (8)
1,000	1	0.58	0.55	0.38	0.47	0.62	0.44	0.93	0.15
1,000	2	0.77	0.45	0.36	0.49	0.77	0.41	0.85	0.16
1,000	3	0.66	0.46	0.36	0.53	0.66	0.49	0.96	0.17
1,000	4	0.63	0.17	0.38	0.52	0.62	0.46	0.78	0.15
1,000	5	0.76	0.39	0.35	0.55	0.75	0.48	0.85	0.20
10,000	1	0.90	0.28	0.35	0.56	0.85	0.18	0.22	0.01
10,000	2	0.90	0.28	0.32	0.55	0.88	0.16	0.13	0.01
10,000	3	0.82	0.07	0.36	0.54	0.78	0.19	0.23	0.01
10,000	4	0.86	0.61	0.44	0.54	0.82	0.17	—	0.02
10,000	5	0.89	0.62	0.40	0.56	0.86	0.17	—	0.00
100,000	1	0.82	0.57	0.34	0.35	0.85	—	—	—
100,000	2	0.83	0.44	0.33	0.33	0.81	—	—	—
100,000	3	0.81	0.50	0.35	0.34	0.81	—	—	—
100,000	4	0.82	0.43	0.33	0.35	0.84	—	—	—
100,000	5	0.83	0.58	0.33	0.35	0.84	—	—	—
1,000,000	1	0.76	0.52	0.22	—	0.76	—	—	—
1,000,000	2	0.77	0.50	0.16	—	0.76	—	—	—
1,000,000	3	0.78	0.53	0.17	—	0.77	—	—	—
1,000,000	4	0.76	0.49	0.23	—	0.79	—	—	—
1,000,000	5	0.77	0.51	0.14	—	0.77	—	—	—

The networks were generated with mixing parameter set to 0. The four algorithms that scale to a million nodes are the SVI algorithm, the Poisson EM algorithm (5), INFOMAP (3), and COPRA (2). The SVI algorithm performs better than INFOMAP and COPRA and is as accurate as the Poisson EM algorithm.

1. Lancichinetti A, Fortunato S (2009) Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys Rev E Stat Nonlin Soft Matter Phys* 80(1 Pt 2):016118.
2. Gregory S (2010) Finding overlapping communities in networks by label propagation. *New J Phys* 12:103018.
3. Esquivel AV, Rosvall M (2011) Compression of flow can reveal overlapping-module organization in networks. *Phys Rev X* 1:021025.
4. McDaid AF, Hurley NJ (2010) Detecting highly overlapping communities with model-based overlapping seed expansion. (*Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*) (IEEE Computer Society, Washington, DC), pp 112–119.
5. Ball B, Karrer B, Newman MEJ (2011) Efficient and principled method for detecting communities in networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 84(3 Pt 2):036103.
6. Lancichinetti A, Radicchi F, Ramasco JJ, Fortunato S (2011) Finding statistically significant communities in networks. *PLoS One* 6(4):e18961.
7. Derényi I, Palla G, Vicsek T (2005) Clique percolation in random networks. *Phys Rev Lett* 94(16):160202.
8. Ahn YY, Bagrow JP, Lehmann S (2010) Link communities reveal multiscale complexity in networks. *Nature* 466(7307):761–764.