Detecting Commmunities via Simultaneous Clustering of Graphs and Folksonomies

Akshay Java, Anupam Joshi, and Tim Finin

University of Maryland, Baltimore County
1000 Hilltop Circle
Baltimore MD 21250, USA
aks1,joshi,finin@cs.umbc.edu

Abstract. We present a simple technique for detecting communities by utilizing both the link structure and folksonomy (or tag) information. A simple way to describe our approach is by defining a community as a set of nodes in a graph that link more frequently within this set than outside it *and* they share similar tags. Our technique is based on the Normalized Cut (NCut) algorithm and can be easily and efficiently implemented. We validate our method by using a real network of blogs and tag information obtained from a social bookmarking site. We also verify our results on a citation network for which we have access to ground truth cluster information. Our method, Simultaneous Cut (SimCut), has the advantage that it can group related tags and cluster the nodes simultaneously.

1 Introduction

Participants in social media systems like blogs and social networking applications tend to cluster around common topics of interest. An important task in analyzing such networked information sources is to identify the significant communities that are formed. Communities are one of the essential elements of social media and add to their richness and utility. A community in the real world is often reflected in the graph representation as a group of nodes that have more links within the set than outside it.

Many social media systems and Web 2.0 applications support free form tagging, also known as a *folksonomy*. A typical example of such a system is del.icio.us[1], where items are bookmarked with descriptive terms associated with the resource. Analysis of tagging systems has shown the utility of folksonomies in providing an intuitive way to organize, share and find information [1]. One approach to group related resources together is by utilizing the tag information. Two URLs belong to the same cluster if they are tagged or categorized under similar sets of tags. This approach was used by Java et al. [2] for clustering related blog feeds and to identify the popular feeds for a given topic.

Clustering based on tags or folksonomy exclusively misses the information available from the link structure of the Web graph. On the other hand, partitioning the graph based on links exclusively ignores tags and other user-generated

[1] http://del.icio.us

meta data available in most social media systems. In this work, we address the problem of combining both the graph and folksonomy data to obtain significant communities in a social network or a blog graph. The intuition behind this technique is that a community is

> *a set of nodes in a graph that link more frequently within this set than outside it and they share similar tags.*

Figure 1 describes the above definition pictorially. The nodes in circles represent entities (URLs, blogs or research papers). Such entities often link to each other via hyperlinks or citations. The square nodes represent the tag information or any user-generated content associated with a given resource. Several entities can share the same descriptive tags. Our extended definition of a community requires us to find a partition of the above graph such that it minimizes the number of edges cut in both the entity-entity and the entity-tag edge set. The Normalized Cut (NCut) algorithm [3] is an efficient technique to find these partitions. Our method, which is based on the NCut algorithm, can be efficiently implemented and provides a good clustering of the graph into its constituent communities.

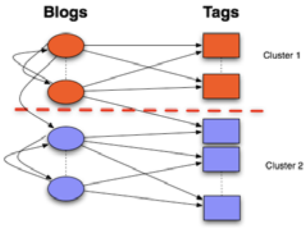

Fig. 1. A community can be defined as a set of nodes in a graph that link more frequently within this set than outside it and the set shares similar tags.

The detailed description of this technique is provided in the following sections. First we begin with the basics of spectral clustering and community detection. Section 2 provides an outline of NCut, co-clustering, constrained dimensionality reduction and related methods. In section 3 we present the basic algorithm for simultaneously clustering graph and tag information. Section 4 provides experimental results and compares the results of NCut and Simultaneous Clustering technique described here. Finally, we summarize and conclude with a discussion of advantages and limitations of the proposed algorithm in Section 5.

2 Related Work

Spectral clustering is a method that is based on the analysis of eigenvectors of a graph or more generally, any similarity matrix. It has been used to efficiently cluster data and partition graphs into communities. Shi and Malik [3] developed a normalized cut criteria to find balanced partitions of an image. The proposed method optimizes the inter-cluster similarity as well as similarity within clusters. Though this method was originally applied for image segmentation, it has found several applications in graph mining and community detection [4]. A comprehensive survey of spectral clustering is provided by von Luxburg [5].

Most spectral clustering techniques use either the un-normalized or normalized form of graph Laplacian. The graph Laplacian is a representation of the similarity matrix that has a number of important properties [6, 7]. Consider a graph $G = (V, E)$ where V is the set of vertices and E represents the set of edges. If $W \in \Re^{n \times n}$ represents the similarity or adjacency matrix such that $W_{ij} = 1$ if an edge, $e_{ij} \in E$ exists. The general format of a graph Laplacian is given by:

$$L = D - W \tag{1}$$

where $D \in \Re^{n \times n}$ is a diagonal matrix representing the degrees of nodes in the graph.

An important property of the graph Laplacian is that the smallest eigenvalue of L is 0 and the smallest non-zero eigenvalue corresponds to the *algebraic connectivity* of the graph [8]. The vector corresponding to the second smallest non-zero eigenvalue is also known as *Fiedler vector* [8]. The algebraic connectivity of the graph is an indicator of how well connected the graph is. The original graph can be easily partitioned using only the sign of the values in the *Fiedler vector*.

Recently, spectral methods have been applied to community detection [9] and shown to have a relation to optimizing the modularity score [10]. The modularity function defined by Newman et al. [11] is an intuitive measure of the quality of any clustering algorithm. The modularity function, Q, measures the fraction of all the edges, e_{ii} that connect within the community to the fraction of edges, a_i that are across communities. The measure Q is defined as

$$Q = \sum_i (e_{ii} - a_i^2) \tag{2}$$

Determining the "best" community structure by finding the optimal modularity value has been shown to be NP-Hard [12] and is thus not a viable approach for even networks of relatively modest sizes. It has been shown [4] that the problem of maximizing the modularity score, Q can be expressed in terms of the graph Laplacian as follows

$$max\ Tr(X^T(W - D)X) \tag{3}$$

where $X \in \Re^{n \times k}$ is the cluster assignment matrix that indicates the membership of a node to a particular partition of the graph. The solution to the optimization problem in Equation 3 can be found by partitioning the nodes using the *Fiedler vector*, or the second smallest eigenvector of the graph Laplacian.

However, this technique for finding communities relies entirely on the link structure. In social media, there are a number of additional sources of meta-data information and annotation that can be obtained. Folksonomies or tags are one form of user-generated meta-data. There can possibly exist many more features that can be additionally used to identify communities. A few examples of these are sentiments and link polarity [13], related Wikipedia entries [14], links to main stream media sites, comments in blog posts, tags used by the blogger (as opposed to the tags used by readers or in social bookmarking sites). All these features provide additional cues and can be potentially useful in community detection algorithms. However, it is not always clear how to integrate these into an unsupervised learning method.

A closely related clustering technique is co-clustering [15]. Co-clustering works by mapping an $m \times n$ term document matrix, A into a bipartite graph. The adjacency matrix of the bipartite graph is represented as

$$M = \begin{pmatrix} 0 & C \\ C^T & 0 \end{pmatrix}$$

where $C_{ij} = 1$ if the word j occurs in document i. It was shown [15] that the optimal clustering can be found by partitioning the graph represented by M. However, note that in this technique the links between the document set are never used. In the following section, we will describe the relation of our methods with the co-clustering.

3 Clustering of Graph and Tags

Our approach for simultaneously clustering graphs and tags was inspired by the classification constrained dimensionality reduction method proposed by Costa and Hero [16] and the co-clustering algorithm proposed by [15]. The constrained dimensionality reduction technique tries to incorporate the class label information to represent a high dimensional data into a lower dimensional space. For example, if the goal was to classify the collection of documents, using the approach presented by Costa and Hero, the known class labels (from the training data) are incorporated into the dimensionality reduction step. The algorithm optimizes a cost function such that the class estimates for the training data is close to the final cluster center and it also satisfies the normalized cut criteria. Their approach belongs to a general class of semi-supervised learning methods.

Following the notations used by Costa and Hero [16], let $W \in \Re^{n \times n}$ represent the adjacency matrix for a set of n nodes. Let $C \in \Re^{n \times k}$ be a matrix that represents if a node is associated with one of the k tag and $\beta > 0$ be a scaling parameter that regulates which of the two information link structure or tags is given more importance. Then the partitioning of the nodes into communities such that the membership is determined based on both the link structure and tags can be found by the eigenvectors associated with the matix

$$W' = \begin{pmatrix} I & C \\ C^T & \beta W \end{pmatrix}$$

The matrix W' combines information from both the graph and the folksonomy. The first k columns correspond to the entity-tag edges in Figure 1 while the last n columns represent the entity-entity links. Finding a partition in the above graph that minimizes the number of edges that are cut, will result in clusters that have more links within the set than outside it and at the same time share similar sets of tags. This satisfies our extended definition of a community. Also note the relation to co-clustering in the above matrix. If the parameter β is set to 0, it would lead to the bipartite graph model used by Dhilon [15]. In our experiments that follow, we set $\beta = 1$ indicating an equal importance to tag information and graph structure.

A related technique is the constrained spectral clustering approach discussed in Xu et al. [17]. Their work utilizes the pairwise constraint information that describe if two nodes *must-link* or *cannot-link* [18]. In some cases this information can be available from domain knowledge or directly derived from the data.

4 Experimental Results

The following section presents the experimental results on two datasets. One is a network of academic paper citations and the associated text with these publications. This dataset contains six clusters for which ground truth label information is available. The other dataset is a blog graph network and the corresponding folksonomy extracted from a social bookmarking site.

4.1 Dataset Description

Table 1. Table summarizing the statistics for the data used in this experiment. The first dataset is a paper citation network while the other is a blog graph network. Both datasets are comparable in size.

Citeseer Data		
1	Number of Papers	3312
2	Number of Words	3703

Blog Data		
1	Number of Documents	3286
2	Number of Tags	3047
3	Number of Homepages	3111
4	Number of stemmed words	10191

For our experiments, we have used two datasets, summarized in Table 1. The first dataset is a citation network of academic publications derived from Citeseer[2] [19]. It consists of 3286 papers from six different categories: Agents, Artificial Intelligence (AI), Databases (DB), Human Computer Interaction (HCI), Information Retrieval (IR) and Machine Learning (ML). The category information was provided in the dataset along with a binary document-term matrix indicating the presence or absence of a term in a given publication. Since, this dataset

[2] http://citeseer.ist.psu.edu/

has the ground truth for classification, it makes it ideal for our experiments. Since we do not have any folksonomy information associated with the publications, we use the words as a substitute for tag information. Since only a binary term vector for each document is provided in this collection, we use an Radial Bias Function (RBF) kernel, $K_{ij} = exp(-||x_i - x_j||^2/2\sigma^{-2})$ to compute the document similarities.

The second dataset is a subset of the Weblogging Ecosystems (WWE) workshop dataset. The original dataset consists of about 10M posts from 1M weblogs over a 20 week period. From the original dataset, we extracted a subgraph corresponding to the top five thousand high PageRank nodes. Next, for each of these blogs we fetched the tags associated with its URL in del.icio.us, a social bookmarking tool. We found 3286 blogs that had some tags associated with them in this system. We chose to use the folksonomy from del.icio.us since it is currently the most popular social bookmarking tool. As opposed to self-identified tags specified by the blogger in blog search engines like Technorati[3], del.icio.us ranks the most popular tags associated with a given URL is aggregated over several users. User-generated labels or tag information provide descriptive meta-data that are helpful in determining the topic or theme of a resource and hence can be helpful in community detection. In general, we can extend our method to use any additional meta-data such as opinions, machine learned categories, etc. Although both the datasets contain directed edges, for our analysis we have convert the graph into an undirected network. This was primarily done due to ease of computation of Normalized Cuts over undirected representation of the graph. As future work, we plan to use directed, weighted normalized cut algorithm [20] that may be more applicable for Web graphs and citation networks.

Finally for the 3286 blogs, the corresponding homepages (or cached versions when available in Google) were downloaded. There were in all 3111 homepages that were retrievable. Since this dataset was originally obtained from a crawl performed in 2005, some of the homepages were non-existent. Using the set of available homepages, a hundred topics were learned using the Latent Dirichilet (LDA) model [21]. This was done primarily as a means for dimensionality reduction. Previously, LDA has been used in clustering blogs and has been shown to be an effective tool in summarizing the key topics [22].

4.2 Evaluation

First we present some empirical results using the blog dataset. The NCut algorithm partitions the graph to determine a set of communities by using only the link information. Once the communities are determined we would like to identify the tags associated with each of these communities. We use a simple approach of identifying the most frequently occurring tags in a given community. Table 2 presents the top five tags associated with 10 communities (out of 35) as identified using NCut.

[3] http://technorati.com

One advantage of using SimCut over NCut algorithm is that it can be effectively used to cluster both the blogs and the tags simultaneously. Table 3 presents the top five tags associated with 10 communities (out of 35) as identified using SimCut. Empirically, the tags associated with the communities form coherent clusters and can be easily associated with the general theme of the community.

Table 2. Top five tags associated with 10 communities found using NCut. For each community the most frequently used tags are shown in this table.

1	blog, blogs,technology, news, web
2	blog , poet, tags, browser, sustainability
3	blog, blogs, news, conspiracy, patterns
4	blog, blogs, kids, china, parenting
5	blog, crafts, craft, blogs, crafty
6	tutorials, graphics, webdesign, design, blogs
7	blog, programming, news, forum, .net
8	blog, cinema, french, literature, religion
9	blog, blogs, music, culture, art
10	blog, knitting, blogs, knitblogs, knitblog

Table 3. Top five tags associated with 10 communities found using SimCut.

1	food, cooking, recipes, foodblog, foodblogs
2	technology, business, web2.0, marketing, advertising
3	israel, jewish, judaism
4	christian, religion, philosophy, christianity, church
5	knitting, knitblogs, knitblog, knit
6	law, economics, legal, academic, libertarian
7	blogs, daily, culture, humor, funny
8	politics, media, liberal, political, progressive
9	design, web, webdesign, inspiration, css
10	tech, geek, gadgets, games, computer

Next we look at some of the statistics of communities extracted using the two methods discussed. First, we present results on the citeseer citation network. Given that the hand-labeled ground truth information is available, this dataset has the advantage that the results can be compared to the actual communities present in the graph. Tables 4 and 5 show the confusion matrix for the two clustering methods. The results indicate that while the clusters are easily identifiable using SimCut approach, the NCut approach fails to find the right clusters. In general NCut finds very large partitions in the graph that are determined by using the link information alone. The overall accuracy obtained using SimCut is around 62%.

Table 4. Confusion matrix for NCut on Citeseer data giving an overall accuracy of 36%

	NCut					
	IR	**HCI**	**DB**	**AI**	**ML**	**Agents**
1	**461**	50	81	29	182	19
2	0	**2**	9	2	0	0
3	122	154	**186**	93	199	82
4	45	1	174	**22**	2	8
5	2	0	66	1	**44**	0
6	38	301	251	104	163	**487**

Table 5. Confusion matrix for SimCut on Citeseer data giving an overall accuracy of 61.7%

	SimCut					
	AI	**IR**	**HCI**	**ML**	**Agents**	**DB**
1	**70**	44	22	78	95	166
2	4	**366**	19	24	4	30
3	23	49	**359**	35	29	16
4	77	104	15	**372**	1	25
5	62	32	66	60	**430**	18
6	13	73	27	21	24	**446**

Figure 2 shows the average cluster similarity for 6 clusters extracted using NCut and SimCut algorithms on the citeseer dataset and the distribution of the community sizes. The similarity scores were obtained by averaging the inter-document scores obtained from the RBF kernel over the term document matrix. The average cluster similarity is computed as follows:

$$\frac{\sum_{d_i \in C, d_j \in C'} K(d_i, d_j)}{p} \tag{4}$$

where $K(d_i, d_j)$ represents the score from RBF kernel and p corresponds to the number of such comparisons. Figure 3 depicts the clusters obtained by the two methods and reflects the true size of the communities found. Notice that NCut results provide a few very small communities while most communities are large and have a relatively low average document similarity score. Finally Figure 4 shows the clusters and sparsity plots obtained by reordering the original adjacency matrix using true cluster labels, NCut Communities and SimCut communities.

Figure 5 shows the average cluster similarity for 35 clusters extracted using NCut and SimCut algorithms for the blog dataset. One difficulty in evaluation for this data set is the lack of availability of any *"ground truth"* information. In order to circumvent this problem we have used the text from the blog homepages as a substitute. However, one thing to note is that this can be subject to a lot of noise that is typically contributed by various elements present on the homepage: navigation menus, advertising content, blogging platform specific templates etc [23]. Using the LDA algorithm, text from the homepages was mapped to topics vectors. The scores represented in the figure reflect the average similarities between the topic vectors for each blog.

From the distribution of community sizes we can find that the NCut algorithm results in partitions that lead to a few large communities and several very

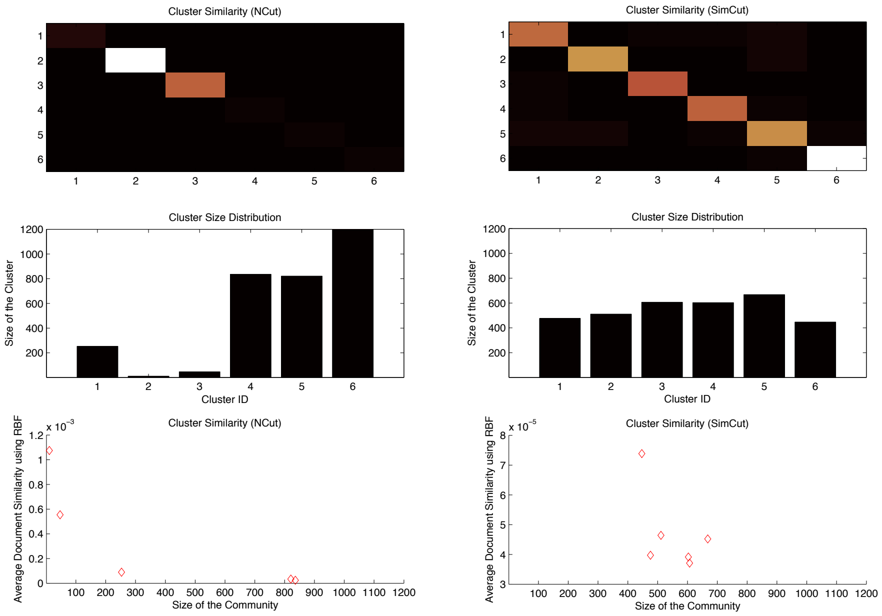

Fig. 2. The above graphs show the average cluster similarity and size distributions of the communities found using NCut and SimCut. The NCut algorithm obtains a few very large communities and a large number of very small ones. In contrast the sizes of the communities found using SimCut is more balanced.

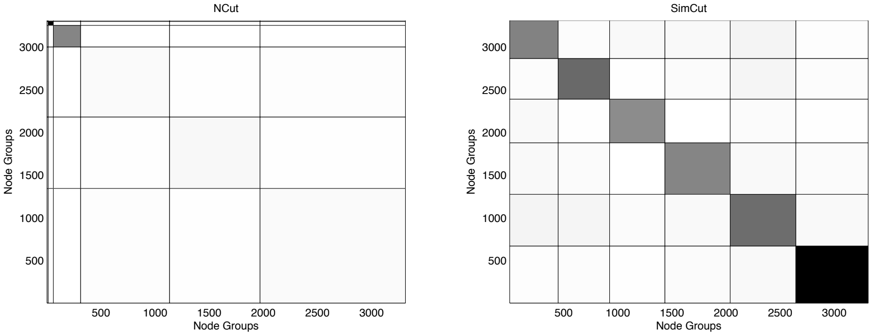

Fig. 3. 6 Clusters obtained using NCut and SimCut algorithm on the citeseer dataset. Each square in the diagonal corresponds to the communities. The shade of the squares represents the average inter/intra cluster similarity scores.

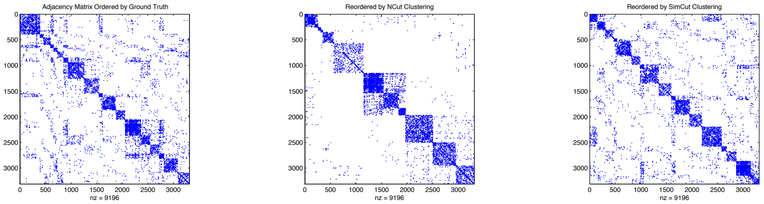

Fig. 4. The above sparsity plots show the adjacency matrix ordered by a) the true cluster labels b) Communities found by NCut approach and c) Communities found by SimCut.

small communities. This can be explained by the fact that the NCut algorithm only uses the link information and it does not have the additional meta-data (via tag information that is available to the SimCut algorithm). In comparison the SimCut algorithm finds several communities of moderate sizes. NCut yields several very small or tightly knit communities of high similarity and a few large communities of very low similarity.

One benefit of using the SimCut algorithm is that even if a few links are missed due to crawling or parsing issues, it can still find the appropriate membership information since it relies on the additional feature of tags to compare the two documents. Finally Figure 7 shows the sparsity plots for the communities found using the two techniques. A point to note here is that although the SimCut criteria does not directly optimize for the modularity score, it does not degrade it significantly either. For example in the clustering results shown in this figure, for 35 communities, the modularity scores [10] determined using NCut is 0.4939 and the corresponding value using SimCut is 0.486. We use 35 communities since it resulted in the best modularity scores over a number of empirical runs.

Given the difficulty and high cost (two to three minutes per blog) of providing human annotation and judgement for the clustering results, one way to verify the performance of the two algorithms is to use the topic vectors generated by LDA. We construct a similarity matrix, $K \in \Re^{n \times n}$, where n is the number of documents (blog homepages). We use the NCut algorithm to identify the clusters in this document similarity matrix. If there was no link or tag information available, this would be the 'best' that we can approximate the ground truth without manually annotating each blog. Table 7 compares the effect of adding tag information and varying the number of clusters. In order to compare the two clustering techniques, NCut and SimCut we use the clusters found using the topic vectors as the "ground truth". Normalized Mutual Information is used to obtain the distance measure between the two clustering results. Mutual information between two clustering results C, C' is defined as

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \log_2 \frac{p(c_i, c'_j)}{p(c_i).p(c'_j)} \quad (5)$$

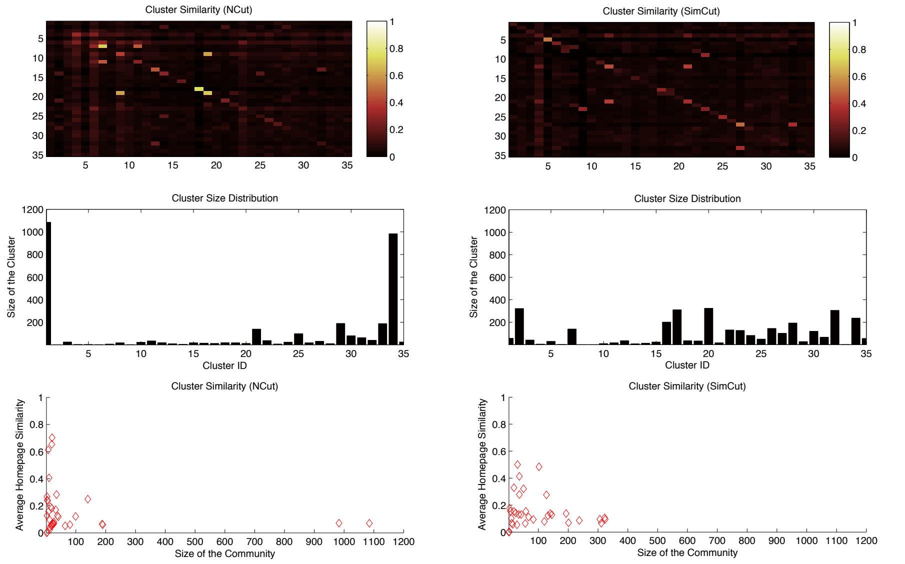

Fig. 5. The above graphs show the average cluster similarity and size distributions of the communities found using NCut and SimCut. The NCut algorithm obtains a few very large communities and a large number of very small ones. In contrast the sizes of the communities found using SimCut is more balanced.

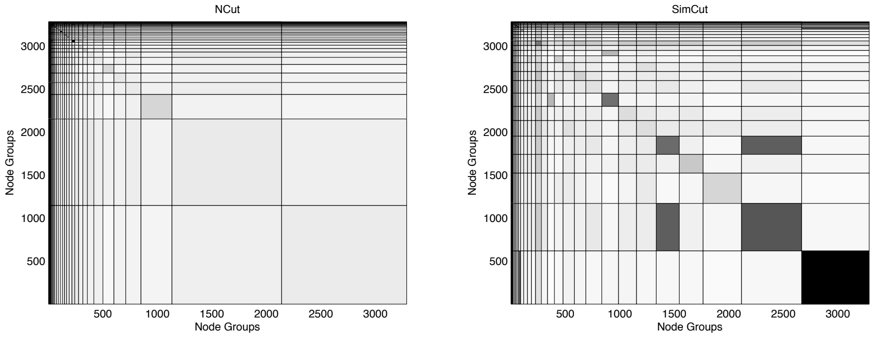

Fig. 6. 35 Clusters obtained using NCut and SimCut algorithm. Each square in the diagonal corresponds to the communities. The shade of the squares represents the average inter/intra cluster similarity scores.

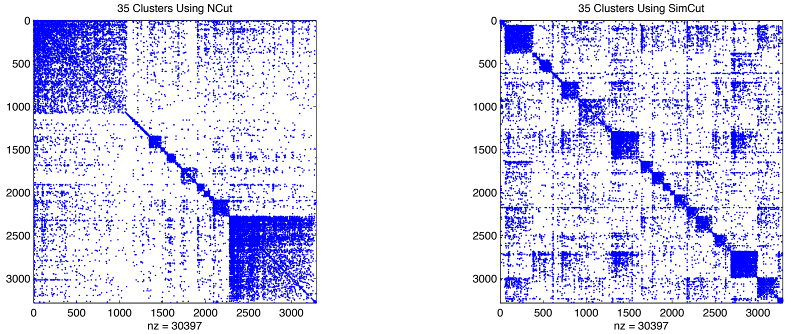

Fig. 7. The above sparsity graphs show the communities found using the two clustering approaches. The original graph of 3286 nodes was first partitioned into 35 communities using NCut. Next, by adding the top 500 tags from del.icio.us, a social book marking site, the SimCut algorithm constrains the partitions such that a communities also share similar labels or tags, thus resulting in better clustering.

where $p(c_i)$, $p(c'_j)$ are the probabilities that an arbitrary document belongs to cluster c_i and c'_j respectively. The Normalized Mutual Information score is a value between 0 and 1 that represents how close two clustering results are.

From the results shown in Figures 6 and 7, we can find that the normalized mutual information increases as more tags. For example, the score is highest at around 35 communities determined using 500 tags, in the case of the blog dataset. However, adding even more tag information does not help. The mutual information is higher than the clusters found using the link graph alone.

Table 6. Table Summarizing the Normalized Mutual Information Scores for citeseer dataset as more words are used in determining the clusters. Values reported here are averaged over 10 runs.

		SimCut (Number of Words Used)			
Clusters	NCut	50	200	500	1000
2	0.16293	0.1822	0.31934	0.35692	0.35071
3	0.16283	0.18196	0.31921	**0.35694**	0.35021
4	0.16443	0.18106	0.31949	0.35670	0.35042
5	0.16443	0.18161	0.31946	0.35665	0.35030
6	0.16126	0.17801	0.31942	**0.35682**	0.35019

5 Conclusions

Many social media sites allow users to tag resources. In this work, we have shown how incorporating folksonomy information in calculating communities can yield

Table 7. Table Summarizing the Normalized Mutual Information Scores for blog dataset as more tag information is used in determining the clusters. Values reported here are averaged over 10 runs.

		SimCut (Number of Tags Used)			
Clusters	NCut	50	200	500	1000
25	0.20691	0.22720	0.27000	0.27970	0.25878
30	0.20693	0.22615	0.27109	0.27978	0.25928
35	0.20901	0.22521	0.26998	**0.2803**	0.25791
40	0.20895	0.22584	0.27208	**0.2800**	0.25840
45	0.20861	0.22503	0.27090	0.27938	0.26004
50	0.20986	0.22767	0.27139	0.27954	0.25633

better results. The SimCut algorithm presented in this paper is based on the Normalized Cut algorithm and can be easily extended to include additional user-generated meta-data (ratings, comments, tags in blog posts, etc). A key advantage of our approach is that it clusters both the tags and graph simultaneously. One challenge in community detection algorithms is that of labeling. Providing the right label that identifies the community is beneficial in visualization and graph analysis. We are currently investigating how our technique could be used to provide intuitive labels for communities. Finally, we are focussing our study on extending SimCut to weighted, directed networks.

6 Acknowledgements

We would like to thank Nielsen Buzzmetrics for the weblogs dataset, Indrajit Bhattacharya and Lise Getoor for the Citeseer dataset. We also thank Timothee Cour, Stella Yu and Jianbo Shi for the MATLAB implementations of NCut algorithm. Finally, we thank Deepayan Chakrabarti for MATLAB scripts to help visualize some of the clustering results.

References

1. Heymann, P., Garcia-Molina, H.: Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical Report 2006-10, Stanford University (April 2006)
2. Java, A., Kolari, P., Finin, T., Joshi, A., Oates, T.: Feeds That Matter: A Study of Bloglines Subscriptions. In: Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007). (2007)
3. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8) (2000) 888–905
4. White, S., Smyth, P.: A spectral clustering approach to finding communities in graph. In: SDM. (2005)
5. von Luxburg, U.: A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics (2006) Technical Report No TR-149.

6. Grone, R., Merris, R., Sunder, V.S.: The laplacian spectrum of a graph. SIAM J. Matrix Anal. Appl. **11**(2) (1990) 218–238
7. Mohar, B.: Some applications of laplace eigenvalues of graphs (1997)
8. Chung, F.R.K.: Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92) (Cbms Regional Conference Series in Mathematics). American Mathematical Society (February 1997)
9. Chi, Y., Song, X., Zhou, D., Hino, K., Tseng, B.L.: Evolutionary spectral clustering by incorporating temporal smoothness. In: KDD. (2007) 153–162
10. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Physical Review E **74** (2006) 036104
11. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks (August 2003)
12. Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. Physical Review E **72** (2005) 027104
13. Kale, A., Karandikar, A., Kolari, P., Java, A., Joshi, A., Finin, T.: Modeling Trust and Influence in the Blogosphere Using Link Polarity. In: Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007). (March 2007)
14. Syed, Z., Finin, T., Joshi, A.: Wikipedia as an Ontology for Describing Documents. In: Proceedings of the Second International Conference on Weblogs and Social Media, AAAI Press (March 2008)
15. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2001) 269–274
16. Costa, J., Hero, A.O., I.: Classification constrained dimensionality reduction. Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on **5** (2005) v/1077–v/1080 Vol. 5
17. Xu, Q., desJardins, M., Wagstaff, K.: Active constrained clustering by examining spectral eigenvectors. In: Discovery Science. (2005) 294–307
18. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2001) 577–584
19. Bhattacharya, I., Getoor, L.: Relational clustering for multi-type entity resolution. In: The ACM SIGKDD Workshop on Multi Relational Data Mining (MRDM), Chicago, IL, USA (2005)
20. Meila, M., Pentney, W.: Clustering by weighted cuts in directed graphs. In: SDM. (2007)
21. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3** (2003) 993–1022
22. Nallapati, R., W., C.: Link-PLSA-LDA: A new unsupervised model for topics and influence in blogs. In: Proceedings of the Second International Conference on Weblogs and Social Media, AAAI Press (March 2008)
23. Java, A., Kolari, P., Finin, T., Mayfield, J., Joshi, A., Martineau, J.: BlogVox: Separating Blog Wheat from Blog Chaff. In: Proceedings of the Workshop on Analytics for Noisy Unstructured Text Data, 20th International Joint Conference on Artificial Intelligence (IJCAI-2007). (2007)