User-guided Large Attributed Graph Clustering with Multiple Sparse Annotations

Jianping Cao[1,*], Senzhang Wang[2,**], Fengcai Qiao[1], Hui Wang[1], Feiyue Wang[1] and Philip S. Yu[3]

[1]College of Information Systems and Management,
National University of Defense Technology, Changsha, Hunan, China
[2]State Key Laboratory of Software Development Environment,
Beihang University, Beijing, China
[3]Department of Computer Science, University of Illinois at Chicago, IL, USA

Abstract. One of the key challenges in large attributed graph clustering is how to select representative attributes. Previous studies introduce user-guided clustering methods by letting a user select samples based on his/her knowledge. However, due to knowledge limitation, a single user may only pick out the samples that s/he is familiar with while ignore the others, such that the selected samples are often biased. We propose a framework to address this issue which allows multiple individuals to select samples for a specific clustering. With wider knowledge coming from multiple users, the selected samples can be more relevant to the target cluster. The challenges of this study are two-folds. Firstly, as user selected samples are usually sparse and the graph can be large, it is non-trivial to effectively combine the different annotations given by the multiple users. Secondly, it is also difficult to design a scalable approach to cluster large graphs with millions of nodes. We propose the approach *CGMA* (*C*lustering *G*raphs with *M*ultiple *A*nnotations) to address these challenges. *CGMA* is able to combine the crowd's consensus opinions in an unbiased way, and conducts an effective clustering with low time complexity. We show the effectiveness and efficiency of the proposed approach on real-world graphs, by comparing with existing attributed graph clustering approaches.

Keywords: user-guided; large attributed graph; clustering; sparse

1 Introduction

With the coming of big data era, the clustering of large attributed graphs has drawn a lot of research attention. One of the major challenges in this field is the attribute selection, which has not been fully explored. Recent research addressed this problem either by using the properties of datasets (e.g. the data density, the topology) [1, 2] or by applying user preference to guide clustering

[*] This work is supported by NSFC grants: 71331008, 61105124 and 61303017.

[**] Corresponding author (Email: szwang@buaa.edu.cn)

[7]. As user-guided clustering is more interpretable and flexible, it attracts more research attentions recently [4–7]. Different from conventional unsupervised clustering methods, user-guided clustering is semi-supervised which allows a user to select a small amount of samples for a particular cluster based on his/her preference. However, existing user-guided clustering assumes there is only one user to annotate the preferred samples. The clustering results largely rely on the labeled samples given by the user. Thus the clustering results may largely rely on the user selected samples based on his/her knowledge on the graph. A potential issue is that the clustering can be biased to the user's preference or knowledge.

In this paper, we propose a framework for user-guided large attributed graph clustering which allows multiple users to annotate their preferred samples independently. An individual may only have partial knowledge about the target clusters, and multiple annotators provide us an effective way to reveal the common knowledge toward a specific issue. Here we borrow the idea of crowdsourcing [18, 24] for user-guided clustering with multiple annotations. The general idea can be illustrated by Fig. 1. Suppose three annotators answer such a common question, e.g., "who are the data mining researchers?" As depicted in Fig. 1, each of the annotators gives his/her own annotation based on their own knowledge independently. However, since the data is too big, the annotations are sparse and may hardly overlap [23]. If we only use one of the annotations, we may get the clusters of "IBM Ph.D students," "computer science professors," or "PAKDD authors." However, if we combine the annotations together, we may find out the clusters of "data mining researchers" through the shared conferences of "data mining."

There are two major challenges for introducing multiple annotators to annotate the samples for cluster analysis. (1) It is challenging to combine the annotations of different users due to the fact that the annotations are not only sparse but also overlap very little. Thus it is hard to apply conventional techniques like majority voting to combine them in a straightforward manner. (2) It is also challenging to address the scalability issue of the proposed approach. Under the background of big data, the graph scale can be extremely large and the attribute dimensions can be very high, developing a scalable algorithm is becoming critically important.

To address the above mentioned challenges, we propose an approach for **C**lustering **G**raphs with **M**ultiple **A**nnotations ($CGMA$). A basic assumption here is that each annotator may label the samples of the preferred clusters based on only a few of the sample attributes instead of all of them [7]. With such an assumption, we map each annotation to the attribute space to obtain the weight vector denoting the relevance of attributes. In this way, the problem of combining sparse annotations is transformed into combining the weight vectors corresponding to the annotators in the common attribute space. Once the combined weight vector is obtained, we use it re-weigh the entire network to obtain a pure seed set which we used it for further clustering. The target cluster will be obtained by expanding these seed sets using a local partitioning method. The contributions of this paper can be addressed as follows,

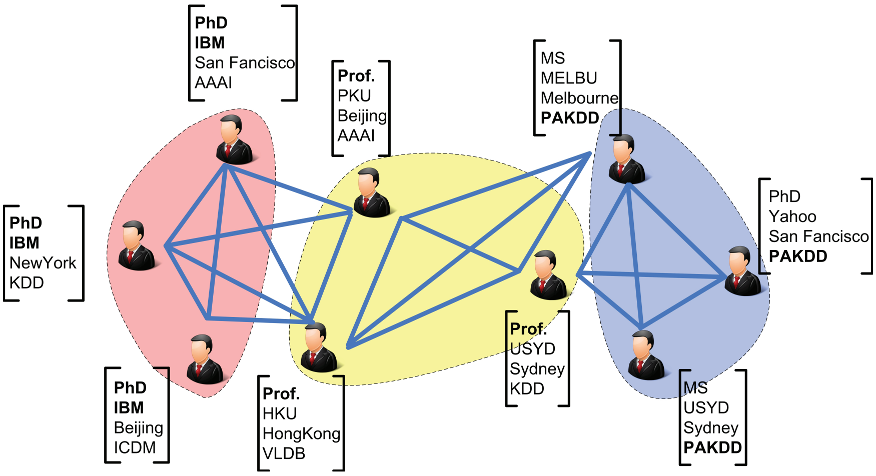


Fig. 1. A toy example of the studied problem. Different annotations with little overlapping are given. Each annotation contains several objects sharing a few focused attributes within it. But the sharing attributes among different annotations may not be the same. The aim of this paper is to identify the target cluster that complies to the multiple annotations as much as possible.

• We introduce a novel problem of user-guided clustering in large attributed networks with multiple annotations. Different from previous user-preference guided clustering, which is often biased, using multiple annotations can alleviate the bias. To the best of our knowledge, this is the first paper applying multiple annotations for graph clustering.

• We propose a two-step clustering approach $CGMA$ to address the proposed problem. $CGMA$ combines multiple annotations in an unbiased way, and it also amplifies the sparse annotations by re-sampling and expansion process. The proposed approach has near-linear time complexity.

• We conduct a series of experiments on various large networks to examine $CGMA$. The experimental results show the effectiveness and efficiency of our method.

The rest of this paper is organized as follows. Section 2 will introduce the related work of this research. Section 3 gives the details of the $CGMA$ algorithm. Next, we will show the experimental results of $CGMA$ on real networks compared with some competitive baselines in section 4. Finally, section 5 concludes the paper.

2 Related Work

Clustering of homogeneous graphs can be sorted into two groups, the plain graph clustering and the attributed graph clustering. Traditional methods mostly target at plain graphs, and they have been well studied in literatures, for example, the partitioning methods $METIS$ [27] and spectral clustering [14] aim to find a k-way partitioning of the graph. Community detection methods [16] cluster the graph into variable size communities, which is significantly different from partitioning-based methods. Autopart, cross-associations [4], and information theoretic co-clustering [13] are parameter-free examples to graph clustering methods. Several methods [19, 20] also allow clusters to overlap as observed in real-world social and communication networks. However, all of these methods are limited to plain graphs (without attributes). Compared to the wide range of works on plain graph mining, there has been much less works on attributed graphs. The representative methods [2, 11] aim to partition the given graph into structurally dense and attribute wise homogeneous clusters. These methods, however, enforce attribute homogeneity in all attributes. Recently some methods loosen this constraint by unsupervised feature selection [1] to extract cohesive subgraphs with homogeneity in a subset of attributes. However, all of these methods either do not perform a selection of attributes, or select the attributes in a biased way.

Semi-supervised clustering applies a small amount of labeled data to aid and bias the clustering of unlabeled data [8]. There are various kinds of methods for semi-supervised clustering considering user-given pairwise constraints like 'must-link' and 'cannot-link' [10]. It is also known as constraint-based clustering where the constraints are often strict to follow [12]. However, most of these methods are based on vector data, thus they are not applicable to graphs with attributes. Methods on seeded community mining [19, 22] find communities around (user-given) seed nodes. However, those methods find structural communities on plain graphs and neither are applicable to attributed graphs, nor enable user guidance on attributes. Our proposed method has two advantages compared with above mentioned methods. First, we apply user-given example sets to automatically infer the possible combination of representative attributes. Second, the constraints of traditional semi-supervised clusterings are hard, while the constraints given by different users are soft, causing the combination problem to be addressed in this study.

3 Method CGMA

In this section, we will present the framework of $CGMA$ to address the problem of using multiple annotations to guide attributed graph clustering. First of all, we give the formulation of our problem. *Given a large attributed graph $G(V, E, F)$ with $|V| = n$ nodes and $|E| = m$ edges, where each node is associated with $|F| = d$ attributes, we target to extract cluster C from G with the guidance of K users. Each user independently labels the samples based on his/her own knowledge. The samples annotated by the k-th user are denoted as U^k.* For each set U^k, we assume that nodes inside it are similar to each other, and they are dissimilar to the nodes outside the set.

3.1 Framework

The proposed approach $CGMA$ combines the annotations first in an unbiased way to obtain the guidance information. Then, a local clustering method is applied to cluster the graph with the guidance of combined annotations. Thus, $CGMA$ addresses the problem in two phases, the annotations combination and cluster extraction.

Annotations Combination. Since the annotations are sparse labels with little overlaps, straightforward methods like majority voting may not effectively capture the relations among the annotations. In this paper, we combine the annotations through each one's inferred weights in relevance to the feature space. Here are two major steps. The first step is mapping the annotations to the attribute space to facilitate measuring the similarity of the annotations. For different annotators, the attributes they think are essential to a particular cluster may be different due to their biased knowledge. Our first goal is to infer the attribute weights of $U^k (k \in \{1, \cdots, K\})$ that make the example nodes as similar to each other as possible. The similarity between two nodes can be measured by the (inverse) Mahalanobis distance: the distance between two nodes with feature vectors f_i and f_j is $(f_i - f_j)^T A^k (f_i - f_j)$. To ensure it as a metric, we set the weight matrix A^k as a positive definite matrix [3], and it denotes the attribute weight that is relevant to annotator $k's$ preference.

The process of learning A^k from annotation U^k is known as the distance metric learning problem [3]. The essence is to minimize the distance among the nodes in U^k. The optimal A^k can be obtained by solving the following convex optimization problem.

$$\min_{A^k} \sum_{(i,j)\in P_C^k} (f_i - f_j)^T A^k (f_i - f_j) - \gamma log(\sum_{(i,j)\in P_D^k} \sqrt{(f_i - f_j)^T A^k (f_i - f_j)}) \quad (1)$$

Here, P_C^k and P_D^k denote the similar and dissimilar set of the k-th annotation, respectively. Following [7], we consider the annotated node pairs as similar set, and the un-annotated node pairs as dissimilar set. The un-annotated pairs are randomly selected from the edges of un-annotated part. To emphasize the difference between similar and dissimilar set, we set $|P_D^k| = d|P_C^k|$ by over-sampling of P_D^k [17]. According to [3], the above objective function is convex and enables efficient, local-minima-free algorithms to solve it, especially for a diagonal solution.

The second step in this phase is to combine the attribute weights A^k of each sample set. Since A^k is a diagonal matrix, we assign attribute vector $\beta^k = diag(A^k)(k \in 1, \cdots, K)$ and combine the vectors β^k according to its importance [15]. Each weight vector β^k can be viewed as a point in a d-dimensional Euclidean space, where the distances d_{ij} $(i, j \in 1, \cdots, K)$ between β^i and β^j be measured by Euclidean distance. For each point β^k, we first compute its local density ρ_k as its importance. Here, the local density of β^k refers to the number of points within a distance d_c to it (Eq. 2).

Algorithm 1 Combination: The Combining of Annotations

```
Input: example annotations $U^1, \cdots, U^K$
Output: combined attribute weights vector $\beta$
 1: //Computing attribute weights vectors
 2: for all $U^k$ do
 3:     Similar pairs $P_S^k = \emptyset$, Dissimilar pairs $P_D^k = \emptyset$
 4:     for all $u \in U^k$, $v \in U^k$ do
 5:         $P_S^k = P_S^k \cup (u, v)$
 6:     end for
 7:     repeat
 8:         Random sample $u$ from set $V \backslash U^k$
 9:         Random sample $v$ from set $V \backslash U^k$
10:         $P_D^k = P_D^k \cup (u, v)$
11:     until $d|P_S|$ dissimilar pairs are generated, $d = |F|$
12:     Oversample from $P_S$ such that $|P_S| = |P_D|$
13:     Solve objective function in Eq. (1) for diagonal $A^k$
14:     $\beta^k = diag(A^k)$
15: end for
16: //Combining the attribute weights vectors
17: for all $\beta^k$ do
18:     Compute $\rho_k$ by Eq. (2)
19: end for
20: Calculate $\beta = norm(\sum_k \rho_k \beta^k)$
21: return combined attribute weights vector $\beta$
```

$$\rho_k = \sum_{l=1, l \neq k}^{K} \chi(d_{kl} - d_c) \tag{2}$$

where $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ otherwise, and d_c is a distance threshold. The algorithm is only sensitive to the relative magnitude of ρ_k in different points. Thus the results of analysis are robust with respect to the choice of d_c [15]. Finally, we get the combination β of weight vectors $\beta^1, \cdots, \beta^K$, according to each vector's importance ρ_k.

We give the details of combining the annotation results in $Algorithm$ 1. The step of inferring the attribute weights of an annotation is illustrated in $A1$ Line $2-15$, and the combination of the attribute vectors is shown in $A1$ Line $17-22$. In our setting, all pairs of example nodes in U^k constitute P_S^k ($A1$ Line 3). We create P_D^k by randomly drawing pairs of nodes that do not belong to user k's example set ($A1$ Lines $7-11$). Note that if $\rho_k = 0$ ($A1$ Line 18), β^k will have no contribution to the combined vector β. That denotes user k's opinion will be ignored ($A1$ Line 20). In the last step, we get a combined β, and then we normalize it.

Cluster Extraction. We use the information of combined annotations to extract the target cluster from the graph. Since a global clustering method would be time-consuming and can not scale well to large graph, we apply a seed-set-expansion algorithm to identify clusters locally with lower computational complexity.

There are two major problems in this step. First, how to extract the seed set samples of the cluster based on the different annotations from multiple annotators and the combined attribute weight vector? Second, how to develop a local partitioning method so that the expansion of seed sets will be scalable to large graphs? Therefore, we explain our algorithm focusing on two parts, the identification of seed set S of the cluster and its expansion rules.

In the process of identifying a pure seed set, we first apply the combined vector β to re-weigh the entire graph, then select the edges with high weight (similarity) to shape the seed set. We call this process as "re-sampling", which aims to enrich the samples space for the expansion process. Specifically, we firstly measure the weights of all the edges. Then, we assign the edges with high weights over threshold w_r as seeds. Simply, we assign a linear interpolation as w_r over the weights of samples, $w_r = \lambda w_{max} + (1 - \lambda)w_{min}$, where λ is a parameter falls in $[0, 1]$. w_{max} and w_{min} represent the maximum and minimum value of the example edges weighted by β, respectively. Algorithm 2 details the process of finding the pure seed set by re-sampling.

Next, we expand the seed set S to the target clusters C through a series of strict rules. Following the expansion process in [19], the expansion process carefully adds new nodes to each component of S. In this paper, we apply conductance [19] to measure the quality of a cluster as it accounts for both the cut size and the total volume/density retained within the cluster. The weighted conductance $\phi^{(w)}(S, G)$ of a set of nodes in graph $G(V, E, F)$ is defined as follows,

$$\phi^{(w)}(S, G) = \frac{W_{cut}(S)}{W_{vol}(S)} = \frac{\sum_{(i,j)\in E, i\in S, j\in V\setminus S} w_{ij}}{\sum_{(i,j)\in S} \sum_{(i,j)\in E} w_{ij}} \tag{3}$$

Here, $W_{cut}(S)$ and $W_{vol}(S)$ are the total weight of cut edges and within edges of S, respectively. The lower the conductance of a cluster is, the better the quality of the cluster is with few cross-cut edges and large within-density.

In each step, the expansion process selects all the nodes in the margin of a component, and adds the ones that will decrease the conductance of the cluster. The process will simultaneously kick out the (nodes) edges within a cluster that will decrease the conductance of the cluster. The process continues until there is no node changing that would decrease the quality of a component. Due the page limitation, we do not illustrate the algorithm, please refer [15] for more details.

3.2 Complexity analysis

(1) The combination of annotations. Since every annotator provides the same amount of examples, we take $U^k (k \in 1, \cdots K)$ as an example to analyze the time complexity of this step. First, we create similar and dissimilar node pairs which we use to infer the attribute weights. Since the optimization objective in Eq. 1 is convex and we aim to find a diagonal solution, local-optima-free gradient descent

Algorithm 2 find seed set by re-sampling

```
Input: attributed graph $G(V, E, F)$, combined weight vector $\beta$, annotations $U^1, \cdots, U^K$
Output: seed set $S$ for expansion
re-weigh edges by $\beta$ getting edge re-weight $w(u, v)$
for all $(u, v) \in E$ do
    $w(u, v) = 1/(\sqrt{(f_u - f_v)^T diag(\beta)(f_u - f_v)} + \epsilon)$
end for
seed node set $V' = \emptyset$
$w_{max}(w_{min}) = max(min)\{w(u, v)|u, v \in \{U^1 \cup \cdots \cup U^K\}\}$
if $w(u, v) > w_r = \lambda w_{max} + (1 - \lambda)w_{min}$ then
    seed nodes $V' = V' \cup \{u, v\}$
end if
build seed set graphs $g(V', E', F)$ where
  $\forall u, v \in V', (u, v) \in E, w(u, v) \geq w'\text{iff}(u, v) \in E'$
seed set $S \leftarrow G(V', E', F)$
return seed set $S$
```

techniques will take $O(d/\epsilon^2)$ for an $\epsilon-$approximate answer [26]. The clustering process of combination is not time consuming because the number of annotators is significantly small to the data scale, $K << n$. According to [15], we calculate that the complexity is $O(K^2)$. Therefore, the total computational complexity of the first part is $O(Kd/\epsilon^2) + O(K^2)$.

(2) The finding and expansion of seed set. Since β is supposed to be sparse with only a few non-zero entries for focused attributes, the multiplicative factor becomes effectively constant yielding a complexity of $O(m)$. In the process of expansion, we enlist all the non-member neighbors as the candidate set and evaluate their weighted $\varDelta$ conductance. As discussed above, the complexity is $\sum_{n \in S} d(n)$. Since $S \subseteq V$, it is equivalent $O(m)$. As we add one node at each iteration, the total complexity becomes $O(|S|m)$ where $|S|$ is the node scale of the seed set, and $|S| << n$.

To sum up, the complexity of the two phases comes to $O(Kd/\epsilon^2 + K^2 + |S|m)$. It is critically low comparing to the large scale of graphs.

4 Experiments

In order to evaluate the clustering quality and scalability of $CGMA$, we compare it with two representative graph clustering techniques $METIS$ [27] and $FocusCO$ [7] on real-world datasets. $METIS$ is a classical graph partitioning algorithm which expects the number of clusters as input. $FocusCO$ is a local clustering approach proposed recently using the guidance of a single user.

To introduce $CGMA$ clearly, we conduct our experiments on the "four-area" dataset, a co-authorship network of computer science researchers. The attributes of the authors are the conferences in which they have published papers in the areas of database(DB), data mining(DM), machine learning(ML), and information retrieval(IR). We use multiple annotations from 50 persons, each person gives 20 sample nodes in responding to the same question in one experiment. For the convenience of study, our problem is identifying the researchers belonging to the four areas, respectively. The ground truth clusters of an area consists of all the researchers whoever published at least one paper in the area.

Since the re-sampling process affects the final clusters significantly, we conduct a parameter study of λ, and we show the F1-score of the clustering results with different settings of λ. As shown in Fig. 2, the F1-score of the final clustering results is not linearly related to λ. One can see that without the re-sampling step the F1-score of the final results in each of experiments is critically low, about or less than half of the value when $\lambda = 0.4$. The F1-score of the final clustering results presents that the re-sampling properly will improve clustering performance significantly.

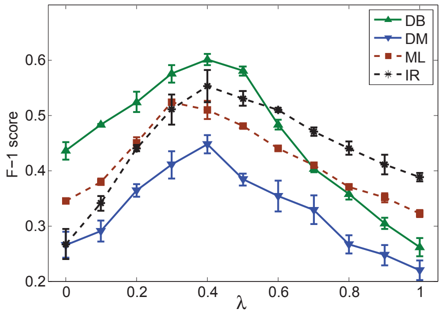


Fig. 2. The λ effects on the clustering results, 50 annotations in each of the experiment.

Accuracy. We compare the cluster results with *METIS* and *FocusCO*. Here we set the clusters number of *METIS* as four, which performs the best on this dataset. As shown in Fig. 3, one can see that F1-score of our method is significantly higher than that of the two baselines, which shows the superior performance of the proposed method. The experimental results show that our method significantly outperforms the other two methods.

Stability. We also examine the stability of the proposed *CGMA*. Although we have different annotations as inputs, they are annotated under the same question. Therefore, the annotations are all theoretically related a common clustering. We use the normalized mutual information (NMI) to evaluate the stability of the proposed clustering approach. Here, we use average NMI between each pair of clustering results to indicate the stability of a method. Higher NMI implies a more stable clustering result. As shown in Fig. 5, the proposed approach *CGMA* gets more stable results than other two methods. With the increasing of λ, the stability of *CGMA* improves significantly.

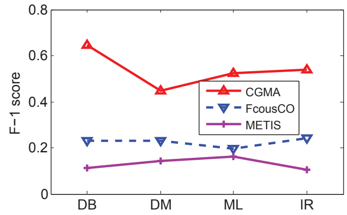


Fig. 3. The accuracy of *CGMA*, *FocusCO* and *METIS*.

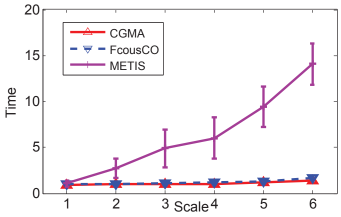
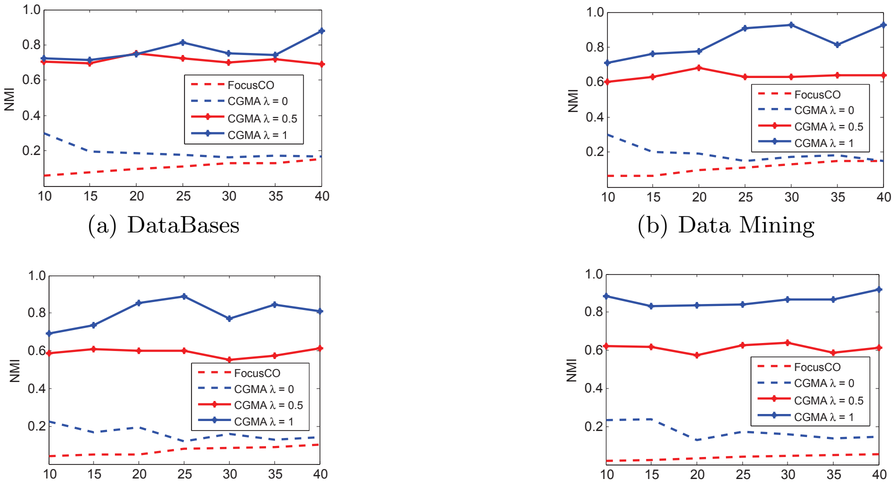


Fig. 4. The scalability of *CGMA*, *FocusCO* and *METIS*.



(a) DataBases

(b) Data Mining

(c) Machine Learning

(d) Information Retrieval

Fig. 5. The comparison of average NMI value in the 4 clusterings. The horizontal axis in all sub-figures represents the number of annotations we randomly selected.

Scalability. We select five subsets of "four-area" with the size from 100 to 27200. Each dataset scale is three times larger than its previous one. Then we conduct extensive experiments on these datasets. Note that the annotation volumes change with the scale of experimental graphs. Larger scale of the graph needs more annotations. For each dataset, we run the experiments for ten times and average the results. The experimental results are shown in Fig. **??**. Note that for *CGMA*, the extraction of cluster can be performed in parallel, thus the computing time can be significantly reduced. As the figure shown, the running time of *METIS* increases with the increasing of graph scale. However, the running time of *CGMA* and *FocusCO* is stable. In such case, the running time of *CGMA* is also comparable to *METIS* and *FocusCO*.

To further examine the scalability of *CGMA*, we conduct more experiments on two different types of real world attributed networks. The first is crawled from

PolBlogs, a citation network among a collection of online blogs that discuss political issues. The attributes of *PolBlogs* are the keywords in the blogs. The second dataset is crawled from *Twitter*, and it is a following network with a collection of discussed topics. The attributes are the keywords in their posts. Statistics of the datasets are given in also given in Table 1. The average running time (total) and their standard deviations are in Table 1. The running time of *CGMA* is the total running time including the annotation combination and clustering extractions steps. As it shows, the running time demonstrates the efficiency of our approach. It only takes less than 2 seconds to cluster the Twitter dataset with more than 14 thousand nodes, which shows CGMA can be scalable to very large graphs. The experimental results prove that *CGMA* is a scalable approach that can deal with various datasets.

Table 1. Comparisons on the Scalability of *CGMA*

Dataset	$\|V\|$	$\|E\|$	$\|F\|$	$\|C\|$	**Running time (sec)**
PolBlog	362	1288	44839	10	0.4772 ± 0.0591 **(CGMA)** 0.8772 ± 0.0839 **(METIS)** 3.0561 ± 0.0471 **(FocusCO)**
Twitter	14078	44619	17839	10	1.2135 ± 0.0322 **(CGMA)** 1.9425 ± 0.0381 **(METIS)** 6.8772 ± 0.0491 **(FocusCO)**

5 Conclusions

In this work, we introduced a novel problem of finding clusters with multi-example sets in large attributed graphs. The challenge here is how to combine them in an unbiased way in order to conduct a clustering. To address these challenges, we proposed *CGMA* in this paper which has two major phases: 1) combining the various example sets, 2) re-sampling the seed sets and expanding them to find a batch of densely connected clusters. Extensive experiments are conducted to examine the *CGMA*, and the experimental results showed that the proposed approach outperforms baseline methods.

References

1. Tang, Jiliang and Liu, Huan. Unsupervised feature selection for linked social media data. SIGKDD. (2012)
2. Akoglu, Leman and Tong, Hanghang and Meeder, Brendan and Faloutsos, Christos. PICS: Parameter-free Identification of Cohesive Subgroups in Large Attributed Graphs. SDM. (2012)
3. Xing E P, Jordan M I, Russell S, et al. Distance metric learning with application to clustering with side-information Advances in neural information processing systems. (2002)

4. Yin, Xiaoxin and Han, Jiawei and Yu, Philip S. Cross-relational clustering with user's guidance. SIGKDD. (2005)
5. Yin, Xiaoxin, Jiawei Han, and S. Yu Philip. CrossClus: user-guided multi-relational clustering. SIGKDD. (2007)
6. Sun Y, Norick B, Han J, et al. Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. SIGKDD. (2012)
7. Perozzi B, Akoglu L, Iglesias Snchez P, et al. Focused clustering and outlier detection in large attributed graphs. SIGKDD. (2014)
8. Basu S, Banerjee A, Mooney R. Semi-supervised clustering by seeding. ICML. (2002).
9. Sánchez, P I, Muller E, Laforet F, et al. Statistical Selection of Congruent Subspaces for Mining Attributed Graphs. ICDM (2013)
10. Chapelle, Olivier and Schölkopf, Bernhard and Zien, Alexander and others, Semi-supervised learning. MIT press Cambridge (2006)
11. Zhou Y, Cheng H, Yu J X. Zhou Y, Cheng H, Yu J X. Graph clustering based on structural/attribute similarities. J. VLDB, 2(1): 718-729 (2009)
12. Han J, Kamber M, Pei J. Data mining: concepts and techniques: concepts and techniques. Elsevier. (2011)
13. Dhillon, Inderjit S and Mallela, Subramanyam and Modha, Dharmendra S. Information-theoretic co-clustering. SIGKDD. (2003)
14. Ng, Andrew Y and Jordan, Michael I et. al: On spectral clustering: Analysis and an algorithm. NIPS. (2002)
15. Rodriguez, Alex and Laio, Alessandro. Clustering by fast search and find of density peaks. Science. 344. (2014)
16. Flake, Gary William and Lawrence, Steve and Giles, C Lee. Efficient identification of web communities. SIGKDD. (2000)
17. Senzhang Wang, Zhoujun Li, Wen-Han Chao, Qinghua Cao: Applying adaptive over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning. IJCNN. (2012)
18. Zhou, Dengyong and Liu, Qiang and Platt, John C and Meek, Christopher. Aggregating Ordinal Labels from Crowds by Minimax Conditional Entropy. ICML. (2014)
19. Andersen, Reid and Chung, Fan and Lang, Kevin. Local graph partitioning using pagerank vectors. In IEEE SFCS. (2006)
20. Yang, Jaewon and Leskovec, Jure. Overlapping community detection at scale: a nonnegative matrix factorization approach. WSDM. (2013)
21. Tong, Hanghang and Lin, Ching-Yung. Non-Negative Residual Matrix Factorization with Application to Graph Anomaly Detection. SDM. (2011)
22. Gleich, David F and Seshadhri, C. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. SIGKDD. (2012)
23. Senzhang Wang, Sihong Xie, Xiaoming Zhang, Zhoujun Li, Philip S. Yu, Xinyu Shu: Future Influence Ranking of Scientific Literature. SDM. (2014)
24. Ruvolo, Paul and Whitehill, Jacob and Movellan, Javier R. Exploiting Commonality and Interaction Effects in Crowdsourcing Tasks Using Latent Factor Models. NIPS. (2013)
25. Zhou, Dengyong and Basu, Sumit and Mao, Yi and Platt, John C. Learning from the wisdom of crowds by minimax entropy. NIPS. (2012)
26. Boyd, Stephen and Vandenberghe, Lieven.Convex optimization. Cambridge university press. (2004)
27. Karypis, George and Kumar, Vipin. Multilevel algorithms for multi-constraint graph partitioning. In ACM/IEEE conference on Supercomputing (1998)