

Research Article

Link Prediction via Convex Nonnegative Matrix Factorization on Multiscale Blocks

Enming Dong, Jianping Li, and Zheng Xie

College of Science, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Zheng Xie; dream0617@163.com

Received 11 March 2014; Revised 21 June 2014; Accepted 7 July 2014; Published 17 July 2014

Academic Editor: Hu Ning

Copyright © 2014 Enming Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Low rank matrices approximations have been used in link prediction for networks, which are usually global optimal methods and lack of using the local information. The block structure is a significant local feature of matrices: entities in the same block have similar values, which implies that links are more likely to be found within dense blocks. We use this insight to give a probabilistic latent variable model for finding missing links by convex nonnegative matrix factorization with block detection. The experiments show that this method gives better prediction accuracy than original method alone. Different from the original low rank matrices approximations methods for link prediction, the sparseness of solutions is in accord with the sparse property for most real complex networks. Scaling to massive size network, we use the block information mapping matrices onto distributed architectures and give a divide-and-conquer prediction method. The experiments show that it gives better results than common neighbors method when the networks have a large number of missing links.

1. Introduction

As a fundamental problem in the network researches, link prediction attempts to estimate the likelihood of relationship between two individuals by the study of observed links and the property of nodes. Researches on the problem can benefit a variety of fields. For example, researchers in different areas can efficiently find their cooperative partners or assistants. Security agencies can more precisely focus their efforts on probable relationships in malicious networks. In social networks, people can find companions based on the prediction of their surrounding networks.

The natural framework of link prediction methods is the similarity-based algorithm. Simple similarity-based measures such as neighborhood-based measures, for example, Adamic-Adar score [1] and common neighbors [2], require consideration of the local structure of the networks. Recently, a considerable amount of work which draws attention to community structure and scalable proximity estimation [3, 4] gives good prediction accuracy. Some similarity-based measures such as the path based methods, for example, Katz [5] and Rooted PageRank [6, 7], which focus on the global structure of the networks, are more effective but have a

high computational complexity. A new measure based on neighbor communities has a good performance with a low complexity [8]. Maximum likelihood estimation, such as hierarchical structure model [3] and stochastic block model [9, 10], presuppose some organizing principles of the network structure. Some algorithms, such as probabilistic relational models [11], probabilistic entity-relationship models [12], and stochastic relational models [13], learn the underlying structure from the observed network and then predict the missing links. Lichtenwalter et al. [14] designed a flow based method for link prediction, which is more localized. Low rank matrices approximations can also be used in link prediction for network [15–17]. Based on the technique of cluster low rank approximation for massive graphs, Shin et al. proposed a multiscale link prediction method [18], which captures the information of global structure of network and can handle massive networks quickly.

In order to capture the information of both global structure and clustering structure of network, we consider low rank approximations as well as blocks in networks' adjacent matrices. Low rank approximations algorithms are good techniques to get the global information of the matrices. Meanwhile block structure is an important feature of matrices

and it is often true that links in the same blocks have similar properties. Indeed, links are easy to be found in dense blocks. Good block detection algorithms have error tolerance: they are unaffected by a few missing edges in a network. This suggests that the principle of block detection could be applied to edge prediction.

Theoretically, a probabilistic latent variable model is proposed that combines both the concepts of block structure and low rank approximations for matrices. The model provides a framework for predicting links. Firstly, any modularity clustering algorithm can be used to generate blocks, while the only limit is the computational complexity. Then different from the low rank matrices approximations algorithms already used for link predictions, we use a new low rank matrices approximations algorithm named convex nonnegative matrix factorization (CNMF) [19] to get the predicting results within the blocks. The reason we use CNMF is that the sparseness of solutions is in accord with the sparse property for most real complex networks, so the predicting results are more reliable. In small networks we use k -means to detect the block structure of a network's adjacency matrix and average the prediction matrices for some k to get the predicting results. Experiments show that our method shows better performance. Scaling to the massive size networks, it is infeasible to use CNMF directly for the high computational complexity. In this case, we use the block information mapping matrices onto distributed architectures and give a divide-and-conquer prediction method to embrace distributed computing.

2. Background

The network of n nodes can be represented mathematically by an adjacency $n \times n$ matrix A . Here we set the diagonal entries to be 1, which means each node has a link to itself. This adjacency matrix can be treated as an object-feature matrix. Reduced rank method CNMF gives an approximation

$$A \approx AWG, \quad (1)$$

which has an interesting property: if A is sparse, the factors W and G both tend to be sparse.

CNMF has a direct interpretation: $N \times L$ ($L \ll N$) matrix $F = AW$ is a convex combinations of the columns of A ; thus we could interpret its columns as weighted sums of certain objects' coordinates (the coordinate of i -object is given by i -column of A). So the columns of F can be treated as cluster centroids of objects and W weights the association between objects and clusters. Meanwhile G measure the strength of relationship between clusters and features; that is, $G_{ki} = 1$ if cluster k has feature i ; $G_{ki} = 0$ otherwise. So $(FG)_{ij}$ can measure the strength of relationship between object i with feature j and then can be used to predict link between i and j .

3. A Probabilistic Latent Variable Model

Although the background gives an intuitionistic interpretation of CNMF used in link prediction, we still need

theoretical guarantee. Here we propose a probabilistic latent variable model, and the model ensures that the probability of a link between two nodes can be expressed as a combination of CNMF and the block structure of a given adjacent matrix.

In probabilistic view, the observed network data is a realization of an underlying probabilistic model, either because it is itself the result of a stochastic process, or because the sampling has uncertainty. We can think of the adjacent matrices of network data $A^o = \{A_k^o, k = 1, \dots, K\}$ as the K object-feature matrices for objects $\{x_i, i = 1, \dots, M\}$ and features $\{y_j, j = 1, \dots, N\}$. In this paper, A^o contains an adjacency matrix and its blocks found by clustering methods such as K -means. The joint occurrence probability of an object and a feature can be factorized as

$$P(X = i, Y = j) = \sum_{k=1}^K P(X = i | Y = j, A = k) \times P(A = k | Y = j) P(Y = j), \quad (2)$$

where X is a variable for the index of objects, Y is a variable for the index of features, and A is a variable for the index of sampling. $P(X = i, Y = j | A = k)$ is the joint occurrence conditional probability given the observation A_k^o , and $P(A = k)$ is the priori probability that A_k^o is observed.

Objects in real data are often organized into modules or clusters and the probability that an object has a feature depends on the groups to which they belong. These clusters memberships are unknown to us. In the language of statistical inference, they are latent variable. Assuming each cluster is a combination of objects, the joint occurrence probability can be factorized as

$$P(X = i, Y = j) = \sum_{l=1}^L \sum_{k=1}^K P(X = i, A = k, C = l) \times P(Y = j | A = k, C = l), \quad (3)$$

where C is the variable for the index of cluster. Here, we assume that the random variables X, Y are conditional independent given C .

Define a random variable

$$V_{kij} = \begin{cases} 1, & (X = i, Y = j, A = k) \text{ occurrence;} \\ 0, & \text{else.} \end{cases} \quad (4)$$

If observing once, the expected value is

$$E(V_{kij}) = P(X = i, Y = j, A = k). \quad (5)$$

Let Z_{ij} be a random variable of the occurrence frequency of $(X = i, Y = j)$ in observing $\sum_{k=1}^K (\sum_{s,t} A_{kst}^o)$ times. Then the expected value is

$$\begin{aligned} E(Z_{ij}) &= \left(\sum_{k=1}^K \sum_{s,t} A_{kst}^o \right) E(V_{kij}) \\ &= \left(\sum_{k=1}^K \sum_{s,t} A_{kst}^o \right) P(X = i, Y = j, A = k) \end{aligned}$$

$$\begin{aligned}
&= \sum_{l=1}^L \left(\sum_{k=1}^K \sum_{s,t} A_{kst}^o \right) P(X = i \mid A = k, C = l) \\
&\quad \times P(Y = j \mid A = k, C = l) P(A = k, C = l) \\
&= \sum_{l=1}^L \left(\sum_{k=1}^K \sum_{s,t} A_{kst}^o \right) P(X = i, A = k, C = l) \\
&\quad \times P(Y = j \mid A = k, C = l). \tag{6}
\end{aligned}$$

For the reason of interpretability, we suppose the joint concurrent probability of i th object, l th cluster, and k th data sampling is given by a combination of data A_{kst}^o as follows:

$$P(X = i, A = k, C = l) = \frac{\sum_{t=1}^N A_{kit}^o W_{ktl}}{\sum_{k=1}^K \sum_{s,h} A_{ksh}^o}, \tag{7}$$

where $W_{ktl} \geq 0$ and

$$\sum_{i,l} \left(\sum_{t=1}^N A_{kit}^o W_{ktl} \right) = \sum_{s,h} A_{ksh}^o. \tag{8}$$

Constraint (8) ensures that the probability defined by (7) is well defined.

This constraint has the advantage that we could interpret $P(X = i, A = k, C = l)$ as weighted sums of certain joint concurrence probability of object, features, and data, given by

$$P(X = i, Y = t, A = k) = \frac{A_{kit}^o}{\left(\sum_{k=1}^K \sum_{s,h} A_{ksh}^o \right)}. \tag{9}$$

Therefore, (6) can be expressed as

$$E(Z_{ij}) = \sum_{k=1}^K \sum_{l=1}^L \sum_{t=1}^N A_{kit}^o W_{ktl} G_{klj}, \tag{10}$$

where $G_{klj} = P(Y = j \mid A = k, C = l)$. Our goal is to compute two 3-order tensors W and G .

If we are only inputting the adjacency matrix, we can drop the index for sampling; then

$$E(Z_{ij}) = \sum_{l=1}^L \sum_{t=1}^N A_{it}^o W_{tl} G_{lj}. \tag{11}$$

Equation (11) can be expressed by matrix as

$$(E(Z_{ij})) = A^o W G. \tag{12}$$

Now, we show that this factorization is equivalent to CNMF.

In fact, for any CNMF solution $(\widetilde{W}, \widetilde{G})$, $\sum_{j=1}^N \widetilde{G}_{lj} = 1$ does not hold and $\sum_{j=1}^N \widetilde{W}_{jl} = 1$ holds for any l . Let $D_{\widetilde{G}}$ be the diagonal matrix, containing the row sums of \widetilde{G} . We say that the matrix $\widetilde{W} D_{\widetilde{G}}$ approximately satisfies (8). This can be proved as follows:

$$\begin{aligned}
\sum_{i=1}^M \sum_{j=1}^N A_{ij}^o &\approx \sum_{i=1}^M \sum_{j=1}^N \sum_{l=1}^L \sum_{t=1}^N A_{it}^o (\widetilde{W}_{tl} D_{\widetilde{G}ll}) (D_{\widetilde{G}ll}^{-1} \widetilde{G}_{lj}) \\
&= \sum_{i=1}^M \sum_{l=1}^L \sum_{t=1}^N A_{it}^o (\widetilde{W}_{tl} D_{\widetilde{G}ll}). \tag{13}
\end{aligned}$$

So the factorization

$$(E(Z_{ij})) = A^o (\widetilde{W} D_{\widetilde{G}}) (D_{\widetilde{G}}^{-1} \widetilde{G}) \tag{14}$$

satisfies the condition of (11). If we are also inputting blocks, (10) can be solved as

$$(E(Z_{ij})) = \sum_{k=1}^K A_k^o \widetilde{W}_k \widetilde{G}_k, \tag{15}$$

where \widetilde{W}_k and \widetilde{G}_k are solutions of CNMF for A_k^o . This can be proved as the case of only inputting the adjacency matrix.

The algorithm of CNMF gives a global optimal solution to $\min \|A - AWG\|^2$. The computational complexity of it (the most time-consuming step in our method) for $M \times N$ matrix is of order $(N^2 M + t(2N^2 L + NL^2))$ for $K \times K$ factor W and is of order $t(2N^2 L + 2NL^2)$ for $K \times N$ factor G , where t is the number of iterations [19].

4. Algorithms

Inputting the network data, the missing link prediction by calculating (15) has three steps. First, partition the observed adjacent matrix into k^2 blocks, using any modularity clustering algorithms. Secondly, the predicting matrix is given by doing CNMF to approximate each block. Thirdly, sum the corresponding entities of predicting matrices for all k to make the final prediction. In small networks, We call our method K -CNMF, as we use K -means to partition the matrix. The diagram of K -CNMF for $k = \{1, 2, 3, \dots\}$ is shown in Figure 1.

The purpose of the first step is to use several scales structures information of the observed network. For small networks, CNMF approximation can be computed directly on the original matrix (block generated by K -means with $k = 1$) to use the global information. A simple interpretation of our method is that if an edge is predicted to exist in many scales, it should be a missing link with high probability. The input of K -CNMF also needs two parameters: desired rank L and scale parameter K . Algorithm 1 shows the algorithm for K -CNMF.

When predicting links in massive size network, K -means is unsuitable for high dimensional data clustering. Meanwhile, the high computational complexity of CNMF makes it also infeasible to be used on the large adjacent matrix directly. So we use fast modularity clustering algorithm [20] to generate blocks. Based on block structures, we give a divideand-conquer algorithm (M -CNMF) to predict links, which is shown in Algorithm 2. The algorithm works by partitioning a matrix into blocks which are small enough for CNMF directly. Then the predicting results for the small blocks are combined to give the final predicting result for the original matrix. In order to give a solution for CPU load balancing, the size of blocks should be similar, which is achieved by splitting the large blocks and combining the small blocks to make their sizes in the neighborhood of a given threshold.

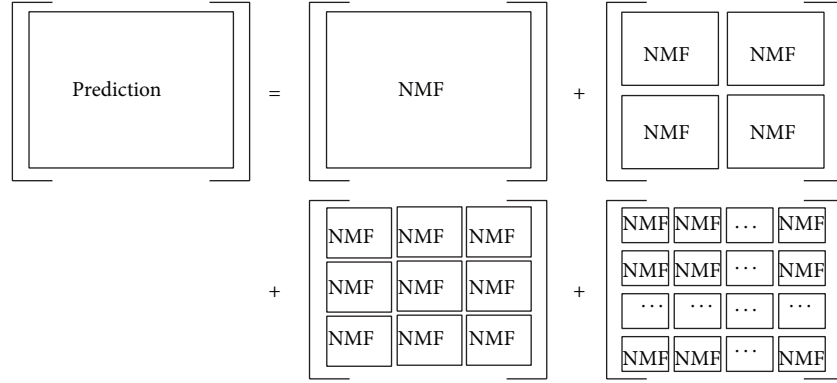


FIGURE 1: Diagram of K-CNMF.

```

input:  $A^o$ //observed network
       $L, K$ //desired rank, scale parameter
output:  $A$ //predicting matrix
 $A = (0)$ 
for  $k$  in range of  $(1, K)$ 
  do  $K$ -means to partition matrix into  $B = \{B_1, \dots, B_{k^2}\}$ 
  for block  $B_i \in B$ 
    do CNMF with rank  $\min(\text{col}(B_i), \text{row}(B_i), L)$ 
     $A = A + B_i W G$ //sum the corresponding entities
  end for
end for

```

ALGORITHM 1: Algorithm for K-CNMF.

5. Experiments and Comparison

In general, links between different nodes may have different weights in networks, representing their relative importance in the network. In our experiments, we set all weights to be one and get the original adjacency matrix A^T of the network. The observed network A^o is generated by removing a fraction of links randomly from the original network A^T , which will be called the missing edges. Then we use the two algorithms K-CNMF and M-CNMF to get the probability of links between nodes, which appears to be links' weight in the observed network.

5.1. Evaluation Method. To measure the accuracies of link prediction methods, the main metric we use is AUC [21], area under the receiver operating characteristic (ROC) curve, which is widely used in the machine learning and social network communities. If we rank pairs of nodes in order of decreasing, AUC is mean value of the probability that a missing link ($A_{ij}^o = 0$ & $A_{ij}^T \neq 0$) has a higher ranking than a nonexistent link ($A_{ij}^T = 0$). In practice, we do n independent comparisons. At each time we randomly pick a missing link and a nonexistent link to compare their ranking. If there are n^0 times the missing link has a higher ranking than the missing one and n^* times they have the same ranking, the AUC value is

```

input:  $A^o$ //observed network
       $L, K$ //desired rank, scale parameter
output:  $A$ //predicting matrix
 $A = (0)$ 
find community structure  $C = \{C_1, \dots, C_m\}$ 
for  $C_i \in C$ 
  if  $\text{size}(C_i) > K$ 
    divide  $C_i$  into  $\text{int}(\text{size}(C_i)/K + 1)$  equal parts
    append each part to  $C$ 
    delete  $C_i$ 
  end if
end for
for  $C_i \in C$ 
  if  $\text{size}(C_i) < K$ 
    for  $C_j \in C, i \neq j$ 
      if  $\text{size}(C_i \cup C_j) < K$ 
         $C_i = C_i \cup C_j$ 
        delete  $C_j$ 
      end if
    end for
  end if
end for
partition matrix into  $B = \{B_1, \dots, B_{|C|^2}\}$  by  $C$ 
for block  $B_i \in B$ 
  do CNMF with rank  $\min(\text{col}(B_i), \text{row}(B_i), L)$ 
   $A = A + B_i W G$ //sum the corresponding entities
end for

```

ALGORITHM 2: Algorithm for M-CNMF.

$$\text{AUC} = \frac{n^0 + 0.5n^*}{n}. \quad (16)$$

The missing links fraction f ranges from 0.05 to 0.95, and the interval is set at 0.05.

5.2. Methods Used to Compare. We compare our algorithm with three prediction methods: Common Neighbors, Block Model, and Hierarchical Random Graphs.

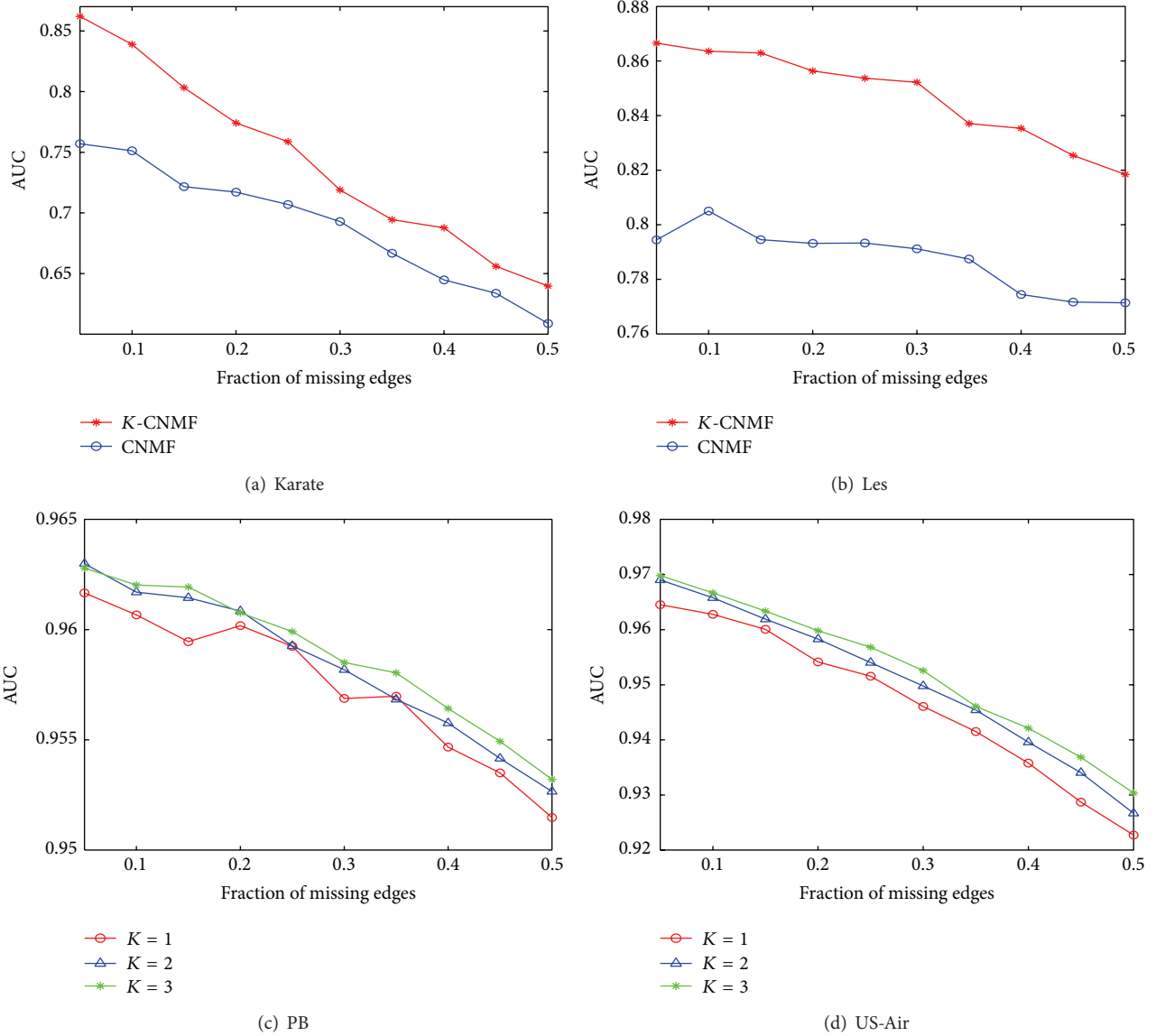


FIGURE 2: Comparison of K -CNMF for different K .

(1) *Common Neighbors (CN)* [2]. If two nodes, a and b have many common neighbors, they are more likely to have a link. The measure of this is

$$s_{ab} = \|\tau(a) \cap \tau(b)\|, \quad (17)$$

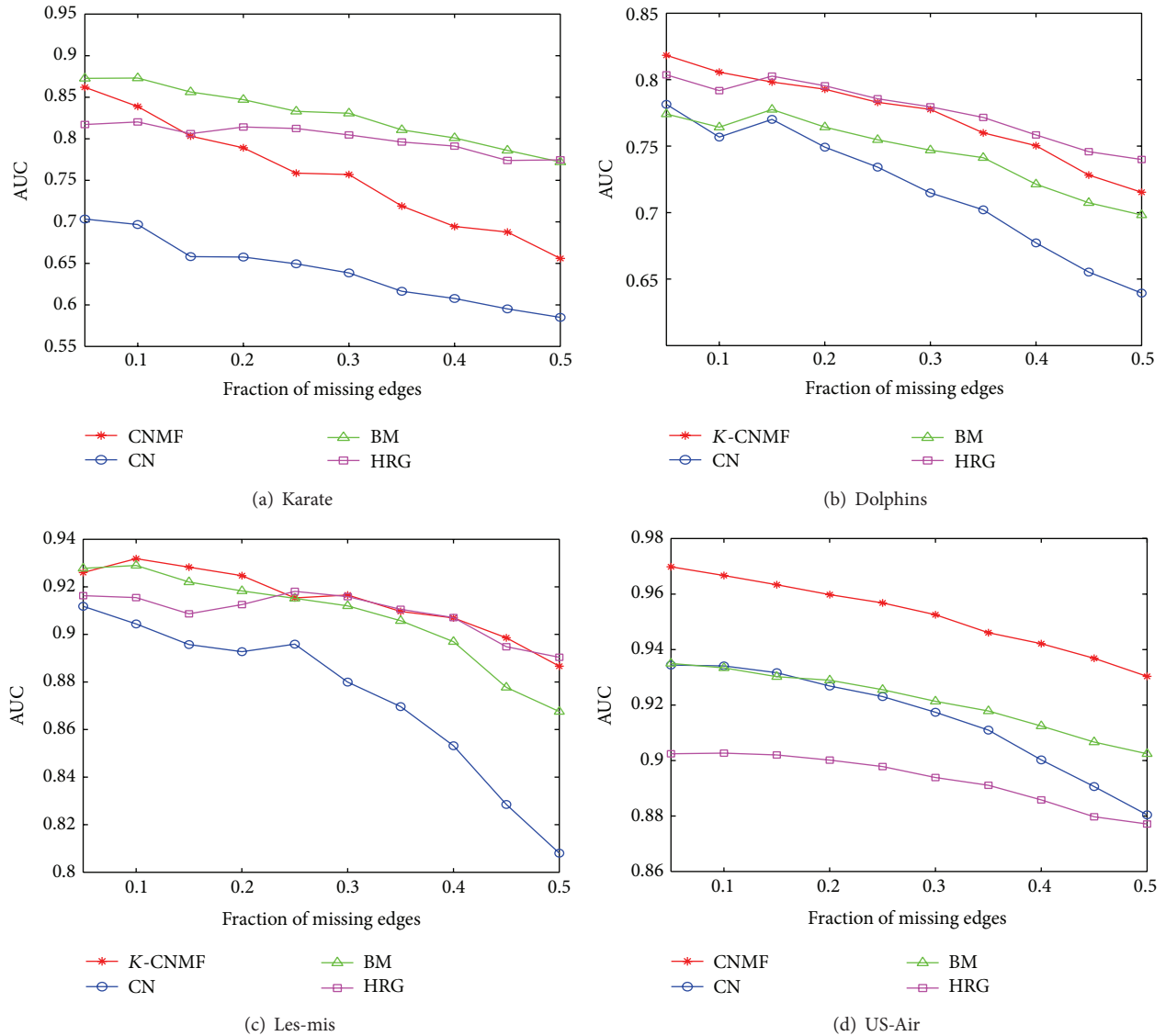
where $\tau(a)$ is the set of neighbors of a .

(2) *Block Model (BM)* [4]. In block models, nodes are partitioned into groups and the connecting probability of two nodes only depends on the groups they belong to. Given a partition P of the network, $l_{\alpha\beta}^o$ is the number of edges in the observed network between nodes in groups α and β , and $r_{\alpha\beta}$ is the maximum possible number of links between α and β . The reliability of an individual link is

$$p(A_{ij} = 1 | A^o) = \frac{1}{Z} \sum_{P \in \mathcal{P}} \left(\frac{l_{\sigma_i \sigma_j}^o + 1}{r_{\sigma_i \sigma_j} + 2} \right) \exp(-\mathcal{H}(P)), \quad (18)$$

where the sum is over partitions P in the space \mathcal{P} of all possible partitions of the network, σ_i is node i 's group, $\mathcal{H}(P) = \sum_{\alpha \leq \beta} (\ln(r_{\alpha\beta} + 1) + \ln C_{r_{\alpha\beta}}^{l_{\alpha\beta}^o})$, and $Z = \sum_{P \in \mathcal{P}} \exp(-\mathcal{H}(P))$.

(3) *Hierarchical Random Graphs (HRG)* [3]. The hierarchical structure of a network can be represented by a dendrogram with n leaves (the vertices from the given network) and $n - 1$ internal nodes. A probability p_r is associated with internal node r and the connecting probability of a pair of leaves is equal to p_r , where r is the deepest common ancestor of these two leaves. HRG combines the maximum likelihood method and Markov chain Monte Carlo method to sample the hierarchical structure with probability proportional to their likelihood from the observed network and then calculate p_r .

FIGURE 3: Comparison of K -CNMF with CN, BM, and HRG.

5.3. Performance of K -CNMF. We evaluate the performance of K -CNMF using four high-quality small networks, and they are listed in Table 1: the social network of interactions between people in a karate club [22], the social network of frequent associations between dolphins [23], the air transportation network of USA, the coappearance network of characters in the novel Les Miserables [24], and a network of hyperlinks between weblogs on US politics [25]. Each AUC is obtained by averaging over 100 independent realizations.

Communities are basic structure in networks, which is widely used to predict missing links. Using block structure, our combined method is also dependent on communities of the networks. As Figures 2(a) and 2(b) show, K -CNMF ($L = 2$, $K = 3$) performs much better than CNMF ($L = 2$) alone on Karate and Les-Mis, because both of those networks have more than two communities. However the enhancement by block structure is small on PB with desired rank $L = 2$ (see Figure 2(c)), which has two main communities. As the

TABLE 1: Networks.

Name	Nodes	Edges	Average degree
Karate club	34	78	4.588
Dolphins	62	159	5.129
Les Miserables	77	254	6.597
Politics weblogs	1490	19025	22.436
US-Air 97	332	2126	12.807
Power	4941	6594	2.669

desired rank increases, the enhancement by block structure decreases. That is because the local information can be revealed by the richness of clustering structures given by CNMF with high desired rank. So the enhancement is also small on US-Air (nodes: 332) with desired rank $L = 300$ (see Figure 2(d)).

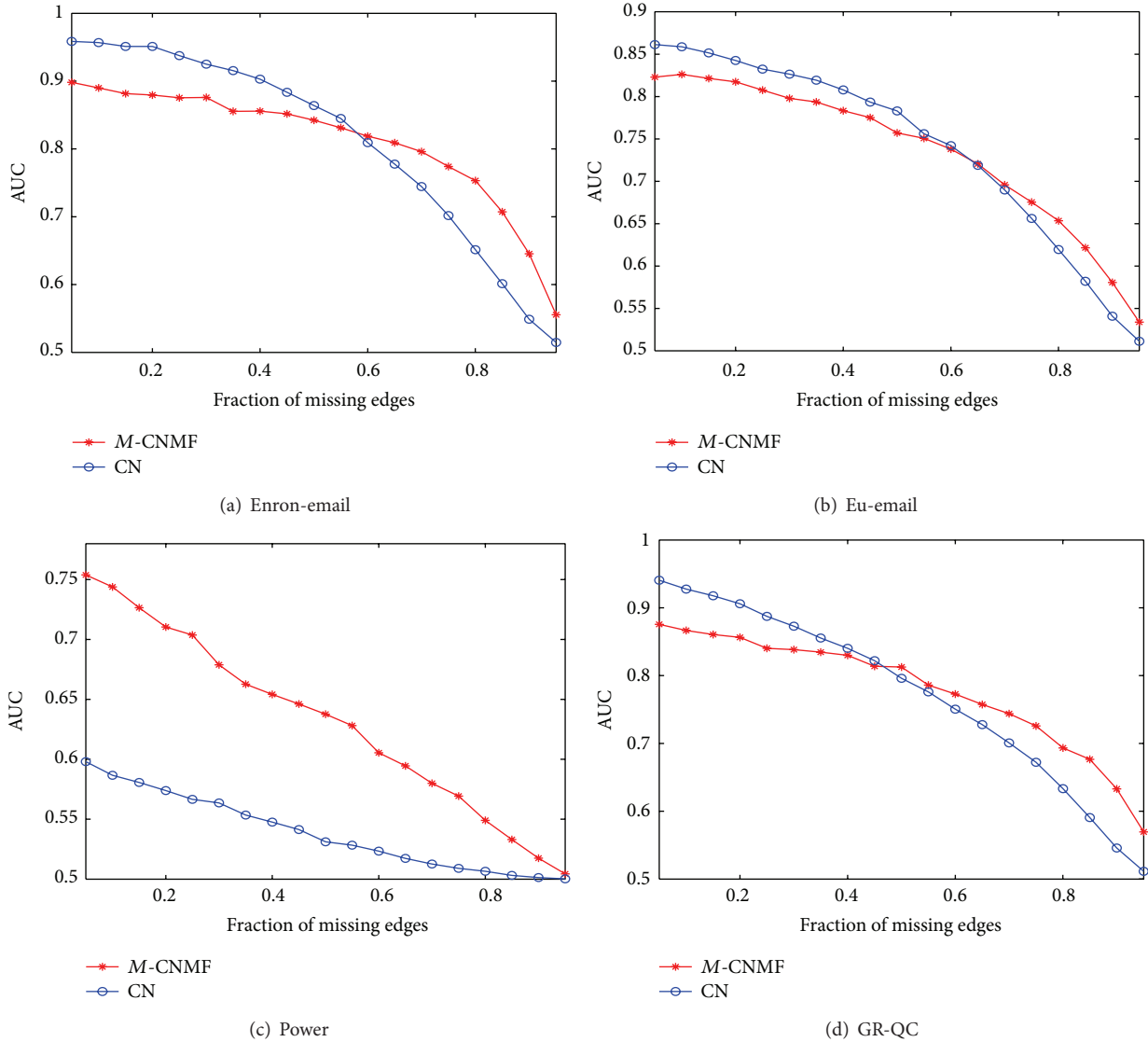


FIGURE 4: Comparison of M -CNMF with CN.

Will more block information usage bring more accuracy?

If partitioning matrix into too small blocks, K -CNMF will have too many parameters relative to the observed data, and then overfitting will occur. An overfitted model describes noise instead of the underlying relationship and generally has poor predictive performance. From the performance of K -CNMF ($L = 62$) with different K on Dolphins (see Table 2), we can see that K -CNMF ($L = 62, K = 1$) has revealed enough local information, and increasing K caused overfitting.

Figure 3 shows the comparison of K -CNMF with CN, BM, and HRG on Karate (inputting parameters of K -CNMF: $K = 3, L = 34$) and Dolphins ($K = 1, L = 62$), Les-Mis ($K = 40, L = 50$), and US-Air ($K = 3, L = 300$). K -CNMF performs better than CN, because it concerns the property of both local and global information. The performance of

TABLE 2: Results for predicting missing links: AUC of K -CNMF on Dolphins with $K = 1, 2, 3, L = 62$.

f	$K = 1$	$K = 2$	$K = 3$
0.05	0.818327	0.809745	0.799961
0.10	0.805632	0.802360	0.795246
0.15	0.798197	0.793669	0.786697
0.20	0.792776	0.786535	0.778436
0.25	0.782948	0.77611	0.773828
0.30	0.777660	0.769801	0.758317
0.35	0.759993	0.759392	0.761205
0.40	0.750356	0.742400	0.732967
0.45	0.728203	0.725031	0.717759
0.50	0.715353	0.720560	0.705817

K -CNMF is comparable with BM and HRG, but faster, as it does not need Monte Carlo samplings.

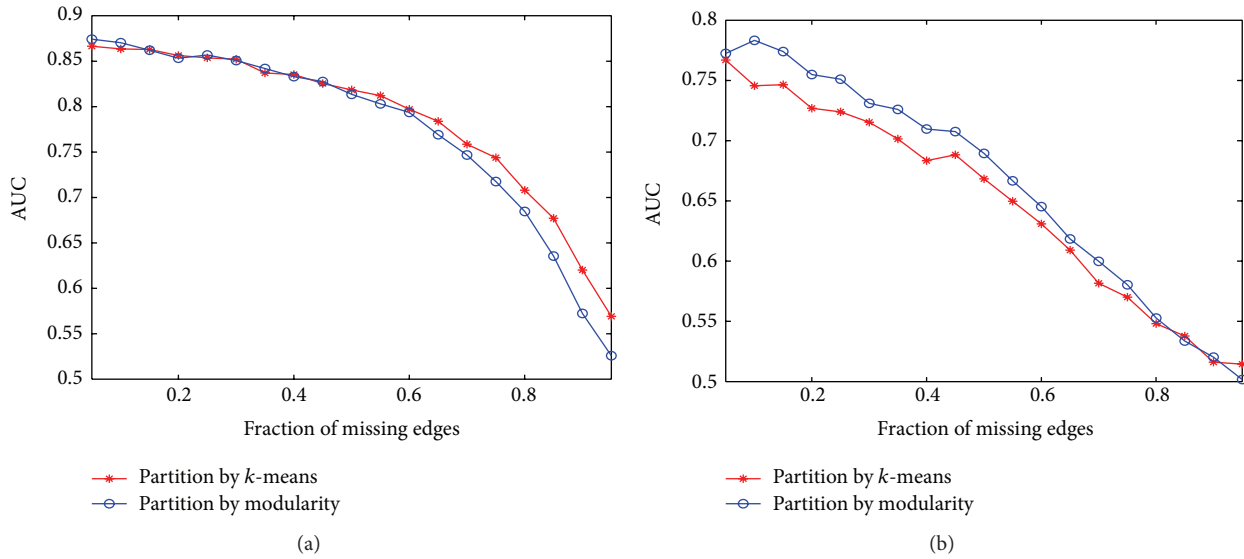


FIGURE 5: Comparison of partition methods.

5.4. Performance of M -CNMF. We examine the performance of M -CNMF by the main components of four real-world networks: Arxiv GR-QC collaboration network (inputting parameters: $L = 2$, $K = 1000$) [26], the Western States Power Grid of the United States ($L = 2$, $K = 250$) [27], Enron email network ($L = 2$, $K = 2000$) [28], and the subnetwork of EU email communication network generated by email form the first 5000 nodes to first 10000 nodes ($L = 2$, $K = 2000$) [26].

Comparisons are made only between M -CNMF and CN, for the reason that BM and HRG are not suitable for large networks. Figure 4 shows the comparison of M -CNMF with CN. The performances of M -CNMF are better than CN when the observed networks are much sparse, because common neighbors miss too much in sparse case and CN only concerns this property. In the power network, our method is obviously better than CN, because the original network is sparse.

Figures 5(a) and 5(b) show the comparison of AUC results between M -CNMF and K -CNMF on Karate and Dolphins, respectively, where $K = 2$, $L = 2$. There are no obvious rules that different modularity clustering algorithms will influence the results of AUC.

6. Conclusions

We have introduced a probabilistic latent variable model for finding missing edges, which combines convex nonnegative matrix factorization with block structures detection. It is inspired by two properties of block structure for matrices: the facts that entities in the same block tend to be similar and that good block detection algorithms have tolerance to missing edges. Scaling to massive size network, we use fast modularity clustering algorithm to generate blocks and give a divide-and-conquer algorithm (M -CNMF) for predicting links. For the load balancing of CPU, we split the large blocks and combine the small blocks to make their sizes in the neighborhood of a given threshold.

Since most applications of link prediction are facing the problems of sparse data, such as personalized recommendation, we plan to combine other sparse low rank approximation algorithms with block detection methods to get effective link prediction algorithms for massive networks in the future.

Conflict of Interests

The authors declare that they do not have any commercial or associative interest that represents a conflict of interests in connection with the work.

References

- [1] L. A. Adamic and E. Adar, "Friends and neighbors on the Web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.
- [2] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, Article ID 025102, 2001.
- [3] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.
- [4] R. Guimerà and M. Sales-Pardo, "Missing and spurious interactions and the reconstruction of complex networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 52, pp. 22073–22078, 2009.
- [5] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.
- [6] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proceedings of the 12th ACM International Conference on Information and Knowledge Management (CIKM '03)*, pp. 556–559, ACM, November 2003.
- [7] L. Lü and T. Zhou, "Link prediction in complex networks: a survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

- [8] Z. Xie, E. Dong, J. Li, D. Kong, and N. Wu, "Potential links by neighbor communities," *Physica A*, vol. 406, pp. 244–252, 2014.
- [9] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [10] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, "Classes of complex networks defined by role-to-role connectivity profiles," *Nature Physics*, vol. 3, no. 1, pp. 63–69, 2007.
- [11] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI '99)*, pp. 1300–1307, Stockholm, Sweden, August 1999.
- [12] D. Heckerman, C. Meek, and D. Koller, "Probabilistic entity-relationship models, PRMs, and plate models," in *Proceedings of the 21st International Conference on Machine Learning*, p. 55, Banff, Canada, 2004.
- [13] K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu, "Stochastic relational models for discriminative link prediction," in *Proceedings of the Neural Information Processing Systems*, p. 1553, MIT Press, Cambridge, Mass, USA, 2006.
- [14] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*, pp. 243–252, July 2010.
- [15] B. Savas and I. S. Dhillon, "Clustered low rank approximation of graphs in information science applications," in *Proceedings of the 11th SIAM International Conference on Data Mining (SDM '11)*, pp. 164–175, April 2011.
- [16] A. Menon and C. Elkan, "Link prediction via matrix factorization," in *Proceedings of the 2011 ECML PKDD European Conference*, pp. 437–452, Athens, Greece, September 2011.
- [17] D. M. Dunlavy, T. G. Kolda, and E. Acar, "Temporal link prediction using matrix and tensor factorizations," *ACM Transactions on Knowledge Discovery from Data*, vol. 5, no. 2, article 10, 2011.
- [18] D. Shin, S. Si, and I. S. Dhillon, "Multi-scale link prediction," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12)*, pp. 215–224, ACM, November 2012.
- [19] C. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2010.
- [20] U. Brandes, D. Delling, M. Gaertler et al., "On modularity clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 172–188, 2008.
- [21] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Machine Learning*, vol. 45, no. 2, pp. 171–186, 2001.
- [22] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, pp. 452–473, 1977.
- [23] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Sloaten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: can geographic isolation explain this unique trait?" *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.
- [24] D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing*, Addison-Wesley, Reading, Mass, USA, 1993.
- [25] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 US Election," in *Proceedings of the Workshop on the Weblogging Ecosystem (WWW '05)*, 2005.
- [26] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, Article ID 1217301, 2007.
- [27] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [28] B. Klimmt and Y. Yang, "Introducing the Enron corpus," in *Proceedings of the 1st Conference on Email and Anti-Spam (CEAS '04)*, 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

