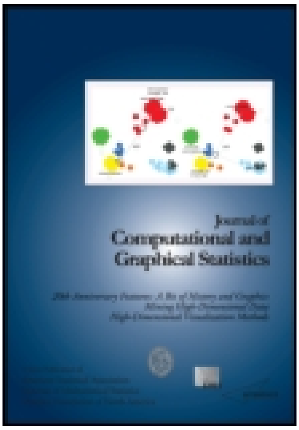

Journal of Computational and Graphical Statistics

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ucgs20

Fast Inference for the Latent Space Network Model Using a Case-Control Approximate Likelihood

Adrian E. Raftery [a], Xiaoyue Niu [b], Peter D. Hoff [a] & Ka Yee Yeung [c]

[a] Department of Statistics, University of Washington, Box 354322, Seattle, WA, 98195-4322

[b] Department of Statistics, Penn State University, University Park, PA, 16802

[c] Department of Microbiology, University of Washington, Box 358070, Seattle, WA, 98195-8070

Accepted author version posted online: 04 Apr 2012.Version of record first published: 14 Dec 2012.

To cite this article: Adrian E. Raftery , Xiaoyue Niu , Peter D. Hoff & Ka Yee Yeung (2012): Fast Inference for the Latent Space Network Model Using a Case-Control Approximate Likelihood, Journal of Computational and Graphical Statistics, 21:4, 901-919

To link to this article: http://dx.doi.org/10.1080/10618600.2012.679240

Fast Inference for the Latent Space Network Model Using a Case-Control Approximate Likelihood

Adrian E. RAFTERY, Xiaoyue NIU, Peter D. HOFF, and Ka Yee YEUNG

Network models are widely used in social sciences and genome sciences. The latent space model proposed by Hoff et al. (2002), and extended by Handcock et al. (2007) to incorporate clustering, provides a visually interpretable model-based spatial representation of relational data and takes account of several intrinsic network properties. Due to the structure of the likelihood function of the latent space model, the computational cost is of order $O(N^2)$, where N is the number of nodes. This makes it infeasible for large networks. In this article, we propose an approximation of the log-likelihood function. We adapt the case-control idea from epidemiology and construct a case-control log-likelihood, which is an unbiased estimator of the log-full likelihood. Replacing the full likelihood by the case-control likelihood in the Markov chain Monte Carlo estimation of the latent space model reduces the computational time from $O(N^2)$ to $O(N)$, making it feasible for large networks. We evaluate its performance using simulated and real data. We fit the model to a large protein–protein interaction data using the case-control likelihood and use the model fitted link probabilities to identify false positive links. Supplemental materials are available online.

Key Words: Clustering; Genome science; Graph; Markov chain Monte Carlo; Protein–protein interaction; Social network.

1. INTRODUCTION

Networks consist of the links between members of a set of actors that are connected by a specific kind of relationship. Networks have many applications in social science, political science, and, recently, in genome science. Examples include friendship among students in a high school, international trade and conflicts, and protein–protein interactions (PPIs). Statistical models are widely used to represent such data. Statistical network models include exponential random graph models (Frank and Strauss 1986; Wasserman and Pattison

Adrian E. Raftery is Professor of Statistics and Sociology, Department of Statistics, University of Washington, Box 354322, Seattle, WA 98195-4322 (E-mail: *raftery@u.washington.edu*). Xiaoyue Niu is Assistant Professor, Department of Statistics, Penn State University, University Park, PA 16802. Peter Hoff is Professor, Department of Statistics, University of Washington, Box 354322, Seattle, WA 98195-4322. Ka Yee Yeung is Research Associate Professor, Department of Microbiology, University of Washington, Box 358070, Seattle, WA 98195-8070.

Journal of Computational and Graphical Statistics, Volume 21, Number 4, Pages 901–919
DOI: 10.1080/10618600.2012.679240

1996), the homogeneous monadic Markov model (Frank and Strauss 1986), the stochastic blockmodel (Wang and Wong 1987), the latent class membership model (Nowicki and Snijders 2001), and the mixed membership stochastic blockmodel (Airoldi et al. 2008).

Hoff, Raftery, and Handcock (2002) proposed a stochastic model for representing network data based on the concept of "social space" (McFarland and Brown 1973). The key idea behind this model is that the observed relations are determined by the unobserved latent characteristics of the actors. These latent characteristics are represented by the actors' unobserved latent positions in a Euclidean space. The probability of a link between two actors is determined by the distance between their latent positions and, given the latent positions of the actors, the relational ties are independent.

The data to be modeled are as follows. The network consists of N actors (or nodes), and $y_{ij} = 1$ if there is a link from actor i to actor j, while $y_{ij} = 0$ otherwise, for $i, j = 1, \ldots, N$. We denote by x_{ij} the vector of p covariates pertaining to the (i, j) dyad. This can include covariates relating to actors i and j individually as well as to the dyad as such. For example, x_{ij} could include the race of actor i, the race of actor j, and whether or not they are of the same race. Note that the link from i to j is not necessarily exchangeable with the link from j to i, and x_{ij} can be different from x_{ji}. We denote by Y the $N \times N$ matrix with elements y_{ij}, sometimes called the sociomatrix. The ith row of Y identifies the links from actor i to the other actors. We denote by X the $N \times N \times p$ array containing the vectors x_{ij}.

The latent space model is a probability model for the network based on assuming that each actor i occupies an unobserved or latent position z_i in a k-dimensional Euclidean "social space." We denote by Z the $N \times k$ matrix whose ith row consists of the k-vector z_i. The model is defined as follows:

$$\Pr(Y|Z, X, \boldsymbol{\theta}) = \prod_{i \neq j} P(y_{ij}|z_i, z_j, x_{ij}, \alpha, \beta), \tag{1}$$

where α is a scalar parameter and β is a p-vector of parameters. Both the parameters α and β and the latent positions Z are to be estimated. The probability of a link between actors i and j, $\Pr(y_{ij}|z_i, z_j, x_{ij}, \alpha, \beta)$, can be conveniently modeled by a logistic regression model, namely

$$\eta_{ij} = \log \text{odds}(y_{ij} = 1|z_i, z_j, x_{ij}, \alpha, \beta) = \alpha + \beta' x_{ij} - |z_i - z_j|. \tag{2}$$

Note that the use of the absolute value in determining distance in the latent space is a choice in the formulation of such models, and Hoff, Raftery, and Handcock (2002) also proposed a model that used an inner product distance. The methodology proposed here would be applicable also with other choices of distance.

To complete the model specification, we assume that the z_i's themselves are independent draws from a distribution. Hoff, Raftery, and Handcock (2002) assumed that this was a spherical multivariate normal distribution, so that

$$z_1, \ldots, z_N \overset{\text{iid}}{\sim} \text{MVN}_k\big(0, \sigma_z^2 I_k\big). \tag{3}$$

Thus, the latent space network model is a hierarchical model for the y_{ij}, where the distribution of the y_{ij} depends on the z_i, and the z_i in turn have a distribution specified by σ_z^2.

The latent space model provides a visual and interpretable model-based spatial representation of relational data and takes account of several intrinsic network properties, including transitivity and homophily on both unobserved and observed attributes. Handcock, Raftery, and Tantrum (2007) extended the latent space model to allow clustering of the subjects in the network, beyond what would be implied by simple transitivity. Instead of assuming that the z_i come from a multivariate normal distribution, they assumed that the z_i are from a mixture of multivariate normal distributions, namely

$$z_i \sim \sum_{g=1}^{G} \lambda_g \mathrm{MVN}_k\left(\mu_g, \sigma_g^2 I_k\right),$$

where λ_g is the probability that an actor belongs to the gth group, $\lambda_g \geq 0$ and $\sum_{g=1}^{G} \lambda_g = 1$.

The log-likelihood of α, β, and the z_i's for the latent space model is as follows:

$$\log \Pr(Y|\eta) = \sum_{i \neq j} \{\eta_{ij} y_{ij} - \log(1 + e^{\eta_{ij}})\}, \tag{4}$$

where $\eta_{i,j} = \alpha + \beta' x_{i,j} - |z_i - z_j|$. Calculation of this log-likelihood involves a sum over $N(N-1)$ terms, which is of the order of $O(N^2)$ terms. When the relationship is undirected, the number of terms is $\binom{N}{2} = \frac{1}{2}N(N-1)$, which is still $O(N^2)$.

To estimate the regression coefficients α, β, the latent positions Z, and their variance σ_z^2, it is standard to apply a Bayesian approach by constructing a Markov chain with stationary distributions equal to the posterior distribution of the parameters (Hoff, Raftery, and Handcock 2002; Handcock, Raftery, and Tantrum 2007; Krivitsky et al. 2009). The algorithm proceeds with Metropolis updates for α, β, and the z_i's, generating random proposal values of these parameters from symmetric distributions centered around their current values, and then accepting these proposals with the appropriate probability. The σ_z^2 parameter is updated with a Gibbs step, generated by sampling a new value from the full conditional distribution:

- For each i in a random order, propose a value z_i^* from the distribution $\phi_1(\cdot)$ and accept with probability $\frac{p(Y|z^*,X,\beta)\phi(z_i^*)}{p(Y|z,X,\beta)\phi(z_i)}$.
- Propose α^*, β^* from the distribution $\phi_2(\cdot,\cdot)$ and accept with probability $\frac{p(Y|z,X,\alpha^*,\beta^*)\phi(\alpha^*,\beta^*)}{p(Y|z,X,\alpha,\beta)\phi(\alpha,\beta)}$.
- Sample a new value σ_z^2 from its full conditional distribution.

The full conditional distribution of σ_z^2 will be an inverse-gamma distribution if the prior is also inverse-gamma.

Due to the number of terms in the log-likelihood, this algorithm is time consuming, especially for large datasets, for the following reasons:

1. For each $i = 1, \ldots, N$, updating z_i requires calculation of $(N-1)$ terms of the log-likelihood.
2. The updating of α and β requires calculation of all $O(N^2)$ terms of the log-likelihood.

Both sets of updates require $O(N^2)$ calculations at each iteration of the Markov chain Monte Carlo (MCMC) algorithm. The computing time increases with the square of the size of the network. This computational cost makes the latent space model infeasible for large networks, typically in practice when the size of the network is above 1000. To make likelihood-based inference (including Bayesian inference via MCMC) feasible, we propose an approximation to the log-likelihood function in Equation (4). Using this approximation, we show that the computational cost can be reduced from $O(N^2)$ to $O(N)$. Throughout this article, we will focus on the computation of the latent space model. The general idea of the approximation will also apply to other statistical network models, such as latent class models (Nowicki and Snijders 2001; Airoldi, et al. 2008), latent factor models (Hoff 2009), or latent cluster random effects models (Krivitsky et al. 2009).

We describe the approximation in Section 2. In Sections 3 and 4, we evaluate its performance using simulated data and a subset of a PPI dataset. We also fit the PPI data using the proposed approximation in Section 4 and use the fitted model to identify false positive links, a practical application of the latent space network models.

2. CASE-CONTROL APPROXIMATE LIKELIHOOD

Large networks are usually sparse. For example, in most cellular networks in biology, including metabolic, physical interaction, and regulatory networks, there are small numbers of highly connected hub nodes and the majority of the nodes have low degrees (Barabási and Oltvai 2004). Therefore, the summation in Equation (4) involves mostly terms in which $Y_{ij} = 0$.

In epidemiology, case-control studies are widely used to compare a group having the outcome of interest ("case") to a control group with regard to one or more characteristics (Breslow 1996). Usually, the cases are so rare that it is impossible or too expensive to draw a simple random sample with enough cases to draw conclusions, or to conduct a cohort study. This is because in a cohort study the case cohort has far fewer members than the control cohort. In a case-control study, available cases are collected and corresponding controls are sampled from the disease-free cohort. Statistics plays an important role in analyzing case-control studies, with regard to finding causal relations, designing efficient sampling methods, and controlling sampling bias and confounding factors. (For a comprehensive history of case-control studies and a manual of statistical methods in case-control studies, see Breslow and Day (1980) and Breslow (1996).)

In network data, if we view the 1's as cases and the 0's as controls, we are interested in studying the observed and unobserved factors that distinguish these two populations. This is similar to identifying the risk factors of disease in an epidemiological study. This analogy suggests an approximation to the log-likelihood function, which can be written as follows:

$$\ell \equiv \log \Pr(Y|\eta) = \sum_{i=1}^{N} \ell_i, \tag{5}$$

where

$$\begin{aligned}\ell_i &\equiv \sum_{j\neq i}\{\eta_{ij}y_{ij} - \log(1+e^{\eta_{ij}})\}\\ &= \sum_{j\neq i, Y_{ij}=1}\{\eta_{ij} - \log(1+e^{\eta_{ij}})\} + \sum_{j\neq i, Y_{ij}=0}\{-\log(1+e^{\eta_{ij}})\}\\ &= \ell_{i,1} + \ell_{i,0}.\end{aligned}$$

Thus, the sum in Equation (5) is a sum over rows of the sociomatrix Y, that is, ℓ_i is the contribution to the likelihood of the links from node i.

The quantity $\ell_{i,0}$ can be viewed as a population total statistic. This population total can be estimated by a simple random sample of the population:

$$\tilde{\ell}_{i,0} = \frac{N_{i,0}}{n_{i,0}}\sum_{k=1}^{n_{i,0}}\{-\log(1+e^{\eta_{ik}})\}, \tag{6}$$

where $N_{i,0}$ is the total number of 0's in the ith row, $n_{i,0}$ is the number of samples selected from the ith row, and the sum is over those selected entries. Since $\tilde{\ell}_{i,0}$ is based on a random sample from among the 0's, $\mathbf{E}[\tilde{\ell}_{i,0}] = \ell_{i,0}$. For a large network, we can choose a relatively small $n_{i,0}$ to get an unbiased estimator of $\ell_{i,0}$ and thus greatly reduce the amount of computation.

However, $\tilde{\ell}_{i,0}$ might not be the best estimator of $\ell_{i,0}$. The latent space model assumes that nodes that are "closer" to each other are more likely to form a tie than those farther apart. This is often true in real networks. Therefore, for each node i, the population of 0's is not homogeneous. The nodes that are "closer" to node i may contain more information and be more relevant in estimating the latent position of node i. We use the shortest path length from node i to node j in the network (D_{ij}) to define "closeness."

Similarly to stratified sampling, we divide the 0's into M strata according to D_{ij}, leading to the following decomposition of the contribution to the log-likelihood by the relations $y_{i,j}$ involving node i:

$$\begin{aligned}\ell_i = &\sum_{j:Y_{ij}=1}\{\eta_{ij} - \log(1+e^{\eta_{ij}})\} + \sum_{j:D_{ij}=2}\{-\log(1+e^{\eta_{ij}})\}\\ &+ \cdots + \sum_{j:D_{ij}=M}\{-\log(1+e^{\eta_{ij}})\}.\end{aligned} \tag{7}$$

Therefore, the following is an unbiased estimator of ℓ_i based on a stratified sample:

$$\hat{\ell}_i = \sum_{j:Y_{ij}=1}\{\eta_{ij} - \log(1+e^{\eta_{ij}})\} + \sum_{h=2}^{M}\frac{N_{i,h}}{n_{i,h}}\sum_{j:D_{ij}=h}\{-\log(1+e^{\eta_{ik}})\}, \tag{8}$$

where $N_{i,h}$ is the total number of nodes j with $D_{ij} = h$, and $n_{i,h}$ is the number of selected samples in the hth stratum.

Now we describe how we determine $n_{i,h}$. First, we pick a global control-to-case rate r and set the total control size of each node $n_{i,0} = r\bar{d} \equiv n_0$, where $\bar{d}$ is the mean degree of the entire network. It is also possible to vary $n_{i,0}$ across different nodes. Given a fixed $n_{i,0} = \sum_{h=1}^{M} n_{i,h}$, we choose $n_{i,h}$ to be proportional to the hth stratum's contribution to the log-likelihood change in sampling z_i. We first draw a simple random sample of size $n_{i,0}$ and conduct a pilot MCMC run.

At each iteration t of the pilot run (after removing the initial burn-in iterations and thinning if appropriate), we calculate the log-likelihood change as follows:

$$\begin{aligned}\Delta\tilde{\ell}_i^{(t)} &\equiv \tilde{\ell}_i\big(z_i^{(t)*}\big) - \tilde{\ell}_i\big(z_i^{(t)}\big) = \ell_{i,1}\big(z_i^{(t)*}\big) - \ell_{i,1}\big(z_i^{(t)}\big) + \sum_h \big\{\tilde{\ell}_{i,h}\big(z_i^{(t)*}\big) - \tilde{\ell}_{i,h}\big(z_i^{(t)}\big)\big\} \\ &\equiv \Delta\ell_{i,1}^{(t)} + \sum_h \Delta\tilde{\ell}_{i,h}^{(t)},\end{aligned}$$

where $z_i^{(t)*}$ is the proposed new value of z_i. We define $w_{i,h}^{(t)} = |\Delta\tilde{\ell}_{i,h}^{(t)} / \sum_{g=2}^{M} \Delta\tilde{\ell}_{i,g}^{(t)}|$. Then we calculate the relative weights as $w_{i,h} = \frac{1}{T-1}\sum_{t=1}^{T-1} w_{i,h}^{(t)}$, where T is the number of iterations in the pilot MCMC run after burn-in and thinning. Finally, we set the size of the h-stratum for the i-node to be

$$n_{i,h} = n_{i,0} w_{i,h} \Big/ \sum_{g=2}^{M} w_{i,g}, \tag{9}$$

for $h = 2, \ldots, M$.

Typically, n_0 is small compared to N. Therefore, at every evaluation of the log-likelihood function, the summation is over $O(n_0)$ terms, which does not grow with the network size N. We call $\hat{\ell} = \sum_i \hat{\ell}_i$ the *stratified case-control log-likelihood*. Using the case-control log-likelihood reduces the computational cost from $O(N^2)$ to $O(N)$.

To summarize, the algorithm is as follows.

1. Carry out a pilot MCMC run as follows:
   (a) For each node $i = 1, \ldots, N$, choose $n_{i,0}$ other nodes at random among those for which $Y_{i,j} = 0$.
   (b) Run the MCMC algorithm with the usual log-likelihood, ℓ, replaced by the approximate log-likelihood $\tilde{\ell} = \sum_{i=1}^{N}(\ell_{i,1} + \tilde{\ell}_{i,0})$, where $\tilde{\ell}_{i,0}$ is given by Equation (6).
2. For each node $i = 1, \ldots, N$ and for each path length $h = 2, \ldots, M$:
   (a) Calculate the number of other nodes to be sampled, $n_{i,h}$, from Equation (9).
   (b) Sample $n_{i,j}$ nodes from among those for which $Y_{i,j} = 0$ and $D_{i,j} = h$.
3. Carry out the full MCMC run with the usual log-likelihood, ℓ, replaced by the stratified case-control log-likelihood, $\hat{\ell}$.

The algorithm has a control parameter, the global control-to-case rate r, which has to be chosen by the user. The stratified case-control log-likelihood is an unbiased estimator of the true log-likelihood regardless of r, but different choices may yield more efficient estimators. Our numerical experiments led us to the values we used in our examples, but it would be desirable to develop a more systematic way of choosing r.

3. SIMULATION STUDIES

The latent space model provides an easy way to simulate networks with certain degrees and structures. To evaluate the performance of the proposed case-control likelihood, we simulated several networks from the latent space model by Hoff, Raftery, and Handcock (2002), and also from the latent position cluster model by Handcock, Raftery, and Tantrum (2007) with two clusters. We set the dimension of the latent space to be 2. We simulated three scenarios, and for each scenario we simulated networks of three different sizes: 100 nodes, 200 nodes, and 500 nodes.

The first scenario was the latent space model. We set the intercept to be a value that makes the average degree of the network approximately 10. The latent positions were generated independently from the bivariate normal distribution with mean (0, 0) and covariance matrix $2I_2$.

The second scenario was also the latent space model, but was sparser so as to assess the robustness of the method to sparsity. We set the intercept to be a value that makes the average degree of the network approximately 5, half as much as in the first scenario.

The third scenario was the latent position cluster model with two clusters. We generated half of the latent z_i's from a bivariate normal((2, 2), $2I_2$) and the other half from a bivariate normal ((2, −2), $2I_2$). For each case, we generated three networks of sizes 100, 200, and 500. (Due to computational costs, it is not feasible to compute the full likelihood for networks with sizes much greater than 500.)

For each of the networks, we fit the latent space model with the original full-likelihood algorithm and the proposed case-control likelihood. When constructing the case-control likelihood, we set the number of selected controls per row equal to 50. This corresponded to a control-to-case rate of 5 for the first and third scenarios, and a rate of 10 for the sparser second scenario.

We evaluated the performance of the case-control approximation by:

1. comparing the CPU time needed to evaluate the two likelihoods;
2. comparing the case-control likelihood function with the full-likelihood function evaluated at a series of parameter values;
3. comparing the estimated link probabilities p_{ij}; and
4. comparing the ROC curves produced by the estimated link probabilities from both likelihoods.

The CPU times needed to evaluate the case-control likelihood and full likelihood for different sizes of the networks under the first scenario are summarized in Table 1. They are all in seconds per 1000 likelihood evaluations. Comparing the time cost ratios, we can see that the CPU time for the full likelihood did indeed increase at a rate close to $O(N^2)$ even for these relatively small networks. The CPU time for the case-control likelihood increased at a rate close to $O(N)$.

The case-control likelihood reduced the CPU time by 30% for $N = 100$, and by 83%, or by a factor of 6, for $N = 500$. It is not surprising that the savings for $N = 100$ were

Table 1. CPU time of case-control likelihood and full likelihood, for different network sizes for the latent space model with average degree 10. All times are in seconds per 1000 likelihood evaluations

	$N = 100$	$N = 200$	$N = 500$
Full likelihood	1.89	6.95	45.08
Case-control likelihood	1.34	2.82	7.60

relatively small, because the case-control log-likelihood involves evaluating about 60% of the components in the full log-likelihood. These empirical results indicate that the computational overhead involved in setting up the case-control likelihood is a small part of the overall CPU time needed.

We compare the result from estimating the models using Bayesian MCMC with the case-control likelihood and the full likelihood in Figure 1 for the latent space model and Figure 3 for the latent position cluster model with two clusters.

The log-likelihoods from the case-control method tracked the full log-likelihood well, as indicated by the left-most panels in Figures 1, 2, and 3. In all cases, the correlations between the two log-likelihoods across values of the parameters visited by the MCMC algorithm were at least 0.88.

The fitted link probabilities using the case-control likelihood were similar to those estimated by the full likelihood, as shown by the middle columns of plots in Figures 1, 2, and 3. The scatterplots are symmetric around the diagonal line, which is in line with the fact that the case-control log-likelihood is an unbiased estimator of the full log-likelihood. The link probabilities estimated by the two methods are highly correlated, with correlations of at least 0.95 for $N = 100$ and 200, and 0.91 for $N = 500$.

The ROC curves generated by the case-control likelihood were indistinguishable from those generated by the full likelihood, as shown by the right-most plots in Figures 1, 2, and 3. This indicates that the two estimation methods provided essentially identical overall fits to the data in terms of predicting links.

4. PROTEIN–PROTEIN INTERACTION DATA

Proteins are involved in most cell functions, and they typically bind together to form complex structures. Such PPIs are intrinsic to virtually every cellular process (Phizicky and Fields 1995). A variety of experimental techniques are available to identify proteins that interact and to determine the strengths of these interactions. These PPIs, when measured on a genome scale, can be used to create a PPI network in which two proteins are connected if they are observed to interact. A PPI network is an undirected graph in which the nodes are proteins and two proteins are linked by an edge if they interact.

PPIs are important for many biological processes, and understanding such networks can give insights into the function of individual proteins, protein complexes and cellular machinery (Uetz et al. 2000; Kuchaiev, et al. 2009). For example, PPI networks have been used to predict the functions of uncharacterized proteins (Huynen et al. 2003; Bandyopadhyay, Sharan, and Ideker 2006). Many tools and methods have been developed to study and visualize the topology of PPI networks, such as Cytoscape (Shannon et al. 2003), Osprey

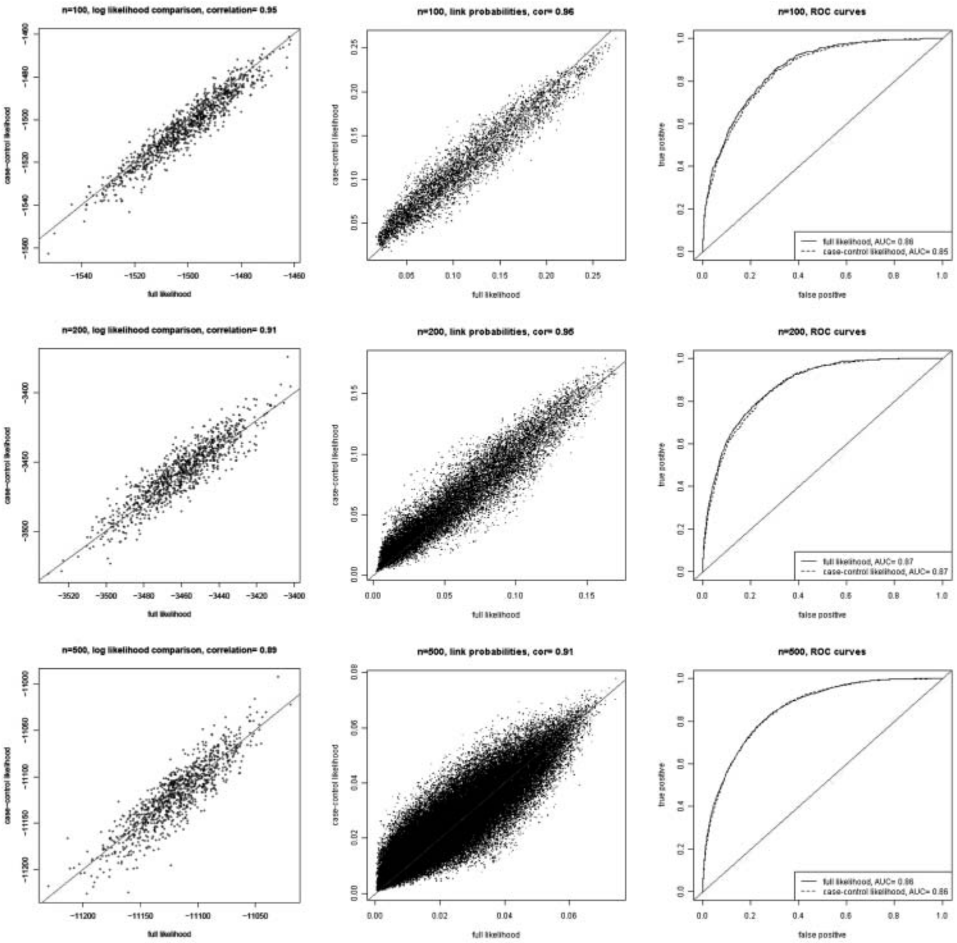

Figure 1. Comparison of the results using the exact likelihood with those using the case-control approximate likelihood for the latent space model with average degree 10 and network sizes 100 (top row), 200 (middle row), and 500 (bottom row). The left panels show the exact log-likelihood function on the x-axis and the case-control approximate log-likelihood function on the y-axis; each point corresponds to one of the parameter vectors visited by the MCMC algorithm. The panels in the middle column show the estimated link probabilities estimated from the two likelihoods, with the exact log-likelihood function on the x-axis and the case-control approximate log-likelihood function on the y-axis; each point corresponds to one directed pair of actors in the network. The right panels show the ROC curves generated by the estimated probabilities from the exact and approximate approaches; in each case these are almost identical.

(Breitkreut, Stark, and Tyers 2003), VisANT (Hu et al. 2005) and PIANA (Aragues, Jaeggi, and Oliva (2006). PPI networks have also been used to identify new disease-related genes. (For a review of this work, see Ideker and Sharan (2008)).

Many experimental methods have been developed to identify PPIs, such as yeast two-hybrid or tandem affinity purification. However, these high throughput techniques are known to produce many false positives and false negatives. For example, the false positive rates could be as high as 64% for yeast two-hybrid experiments and 77% for tandem affinity purification experiments (Edwards et al. 2002). These false positive links can yield

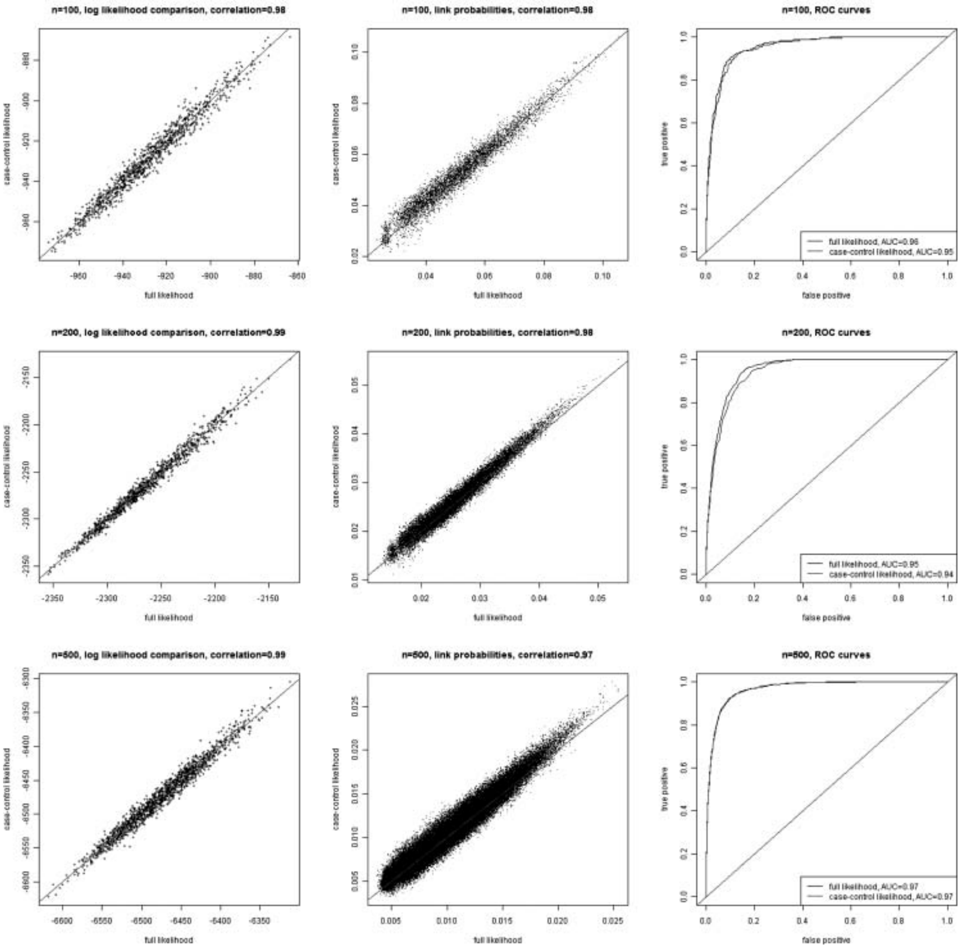

Figure 2. Comparison of the results using the exact likelihood with those using the case-control approximate likelihood for the latent position model with average degree 5. The panels are as described in Figure 1.

misleading scientific hypotheses and lead to costly and unproductive biological validation experiments. Hence, there is great interest in finding ways to assess the reliability of PPIs (Deng, Sun, and Chen 2003; Lin 2009), and to identify and remove these false positive links (Mahdavi and Lin 2007; Kuchaiev, Lici, Chen et al.2009)

Here, we use the PPI data for the yeast *Saccharomyces cerevisiae* as an example of the usefulness of the latent space model. The latent space model assumes that the presence of a link depends on the distance between the latent positions of two nodes. One possible use of the latent space model is to help identify the false positive links in the PPI network. We downloaded the PPI data from the Saccharomyces Genome Database (SGD; *http://downloads.yeastgenome.org*) compiled from the Biological General Repository for Interaction Datasets (BioGRID) database (Stark 2006). In Section 4.1, we show that our approximate case-control likelihood yields similar results to the full likelihood using a

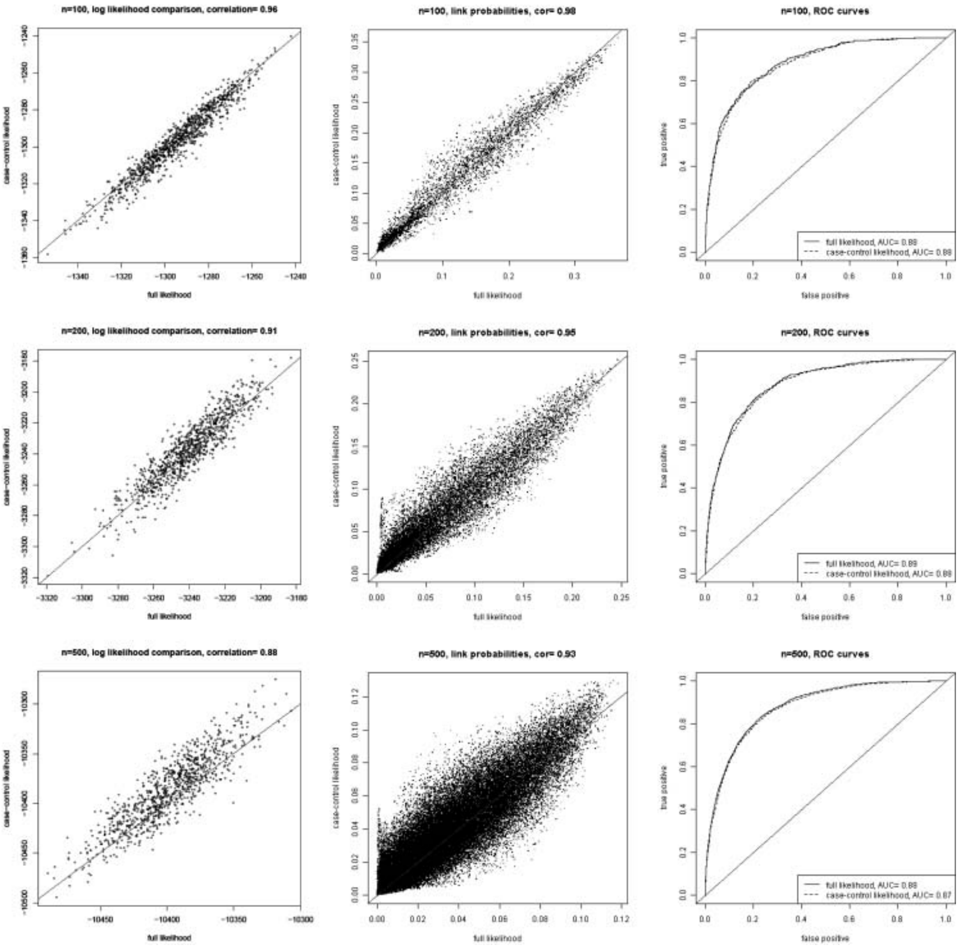

Figure 3. Comparison of the results using the exact likelihood with those using the case-control approximate likelihood for the latent position cluster model with two clusters. The panels are as described in Figure 1.

small connected subnetwork. In Section 4.2, we show the effectiveness of the latent space model in identifying false positives in a large PPI network.

4.1 A SMALL CONNECTED SUBSET OF THE PPI DATA

Our previous simulation results showed that when the data are generated from a latent space model, the case-control approximation can provide similar estimation to the full likelihood. These results are based on simulated networks from the latent space and latent position cluster models, when we know that the model we fit is the correct one. Before applying the case-control likelihood approximation to the full data, we would like to evaluate the performance of the case-control likelihood for this real dataset, when we do not know the true model.

We selected a connected subnetwork with 200 nodes and 1524 links, counting the symmetric pairs twice. For this subnetwork, we were able to fit the latent space model

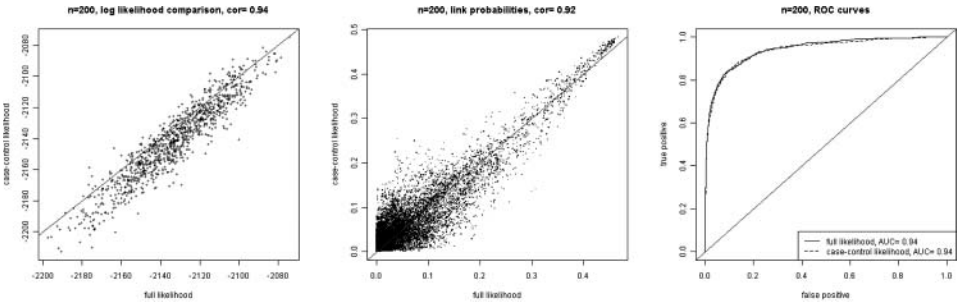

Figure 4. Comparison of the results using the exact likelihood with those using the case-control approximation for the PPI subnetwork of size 200. The panels are as described in Figure 1.

using the full likelihood and to compare the results with those using the case-control likelihood. We fit the latent space model using both the exact likelihood and the case-control approximation, and we evaluated the case-control approximation in the same way as in Section 3. The results are summarized in Figure 4. The results from fitting the subnetworks show that the case-control likelihood works well for the real data too.

To evaluate whether we can identify false positive links, we randomly implanted false positives by adding about 150 nonexistent edges to the PPI data. This increases the number of links by about 20%. Then we fit the latent space model to this perturbed dataset using both the full likelihood and the case-control approximate likelihood to see whether we are able to identify the false links.

The case-control likelihood and the full likelihood produced similar results. Among the fitted probabilities for the perturbed data, the nine smallest probabilities were from the false positives, and 26 of the 50 smallest probabilities were from false positives. The boxplots of the fitted probabilities of the true positives, false positives, and true negatives in Figure 5 indicate that the false positives had much lower fitted probabilities of being linked than the true positives on average. These results suggest that the latent space network model is potentially useful for identifying false positive links in PPI network data, and that its usefulness is not diminished by using the much more computationally efficient case-control approximation.

4.2 A LARGE PPI DATASET

4.2.1 Visualization of the PPI Data. The large PPI physical interaction network we are using has 2716 proteins with a total of 27,586 links, where the symmetric pairs have been counted twice. This is a sparse network with a mean degree of about 10 links per protein, and a median degree of only 5. This discrepancy between the mean and median indicates different activity levels across the nodes. The most active protein has 194 links while 495 proteins have only one link and a quarter of the proteins have two links. Figure 6 plots the ROC curve constructed by the fitted probabilities, which suggests that the latent space model fits the data reasonably well.

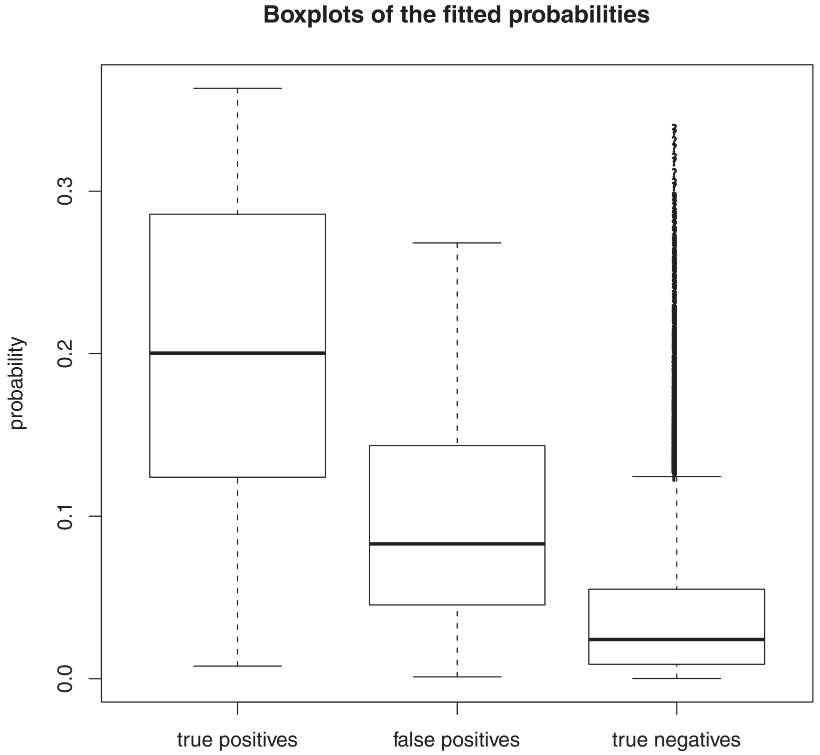

Figure 5. Perturbed PPI subnetwork of size 200: boxplots of the fitted probabilities of the true positives (left), false positives (middle), and true negatives (right). The fitted probabilities were calculated using an MCMC algorithm based on the case-control likelihood.

One advantage of the latent space model is that it provides a visualization of the network data. We plot the latent positions of all the proteins in the left panel of Figure 7. To facilitate visualization, the right panel of Figure 7 shows latent positions of the nodes connected to the three most active proteins (YPL240C, YJL164C, and YBR160W), that is, those with the highest degree. Nodes YPL240C and YJL164C are estimated as being closer together in the latent space than they are to node YBR160W. This makes sense, as the former two nodes are connected to each other but not to the latter. Additionally, YPL240C and YJL164C are more similar in terms of their ties to other nodes: The correlation between the ties from YPL240C and the ties from YJL164C is 0.023, which, while low, is several times higher than the correlation between YPL240C and YBR160W or between YJL164C and YBR160W (both about 0.006).

4.2.2 Identifying False Positive Links. Similarly to what we did for the subset of the data, we first randomly perturbed the data by changing a number of 0's into 1's equal to 20% of the number of actual links in the original dataset. We then fit the latent space model to this perturbed dataset. The boxplots of the fitted probabilities of the true positives, false positives, and true negatives in Figure 8 indicate that the fitted probabilities for the false positives were much lower than those for the true positives, and indeed were closer on average to those for the true negatives. Eight of the 10 smallest fitted link probabilities

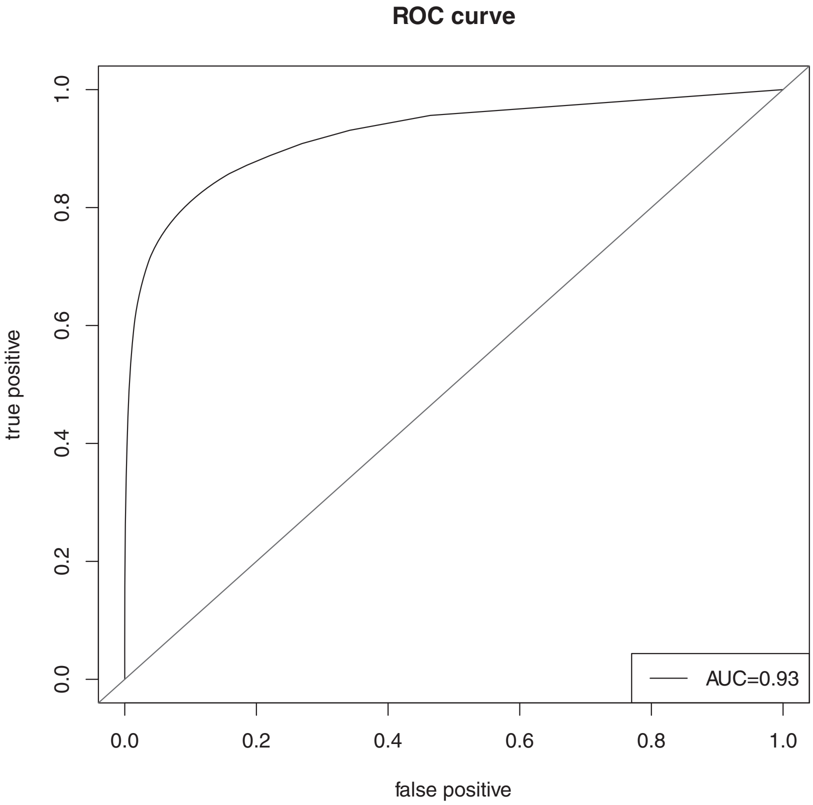

Figure 6. ROC curve of the fitted link probabilities for the large PPI dataset.

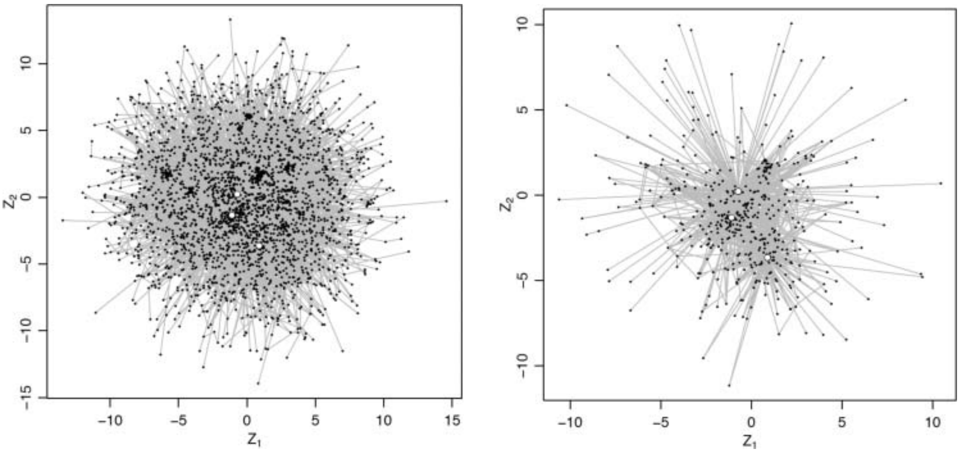

Figure 7. Latent positions for the large PPI dataset. The left panel shows latent positions for all nodes, and the right panel shows positions for the subnetwork formed by nodes connected to the three most active proteins, that is, those with the highest degree. The three most active proteins are shown by white circles in the figure, and are YPL240C, YJL164C, and YBR160W in order of decreasing vertical position.

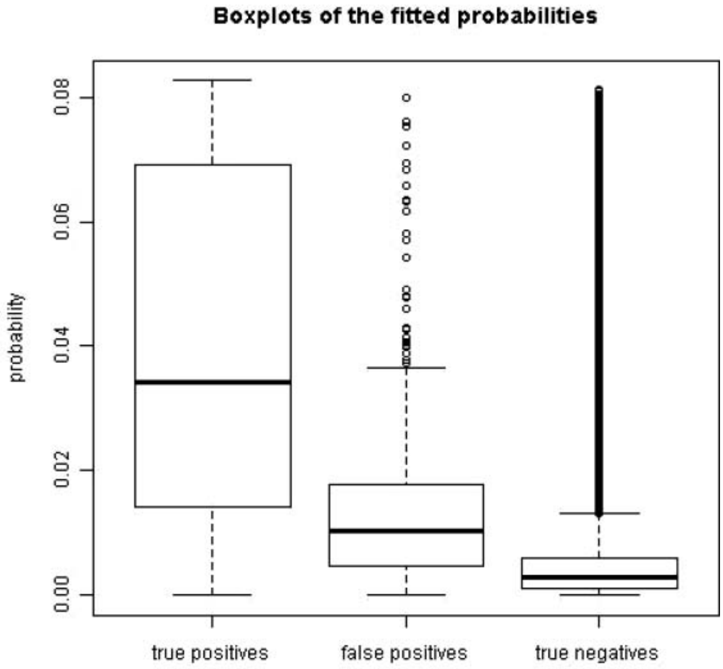

Figure 8. Perturbed full PPI data: boxplots of the fitted probabilities of the true positives (left), false positives (middle), and true negatives (right).

were from the false positives. These results for the full dataset suggest that the latent space model may provide a promising way to identify false positive links. Note that these results were obtained using the case-control approximate likelihood only.

The Gene Ontology (GO) project aims to standardize the representation of gene and gene product attributes across species and databases (Ashburner et al. 2000). The project provides a controlled vocabulary of terms for describing gene product characteristics and annotations. The relationships between GO terms are represented as directed acyclic graphs. On the other hand, GO-slim terms are cut-down versions of the GO ontologies containing a subset of the terms in the whole GO. They give a broad overview of the ontology content without the detail of the specific fine-grained terms. GO-slim terms are not hierarchical in nature, unlike GO terms. Therefore, GO-slim is particularly useful for providing a summary of function annotations. GO-slim terms are available from the gene ontology (GO) web site *(http://www.geneontology.org/GO.slims.shtml*). Since GO-slim terms represent curated annotations from the literature, they are considered standard gene function annotations. Interacting proteins are expected to share GO-slim terms because PPIs can be used to predict gene functions of uncharacterized proteins (Huynen et al. 2003; Bandyopadhyay, Sharan, and Ideker (2006)), and proteins with similar functions are expected to share GO and GO-slim terms. Comparing GO terms and PPIs is a standard technique in the literature (Wu et al. 2006; Mahdavi and Lin 2007; Brady et al. 2009; Kuchaiev et al. 2009; Lavallee-Adam, Coulombe, and Blanchette 2010). In particular, Mahdavi and Lin (2007) used GO annotations to reduce false positive PPI pairs resulting from computational predictions.

Next we used the fitted link probabilities from the real data to identify false positive links. We hypothesize that links in the data with very low fitted probabilities are probably false positive links. Mahdavi and Lin (2007) used GO annotations to reduce false PPI pairs resulting from computational predictions. The key idea is that interacting proteins are likely to share GO slim terms. We used this criterion to evaluate the fitted link probabilities we get from the latent space model. We used the GO-slim terms (Ashburner 2000) from the SGD database *(http://downloads.yeastgenome.org)*. For each pair of interacting proteins, if it is documented to share one or more GO slim terms, we called it a true positive, otherwise a false positive. Note that this definition of true and false positives is itself subject to error.

We compare the boxplots of the fitted link probabilities of the true positives and false positives in Figure 9. We can clearly see a difference in the two populations, which indicates that there is considerable agreement between the identification of false positives by the latent space model and by the GO-slim terms. Of course we would not expect perfect or even very strong agreement, even if the latent space model discriminated perfectly between true and false positives, because the GO-slim terms provide only an approximate identification of false positives.

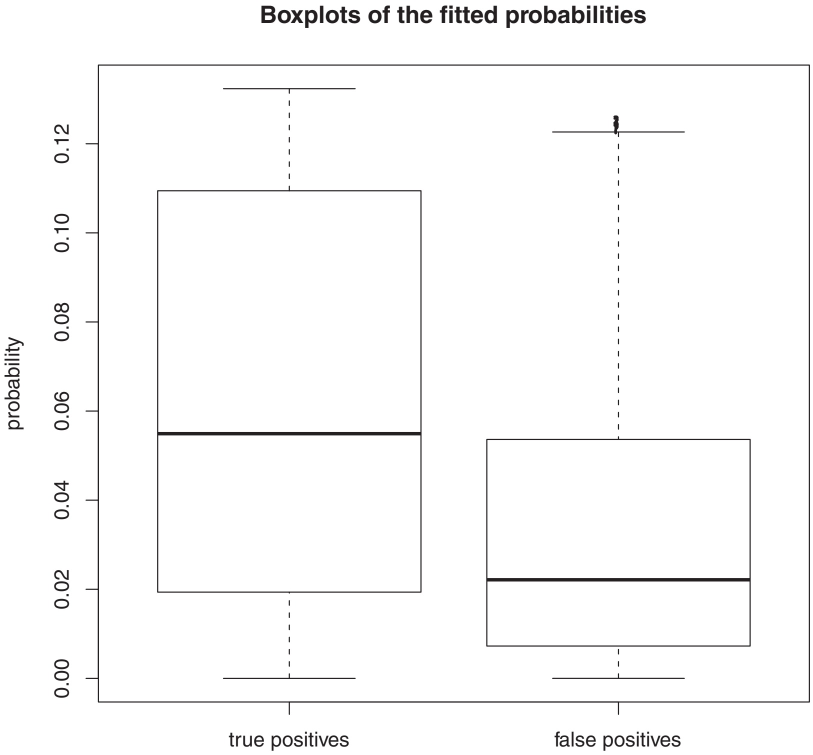

Figure 9. Full PPI data: fitted link probabilities grouped by whether or not the edges share a GO-slim term.

5. DISCUSSION

An obstacle to the use of latent space models for networks has been the fact that existing likelihood and Bayesian estimation methods do not scale to large networks, because the required computation is $O(N^2)$, where N is the number of nodes or actors in the network. We have proposed an approximate likelihood based on the same idea that underlies case-control studies, and we have found it to perform well in simulated and real data. This reduces computation from $O(N^2)$ to $O(N)$, and makes it feasible to do Bayesian estimation via MCMC for large networks.

We have implemented our method for estimating the latent space model (Hoff, Raftery, and Handcock 2002) and the latent position cluster model (Handcock Raftery, and Tantram 2007), but the basic idea can be applied to other statistical network models as well. They can be used to reduce computation for likelihood-based estimation for network models for which the log-likelihood involves a sum of contributions from all or most of the pairs of actors. These include the latent position random effects model (Krivitsky et al. 2009), which is a direct extension of the latent space model, and explicitly models different activity levels of nodes. They also include the latent class model by Nowicki and Snijders (2001) and the latent factor model by Hoff, Raftery, and Handcock (2002). (See Goldenberg et al. (2009) for a survey of these and other network models.)

Other approaches to efficient computation for statistical network models have been explored, notably the variational Bayes approach by Attias (1999). This was applied to stochastic blockmodels by Airoldi et al. (2008) and extended to the latent position cluster model by Salter-Townshend and Murphy (2010). Rather than try to approximate the likelihood, this attempts to find and use a lower bound for the likelihood.

6. SUPPLEMENTARY MATERIALS

Supplementary Materials are available online in a single archive as follows:

README file: Text file “readme.txt” containing a list of the files in the Supplementary Materials.

C program to calculate likelihoods: C program “if.c” to calculate the likelihood and the case-control approximate likelihood.

R program for pilot run: R program “pilot_new.r” to carry out the pilot MCMC run and obtain the sampling weights and initial values.

R program for case-control likelihood: R program “mcc.r” to carry out a full MCMC realization using the case-control approximate likelihood.

Example R code: Example R code “mccfit.r” to run the case-control likelihood.

PPI data: File “PPI” containing the PPI data.

ACKNOWLEDGMENTS

This research was supported by NIH grants R01 GM-84163, R01 HD-54511, and R01 HD-70936. The authors are grateful to Pavel Krivitsky, the editor, the associate editor, and an anonymous referee for very helpful comments and discussions.

[Received July 2010. Revised July 2011.]

REFERENCES

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008), "Mixed Membership Stochastic Blockmodels," *Journal of Machine Learning Research*, 9, 1981–2014. [902,904,917]

Aragues, R., Jaeggi, D., and Oliva, B. (2006), "Piana: Protein Interactions and Network Analysis," *Bioinformatics*, 22, 1015–1017. [909]

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, M., Cherry, J. M., Davis, A. P., Dolinsky, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000), "Gene Ontology: Tool for the Unification of Biology. The Gene Ontology Consortium," *Nature Genetics*, 25, 25–29. [915,916]

Attias, H. (1999), "Inferring Parameters and Structure of Latent Variable Models by Variational Bayes," in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 21–30. [917]

Bandyopadhyay, S., Sharan, R., and Ideker, T. (2006), "Systematic Identification of Functional Orthologs Based on Protein Network Comparison," *Genome Research*, 16, 428–435. [908,915]

Barabási, A. L., and Oltvai, Z. N. (2004), "Network Biology: Understanding the Cell's Functional Organization," *Nature Reviews Genetics*, 5, 101–113. [904]

Brady, A., Maxwell, K., Daniels, N., and Cowen, L. J. (2009), "Fault Tolerance in Protein Interaction Networks: Stable Bipartite Subgraphs and Redundant Pathways," *PLoS ONE*, 4, e5364. [915]

Breitkreut, B. J., Stark, C., and Tyers, M. (2003), "Osprey: A Network Visualization System," *Genome Biology*, 4, R22. [909]

Breslow, N. E. (1996), "Statistics in Epidemiology: The Case-Control Study," *Journal of the American Statistical Association*, 91, 14–28. [904]

Breslow, N. E., and Day, N. (1980), *Statistical Methods in Cancer Research: Volume 1 – The Analysis of Case-Control Studies*, Lyon: IARC Scientific Publications. [904]

Deng, M., Sun, F., and Chen,T. (2003), "Assessment of the Reliability of Protein–Protein Interactions and Protein Function Prediction," *Pacific Symposium of Biocomputing*, 8, 140–151. [910]

Edwards, A. M., Kus, B., Jansen, R., Greenbaum, D., Greenblatt, J., and Gerstein, M. (2002), "Bridging Structural Biology and Genomics: Assessing Protein Interaction Data With Known Complexes," *Trends in Genetics*, 18, 529–536. [909]

Frank, O., and Strauss, D. (1986), "Markov Graphs," *Journal of the American Statistical Association*, 81, 832–842. [901]

Goldenberg, A., Zheng, A. X., Fienberg, S. E., and Airoldi, E. M. (2009), "A Survey of Statistical Network Models," *Foundations and Trends in Machine Learning*, 2, 129–233. [917]

Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007), "Model-Based Clustering for Social Networks" (with discussion), *Journal of the Royal Statistical Society,* Series A, 170, 301–354. [901,903,907,917]

Hoff, P. D. (2009), "Multiplicative Latent Factor Models for Description and Prediction of Social Networks," *Computational and Mathematical Organization Theory*, 15, 261–272. [904]

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. [901,902,903,907,917]

Hu, Z., Mellor, J., Wu, J., Yamada, T., Holloway, D., and DeLisi, C. (2005), "Visant: Data-Integrating Visual Framework for Biological Networks and Modules," *Nucleic Acids Research*, 33, W352–W357. [909]

Huynen, M. A., Snel, B., von Mering, C., and Bork, P. (2003), "Function Prediction and Protein Networks," *Current Opinions in Cell Biology*, 15, 191–198. [908,915]

Ideker, T., and Sharan, R. (2008), "Protein Networks in Disease," *Genome Research*, 18, 644–652. [909]

Krivitsky, P. N., Handcock, M. S., Raftery, A. E., and Hoff, P. D. (2009), "Representing Degree Distributions, Clustering, and Homophily in Social Networks With Latent Cluster Random Effects Models," *Social Networks*, 31, 204–231. [903,917]

Kuchaiev, O., Rašajski, M., Higham, D. J., and Pržulj, N. (2009), "Geometric Denoising of Protein–Protein Interaction Networks," *PLoS Computational Biology*, 5, e1000454. [908,910,915]

Lavallee-Adam, M., Coulombe, B., and Blanchette, M. (2010), "Detection of Locally Over-Represented GO Terms in Protein–Protein Interaction Networks," *Journal of Computational Biology*, 17, 443–457. [915]

Lin, X., Liu, M., and Chen, X. W. (2009), "Assessing Reliability of Protein–Protein Interactions by Integrative Analysis of Data in Model Organisms," *BMC Bioinformatics*, 10(Suppl 4), S5. [910]

Mahdavi, M. A., and Lin, Y.-H. (2007), "False Positive Reduction in Protein–Protein Interaction Predictions Using Gene Ontology Annotations," *BMC Bioinformatics*, 8, 262. [910,915,916]

McFarland, D. D., and Brown, D. J. (1973), "Social Distance as a Metric: A Systematic Introduction to Smallest Space Analysis," in *Bonds of Pluralism: The Form and Substance of Urban Social Networks*, ed. E. Laumann, New York: Wiley, pp. 213–253. [902]

Nowicki, K., and Snijders, T. A. B. (2001), "Estimation and Prediction for Stochastic Blockstructures," *Journal of the American Statistical Association*, 96, 1077–1087. [902,904,917]

Phizicky, E. M., and Fields, S. (1995), "Protein–Protein Interactions: Methods for Detection and Analysis," *Microbiological Reviews*, 59, 94–123. [908]

Salter-Townshend, M., and Murphy, T. B. (2010), *Variational Bayesian Inference for the Latent Position Cluster Model*, Technical Report, Dublin: School of Mathematical Sciences, University College Dublin. [917]

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Scwilowski, B., and Ideker, T. (2003), "Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks," *Genome Research*, 13, 2498–2504. [908]

Stark, C., Breitkreutz, B. J., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. (2006), "Biogrid: A General Repository for Interaction Datasets," *Nucleic Acids Research*, 34, D536–D539. [910]

Uetz, P., Giot, L., Cagney, G., Mansfield, T. A., Judson, R. S., Knight, J. R., Lockshon, D., Narayan, V., Srinivasan, M., Pochart, P., Qureshi-Emili, A., Li, A., Godwin, B., Conover, D., Kalbfleisch, T., Vijayadamodar, G., Yang, M., Johnston, M., Fields, S., and Rothberg, J. M. (2000), "A Comprehensive Analysis of Protein–Protein Interactions in *Saccharomyces cerevisiae*," *Nature*, 403, 623–627. [908]

Wang, Y. J., and Wong, G. Y. (1987), "Stochastic Blockmodels for Directed Graphs," *Journal of the American Statistical Association*, 82, 8–19. [902]

Wasserman, S., and Pattison, P. (1996), "Logit Models and Logistic Regressions for Social Networks: I. An Introduction to Markov Graphs and p^*," *Psychometrika*, 61, 401–425. [902]

Wu, X., Zhu, L., Guo, J., Zhang, D.-Y., and Lin, K. (2006), "Prediction of Yeast Protein–Protein Interaction Network: Insights From the Gene Ontology and Annotations," *Nucleic Acids Research*, 34, 25–29. [915]