placed into one cluster or placed into different clusters randomly. The smaller value of Q_d means the better quality of clustering.

Similar to [7, 12], if we combine any two modules C_s and C_t, the distance-based modularity gain $\triangle Q_d$ achieved from the combination can be computed by

$$\triangle Q_d = Q_d^{C_s \cup C_t} - Q^{C_s} - Q^{C_t} = \frac{2D_{st}^U}{D^T} - \frac{2D_s^C D_t^C}{(D^T)^2} \quad (9)$$

where $D_{st}^U = \sum_{u \in C_s, v \in C_t} d_{\mathcal{M}}(u, v)$ is the sum of distance between any two nodes in modules C_s and C_t respectively.

According to Equation (9), we compute the gain of distance-based modularity $\triangle Q_d$ for combing j clusters $C_1, C_2, ..., C_j$ into a new community by

$$\triangle Q_d(j) = \frac{\sum_{s,t \in \{1,...,j\}, s \neq t} 2D_{st}^U}{D^T} - \frac{\sum_{s,t \in \{1,...,j\}, s \neq t} 2D_s^C D_t^C}{(D^T)^2} \quad (10)$$

5.2 Clustering Algorithm

Before addressing our algorithm in detail, we give the following definitions.

Definition 6 (Nearest Neighbor) *Given an incomplete information network $G = (V, E, A, \mathcal{M})$, the nearest neighbor set for $\forall v \in V$ is defined as*

$$NN(v) = \{y | y = \arg\min_x d_{\mathcal{M}}(v, x), x \in V \wedge x \neq v\}. \quad (11)$$

Definition 7 (Mutual Nearest Neighbor) *Given an incomplete information network $G = (V, E, A, \mathcal{M})$, any pair of nodes $u, v \in V$ is said to be mutual nearest neighbor, denoted by $u \overset{\gamma}{\leftrightarrow} v$, i $iff v \in NN(u) \wedge u \in NN(v) \wedge d_{\mathcal{M}}(u, v) = \gamma$, where $\gamma \in \mathcal{R}^+$.*

Definition 8 (Local Community) *Given an incomplete information network $G = (V, E, A, \mathcal{M})$, we call the subgraph $C(v) = (V', E', A', \mathcal{M}, \gamma)$ of G as a local community iff* (1) *$v \in V'$;* (2) *$\forall u \in V', \exists v \in V' \wedge (u \overset{\gamma}{\leftrightarrow} v)$;* (3) *$\{u | u \in V' \wedge u \overset{\gamma}{\leftrightarrow} v \wedge v \notin V'\} = \emptyset$. $\gamma \in \mathcal{R}^+$ is the radius of the local community $C(v)$.*

The distance-based shrinking approach DSHRINK is presented in Figure 3. Our approach can be divided into two phases. At the first phase, we compute the distance between any pair of nodes in graph G and store the distance as a 3-tuple $(v_i, v_j, d_{\mathcal{M}}(v_i, v_j))$ into a map structure (see Figure 2). For any $v_i \in V$, the sum of the distance between node v_i and any other node $v_j \in V$ is saved in S_i^T. Since $D_s^C = \sum_{v_i \in C_s, v_j \in V} d_{\mathcal{M}}(v_i, v_j) = \sum_{v_i \in C_s} S_i^T$, computing S_i^T in advance can speed up computing D_s^C when computing Q_d. For the same purpose, the total distance D^T between any pair of node is also be computed.

At the second phase, (1) we first begin at an arbitrary node and span the node to a local community based on Definition 8. All the nodes, which are in the local community, will be tagged as "visited". Then, we choose the next unvisited node in graph G and repeat the above step. This process will not stop until all the nodes are visited. (2) Secondly, for each local community discovered by the first step, we view each single node in it as a community. Then $\triangle Q_d$ is computed according to Equation (10). If $\triangle Q_d < 0$,

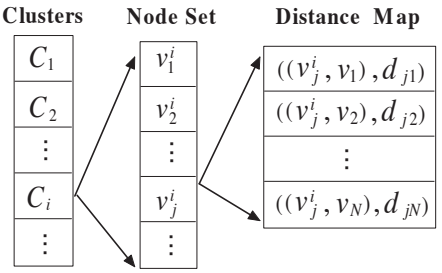


Figure 2: Data structure used in DSHRINK

which means the combination of the communities can decrease the total distance-based modularity Q_d, we shrink the local community as a super node. Otherwise, the local community will not be shrunk. (3) Thirdly, we tag all the nodes including super nodes and the common nodes as "unvisited" and repeat the first and second steps. The above steps will be repeated many times until shrinking any local maximal community can not decrease the Q_d any more. Finally, the nodes condensed in a super node form a community, and different super nodes stand for different communities.

According to the definition of local community, we know that the order of traversing the nodes in the incomplete information network G does not change the final members of the local community. Moreover, from the clustering process, we know that the local community, in which nodes have a shorter distance, will be shrunk at the prior or the same iteration than the local community, in which nodes have a longer distance. Single nodes, which have not been shrunk to any other super nodes, are viewed as hubs or outliers depending on how many communities they are close to. If we want to form the overlapped communities, the hub nodes will be placed into more than one communities. Otherwise, each hub node will only be placed into the community which makes the most decrease of Q_d by adding the hub node. If we view the distance between each pair of nodes as the structure similarity, the above shrinking process is similar to [12].

5.3 Speeding up the Clustering Process with Approximation

It is possible to speed up the clustering process by allowing some approximation in the determination of the local community. We define ϵ-approximate mutual nearest neighbor and ϵ-approximate local community as follows:

Definition 9 (ϵ-approximation Mutual Nearest Neighbor) *Given an incomplete information network $G = (V, E, A, \mathcal{M})$, any pair of nodes $u, v \in V$ is said to be ϵ-approximation mutual nearest neighbor in G, denoted by $u \overset{\epsilon}{\leftrightarrow} v$, iff $(v \in NN(u) \wedge |d_{\mathcal{M}}(u, v) - d_{\mathcal{M}}(v, x)| \leq \epsilon) \vee (u \in NN(v) \wedge |d_{\mathcal{M}}(u, v) - d_{\mathcal{M}}(u, y)| \leq \epsilon)$, where $x \in NN(v)$, $y \in NN(u), \epsilon \in \mathcal{R}^+$.*

Obviously, ϵ-approximate mutual nearest neighbor is an extended version of mutual nearest neighbor.

Definition 10 (ϵ-approximation Local Community) *Given an incomplete information network $G = (V, E, A, \mathcal{M})$, $C(v) = (V', E', A', \mathcal{M}, \epsilon)$ is a subgraph of network G. $C(v)$ is said to be a ϵ-approximation local community of G iff* (1)

DSHRINK($G = (V, E, A, \mathcal{M})$)

Input:
$G = (V, E, A, \mathcal{M})$: Incomplete information network.
Output:
$C = \{C_1, C_2, ..., C_k\}$: Cluster set.
HO : Hubs and outliers.
Process:

```
1  Initialize each v_i ∈ V as a community and put it in C;
2  for each v_i ∈ V do
3    for each v_j ∈ V ∧ v_i ≠ v_j do
4      Compute d_M(v_i, v_j) according to Equation 4;
       Store (key(v_i, v_j), value(d_M(v_i, v_j))) with
       ascending order into the distance map.
5      S_i^T += d_M(v_i, v_j);
6      D^T += d_M(v_i, v_j);
7    end
8  end
9  while true do
10   for each v_i ∈ V do
11     if v_i.visited then continue
12     Span a local community C(v_i) according to
       Definition 8;
13     for each v_j ∈ V(C(v_i)) do
14       v_j.visited = ture;
15     end
16     C ← C ∪ C(v_i);
17   end
18   Q_d.descrease = false;
19   for each C_j ∈ C do
20     Compute △Q_d according to Equation 10;
21     if △Q_d < 0 then
22       v_s ← V(C_j);
23       C ← (C − C_j) ∪ v_s;
24       Q_d += △Q_d;
25       Q_d.descrease = true;
26       v_s.visited = false;
27     end
28   end
29   if !(Q_d.descrease) then break;
30 end
31 Get single nodes from C and put them into HO
32 return C, HO;
```

Figure 3: The Description of DSHRINK

$v \in V'$; (2) $\forall u \in V', \exists v \in V' \wedge (u \underset{\epsilon}{\leftrightarrow} v)$; (3) $\{u | u \in V' \wedge u \underset{\epsilon}{\leftrightarrow} v \wedge v \notin V'\} = \emptyset$; (4) *let* $f(r) = \{r | r = d_{\mathcal{M}}(s,t), s \underset{\epsilon}{\leftrightarrow} t \wedge s \in V' \wedge t \in V'\}$, $|Max(f(r)) - Min(f(r))| \leq \epsilon$; (5)*when* (3) *and* (4) *can not be held at the same time,* (4) *is prior to* (3) *to be guaranteed.* $\epsilon, r \in \mathcal{R}^+$.

We note that this relaxation of the definition of local community can greatly speed up the clustering process. In order to take advantage of ϵ-approximation local community, the only difference in DSHRINK is to span a ϵ-approximation local community instead of a local community in step (12). When we span the ϵ-approximation local community ($\epsilon > 0$), the final clustering result may rely on the visiting sequence of the nodes. In this paper, we give priority to the shorter distance nodes among all of the ϵ-approximation neighbours when spanning the ϵ-approximation local communities. Our experimental results show that the final clustering effect is almost not affected by the order of the visiting sequence of nodes by taking the above strategy. Furthermore, given an appropriate parameter ϵ, we find that this relaxation does not affect the practical quality of the communities obtained.

Table 1: Summary of experimental data sets

Dataset	# Nodes	# Links	# Attributes	# Classes
DBLP-A	4638	16,447	102	6
DBLP-B	4559	14,407	102	6

6. EXPERIMENTS

In this section, we use two real-world data sets to validate the effectiveness and efficiency of our approach. All the experiments are conducted on a machine with Intel 8-core 2.7 GHz processors and 28GB memory.

6.1 Data Sets

DBLP-A Dataset: DBLP-A is the data set extracted from DBLP database[1] which provides bibliographic information on computer science journals and proceeding. We extract paper information from 16 top conferences which cover 6 research fields including *Artificial Intelligence*, *Information Retrieval*, *Computer Vision*, etc. We create the coauthor network by choosing authors, who published at least 2 papers during $2000 - 2010$, as the nodes of the network. Any pair of authors who have coauthored are linked in this coauthor network. This coauthor network contains 4638 nodes and 16447 links in total. Each node is attached with a bag-of-words which extracted from the paper titles published by him/her. We first apply the standard text preprocessing such as stemming, stop words removal. Then we reduce the dimension of the bag-of-words to 100 by PCA and use them as the features of the corresponding node. In addition, the number of co-authors and publications are also used as features of the nodes.

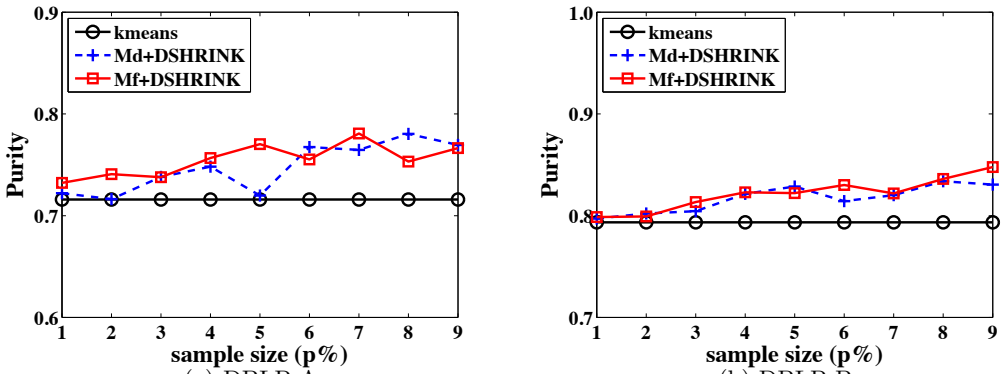
DBLP-B Dataset: We also extract paper information from 16 top conferences of 6 research fields such as *Algorithms & Theory*, *Natural Language Processing*, *Bioinformatics*, etc. The same setups with DBLP-A are also used here to build the coauthor network as our second data set, called DBLP-B.

We summarize our data sets in Table 1.

6.2 Incomplete Information Network Generation

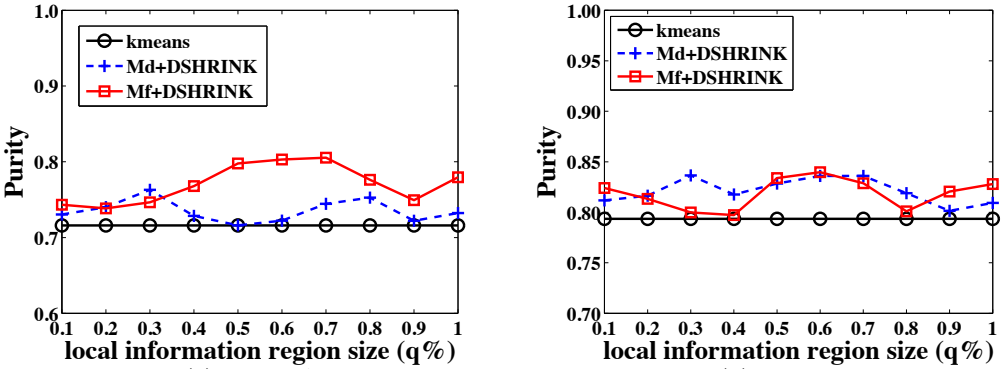
In order to simulate the incomplete information networks with local information regions, we use the following experiment setting. If we perform random sampling on the nodes, the sampled network usually ends up being sparsely connected, without local information regions. In this paper, we use the snowball sampling [23] to sample a group of connected local region at a time. We randomly sample one node and use BFS to include its neighboring nodes into the sampled region until a fixed number of nodes are sampled. We repeat this process until a number of local regions are sampled. Then we assume the links within the local informative regions are available to the algorithms, while the remaining links in the network are removed. In order to control the total number of nodes being sampled, we introduce a parameter p, called *sample ratio*, i.e., the ratio of the nodes in the network being sampled into the local region. In addition, we introduce another parameter q, called *local information region size*, to control the size of each local information region. In detail, we first randomly choose a node

[1]http://www.informatik.uni-trier.de/∼ley/db/



(a) DBLP-A (b) DBLP-B

Figure 4: Accuracy comparison between different methods ($q\% = 0.3\%$).



(a) DBLP-A (b) DBLP-B

Figure 5: Accuracy comparison between different methods ($p\% = 10\%$).

in the network. We then include $q\%$ nodes from its neighbors using BFS search. Common neighbors of any pair of nodes in the sampled region are further included into the sampled local region. The above sampling process continues until we sample $p\%$ of the nodes in the network. In addition to the local regions, we sample the same number of nodes and use them to generate dissimilar pairwise constraints. In the sampled group, the pairs of nodes that are in different classes are then used as the dissimilar node-pair set $\mathcal{D}$. More concretely, for DBLP-A and DBLP-B datasets, we choose the pair of authors, whose research fields are not overlapped as the dissimilar node pair.

6.3 Evaluation Measures

In order to measure the effectiveness of our approach, we adopt *Purity* to evaluate the quality of the communities generated by different approaches. The definition of purity is as follows: each cluster is first assigned with the most frequent class in the cluster, and then the purity is measured by computing the number of the instances assigned with the same labels in all clusters. Formally:

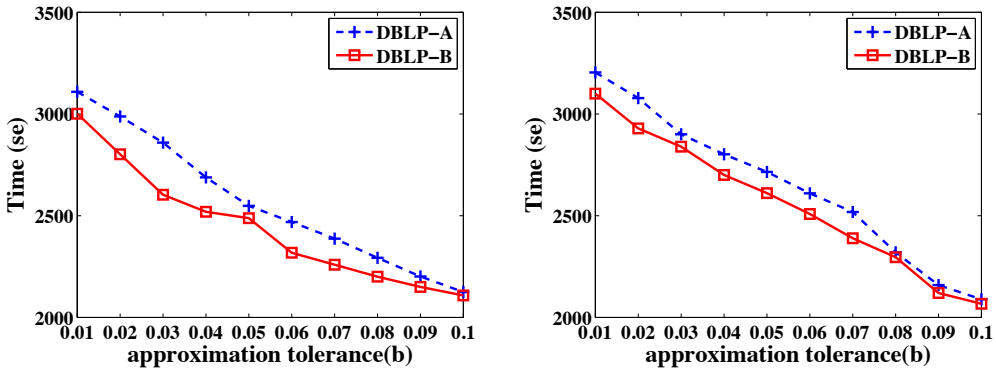
$$\text{Purity} = \frac{1}{n}\sum_{i=1}^{k}\max_{j}|C_i \cap l_j| \tag{12}$$

where $\{C_1, \cdots, C_k\}$ is the set of clusters, l_j is the j-th class label. The value of purity ranges from 0 to 1. The community structure generated by each compared method will be evaluated using the true label of each node such that the higher purity value means the higher accuracy of the method. Since each author can have multiple research areas as its class labels. We computed the purity of the clustering results based on each label separately, and the average results over 6 labels are reported.

6.4 Compared Methods

In order to demonstrate the effectiveness and efficiency of our approach, we compare our approach with the following methods:

- Kmeans: We use the default Euclidean metric to measure the distance between any node x_i and the centroid x_k. The K value used in the dataset of DBLP-A and DBLP-B is 6, which is the same number of clusters with the ground truth.
- $\mathcal{M}_d$+ DSHRINK: We learn a diagonal Mahalanobis matrix $\mathcal{M}_d$ and use it as the input of $\mathcal{M}$ for DSHRINK.
- $\mathcal{M}_f$+ DSHRINK: We learn a full Mahalanobis matrix $\mathcal{M}_f$ and use it as the input of $\mathcal{M}$ for DSHRINK.



(a) $\mathcal{M}_f$+DSHRINK (b) $\mathcal{M}_d$+DSHRINK

Figure 6: The computation time with the different values of b.

6.5 Effectiveness Results

The variation of purity scores under different values of $p\%$ is given in Figure 4. In this experiments, $q\% = 0.3\%$ is used. Since the accuracy of KMeans is not affected by the number of sampling nodes, the purity value of the KMeans is a horizontal line in all cases. We notice that the purity scores of $\mathcal{M}_f$+DSHRINK and $\mathcal{M}_d$+DSHRINK ascend quickly with the increasing number of local information regions sampled. Especially, when $p\% > 2\%$, the purity values of $\mathcal{M}_f$+DSHRINK and $\mathcal{M}_d$+DSHRINK exceed kmeans over all data sets. That is because, firstly, the learned Mahalanobis matrix $\mathcal{M}$ rescales all of the nodes into a new feature space, where the similar nodes are closer, and the dissimilar nodes are further away than the original Euclidean space. Secondly, DSHRINK can automatically detect the most appropriate number of communities by minimizing the distance-based modularity. Since the number of communities in the networks is unknown, $\mathcal{M}_f$+DSHRINK and $\mathcal{M}_d$+DSHRINK have more advantage for discovering the most appropriate community structures than Kmeans. Another observation is that, in most of the cases, with the same value of $p\%$, the purity scores of $\mathcal{M}_f$+DSHRINK are a little higher than $\mathcal{M}_d$+DSHRINK in both DBLP-A and DBLP-B data sets. This demonstrates that the full Mahalanobis matrix performs better rescaling function for separating the similar nodes from the dissimilar nodes than diagonal Mahalanobis matrix in DBLP-A and DBLP-B data sets. However, this principle dose not not always hold. For instance, in DBLP-A data set, the purity score of $\mathcal{M}_f$+DSHRINK is less than $\mathcal{M}_d$+DSHRINK when $p\% = 8\%$.

In Figure 5, given a specified value of $p\% = 10\%$, we also present the changes of purity scores with the different value of $q\%$. We notice that, on the one hand, for a specified $p\% = 10\%$, the larger value of $q\%$ makes more similar node pairs be captured in each local information region, but fewer local information regions get chosen in the whole incomplete information network. On the other hand, with the smaller value of $q\%$, fewer similar node pairs can be captured in each local information region, but more local information regions can be sampled. Since both the number of local information regions and the number of similar node pairs can affect the learning of the metric, finding the balance point of $q\%$ is critical for achieving a better clustering result. From Figures 5 (a) to (b), we know that the balance point of $q\%$ can be gotten between 0.5% to 0.7%.

6.6 Efficiency Results

We notice that, computing the optimal Mahalanobis matrix and the distance between any pair of nodes can be accomplished in advance before clustering process. In this part, we mainly focus on the clustering process and test how ϵ-approximation local community speeds up the clustering process and affects the quality of clusters. The distance here is relative and changeable according to different values of w in Equation (5). Hence, discussing the value of ϵ is meaningless with a special value of w. Fortunately, we find the top k nearest nodes of each node is a good base for us to compute the appropriate ϵ value. In order to compute an appropriate ϵ value, we average the sum of distance between each node and its corresponding top k nearest nodes as follows:

$$d = \frac{\sum_{i=1}^{N} \sum_{j \in TopK(i)} d_{\mathcal{M}}(v_i, v_j)}{k|N|} \tag{13}$$

where $TopK(i)$ is the set of node index, whose distance to v_i ranks in the top k among all of the nodes to v_i, and $|N|$ is the total number of nodes in the incomplete information network. In this paper, we choose $k = 10$ and give the value of ϵ as $\epsilon = d \times b$. For a specified incomplete information network G and a metric $\mathcal{M}$, the value of d is a constant. Therefor, changing the value of b is equivalent to change the value of ϵ.

In Figures 6(a) to (b), we have illustrated the variation in the efficiency of different b values for $\mathcal{M}_f$+DSHRINK and $\mathcal{M}_d$+DSHRINK. We observe that the computation time decreases quickly with the increasing value of b. It is because relaxing the definition of local community to a certain extent can decrease the iteration times in the clustering process. We also find that the purity score are not changed dramatically with differenct b values.

7. CONCLUSION

In this paper, we presented the first approach for community detection in incomplete information networks with local information regions. While the traditional community detection algorithms make the assumption of the full knowledge of linkage information, they can not solve the problem

of community detection in incomplete information networks. In order to resolve this problem, we explored the metric learning idea and learned a global metric from the side information of the incomplete information network. Moreover, we proposed the distance-based modularity function. Based on this function, we further devised a distance-based clustering algorithm DSHRINK. In order to speed up the clustering process, some helpful approximation strategies were also proposed. Experimental results illustrated the effectiveness and efficiency of our approach.

8. ACKNOWLEDGMENTS

The first author is supported by National Natural Science Foundation of China through grants 60933005, 60873204 and the National High-Tech Program through grant 2010AA012505. The third author is supported in part by US NSF through grants IIS 0905215, DBI-0960443, and Google Mobile 2014 Program.

9. REFERENCES

[1] G. Agarwal and D. Kempe. Modularity-maximizing graph communities via mathematical programming. *European Physical Journal B*, 66:409–418, 2008.

[2] C. Aggarwal, Y. Xie, and P. Yu. Towards community detection in locally heterogeneous networks. SDM, pages 391–402, 2011.

[3] Y. Ahn, J. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466:761–764, 2010.

[4] H. Alani, S. Dasmahapatra, K. O'Hara, and N. Shadbolt. Identifying communities of practice through ontology network analysis. *Intelligent Systems*, 18(2):18–25, 2003.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] T. Evans and R. Lambiotte. Line graphs, link partitions and overlapping communities. *Physical Review E*, 80:016105, 2009.

[7] Z. Feng, X. Xu, N. Yuruk, and T. A. J. Schweiger. A novel similarity-based modularity function for graph partitioning. In *DaWak*, pages 385–396, 2007.

[8] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proceedings of The National Academy of Sciences*, 104(1):36–41, 2007.

[9] R. Ge, M. Ester, B. J. Gao, Z. Hu, B. K. Bhattacharya, and B. Ben-moshe. Joint cluster analysis of attribute data and relationship data: The connected k-center problem, algorithms and applications. *ACM Transactions on Knowledge Discovery From Data*, 2:1–35, 2008.

[10] B. H. Good, Y. A. de Montjoye, and A. Clauset. The performance of modularity maximization in practical contexts. *Physical Review E*, 81:046106, 2010.

[11] R. Guimera and L. N. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005.

[12] J. H. H. D. Y. S. Y. L. J. Huang, H. Sun. SHRINK: a structural clustering algorithm for detecting hierarchical communities in networks. In *CIKM*, pages 219–228, 2010.

[13] M. Kim and J. Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *SDM*, pages 47–58, 2011.

[14] J. Z. Z. N. L. Tang, H. Liu. Community evolution in dynamic multi-mode networks. KDD, pages 677–685, 2008.

[15] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5):056117, 2009.

[16] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Statistical properties of community structure in large social and information networks. WWW, pages 695–704, 2008.

[17] J. Leskovec, K. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. WWW, pages 631–640, 2010.

[18] Y. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. Metafac: community discovery via relational hypergraph factorization. KDD, pages 527–536, 2009.

[19] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[20] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814, 2005.

[21] M. Rosvall and C. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105:1118, 2008.

[22] M. Sales-Pardo, R. Guimerà, A. Moreira, and L. Amaral. Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences*, 104(39):15224–15229, 2007.

[23] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[25] M. Shiga, I. Takigawa, and H. Mamitsuka. A spectral clustering approach to optimally combining numerical vectors with a modular network. KDD, pages 647–656, 2007.

[26] Y. Tian, R. Hankins, and J. Patel. Efficient aggregation for graph summarization. In *SIGMOD*, pages 567–580, 2008.

[27] C. Tsai and C. Chiu. Developing a feature weight self-adjustment mechanism for a k-means clustering algorithm. *Computational Statistics Data Analysis*, 52:4658–4672, 2008.

[28] K. Wakita and T. Tsurumi. Finding community structure in mega-scale social networks. In *WWW*, pages 1275–1276, 2007.

[29] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, pages 505–512, 2002.

[30] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. Scan: A structural clustering algorithm for networks. In *KDD*, pages 824–833, 2007.

[31] Y. Zhou, H. Cheng, and J. Yu. Graph clustering based on structural/attribute similarities. *VLDB Endowment*, 2:718–729, 2009.