Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ucgs20

Link Prediction for Egocentrically Sampled Networks

Tianxi Li, Yun-Jhong Wu, Elizaveta Levina & Ji Zhu

To cite this article: Tianxi Li, Yun-Jhong Wu, Elizaveta Levina & Ji Zhu (2023): Link Prediction for Egocentrically Sampled Networks, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2022.2163648

To link to this article: https://doi.org/10.1080/10618600.2022.2163648

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2023, VOL. 00, NO. 0, 1–24
https://doi.org/10.1080/10618600.2022.2163648

Taylor & Francis
Taylor & Francis Group

Link Prediction for Egocentrically Sampled Networks

Tianxi Li[a], Yun-Jhong Wu[b], Elizaveta Levina[c], and Ji Zhu[c]

[a]Statistics, University of Virginia, Charllottesville, VA; [b]Department of Statistics, University of Michigan, Ann Arbor, MI; [c]University of Michigan, Ann Arbor, MI

ABSTRACT
Link prediction in networks is typically accomplished by estimating or ranking the probabilities of edges for all pairs of nodes. In practice, especially for social networks, the data are often collected by egocentric sampling, which means selecting a subset of nodes and recording all of their edges. This sampling mechanism requires different prediction tools than the typical assumption of links missing at random. We propose a new computationally efficient link prediction algorithm for egocentrically sampled networks, estimating the underlying probability matrix by estimating its row space. We empirically evaluate the method on several synthetic and real-world networks and show that it provides accurate predictions for network links. Supplemental materials including the code for experiments are available online.

ARTICLE HISTORY
Received July 2020
Accepted December 2022

KEYWORDS
Binary data analysis; Machine learning; Network modeling

1. Introduction

Networks are useful for representing connections or relations between individual units, and a large body of work spread over several disciplines, including statistics, has been devoted to network analysis. In many network-related problems, especially in social sciences, the network structure is recorded with noise and missing values (Newman 2018). These problems can be especially severe when the data are collected by a survey, which is popular in social studies. Link prediction addresses this problem by denoising the observed network and/or predicting missing links. Many techniques have been developed for this task; see Liben-Nowell and Kleinberg (2007) and Lü and Zhou (2011) for reviews.

The scenario we focus on in this article is when the network structure is constructed by egocentric sampling, a procedure where a subset of nodes is sampled first, and all links involving this subset of nodes are then recorded, while all the other information is missing. Egocentric sampling can be used in surveys and other data collection procedures under budget constraints and accessibility of subjects (Laumann et al. 1995; Hurlbert, Beggs, and Haines 2005; Morris et al. 2009; Banerjee et al. 2013). Link prediction in this context can be directly applied to scientifically meaningful problems, for example, to identify social ties for different levels of socio-economic factors in the post-disaster policy-making (Hurlbert, Beggs, and Haines 2005), or to identify high-risk connections between terrorists (Anil et al. 2015). Even when the network structure itself is not of primary interest, accurately predicting the links may help downstream analysis of the data. For example, Chandrasekhar and Lewis (2011) have shown that in regression models of individuals on an egocentrically sampled network, it is crucial to identify the true connections between individuals to avoid serious bias in covariate effects estimation.

Sampled or partially observed relational data have been studied, and multiple methods have been proposed to handle independently missing links for network estimation (Butts 2003; Newman 2018) and longitudinal studies (Manresa 2013). In particular, Butts (2003) proposes a Bayesian model with individual missing status for every dyad under the ignorable missing mechanism as defined in Rubin (1976), which can be viewed as a more systematic model-based approach to the informant inaccuracy problem discussed in Romney, Weller, and Batchelder (1986) and Batchelder and Romney (1988). Newman (2018) improves this framework with more flexible parameterization. However, the missing status is assumed to be independent at the dyad level and does not adapt to egocentric sampling. Moreover, these methods involve heavy computation and are not feasible for large networks.

Methods for egocentric sampling have also been studied in quantitative social sciences (Freeman 1982; Almquist 2012) and more recently in physics, computer science and statistics (e.g., Newman 2003; Mcauley and Leskovec 2012; Handcock and Gile 2010; Krivitsky and Morris 2017). In particular, Handcock and Gile (2010) introduce a general framework for handling a class of missing mechanisms, including egocentric sampling, in the exponential random graph model (ERGM) framework. The model can be fitted with a pseudo-likelihood method and scales up to moderate-sized networks. More recently, Krivitsky and Morris (2017) extend the ERGM to model egocentric networks in which even though links are recorded, one cannot identify who are the corresponding nodes. An efficient model fitting method is proposed. However, in this situation, the link

CONTACT Tianxi Li ✉ tianxili@virginia.edu Statistics, University of Virginia, 148 Amphitheater Way, Charllottesville, VA 22904.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

prediction problem is not well-defined, so their method is not applicable in our settings. In general, though the ERGM framework admits many attractive statistical properties, the model is known to suffer from multi-modal likelihood and degenerate model space (Chatterjee and Diaconis 2013; Shalizi and Rinaldo 2013), significantly limiting its flexibility to predict links. In economics, Chandrasekhar and Lewis (2011) introduce the egocentric sampling in a similar form, but the network recovery only works on the unrealistically simple configuration model (Chung and Lu 2002) or relies on additional attributes as side information.

In this article, we propose a computationally efficient algorithm for link prediction specifically designed for egocentrically sampled network data. The method is motivated by a matrix completion algorithm, with added subspace estimation. Empirically, we observed competitive performance compared with a wide range of benchmark methods for link prediction on both synthetic networks from different models and on real-world networks.

We start by setting up the probabilistic framework for our discussion. An undirected network on N nodes can be represented with a symmetric adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$. The ijth entry of $\mathbf{A}$, denoted by A_{ij}, is the edge indicator, such that $A_{ij} = A_{ji} = 1$ if and only if there is an edge between nodes i and j. There are many successful statistical models for random networks (Frank and Strauss 1986; Robins et al. 2007; Crane and Dempsey 2018; Lauritzen, Rinaldo, and Sadeghi 2018). Here we follow the popular inhomogeneous Erdős-Rényi framework for random networks: given an underlying probability matrix $\mathbf{P}$, all the upper triangular entries of the adjacency matrix A are generated as independent Bernoulli random variables, with probability of $A_{ij} = 1$ given by the ijth entry of $\mathbf{P}$, p_{ij}. We treat the probability matrix $\mathbf{P}$ as fixed throughout the article.

1.1. Matrix Completion and Link Prediction

The link prediction problem can be viewed as the task of classifying pairs of nodes into two categories, "linked" and "not linked." In practice, link prediction is frequently made on the basis of a score for each pair (Liben-Nowell and Kleinberg 2007; Lichtenwalter, Lussier, and Chawla 2010; Zhao et al. 2017). Such a score function can be an estimator of $\mathbf{P}$ or, alternatively, a monotone function of the link probabilities if only a relative ranking of links is important. This is closely related to the problem of matrix completion.

Matrix completion is the task of completing a data matrix where only some entries of the matrix A are observed, often solved through a low-rank approximation. We assume that $A_{ij} = p_{ij} + e_{ij}$, where e_{ij}'s are independent random errors with $\mathbb{E}[e_{ij}] = 0$ and $\mathbf{P}$ is the true signal matrix that is assumed to be at least approximately low-rank. When A is constrained to be symmetric, e_{ij}'s are only independent across upper triangular entries and the symmetric restriction is applied to the lower triangular entries.

A basic matrix completion procedure solves the optimization problem

$$\begin{aligned} &\underset{\mathbf{X}}{\text{minimize}} && \ell(\Omega(\mathbf{X}), \Omega(\mathbf{A})) \\ &\text{subject to} && \text{Rank}(\mathbf{X}) \le r, \end{aligned}$$

where Ω is an entry-wise mask operator with $\Omega(A_{ij}) = A_{ij}$ if the entry A_{ij} is observed and $\Omega(A_{ij}) = 0$ otherwise, and ℓ is a loss function. This problem and its variants have been extensively studied in recent years (Candès and Tao 2010; Candès and Plan 2010; Keshavan, Montanari, and Oh 2010; Davenport et al. 2014); we refer readers to the review of Chi and Li (2019). All of this work is under the assumption that entries in the data matrix are missing independently, though the missing probability can be different across entries. Under this assumption, matrix completion methods have been successfully used for many link prediction problems (Pech et al. 2017; Gao et al. 2018; Li, Levina, and Zhu 2020b). The models of Butts (2003) and Newman (2018) can also be viewed as probabilistic binary matrix completion methods.

1.2. Egocentric Networks

The sampling mechanism we focus on in this article, egocentric sampling, often arises from surveys that ask a sample of subjects to name all of their social connections. We model this process as sampling n nodes without replacement from the true network of size N, and recording all edges involving these n nodes, giving a random *egocentric sample* of n rows (and the corresponding columns, since A is symmetric) from the full $N \times N$ adjacency matrix A.

Formally, suppose that the network $G = (V, E)$ has the node set $V = \{1, \dots, N\}$ and the edge set E with $|E| = m$. We sample nodes $\mathcal{I} = \{i_1, \dots, i_n\} \subset V$, and observe the egocentriclaly sampled network, or *ego-network* for short, $G_{\text{ego}} = (V_{\text{ego}}, E_{\text{ego}})$, where $V_{\text{ego}} = V$, and $E_{\text{ego}} = \cup_{u\in I}\{(u,v) : (u,v) \in E\}$. See Figure 1 for an illustration. Equivalently, when node i is sampled, the ith row and ith column of $\mathbf{A}$ are observed.

Egocentric sampling evidently results in a different missing mechanism from the traditional matrix completion settings discussed in Section 1.1. Therefore, new approaches are needed to account for this difference.

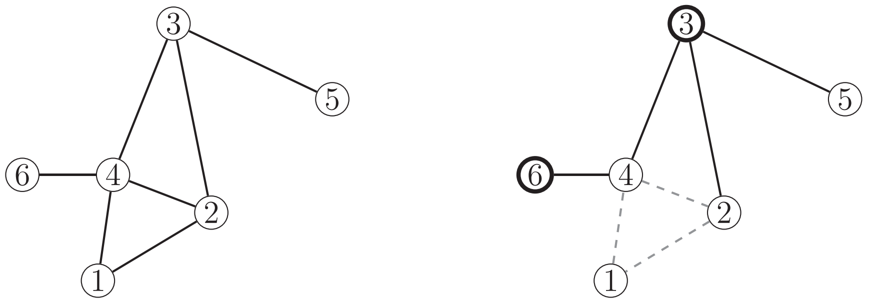
The rest of this article is organized as follows. In Section 2, we propose a new computationally efficient link prediction method for egocentrically sampled networks. The key idea is subspace estimation, since the observed rows allow us to estimate the approximate row space of the probability matrix $\mathbf{P}$. Empirical results on synthetic data and real networks are presented in Section 3 and Section 4. Section 5 concludes the article with a discussion and future work.

2. A Subspace Estimation Algorithm

Without loss of generality, we assume that the first n nodes are sampled, and the observed adjacency matrix can be partitioned into blocks $\mathbf{A}_{ij}$ for $i, j \in \{1, 2\}$, where $\mathbf{A}_{11} \in \{0,1\}^{n\times n}$, $\mathbf{A}_{12} \in \{0,1\}^{n\times(N-n)}$, $\mathbf{A}_{21} = \mathbf{A}_{12}^{\top}$, and the block $\mathbf{A}_{22} \in \{0,1\}^{(N-n)\times(N-n)}$ is unobserved, as illustrated in Figure 2. The corresponding submatrices of $\mathbf{P}$ are similarly labeled $\mathbf{P}_{ij}$ for $i, j = 1, 2$. We also define $\mathbf{A}_{\text{in}} = [\mathbf{A}_{11}\ \mathbf{A}_{12}]_{n\times N}$ with the corresponding probability sub-matrix $\mathbf{P}_{\text{in}} = [\mathbf{P}_{11}\ \mathbf{P}_{12}]_{n\times N}$.

2.1. Motivating Background: The CUR Decomposition

Our goal is to predict links between the nodes that are not sampled, or equivalently to estimate $\mathbf{P}_{22}$ from the available



(a) Full network G

(b) Ego-network G_{ego}, constructed by sampling nodes $\{3, 6\}$; dashed edges are not observed.

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\mathbf{A}_{\text{ego}} = \begin{pmatrix} ? & ? & 0 & ? & ? & 0 \\ ? & ? & 1 & ? & ? & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ ? & ? & 1 & ? & ? & 1 \\ ? & ? & 1 & ? & ? & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

(c) Full adjacency matrix

(d) Observed adjacency matrix

Figure 1. An illustration of egocentric sampling.

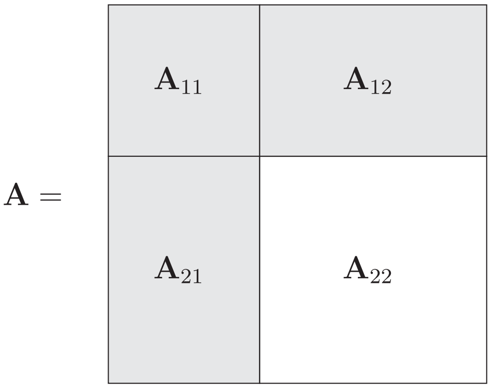


Figure 2. Grey: observed blocks. White: the unobserved block.

information $\mathbf{A}_{\text{in}}$. For illustrative purposes, consider the simpler noiseless setting, where we observe $\mathbf{P}_{\text{in}}$ and aim to estimate $\mathbf{P}_{22}$ or equivalently, the entire matrix $\mathbf{P}$. The egocentric sampling mechanism gives a special structure to missing entries, since we observe full rows (and the corresponding columns) of the matrix. An effective way to recover $\mathbf{P}$ given this structure is by using the so-called CUR decomposition from matrix algebra, studied by Drineas, Kannan, and Mahoney (2006), Drineas, Mahoney, and Muthukrishnan (2008), and Mahoney and Drineas (2009). Mahoney and Drineas (2009) show that any $N \times N$ symmetric low-rank matrix $\mathbf{M}$ can be approximately represented by

$$\mathbf{M} \approx \mathbf{R}^T\mathbf{U}\mathbf{R}$$

where $\mathbf{R}$ is a matrix of q rows of $\mathbf{M}$ and $\mathbf{U}$ is $q \times q$ loading matrix, for some q such that $\text{rank}(\mathbf{M}) < q \ll N$.

Motivated by this representation, we make the assumption

$$\mathbf{P} = \mathbf{P}_{\text{in}}^T\mathbf{U}\mathbf{P}_{\text{in}} \,, \tag{1}$$

where $\mathbf{U}$ is an $n \times n$ matrix.

A natural way to compute $\mathbf{U}$ is to solve the following least square problem, restricted to the observed entries as in other matrix completion problems:

$$\mathbf{U} = \arg\min_{\mathbf{U}} \|\Omega(\mathbf{P}) - \Omega(\mathbf{P}_{\text{in}}^T\mathbf{U}\mathbf{P}_{\text{in}})\|_F^2, \tag{2}$$

where $\Omega(p_{ij}) = I(i \text{ or } j \text{ is sampled}) \cdot p_{ij}$, and $\|\cdot\|_F$ is the matrix Frobenius norm. This problem admits a closed-form solution in the egocentric setting.

Theorem 2.1. Under the egocentric sampling model for Ω and the CUR assumption (1),

$$\mathbf{U} = \mathbf{P}_{11}^{+} \tag{3}$$

is a solution to problem (2), where $\mathbf{P}_{11}^{+}$ is the Moore-Penrose inverse of $\mathbf{P}_{11}$ and the corresponding minimum of the objective function is 0.

Based on Theorem 2.1, one can then use $\mathbf{P}_{\text{in}}^T\mathbf{P}_{11}^{+}\mathbf{P}_{\text{in}}$ as the estimator of $\mathbf{P}$. Such a procedure is not directly applicable in practice when we observe $\mathbf{A}_{\text{in}}$ rather than $\mathbf{P}_{\text{in}}$, but it serves as the motivation of our algorithm.

2.2. Estimation Algorithm for Egocentric Link Prediction

In practice, we observe $\mathbf{A}_{\text{in}}$, a noisy version of $\mathbf{P}_{\text{in}}$. While it is tempting to directly "plug-in" the data by replacing $\mathbf{P}$ and $\mathbf{P}_{\text{in}}$

in (2) and (3) by $\mathbf{A}$ and $\mathbf{A}_{\text{in}}$, this naive approach does not work well in practice, because it overfits the data and gives 0 training error in the objective (2). One way to see this is to observe that $|A_{ij} - p_{ij}|$ is either p_{ij} or $1 - p_{ij}$, so the error is as large as the signal p_{ij} itself. Empirically this estimator often gives poor predictions (see Section 3).

Regularization is a general approach to counteracting overfitting. A natural option in this setting is constraining the rank of the estimator and replying on the low-rank approximation to remove some of the noise. Such a strategy has also been widely used in spectral algorithms in network settings (Rohe, Chatterjee, and Yu 2011; Lei and Rinaldo 2015). Specifically, we compute a regularized estimator of $\mathbf{P}_{\text{in}}$ as the best rank r approximation to $\mathbf{A}_{\text{in}}$, given by the partial SVD (Golub and Van Loan 1989). Denoting this estimator by $\widetilde{\mathbf{P}}_{\text{in}}$, we can then plug it in (2) in the place of $\mathbf{P}_{\text{in}}$ and obtain a regularized estimator of $\mathbf{P}$ from the CUR algorithm. This procedure is summarized as follows.

1. Compute $\widetilde{\mathbf{P}}_{\text{in}} = \mathbf{U}_r\mathbf{D}_r\mathbf{V}_r^{\top}$, the optimal rank-$r$ approximation of $\mathbf{A}_{\text{in}}$ by partial SVD.
2. Let $\widetilde{\mathbf{P}}_{11}$ be the $n \times n$ sub-matrix of $\widetilde{\mathbf{P}}_{\text{in}}$ consisting of the first n columns. Set
$$\widehat{\mathbf{X}} = \frac{1}{2}(\widetilde{\mathbf{P}}_{11}^{+} + \widetilde{\mathbf{P}}_{11}^{\top +}). \tag{4}$$
3. Estimate the probability matrix $\mathbf{P}$ by
$$\widehat{\mathbf{P}} = \widetilde{\mathbf{P}}_{\text{in}}^{\top}\widehat{\mathbf{X}}\widetilde{\mathbf{P}}_{\text{in}}. \tag{5}$$

The algorithm for computing $\mathbf{V}_r$ can be viewed as estimating the principal subspace of the row space of $\mathbf{P}_{\text{in}}$, which is also the row space of $\mathbf{P}$ according to the CUR assumption (1). From this perspective, we can view our method as a *subspace estimation* procedure (Cichocki and Amari 2002). Computationally, the main step of this algorithm is the partial SVD, which can be efficiently done for large networks. In our empirical evaluation, the method can easily handle sparse networks of order 10^5 in an ordinary laptop.

Another interpretation of our algorithm comes from a low-rank approximation view. A rank-r approximation to $\mathbf{P}$ can always be written as $\mathbf{P}_r = \mathbf{R}^{\top}\mathbf{Z}\mathbf{R}$, where $\mathbf{R} \in \mathbb{R}^{r\times N}$ and $\mathbf{Z} \in \mathbb{R}^{r\times r}$. Recall that $\widetilde{\mathbf{P}}_{\text{in}} = \mathbf{U}_r\mathbf{D}_r\mathbf{V}_r^{\top}$. Therefore, we can rewrite (5) as

$$\widehat{\mathbf{P}} = \mathbf{V}_r\mathbf{D}_r\mathbf{U}_r^{\top}\widehat{\mathbf{X}}\mathbf{U}_r\mathbf{D}_r\mathbf{V}_r^{\top} = \mathbf{V}_r(\mathbf{D}_r\mathbf{U}_r^{\top}\widehat{\mathbf{X}}\mathbf{U}_r\mathbf{D}_r)\mathbf{V}_r^{\top} := \widehat{\mathbf{R}}^{\top}\widehat{\mathbf{Z}}\widehat{\mathbf{R}}.$$

Thus, $\mathbf{V}_r$ gives an estimated embedding of the network in a space equipped with an inner product represented by the matrix $\widehat{\mathbf{Z}}$.

Our method relies on both the CUR form and the low-rank assumption, which can in practice be relaxed to approximately low-rank. Checking the approximately low rank assumption in practice can be achieved via computing the stable rank $\text{SR}(\mathbf{A}) = \|\mathbf{A}\|_F^2/\|\mathbf{A}\|^2$. To understand where this set of assumptions fits in with existing models, consider two special cases.

The first case is the stochastic block model (SBM) (Holland and Leinhardt 1981), perhaps the most widely studied model for networks with communities. It is easy to see that the SBM can be written as a special case of the CUR form (1) because there are only K different rows of $\mathbf{P}$; thus, our assumptions include the SBM and are strictly more general. The CUR condition holds as long as we observe at least one node from each community. We can get a rough estimate of the sample size n for which this happens. Assume a data collection procedure from the population where each community has the same proportion. When n subjects are collected, the number of observations from each community follows a binomial distribution with expectation n/r. By Hoeffding's inequality (e.g., chap. 2.6 of Vershynin 2018), the probability of having zero subjects from one community is bounded by $2\exp(-\frac{cn}{r^2})$ with some absolute constant c. Therefore, the probability of missing at least one community in the n subjects is upper bounded by

$$2r\exp(-\frac{cn}{r^2}).$$

So for any $\delta > 0$, if $n > r^2 \log\frac{2r}{\delta}$, we can ensure that all communities have at least one subject in the sample (thus, the CUR assumption holds exactly) with probability at least $1 - \delta$.

The other case is a general low-rank model for $\mathbf{P}$. While assuming a low-rank structure in $\mathbf{P}$ is more general than our model, Drineas, Mahoney, and Muthukrishnan (2008) and Mahoney and Drineas (2009) show that if one can control which rows to sample, sampling $O(r\log r/\epsilon^2)$ rows would ensure that the CUR decomposition can approximate the general low-rank matrix up to a multiplicative factor ϵ with probability at least 0.7. Under uniform egocentric sampling, this statement still holds if the incoherence condition of Candès and Tao (2010) is further assumed for $\mathbf{P}$. Thus, when the sampling fraction is sufficiently large and the true model is low-rank but does not satisfy the CUR assumption, our method may still perform similarly to a generic low-rank model. In contrast, when n is small, as is frequently the case with egocentric sampling in practice (Fafchamps and Lund 2003; Bandiera and Rasul 2006; Ali and Dwyer 2009; Conley and Udry 2010; Banerjee et al. 2013), the general low-rank model may be too flexible and lead to overfitting, whereas our method can be viewed as a form of regularization that can mitigate overfitting. These conjectures are confirmed by our experiments in Section 3.

2.3. Tuning Parameter Selection

The approximation rank r is a tuning parameter we have to choose in a data-driven fashion. We do this with a network cross-validation method which can be viewed as a hybrid of Chen and Lei (2018) and Li, Levina, and Zhu (2020b), adapted to the ego-sampled setting. Specifically, we randomly sample a subset of nodes $\mathcal{T} \subset \{1, \ldots, n\}$ and set $\mathbf{A}_{-\mathcal{T},in}$ to be the matrix resulting from deleting the rows in $\mathcal{T}$ from $\mathbf{A}_{\text{in}}$. Applying the subspace estimation algorithm to $\mathbf{A}_{-\mathcal{T},in}$ with a sequence of r values, we then estimate predictive accuracy for each r by computing the area under the ROC curve (AUC) on the entries $A_{\mathcal{T},\text{in}}$, and choose the value of r that achieves the maximum AUC. Moreover, as suggested in Chen and Lei (2018), repeating the above procedure multiple times and aggregating the results by picking a quantile or mode of the selection can significantly improve the tuning stability. In our examples, we set $|\mathcal{T}| = 0.1n$, and search for the rank from 2 to the stable rank plus 10. We also replicate this procedure 20 times, and picked the 80% quantile of the 20 selected ranks as our rank to use for model fitting. Similar methods are used for other benchmark methods involving tuning parameters in our empirical studies in Sections 3 and 4. Self-tuning or tuning-free methods, such as the universal singular value thresholding (Chatterjee 2015), will further reduce computational cost, but they give inferior results in most settings of our evaluation.

3. Evaluation on Synthetic Networks

In this section, we compare our subspace estimation method (SE) to several benchmark algorithms for link prediction. The benchmark algorithms can be grouped into the following classes.

1. Matrix completion methods. The first method applies the CUR algorithm to the "naive" plug-in estimate $\mathbf{A}_{11}$ for $\mathbf{P}_{11}$ in (3), as discussed at the beginning of Section 2.2. This method is labeled "CUR" in the results. The second method in this class is the nuclear norm regularization with inexact augmented Lagrange multiplier method (Lin, Chen, and Ma 2010), labeled "MC." The MC method is tuned by the cross-validation method of Chen and Lei (2018), with five replicates for stability selection.
2. Graphon methods. These include the popular universal singular value thresholding method (Chatterjee 2015), labeled "USVT," and the neighborhood smoothing method for graphon estimation (Zhang, Levina, and Zhu 2017), labeled "NS." The NS method is based on a similarity measure between nodes and thus can be applied in the egocentric sampling setting, with $\mathbf{A}^2$ replaced by $\mathbf{A}_{in}^{\top}\mathbf{A}_{in}$ in the algorithm of Zhang, Levina, and Zhu (2017).
3. Parametric models. These include the exponential random graph model (labeled "ERGM") from Handcock and Gile (2010) with egocentric sampling and the geometrically-weighted edgewise shared partnerships of Hunter (2007), as implemented in the R package `ergm` (Handcock et al. 2019). The second parametric model is the stochastic block model (labeled "SBM"), adapted to egocentric sampling with the fitting strategy of Chen and Lei (2018) which is also used to estimate the number of blocks, with 20 replicates for stability selection. The third parametric model is the Bayesian version of the random dot product graph model (labeled "RDPG"). It is unclear how to fit the standard RDPG in the egocentric setting, but the additive and multiplicative effect models of Hoff (2009) include the Bayesian version of the RDPG as a special case. The model fitting is based on the R package `amen` (Hoff, Fosdick, and Volfovsky 2020), tuned by the cross-validation of Chen and Lei (2018).
4. The oracle method. We directly use the true model $\mathbf{P}$ as the oracle reference.
5. Ensemble learning for link prediction. In practice, combining multiple single models or algorithms can result in significant performance improvement for link prediction (Ghasemian et al. 2020). In particular, if link prediction is the primary task, Ghasemian et al. (2020) proposed using an approach they called Optimal Link Prediction (OLP), exploiting a hierarchical learning strategy to combine strengths of individual methods by random forest. The final ensemble prediction can be better than any of the individual methods (on average). We include two methods based on this strategy. One is topological stacking ("Topo") based on the implementation of Ghasemian et al. (2020), with the one difference that we drop the Adamic-Adar index, because it cannot be computed for egocentrically sampled data, and rely on the remaining 41 topological link prediction metrics. We also include a version that combines our SE method with the 41 topological metrics ("SeTopo"), as the strategy of Ghasemian et al. (2020) is general and, in principle, can combine any link prediction methods. There are other ensemble learning link prediction approaches in the literature, such as Peixoto (2015), Peixoto (2018), Yao et al. (2021), and Li and Le (2021). We only include Topo and SeTopo to demonstrate the potential of ensemble learning for link prediction; we note that these methods cannot be used to learn anything model-based, such as network summary statistics, which are also part of the evaluation. For both "Topo" and "SeTopo," we tune the random forest using the cross-validation strategy of Ghasemian et al. (2020), with the number of trees selected from {50, 100} and tree depths selected from {3, 6, 10}.

The methods without publicly available software have been implemented in Python. While only the SE and CUR methods are designed specifically for the egocentric sampling setting, the algorithms fitting NS, MC, ERGM, RDPG, and the SBM are applicable to this setting either directly or with minor modifications which we implemented. We conjecture that USVT is not valid under egocentric sampling because it relies on the uniform dyad sampling assumption for its threshold and scaling values. The ensemble methods do not rely on any statistical model or assumptions, so it is difficult to assess their validity, but our results below show that empirically they work well under egocentric sampling.

3.1. Network Generating Models

The synthetic networks are generated from three widely used network models, which all fall under the inhomogeneous Erdös-Renyi framework. The first model is the stochastic block model (SBM) proposed by Holland, Laskey, and Leinhardt (1983), with $K = 5$ communities. We randomly assign the community label Z_i for nodes $i = 1, \ldots, N$, a value from $\{1, 2, 3, 4, 5\}$, all with equal probability. The connection probability between nodes is set to $p_{ij} = 0.05$ if $Z_i \neq Z_j$, and to $p_{ij} = .05 + \frac{Z_i - 0.3}{6}$ if $Z_i = Z_j$. The second model is the random dot product graph (RDPG) model (Young and Scheinerman 2007). Following Athreya et al. (2018), we generate five-dimensional vectors $Z_i, i = 1, \ldots, N$ independently with each coordinate being sampled from Beta(0.5, 1) and define $p_{ij} = Z_i^T Z_j$. The third model is the latent space model proposed by Hoff, Raftery, and Handcock (2002). Here we generate five-dimensional latent vectors $Z_i, i = 1, \ldots, N$ from $N(0, I_5)$ and set $p_{ij} = 1/(1 + \exp(\|Z_i - Z_j\|))$. We call this the "distance model," as the strength of connectivity between two nodes is determined by the distance between their positions in the latent space. The three models are

Table 1. Generative models for synthetic networks: distributions of latent variables, link functions, and the rank of the matrix **P**. Unif$\{1, \ldots, 5\}$ is the uniform distribution on the integers $\{1, \ldots, 5\}$; Beta$(0.5, 1)$ is a beta distribution with parameters 0.5 and 1; $N(0, I_5)$ is a five-dimensional multivariate standard Gaussian.

Model	Distr. of Z_i	$f(Z_i, Z_j)$	Rank(**P**)
SBM	Unif$\{1, \ldots, 5\}$	$0.05 + \frac{i-0.3}{6}1(i=j)$	5
RDPG	Beta$(0.5, 1)$	$Z_i^\top Z_j$	5
Distance	$N(0, I_5)$	$(1 + e^{(\|Z_i - Z_j\|)})^{-1}$	full

summarized in Table 1. The SBM is the simplest model of the three, which admits community structures but does not allow for any heterogeneity within a community. The product model is a more general low-rank model that includes the (assortative) SBM as a special case (Athreya et al. 2018). The latent vectors Z_i's can be generated from many general distributions, as long as their inner products give valid probabilities. The distance model is the most general, with no constraint on the distribution of Z_i's at all. It does not necessarily have a low-rank probability matrix. For all simulations, we first generate iid Z_i's for $i = 1, \dots, N$, and then generate $A_{ij} \sim \text{Bernoulli}(\phi f(X_i, X_j))$, where ϕ controls the average degree and f is specified by the model (see Table 1).

For a given network model, the difficulty of the problem is primarily controlled by two quantities, the sampling fraction ρ and the average node degree determined by ϕ. Different configurations of the two quantities are evaluated. All results are based on average performance over 50 replications.

3.2. Evaluation Criterion

We will evaluate the methods in three different aspects that can be crucial in practice: *the link prediction accuracy*, *prediction of network global statistics*, and *timing*.

For link prediction accuracy, since the links are always either 1 or 0, a threshold set on the probability for prediction would result in both true and false positives. There are various ways to measure performance in this context. In particular, varying the threshold allows us to trace the entire ROC curve described by the true positive rate (TPR) and the false positive rate (FPR), defined as

$$\text{TPR} = \frac{\#\{\text{correctly predicted edges}\}}{\#\{\text{All true edges}\}},$$
$$\text{FPR} = \frac{\#\{\text{falsely predicted edges}\}}{\#\{\text{All null edges}\}}$$

as well as the Precision-Recall (PR) curve described by precision and recall, defined as

$$\text{Precision} = \frac{\#\{\text{correctly predicted edges}\}}{\#\{\text{All predicted edges}\}},$$
$$\text{Recall} = \frac{\#\{\text{correctly predicted edges}\}}{\#\{\text{All true edges}\}}.$$

Note that recall and the true positive rate are the same metric, so the difference between the ROC and the PR curves lies in the difference between the false positive rate and precision. Generally speaking, the ROC curve gives a more balanced measure, while the PR curve focuses on performance on true edges rather than both true and absent edges. When the network is very sparse, the ROC curve may favor methods that do not predict any edges at all, and the PR curve does not have that problem, so we will report both measures when we can. Both curves depend only on the ordering of the predicted probabilities of links, not their magnitude, and are invariant under a monotone transformation applied entry-wise to $\widehat{\mathbf{P}}$.

Model-based methods can also be used to predict global network summary statistics, from the partially observed network. We focus on three widely used statistics: edge density, global clustering coefficient and global eigencentrality. The density is defined by

$$\lambda = \frac{\sum_{i<j} A_{ij}}{n(n-1)/2}.$$

The global clustering coefficient of a network is defined to be

$$CC = \frac{\#\text{ of closed triplets}}{\#\text{ of all triplets}}.$$

Given a connection probability estimate $\widehat{\mathbf{P}}$, we can estimate the clustering coefficient by replacing both the numerator and denominator with their expectations under $\widehat{\mathbf{P}}$.

The eigencentrality score of a node is defined to be the absolute value of its coordinate in the leading eigenvector of $\mathbf{A}$. Let w be the leading eigenvector of $\mathbf{A}$, and the global eigencentrality is defined as

$$C = \sum_i (\max_i w_i - w_i)/c,$$

where c is a computable constant that only depends on N. Eigencentrality is widely used in many network algorithms, such as Google's PageRank (Page et al. 1999).

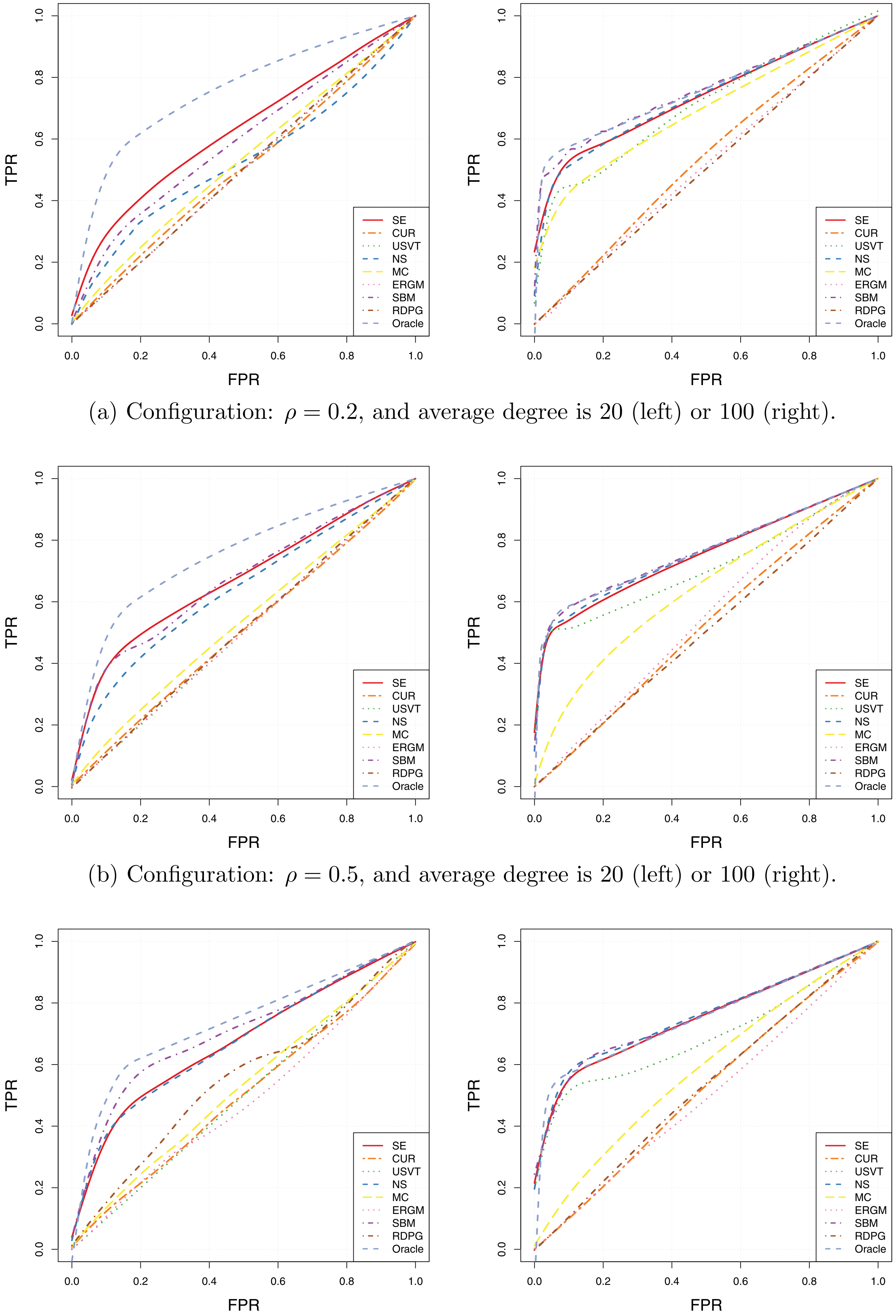
When only $\mathbf{A}_{22}$ is missing, we use the link prediction algorithms to impute it by $\widehat{\mathbf{P}}_{22}$ as the point estimate. The network statistics can then be computed from this imputed adjacency matrix. As suggested by the associate editor, a natural predictor for the density may be empirical estimator $\lambda_{\text{emp}} = \frac{\sum_{i<j,(i,j)\in\Omega} A_{ij}}{|\Omega|}$. This empirical estimator will also be included for evaluation. For completeness, we also include two "naive" empirical estimators for the global clustering coefficient and eigencentrality. These are calculated by directly treating the missing entries of $\mathbf{A}$ as zeros. Performance is measured by the relative absolute error $|\hat{C} - C|/C$, which does not depend on the constant c in the definition of eigenvector centrality.

For the timing comparison, since many methods involve a tuning procedure for which the configuration and preference can vary a lot across users, we do not include the tuning procedure in timing evaluation and only focus on one model-fitting procedure. For reference, SE can be tuned by cross-validation for which the timing would be the number of cross-validations times the single model fitting timing. Our implementation does not take advantage of the sparse matrix structure, so in principle the times for all of SE, CUR, USVT, NS, and the SBM can be further improved. Also, some of the methods are implemented in R and others in Python. However, we believe these approximate comparisons are enough to get a sense of the relative scalability of different methods.

3.3. Link Prediction Accuracy

For each of the three models, we consider the settings of $\rho = 0.2, 0.5, 0.9$ and the average degree of 20 or 100. The ROC curves and PR curves are shown in Figures 3–4 (SBM), Figures 5–6 (product model) and Figures 7–8 (distance model). We start from the single model-based methods.

When the true model is the SBM, the ROC curves in Figure 3 show that sometimes (when $\rho = 0.2$ and the degree is 20), our



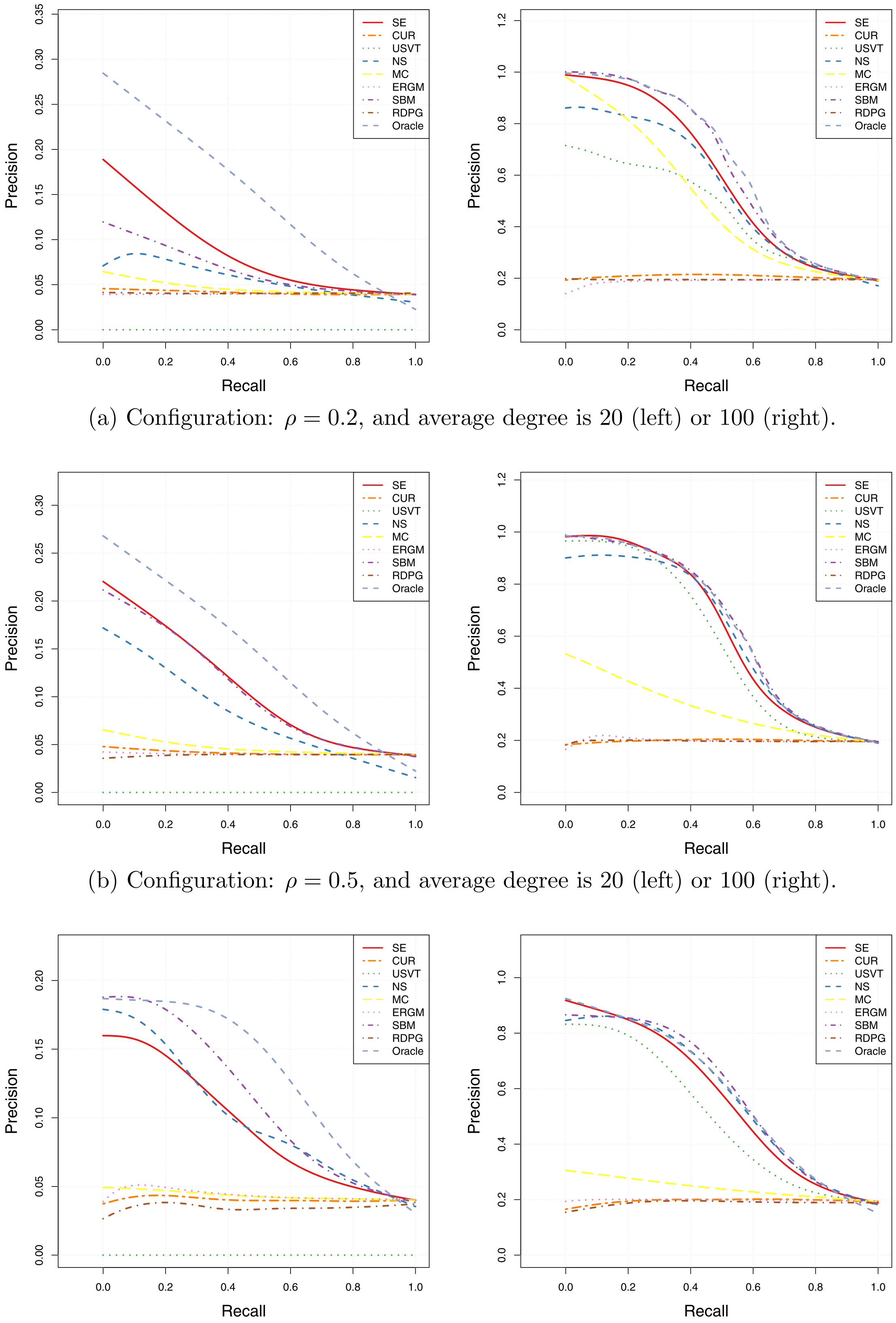
(a) Configuration: $\rho = 0.2$, and average degree is 20 (left) or 100 (right).

(b) Configuration: $\rho = 0.5$, and average degree is 20 (left) or 100 (right).

(c) Configuration: $\rho = 0.9$, and average degree is 20 (left) or 100 (right).

Figure 3. ROC curves of link prediction performance when the network is generated by the SBM.

method (SE) is more accurate than fitting the SBM model itself. This is because cross-validation tends to be conservative for the SBM and selects a smaller number of blocks than necessary. As the network becomes denser, most methods improve, and in particular, the NS method performs similarly to our method. Fitting the SBM eventually becomes the best for dense networks and



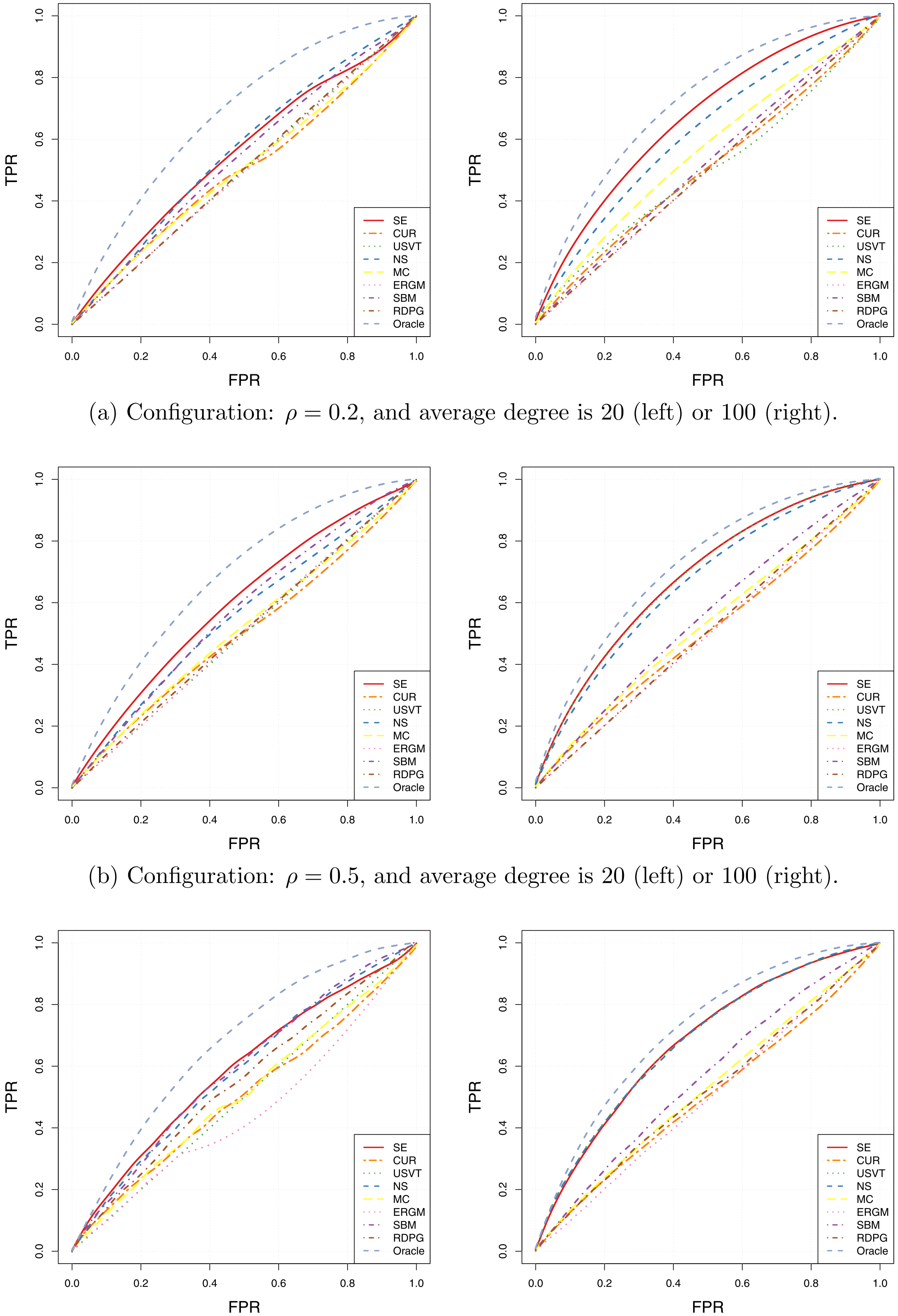
(a) Configuration: $\rho = 0.2$, and average degree is 20 (left) or 100 (right).

(b) Configuration: $\rho = 0.5$, and average degree is 20 (left) or 100 (right).

(c) Configuration: $\rho = 0.9$, and average degree is 20 (left) or 100 (right).

Figure 4. PR curves of link prediction performance when the network is generated by the SBM.

matches the oracle performance, as expected. The performance also improves when ρ increases, especially going from 0.2 to 0.5, and less so from 0.5 to 0.9. USVT also performs well on dense networks, but does not adapt well to sparsity. The comparison remains overall similar when the performance is measured by PR curves instead (Figure 4).



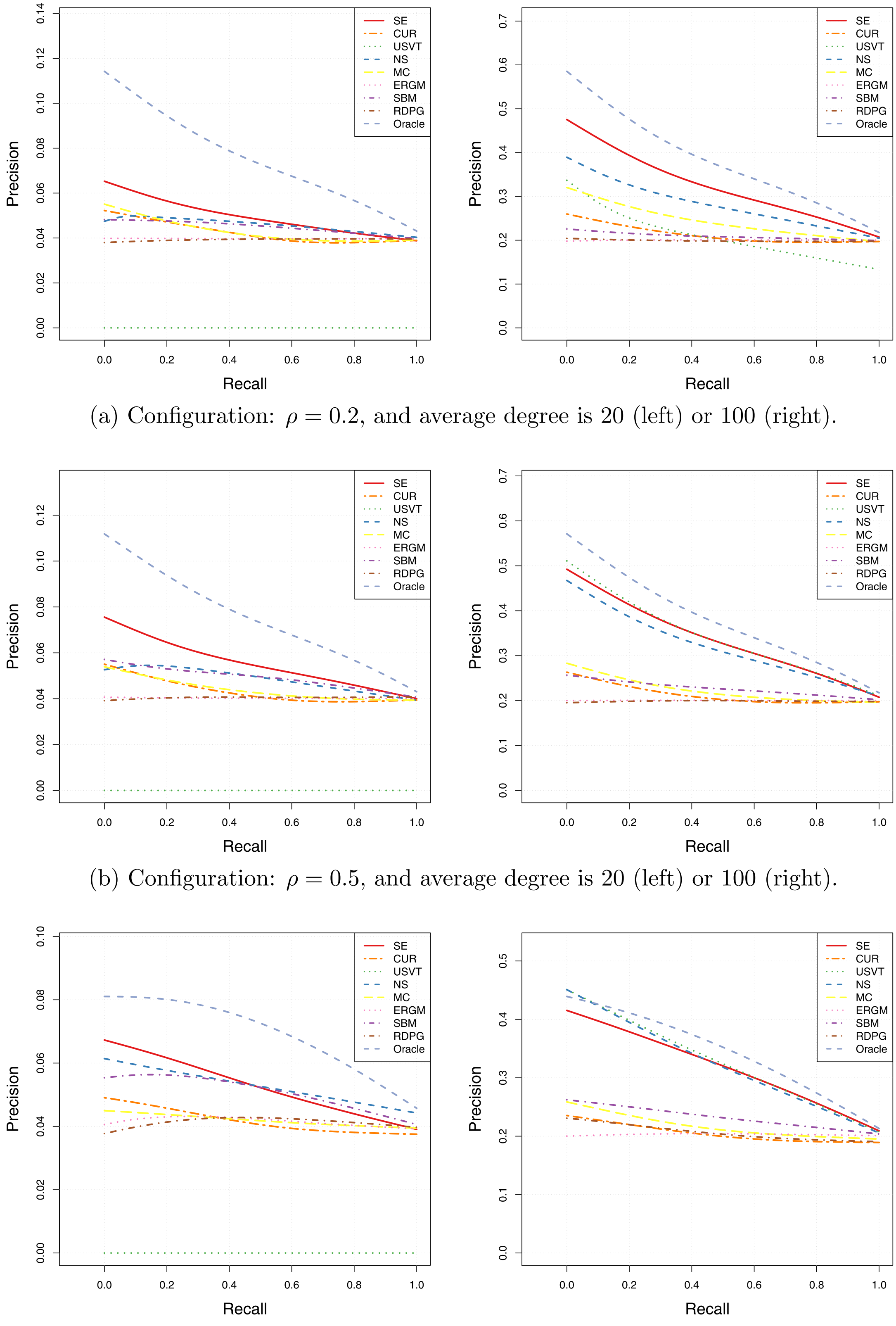
(a) Configuration: $\rho = 0.2$, and average degree is 20 (left) or 100 (right).

(b) Configuration: $\rho = 0.5$, and average degree is 20 (left) or 100 (right).

(c) Configuration: $\rho = 0.9$, and average degree is 20 (left) or 100 (right).

Figure 5. ROC curves of link prediction performance when the network is generated by the product model.

The product model and the distance model produce similar patterns. In the most challenging setting of $\rho = 0.2$ and the average degree of 20, our method has the biggest advantage, with smaller differences in easier settings. Overall, the NS method is also competitive if ρ is not too small, and the network is not too sparse. The USVT works well only for dense networks. The



(a) Configuration: $\rho = 0.2$, and average degree is 20 (left) or 100 (right).

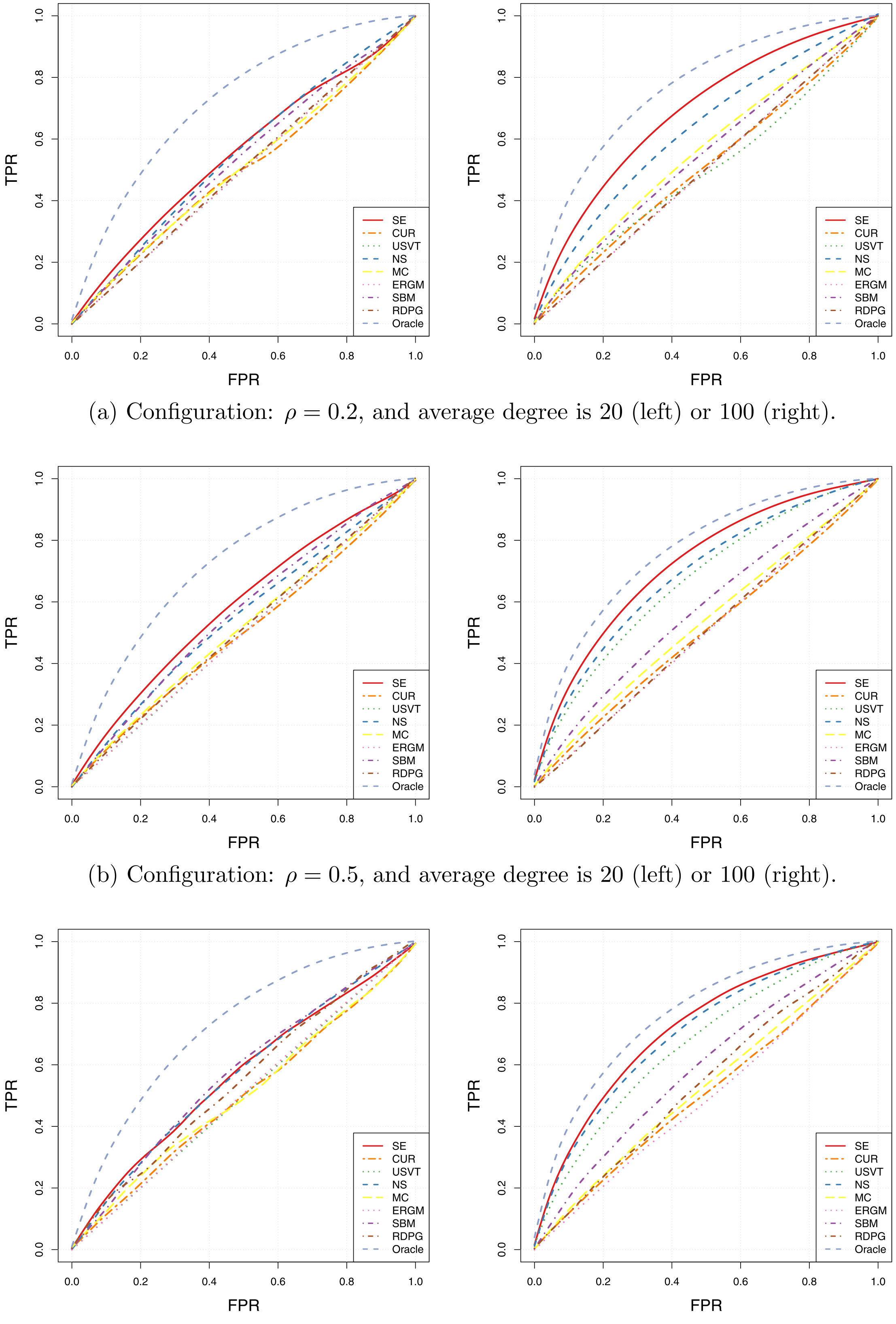
(b) Configuration: $\rho = 0.5$, and average degree is 20 (left) or 100 (right).

(c) Configuration: $\rho = 0.9$, and average degree is 20 (left) or 100 (right).

Figure 6. PR curves of link prediction performance when the network is generated by the product model.

SBM can achieve moderately good link prediction performance in some settings.

Next, we compare our SE method with the ensemble learning methods Topo and SeTopo based on the OLP strategy proposed by Ghasemian et al. (2020). Due to space limitations, we only show the PR curves, given in Figures 9–10. The corresponding ROC curves are included in supplementary materials Section C for completeness.



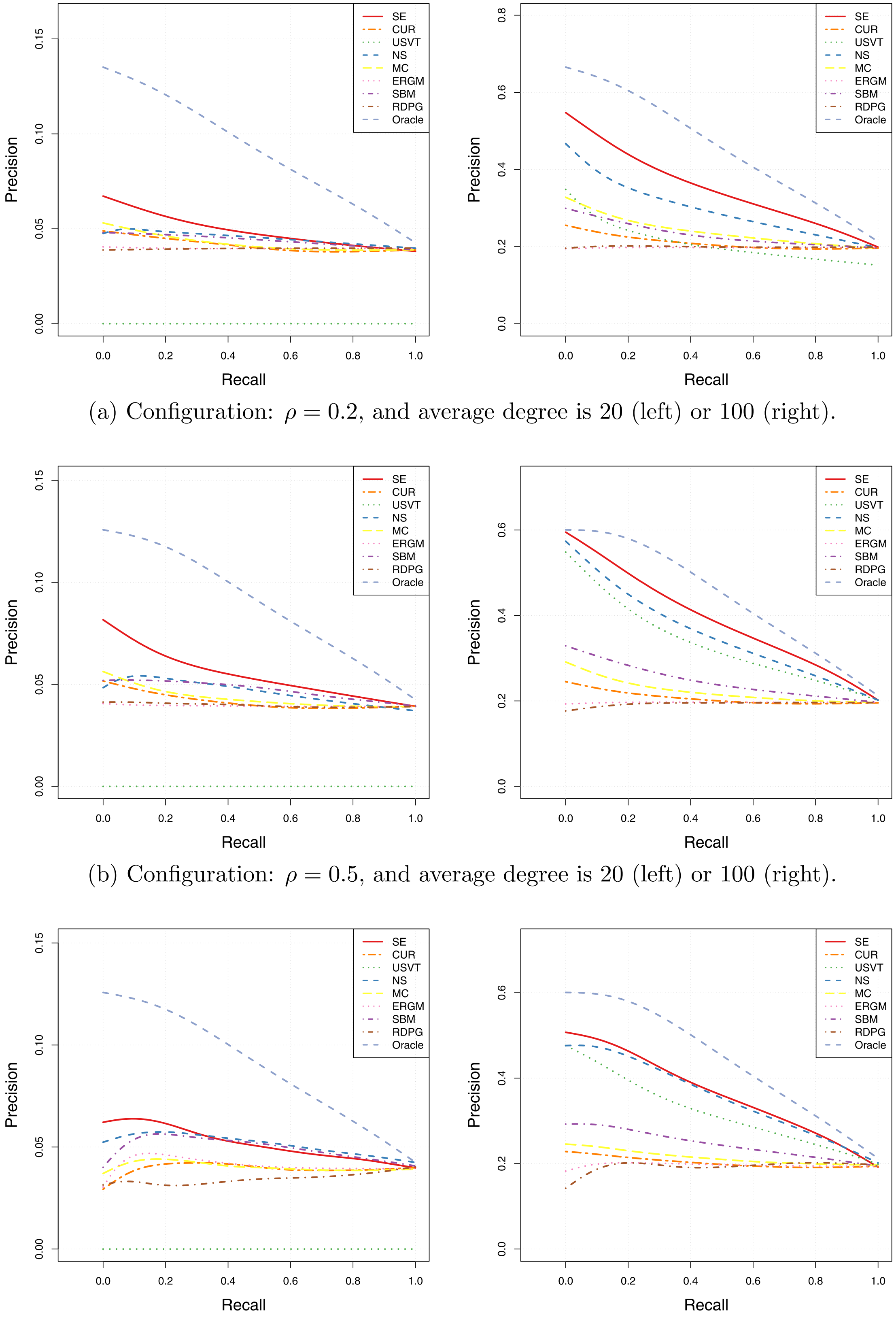
(a) Configuration: $\rho = 0.2$, and average degree is 20 (left) or 100 (right).

(b) Configuration: $\rho = 0.5$, and average degree is 20 (left) or 100 (right).

(c) Configuration: $\rho = 0.9$, and average degree is 20 (left) or 100 (right).

Figure 7. ROC curves of link prediction performance when the network is generated by the distance model.

Our results show that while Topo can make reasonably good predictions in most settings, it is less accurate than the SE method. This is expected since Topo does not match the true setting. SeTopo, in contrast, is always similar to or better than SE. These results confirm the observation of Ghasemian et al. (2020) that, for link prediction, an



(a) Configuration: $\rho = 0.2$, and average degree is 20 (left) or 100 (right).

(b) Configuration: $\rho = 0.5$, and average degree is 20 (left) or 100 (right).

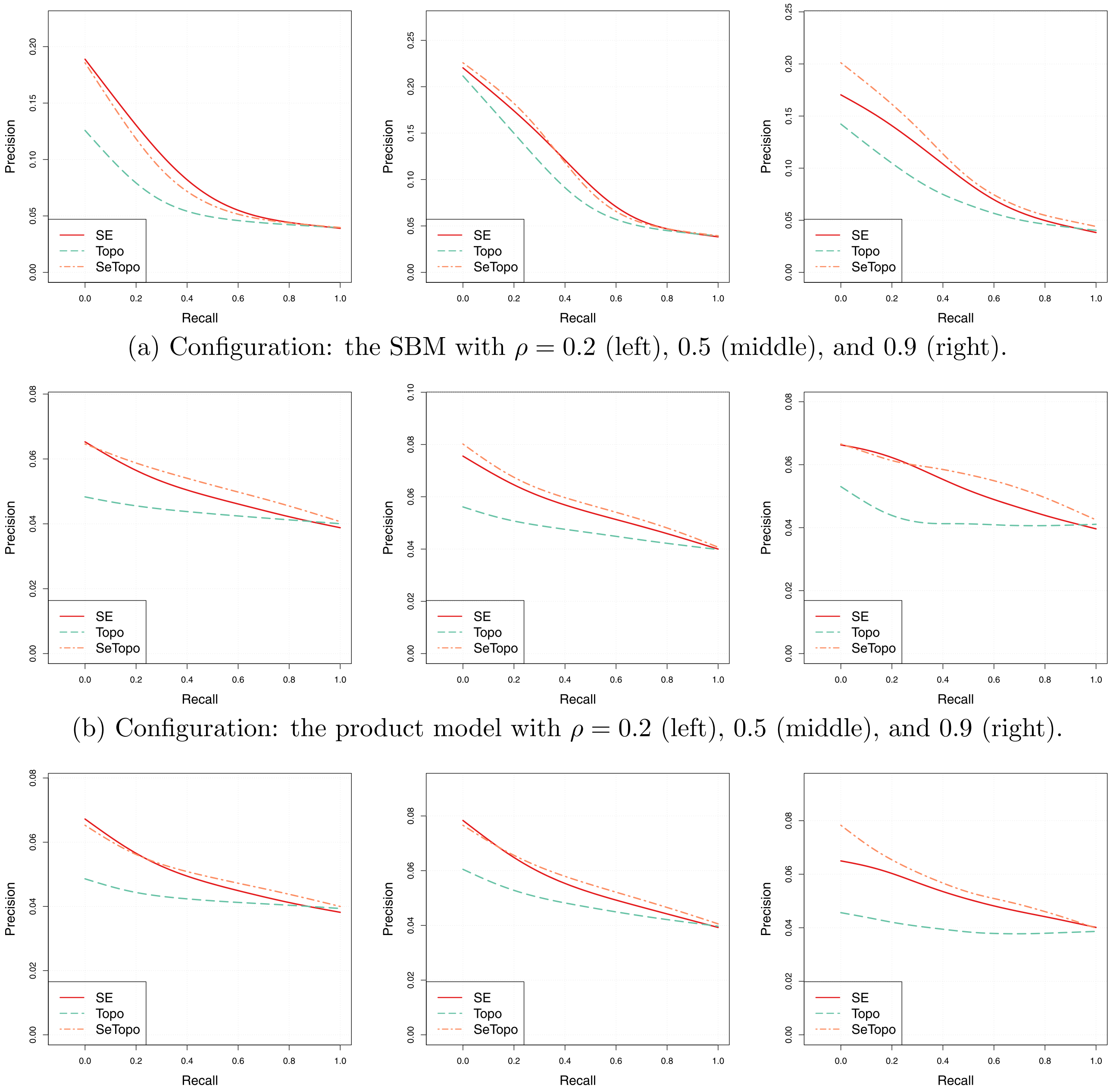
(c) Configuration: $\rho = 0.9$, and average degree is 20 (left) or 100 (right).

Figure 8. PR curves of link prediction performance when the network is generated by the distance model.

ensemble can outperform any individual method, and show that including SE in the ensemble substantially improves performance.

3.4. Predicting Global Network Summaries

Tables 2–4 present the results for global network density, global clustering coefficient, and eigencentrality prediction.



(a) Configuration: the SBM with $\rho = 0.2$ (left), 0.5 (middle), and 0.9 (right).

(b) Configuration: the product model with $\rho = 0.2$ (left), 0.5 (middle), and 0.9 (right).

(c) Configuration: the distance model with $\rho = 0.2$ (left), 0.5 (middle), and 0.9 (right).

Figure 9. PR curves of link prediction performance in comparison with the ensemble methods when the average degree is 20.

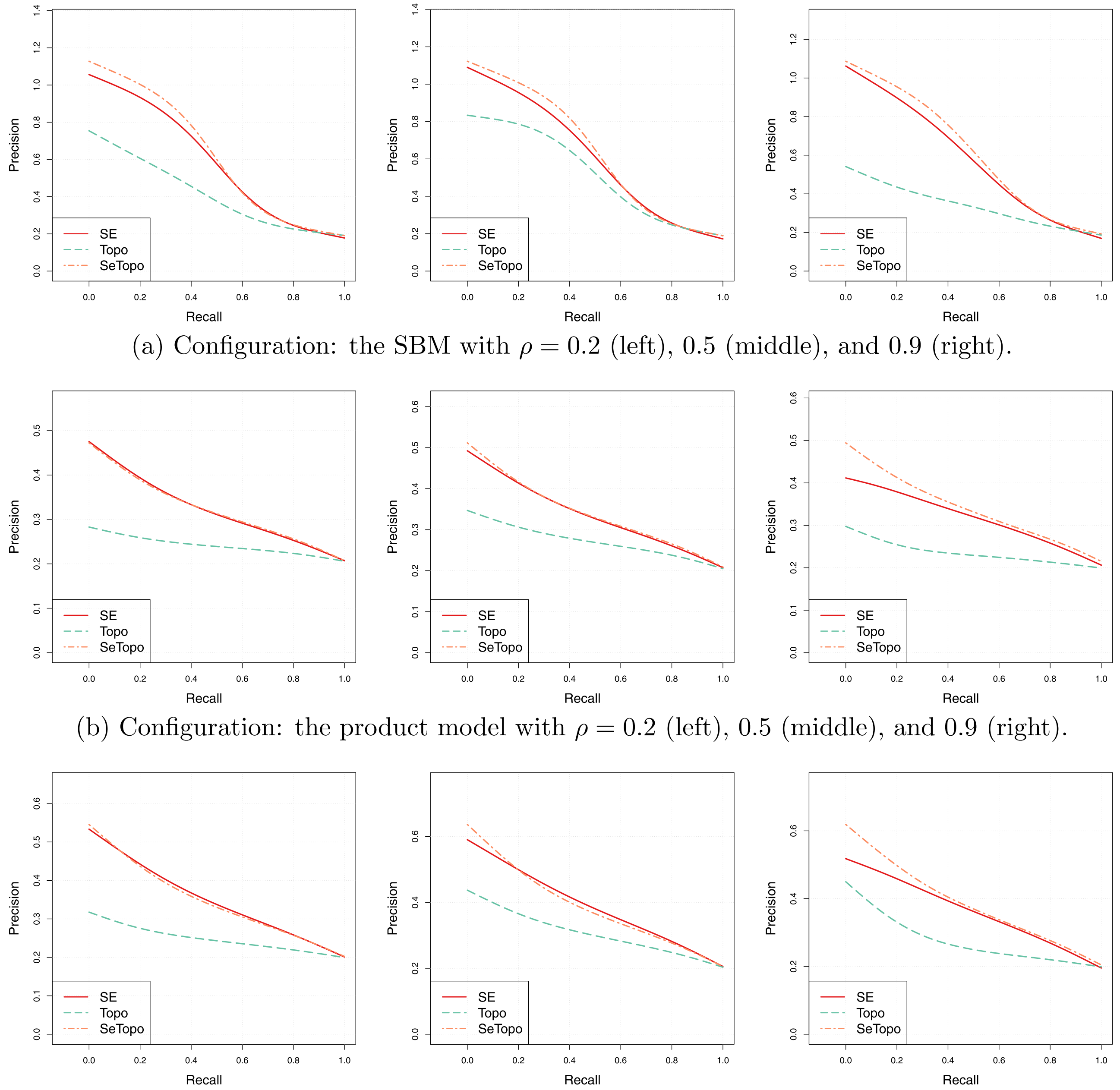
The smallest error in each setting is shown in bold. If multiple methods show no statistically significant difference from the best one, they are also shown in bold.

3.4.1. *Network Density Prediction* (Table 2)

The SBM, as expected, is the best or close to the best in all configurations when the true model is the SBM. It also gives competitive estimators when ρ is large under other models. The empirical estimator turns out to be the best in sparse settings and close to the best in dense networks. Our proposed method, SE, is the best in dense networks and close to the best in sparse settings. The NS and ERGM are competitive in a few settings but give poor results in other settings. For example, the ERGM appears to work relatively well for sparse networks, but not for denser ones.

3.4.2. *Global Clustering Coefficient Prediction* (Table 3)

The empirical estimator performs poorly on this task. The SBM estimator is still the best under the SBM, as well as for sparse yet high sampling proportion settings under the other two models. The SE and NS are generally competitive for predicting this statistic. The SE is again the best in dense networks and close to the best in sparse settings, while the NS is the best in sparse settings and close to the best in dense ones. The MC and USVT give good estimators occasionally but are not competitive in most settings.



(a) Configuration: the SBM with $\rho = 0.2$ (left), 0.5 (middle), and 0.9 (right).

(b) Configuration: the product model with $\rho = 0.2$ (left), 0.5 (middle), and 0.9 (right).

(c) Configuration: the distance model with $\rho = 0.2$ (left), 0.5 (middle), and 0.9 (right).

Figure 10. PR curves of link prediction performance in comparison with the ensemble methods when the average degree is 100.

3.4.3. *Eigencentrality Prediction* (Table 4)

There is not a clear winner for this statistic. The SBM is the best or close to the best under the SBM. The SE, NS, and ERGM are the most competitive overall. The empirical estimator, CUR, and MC are not competitive with the best methods.

Overall, our proposed method, the SE, provides the best or close to the best predictions for all three statistics when the network is dense and highly competitive in sparse settings. The NS generally works well for global clustering coefficient and eigencentrality, but is less accurate for network density. The SBM estimator is mainly helpful if the network is generated from the SBM. The empirical estimator is accurate for network density but not for either global clustering coefficient or eigencentrality.

3.5. *Timing Comparisons*

There are three main factors that impact timing: the network size N, the sampling fraction $\rho = n/N$, and the average degree of the network. We evaluate the average time that it takes to fit a single model in the settings of the previous section. We also include additional settings with $N = 2000, 5000$, the degree is set to $\log n^{1.5}$, and $\rho \in (0.2, 0.5, 0.9)$. The computation was done on a Linux system with 128GB of RAM. Times longer than 3 hr are considered not competitive and are not reported. The results

Table 2. Relative errors in predicting the densities of the synthetic networks.

Model	(ρ, degree)	avg. λ	SE	CUR	NS	MC	USVT	SBM	ERGM	RDPG	Empirical
SBM	(0.2, 20)	0.039	0.074	5.053	0.113	2.752	0.638	0.037	0.027	0.692	**0.023**
	(0.2, 100)	0.197	0.018	0.865	0.092	0.291	0.494	**0.008**	0.504	0.054	0.017
	(0.5, 20)	0.039	0.013	2.454	0.024	1.266	0.250	**0.009**	0.012	0.052	0.010
	(0.5, 100)	0.197	**0.005**	0.359	0.017	0.042	0.104	**0.004**	0.184	0.055	0.007
	(0.9, 20)	0.039	**0.001**	0.100	**0.001**	0.010	0.010	**0.001**	**0.001**	0.002	**0.001**
	(0.9, 100)	0.197	**<0.001**	0.014	**<0.001**	0.006	0.002	0.001	0.003	0.001	0.001
Product	(0.2, 20)	0.039	0.074	4.568	0.143	2.577	0.637	0.051	**0.029**	0.784	**0.030**
	(0.2, 100)	0.197	**0.017**	0.851	0.048	0.355	0.632	0.026	0.271	0.173	0.022
	(0.5, 20)	0.039	0.015	2.403	0.045	1.219	0.251	0.013	0.013	0.053	**0.012**
	(0.5, 100)	0.197	**0.005**	0.342	0.022	0.049	0.103	0.009	0.078	0.075	0.009
	(0.9, 20)	0.039	0.002	0.100	0.002	0.007	0.010	**0.001**	**0.001**	0.003	0.002
	(0.9, 100)	0.197	**0.001**	0.014	**0.001**	0.006	0.002	**0.001**	0.003	0.003	**0.001**
Distance	(0.2, 20)	0.039	0.066	4.779	0.133	2.589	0.640	0.050	0.034	0.826	**0.022**
	(0.2, 100)	0.197	**0.017**	0.836	0.045	0.359	0.629	0.024	0.267	0.067	0.022
	(0.5, 20)	0.039	0.013	2.440	0.041	1.214	0.250	0.013	**0.010**	0.072	**0.010**
	(0.5, 100)	0.197	**0.004**	0.356	0.019	0.052	0.103	0.012	0.076	0.058	0.010
	(0.9, 20)	0.039	0.002	0.103	0.002	0.009	0.010	**0.001**	**0.001**	0.002	**0.001**
	(0.9, 100)	0.197	**<0.001**	0.014	0.001	0.006	0.002	0.001	0.003	0.004	0.001

NOTE: The relative error is defined as $|\hat{\lambda} - \lambda|/\lambda$

Table 3. The relative errors in predicting the global clustering coefficients of the synthetic networks.

Model	(ρ, degree)	avg. CC	SE	CUR	NS	MC	USVT	SBM	ERGM	RDPG	Empirical
SBM	(0.2, 20)	0.085	0.242	3.787	**0.130**	2.033	0.563	0.220	0.474	0.254	0.590
	(0.2 ,100)	0.405	0.028	0.295	0.158	0.233	0.334	**0.006**	0.570	0.448	0.548
	(0.5, 20)	0.089	0.063	2.926	0.047	1.770	0.206	**0.039**	0.247	0.097	0.216
	(0.5, 100)	0.409	0.007	0.172	0.035	0.113	0.079	**0.002**	0.209	0.238	0.204
	(0.9, 20)	0.086	**0.004**	0.313	**0.004**	0.010	0.008	**0.003**	0.014	0.014	0.010
	(0.9, 100)	0.406	**<0.001**	0.015	**<0.001**	0.011	0.002	**<0.001**	0.013	0.009	0.010
Product	(0.2, 20)	0.052	0.335	5.657	**0.048**	3.931	0.567	0.124	0.176	1.187	0.558
	(0.2, 100)	0.258	**0.041**	0.845	0.045	0.219	0.545	0.171	0.348	0.062	0.560
	(0.5, 20)	0.051	0.062	5.342	**0.030**	3.626	0.206	0.029	0.091	0.188	0.212
	(0.5, 100)	0.263	0.014	0.373	0.022	**0.010**	0.083	0.081	0.128	0.127	0.203
	(0.9, 20)	0.051	0.005	0.691	**0.003**	0.008	0.010	**0.003**	0.005	0.008	0.009
	(0.9, 100)	0.261	**0.001**	0.006	**0.001**	0.002	**0.001**	0.004	0.006	0.005	0.009
Distance	(0.2, 20)	0.052	0.310	7.183	**0.056**	3.717	0.560	0.140	0.220	1.043	0.562
	(0.2, 100)	0.270	**0.057**	0.514	0.072	0.243	0.543	0.190	0.367	0.177	0.553
	(0.5, 20)	0.051	0.055	5.309	**0.035**	3.346	0.212	0.040	0.101	0.184	0.211
	(0.5, 100)	0.268	**0.018**	0.361	0.035	0.029	0.095	0.096	0.141	0.128	0.200
	(0.9, 20)	0.052	**0.005**	0.636	**0.004**	0.008	0.010	**0.004**	**0.005**	0.007	0.010
	(0.9, 100)	0.269	**0.001**	0.005	0.002	**0.001**	0.003	0.005	0.007	0.006	0.009

NOTE: The relative error is defined as $|\widehat{CC} - CC_{\text{true}}|/CC_{\text{true}}$

Table 4. Relative errors in predicting the eigencentralities of the synthetic networks.

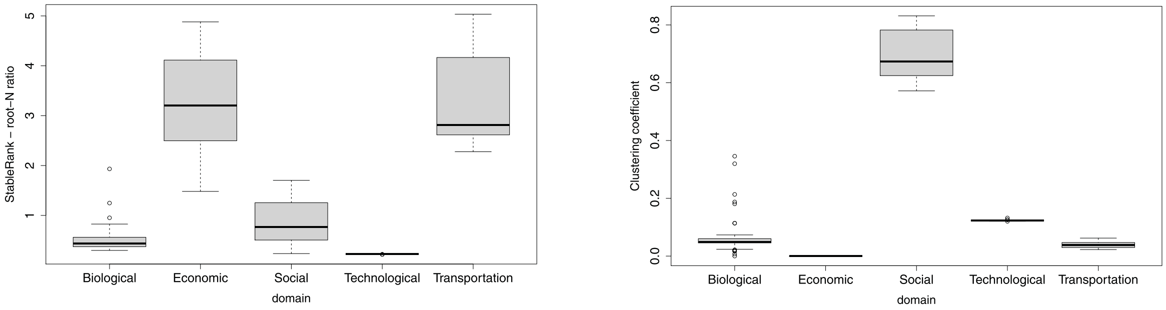
Model	(ρ, degree)	avg. C	SE	CUR	NS	MC	USVT	SBM	ERGM	RDPG	Empirical
SBM	(0.2, 20)	45.2	0.329	0.636	0.171	0.233	1.002	**0.173**	0.276	0.222	0.985
	(0.2, 100)	22.4	0.068	0.545	0.180	0.515	0.748	**0.028**	1.047	0.253	1.373
	(0.5, 20)	44.4	0.149	0.484	**0.053**	0.090	0.248	0.068	0.094	0.087	0.262
	(0.5, 100)	22.4	0.020	0.323	0.037	0.121	0.161	**0.011**	0.257	0.064	0.370
	(0.9, 20)	44.7	0.011	0.314	0.006	0.029	0.014	0.004	**0.011**	0.003	0.019
	(0.9, 100)	22.5	**0.003**	0.024	**0.001**	0.015	**0.003**	**0.002**	0.006	**0.002**	0.016
Product	(0.2, 20)	30.1	0.483	0.446	0.288	0.156	1.237	0.236	**0.136**	0.486	1.280
	(0.2, 100)	23.4	0.156	0.563	0.113	0.499	1.299	**0.103**	0.530	0.306	1.373
	(0.5, 20)	32.4	0.173	0.279	**0.079**	0.487	0.317	0.089	0.084	0.637	0.300
	(0.5, 100)	22.8	**0.044**	0.345	0.047	0.101	0.129	0.064	0.141	0.108	0.346
	(0.9, 20)	31.0	0.013	0.736	**0.008**	0.028	0.020	0.011	0.021	0.011	0.019
	(0.9, 100)	23.1	**0.004**	0.023	0.006	0.008	**0.004**	0.009	0.009	0.010	0.023
Distance	(0.2, 20)	29.7	0.664	0.380	0.280	0.181	1.320	0.218	**0.140**	0.540	1.316
	(0.2, 100)	20.4	0.179	0.526	**0.096**	0.596	1.367	0.138	0.510	0.202	1.424
	(0.5, 20)	30.0	0.175	0.224	0.085	0.556	0.332	0.094	**0.078**	0.801	0.349
	(0.5, 100)	20.5	0.063	0.271	**0.050**	0.121	0.139	0.054	0.133	0.108	0.357
	(0.9, 20)	30.1	0.015	0.873	0.012	0.035	0.020	0.012	0.008	**0.007**	0.019
	(0.9, 100)	20.1	**0.002**	0.027	0.005	0.016	0.004	0.007	0.014	0.034	0.026

NOTE: The relative error is defined as $|\widehat{C} - C_{\text{true}}|/C_{\text{true}}$.

Table 5. Timing comparisons with different configurations under the distance model (in seconds) averaged over 50 replications.

N, degree, ρ	SE	CUR	NS	MC	USVT	SBM	ERGM	RDPG	Topo	SeTopo
(500, 20, 0.2)	0.007	0.003	0.183	1.865	0.022	0.057	25.84	75.60	110.86	124.11
(500, 20, 0.5)	0.032	0.019	0.223	2.215	0.022	0.069	13.75	80.19	118.98	144.73
(500, 20, 0.9)	0.114	0.076	0.298	2.258	0.023	0.089	9.60	57.80	131.75	139.62
(500, 100, 0.2)	0.007	0.003	0.185	1.821	0.021	0.032	80.96	61.69	467.63	484.78
(500, 100, 0.5)	0.031	0.019	0.221	2.208	0.022	0.045	160.64	42.97	503.25	537.46
(500, 100, 0.9)	0.113	0.076	0.300	2.253	0.023	0.063	190.95	48.29	514.32	476.119
(2000, 20.95, 0.2)	0.254	0.100	13.32	147.16	0.272	0.456	969.08	890.67	1873.12	2017.78
(2000, 20.95, 0.5)	1.616	0.719	20.48	156.22	0.269	0.988	296.33	554.68	2201.72	2381.38
(2000, 20.95, 0.9)	5.954	4.026	25.79	157.00	0.330	1.450	71.061	564.66	2264.30	2474.97
(5000, 24.85, 0.2)	5.013	1.439	224.83	>3hr	1.883	3.386	>3hr	>3hr	>3hr	>3hr
(5000, 24.85, 0.5)	33.851	12.823	321.77	>3hr	2.066	9.252	3354.88	>3hr	>3hr	>3hr
(5000, 24.85, 0.9)	107.453	72.886	449.51	>3hr	2.124	15.132	255.93	>3hr	>3hr	>3hr

NOTE: The standard deviations are small and omitted.



(a) The ratio of $\mathrm{SR}(\mathbf{A})/\sqrt{N}$.

(b) Clustering coefficients.

Figure 11. Summary statistics of the 276 binary undirected networks.

are similar for different models, so we only show the times under one of them, the distance model.

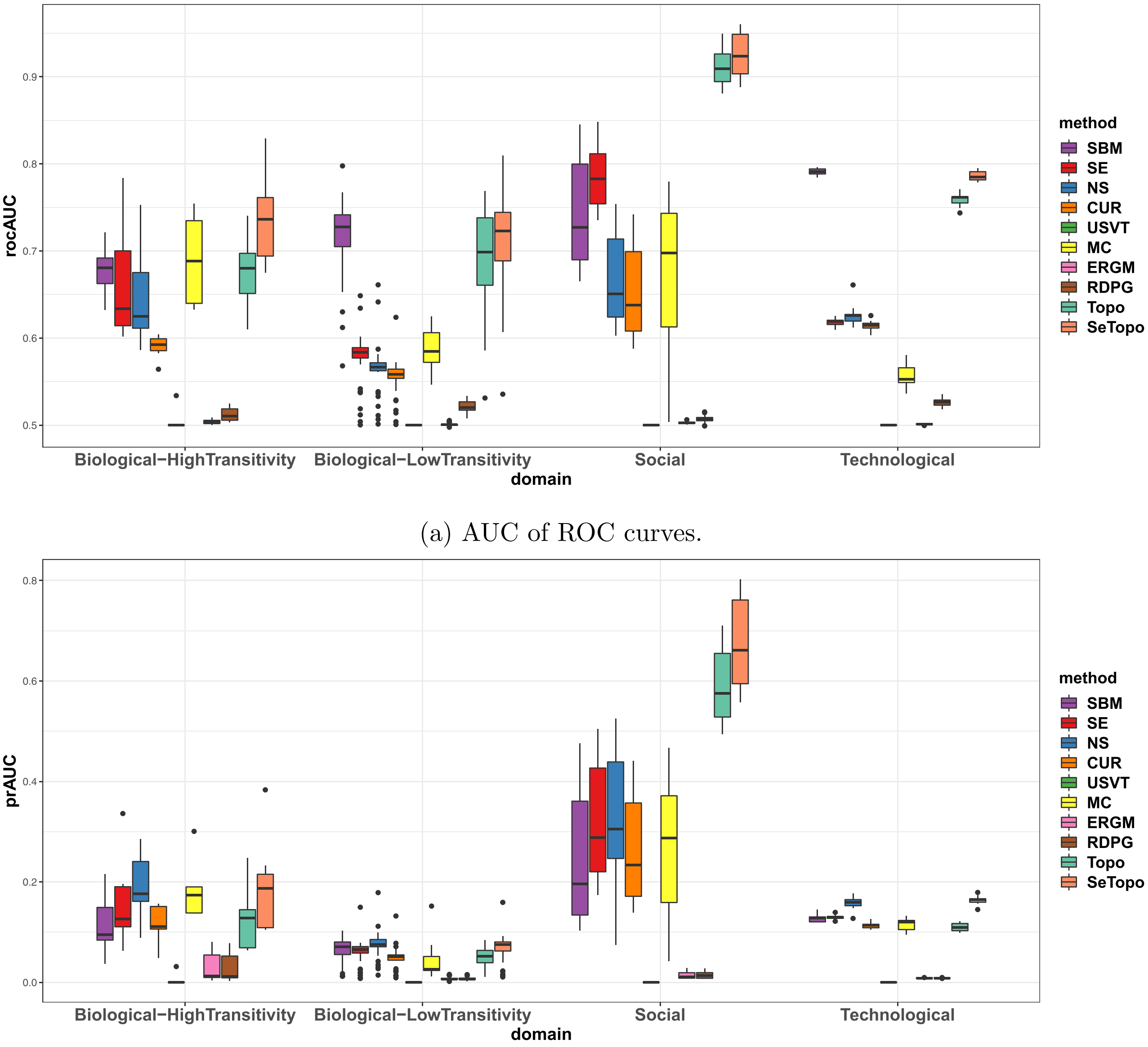
Table 5 shows results in seconds. Generally, it takes all methods longer as ρ grows, except for the ERGM. This is because higher values of ρ usually correspond to models of higher complexity chosen by cross-validation (e.g., a larger r in SE, and a larger $\mathbf{A}_{11}$ for computing the pseudo-inverse). The times for Topo, SeTopo, and the ERGM depend heavily on the density of the network. Overall, the CUR and USVT are the fastest, followed by the SE and the SBM. The NS is slower than these four by an order of magnitude and is barely feasible for $N = 5000$. The ERGM is even slower, unless $\rho = 0.9$. The RDPG, Topo, and SeTopo are even slower and cannot easily handle $N = 2000$ or more. These results do not take memory usage into account, but we have observed that SE, CUR, USVT, and the SBM are also the most memory-efficient among all methods considered.

4. Link Prediction on Real-World Networks

Here we report the performance of all the methods considered so far on a large set of real-world networks from Ghasemian et al. (2020). The dataset contains networks from different domains (biological, economic, informational, social, technological, and transportation). The topological differences between different domains have been investigated by Ikehara and Clauset (2017), and the dataset has been used as a benchmark in many link prediction evaluations (Mara, Lijffijt, and De Bie 2020; Kitsak, Voitalov, and Krioukov 2020; Yao et al. 2021; Li and Le 2021). A "no-free-lunch" phenomenon was observed by Ghasemian et al. (2020) and further verified in other follow-up work, showing that performance of any single method can vary significantly across domains, and there is no single dominant method for all domains, unless aggregation of multiple methods is used (Ghasemian et al. 2020; Yao et al. 2021; Li and Le 2021). Our goal here is to evaluate the SE method in real-world egocentric settings across different domains.

The dataset contains 276 undirected networks with unweighted edges, which is our focus: 43 biological, 106 economic, 108 social, 13 technological, and 6 transportation networks. Further, many of the methods we test, including ours, assume a low-rank structure and thus would not be appropriate to apply to networks that are not even approximately low lank. While we cannot check this assumption on the population probability matrix, we can assess its plausibility by looking at the stable rank of the adjacency matrix $\mathrm{SR}(\mathbf{A})$, as suggested in Section 2.2. Random matrix theory (Eldridge, Belkin, and Wang 2017) implies that if the true $\mathbf{P}$ has a low stable rank, the adjacency matrix $\mathbf{A}$ also has a low stable rank. The distribution of the ratio $\mathrm{SR}(\mathbf{A})/\sqrt{N}$ for these 276 networks shown in Figure 11(a). Intuitively, we treat $\mathrm{SR}(\mathbf{A})/\sqrt{N} < 2$ as a "low-rank" criterion for our focus. It turns out that all transportation networks and almost all economic networks (except for 3) are far from low rank. Therefore, we remove these two domains. This procedure results in the final evaluation set of 163 networks, of which 107 are social networks, 43 are biological networks, and 13 are technological networks.

These networks also tend to have different levels of transitivity, as shown in Figure 11(b). Social networks are more transitive with significantly larger clustering coefficients than networks from other domains, as previously observed by network scien-



(a) AUC of ROC curves.

(b) AUC of PR curves.

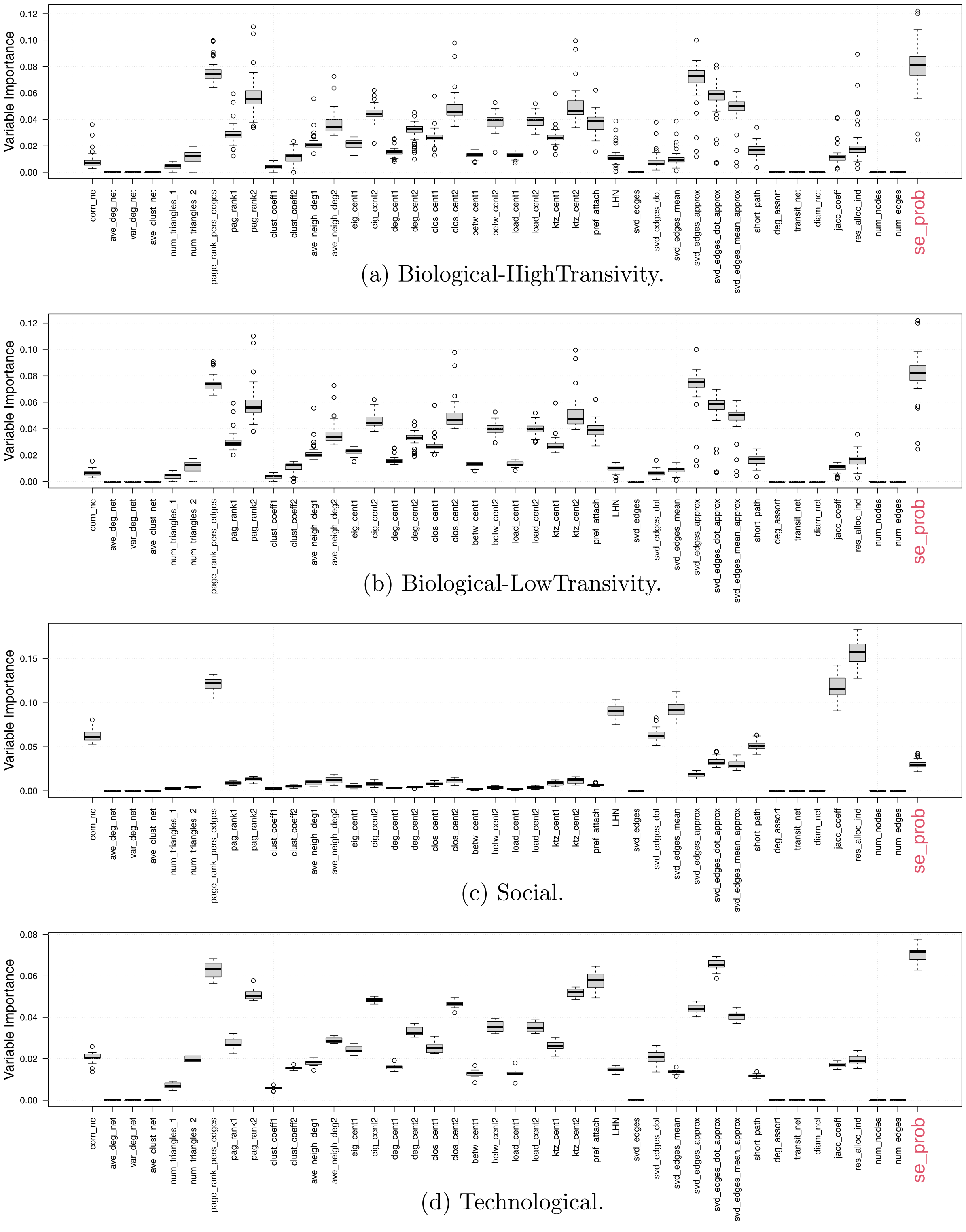
Figure 12. Link prediction performance on the 163 "approximately low-rank" networks with a low sampling fraction, $\rho = n/N = 0.2$.

tists (Holland and Leinhardt 1971; Watts and Strogatz 1998). Biological networks exhibit a wide range of transitivity, and thus, we divided them into "high transitivity" (global clustering coefficient > 0.1) and "low transitivity" (the rest). As a reminder, these networks are not a random sample of real-world approximately low-rank networks: for instance, all 13 technological networks are from the same study of BGP traffic among autonomous systems (ASs) (Ghasemian et al. 2020), and they do not differ much. Moreover, egocentric sampling is more likely to arise in social networks in practice than in other domains.

For evaluation purposes, we randomly sample rows with probability ρ from a given network to create an egocentrically sampled network, and then predict unobserved entries, repeating this 50 times for each network. As before, we evaluate the methods by (a) individual link prediction accuracy, measured by the average AUC of the ROC curves (rocAUC) and the average AUC of the PR curves (prAUC), reported in Section 4.1; and (b) global statistics prediction, measured by the average relative prediction error on each network, reported in Section 4.2.

4.1. Link Prediction Accuracy

Figure 12(a) shows boxplots of the rocAUC of all the methods with sampling faction $\rho = 0.2$, and Figure 12(b) shows the prAUC in the same setting. As Ghasemian et al. (2020) showed, methods fitting a single model perform very differently in different domains. In fact, this is also true across different performance metrics. Overall, among single model methods, the SE, SBM, and NS are the best on both ROC and PR measures. For social networks, SE is the best on ROC and marginally inferior to NS on PR. The SBM is the clear winner in ROC for technological networks, while NS is better in PR curves. Recall that the technological networks are very similar, so variability for that domain mainly reflects the randomness of data splitting. For



(a) Biological-HighTransivity.

(b) Biological-LowTransivity.

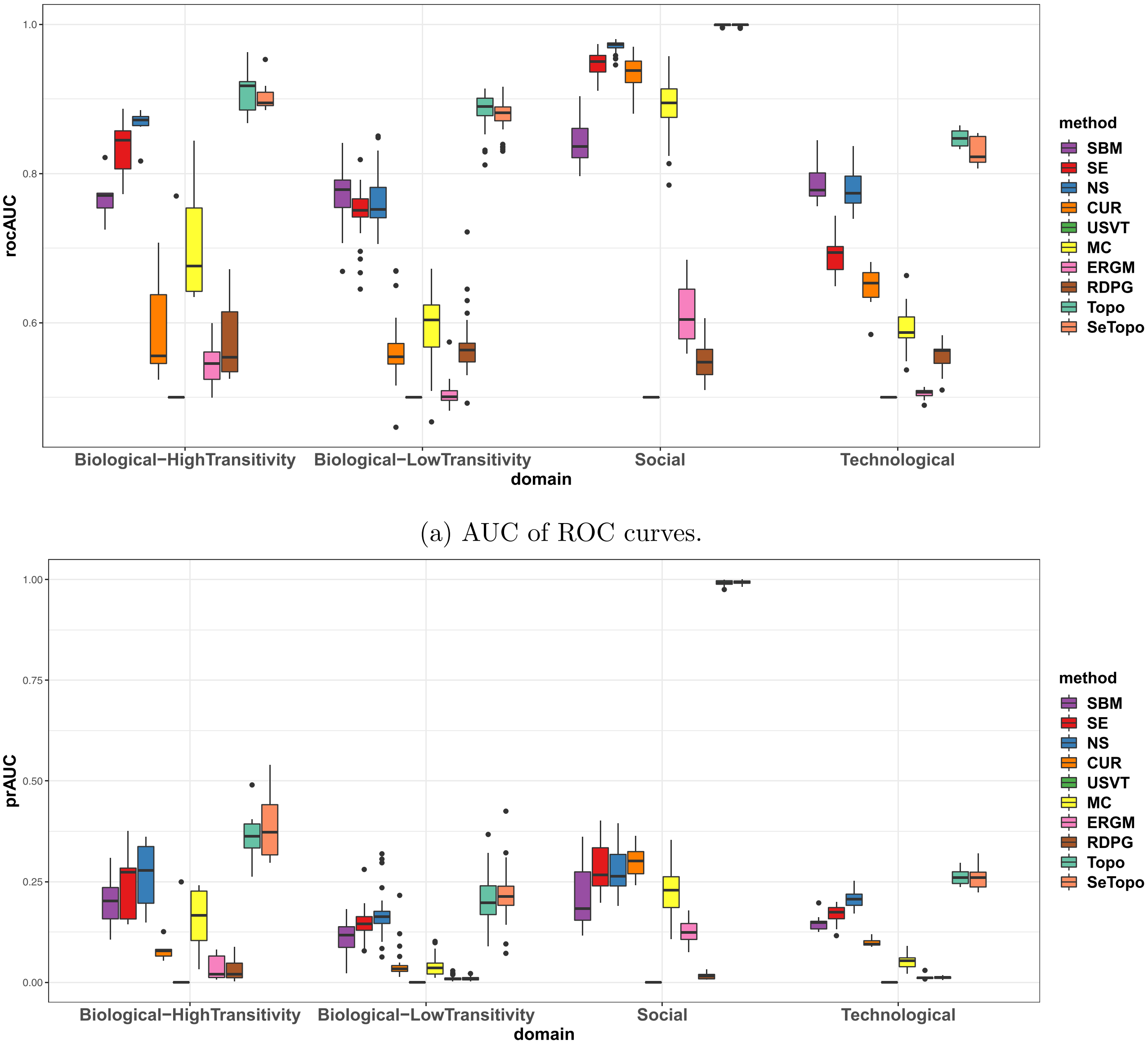
(c) Social.

(d) Technological.

Figure 13. Variable importance scores of all 42 features in SeTopo on the 163 “approximately low-rank” networks with a low sampling fraction, $\rho = n/N = 0.2$.

high-transitivity biological networks, SE, SBM, and NS are fairly similar, while for low-transitivity biological networks, SBM is the winner.

Ensemble methods outperform single models on link prediction accuracy uniformly across all domains, which is to be expected for a prediction task. As shown in Figure 12(a)–(b), SeTopo and Topo are much better than the single models on social networks and similar to the best single model on biological and technological networks. The SeTopo is uniformly better than Topo, reflecting the importance of the additional information provided by the SE method. This is also reflected in the variable importance scores from the random forest shown in Figure 13 for the 42 predictors on the 163 networks, where the SE probability labeled “se_prob.” Variable importance scores were proposed by (Breiman 2001) to measure the predictive contributions of different variables, and while they are not direct



(a) AUC of ROC curves.

(b) AUC of PR curves.

Figure 14. Link prediction performance on the 163 networks with $\rho = n/N = 0.9$.

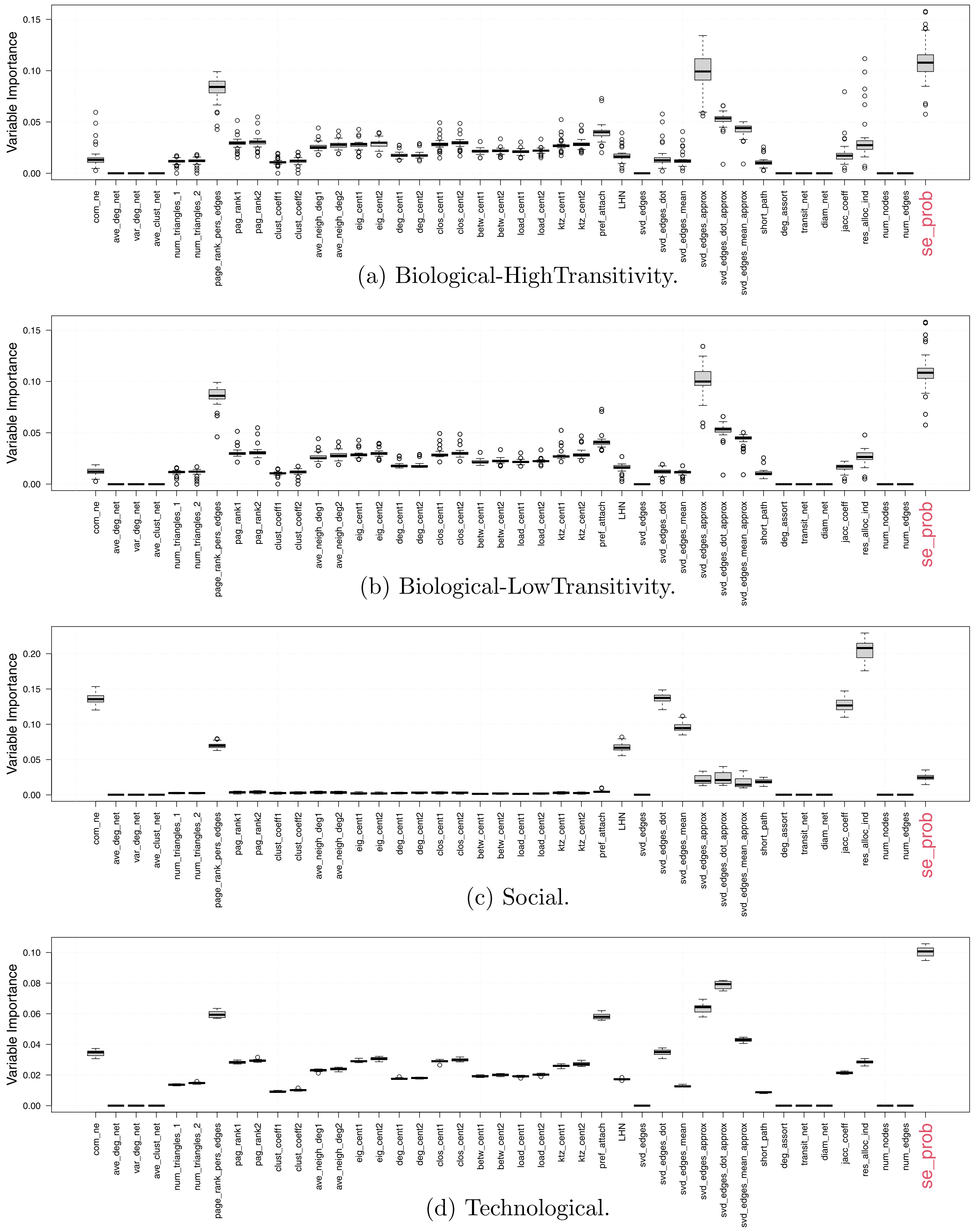
evidence of predictive power (Hastie et al. 2009), a high score indicates that the variable is frequently used by the forest. The SE probability is the most important variable on biological and technological networks, and a fairly important one on social networks.

Next, we evaluate the scenario with a high sampling fraction, with $\rho = 0.9$, with results shown in Figure 14. A large sampling fraction significantly helps NS, which becomes the best single model except on technological networks. The SE method remains competitive for both social networks and highly transitive biological networks, and is comparable to SBM on low transitivity biological networks. The CUR and MC are more effective on social networks but not competitive in other domains. The SBM remains the best for technological networks.

For the ensemble methods, both Topo and SeTopo give better predictions than any one of the individual models across all domains and, in particular, deliver almost perfect predictions on social networks. There is not much difference in ROC and PR curves between Topo and SeTopo in this scenario, likely because the signal is very strong for both. The variable importance score for the SE probability remains high, however, as shown in Figure 15, at least for biological and technological networks.

4.2. Global Statistics Prediction

Similarly to the simulations, we evaluate the methods for the task of predicting global network statistics, with results shown in Figure 16 for the low sampling fraction $\rho = 0.2$ and in Figure 17 for the high sampling fraction $\rho = 0.9$. For these tasks, Topo and SeTopo are no longer applicable, but we can still use the empirical estimators. In the low sampling scenario, the empirical estimator is the best for density across all domains, but it cannot predict the global clustering coefficient and eigencentrality effectively. The SE does worse on predicting the clustering coefficient



(a) Biological-HighTransitivity.

(b) Biological-LowTransitivity.

(c) Social.

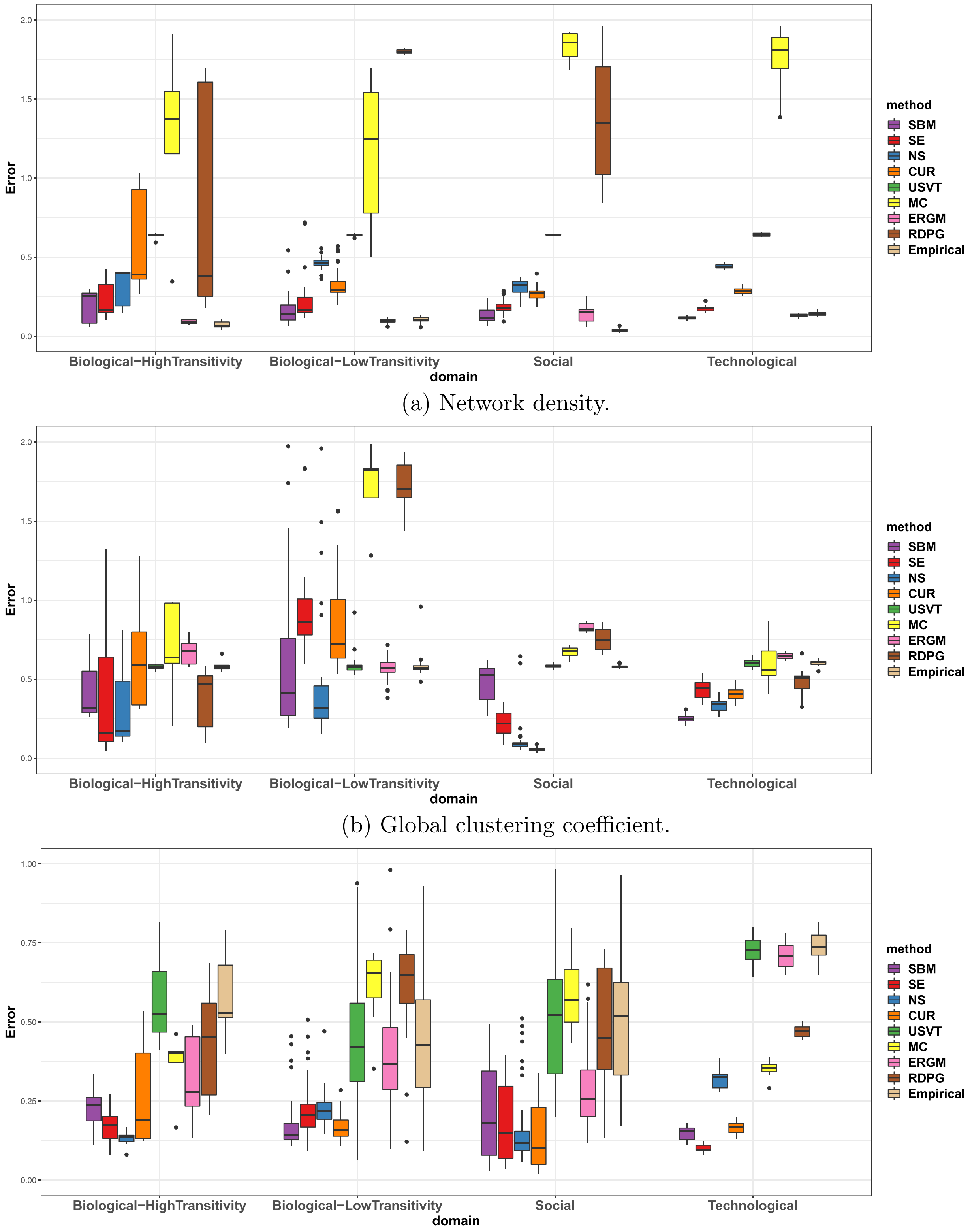
(d) Technological.

Figure 15. Variable importance scores of all 42 features in SeTopo on the 163 networks with a high sampling fraction, $\rho = n/N = 0.9$.

on low-transitivity biological networks, but remains competitive in other domains on all three metrics. The NS estimator is less effective in predicting the density but works well for the other two metrics most of the time. For the high sampling fraction, the SE is the best or close to the best in all three metrics across all domains.

4.2.1. Summary of Results on the Real-World Networks

The choice of method in practice depends on the task at hand, network properties that may correlate with the domain, and computational constraints. For link prediction, our method (SE) has the most significant advantage for small sampling fractions and is the best for social networks in that setting; for high sampling fractions, the NS method performs better due to its generality. The SE is also more accurate on networks with higher transitivity, including social networks and some biological networks. The ensemble methods based on feature stacking generally outperform single models, and more so in real-world networks. Including the SE predictions in the ensemble improves the results, especially for small sampling fractions. For global



(a) Network density.

(b) Global clustering coefficient.

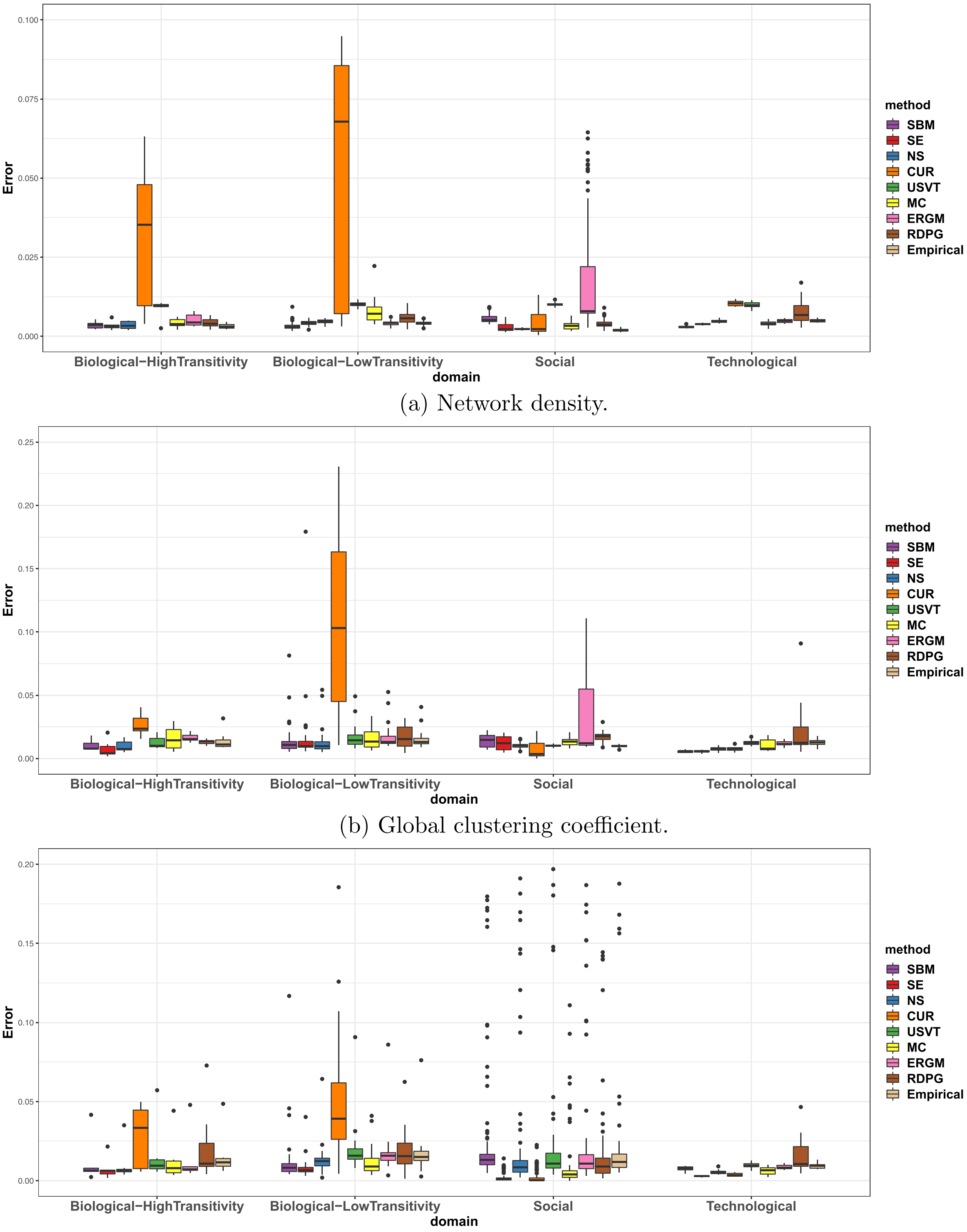
(c) Eigencentrality.

Figure 16. The prediction error of the global statistics on the 163 networks with $\rho = n/N = 0.2$.

statistics prediction where ensemble methods are not applicable, the SE delivers competitive predictions in most situations, and NS achieves similar results but at a much higher computational cost. The computational costs of the ensemble methods are even higher.

5. Discussion

The subspace estimation link prediction algorithm we proposed is, to the best of our knowledge, the first algorithm designed specifically for egocentrically sampled networks and not generic link prediction. The comparison with results from standard



(a) Network density.

(b) Global clustering coefficient.

(c) Eigencentrality.

Figure 17. The prediction error of the global statistics on the 163 networks with $\rho = n/N = 0.9$.

matrix completion and link prediciton methods shows that egocentrically sampled networks deserve their own careful treatment, given how often they are encountered in practice, and should not be treated as generic matrices with missing entries. Our method is computationally efficient and especially powerful in the most challenging scenarios of low sampling rates and sparse networks. It relies on assumptions more general than block models but more parsimoniuous than simply low rank, which may help explain its good performance with small sampling fractions. In practice, the plausibility of these assumptions can be assessed by cross-validation (Chen and Lei 2018; Li, Levina, and Zhu 2020b), and the best prediction method is chosen accordingly.

Many challenges in modeling realistic network data collection remain open. We have developed a method for egocentric sampling, but even within that framework, practical complications may occur—for instance, a survey of a social network may limit how many friends a subject can name, or the subject can give a biased and incomplete representation of their true connections. For these scenarios, our approach provides the first step that can be followed by an additional step modeling subject's reporting preferences, as in, for example, Li, Levina, and Zhu (2020a), who modeled friendship reporting preferences based on communities, and new approaches may be needed for more complicated settings, such as snowball sampling and others (Rohe 2019).

Another limitation of our method and matrix completion methods in general is the assumption of missing at random. More complex hierarchical models for missing data inference are available (Little and Rubin 2019), but one would need to balance the tradeoffs between model complexity, computational efficiency, and sample size requirements. Traditional missing data procedures for inference and bias correction tend to require a large sample size and complicated models which are frequently not scalable. Although not designed for inference and bias correction, matrix completion methods can work with a much smaller sampling fraction and are computationally scalable. In our view, finding a nuanced compromise between these competing considerations will be an important avenue for future work.

Supplementary Materials

The following are included in the supplementary materials available online.

Appendix (`Appendix.pdf`): References about empirical studies involving egocentric sampling of networks, proof of Theorem 2.1 and additional simulation results.

Code (`Code.zip`): Code for experiments of the article. Each subfolder in the file has its own `Readme.txt` about the files and examples.

Acknowledgments

We would like to thank the Editor, Associate Editor, and the referees for their constructive and helpful feedback.

Disclosure Statement

The authors declare that they have no financial or nonfinancial interests that relate to the research described in this article.

Funding

T. Li's research is supported in part by NSF DMS grant 2015298 and the 3Caverliers Award at the University of Virginia. E. Levina's research is supported in part by NSF DMS grants 1521551 and 1916222. J. Zhu's research is supported in part by NSF DMS grant 1407698 and 1821243.

References

Ali, M. M., and Dwyer, D. S. (2009), "Estimating Peer Effects in Adolescent Smoking Behavior: A Longitudinal Analysis," *Journal of Adolescent Health*, 45, 402–408. [4]

Almquist, Z. W. (2012), "Random Errors in Egocentric Networks," *Social Networks*, 34, 493–505. [1]

Anil, A., Kumar, D., Sharma, S., Singha, R., Sarmah, R., Bhattacharya, N., and Singh, S. R. (2015), "Link Prediction Using Social Network Analysis Over Heterogeneous Terrorist Network," in *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pp. 267–272. IEEE. [1]

Athreya, A., Tang, M., Park, Y., and Priebe, C. E. (2018), "On Estimation and Inference in Latent Structure Random Graphs," arXiv preprint arXiv:1806.01401. [5,6]

Bandiera, O., and Rasul, I. (2006), "Social Networks and Technology Adoption in Northern Mozambique," *The Economic Journal*, 116, 869–902. [4]

Banerjee, A., Chandrasekhar, A. G., Duflo, E., and Jackson, M. O. (2013), "The Diffusion of Microfinance," *Science*, 341, 1236498. [1,4]

Batchelder, W. H., and Romney, A. K. (1988), "Test Theory Without an Answer Key," *Psychometrika*, 53, 71–92. [1]

Breiman, L. (2001), "Random Forests," *Machine Learning*, 45, 5–32. [18]

Butts, C. T. (2003), "Network Inference, Error, and Informant (in) Accuracy: A Bayesian Approach," *Social Networks*, 25, 103–140. [1,2]

Candès, E. J., and Plan, Y. (2010), "Matrix Completion with Noise," in *Proceedings of the IEEE*. [2]

Candès, E. J., and Tao, T. (2010), "The Power of Convex Relaxation: Near-Optimal Matrix Completion," *IEEE Transactions on Information Theory*, 56, 2053–2080. [2,4]

Chandrasekhar, A., and Lewis, R. (2011), "Econometrics of Sampled Networks," Unpublished manuscript, MIT.[422]. [1,2]

Chatterjee, S. (2015), "Matrix Estimation by Universal Singular Value Thresholding," *The Annals of Statistics*, 43, 177–214. [5]

Chatterjee, S., and Diaconis, P. (2013), "Estimating and Understanding Exponential Random Graph Models," *The Annals of Statistics*, 41, 2428–2461. [2]

Chen, K., and Lei, J. (2018), "Network Cross-validation for Determining the Number of Communities in Network Data," *Journal of the American Statistical Association*, 113, 241–251. [4,5,22]

Chi, E. C., and Li, T. (2019), "Matrix Completion from a Computational Statistics Perspective," *Wiley Interdisciplinary Reviews: Computational Statistics*, 11, e1469. [2]

Chung, F., and Lu, L. (2002), "The Average Distances in Random Graphs with Given Expected Degrees," *Proceedings of the National Academy of Sciences*, 99, 15879–15882. [2]

Cichocki, A., and Amari, S.-i. (2002), *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*, Chichester: Wiley. [4]

Conley, T. G., and Udry, C. R. (2010), "Learning About a New Technology: Pineapple in Ghana," *American Economic Review*, 100, 35–69. [4]

Crane, H., and Dempsey, W. (2018), "Edge Exchangeable Models for Interaction Networks," *Journal of the American Statistical Association*, 113, 1311–1326. [2]

Davenport, M. A., Plan, Y., Berg, E. V. D., Wootters, M., van den Berg, E., and Wootters, M. (2014), "1-bit Matrix Completion," *Information and Inference*, 3, 189–223. [2]

Drineas, P., Kannan, R., and Mahoney, M. (2006), "Fast Monte Carlo Algorithms for Matrices III: Computing a Compressed Approximate Matrix Decomposition," *SIAM Journal on Computing*, 36, 184–206. [3]

Drineas, P., Mahoney, M., and Muthukrishnan, S. (2008), "Relative-Error CUR Matrix Decompositions," *SIAM Journal on Matrix Analysis and Applications*, 30, 844–881. [3,4]

Eldridge, J., Belkin, M., and Wang, Y. (2017), "Unperturbed: Spectral Analysis Beyond Davis-Kahan," arXiv preprint arXiv:1706.06516. [16]

Fafchamps, M., and Lund, S. (2003), "Risk-Sharing Networks in Rural Philippines," *Journal of Development Economics*, 71, 261–287. [4]

Frank, O., and Strauss, D. (1986), "Markov Graphs," *Journal of the American Statistical Association*, 81, 832–842. [2]

Freeman, L. C. (1982), "Centered Graphs and the Structure of Ego Networks," *Mathematical Social Sciences*, 3, 291–304. [1]

Gao, M., Chen, L., Li, B., and Liu, W. (2018), "A Link Prediction Algorithm based on Low-Rank Matrix Completion," *Applied Intelligence*, 48, 4531–4550. [2]

Ghasemian, A., Hosseinmardi, H., Galstyan, A., Airoldi, E. M., and Clauset, A. (2020), "Stacking Models for Nearly Optimal Link Prediction in Complex Networks," *Proceedings of the National Academy of Sciences*, 117, 23393–23400. [5,10,11,16,17]

Golub, G. H., and Van Loan, C. F. (1989), *Matrix Computations* (2nd ed.), Baltimore, MD: The John Hopkins University Press. [4]

Handcock, M. S., and Gile, K. J. (2010), "Modeling Social Networks from Sampled Data," *The Annals of Applied Statistics*, 4, 5–25. [1,5]

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., Krivitsky, P. N., and Morris, M. (2019), *ergm: Fit, Simulate and Diagnose Exponential-Family Models for Networks*. The Statnet Project. R package version 3.10.4. [5]

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Vol. 2), New York: Springer. [19]

Hoff, P., Fosdick, B., and Volfovsky, A. (2020), *amen: Additive and Multiplicative Effects Models for Networks and Relational Data*, R package version 1.4.4. [5]

Hoff, P. D. (2009), "Multiplicative Latent Factor Models for Description and Prediction of Social Networks," *Computational and Mathematical Organization Theory*, 15, 261–272. [5]

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002), "Latent Space Approaches to Social Network Analysis," *Journal of the American Statistical Association*, 97, 1090–1098. [5]

Holland, P. P. W., and Leinhardt, S. (1981), "An Exponential Family of Probability Distributions for Directed Graphs," *Journal of the American Statistical Association*, 76, 33–50. [4]

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983), "Stochastic Blockmodels: First Steps," *Social Networks*, 5, 109–137. [5]

Holland, P. W., and Leinhardt, S. (1971), "Transitivity in Structural Models of Small Groups," *Comparative Group Studies*, 2, 107–124. [17]

Hunter, D. R. (2007), "Curved Exponential Family Models for Social Networks," *Social Networks*, 29, 216–230. [5]

Hurlbert, J., Beggs, J., and Haines, V. (2005), "Bridges Over Troubled Waters: What are the Optimal Networks for Katrina's Victims," in *Online Forum and Essays]–Social Science Research Council*. [1]

Ikehara, K., and Clauset, A. (2017), "Characterizing the Structural Diversity of Complex Networks Across Domains," arXiv preprint arXiv:1710.11304. [16]

Keshavan, R., Montanari, A., and Oh, S. (2010), "Matrix Completion from Noisy Entries," *Journal of Machine Learning Research*, 11, 2057–2078. [2]

Kitsak, M., Voitalov, I., and Krioukov, D. (2020), "Link Prediction with Hyperbolic Geometry," *Physical Review Research*, 2, 043113. [16]

Krivitsky, P. N., and Morris, M. (2017), "Inference for Social Network Models from Egocentrically Sampled Data, with Application to Understanding Persistent Racial Disparities in HIV Prevalence in the US," *The Annals of Applied Statistics*, 11, 427–455. [1]

Laumann, E. O., Gagnon, J. H., Michael, R. T., and Michaels, S. (1995), *National Health and Social Life Survey*, Chicago, IL: University of Chicago and National Opinion Research Center. [1]

Lauritzen, S., Rinaldo, A., and Sadeghi, K. (2018), "Random Networks, Graphical Models and Exchangeability," *Journal of the Royal Statistical Society*, Series B, 80, 481–508. [2]

Lei, J., and Rinaldo, A. (2015), "Consistency of Spectral Clustering in Stochastic Block Models," *The Annals of Statistics*, 43, 215–237. [4]

Li, T., and Le, C. M. (2021), "Network Estimation by Mixing: Adaptivity and More," arXiv preprint arXiv:2106.02803. [5,16]

Li, T., Levina, E., and Zhu, J. (2020a), "Community Models for Networks Observed through Edge Nominations," arXiv e-prints, page arXiv:2008.03652. [23]

——— (2020b), "Network Cross-Validation by Edge Sampling," *Biometrika*, 107, 257–276. [2,4,22]

Liben-Nowell, D., and Kleinberg, J. (2007), "The Link-Prediction Problem for Social Networks," *Journal of the American Society for Information Science and Technology*, 58, 1019–1031. [1,2]

Lichtenwalter, R. N., Lussier, J. T., and Chawla, N. V. (2010), "New Perspectives and Methods in Link Prediction," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 243–252. [2]

Lin, Z., Chen, M., and Ma, Y. (2010), "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices," arXiv preprint arXiv:1009.5055. [5]

Little, R. J., and Rubin, D. B. (2019), *Statistical Analysis with Missing Data* (Vol. 793), Hoboken, NJ: Wiley. [23]

Lü, L., and Zhou, T. (2011), "Link Prediction in Complex Networks: A Survey," *Physica A: Statistical Mechanics and its Applications*, 390, 1150–1170. [1]

Mahoney, M. W., and Drineas, P. (2009), "CUR Matrix Decompositions for Improved Data Analysis," *Proceedings of the National Academy of Sciences of the United States of America*, 106, 697–702. [3,4]

Manresa, E. (2013), "Estimating the Structure of Social Interactions using Panel Data," *Unpublished Manuscript. CEMFI, Madrid*. [1]

Mara, A. C., Lijffijt, J., and De Bie, T. (2020), "Benchmarking Network Embedding Models for Link Prediction: Are We Making Progress?" in *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 138–147, IEEE. [16]

Mcauley, J., and Leskovec, J. (2012), "Learning to Discover Social Circles in Ego Networks," in *Advances in Neural Information Processing Systems* (Vol. 25), pp. 539–547. [1]

Morris, M., Kurth, A. E., Hamilton, D. T., Moody, J., and Wakefield, S. (2009), "Concurrent Partnerships and HIV Prevalence Disparities by Race: Linking Science and Public Health Practice," *American Journal of Public Health*, 99, 1023–1031. [1]

Newman, M. (2018), "Network Structure from Rich but Noisy Data," *Nature Physics*, 14, 542–545. [1,2]

Newman, M. E. J. (2003), "Ego-Centered Networks and the Ripple Effect," *Social Networks*, 25, 83–95. [1]

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999), "The Pagerank Citation Ranking: Bringing Order to the Web," Technical report, Stanford InfoLab. [6]

Pech, R., Hao, D., Pan, L., Cheng, H., and Zhou, T. (2017), "Link Prediction via Matrix Completion," *EPL (Europhysics Letters)*, 117, 38002. [2]

Peixoto, T. P. (2015), "Inferring the Mesoscale Structure of Layered, Edge-Valued, and Time-Varying Networks," *Physical Review E*, 92, 042807. [5]

——— (2018), "Reconstructing Networks with Unknown and Heterogeneous Errors," *Physical Review X*, 8, 041011. [5]

Robins, G., Snijders, T., Wang, P., Handcock, M., and Pattison, P. (2007), "Recent Developments in Exponential Random Graph (p*) Models for Social Networks," *Social Networks*, 29, 192–215. [2]

Rohe, K. (2019), "A Critical Threshold for Design Effects in Network Sampling," *The Annals of Statistics*, 47, 556–582. [23]

Rohe, K., Chatterjee, S., and Yu, B. (2011), "Spectral Clustering and the High-Dimensional Stochastic Blockmodel," *The Annals of Statistics*, 39, 1878–1915. [4]

Romney, A. K., Weller, S. C., and Batchelder, W. H. (1986), "Culture as Consensus: A Theory of Culture and Informant Accuracy," *American Anthropologist*, 88, 313–338. [1]

Rubin, D. B. (1976), "Inference and Missing Data," *Biometrika*, 63, 581–592. [1]

Shalizi, C. R., and Rinaldo, A. (2013), "Consistency Under Sampling of Exponential Random Graph Models," *The Annals of Statistics*, 41, 508–535. [2]

Vershynin, R. (2018), *High-Dimensional Probability: An Introduction with Applications in Data Science* (Vol. 47), Cambridge: Cambridge University Press. [4]

Watts, D. J., and Strogatz, S. H. (1998), "Collective Dynamics of "Small-World" Networks," *Nature*, 393, 440–442. [17]

Yao, Y., Pirš, G., Vehtari, A., and Gelman, A. (2021), "Bayesian Hierarchical Stacking," arXiv preprint arXiv:2101.08954. [5,16]

Young, S. J., and Scheinerman, E. R. (2007), "Random Dot Product Graph Models for Social Networks," in *International Workshop on Algorithms and Models for the Web-Graph*, pp. 138–149, Springer. [5]

Zhang, Y., Levina, E., and Zhu, J. (2017), "Estimating Network Edge Probabilities by Neighbourhood Smoothing," *Biometrika*, 104, 771–783. [5]

Zhao, Y., Wu, Y.-J., Levina, E., and Zhu, J. (2017), "Link Prediction for Partially Observed Networks," *Journal of Computational and Graphical Statistics*, 26, 725–733. [2]