A Spectral Clustering Approach to Optimally Combining Numerical Vectors with a Modular Network

Motoki Shiga
Bioinformatics Center
Kyoto University
Gokasho Uji 611-0011, Japan
shiga@kuicr.kyoto-u.ac.jp

Ichigaku Takigawa
Bioinformatics Center
Kyoto University
Gokasho Uji 611-0011, Japan
takigawa@kuicr.kyoto-u.ac.jp

Hiroshi Mamitsuka
Bioinformatics Center
Kyoto University
Gokasho Uji 611-0011, Japan
mami@kuicr.kyoto-u.ac.jp

ABSTRACT

We address the issue of clustering numerical vectors with a network. The problem setting is basically equivalent to constrained clustering by Wagstaff and Cardie [20] and semi-supervised clustering by Basu et al. [2], but our focus is more on the optimal combination of two heterogeneous data sources. An application of this setting is web pages which can be numerically vectorized by their contents, e.g. term frequencies, and which are hyperlinked to each other, showing a network. Another typical application is genes whose behavior can be numerically measured and a gene network can be given from another data source. We first define a new graph clustering measure which we call *normalized network modularity*, by balancing the cluster size of the original modularity. We then propose a new clustering method which integrates the cost of clustering numerical vectors with the cost of maximizing the normalized network modularity into a spectral relaxation problem. Our learning algorithm is based on spectral clustering which makes our issue an eigenvalue problem and uses k-means for final cluster assignments. A significant advantage of our method is that we can optimize the weight parameter for balancing the two costs from the given data by choosing the minimum total cost. We evaluated the performance of our proposed method using a variety of datasets including synthetic data as well as real-world data from molecular biology. Experimental results showed that our method is effective enough to have good results for clustering by numerical vectors and a network.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

KDD'07, August 12–15, 2007, San Jose, California, USA.

General Terms

Algorithms and Experimentation

Keywords

Network modularity, spectral clustering, heterogeneous data sources, k-means, eigenvalue problem

1. INTRODUCTION

Clustering, a major research subject in data mining, has been successfully applied to a wide variety of areas in the real world. In this paper, we address the issue of clustering numerical vectors with a network. This is a general setting which can be found in a lot of applications and basically equivalent to constrained clustering by Wagstaff and Cardie [20] and semi-supervised clustering by Basu et al. [2], but our focus is more on the optimal combination of two heterogeneous data sources, numerical vectors and a network.

A typical application is web pages. This case, web pages will be clustered by their contents, say term frequencies, based on the assumption that if the contents of a page are very similar to those of another, these pages can be in the same cluster. In contrast, web pages are linked together, forming a network in which nodes and edges correspond to web pages and hyperlinks between them, respectively, and can be clustered based on the hyperlink connectivity. Clustering web pages based on hyperlinks is exactly graph partitioning. Standard criteria for graph partitioning are ratio cut [8] and normalized cut [15]. Simply speaking, these criteria are to minimize the number of inter-cluster edges relevant to the size of a cluster. The number of inter-cluster edges which is called *graph min-cut* is to check the partitioning performance while the size of a cluster is to avoid a small cluster which might be generated by outliers. In other words, these criteria are obtained by normalizing the graph min-cut by the cluster size. Given a network, minimizing normalized cut (or ratio cut) is a trace optimization problem which is NP-hard. Thus usually this problem is converted by spectral relaxation into an optimization problem with a constraint which can be solved by Lagrange multipliers, and the solution is given by an eigenvalue problem. This is called spectral graph clustering which is difficult to assign a cluster label to each node definitely. Thus usually k-means is finally applied to cluster assignments using the resultant eigenvectors.

In the problem setting of graph partitioning, a numerical

weight can be attached to each edge of a given graph, and you may think that the similarity between two web pages can be a weight for the hyperlink between them. However, we assume that a given graph and a given set of numerical vectors are independently observed. This assumption is natural. For example, the contents of web pages and their links are independently generated. More concretely, there is a case that two web pages are very similar to each other, even if there are no links between them. Thus we note that our problem cannot be solved directly by using an existing graph partitioning method only.

Another application is genes. Genes are expressed and function in a cell. Currently the quantitative expression of thousands of genes can be measured simultaneously by using the latest technology in genetic engineering, called cDNA microarray. We can have a numerical vector (generally called a profile) for each gene by repeating the experiment of cDNA microarray. However cDNA microarray data is very noisy and unreliable. Thus naturally we often need another data source in clustering genes, since precise gene clustering is important in predicting gene function [16]. We can have more reliable information on genes as a gene network, although they are confined to relatively well-studied genes. For example, literature information provides us with the co-occurrence frequencies of genes in medical documents which can be turned into a network of genes by setting a cut-off value against the frequencies. Similarly, metabolic or gene regulatory networks which are generated from literature are much more reliable than microarray data.

A potential approach for our problem setting would be to integrate the two data sources, numerical vectors and a network. A typical example of this direction is semi-supervised clustering based on a hidden Markov random field (HMRF) [2]. Semi-supervised clustering by HMRF is clustering numerical vectors by minimizing the objective function containing squared Euclidean distance as well as weighted network constraints. This method was extended to a more general framework in which the objective function can be expressed as a trace optimization problem for which an efficient weighted kernel k-means algorithm was proposed [11]. This work is based on the idea that minimizing the cost (or the objective function) of semi-supervised clustering by HMRF can be a trace optimization problem, which is true of minimizing the objective function of the weighted kernel k-means and more generally, minimizing a graph cut criterion such as normalized cut or ratio cut [3]. This work inspired us to combine numerical vectors with a network, but we note that our criterion for graph partitioning is clearly different from that in [11] and our focus is more on optimally combining the two data sources in terms of clustering.

Recent analysis on networks in the real-world data have revealed that they have some common global characteristics such as small-world phenomena [21], scale-free property [1], self-similarity [17] and hierarchical modularity [14]. We can expect that the performance of clustering in our problem setting might be improved by using some global network property than the vicinity information like the Markov property.

In light of the above, we propose a new method for clustering numerical vectors with a network. We focus on *network modularity*, a global feature found in a lot of real-world networks such as gene networks [14] and must be a powerful criterion for clustering by the graph connectivity. The original network modularity [13, 7, 6] is, intuitively, the number of intra-cluster edges minus the square of the number of inter-cluster edges. That is, this measure is given by using only the edges of a graph and is not balanced by the cluster size. Thus we first define *normalized network modularity* which is obtained by dividing the original network modularity by the cluster size. We then integrate the normalized network modularity with the cost of clustering numerical vectors into the framework of a trace optimization problem. Our clustering algorithm is based on spectral clustering by which our issue is relaxed into an eigenvalue problem and the final clusters are assigned by k-means clustering algorithm from the resultant eigenvectors. We stress that our work is an approach for clustering with not only numerical vectors but the network modularity. A significant merit of our method is that we can optimize the weight parameter for balancing the two data sources by choosing that which minimizes the total cost.

We evaluated the performance of our method using three types of datasets including both synthetic and real-world data. Our first dataset was synthetic both in numerical vectors and a network. Numerical vectors were generated randomly according to a mixture of von Mises-Fisher distributions, and a network was generated by selecting node pairs randomly. We confirmed the effectiveness of our method by checking normalized mutual information (NMI), a measure to check the performance of clustering methods, and the total cost of clustering, with varying the value of the weight parameter for balancing the two data sources. In particular, we found that NMI was mostly maximized at the weight parameter value which minimized the total cost, meaning that our strategy of choosing the weight parameter value of the minimum total cost worked successfully. The second dataset was synthetic numerical vectors and a real metabolic network having the scale-free property and unbalanced cluster sizes. From the experimental results on this dataset, we showed that our method of optimally combining numerical vectors with a given network worked favorably against even a real scale-free network with unbalanced cluster sizes. The third dataset was real microarray expression profiles corresponding to numerical vectors and the real gene network used in the second dataset. We note that gene expression measured by microarray is heavily noisy and unreliable while the network we used is from a database which is manually curated and trustworthy. Interestingly the resultant weight parameter value was extremely biased to the network information, being consistent with the above reliability fact of the two input data sources.

2. METHOD

2.1 Preliminaries and Notations

We describe the notations that will be used throughout this paper.

Let N be the number of given numerical vectors (data points). Let $\boldsymbol{E}$ be the $N \times N$ matrix whose entries are all one. Let $\boldsymbol{X} := (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N)$ be given numerical vectors. Each $\boldsymbol{x}_n$ has p entries, and let $x_n(i)$ be the i-th entry of $\boldsymbol{x}_n$. Let $\bar{\boldsymbol{X}} := (\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_2, \ldots, \bar{\boldsymbol{x}}_N)$ where $\bar{\boldsymbol{x}}_n := \boldsymbol{x}_n / \sqrt{\sum_{i=1}^{p} x_n(i)^2}$. $\boldsymbol{Y} = \bar{\boldsymbol{X}}^T \bar{\boldsymbol{X}}$. Let $\boldsymbol{G}$ be a given network with N nodes and edges. Let $\boldsymbol{W} \in \Re^{N \times N}$ be a non-negative, symmetric matrix whose (i, j) entry, w_{ij} is a non-negative weight

between nodes i and j. If there is no edge between nodes i and j, w_{ij} is zero. We note that in our problem setting, $\boldsymbol{W}$ is an input having all information on a given graph $\boldsymbol{G}$ and is often called a *weight matrix* or an *affinity matrix*. Let $\bar{\boldsymbol{W}}$ be a matrix whose (i,j) entry $\bar{w}_{ij}$ satisfies that $\bar{w}_{ij} = w_{ij}/\sum_{j=1}^{N} w_{ij}$. Let $\boldsymbol{D}_d$ be a $N \times N$ diagonal matrix whose (i,i) entry d_i satisfies that $d_i = \sum_{j=1}^{N} w_{ij}$. Let $\boldsymbol{D} := \boldsymbol{d}^T\boldsymbol{d}$ where $\boldsymbol{d} := (d_1, \cdots, d_N)$. Let $\boldsymbol{M}$ be a matrix which satisfies $\boldsymbol{M} := \boldsymbol{D}_d^{-1}\boldsymbol{W}$.

Let K be the number of clusters which is an input. Let $\boldsymbol{I}_K$ be the identity matrix of size K. Let $\boldsymbol{Z} := (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_K)$ be an unsigned cluster assignment in which $\boldsymbol{z}_k^T = (z_{k,1}, \cdots, z_{k,N})$ where $z_{k,n}$ ($\in \{0,1\}$) is 1 if $\boldsymbol{x}_n$ is in cluster k, otherwise zero. $\tilde{\boldsymbol{Z}} := \frac{\boldsymbol{Z}}{\sqrt{\boldsymbol{Z}^T\boldsymbol{Z}}}$. Let $\boldsymbol{\mu} := (\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K)$ where $\boldsymbol{\mu}_k$ be the representative (or the cluster center) of cluster k. Let $\mathcal{Z}_k$ be a set of nodes in a given graph (or numerical vectors) in cluster k, and let $|\mathcal{Z}_k|$ be the number of all nodes in cluster k. $\mathcal{Z} := \cup_{k=1}^{K}\mathcal{Z}_k$. Let $\boldsymbol{V}$ be a diagonal matrix whose (k,k) entry is $|\mathcal{Z}_k|$. $L(\mathcal{Z}_k, \mathcal{Z}_{k'}) := \sum_{i\in\mathcal{Z}_k}\sum_{j\in\mathcal{Z}_{k'}} w_{ij}$, and $L := L(\mathcal{Z}, \mathcal{Z})$.

Let J be a cost of clustering numerical vectors $\boldsymbol{X}$ (or/and nodes in network $\boldsymbol{G}$). Let ω be a numerical parameter which takes a value between zero and one, and balances the two data sources, i.e. numerical vectors $\boldsymbol{X}$ and network $\boldsymbol{G}$.

2.2 k-means

We first briefly review the k-means clustering algorithm which is widely used in a lot of applications. The cost (or the objective function) of the k-means algorithm is given as follows:

$$J_{num}(\boldsymbol{X}, \mathcal{Z}; \boldsymbol{\mu}) = \frac{1}{N}\sum_{k=1}^{K}\sum_{i\in\mathcal{Z}_k} Dist(\boldsymbol{x}_i, \boldsymbol{\mu}_k), \qquad (1)$$

where $Dist(\boldsymbol{x}_i, \boldsymbol{\mu}_k)$ is a distance between numerical vector $\boldsymbol{x}_i$ and cluster representative $\boldsymbol{\mu}_k$ of cluster k. We can use any distance such as Euclidean distance, cosine similarity and 1−Pearson correlation coefficient. In our experiments, we used cosine similarity which is used for clustering high-dimensional data such as text documents [25]:

$$Dist(\boldsymbol{x}_i, \boldsymbol{\mu}_k) = \frac{1}{2}\left(1 - \frac{\boldsymbol{x}_n^T\boldsymbol{\mu}_k}{\sqrt{\boldsymbol{x}_n^T\boldsymbol{x}_n}}\right).$$

The k-means algorithm minimizes the cost of Eq. (1) by repeating the following two steps alternately until convergence: 1) updating the cluster representative and 2) updating cluster labels. Figure 1 shows the pseudocode of the k-means clustering algorithm.

2.3 Maximizing Normalized Network Modularity

2.3.1 Spectral Graph Partitioning

We briefly review k-way graph partitioning which divides nodes of a given graph (network) into k clusters. The standard criteria to be minimized in k-way graph partitioning are *ratio cut* and *normalized cut* which are given as follows:

$$\text{Ratio cut:} \quad \sum_k \frac{L(\mathcal{Z}_k, \mathcal{Z}\setminus\mathcal{Z}_k)}{|\mathcal{Z}_k|}.$$

$$\text{Normalized cut:} \quad \sum_k \frac{L(\mathcal{Z}_k, \mathcal{Z}\setminus\mathcal{Z}_k)}{L(\mathcal{Z}_k, \mathcal{Z})}.$$

Input : $\boldsymbol{X}$, K, $\mathcal{Z}^{(0)}$, $\boldsymbol{\mu}^{(0)}$
Output : $\mathcal{Z}$, $\boldsymbol{\mu}$, J

k-means $(\boldsymbol{X}, K, \mathcal{Z}^{(0)}, \boldsymbol{\mu}^{(0)})$
1: $\mathcal{Z} \leftarrow \mathcal{Z}^{(0)}$, $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}^{(0)}$
2: $\bar{\boldsymbol{X}} \leftarrow \boldsymbol{X}/\sqrt{\boldsymbol{X}^T\boldsymbol{X}}$
3: **while** $J(\bar{\boldsymbol{X}}, \mathcal{Z}; \boldsymbol{\mu})$ is not converged **do**
4: $\quad \boldsymbol{\mu}_k^{(t+1)} \leftarrow \frac{1}{|\mathcal{Z}_k^{(t)}|}\sum_{j\in\mathcal{Z}_k^{(t)}} \bar{\boldsymbol{x}}_j \quad (k = 1, \cdots, K)$
5: $\quad \mathcal{Z}^{(t+1)} \leftarrow \arg\min_{\mathcal{Z}} J(\bar{\boldsymbol{X}}, \mathcal{Z}; \boldsymbol{\mu}^{(t+1)})$
6: $\quad \mathcal{Z} \leftarrow \mathcal{Z}^{(t+1)}$, $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}^{(t+1)}$, $J \leftarrow J(\bar{\boldsymbol{X}}, \mathcal{Z}; \boldsymbol{\mu})$
7: **end while**

Figure 1: Pseudocode of k-means.

The numerator which is common to the above two cuts is the so-called *graph min-cut*. If we use the graph min-cut only, the clustering result is very sensitive to outliers. That is, a small cluster might be formed, if this cluster is relatively isolated from other nodes. So we need to normalize the graph min-cut by the number of nodes (ratio cut) or the total weight (normalized cut) in each cluster.

We can see that the above criteria can be rewritten as follows:

$$\text{Ratio Cut:} \quad \sum_k \frac{\boldsymbol{z}_k^T(\boldsymbol{D}_d - \boldsymbol{W})\boldsymbol{z}_k}{\boldsymbol{z}_k^T\boldsymbol{z}_k}.$$

$$\text{Normalized Cut:} \quad \sum_k \frac{\boldsymbol{z}_k^T(\boldsymbol{D}_d - \boldsymbol{W})\boldsymbol{z}_k}{\boldsymbol{z}_k^T\boldsymbol{D}_c\boldsymbol{z}_k}.$$

Finding a set of clusters which minimizes this criterion is an NP-hard problem, and we then solve this problem by relaxing the discrete cluster indicator matrix to a real valued one. We can then have the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \mathrm{tr}(\tilde{\boldsymbol{Z}}^T(\boldsymbol{D}_d - \boldsymbol{W})\tilde{\boldsymbol{Z}}) \\ \text{subject to} \quad & \tilde{\boldsymbol{Z}}^T\tilde{\boldsymbol{Z}} = \boldsymbol{I}_K. \ \text{(Ratio Cut)} \\ & (\tilde{\boldsymbol{Z}}^T\boldsymbol{D}_d\tilde{\boldsymbol{Z}} = \boldsymbol{I}_K. \ \text{(Normalized Cut))} \end{aligned}$$

By using Lagrange multipliers, we can easily find that the solution of this optimization can be an eigenvalue problem. In a standard manner of spectral clustering, after solving the eigenvalue problem, we first select the resultant $K-1$ eigenvectors with the minimum eigenvalues. Then, since we cannot directly assign a cluster label to each numerical vector (or each node in a given graph) by the resultant eigenvectors, we apply the k-means clustering algorithm to the selected $K-1$ eigenvectors after their normalization.

2.3.2 Maximizing Normalized Network Modularity with Spectral Graph Partitioning

Network modularity is originally defined as follows [6]:

$$Q(\boldsymbol{G}) = \sum_{k=1}^{K}\left\{\frac{2e_k(\boldsymbol{G})}{L} - \left(\frac{g_k(\boldsymbol{G})}{L}\right)^2\right\}, \qquad (2)$$

where $e_k(\boldsymbol{G})$ is the number of edges in cluster k and $g_k(\boldsymbol{G})$ is the total sum of degrees over all nodes in cluster k. As shown in the above, the weight attached to each edge was not considered in the original definition of network modularity.

Hereafter, we incorporate the edge weight into the network modularity, meaning that the binary definition on each edge is turned into a numerical one. We note that the property of the network modularity is totally kept in this extension. Eq. (2) can then be rewritten by using only L as follows:

$$Q(\boldsymbol{W}) = \sum_{k=1}^{K} \left\{ \frac{L(\mathcal{Z}_k, \mathcal{Z}_k)}{L} - \left(\frac{L(\mathcal{Z}_k, \mathcal{Z})}{L} \right)^2 \right\}.$$

This measure is defined by the (weighted) number of edges only, meaning that the original modularity considers the number of edges only. More importantly, the original network modularity is not balanced by the cluster size, meaning that a cluster might become small when affected by outliers. Thus we define the new measure which we call *normalized network modularity* whose cost (or the objective function) is given as follows:

$$J_{net}(\boldsymbol{W}, \mathcal{Z}) = \sum_{k=1}^{K} \frac{N}{|\mathcal{Z}_k|} \left\{ \left(\frac{L(\mathcal{Z}_k, \mathcal{Z})}{L} \right)^2 - \frac{L(\mathcal{Z}_k, \mathcal{Z}_k)}{L} \right\}.$$

This equation indicates that the larger negative value of this cost a clustered network has, the higher normalized network modularity this network has. The problem of finding the set of clusters which minimizes this cost is NP-hard. We then apply the spectral clustering approach to minimize the cost of normalized network modularity, $J(\boldsymbol{W}, \mathcal{Z})$. In the following derivation, we partly borrow the idea of the spectral graph clustering by White and Smith which was developed for the original network modularity [22].

We can first modify the problem of minimizing $J(\boldsymbol{W}, \mathcal{Z})$ into the problem of minimizing the following trace:

$$J_{net}(\boldsymbol{W}, \mathcal{Z}) \quad = \quad \mathrm{tr} \left(\frac{\boldsymbol{Z}^T N \left(\frac{1}{L^2} \boldsymbol{D} - \frac{1}{L} \boldsymbol{W} \right) \boldsymbol{Z}}{\boldsymbol{Z}^T \boldsymbol{Z}} \right). \quad (3)$$

As shown in the spectral graph clustering using ratio (or normalized) cut, matrix $\boldsymbol{Z}$ must satisfy the following constraint since each node falls into one cluster only:

$$\boldsymbol{Z}^T \boldsymbol{Z} = \boldsymbol{V}.$$

We can then replace $\boldsymbol{Z}$ with $\tilde{\boldsymbol{Z}}$ and rewrite the trace optimization problem in the following by relaxing the discrete cluster indicators into real-valued indicators:

$$\begin{array}{ll} \text{minimize} & \mathrm{tr} \left(\tilde{\boldsymbol{Z}}^T \left(\frac{1}{L^2} \boldsymbol{D} - \frac{1}{L} \boldsymbol{W} \right) \tilde{\boldsymbol{Z}} \right) \\ \text{subject to} & \tilde{\boldsymbol{Z}}^T \tilde{\boldsymbol{Z}} = \boldsymbol{I}_K. \end{array}$$

The solution can be found via Lagrange multipliers in the following typical eigenvalue problem:

$$\left(\frac{1}{L^2} \boldsymbol{D} - \frac{1}{L} \boldsymbol{W} \right) \tilde{\boldsymbol{Z}} = \tilde{\boldsymbol{Z}} \Lambda,$$

where Λ is a diagonal matrix of Lagrange multipliers. This can be further modified using $\bar{\boldsymbol{W}}$, the normalized $\boldsymbol{W}$, into:

$$\left(\frac{1}{L} \bar{\boldsymbol{W}} - \frac{1}{L^2} \boldsymbol{E} \right) \tilde{\boldsymbol{Z}} = \tilde{\boldsymbol{Z}} \Lambda. \quad (4)$$

When $n \to \infty$, the second term of Eq. (4) approaches zero faster than the first term. In addition, we can approximate $\bar{\boldsymbol{W}}$ by $\boldsymbol{D}_d^{-1} \boldsymbol{W}$. We then approximate Eq. (4) by the following simple eigenvalue problem:

$$\boldsymbol{M} \tilde{\boldsymbol{Z}} = \tilde{\boldsymbol{Z}} \Lambda. \quad (5)$$

Input : $\boldsymbol{W}$, K, $\mathcal{Z}^{(0)}$, $\boldsymbol{\mu}^{(0)}$
Output : $\mathcal{Z}$

1: Compute $\boldsymbol{D}_d$ from $\boldsymbol{W}$.
2: $\boldsymbol{M} \leftarrow \boldsymbol{D}_d^{-1} \boldsymbol{W}$
3: Compute $\boldsymbol{H}$ of $\boldsymbol{M}$.
4: Normalize $\boldsymbol{H}$ into $\bar{\boldsymbol{H}}$.
5: $[\mathcal{Z}, \boldsymbol{\mu}, J] \leftarrow$ **k-means**$(\bar{\boldsymbol{H}}, K, \mathcal{Z}^{(0)}, \boldsymbol{\mu}^{(0)})$

Figure 2: Pseudocode of spectral clustering for normalized network modularity.

This modification allows $\boldsymbol{M}$ to be a very sparse matrix, meaning that we can reduce the computational cost of solving the eigenvalue problem in Eq. (5).

After solving Eq. (5), we have the resultant $K-1$ eigenvectors with the minimum eigenvalues. That is, we can have $N \times (K-1)$ matrix, $\boldsymbol{H} = (\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{K-1})$ where $\boldsymbol{h}_i$ be the i-th eigenvector of the selected $K-1$ eigenvectors. We then normalize this matrix into $N \times (K-1)$ matrix, $\bar{\boldsymbol{H}}$ in which (n,k) entry $\bar{h}_{nk}$ satisfies $\bar{h}_{nk} = h_{nk} / \sqrt{\sum_{k=1}^{K-1} h_{nk}^2}$ where h_{nk} is (n,k) entry of $\boldsymbol{H}$. This eigenmatrix $\bar{\boldsymbol{H}}$ can be an input of k-means. Figure 2 shows the pseudocode of this process.

2.4 Proposed Algorithm for Optimally Combining Numerical Vectors with Normalized Network Modularity

2.4.1 Spectral Clustering with Numerical Values and a Network

We describe our proposed algorithm for combining two data sources: numerical vectors and a given weighted network.

We first set the cost (the objective function) of clustering numerical vectors as follows:

$$J_{num}(\bar{\boldsymbol{X}}, \mathcal{Z}; \boldsymbol{\mu}) \quad = \quad \frac{1}{N} \sum_{k=1}^{K} \sum_{i \in \mathcal{Z}_k} \frac{1}{2} (1 - \bar{\boldsymbol{x}}_i^T \boldsymbol{\mu}_k),$$

where the cluster representative $\boldsymbol{\mu}_k$ of cluster k is given as follows:

$$\boldsymbol{\mu}_k \quad = \quad \frac{1}{|\mathcal{Z}_k|} \sum_{j \in \mathcal{Z}_k} \bar{\boldsymbol{x}}_j$$

The cost of k-means can be rewritten as follows:

$$\begin{aligned} J_{num}(\bar{\boldsymbol{X}}, \mathcal{Z}) &= \frac{1}{2N} \sum_{k=1}^{K} \sum_{i \in \mathcal{Z}_k} \left(1 - \bar{\boldsymbol{x}}_i^T \boldsymbol{\mu}_k \right) \\ &= \frac{1}{2N} \sum_{k=1}^{K} \sum_{i \in \mathcal{Z}_k} \left(1 - \bar{\boldsymbol{x}}_i^T \frac{1}{|\mathcal{Z}_k|} \sum_{j \in \mathcal{Z}_k} \bar{\boldsymbol{x}}_j \right) \\ &= \frac{1}{2} - \frac{1}{2N} \sum_{k=1}^{K} \frac{1}{|\mathcal{Z}_k|} \sum_{i \in \mathcal{Z}_k} \sum_{j \in \mathcal{Z}_k} \bar{\boldsymbol{x}}_i^T \bar{\boldsymbol{x}}_j \end{aligned}$$

and it can then be a trace optimization problem:

$$J_{num}(\bar{\boldsymbol{X}}, \mathcal{Z}) = \frac{1}{2} - \mathrm{tr} \left(\frac{\boldsymbol{Z}^T (2N)^{-1} \boldsymbol{Y} \boldsymbol{Z}}{\boldsymbol{Z}^T \boldsymbol{Z}} \right) \quad (6)$$

We then combine the cost of clustering numerical vectors which is shown in Eq. (6) with the cost of the normalized network modularity which is given in Eq. (3), using ω for balancing the two costs:

$$\begin{aligned} J_{total}(\bar{\boldsymbol{X}}, \boldsymbol{W}, \mathcal{Z}) &= \omega J_{net}(\boldsymbol{W}, \mathcal{Z}) + (1-\omega) J_{num}(\bar{\boldsymbol{X}}, \mathcal{Z}) \\ &= \text{tr}\left\{ \frac{\boldsymbol{Z}^T \left(\frac{\omega N}{L^2}\boldsymbol{D} - \frac{\omega N}{L}\boldsymbol{W} - \frac{1-\omega}{2N}\boldsymbol{Y}\right)\boldsymbol{Z}}{\boldsymbol{Z}^T\boldsymbol{Z}} \right\} \\ &= \text{tr}\left\{ \tilde{\boldsymbol{Z}}^T \left(\frac{\omega N}{L^2}\boldsymbol{D} - \frac{\omega N}{L}\boldsymbol{W} - \frac{1-\omega}{2N}\boldsymbol{Y}\right)\tilde{\boldsymbol{Z}} \right\} \end{aligned}$$

Finding a set of clusters minimizing the integrated cost is also an NP-hard problem, and so we solve this problem by relaxing discrete cluster indicators into real-valued ones. By doing so, we can have the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \text{tr}\left\{ \tilde{\boldsymbol{Z}}^T \left(\frac{\omega N}{L^2}\boldsymbol{D} - \frac{\omega N}{L}\boldsymbol{W} - \frac{1-\omega}{2N}\boldsymbol{Y}\right)\tilde{\boldsymbol{Z}} \right\} \\ \text{subject to} \quad & \tilde{\boldsymbol{Z}}^T\tilde{\boldsymbol{Z}} = \boldsymbol{I}_K \end{aligned} \tag{7}$$

This optimization problem can be also turned by Lagrange multipliers into the following eigenvalue problem for the total cost:

$$\boldsymbol{M}_\omega \tilde{\boldsymbol{Z}} = \tilde{\boldsymbol{Z}}\Lambda, \tag{8}$$

where

$$\boldsymbol{M}_\omega = \frac{\omega N}{L^2}\boldsymbol{D} - \frac{\omega N}{L}\boldsymbol{W} - \frac{1-\omega}{2N}\boldsymbol{Y}.$$

Once we have the above eigenvalue problem, we can use the same manner of clustering as done in spectral graph clustering. That is, given $N \times N$ matrix $\boldsymbol{M}$, we can obtain the resultant $K-1$ eigenvectors with the minimum eigenvalues, to generate $N \times (K-1)$ matrix $\boldsymbol{S} = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_{K-1})$ where $\boldsymbol{s}_i$ is the i-th eigenvector of the selected $K-1$ eigenvectors. We then use the k-means clustering algorithm, after normalizing $\boldsymbol{S}$ into $\bar{\boldsymbol{S}}$ which is the $N \times (K-1)$ matrix in which (n,k) entry $\bar{s}_{nk}$ satisfies $\bar{s}_{nk} = s_{nk}/\sqrt{\sum_{k=1}^{K-1} s_{nk}^2}$ where s_{nk} is (n,k) entry of $\boldsymbol{S}$.

As we saw in spectral graph clustering, the constraint given in Eq. (7) can be modified into another form. For example, we can use $\tilde{\boldsymbol{Z}}^T\boldsymbol{D}_d\tilde{\boldsymbol{Z}} = \boldsymbol{I}_K$ which was used for normalized cut in graph partitioning. This case, $\boldsymbol{M}_\omega$ is given as follows:

$$\boldsymbol{M}_\omega = \boldsymbol{D}_d^{-\frac{1}{2}}\left(\frac{\omega N}{L^2}\boldsymbol{D} - \frac{\omega N}{L}\boldsymbol{W} - \frac{1-\omega}{2N}\boldsymbol{Y}\right)\boldsymbol{D}_d^{-\frac{1}{2}}.$$

We used this constraint in our experiments, since normalized cut is more often used in graph partitioning than ratio cut. In addition, White and Smith [22] also used this modification when they practically applied their spectral graph clustering with the original network modularity to the real world datasets.

2.4.2 Estimating ω

The parameter ω depends on the spectral space which is generated by combining the two data sources, meaning that the choice of ω will heavily affect the clustering result. So we propose a method to estimate the optimal ω value from given two data sources. In this method, varying ω from zero to one, we choose the ω which gives the minimum total cost, J_{total}. Figure 3 shows the pseudocode of the whole procedure of our proposed algorithm.

Input : $\boldsymbol{X}$, $\boldsymbol{W}$, K, $\mathcal{Z}^{(0)}$, $\boldsymbol{\mu}^{(0)}$
Output : $\mathcal{Z}$

$\bar{\boldsymbol{X}} \leftarrow \boldsymbol{X}/\sqrt{\boldsymbol{X}^T\boldsymbol{X}}$
$\boldsymbol{Y} \leftarrow \bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}$
Compute $\boldsymbol{D}_d$ and $\boldsymbol{D}$ from $\boldsymbol{W}$
for $\omega = 0$ to 1 **do**
  $\boldsymbol{M}_\omega \leftarrow \boldsymbol{D}_d^{-\frac{1}{2}}\left(\frac{\omega N}{L^2}\boldsymbol{D} - \frac{\omega N}{L}\boldsymbol{W} - \frac{1-\omega}{2N}\boldsymbol{Y}\right)\boldsymbol{D}_d^{-\frac{1}{2}}$.
  Compute $\boldsymbol{S}$ of $\boldsymbol{M}_\omega$.
  Normalize $\boldsymbol{S}$ into $\bar{\boldsymbol{S}}$.
  $[\mathcal{Z}_\omega, \boldsymbol{\mu}_\omega, J_\omega] \leftarrow$ **k-means**$(\bar{\boldsymbol{S}}, K, \mathcal{Z}^{(0)}, \boldsymbol{\mu}^{(0)})$
end for
$\omega_{\min} \leftarrow \arg\min_\omega J_\omega$
$\mathcal{Z} \leftarrow \mathcal{Z}_{\omega_{\min}}$

Figure 3: Pseudocode of the proposed algorithm.

3. EXPERIMENTS

3.1 Dataset 1: Synthetic Numerical Vectors and Synthetic Random Network

3.1.1 Data

1. Synthetic Numerical Vectors: We first assume that numerical vectors are randomly generated according to a mixture of von Mises-Fisher distributions [12] in which each component corresponds to a cluster:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \alpha_k c_p(\kappa) e^{\kappa \boldsymbol{\mu}_k^T \boldsymbol{x}},$$

where α_k $(k = 1, \cdots, K)$ are mixture proportions satisfying that $\sum_{k=1}^{K}\alpha_k = 1$ and $0 \le \alpha_k \le 1$, and $c_p(\kappa)$ is given as follows:

$$c_p(\kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{p/2-1}(\kappa)},$$

where $I_{p/2-1}(\kappa)$ is the type 1 Bessel function of order p which is given as follows:

$$I_p(\kappa) = \frac{1}{\pi}\int_0^\pi e^{\kappa \cos t}\cos(pt)dt.$$

Interested readers should refer [4] on the method for randomly generating numerical values according to this distribution. We used the following settings: $K = 4$, $p = 3$, $\alpha_k = 0.25$ $(k = 1, \ldots, 4)$, $\boldsymbol{\mu}_1 = (0, 0, 1)^T$, $\boldsymbol{\mu}_2 = (0, 0, -0.5)^T$, $\boldsymbol{\mu}_3 = (-0.5, \frac{\sqrt{3}}{2}, -0.5)^T$, $\boldsymbol{\mu}_4 = (-0.5, -\frac{\sqrt{3}}{2}, -0.5)^T$. Another parameter, κ which is called *concentration parameter*, behaves like the inverse of the variance of numerical vectors in a cluster. Thus numerical vectors which are generated with a larger κ will be more concentrated on cluster representatives and be more easily clustered. Thus we changed κ in our experiments to check the effect of κ on clustering results.

2. Synthetic Random Network: To generate a network, we first assigned a cluster label to each node. We generated a network in which the number of nodes and the number of edges in each cluster are kept the same for all clusters. Thus the generated network was defined by three parameters: N_v (the number of nodes in a cluster), N_e (the

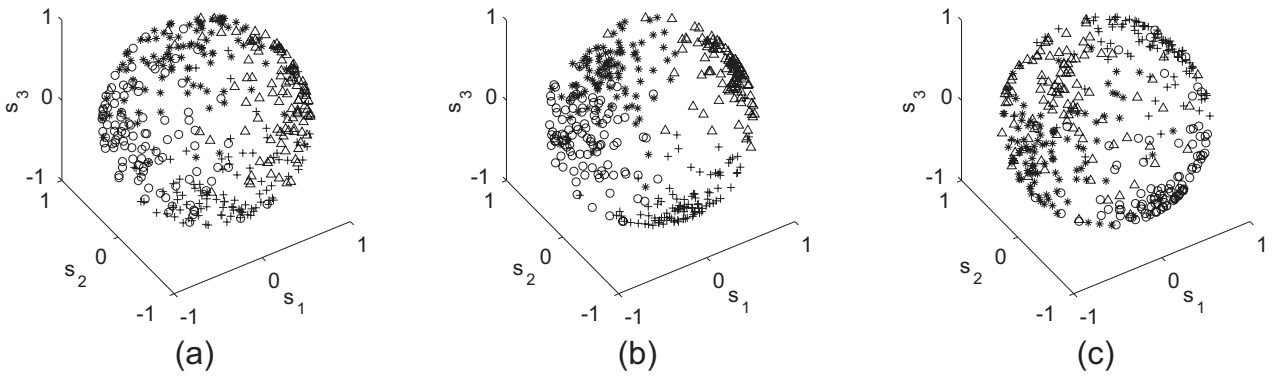


Figure 4: Dataset 1: The distribution of eigenvectors $\boldsymbol{s}_n$ $(n = 1, \ldots, 400)$ which are shown by four different symbols $(\circ, *, +, \triangle)$ corresponding to four different clusters at $N_{in} = 250$, $\kappa = 5$, (a) $\omega = 0$ and $J = 0.0932$, (b) $\omega = 0.3$ and $J = 0.0538$ and (c) $\omega = 1$ and $J = 0.0809$.

total number of edges) and N_{in} (the number of edges in a cluster) which is equal to $\frac{1}{2}L(\mathcal{Z}_k, \mathcal{Z}_k)$ $(k = 1, \cdots, K)$. We then chose a value of N_v to generate a set of nodes and randomly chose node pairs which are connected by edges to satisfy the values of N_e and N_{in}. In Dataset 1, we used $N_v = 100$ (meaning that the total number of nodes is 400) and $N_e = 1,600$, while N_{in} was changed to check how the clustering performance is affected by N_{in}.

3.1.2 Performance Results

To evaluate our clustering results using true cluster labels, we used normalized mutual information (NMI) [18] which has been used in a lot of applications to measure the performance of clustering methods [24]. A larger NMI value indicates a better clustering result. Interested readers should see [18] for the detail of NMI.

We first checked the distribution of $\boldsymbol{s}_n$ $(n = 1, \cdots, 400)$, i.e. the resultant eigenvectors of the eigenvalue problem in Eq. (8), when we fixed $\kappa = 5$ and $N_{in} = 250$. Figure 4 shows the three distributions of eigenvectors which are shown in four different symbols $(\circ, *, +, \triangle)$ corresponding to four different clusters, when ω was at 0, 0.3 and 1. This figure shows that the distribution of eigenvectors changes with varying ω. In particular, we can see that when $\omega = 0.3$, the eigenvectors were separated most clearly among the three cases. In fact, when ω=0, 0.3 and 1, J was 0.0932, 0.0538 and 0.0809, respectively, indicating that eigenvectors at $\omega = 0.3$ were the most concentrated on the cluster representatives among the three cases of ω.

We then checked the effectiveness of our method of optimally combining two different data sources by using the cost J and NMI, when ω is changed. We used each data set of all combinations of $\kappa = 1, 5$ and 50 and $N_{in} = 250, 280$ and 300. Figure 5 shows J of all these cases, and Figure 6 shows NMI of all these cases. From these figures, first of all, we can see that with increasing N_{in} ((a) $\rightarrow$ (b) $\rightarrow$ (c)), the modularity became higher, resulting with decreasing the cost (J) and increasing NMI. This might be clearer if we focus on the ω of one. On the other hand, we can see that with increasing κ $(1 \rightarrow 5 \rightarrow 50)$, numerical vectors were more concentrated on their cluster representatives, resulting with decreasing the cost (J) and increasing NMI. This might be also clearer if we focus on the ω of zero.

In all 18 curves in Figures 5 and 6, the best value is obtained when $0 < \omega < 1$, indicating that combining two data sources improved the cost J and NMI of $\omega = 0$ and $\omega = 1$. In particular, we emphasize that the ω value of the minimum J was mostly consistent with that of the maximum NMI. For example, when $N_{in} = 250$ and $\kappa = 1$, J was minimized at $\omega = 0.5$ and the maximum NMI was at $\omega = 0.4$. Similarly, at $\kappa = 5$, the minimum J was at $\omega = 0.4$ where the maximum NMI was obtained. This was true of $N_{in} = 280$ and $\kappa = 1$ where $\omega = 0.5$ provided the best in both NMI and J. These results imply that our method of selecting ω worked effectively for optimally combining numerical vectors with a given network. An interesting finding is that in a balanced case in which the cost (and NMI) of $\omega = 0$ is almost the same as that of $\omega = 1$, the curve became concave (and convex for NMI), indicating that the minimum cost (and the maximum NMI) can be easily found. On the other hand, in an unbalanced case such as $\kappa = 1$ and $N_{in} = 310$ in (c), the curve was not necessarily concave (and convex for NMI), meaning that only one data source (i.e. network information), is much more informative for clustering than the other (i.e. numerical vectors). This is natural, because numerical vectors at $\kappa = 1$ were widely distributed, not concentrating on the cluster representatives, and it would be difficult to do clustering by them, comparing to the case of $N_{in} = 310$ where clustering could be relatively easy by using network information only. Thus in such a case, our method might choose $\omega = 1$ (or $\omega = 0$), but this would be the right selection, since the NMI at $\omega = 1$ (or $\omega = 0$) must be the maximum or very close to the maximum in this case.

Using the same parameter setting as that in Figure 6, we finally checked NMI by repeating randomly generating datasets 400 times and averaging the performance over them. Figure 7 shows the averaged NMI obtained by this experiment. The curves in this figure were almost similar to those in Figure 6, implying that our results were very stable. Totally we can say that our method optimally combined the two synthetic data sources.

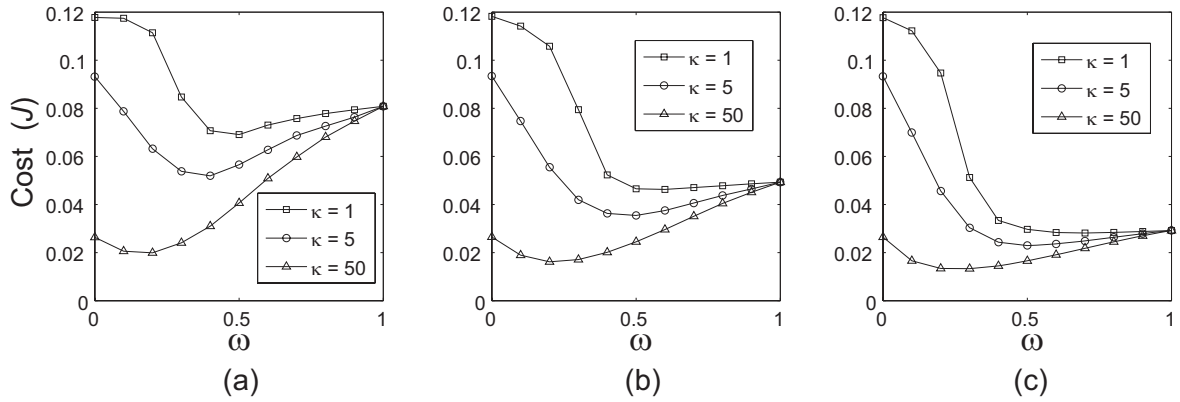


Figure 5: Dataset 1: J at N_{in} = (a) 250, (b) 280 and (c) 310.

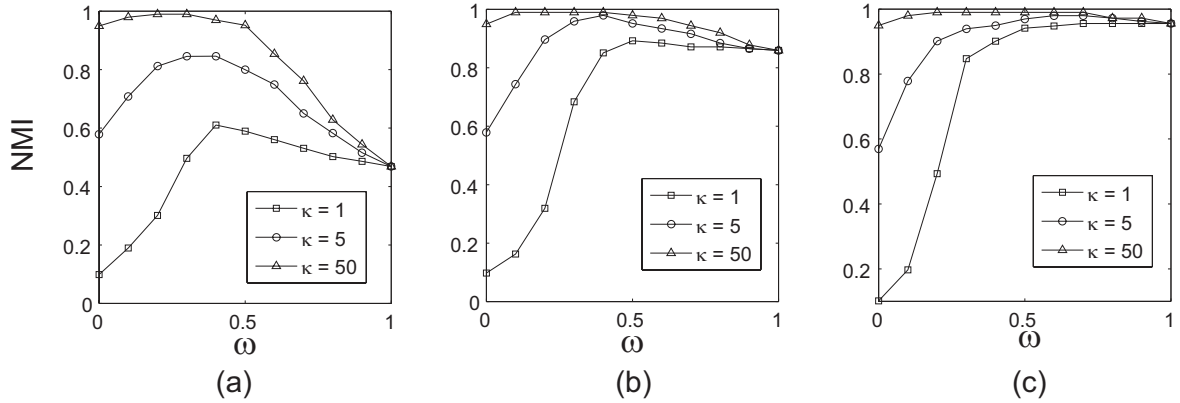


Figure 6: Dataset 1: NMI at N_{in} = (a) 250, (b) 280 and (c) 310.

3.2 Dataset 2: Synthetic Numerical Vectors and Real Scale-free Network

3.2.1 Data

1. Synthetic Numerical Vectors: Numerical values of a gene, such as gene expression, are usually measured experimentally, meaning that these values are noisy, comparing to the network which we can derive from a curated database in molecular biology[1]. Thus in this experiment, we generated synthetic numerical vectors to check the performance of our method of combining two data sources.

As we used a real metabolic network of 636 *Saccharomyces cerevisiae* genes (See below the way to generate this network.), we first assigned a cluster label to each node of the network in the following way: 1) We fixed the number of clusters and repeated running spectral graph clustering by White and Smith [22] 1,000 times on the real metabolic network, measuring the original network modularity on the resultant clusters at each time. 2) Out of the 1,000 runs, we then chose the clusters with the highest network modularity[2], and these clusters were used to assign a cluster label to each node. That is, these clusters were used as standard data for evaluation.

We then generated numerical vectors (corresponding to nodes in the metabolic network) in each cluster, assuming that they can be generated according to a component of the von Mises-Fisher distribution mixture as in Section 3.1.1. We used the following settings: $K = 10$, $p = 5$, $\alpha_k = 0.1$ $(k = 1, \cdots, 10)$, $\boldsymbol{\mu}_1 = (1, 0, 0, 0, 0)^T$, $\boldsymbol{\mu}_2 = (-1, 0, 0, 0, 0)^T$, $\boldsymbol{\mu}_3 = (0, 1, 0, 0, 0)^T$, $\boldsymbol{\mu}_4 = (0, -1, 0, 0, 0)^T$, $\boldsymbol{\mu}_5 = (0, 0, 1, 0, 0)^T$, $\boldsymbol{\mu}_6 = (0, 0, -1, 0, 0)^T$, $\boldsymbol{\mu}_7 = (0, 0, 0, 1, 0)^T$, $\boldsymbol{\mu}_8 = (0, 0, 0, -1, 0)^T$, $\boldsymbol{\mu}_9 = (0, 0, 0, 0, 1)^T$, $\boldsymbol{\mu}_{10} = (0, 0, 0, 0, -1)^T$. We changed κ to check how clustering results are affected by κ.

2. Real Metabolic Network: Metabolism is represented by a directed graph, called a *metabolic pathway* which shows biochemical processes of synthesizing small molecules in a cell. Each node is labeled by a chemical compound. A directed edge from a node, say node A, to another, say node B, is a chemical reaction, meaning that the compound corresponding to node B is synthesized from that for node A, and each edge is labeled by a (enzyme) gene which catalyzes the corresponding chemical reaction. From a metabolic path-

[1]We show this fact more clearly in the experiment using Dataset 3.

[2]We note that the clusters with the highest modularity cannot be obtained so easily by our method of $\omega = 1$, since in spectral graph clustering, the final solution is obtained by k-means and is always an approximation.

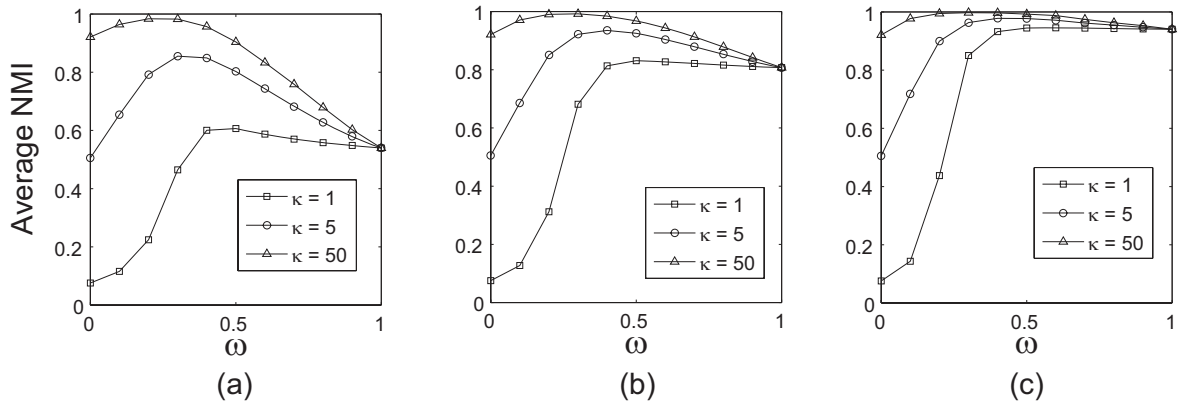


Figure 7: Dataset 1: Average NMI over 400 replicates at N_{in} = (a) 250, (b) 280 and (c) 310.

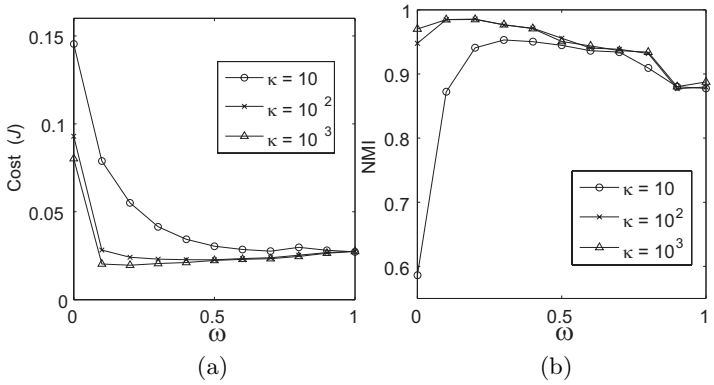


Figure 8: Dataset 2: Cost (J) and NMI.

way which is stored in the KEGG (Kyoto Encyclopedia of Genes and Genomes) database [10], we generated a network by connecting two genes by an undirected edge, if they catalyze two neighboring chemical reactions in the metabolic pathway. The generated network which we call *metabolic network* had 636 nodes and 3,104 edges. Most importantly, this network has two important network features: the scale-free property [1] as well as hierarchical network modularity [14].

3.2.2 Performance Results

We used NMI again to validate the clustering results obtained by our method, with the standard data for evaluation. Figure 8 (a) shows the total cost J with changing ω, and Figure 8 (b) shows NMI with changing ω. The results we can derive from these figures were mostly consistent with those obtained by Dataset 1. For example, at $\kappa = 10^2$ and 10^3 where the distribution of numerical vectors was relatively concentrated on their cluster representatives, the curves could be concave for J (and convex for NMI), and the ω value of the minimum cost and the ω value of the maximum NMI were easily found. Furthermore they were mostly consistent with each other. In addition, at $\kappa = 1$ where numerical vectors were distributed broadly, the curve was not necessarily a strong concave for the cost J (and convex for NMI), implying that this case, the network information would be more useful for clustering than the numerical vectors. In fact, NMI at $\omega = 1$ reached around 0.9, a very high value, whereas that at $\omega = 0$ stays at less than 0.6.

Figure 9 shows the clustering results obtained by our method at three different ω values when $\kappa = 10$. Ten different colors correspond to ten different clusters. Figure 10 shows the true cluster labels we used for both evaluation and generating numerical vectors. From these figures, we can easily see that the clustering result at $\omega = 0.5$ was the most similar to the true cluster labels among the three networks in Figure 9. For example, the center part of the true clusters were colored orange and dark blue, and this was consistent with the network at $\omega = 0.5$ in Figure 9 (b). On the other hand, this center part has a lot of different colors at $\omega = 0$ in Figure 9 (a) and is colored dark blue only at $\omega = 1$ in Figure 9 (c).

From these results, we can say that our method worked effectively for Dataset 2 which contains a real metabolic network with the scale-free property as well as unbalanced cluster sizes.

3.3 Dataset 3: Real Numerical Vectors and Real Scale-free Network

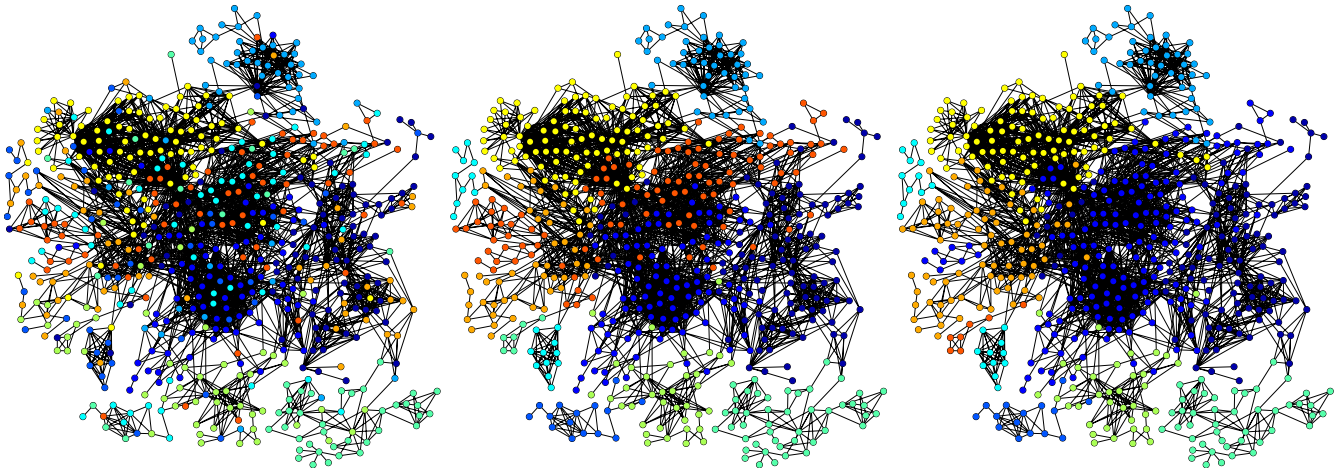
3.3.1 Data

1. Real Numerical Vectors: Microarray Gene Expression We used a microarray gene expression dataset [9] which has 300 expression profiles (numerical vectors) while around 200 missing values only. This dataset has been often used in the literature of microarray expression analysis [26, 23]. All missing values were interpolated by using the 10-nearest least square method by [19].

2. Real Metabolic Network: We used the real metabolic network in Dataset 2.

3.3.2 Performance Results

In order to evaluate the clustering result by our method, we used ten categories in metabolism which were stored in the KEGG database. At least one of the ten categories could be assigned to each metabolic gene. We note that these ten categories are not defined directly from the metabolic pathways in the KEGG database, and so these ten clusters cannot be identified by using the metabolic network in our



(a) (b) (c)

Figure 9: Dataset 2: Clustered metabolic networks at $\kappa = 10$ and ω= (a) 0, (b) 0.5 and (c) 1.

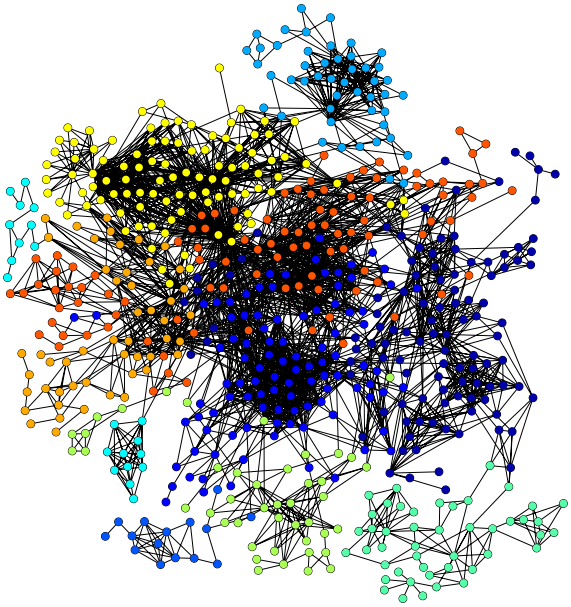
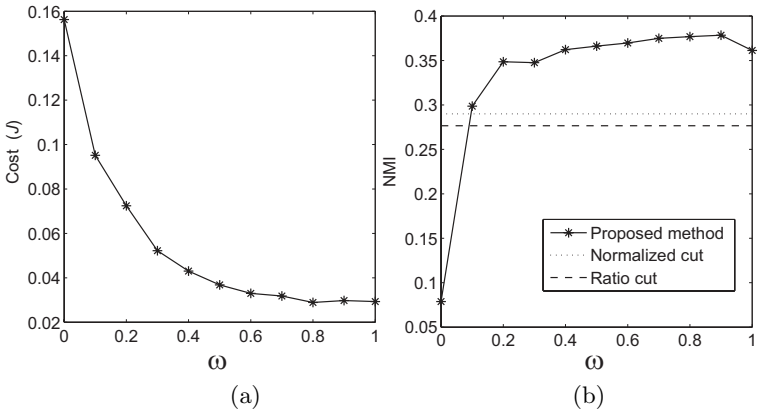


Figure 10: Dataset 2: True cluster labels.



(a) (b)

Figure 11: Dataset 3: Cost (J) and NMI.

experiment only. We then used NMI again to validate the performance of our clustering result.

Figure 11 (a) shows the cost J of our method with changing ω. Figure 11 (b) shows NMI of our method with changing ω. The curves in these figures were not strong concave for J (and convex for NMI), meaning that the two data sources are heavily unbalanced as pointed out in the experimental results of Dataset 1 and Dataset 2. In fact, by looking carefully, we can see that the minimum cost was at around $\omega = 0.8$ to 0.9 where the best NMI was obtained, and the difference between the best NMI and NMI at $\omega = 1$ was insignificant. Also Figure 11 (b) shows the averaged result over 200 runs of each of the two spectral graph partitioning methods using normalized cut and ratio cut. We note that these methods used graph information only, and the results of them are shown as dotted straight lines in the figure. Our method significantly outperformed these two methods even at $\omega = 1$, implying that normalized network modularity is more effective for clustering than normalized cut and ratio cut.

We repeated the same experiment replacing the above microarray dataset with datasets derived from a large microarray database [5], and found that the results were almost similar to the above case. From this result, we can say that the metabolic network derived from a curated database is more reliable in clustering metabolic genes than microarray expression. This result is consistent with existing understanding on data reliability in molecular biology.

For Dataset-3, the ω for the minimum cost took a value which was very close to one, around where NMI was maximum or very close to the maximum. Thus we think that our method succeeded for this dataset. As well in Dataset-2, our method worked favorably for optimally combining the real metabolic network with more reliable numerical vectors. Thus if we have more reliable numerical vectors on genes,

the clustering result would be improved much more. Over all we can say that our method itself is very promising.

4. CONCLUDING REMARKS

We have presented a new spectral approach to clustering numerical vectors with a network. The focus of our method was on network modularity, a key network property in clustering, and we defined a new criterion, normalized network modularity, for combining the two different data sources in the framework of spectral clustering. A significant advantage of our method is that we can optimize the weight parameter for balancing the two data sources, i.e. numerical vectors and a network. Also our algorithm is time-efficient, and practical computation time was less than one minute for any dataset in our experiment. Experimental results obtained by using three different types of datasets showed that our method worked favorably for optimally combining numerical vectors and a network.

In this paper, we have focused on two data sources, i.e. numerical vectors and a network, both in methodology and experiments. Our method can be easily extended to a more general framework for combining multiple heterogeneous data sources for clustering, and by doing so, the resultant clustering performance might be improved by adding another different data source. Thus interesting future work is to apply the proposed method to a variety of real-world datasets to characterize more systematically under which the proposed method works well. And this would be performed by not only two different data sources but also more than two heterogeneous data sources, say numerical vectors, a network and another different type of dataset. It would also be interesting to develop, in the context of clustering, a general criterion which can cover a lot of distances for numerical values as well as graph partitioning criteria such as normalized cut, ratio cut and network modularity.

5. ACKNOWLEDGMENTS

This work is supported in part by Bioinformatics Education Program "Education and Research Organization for Genome Information Science" and Kyoto University 21st Century COE Program "Knowledge Information Infrastructure for Genome Science" with support from MEXT, Japan.

6. REFERENCES

[1] A.-L. Barabási and A. Reka. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[2] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *KDD*, pages 59–68, August 2004.

[3] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means, spectral clustering and normalized cuts. In *KDD*, pages 551–556, 2004.

[4] I. S. Dhillon and S. Sra. Modeling data using directional distributions. Technical Report TR-06-03, University of Texas, Dept. of Computer Sciences, 2003.

[5] R. Edgar, M. Domrachev, and A. E. Lash. Gene expression omnibus: NCBI gene expression and hybridization array data repository. *NAR*, 30(1):207–210, 2002.

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

[7] R. Guimera, M. Sales-Pardo, and L. A. N. Amaral. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E*, 70:025101, 2004.

[8] L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE TCAD*, 11:1074–1085, 1992.

[9] T. R. Hughes et al. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–126, 2000.

[10] M. Kanehisa et al. From genomics to chemical genomics: new developments in KEGG. *NAR*, 34:D354–357, 2006.

[11] B. Kulis, S. Basu, I. Dhillon, and R. J. Mooney. Semi-supervised graph clustering: A kernel approach. In *ICML*, pages 457–464, 2005.

[12] K. V. Mardia and P. E. Jupp. *Directional Statistics*. John Wiley & Sons, second edition, 2000.

[13] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, 2004.

[14] E. Ravasz et al. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5589):1551–1555, 2002.

[15] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE PAMI*, 22(8):888–905, 2000.

[16] M. Shiga, I. Takigawa and H. Mamitsuka. Annotating gene function by combining expression data with a modular gene network. To appear in *ISMB*, 2007.

[17] C. Song, S. Havlin, and H. A. Makse. Self-similarity of complex networks. *Nature*, 433:392–395, 2005.

[18] A. Strehl and J. Ghosh. Relationship-based clustering and visualization for high-dimensional data mining. *INFORMS Journal on Computing*, 15(2):208–230, 2003.

[19] O. Troyanskaya et al. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[20] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *ICML*, pages 1103–1110, 2000.

[21] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[22] S. White and P. Smyth. A spectral clustering approach to finding communities in graphs. In *SDM*, pages 76–84, 2005.

[23] L. F. Wu et al. Large-scale prediction of saccharomyces cerevisiae gene function using overlapping transcriptional clusters. *Nat. Genet.*, 31(3):255–265, 2002.

[24] S. Zhong and J. Ghosh. A unified framework for model-based clustering. *JMLR*, 4:1001–1037, 2003.

[25] S. Zhong and J. Ghosh. Generative model-based document clustering: A comparative study. *KAIS*, 8(3):374–384, 2005.

[26] X. Zhou, M. C. Kao, and W. H. Wong. Transitive functional annotation by shortest-path analysis of gene expression data. *PNAS*, 99(20):12783–12788, 2002.