Uncovering Cross-Dimension Group Structures in Multi-Dimensional Networks

Lei Tang[*] Huan Liu[†]

Abstract

With the proliferation of Web 2.0 and social networking sites, people can interact with each other easily through various social media. For instance, popular sites like Del.icio.us, Flickr, and YouTube allow users to comment sharing content (bookmark, photos, videos), and users can tag her own favorite content. Users can also connect to friends, and subscribe to or become a fan of other users. These diverse individual activities result in a multi-dimensional network among actors, forming cross-dimension group structures with group members focusing on similar topics. It is challenging to effectively integrate the network information of multiple dimensions to find out the cross-dimension group structure. In this work, we propose a two-phase strategy to identify the hidden structures shared across dimensions in multi-dimensional networks. We extract structural features from each dimension of the network via modularity analysis, and then integrate them to find out a robust community structure among actors. Experiments on synthetic and real-world data validate the superiority of our strategy, enabling the analysis of collective behavior underneath diverse individual activities in a large scale.

1 Introduction

The prosperity of online social networking sites (e.g., Del.icio.us, Flickr, YouTube, Facebook, etc.) facilitates human beings to interact with each other more conveniently than ever, and is gaining increasing attentions from a variety of disciplines to study the modeling and prediction of human behavior, as well as scaling up social network analysis from hundreds of people in traditional social science to thousands of hundreds, and more. One basic task in social network analysis is to identify the community structure among social actors [29]. The group information can be utilized for post-analysis or tasks such as group evolution [2, 27], group profiling [26], viral marketing [24], visualization [14], finding influential members within communities [1], or detecting stable clusters across temporal changes [3].

However, most existing works of community detection are focusing on one dimension of interaction among people (i.e., only one single-type network). In reality, human beings interact with each other in assorted forms of activities leading to multiple networks among the same set of actors, or a *multi-dimensional network* with each dimension representing one type of interaction. For instance, as in Figure 1, at popular photo and video sharing sites (Flickr and YouTube), a user can connect to his friends through email invitation or the provided "add as contacts" function; users can also tag/comment on the social contents like photos and videos; a user in YouTube can upload a video to respond to a video post by another user; and users can also become a fan of another user's contribution of social contents by subscription to the user. Based on each type of activity, a network among these users could be constructed representing one dimension of human interactions.

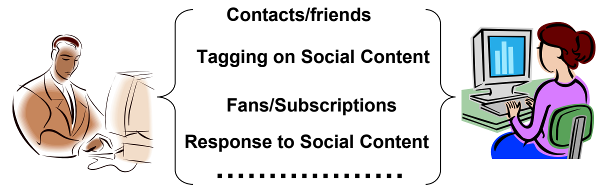


Figure 1: Multi-Dimensional Network

Another example of multi-dimensional networks is email communication network. In email communications, people can act in two different roles: senders and receivers. These two roles are not interchangeable. Spammers in the network send an overwhelming number of emails to normal users but seldom receive responses from them. The sender and receiver roles essentially represent two different interaction patterns.

With a multi-dimensional network, intuitively one can use richer information to infer more accurate latent community structures among actors. However, idiosyncratic personalities lead to varied local correlations between dimensions. Some people interact with other members within the same group in one form of activity consistently, but may be inactive in another. It thus becomes a challenge to identify groups in multi-dimensional network as we need to fuse all the information available from each dimension for joint analysis.

In this work, we propose a two-phase strategy to handle community detection in multi-dimensional networks. We first extract potential structural features from each dimension via modularity analysis. In the second phase, we concatenate all these features and conduct integration to find the most representative groups. As a real-world network typically does not have full information for the ground truth about the group membership, a cross-dimension validation procedure is proposed to compare the clustering results obtained from

[*] Dept. of CSE, Arizona State University, L.Tang@asu.edu
[†] Dept. of CSE, Arizona State University, Huan.Liu@asu.edu

different approaches. Our experiments on both synthetic and real-world network data validate the superiority of our proposed approach. Moreover, our approach can be easily paralleled and thus applicable for large-scale networks.

2 Modularity for 1-D Networks

Three patterns are frequently observed in large-scale social networks [6]: 1) small-world phenomenon; 2) scale-free property, or alternatively, the degree distribution of nodes in a network follows a power law distribution; and 3) community structure. While there have been plenty of clustering algorithms on graphs, modularity [21] is proposed specifically to measure the strength of a community structure for real-world networks by taking into account the degree distribution of nodes. It is shown to be effective in various kinds of complex networks [30, 20]. Below, we briefly review the concept of modularity.

Consider dividing an interaction network A of n vertex and m edges into k non-overlapping communities. Let $s_i \in \{1, \cdots, k\}$ denote the community membership of vertex i, and d_i represent the degree of vertex i. The expected number of edges between two nodes with a uniform random graph model is $d_i d_j / 2m$. Intuitively, modularity measures how large the interaction within communities deviates from a uniform random graph. It is defined as:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{d_i d_j}{2m} \right] \delta(s_i, s_j) \tag{2.1}$$

where $2m$ is a normalization factor, $\delta(s_i, s_j) = 1$ if $s_i = s_j$ and 0 otherwise. Note that Q can be negative if the vertexes are split into bad clusters. $Q > 0$ indicates the clustering captures some degree of community structure. In general, one aims to find a community structure such that Q is maximized.

The modularity in Eq. (2.1) can also be represented in a matrix form. Let $\mathbf{d} \in Z_+^n$ denote the degree of each node, $S \in \{0, 1\}^{n \times k}$ be a community indicator matrix defined as follows:

$$S_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ belongs to community } j \\ 0 & \text{otherwise} \end{cases}$$

and modularity matrix defined as

$$B = A - \frac{\mathbf{d}\mathbf{d}^T}{2m} \tag{2.2}$$

The modularity can be reformulated as

$$Q = \frac{1}{2m} Tr(S^T B S) \tag{2.3}$$

The discreteness of S poses the modularity maximization problem as NP-hard [5], but with a spectral relaxation to allow S to be continuous, the optimal S can be computed as the top-k eigenvectors of the modularity matrix B [20].

Contrast to the sparse interaction matrix A, the modularity matrix B is dense and cannot be computed out and held in memory if n is large (which is typically true for real-world social networks). To calculate the top eigenvectors, power method or Lanczos method can be applied as it relies only on the basic matrix-vector multiplication without holding the full matrix. Since B is the summation of a sparse matrix (A) and a rank-one matrix ($\mathbf{d}\mathbf{d}^T/2m$) , the multiplication of matrix B and a vector $\mathbf{x}$ can be calculated as:

$$B\mathbf{x} = A\mathbf{x} - \frac{\mathbf{d}^T\mathbf{x}}{2m}\mathbf{d}$$

The same trick can be applied to any structured matrix similar to B (a sparse matrix plus low rank update). This strategy is employed later in our baseline approaches as well.

The degree of freedom of k clusters is $k - 1$, so we can compute the top $k - 1$ eigenvectors to form a low-dimensional embedding of the interaction into a Euclidean space. Then a post-processing optimization step like k-means can be applied to find out a discrete community assignment [30].

3 Modularity for M-D Networks

In the previous section, we briefly review the process of modularity maximization to identify communities. Here, we extend the modularity analysis to multi-dimensional networks. A d-dimensional network is represented as

$$Net = \{A^1, A^2, \cdots, A^d\}$$

A^i represents the interactions among social actors in the i-th dimension satisfying

$$A^i \in \mathcal{R}_+^{n \times n}, \;\; A^i = (A^i)^T, \;\; i = 1, 2, \cdots, d$$

where n is the total number of actors. Here, we concentrate on symmetric networks[1]. Given a multi-dimensional network, our goal is to *find out a shared latent community structure among the social actors*. To be specific, we need to find out a community assignment such that Q_i is maximized for $i = 1, \cdots, d$.

3.1 Average Modularity Maximization (AMM) A simple strategy to handle a multi-dimensional network is to treat it as single-dimensional. One straightforward approach is to calculate the average interaction network among social actors:

$$\bar{A} = \frac{1}{d} \sum_{i=1}^{d} A_i \tag{3.4}$$

[1]Directed network can be converted into undirected networks as shown in later parts.

With $\bar{A}$, this boils down to classical community detection in a single-dimensional network, and modularity maximization method as shown in previous section can be applied directly. In reality, social actors often participate in different dimensions of network with varied intensity. Even within the same group, the interaction can be very sparse in one dimension but relatively more observable in another dimension. So if there is one dimension with intensive interaction, simply averaging all the dimensions would overwhelm the structural information in other dimensions.

3.2 Total Modularity Maximization (TMM) Another variant is to maximize the overall modularity among all the dimensions. That is,

$$\max \bar{Q} = \frac{1}{d}\sum_{i=1}^{d} Q_i \tag{3.5}$$

where Q_i is the modularity in the i-th dimension. To compute the top eigenvectors, the matrix-vector multiplication is calculated as:

$$\frac{1}{d}\left[\sum_{i=1}^{d} A_i\mathbf{x} - \sum_{i=1}^{d}\frac{\mathbf{d}_i^T\mathbf{x}}{2m_i}\mathbf{d}_i\right]$$

This strategy considers the degree distribution in each dimension separately whereas the previous AMM does not distinguish any degree distribution difference in each dimension. However, as mentioned in previous subsection, different groups participate in different dimensions of the network in a variety of degrees. It is not clear whether the modularity is directly comparable between dimensions and overall modularity maximization can work in practice.

4 Principal Modularity Maximization (PMM)

In previous section, we have pictured two baseline strategies to integrate multiple dimensions of a network. Now we shall show a two-phase strategy which circumvents the comparability of interactions in different dimensions.

4.1 Phase I: Structural Feature Extraction A concern with previous AMM and TMM is that they are not robust to noisy dimensions of a network. This motivates us to consider denoise each dimension of the network first. Recall that in Section 2, to maximize the modularity, we compute a low-dimensional embedding using the top eigenvectors of the modularity matrix. In other words, those selected eigenvectors represent possible splittings of community structure. Thus, the eigenvectors can be treated as the important structural features extracted from the network. Since those eigenvectors with negative or small eigenvalues contribute marginally to the modularity and are very likely to be noise, they should be abandoned. For a multi-dimensional network, we can extract structural features from each dimension. Only those eigenvectors with a positive eigenvalue should be kept. We can also retain only top-ranking community indicators to reduce noise. In our setting, only the top $2k$ eigenvectors are computed if we aim to find out k communities.

Algorithm: Principal Modularity Maximization

Input: $Net = \{A_1, A_2, \cdots, A_d\}$, k
Output: idx, S_i $(i = 1, 2, \cdots, d)$

1. Compute top $2k$ eigenvectors of the modularity matrix as in Eq (2) for each A_i via Lanczos method;
2. Select the vectors with positive eigenvalues as S_i;
3. Compute slim SVD of $X = [S_1, S_2, \cdots S_d] = UDV^T$;
4. Obtain lower-dimensional embedding $\widetilde{U} = U(:, k-1)$;
5. Calculate the cluster idx with k-means on $\widetilde{U}$.

Figure 2: PMM for Multi-Dimensional Networks

4.2 Phase II: Cross-Dimension Analysis Assume a latent community structure is shared across dimensions in a multi-dimensional network, a natural thinking is that maybe we can simply average the structural features to find out a shared pattern. However, the structural features extracted from each dimension are not unique. Dissimilar structural features do not necessarily indicate that the corresponding community structures are drastically different. Let S be the top-ℓ eigenvectors that maximize Q, and V an orthonormal matrix such that

$$V \in R^{\ell\times\ell},\ VV^T = I, V^TV = I$$

It can be verified that SV also maximize Q:

$$\begin{aligned} & \frac{1}{2m}tr((SV)^T B(SV)) \\ = & \frac{1}{2m}tr(S^T BSVV^T) \\ = & \frac{1}{2m}tr(S^T BS) \\ = & \max Q \end{aligned}$$

In other words, SV and S are equivalent under an orthogonal transformation. In the simplest case, $S' = -S$ is also a valid solution. We cannot average S_i directly.

Instead, we expect the structural features of different dimensions to be highly correlated after transformation. To capture the correlations between multiple sets of variables, (generalized) canonical correlation analysis (CCA) [12, 15, 23] is the standard statistical technique. CCA attempts to find a transformation for each set of variables such that the pairwise correlations are maximized. Here we briefly illustrate one scheme of generalized CCA which turns out to be equivalent to principal component analysis (PCA) in our specific case.

Let $S_i \in R^{n \times \ell_i}$ denote the structural features extracted from i-th dimension of the network, and $w_i \in R^{\ell_i}$ be the linear transformation applied to structural features of dimension i. The correlation between two dimensions after transformation is

$$R(i,j) = (S_i w_i)^T (S_j w_j) = w_i^T (S_i^T S_j) w_j = w_i^T C_{ij} w_j$$

with $C_{ij} = S_i^T S_j$ representing the covariance between the structural features of i-th and j-th dimensions. Generalized CCA attempts to maximize the summation of pairwise correlations as in the following form:

$$\text{(4.6)} \qquad \max \quad \sum_{i=1}^{d} \sum_{j=1}^{d} w_i^T C_{ij} w_j$$

$$\text{(4.7)} \qquad s.t. \quad \sum_{i=1}^{d} w_i^T C_{ii} w_i = 1$$

Using standard Lagrange multiplier and setting the derivatives respect to w_i to zero, we obtain equation below:

$$\text{(4.8)} \qquad \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1d} \\ C_{21} & C_{22} & \cdots & C_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ C_{d1} & C_{d2} & \cdots & C_{dd} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

$$\text{(4.9)} \qquad = \lambda \begin{bmatrix} C_{11} & 0 & \cdots & 0 \\ 0 & C_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_{dd} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

Recall that our structural features extracted from each dimension is essentially the top eigenvectors of the modularity matrix satisfying $S_i^T S_i = I$. Thus, the matrix $diag(C_{11}, C_{22}, \cdots, C_{dd})$ in Eq. (4.9) becomes an identity matrix. Hence $\mathbf{w} = [w_1, w_2, \cdots, w_d]^T$ corresponds the top eigenvector of the full covariance matrix in Eq. (4.8), which is equivalent to PCA applied to data of the following form:

$$\text{(4.10)} \qquad X = [S_1, S_2, \cdots, S_d]$$

To compute the $(k-1)$-dimension embedding, we just need to project the above data into the top $(k-1)$ principal vectors. Suppose $X = UDV^T$ is the SVD of X, it follows that the top $(k-1)$ vectors of U are the lower-dimensional embedding.

The detailed algorithm is summarized in Figure 2. In summary, we first extract structural features from each dimension of the network via modularity maximization; then PCA is applied on the concatenated data as in Eq. (4.10) to select the top eigenvectors. Thus, we name our approach as *Principal Modularity Maximization*. After projecting the data into the principal vectors, we obtain a lower-dimensional embedding which captures the principal pattern across all the dimensions of the network. Then we can perform k-means on this embedding to find out the discrete community assignment.

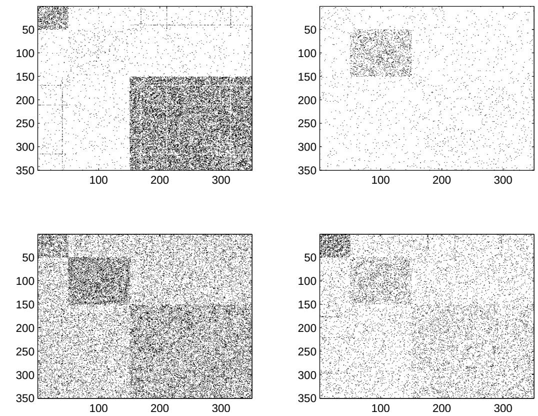
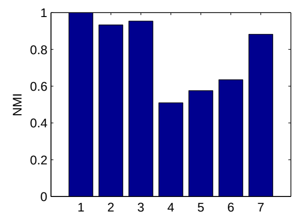
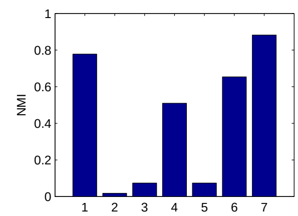


Figure 3: Example of Synthetic 4-Dimensional Network



(a) Comparable Dimensions



(b) Incomparable Dimensions

Figure 4: Performance of Various Strategies

5 Experiments

In this section, we compare and evaluate different strategies applied to multi-dimensional networks.

5.1 Experiments on Synthetic Data As a real-world network typically does not provide the ground truth information of community membership, we resort to synthetic data to conduct some controlled experiments. The synthetic data has 3 clusters, with each having 50, 100, 200 members respectively. There are 4 dimensions of interactions among these 350 social actors. For each dimension, group members connect with each other following a random generated within-group interaction probability. The interaction probability differs among groups of different dimensions. After that, we add some noise to the network by randomly connecting any two actors with low probability. The resulting interaction matrix is boolean with entries being either 0 or 1. Normalized mutual information (NMI) [25] is adopted to measure the clustering performance in the controlled experiments. NMI is a measure between 0 and 1. NMI=1 when two clusters are exactly the same.

Figure 3 shows one example of the generated multi-dimensional network. Clearly, different dimensions demonstrate different interaction patterns. Figure 4a shows the clustering performance in terms of NMI. The first 3 bars represent Principal/Total/Average modularity maximization, respectively. The last 4 bars denote the performance of community structure obtained via a single dimension of the network. Clearly, the first 3 methods which consider all dimensions in the network outperform those on single-dimensional networks. This could be easily explained by the patterns represented in Figure 3. The first dimension of the network actually only shows two groups, and the second dimension involves only one group with the other two hidden behind the noise. Thus, using a single view is very unlikely to recover the correct latent community structure. This is indicated by the low NMI of the first two dimensions (bars 4 and 5). Utilizing all the dimensions helps reduce the noise and uncover the shared community structure.

Comparing the three community detection strategies to handle multi-dimensional networks, our proposed principal modularity maximization outperforms the other two. To show the possible drawback of TMM and AMM, we insert some strong noise (with entries values ranging from 0 to 20) to the interaction matrix of the second dimension. Note that after this change, the performance of using the second dimension alone is decreasing from 0.5 to 0.1. That is, this dimension actually does not help identify the latent structure. With such a dominant dimension, both AMM and TMM fail. On the contrary, our proposed PMM still achieves reasonable good performance. This implies that PMM is more robust to noisy dimensions in multi-dimensional networks.

Figure 4 just shows one example. The average performance of each method plus its standard deviation over 100 runs is reported in Table 1. Clearly, multi-dimensional outperforms single-dimensional community detection method with lower variance. Due to the randomness of each run, it is not surprising that single-dimensional method shows larger variance. Among the three multi-dimensional modularity maximization strategies, PMM, with lowest variance, again outperforms the other two and is more stable.

5.2 Experiments on Social Media Data In this section, we examine our approach on real-world social media. A big challenge for evaluation is that the community membership information is often unknown in reality. Asking human subjects to manually verify and label the community membership for each user is acceptable but hardly can it scale to large online social networks. To address this issue, a cross-dimension network validation procedure is presented following the idea of cross validation in conventional data mining.

5.2.1 Cross-Dimension Network Validation Given a multi-dimensional network $Net = \{A_i | 1 \leq i \leq d\}$, we can use $d-1$ dimensions for training and the remaining one as test data. That is, we learn a community structure from

Table 1: Average Performance Over 100 Runs

	Strategy	Performance
Single-Dimensional	A_1	0.7237 ± 0.1924
	A_2	0.6798 ± 0.1888
	A_3	0.6672 ± 0.1848
	A_4	0.6906 ± 0.1976
Multi-Dimensional	AMM	0.7946 ± 0.1623
	TMM	0.9157 ± 0.1137
	PMM	$\mathbf{0.9351 \pm 0.1059}$

$d-1$ dimensions of the network. Based on the obtained community structure, we measure the modularity on the test dimension. It verifies how the learned community structure from other dimensions matches with the test dimension. A larger modularity implies more accurate community structure is discovered using the training data.

5.2.2 YouTube Data Collection YouTube[2] is currently the most popular video sharing web site. It is reported to "attract 100 million video views per day"[3]. As of March 17th, 2008, there have been 78.3 million videos uploaded, with over 200, 000 videos uploaded per day[4]. This social networking site allows users to interact with each other in various forms such as contacts, subscriptions, sharing favorite videos, etc. We use YouTube Data API[5] to crawl the contacts network, subscription network as well as each user's favorite videos. To avoid sample selection bias, we choose 100 authors of recently uploaded videos as the seed set, and expand the network via their contacts and subscriptions. We crawled a small portion of the whole network, with $30,522$ user profiles reaching in total $848,003$ users and $1,299,642$ favorite videos. After removing those users who decline to share their contact information, we have $15,088$ active user profiles in the network. Figure 5 shows the degree distribution in contacts network and favorite network. Both follow power law distribution as expected.

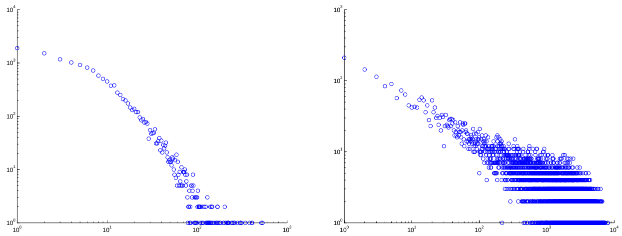
On issue is that the collected subscription network is directional. For such case, simply ignoring the direction mixes the two roles of the directional interaction (similar to the email communication example in the introduction). Instead, we decompose the asymmetric interaction A into two undirectional interactions:

$$A' = A * A^T; \tag{5.11}$$

$$A'' = A^T * A. \tag{5.12}$$

Essentially, if two social actors both subscribe to the same set of users, it is likely that they are similar and share the

[2] http://www.youtube.com/
[3] http://www.usatoday.com/tech/news/2006-07-16-youtube-views_x.htm
[4] http://ksudigg.wetpaint.com/page/YouTube+Statistics?t=anon
[5] http://code.google.com/apis/youtube/overview.html



(a) Contact Network (b) Favorite Network

Figure 5: Power law distribution on Different Dimensions

Table 2: The sparsity of each dimension

Network	Dimension	Density
A_1	contact	6.74×10^{-4}
A_2	co-contact	7.28×10^{-2}
A_3	co-subscription	4.90×10^{-2}
A_4	co-subscribed	1.97×10^{-2}
A_5	favorite	8.91×10^{-2}

same community; On the other hand, if two are referred by the same set of actors, their similarity tends to be higher than that of random pairs. This is similar to the two roles of hub and authority of web pages as mentioned in [16] and is also adopted for semi-supervised learning on directed graphs [32].

To utilize all aspects of information in our collected data, we construct a 5-dimensional network:

A_1: contact network: the contact network among those $15,088$ active users;

A_2: co-contact network: two active users are connected if they both add another user as contact; This is constructed based on all the reachable 848,003 users in our collected data.

A_3: co-subscription network: the connection between two users denotes they subscribe to the same user; constructed following Eq. (5.11);

A_4: co-subscribed network: two users are connected if they are both subscribed by the same user; constructed following Eq. (5.12);

A_5: favorite network: two users are connected if they share favorite videos.

The interactions in all dimensions are represented using boolean values. Table 2 shows the connection density of each dimension. Contact dimension is the most sparse one, while the other dimensions, due to the construction, are denser. As verified in later experiments, the other dimensions do not have a strong community structure as presented in A_1.

Table 3: Performance when k=10

Methods	A_1	A_2	A_3	A_4	A_5
A_1	—	.0327	.0264	.0412	.0088
A_2	.1746	—	.0495	.0879	.0238
A_3	.1613	.0469	—	.0975	.0223
A_4	.2475	.0532	.0735	—	.0171
A_5	.1040	.0426	.0369	.0423	—
AMM	.3236	.0728	.0814	.1228	.0333
TMM	.3374	.0752	.0689	.1192	**.0370**
PMM	**.4714**	**.0796**	**.1039**	**.1526**	.0316

Table 4: Performance when k=20

Methods	A_1	A_2	A_3	A_4	A_5
A_1	—	.0265	.0297	.0438	.0089
A_2	.1770	—	.0251	.0494	.0132
A_3	.1310	.0265	—	.0618	.0119
A_4	.2025	.0353	.0474	—	.0117
A_5	.0867	.0289	.0240	.0316	—
AMM	.2768	.0424	.0473	.0785	.0232
TMM	.3007	.0411	.0486	.0851	.0237
PMM	**.3985**	**.0550**	**.0626**	**.1004**	**.0248**

5.2.3 Comparative Study AMM, TMM and PMM as well as single-dimensional modularity maximization are compared. We cluster the active users involved in the network into different number of communities ranging from 10 to 90. The clustering performance of single-dimensional and multi-dimensional methods when $k = 10$ or 20 are presented in Table 3 and Table 4. We omit the detailed results for other cases as a similar trend is observed. In both tables, the rows represent methods and the columns denote the dimensions used as test data. The bold face denotes the optimal performance in each column. Note that in our cross-dimension network validation procedure, the test dimension is not available during training, thus the diagonal entries for single-dimensional methods are not shown.

PMM is clearly the winner most of the time. The other dimensions except A_1 are quite noisy due to their construction process, hence yielding consistently lower modularity. A closer examination reveals that utilizing information of all the dimensions outperforms single-dimensional clustering. Comparing all the multi-dimensional clustering approaches, both AMM and TMM are not comparable to our proposed PMM. Typically, AMM $<$ TMM $<$ PMM. Part of the reason is TMM considers degree distribution in separate dimensions. Our algorithm, by removing noise in each dimension, achieves the most accurate community structure among all the methods. This is evident in the contact dimension. Figure 6 shows the performance of the multi-dimensional clustering methods with respect to number of clusters (k) on A_1. No matter how many clusters we set, PMM outperforms the other two methods with a significant margin.

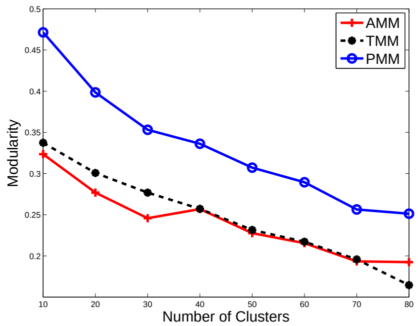


Figure 6: Performance comparison on Contact Dimension

Among all the multi-dimensional methods, AMM is most efficient as it only requires a single-dimensional clustering procedure on the average interaction matrix. Our proposed PMM needs to extract the important structural features from each network dimension, thus the total computation is more expensive. But this step can be easily paralleled with a multi-core CPU or clusters. The number of structural features extracted from each dimension is normally much smaller than the number of actors, resulting in a narrow data set X in Eq. (4.10). So the subsequent SVD computation to find out the shared latent structure is not too expensive. Another advantage we want to emphasize is that PMM not only finds the shared latent community structure, it also allows community identification in a specific dimension by utilizing the extracted structural features. This cannot be accomplished by AMM and TMM.

6 Related Work

Some works attempt to address unsupervised learning with many data sources or clustering results, such as cluster ensemble [25, 28, 9], consensus clustering [19, 13, 22, 10]. Most of the algorithms aim to find a robust clustering based on multiple clustering results either by sampling or different clustering algorithms. A similar idea is applied to community detection in social networks [11], in which a small portion of connections between nodes are randomly removed before each run, leading to multiple different clustering results. Those clusters occurring frequently are considered more stable reflecting the natural communities in reality. However, all the cluster ensemble methods concentrate on either one-dimensional network or attribute based data.

Another related field is multi-view clustering [4, 7, 18]. Bickel and Scheffere [4] propose co-EM and an extension of k-means and hierarchical clustering to handle data with two conditional independent views. Sa [8] creates a bipartite based on the two views and tries to minimize the disagreement. Multi-view clustering is also mentioned in [7] but in a different fashion: find all non-redundant clustering views existing in the same data set. Different spectral frameworks with multiple views are also studied in [31] and [18]. As for real-world social networks, one striking observation [17] is that spectral clustering always finds tight and small-scale but almost trivial communities (say, the community is connecting to the remaining network via one edge). Modularity maximization, on the other hand, tends to achieve better performance as it considers the highly skewed degree distribution of nodes [20]. A comparison between spectral clustering and modularity maximization within a large-scale multi-dimensional network is interesting and worthy of future work.

One concept we want to clarify is multi-mode network [27], which is different multi-dimensional network. Multi-mode network refers to the network with heterogeneous types of actors. Both within-mode and between-mode interactions can occur in a multi-mode network. But multi-dimensional network concentrates on networks with multiple dimensions among the same set of actors. A more general and complicated case is multi-mode multi-dimensional network, but it is beyond the scope of this work.

7 Conclusions

Multi-dimensional networks commonly exist in many social networking sites, reflecting diverse individual activities. In this work, we propose to detect the latent communal structure in a multi-dimensional network. We formally describe the community detection problem in multi-dimensional networks and discuss two straightforward extensions of modularity maximization from single-dimensional to multi-dimensional networks: Average modularity maximization (AMM) and total modularity maximization (TMM) .We show that both methods are not robust to handle networks with noise. We propose principal modularity maximization (PMM) to overcome this limitation. We extract structural features from each dimension of the network, which also effectively removes the noise in that dimension; and then apply cross-dimension analysis on the constructed data to find the lower-dimensional embedding such that the features extracted from all the dimensions are highly correlated to each other. Principal modularity maximization has been empirically shown to outperform single-dimensional clustering as well as AMM and TMM in both synthetic and YouTube data. Its superiority is most observable when a certain dimension of the network is rather noisy.

In current work, the network information of each dimension is assumed to be complete. But in reality, some actors decline to share their friends information. How to identify community structure with noisy incomplete network information is a challenge. It would be exciting as well to study the specific group structure of each dimension. Are there different group structure in each dimension? How are they correlated? Could we utilize dimension information for more effective viral marketing? All these questions are quite interesting for further research.

8 Acknowledgments

This work is, in part, supported by AFOSR-FA95500810132 and ONR-N000140810477.

References

[1] Nitin Agarwal, Huan Liu, Lei Tang, and Philip S. Yu. Identifying the influential bloggers in a community. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 207–218, New York, NY, USA, 2008. ACM.

[2] Sitaram Asur, Srinivasan Parthasarathy, and Duygu Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 913–921, 2007.

[3] Nilesh Bansal, Fei Chiang, Nick Koudas, and Frank Wm. Tompa. Seeking stable clusters in the blogosphere. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 806–817, 2007.

[4] Steffen Bickel and Tobias Scheffere. Multi-view clustering. In *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 19–26, Washington, DC, USA, 2004. IEEE Computer Society.

[5] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. Maximizing modularity is hard. *Arxiv preprint physics/0608255*, 2006.

[6] Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1):2, 2006.

[7] Ying Cui, Xiaoli Z. Fern, and Jennifer G. Dy. Non-redundant multi-view clustering via orthogonalization. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 133–142, 2007.

[8] Virginia R. de Sa. Spectral clustering with two views. In *Proceedings of Workshop of Learning with Multiple Views*, 2005.

[9] Xiaoli Zhang Fern and Carla E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, 2004.

[10] Andrey Goder and Vladimir Filkov. Consensus clustering algorithms: Comparison and refinement. In *SDM*, 2008.

[11] John Hopcroft, Omar Khan, Brian Kulis, and Bart Selman. Natural communities in large linked networks. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 541–546, New York, NY, USA, 2003. ACM.

[12] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[13] Tianming Hu and Sam Yuan Sung. Consensus clustering. *Intell. Data Anal.*, 9(6):551–565, 2005.

[14] Hyunmo Kang, Lise Getoor, and Lisa Singh. Visual analysis of dynamic group membership in temporal social networks. *SIGKDD Explorations, Special Issue on Visual Analytics*, 9(2):13–21, dec 2007.

[15] J.R. Kettenring. Canonical analysis of several sets of variables. *Biometrika*, 58:433–451, 1971.

[16] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[17] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Statistical properties of community structure in large social and information networks. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 695–704, New York, NY, USA, 2008.

[18] Bo Long, Philip S. Yu, and Zhongfei (Mark) Zhang. A general model for multiple view unsupervised learning. In *SDM*, 2008.

[19] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Mach. Learn.*, 52(1-2):91–118, 2003.

[20] Mark Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(3), 2006.

[21] Mark Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, 2006.

[22] Nam Nguyen and Rich Caruana. Consensus clusterings. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, 2007.

[23] Allan Aasbjerg Nielsen. Multiset canonical correlations analysis and multispectral, truly multitemporal remote sensing data. *Image Processing, IEEE Transactions on*, 11(3):293–305, Mar 2002.

[24] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, pages 61–70, 2002.

[25] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2003.

[26] Lei Tang, Huan Liu, Jianping Zhang, Nitin Agarwal, and John J. Salerno. Topic taxonomy adaptation for group profiling. *ACM Trans. Knowl. Discov. Data*, 1(4):1–28, 2008.

[27] Lei Tang, Huan Liu, Jianping Zhang, and Zohreh Nazeri. Community evolution in dynamic multi-mode networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 677–685, New York, NY, USA, 2008. ACM.

[28] Alexander Topchy, Anil K. Jain, and William Punch. Combining multiple weak clusterings. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, 2003. IEEE Computer Society.

[29] Stanley Wasserman and Katherine Faust. *Social Netwok Analysis: Methods and Applications*. Cambridge University Press, 1994.

[30] Scott White and Padhraic Smyth. A spectral clustering approaches to finding communities in graphs. In *SDM*, 2005.

[31] Dengyong Zhou and Christopher J. C. Burges. Spectral clustering and transductive learning with multiple views. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 1159–1166, New York, NY, USA, 2007. ACM.

[32] Dengyong Zhou, Bernhard Schölkopf, and Thomas Hofmann. Semi-supervised learning on directed graphs. In *NIPS*, 2005.