Unsupervised Learning on K-partite Graphs

Bo Long
Computer Science Dept.
SUNY Binghamton
Binghamton, NY 13902
blong1@binghamton.edu

Xiaoyun Wu
Yahoo! Inc.
701 First Avenue
Sunnyvale, California 94089
xiaoyunw@yahoo-inc.com

Zhongfei (Mark) Zhang
Computer Science Dept.
SUNY Binghamton
Binghamton, NY 13902
zzhang@binghamton.edu

Philip S. Yu
IBM Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
psyu@us.ibm.com

ABSTRACT

Various data mining applications involve data objects of multiple types that are related to each other, which can be naturally formulated as a k-partite graph. However, the research on mining the hidden structures from a k-partite graph is still limited and preliminary. In this paper, we propose a general model, the relation summary network, to find the hidden structures (the local cluster structures and the global community structures) from a k-partite graph. The model provides a principal framework for unsupervised learning on k-partite graphs of various structures. Under this model, we derive a novel algorithm to identify the hidden structures of a k-partite graph by constructing a relation summary network to approximate the original k-partite graph under a broad range of distortion measures. Experiments on both synthetic and real data sets demonstrate the promise and effectiveness of the proposed model and algorithm. We also establish the connections between existing clustering approaches and the proposed model to provide a unified view to the clustering approaches.

Categories and Subject Descriptions: E.4 [**Coding and Information Theory**]:Data compaction and compression; H.3.3[**Information search and Retrieval**]:Clustering; I.5.3[**Pattern Recognition**]:Clustering.

General Terms: Algorithms.

Keywords: K-partite graph, Unsupervised learning, Clustering, Relation summary network, Bregman divergence.

1. INTRODUCTION

Unsupervised learning approaches have traditionally focused on the homogeneous data objects, which can be represented either as a set of feature vectors or a homogeneous graph with nodes of a single type. However, many examples of real-world data involve objects of multiple types that are related to each other, which naturally form k-partite graphs of heterogeneous types of nodes. For example, documents and words in a corpus, customers and items in collaborative filtering, transactions and items in market basket, as well as genes and conditions in micro-array data all form a bi-partite graph; documents, words, and categories in taxonomy mining, as well as Web pages, search queries, and Web users in a Web search system all form a tri-partite graph; papers, key words, authors, and publication venues in a scientific publication archive form a quart-partite graph. In such scenarios, using traditional approaches to cluster each type of objects (nodes) individually may not work well due to the following reasons.

First, to apply traditional clustering approaches to each type of data objects individually, the relation information needs to be transformed into feature vectors for each type of objects. In general, this transformation results in high dimensional and sparse feature vectors, since after the transformation the number of features for a type of objects is equal to the number of all the objects which are possibly related to this type of objects. For example, if we transform the links between Web pages and Web users as well as search queries into the features for the Web pages, this leads to a huge number of features with sparse values for each Web page. Second, traditional clustering approaches are unable to tackle the interactions among the cluster structures of different types of objects, since they cluster data of a single type based on static features. Note that the interactions could pass along the relations, i.e., there exists influence propagation in a k-partite graph. Third, in some data mining applications, users are not only interested in the local cluster structure for each type of objects, but also the global community structures involving multi-types of objects. For example, in document clustering, in addition to document clusters and word clusters, the relationship between document clusters and word clusters is also useful information. It is difficult to discover such global structures by clustering each type of objects individually.

An intuitive attempt to mine the hidden structures from k-partite graphs is applying existing graph partitioning approaches to k-partite graphs. This idea may work in some special and simple situations. However, in general, it is infeasible. First, the graph partitioning theory focuses on finding the best cuts of a graph under a certain criterion and it is

KDD'06, August 20–23, 2006, Philadelphia, Pennsylvania, USA.

very difficult to cut different type of relations (links) simultaneously to identify different hidden structures for different types of nodes. Second, by partitioning the whole k-partite graph into m subgraphs, one actually assumes that all different types of nodes have the same number of clusters m, which in general is not true. Third, by simply partitioning the whole graph into disjoint subgraphs, the resulting hidden structures are rough. For example, the clusters of different types of nodes are restricted to one-to-one associations.

Therefore, mining hidden structures from k-partite graphs has presented a great challenge to traditional unsupervised learning approaches. In this study, first we propose a general model, the relation summary network, to find the hidden structures (the local cluster structures and the global community structures) from a k-partite graph. The basic idea is to construct a new k-partite graph with hidden nodes, which "summarize" the link information in the original k-partite graph and make the hidden structures explicit, to approximate the original graph. The model provides a principal framework for unsupervised learning on k-partite graphs of various structures. Second, under this model, based on the matrix representation of a k-partite graph we reformulate the graph approximation as an optimization problem of matrix approximation and derive an iterative algorithm to find the hidden structures from a k-partite graph under a broad range of distortion measures. By iteratively updating the cluster structures for each type of nodes, the algorithm takes advantage of the interactions among the cluster structures of different types of nodes and performs implicit adaptive feature reduction for each type of nodes. Experiments on both synthetic and real data sets demonstrate the promise and effectiveness of the proposed model and algorithm. Third, we also establish the connections between existing clustering approaches and the proposed model to provide a unified view to the clustering approaches.

2. RELATED WORK

Graph partitioning on homogeneous graphs has been studied for decades and a number of different approaches, such as spectral approaches [6, 26, 11] and multilevel approaches [5, 15, 19], have been proposed. However, the research on mining cluster structures from k-partite graphs of heterogeneous types of nodes is limited. Several noticeable efforts include [8, 18] and [13]. [8, 18] extends the spectral partitioning based on normalized cut to a bi-partite graph. After the deduction, spectral partitioning on the bi-partite graph is converted to a singular value decomposition (SVD). [13] partitions a star-structured k-partite graph based on semi-definite programming. In addition to the restriction that they are only applicable to the special cases of k-partite graphs, all these algorithms have the restriction that the numbers of clusters for different types of nodes must be equal and the clusters for different types of objects must have one-to-one associations.

The research on clustering multi-type interrelated objects is also related to this study. Clustering on bi-type interrelated data objects, such as word-document data, is called co-clustering or bi-clustering. Recently, co-clustering has been addressed based on matrix factorization. Both [23] and [21] model the co-clustering as an optimization problem involving a triple matrix factorization. [23] proposes an EM-like algorithm based on multiplicative updating rules and [21] proposes a hard clustering algorithm for binary data. [10] extends the non-negative matrix factorization to symmetric matrices and shows that it is equivalent to the Kernel K-means and the Laplacian-based spectral clustering.

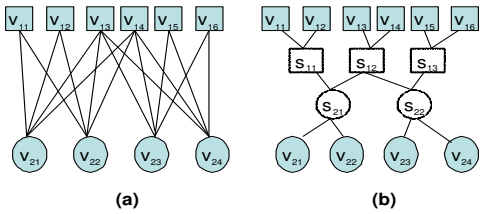

Figure 1: A bi-partite graph (a) and its relation summary network (b).

Some efforts on latent variable discovery are also related to co-clustering. PLSA [16] is a method based on a mixture decomposition derived from a latent class model. A two-sided clustering model is proposed for collaborative filtering by [17]. Information-theory based co-clustering has also attracted attention in the literature. [12] extends the information bottleneck (IB) framework [28] to repeatedly cluster documents and then words. [9] proposes a co-clustering algorithm to maximize the mutual information between the clustered random variables subject to the constraints on the number of row and column clusters. A more generalized co-clustering framework is presented by [3] wherein any Bregman divergence can be used in the objective function.

Comparing with co-clustering, clustering on the data consisting of more than two types of data objects has not been well studied in the literature. Several noticeable efforts are discussed as follows. [30] proposes a framework for clustering heterogeneous web objects, under which a layered structure with link information is used to iteratively project and propagate the cluster results between layers. Similarly, [29] presents an approach named ReCom to improve the cluster quality of interrelated data objects through an iterative reinforcement clustering process. However, there is no sound objective function and theoretical proof on the effectiveness of these algorithms. [22] formulates multi-type relational data clustering as collective factorization on related matrices and derives a spectral algorithm to cluster multi-type interrelated data objects simultaneously. The algorithm iteratively embeds each type of data objects into low dimensional spaces and benefits from the interactions among the hidden structures of different types of data objects.

To summarize, unsupervised learning on k-partite graphs has been touched from different perspectives due to its high impact in various important applications. Yet, systematic research is still limited. This paper attempts to derive a theoretically sound general model and algorithm for unsupervised learning on k-partite graphs of various structures.

3. MODEL FORMULATION

In this section, we derive a general model based on graph approximation to mine the hidden structures from a k-partite graph.

Let us start with an illustrative example. Figure 1(a) shows a bi-partite graph $G = (V_1, V_2, E)$ where $V_1 = \{v_{11}, \ldots, v_{16}\}$ and $V_2 = \{v_{21}, \ldots, v_{24}\}$ denote two types of nodes and E denotes the edges in G. Even though this graph is simple, it is non-trivial to discover its hidden structures. In Figure 1(b), we redraw the original graph by adding two sets of new nodes (called hidden nodes), $S_1 = \{s_{11}, s_{12}, s_{13}\}$ and $S_2 = \{s_{21}, s_{22}\}$. Based on the new graph, the cluster structures for each type of nodes are straightforward: V_1

has three clusters, $\{v_{11}, v_{12}\}$, $\{v_{13}, a_{14}\}$, and $\{v_{15}, v_{16}\}$, and V_2 has two clusters, $\{v_{21}, v_{22}\}$ and $\{v_{23}, b_{24}\}$. If we look at the subgraph consisting of only the hidden nodes in Figure 1(b), we see that it provides a clear skeleton for the global structure of the whole graph, from which it is clear how the clusters of different types of nodes are related to each other; for example, cluster s_{11} is associated with cluster s_{21} and cluster s_{12} is associated with both clusters s_{21} and s_{22}. In other words, by introducing the hidden nodes into the original k-partite graph, both the local cluster structures and the global community structures become explicit. Note that if we apply a graph partitioning approach to the bi-partite graph in Figure 1(a) to find its hidden structures, no matter how we cut the edges, it is impossible to identify all the cluster structures correctly.

Based on the above observations, we propose a model, the Relation Summary Network (RSN), to mine the hidden structures from a k-partite graph. The key idea of RSN is to add a small number of hidden nodes to the original k-partite graph to make the hidden structures of the graph explicit. However, given a k-partite graph, we are not interested in an arbitrary relation network. To ensure a relation summary network to discover the desirable hidden structures of the original graph, we must make RSN as "close" as possible to the original graph. In other words, we aim at an optimal relation summary network, from which we can re-construct the original graph as precisely as possible. Formally, we define an RSN as follows.

Definition 1. Given a distance function $\mathfrak{D}$, a k-partite graph $G = (V_1, \ldots, V_m, E)$, and m positive integers, $k_1, \ldots, k_m$, the relation summary network of G is a k-partite graph $G^s = (V_1, ..., V_m, S_1, ..., S_m, E^s)$, which satisfies the following conditions:

1. each instance node in V_i is adjacent to one and only one hidden node from S_i for $1 \leq i \leq m$ with unit weight;
2. $S_i \sim S_j$ in G^s if and only if $V_i \sim V_j$ in G for $i \neq j$ and $1 \leq i, j \leq m$;
3. $G^s = \arg\min_F \mathfrak{D}(G, F)$,

where S_i denotes a set of hidden nodes for V_i and $|S_i| = k_i$ for $1 \leq i \leq m$; $S_i \sim S_j$ denotes that there exist edges between S_i and S_j, and similarly $V_i \sim V_j$; F denotes any k-partite graph $(V_1, ..., V_m, S_1, ..., S_m, E^f)$ satisfying Condition 1 and 2.

In Definition 1, the first condition implies that in an RSN, the instance nodes (the nodes in V_i) are related to each other only through the hidden nodes. Hence, a small number of hidden nodes actually summarize the complex relations (edges) in the original graph to make the hidden structures explicit. Since in this study, our focus is to find disjoint clusters for each type of nodes, the first condition restricts one instance node to be adjacent to only one hidden node with unit weight; however, it is easy to modify this restriction to extend the model to other cases of unsupervised learning on k-partite graphs. The second condition implies that if two types of instance nodes V_i and V_j are (or are not) related to each other in the original graph, then the corresponding two types of hidden nodes S_i and S_j in the RSN are (or are not) related to each other. For example, Figure 2 shows a tri-partite graph and its RSN. In the original graph Figure 2(a), $V_1 \sim V_2$ and $V_1 \sim V_3$, and hence $S_1 \sim S_2$ and $S_1 \sim S_3$ in its RSN. The third condition states that the RSN is an optimal approximation to the original graph under a certain distortion measure.

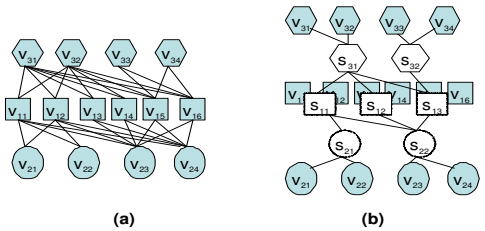

Figure 2: A tri-partite graph (a) and its RSN (b)

Next, we need to define the distance between a k-partite graph G and its RSN G^s. Without loss of generality, if $V_i \sim V_j$ in G, we assume that edges between V_i and V_j are complete (if there is no edge between v_{ih} and v_{jl}, we can assume an edge with weight of zero or other special value). Similarly for $S_i \sim S_j$ in G^s. Let $e(v_{ih}, v_{jl})$ denote the weight of the edge (v_{ih}, v_{jl}) in G. Similarly let $e^s(s_{ip}, s_{jq})$ be the weight of the edge (s_{ip}, s_{jq}) in G^s. In the RSN, a pair of instance nodes v_{ih} and v_{jl} are connected through a unique path $(v_{ih}, s_{ip}, s_{jq}, v_{jl})$, in which $e^s(v_{ih}, s_{ip}) = 1$ and $e^s(s_{jq}, v_{jl}) = 1$ according to Definition 1. The edge between two hidden nodes (s_{ip}, s_{jq}) can be considered as the "summary relation" between two sets of instance nodes, i.e., the instance nodes connecting with s_{ip} and the instance nodes connecting with s_{jq}. Hence, how good G^s approximates G depends on how good $e^s(s_{ip}, s_{jq})$ approximates $e(v_{ih}, v_{jl})$ for v_{ih} and v_{jl} which satisfy $e^s(v_{ih}, s_{ip}) = 1$ and $e^s(s_{jq}, v_{jl}) = 1$, respectively. Therefore, we define the distance between a k-partite graph G and its RSN G^s as follows:

$$\mathfrak{D}(G, G^s) = \sum_{i,j} \sum_{\substack{V_i \sim V_j, \\ v_{ih} \in V_i, v_{jl} \in V_j, \\ e^s(v_{ih}, s_{ip}) = 1, \\ e^s(s_{jq}, v_{jl}) = 1.}} D(e(v_{ih}, v_{jl}), e^s(s_{ip}, s_{jq})), \tag{1}$$

where $1 \leq i, j \leq m$, $1 \leq h \leq |V_i|$, $1 \leq l \leq |V_j|$, $1 \leq p \leq |S_i|$, and $1 \leq q \leq |S_j|$.

Let us have an illustrative example. Assume that the edges of the k-partite graph in Figure 1(a) have unit weights. If there is no edge between v_{ih} and v_{jl}, we let $e(v_{ih}, v_{jl}) = 0$. Similarly for its RSN in Figure 1(b). Assume that D is the Euclidean distance function. Hence, based on Eq. (1), $\mathfrak{D}(G, G^s) = 0$, i.e., from the RSN in Figure 1(b), we can reconstruct the original graph in Figure 1(a) without any error. For example, the path $(v_{13}, s_{12}, s_{21}, v_{22})$ in the RSN implies that there is an edge between v_{13} and v_{22} in the original graph such that $e(v_{13}, v_{22}) = e^s(s_{12}, s_{21})$. Following this procedure, the original graph can be reconstructed completely.

Note that different definitions of the distances between two graphs lead to different algorithms. In this study, we focus on the definition given in Eq.(1). One of the advantages of this definition is that it leads to a nice matrix representation for the distance between two graphs, which facilitates to derive the algorithm.

Definition 1 and Eq. (1) provide a general model, the RSN model, to mine the cluster structures for each type of nodes in a k-partite graph and the global structures for the whole graph. Compared with the traditional clustering approaches, the RSN model is capable of making use of the interactions (direct or indirect) among the hidden structures

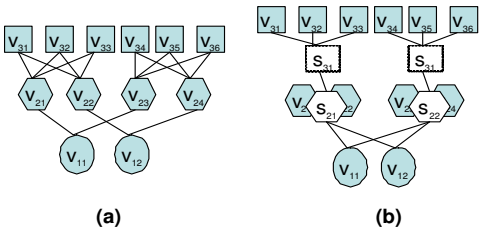

Figure 3: The cluster structures of V_2 and V_3 affect the similarity between v_{11} and v_{12} through the hidden nodes.

of different types of nodes, and through the hidden nodes performing implicit and adaptive feature reduction to overcome the typical high dimensionality and sparsity. Figure 3 shows an illustrative example of how the cluster structures of two types of instance nodes affect the similarity between two instance nodes of another type. Suppose that we are to cluster nodes in V_1 (only two nodes in V_1 are shown in Figure 3(a)). Traditional clustering approaches determine the similarity between v_{11} and v_{12} based on their link features, $[1, 0, 1, 0]$ and $[0, 1, 0, 1]$, respectively, and hence, their similarity is inappropriately considered as zero (lowest level). This is a typical situation in a large graph with sparse links. Now suppose that we have derived hidden nodes for V_2 and V_3 as in Figure 3(b); through the hidden nodes the cluster structures of V_2 change the similarity between v_{11} and v_{12} to 1 (highest level), since the reduced link features for both v_{11} and v_{12} are $[1, 1]$, which is a more reasonable result, since in a sparse k-partite graph we expect that two nodes are similar when they are connected to *similar* nodes even though they are not connected to the *same* nodes. If we continue this example, next, v_{11} and v_{12} are connected with the same hidden nodes in S_1 (not shown in the figure); then after the hidden nodes for V_1 are derived, the cluster structures of V_2 and V_3 may be affected in return. In fact, this is the idea of the iterative algorithm to construct an RSN for a k-partite graph, which we discuss in the next section.

4. ALGORITHM DERIVATION

In this section, we derive an iterative algorithm to find the RSN (local optima) for a k-partite graph. It can be shown that the RSN problem is NP-hard (the proof is omitted here); hence it is not realistic to expect an efficient algorithm to find the global optima.

First we reformulate the RSN problem based on the matrix representation of a k-partite graph. Given a k-partite $G = (V_1, \ldots, V_m, E)$, the weights of edges between V_i and V_j can be represented as a matrix $A^{(ij)} \in \mathbb{R}^{n_i \times n_j}$, where $n_i = |V_i|$, $n_j = |V_j|$, and $A^{(ij)}_{hl}$ denotes the weight of the edge (v_{ih}, v_{jl}), i.e., $e(v_{ih}, v_{jl})$. Similarly in an RSN $G^s = (V_1, \ldots, V_m, S_1, \ldots, S_m, E^s)$, $B^{(ij)} \in \mathbb{R}^{k_i \times k_j}$ denotes the weights of edges between S_i and S_j, i.e., $B^{(ij)}_{pq}$ denotes $e^s(s_{ip}, s_{jq})$; $C^{(i)} \in \{0, 1\}^{n_i \times k_i}$ denotes the weights of edges between V_i and S_i, i.e., $C^{(i)}$ is an indicator matrix such that if $e^s(v_{ih}, s_{ip}) = 1$, then $C^{(i)}_{hp} = 1$. Hence, we represent a k-partite as a set of matrices. Note that under the RSN model, we do not use one graph affinity matrix to represent the whole graph as in the graph partitioning approaches, which may cause very expensive computation on a huge matrix.

Based on the above matrix representation, the distance between two graphs in Eq. (1) can be formulated as the distances between a set of matrices and a set of matrix products. For example, for the two graphs shown in Figure 1, $\mathfrak{D}(G, G^s) = D(A^{(12)}, C^{(1)}B^{(12)}(C^{(2)})^T)$; for the two graphs shown in Figure 2, $\mathfrak{D}(G, G^s) = D(A^{(12)}, C^{(1)}B^{(12)}(C^{(2)})^T) + D(A^{(13)}, C^{(1)}B^{(13)}(C^{(3)})^T)$. Hence, finding the RSN defined in Definition 1 is equivalent to the following optimization problem of matrix approximation (for convenience, we assume that there exists $A^{(ij)}$ for $1 \le i < j \le m$, i.e., every pair of V_i and V_j are related to each other in G).

Definition 2. Given a distance function D , a set of matrices $\{A^{(ij)} \in \mathbb{R}^{n_i \times n_j}\}_{1 \le i < j \le m}$ representing a k-partite graph G, and m positive integers, $k_1, \ldots, k_m$, the RSN G^s represented by $\{C^{(i)} \in \{0, 1\}^{n_i \times k_i}\}_{1 \le i \le m}$ and $\{B^{(ij)} \in \mathbb{R}^{k_i \times k_j}\}_{1 \le i < j \le m}$ is given by the minimization of

$$L = \sum_{1 \le i < j \le m} D(A^{(ij)}, C^{(i)}B^{(ij)}(C^{(j)})^T), \qquad (2)$$

subject to $\sum_{k=1}^{k_i} C^{(i)}_{hk} = 1$ for $1 \le h \le n_i$.

In the above definition, the constraint on $C^{(i)}$ is to restrict $C^{(i)}$ to be an *indicator matrix*, in which each row is an indicator vector. In the definition, the distance between two matrices $D(X, Y)$ denotes the sum of the distances of each pair of elements, i.e., $D(X, Y) = \sum_{h,l} D(X_{hl}, Y_{hl})$.

For the optimization problem in Definition 1 or Definition 2, there are many choices of distance functions, which imply the different assumptions about the distribution of the weights of the edges in the given k-partite graph. For example, by using Euclidean distance function, we implicitly assume the normal distribution for the weights of the edges. Presumably for a specific distance function used in Definition 2, we need to derive a specific algorithm. However, a large number of useful distance functions, such as Euclidean distance, generalized I-divergence, and KL divergence, can be generalized as the Bregman divergences [25, 4]. Based on the properties of Bregman divergences, we derive a general algorithm to minimize the objective function in Eq.(2) under all the Bregman divergences. Table 1 shows a list of Bregman divergences and their corresponding Bregman convex functions. Note that Bregman divergences are non-negative. The definition of a Bregman divergence is given as follows.

Definition 3. Given a strictly convex function, $\phi : S \mapsto \mathbb{R}$, defined on a convex set $S \subseteq \mathbb{R}^d$ and differentiable on the interior of S, $\text{int}(S)$, the Bregman divergence $D_\phi : S \times \text{int}(S) \mapsto [0, \infty)$ is defined as

$$D_\phi(x, y) = \phi(x) - \phi(y) - (x - y)^T \nabla\phi(y), \qquad (3)$$

where $\nabla\phi$ is the gradient of ϕ.

We prove the following theorem which is the basis of our algorithm.

THEOREM 1. *Assume that D in Definition 2 is a Bregman Divergence D_ϕ. If $\{C^{(i)}\}_{1 \le i \le m}$ and $\{B^{(ij)}\}_{1 \le i < j \le m}$ are the optimal solution to the minimization in Definition 2, then*

$$(C^{(i)})^T(C^{(i)}B^{(ij)}(C^{(j)})^T - A^{(ij)})C^{(j)} = 0 \qquad (4)$$

for $1 \le i < j \le m$.

PROOF. For convenience we use Y to denote $C^{(i)}B^{(ij)}(C^{(j)})^T$, $\zeta(x)$ to denote $\nabla\phi(x)$, $\xi(x)$ to denote $\nabla^2\phi(x)$.

Name	$D_\phi(x,y)$	$\phi(x)$	Domain
Euclidean distance	$\|\|\mathbf{x}-\mathbf{y}\|\|^2$	$\|\|\mathbf{x}\|\|^2$	$\mathbb{R}^d$
Generalized I-divergence	$\sum_{i=1}^d x_i \log(\frac{x_i}{y_i}) - \sum_{i=1}^d (x_i - y_i)$	$\sum_{i=1}^d x_i \log(x_i)$	$\mathbb{R}_+^d$
Logistic loss	$x\log(\frac{x}{y}) + (1-x)\log(\frac{1-x}{1-y})$	$x\log(x) + (1-x)\log(1-x)$	$\{0,1\}$
Itakura-Saito distance	$\frac{x}{y} - \log xy - 1$	$-\log x$	$(0,\infty)$
Hinge loss	$\max\{0, -2\text{sign}(-y)x\}$	$\|x\|$	$\mathbb{R}\setminus\{0\}$
KL-divergence	$\sum_{i=1}^d x_i \log(\frac{x_i}{y_i})$	$\sum_{i=1}^d x_i \log(x_i)$	d-Simplex
Mahalanobis distance	$(\mathbf{x}-\mathbf{y})^T\mathbf{A}(\mathbf{x}-\mathbf{y})$	$\mathbf{x}^T\mathbf{A}\mathbf{x}$	$\mathbb{R}^d$

Table 1: A list of Bregman divergences and the corresponding convex functions.

We compute the gradient $\nabla_{B^{(ij)}} L$, where $1 \le i < j \le m$ and L denotes the objective function in Eq.(2). Using the fact that $\partial Y_{hl}/\partial B_{pq}^{(ij)} = C_{hp}^{(i)} C_{lq}^{(j)}$, we see that $\partial L/\partial B_{pq}^{(ij)}$ is given by

$$\begin{aligned}
&\frac{\partial}{\partial B_{pq}^{(ij)}}\{\sum_{h,l} \phi(A_{hl}^{(ij)}) - \phi(Y_{hl}) - (A_{hl}^{(ij)} - Y_{hl})\zeta(Y_{hl})\} \\
= &\sum_{h,l} -\zeta(Y_{hl}) C_{hp}^{(i)} C_{lq}^{(j)} - A_{hl}^{(ij)} \xi(Y_{hl}) C_{hp}^{(i)} C_{lq}^{(j)} + \\
&+ C_{hp}^{(i)} C_{lq}^{(j)} \zeta(Y_{hl}) + Y_{hl}\xi(Y_{hl}) C_{hp}^{(i)} C_{lq}^{(j)} \\
= &\sum_{h,l} \xi(Y_{hl})(Y_{hl} C_{hp}^{(i)} C_{lq}^{(j)} - A_{hl}^{(ij)} C_{hp}^{(i)} C_{lq}^{(j)}) \qquad (5) \\
= &[(C^{(i)})^T(\xi(Y) \odot (Y - A^{(ij)}))C^{(j)}]_{pq} \qquad (6)
\end{aligned}$$

where $\odot$ denotes the Hadamard product or entrywise product of two matrices. By Eq.(6), we have

$$\nabla_{B^{(ij)}} L = (C^{(i)})^T(\xi(Y) \odot (Y - A^{(ij)}))C^{(j)} \qquad (7)$$

According to the KKT conditions, an optimal solution to Definition 2 satisfies $\nabla_{B^{(ij)}} L = 0$, which leads to

$$(C^{(i)})^T(\xi(Y) \odot (Y - A^{(ij)}))C^{(j)} = 0 \qquad (8)$$

According to Definition 3, ϕ is strictly convex, hence, $[\xi(Y)]_{pq} > 0$ for $1 \le p \le k_i$ and $1 \le q \le k_j$. Therefore, $\xi(Y)$ can be canceled from Eq.(8) to obtain

$$(C^{(i)})^T(Y - A^{(ij)})C^{(j)} = 0 \qquad (9)$$

This completes the proof of the theorem. □

The most interesting observation about Theorem 1 is that Eq.(4) does not involve the distance function D_ϕ.

We propose an iterative algorithm to find a local optimal RSN represented by $\{C^{(i)}\}_{1\le i\le m}$ and $\{B^{(ij)}\}_{1\le i<j\le m}$ for a given k-partite graph. At each iterative step, we update one of $\{C^{(i)}\}_{1\le i\le m}$ or one of $\{B^{(ij)}\}_{1\le i<j\le m}$ by fixing all the others.

Since $C^{(i)}$ is an indicator matrix, we adopt the reassignment procedure such as in the k-means algorithm to update $C^{(i)}$. To determine which element of the hth row of $C^{(i)}$ is equal to 1, for $l = 1, \ldots, k_i$, we let $C_{hl}^{(i)} = 1$ and compute the objective function L in Eq.(2) for each l, which is denoted as L_l, then

$$C_{hl^*}^{(i)} = 1 \text{ for } l^* = \arg\min_l L_l \qquad (10)$$

The updating rule in Eq.(10) is equivalent to updating the edges between V_i and S_i in G^s by connecting v_{ih} to each hidden nodes in S_i to find which hidden node gives the smallest values for $D_\phi(G, G^s)$, i.e.,

$$e^s(v_{ih}, s_{il^*}) = 1 \text{ for } l^* = \arg\min_l D_\phi(G, G_l^s). \qquad (11)$$

Algorithm 1 Relation Summary Network with Bregman Divergences

Input: A k-partite graph $G = (V_1, \ldots, V_m, E)$, a Bregman divergence function D_ϕ, and m positive integers, $k_1, \ldots, k_m$.
Output: An RSN $G^s = (V_1, \ldots, V_m, S_1, \ldots, S_m, E^s)$.
Method:

```
Initialize G^s.
repeat
    for i = 1 to m do
        Update the edges between V_i and S_i according to Eq.(11).
    end for
    for each pair of S_i ~ S_j where 1 ≤ i < j ≤ m do
        Update the edges between S_i and S_j according to Eq.(13).
    end for
until convergence
```

where G_l^s denotes the RSN with s_{il} connecting to v_{ih}. Note that the computation for this updating involves only edges between v_{ih} and the related nodes, not all the edges.

Based on Eq.(4) in Theorem 1, after a little algebraic manipulation, we have the following updating rule for each $B^{(ij)}$,

$$B^{(ij)} = ((C^{(i)})^T C^{(i)})^{-1}(C^{(i)})^T A^{(ij)} C^{(j)}((C^{(j)})^T C^{(j)})^{-1} \qquad (12)$$

This updating rule does not really involve computing inverse matrices, since $(C^{(i)})^T C^{(i)}$ is a special diagonal matrix such that $[(C^{(i)})^T C^{(i)}]_{pp} = \sum_{h=1}^{n_i} C_{hp}^{(i)}$, i.e., the number of instance nodes associated with the hidden node s_{ip}, and similarly for $(C^{(j)})^T C^{(j)}$. The updating rule in Eq.(12) is equivalent to updating the edges between S_i and S_j in G^s by re-computing the weight of the edge between a pair of hidden nodes $s_{ip} \in S_i$ and $s_{jq} \in S_j$ as follows,

$$e^s(s_{ip}, s_{jq}) = \frac{1}{|\mathcal{U}| * |\mathcal{Z}|} \sum_{v_{ih}\in\mathcal{U}, v_{jl}\in\mathcal{Z}} e(v_{ih}, v_{jl}), \qquad (13)$$

where $\mathcal{U} = \{v_{ih} : e^s(v_{ih}, s_{ip}) = 1\}$, i.e., the instance nodes associated with s_{ip}; $\mathcal{Z} = \{v_{jl} : e^s(v_{jl}, s_{jq}) = 1\}$, i.e., the instance nodes associated with s_{jq}, $1 \le p \le k_i, 1 \le q \le k_j, 1 \le h \le n_i$, and $1 \le l \le n_j$. This updating rule is consistent with our intuition about the edge between two hidden nodes; i.e., it is the "summary relation" for two sets of instance nodes. It is, however, a surprising observation that the updating does not involve the distance function, i.e., this simple updating rule holds for all Bregman divergences.

The algorithm, Relation Summary Network with Bregman Divergences (RSN-BD), is summarized in Algorithm 1. RSN-BD iteratively updates the cluster structures for different types of instance nodes and summary relations among

the hidden nodes. Through the hidden nodes, the cluster structures of different types of instance nodes interact with each other directly or indirectly. The interactions lead to the implicit adaptive feature reduction for each type of instance nodes which overcomes the typical high dimensionality and sparsity. RSN-BD is applicable to a wide range of problems, since it does not have restrictions on the structures of the input k-partite graph. Furthermore, the graphs from different applications may have different probabilistic distributions on their edges; it is easy for RSN-BD to adapt to this situation by simply using different Bregman divergences, since Bregman divergences correspond to a large family of exponential distributions including most common distributions, such as Normal, Multinomial and Poisson distributions [7].

Note that to avoid clutter, we do not consider weighting different types of edges during the derivation. Nevertheless, it is easy to extend the proposed model and algorithm to the weighted versions.

If we assume that the number of pairs of $V_i \sim V_j$ is $\Theta(m)$ which is typical in real applications, and let $n = \Theta(n_i)$ and $k = \Theta(k_i)$, the computational complexity of RSN-BD can be shown to be $O(tmn^2k)$ for t iterations. If we apply the k-means algorithm to each type of nodes individually by transforming the relations into features for each type of nodes, the total computational complexity is also $O(tmn^2k)$. Hence, RSN-BD is as efficient as k-means. If the edges in the graph are very sparse, the computational complexity of RSN-BD can be reduced to $O(tmrk)$ where we assume that the number of edges between each pair of V_i and V_j is $\Theta(r)$.

Eq.(4) in Theorem 1 is an necessary condition for an optimal solution, but not sufficient for the correctness of the RSN-BD algorithm. The following theorems guarantee the convergence of RSN-BD.

LEMMA 1. *Given a Bregman divergence $D_\phi : S \times int(S) \mapsto [0, \infty)$, $A \in \mathbb{R}^{n_1 \times n_2}$ and two indicator matrices, $C^{(1)} \in \{0,1\}^{n_1 \times k_1}$ and $C^{(2)} \in \{0,1\}^{n_2 \times k_2}$, let*

$$B^* = ((C^{(1)})^T C^{(1)})^{-1} (C^{(1)})^T A C^{(2)} ((C^{(2)})^T C^{(2)})^{-1} \quad (14)$$

then for any $B \in \mathbb{R}^{k_1 \times k_2}$,

$$D_\phi(A, C^{(1)} B (C^{(2)})^T) - D_\phi(A, C^{(1)} B^* (C^{(2)})^T) \geq 0. \quad (15)$$

PROOF. For convenience we use Y to denote $C^{(1)} B (C^{(2)})^T$, Y^* to denote $C^{(1)} B^* (C^{(2)})^T$, $\zeta(x)$ to denote $\nabla\phi(x)$. Let $\mathfrak{J}$ denote the lefthand side of Eq.(15).

$$\begin{aligned}
\mathfrak{J} &= \sum_{h,l} D_\phi(A_{hl}, Y_{hl}) - \sum_{h,l} D_\phi(A_{hl}, Y^*_{hl}) \\
&= \sum_{h,l} \{\phi(Y^*_{hl}) - \phi(Y_{hl}) - (A_{hl} - Y_{hl})\zeta(Y_{hl}) \\
&\quad + (A_{hl} - Y^*_{hl})\zeta(Y^*_{hl})\} \\
&= \sum_{h,l} \{\phi(Y^*_{hl}) - \phi(Y_{hl}) - [A \odot \zeta(Y)]_{hl} + [Y \odot \zeta(Y)]_{hl} \\
&\quad + [A \odot \zeta(Y^*)]_{hl} - [Y^* \odot \zeta(Y^*)]_{hl}\} \\
&= \sum_{h,l} \{\phi(Y^*_{hl}) - \phi(Y_{hl}) - [Y^* \odot \zeta(Y)]_{hl} + [Y \odot \zeta(Y)]_{hl}\} \\
&= D_\phi(Y^*, Y) \\
&\geq 0
\end{aligned}$$

During the above deduction, the second and fifth equalities follow the definition of the Bregman divergences; the fourth equality follows the fact that $\sum_{h,l}[A \odot \zeta(Y^*)]_{hl} = \sum_{h,l}[Y^* \odot \zeta(Y^*)]_{hl}$ and $\sum_{h,l}[A \odot \zeta(Y)]_{hl} = \sum_{h,l}[Y^* \odot \zeta(Y)]_{hl}$ resulting from the special structure of the indicator matrix; the last inequality follows the non-negativity of Bregman divergences. □

THEOREM 2. *The RSN-BD algorithm (Algorithm 1) monotonically decreases the objective function in Eq.(1).*

PROOF. Proving the theorem is equivalent to proving that the updating rules in Eq.(10) and Eq.(12) monotonically decrease the objective function in Eq.(2). Let $L_{(t)}$ denote the objective value after the tth iteration.

$$\begin{aligned}
L_{(t)} &= \sum_{1 \leq i < j \leq m} D_\phi(A^{(ij)}, C^{(i)}_{(t)} B^{(ij)}_{(t)} (C^{(j)}_{(t)})^T) \\
&\geq \sum_{1 \leq i < j \leq m} D_\phi(A^{(ij)}, C^{(i)}_{(t+1)} B^{(ij)}_{(t)} (C^{(j)}_{(t+1)})^T) \\
&\geq \sum_{1 \leq i < j \leq m} D_\phi(A^{(ij)}, C^{(i)}_{(t+1)} B^{(ij)}_{(t+1)} (C^{(j)}_{(t+1)})^T) \\
&= L_{(t+1)}
\end{aligned}$$

where the first inequality follows trivially the criteria used for reassignment in Eq.(10), and the second inequality follows Eq.(12) and Lemman 1. □

Base on Theorem 2 and the fact that the objective function in Eq.(2) has the lower bound 0 for a Bregman divergence, the convergence of RSN-BD is proved.

5. A UNIFIED VIEW TO CLUSTERING APPROACHES

In this section we discuss the connections between existing clustering approaches and the RSN model. By considering them as special cases or variations of the RSN model, we show that RSN provides a unified view to the existing clustering approaches.

5.1 Bipartite Spectral Graph Partitioning

Bipartite Spectral Graph Partitioning (BSGP) [8, 18] uses the spectral approach to partitioning a bi-partite graph to find cluster structures for two types of interrelated data objects, such as words and documents. The objective function of BSGP is the normalized cut on the bi-partite graph, whose affinity matrix is $\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$. After the deduction, the spectral partitioning on the bipartite graph is converted to a singular value decomposition (SVD) [8, 18].

As a graph partitioning approach, BSGP has the restriction that the clusters of different types of nodes have one-to-one associations. Under the RSN model, this restriction is equivalent to letting a hidden node connect with one and only one hidden node. Hence, the affinity matrix representing the edges between two sets of hidden nodes is restricted to a diagonal matrix. The objective function in Eq.(2) can be formulated as

$$L = ||A^{(12)} - C^{(1)} B^{(12)} (C^{(2)})^T||^2 \quad (16)$$

where $||\cdot||$ denotes Frobenius norm, i.e., the Euclidean distance function is adopted, and A may be normalized as described in [8]. Based on this objective function, if we relax $C^{(1)}$ and $C^{(2)}$ to any othornormal matrices as in [8, 18], it immediately follows the standard result of linear algebra [14] that the minimization of L in Eq.(16) with the diagonal

constraint on B is equivalent to partial SVD. Therefore, the RSN model based on Euclidean distance function provides a simple way to understand BSGP. Comparing with BSGP, RSN-BD is more flexible to exploit the cluster structures from a bi-partite graph, since it does not have one-to-one association as a constraint and is capable of adopting different distance functions.

5.2 Binary Data Clustering with Feature Reduction

In [21], a model is proposed to cluster binary data by clustering data points and features simultaneously, i.e., clustering with feature reduction. If we consider data points and features as two different types of nodes in a bi-partite graph and the binary elements of the data matrix denote whether there exists a link between a pair of nodes, then this model is equivalent to the RSN model on a bi-partite graph with unit weight edges. The objective function of this model is given in [21] as

$$O(A, X, B) = ||W - AXB^T||^2, \quad (17)$$

where W denotes the data matrix, A and B denote cluster memberships for data points and features, respectively, and X represents the associations between the data clusters and the feature clusters. We can see that this objective function is exactly the same as the objective function in Eq.(2) on bi-partite graph with Euclidean distance.

The immediate benefit of establishing the connection between the model proposed in [21] and the RSN model is the new solution to binary data clustering with feature reduction. In [21], the model is based on Euclidean distance. Euclidean distance function has very wide applicability, since it implies the normal distribution and most data with a large sample size tend to have a normal distribution. However, since Bernoulli distribution is a more intuitive choice for the binary data, RSN-BD directly provides a new algorithm for clustering binary data with feature reduction by using logistic distance function (see Table 1), which corresponds to Bernoulli distribution.

5.3 Information-Theoretic Co-clustering

[9] proposes a novel theoretic formulation to view the contingency table as an empirical joint probability distribution of two discrete random variables and developes the co-clustering algorithm, Information-Theoretic Co-Clustering (ITCC), to maximize the mutual information between the clustered random variables subject to the constraints on the number of row and column clusters. Let X and Y be discrete random variables that take values in the sets $\{x_1, \ldots, x_{n_1}\}$ and $\{y_1, \ldots, y_{n_2}\}$, respectively, and $\hat{X}$ and $\hat{Y}$ be the cluster random variables that take values in the sets $\{\hat{x}_1, \ldots, \hat{x}_{k_1}\}$ and $\{\hat{y}_1, \ldots, \hat{y}_{k_2}\}$, respectively; then the objective function of ITCC is the loss in mutual information, $I(X;Y) - I(\hat{X}, \hat{Y})$.

The joint distribution of X and Y can be formulated as a bi-partite graph by assigning the probability $p(x_h, y_l)$ to the weight of the edge between $v_{1h} \in V_1$ and $v_{2l} \in V_2$. If we modify the Condition 1 in Definition 1 such that an instance node v_{ih} is connected to one and only one hidden node s_{ip} with weight $\frac{1}{\#s_{ip}}$ where $\#s_{ip}$ is the number of the instance nodes connected to s_{ip}, then in the RSN of aforementioned bi-partite graph, $e^s(v_{1h}, s_{1p})$ and $e^s(v_{2l}, s_{2q})$ can be considered as $p(x_h|\hat{x}_p)$ and $p(x_l|\hat{x}_q)$, respectively, $e^s(s_{1p}, s_{2q})$ can be considered as $p(\hat{x}_p, \hat{x}_q)$. Based on this formulation, it is easy to verify that the objective function of RSN with KL-divergence is equivalent to $I(X;Y) - I(\hat{X}, \hat{Y})$. This connection between the ITCC and a variation of RSN model implies that the ITCC algorithm may be extended to more general cases of more than two random variables and with other loss functions.

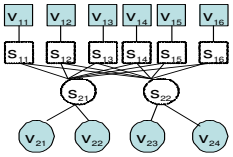

Figure 4: An RSN equivalent to k-means

5.4 K-means Clustering

Due to its simplicity, efficiency, and broad applicability, k-means algorithm has become one of the most popular clustering algorithms. Figure 4 explains the relation between the RSN model and k-means. If we consider data points and features as two different types of nodes, V_2 and V_1, in a bi-partite graph, and restrict feature nodes to have one-to-one associations of their hidden nodes with unit weight, then the objective function in Eq.(2) is given as $L = ||A^{(12)} - C^{(1)}B^{(12)}(C^{(2)})^T||^2$ where $C^{(2)}$ is restricted to an identity matrix. Hence, the objective function is reduced to $L = ||A^{(12)} - C^{(1)}B^{(12)}||^2$, which is exactly the matrix representation for the objective function of the k-means algorithm [31]. From Figure 4, we also see that since the number of feature nodes is equal to the number of their hidden nodes, k-means does not do feature reduction. Finally, we may consider RSN-BD as a generalization of k-means on k-partite graphs with various Bregman divergences and expect that it inherits the simplicity and efficiency of k-means and has much broader applicability.

There are more clustering approaches in the literature that may be considered as the special cases or variations of the RSN model. For example, the subspace clustering [1], which clusters the data points in a high dimensional space around a different subset of the dimensions, can be considered as an extension of Figure 4 such that s_{21} or s_{22} only connects to a subset of S_1. Spectral relational clustering [22] can be considered as using the spectral approach to solve the RSN model under Euclidean distance.

By examining the connections between existing clustering approaches and the RSN model, we conclude that the RSN model provides a unified view to the existing clustering approaches. Moreover, the idea of RSN is more general than the proposed model based on Definition 1 and Eq.(1). For example, if we change the definition of distance between graphs in Eq.(1), we may find totally different ways to mine hidden structures from a k-partite graph, and as a result, we may obtain new variations for existing clustering approaches.

6. EXPERIMENTAL RESULTS

This section provides empirical evidence to show the effectiveness of the RSN model and algorithm. In particular, we apply RSN-BD to two basic types of k-partite graphs, the bi-partite graph and the sandwich structure tri-partite graph (such as Figure 2(a)), which arise frequently in various applications. Note that the application of RSN-BD is not limited to these two types of graphs and it is applicable to various k-partite graphs. Four types of RSN-BD are evaluated in the

Data set	S	Distribution
BP-b1	$\begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix}$	Bernoulli
BP-b2	$\begin{bmatrix} 0.4 & 0.7 \\ 0.5 & 0.6 \end{bmatrix}$	Bernoulli
BP-p	$\begin{bmatrix} 0.5 & 0.6 \\ 0.6 & 0.8 \end{bmatrix}$	Poisson
BP-e	$\begin{bmatrix} 0.4 & 0.5 \\ 0.5 & 0.7 \end{bmatrix}$	Exponential

Table 2: Parameters and distributions for synthetic bi-partite graphs

experiments: RSN with Euclidean Distance (RSN-ED) assumes the normal distribution of the data; RSN with Logistic Loss (RSN-LL) assumes the Bernoulli distribution of the data; RSN with Generalized I-divergence (RSN-GI) assumes the Poisson distribution of the data; RSN with Itakura-Saito distance (RSN-IS) assumes the exponential distribution of the data. Two graph partitioning approaches, BSGP [8] and Consistent Bipartite Graph Co-partitioning (CBGC) [13] (we thank the authors for providing the executable code of CBGC) , are used as the comparison on bi-partite graph and sandwich tri-partite graph, respectively. Four traditional feature-based algorithms, which cluster a type of nodes in a k-partite graph by transforming all the links into features, are also used as comparisons. They are K-Means with Euclidean Distance (KM-ED), K-Means with Logistic Loss (KM-LL), K-Means with Generalized I-divergence (KM-GI) and K-Means with Itakura-Saito (KM-IS).

6.1 Data Sets and Parameter Setting

The data sets used in the experiments include synthetic data sets with various distributions and real data sets based on the 20-Newsgroup data [20].

The synthetic bi-partite graphs are generated such as that both V_1 and V_2 have two clusters (to be fair for BSGP, we use equal number of clusters); each cluster has 100 nodes, hence, both V_1 and V_2 have 200 nodes. The distributions and parameters (the true means of the distributions) used to generate the links in the graphs are documented in Table 2. In the table, distribution parameters for a graph is represented as a matrix S such that S_{pq} denotes the mean parameter of the distribution to generate the links between the pth cluster of V_1 and the qth cluster of V_2.

The real bi-partite graphs are constructed based on various subsets of the 20-Newsgroup data [20] which contains about $20,000$ articles from 20 newsgroups. We pre-process the data by removing stop words and selecting the top 2000 words by the mutual information. The document-word matrix is based on *tf.idf* weighting scheme and each document vector is normalized to a unit L_2 norm vector. Specific details of data sets used to construct bi-partite graphs are listed in Table 3. For example, to construct a *BP-NG3* graph, we randomly and evenly sample 200 documents from the corresponding newsgroups; then we formulate a bi-partite graph consisting of 1600 document nodes and 2000 word nodes.

The synthetic tri-partite graphs are generated similarly to the bi-partite graphs. The distributions and parameters are documented in Table 4. Let V_1 denote the central type nodes. In Table 4, $S^{(12)}$ denotes the true means of distributions for generating the links between V_1 and V_2, and similarly for $S^{(13)}$. The numbers of clusters for each type of nodes are given by dimensions of $S^{(12)}$ and $S^{(13)}$ and each

Data set	$S^{(12)}$	$S^{(13)}$	Distribution
TP-b1	$\begin{bmatrix} 0.4 & 0.7 \\ 0.5 & 0.6 \end{bmatrix}$	$\begin{bmatrix} 0.7 & 0.5 \\ 0.6 & 0.6 \end{bmatrix}$	Bernoulli
TP-b2	$\begin{bmatrix} 0.5 & 0.6 \\ 0.5 & 0.6 \\ 0.7 & 0.7 \end{bmatrix}$	$\begin{bmatrix} 0.6 & 0.6 & 0.6 \\ 0.7 & 0.7 & 0.5 \\ 0.7 & 0.7 & 0.5 \end{bmatrix}$	Bernoulli
BP-p	$\begin{bmatrix} 0.3 & 0.6 \\ 0.2 & 0.7 \end{bmatrix}$	$\begin{bmatrix} 0.4 & 0.4 \\ 0.5 & 0.3 \end{bmatrix}$	Poisson
TP-large	$\mathbb{Z}^{20\times 20}$	$\mathbb{Z}^{20\times 18}$	Poisson
TP-e	$\begin{bmatrix} 0.3 & 0.6 \\ 0.3 & 0.7 \end{bmatrix}$	$\begin{bmatrix} 0.4 & 0.7 \\ 0.5 & 0.6 \end{bmatrix}$	Exponential

Table 4: Parameters and distributions for synthetic tri-partite graphs

Data set	Taxonomy structure
TP-TM1	{rec.sport.baseball, rec.sport.hockey}, {talk.politics.guns, talk.politics.mideast, talk.politics.misc}
TP-TM2	{comp.graphics, comp.os.ms-windows.misc}, {rec.autos, rec.motorcycles}, {sci.crypt, sci.electronics}

Table 5: Taxonomy structures of two data sets for constructing tri-partite graphs

cluster has 100 nodes. In Table 4, *TP-large* is a large graph with 20 clusters of V_1, 20 clusters of V_2, and 18 clusters of V_3 (due to the space limit, the details of parameters are omitted). Each *BP-large* graph contains 5800 nodes and on an average about 3.25 million links.

The real tri-partite graphs are built based on the 20-newsgroups data for hierarchical taxonomy mining. In the field of text categorization, hierarchical taxonomy classification is widely used to obtain a better trade-off between effectiveness and efficiency than flat taxonomy classification. To take advantage of hierarchical classification, one must mine a hierarchical taxonomy from the data set. We see that words, documents, and categories formulate a sandwich structure tri-partite graph, in which documents are central type nodes. The links between documents and categories are constructed such that if a document belongs to k categories, the weights of links between this document and these k category nodes are $1/k$ (please refer [13] for details).

The true taxonomy structures for two data sets, *TP-TM1* and *TP-TM2*, are documented in Table 5. For example, *TP-TM1* data set is sampled from five categories (200 documents for each category), in which two categories belong to the high level category *res.sports* and other three categories belong to the high level category *talk.politics*.

For all the algorithms on all the graphs, we fix the number of iterations to 20 (this also holds true for BSGP and CBGC, since they use classic k-means to do postprocessing) and use the same initialization, random initialization for synthetic data and classic k-means initialization for real data. The final performance score is the average of the twenty runs. At each test run, a graph is constructed by sampling from the corresponding distributions or newsgroups of the 20-newsgroup data. Hence, the variation of a final performance score includes the variance of sampling.

For the number of clusters, we use the true number of clusters for the synthetic graphs. For real data graphs, we use the true number of clusters for documents and categories; however, we do not know the true number of word clusters. How to determine the optimal number of word clusters is beyond the scope of this paper. We simply adopt 40 for all the RSN algorithms. For BSGP and CBGC, the number of

Dataset Name	Newsgroups Included	# Documents per Group	Total # Documents
BP-NG1	rec.sport.baseball, rec.sport.hockey	200	400
BP-NG2	comp.os.ms-windows.misc, comp.windows.x, rec.motorcycles, sci.crypt, sci.space	200	1000
BP-NG3	comp.os.ms-windows.misc, comp.windows.x, misc.forsale, rec.motorcycles,rec.motorcycles,sci.crypt, sci.space, talk.politics.mideast, talk.religion.misc	200	1600

Table 3: Subsets of Newsgroup Data for constructing bi-partite graphs.

Algorithm	*BP-b1*	*BP-b2*	*BP-p*	*BP-e*	*BP-NG1*	*BP-NG2*	*BP-NG3*
RSN-ED	1 ± 0	0.618 ± 0.079	0.549 ± 0.057	0.821 ± 0.064	0.402 ± 0.239	0.599 ± 0.055	0.573 ± 0.037
KM-ED	1 ± 0	0.069 ± 0.089	0.042 ± 0.049	0.632 ± 0.095	0.375 ± 0.236	0.616 ± 0.070	0.601 ± 0.042
RSN-LL	1 ± 0	$\mathbf{0.620 \pm 0.069}$	0.519 ± 0.075	0.819 ± 0.062	$\mathbf{0.638 \pm 0.164}$	$\mathbf{0.747 \pm 0.068}$	$\mathbf{0.698 \pm 0.037}$
KM-LL	1 ± 0	0.060 ± 0.084	0.224 ± 0.099	0.567 ± 0.079	0.443 ± 0.229	0.655 ± 0.070	0.641 ± 0.038
RSN-GI	1 ± 0	0.604 ± 0.062	$\mathbf{0.562 \pm 0.060}$	0.849 ± 0.058	0.619 ± 0.180	0.746 ± 0.066	0.697 ± 0.038
KM-GI	1 ± 0	0.053 ± 0.056	0.025 ± 0.023	0.656 ± 0.188	0.444 ± 0.229	0.655 ± 0.069	0.641 ± 0.038
RSN-IS	1 ± 0	0.549 ± 0.074	0.553 ± 0.064	$\mathbf{0.857 \pm 0.063}$	0.411 ± 0.207	0.414 ± 0.084	0.335 ± 0.056
KM-IS	1 ± 0	0.050 ± 0.059	0.025 ± 0.037	0.635 ± 0.207	0.383 ± 0.242	0.618 ± 0.063	0.596 ± 0.043
BSGP	1 ± 0	0.379 ± 0.079	0.005 ± 0.007	0.004 ± 0.089	0.430 ± 0.252	0.638 ± 0.033	0.501 ± 0.047

Table 6: NMI scores of the algorithms on bi-partite graphs

word clusters must equal the number of document clusters. By the authors' suggestion, the parameter setting for CBGC is $\beta = 0.5$, $\theta_1 = 1$ and $\theta_2 = 1$ [13].

The performance comparison is based on the quality of the clusters of one type of nodes in each graph. In synthetic bi-partite graphs, it is based on V_1 whose clusters correspond to the rows of S in Table 2; in synthetic tri-partite graphs, it is based on the central type nodes V_1; in bi-partite graphs of documents and words, it is based on documents; in tri-partite graphs for taxonomy mining, it is based on categories whose clusters provide the taxonomy structures. For performance measure, we elect to use the Normalized Mutual Information (NMI) [27], which is a standard way to measure the cluster quality.

6.2 Results and Discussion

Table 6 shows the NMI scores of the nine algorithms on the bi-partite graphs. For the *BP-b1* graph, all the algorithms provide perfect NMI score, since the graphs are generated with very clear structures, which can be seen from the parameter matrix in Table 2. For other synthetic bi-partite graphs, the cluster structures are subtle, especially for the nodes V_1, whose cluster structures are our objective. For these graphs, the RSN algorithms perform much better than k-means algorithms, especially for the *BP-b2* and *BP-p* graph, in which the distributions for clusters of V_1 are very close to each other and the links are relatively sparse. This comparison implies that benefiting from the interactions among the cluster structures of different types of nodes, the RSN algorithms are able to identify very subtle cluster structures even when the traditional clustering approaches totally fail. Compared with the RSN algorithms, BSGP performs poorly for all the synthetic bi-partite graphs except *BP-b1*. The possible explanation is that it assumes one-to-one associations between clusters of different types of nodes, which does not hold true for the synthetic bi-partite graphs except *BP-b1*. We also observe that the RSN algorithm with the distance function matching the distribution to generate the graph provides the best NMI score for that graph.

For the real bi-partite graphs consisting of document and word nodes, RSN-LL always provides the best NMI score. For the difficult *BP-NG1* graph based on two "close" newsgroups, RSN-LL shows about 44% improvement in comparison with KM-LL, which is, along with KM-GI, the best among the non-RSN algorithms. Note that since the document vector is L_2-normalized, the KM-ED is actually based on von Mises-Fisher distribution [24], which proved efficient for document clustering [2]. We also observe that for these graphs, in general the algorithms based on logistic loss provide better performance. The possible reason is that logistic loss corresponds to Bernoulli distribution which provides a good approximation to the distribution of the data consisting of a large mount of zeros, such as the sparse links between documents and words. In the meantime, it is also reasonable to assume the Poisson distribution for the frequencies such as the frequency in that a word appears in a document. That is why RSN-GI also shows the performance very close to RSN-LL. The above comparison verifies the assumption that under an appropriate distribution assumption, through the hidden nodes the RSN algorithms perform implicit adaptive feature reduction to overcome the typical high dimensionality and sparseness.

Table 7 shows the NMI scores of the nine algorithms on the tri-partite graphs. As similarly in the synthetic bi-partite graphs, the RSN algorithms perform much better than the k-means algorithms. Except for RSN-ED on the *TP-p* graph, the RSN algorithms perform significantly better than CBGC. The NMI scores of CBGC for some graphs are not available because the CBGC code provided by the authors only works for the case of two clusters and small size graphs. For the large dense *TP-large* graph, the RSN algorithms perform consistently better than the KM algorithms, and this demonstrates the good scalability of the RSN algorithms; the RSN-ED performs best on *TP-large*, and this demonstrates the advantage of the normal distribution for the very large sample size of dense links.

For the real tri-partite graphs for taxonomy mining, the k-means algorithms perform poorly since they cluster categories only based on links between categories and documents. From Table 7, we observe that both RSN-ED and RSN-IS provide the best NMI score for *TP-TM1*. To have an intuition about this score, we check the details of the 20 test runs, which show that in 16 out of the 20 runs the algorithms provide the perfect taxonomy structures and in the other 4 runs one category is clustered incorrectly. We believe that if we assign different weights to different types of links, the RSN algorithms could perform more efficiently

Algorithm	*TP-b1*	*TP-b2*	*TP-p*	*TP-large*	*TP-e*	*TP-TM1*	*TP-TM2*
RSN-ED	0.835 ± 0.061	0.847 ± 0.087	0.573 ± 0.073	$\mathbf{0.715 \pm 0.0433}$	0.612 ± 0.067	$\mathbf{0.887 \pm 0.233}$	0.623 ± 0.178
KM-ED	0.196 ± 0.217	0.258 ± 0.147	0.012 ± 0.016	0.165 ± 0.153	0.017 ± 0.023	0.257 ± 0.211	0.439 ± 0.117
RSN-LL	$\mathbf{0.848 \pm 0.061}$	$\mathbf{0.860 \pm 0.036}$	0.622 ± 0.079	0.335 ± 0.145	0.606 ± 0.063	0.858 ± 0.252	0.645 ± 0.175
KM-LL	0.219 ± 0.214	0.255 ± 0.137	0.025 ± 0.036	0.174 ± 0.153	0.016 ± 0.021	0.218 ± 0.246	0.456 ± 0.127
RSN-GI	0.829 ± 0.062	0.854 ± 0.043	$\mathbf{0.656 \pm 0.066}$	0.658 ± 0.083	0.662 ± 0.071	0.858 ± 0.252	0.637 ± 0.174
KM-GI	0.194 ± 0.197	0.289 ± 0.127	0.014 ± 0.026	0.174 ± 0.153	0.019 ± 0.025	0.245 ± 0.246	0.482 ± 0.173
RSN-IS	0.801 ± 0.064	0.811 ± 0.086	0.616 ± 0.084	0.512 ± 0.183	$\mathbf{0.677 \pm 0.063}$	$\mathbf{0.887 \pm 0.233}$	$\mathbf{0.681 \pm 0.150}$
KM-IS	0.152 ± 0.170	0.310 ± 0.110	0.019 ± 0.030	0.250 ± 0.116	0.012 ± 0.015	0.223 ± 0.243	0.469 ± 0.152
CBGC	0.744 ± 0.076	—	0.575 ± 0.068	—	0.575 ± 0.069	—	—

Table 7: NMI scores of the algorithms on tri-partite graphs

on mining the taxonomy structures. However, this is beyond the scope of this paper.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a general model RSN to find the hidden structures (the local cluster structures and the global community structures) from a k-partite graph. The model provides a principal framework for unsupervised learning on k-partite graphs of various structures. Under this model, we derive a novel algorithm to find the hidden structures from a k-partite graph under a broad range of distortion measures. By iteratively updating the cluster structures for each type of nodes, the algorithm takes advantage of the interactions among the cluster structures of different types of nodes and performs implicit adaptive feature reduction for each type of nodes. Experiments on both synthetic and real data sets demonstrate the promise and effectiveness of the proposed model and algorithm. We also establish the connections between existing clustering approaches and the proposed model to provide a unified view to the existing clustering approaches in the literature. There are a number of interesting potential directions for future research on the RSN model and algorithms, such as extending RSN model to other cases of unsupervised learning on k-partite graphs and applying the RSN algorithms to a wide range of problems involving k-partite graphs.

8. ACKNOWLEDGMENTS

This work is supported in part by NSF (IIS-0535162), AFRL Information Institute (FA8750-04-1-0234, FA8750-05-2-0284), AFOSR (FA9550-06-1-0327), and a summer research internship at Yahoo! Research.

9. REFERENCES

[1] C. C. Aggarwal, C. M. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *SIGMOD Conference*, pages 61–72, 1999.

[2] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Generative model-based clustering of directional data. In *KDD'03*, 2003.

[3] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *KDD*, pages 509–514, 2004.

[4] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. In *SDM*, 2004.

[5] T. N. Bui and C. Jones. A heuristic for reducing fill-in in sparse matrix factorization. In *PPSC*, pages 445–452, 1993.

[6] P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. In *DAC '93*, pages 749–754, 1993.

[7] M. Collins, S. Dasgupta, and R. Reina. A generalizaionof principal component analysis to the exponential family. In *NIPS'01*, 2001.

[8] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, pages 269–274, 2001.

[9] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *KDD'03*, pages 89–98, 2003.

[10] C. Ding, X. He, and H. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SDM'05*, 2005.

[11] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of ICDM 2001*, pages 107–114, 2001.

[12] R. El-Yaniv and O. Souroujon. Iterative double clustering for unsupervised and semi-supervised learning. In *ECML*, pages 121–132, 2001.

[13] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In *KDD '05*, pages 41–50, 2005.

[14] G.Golub and C. Loan. *Matrix Computations*. Johns Hopkins University Press, 1989.

[15] B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. In *Supercomputing '95*.

[16] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999.

[17] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *IJCAI'99*, Stockholm, 1999.

[18] M. X. H.Zha, C.Ding and H.Simon. Bi-partite graph partitioning and data clustering. In *ACM CIKM'01*, 2001.

[19] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.

[20] K. Lang. News weeder: Learning to filter netnews. In *ICML*, 1995.

[21] T. Li. A general model for clustering binary data. In *KDD'05*, 2005.

[22] B. Long, Z. M. Zhang, X. Wu, and P. S. Yu. Spectral clustering for multi-type relational data. In *ICML'06*, 2006.

[23] B. Long, Z. M. Zhang, and P. S. Yu. Co-clustering by block value decomposition. In *KDD'05*, 2005.

[24] K. V. Mardian. Statistics of directional data. *J. Royal Statistical Society. Series B*, 37(3):349–393, 1975.

[25] J. S.D.Pietra, V.D.Pietera. Duality and auxiliary functions for bregman distances. Technical Report CMU-CS-01-109, Carnegie Mellon University, 2001.

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

[27] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining partitionings. In *AAAI 2002*, pages 93–98, 2002.

[28] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[29] J. Wang, H. Zeng, Z. Chen, H. Lu, L. Tao, and W.-Y. Ma. Recom: reinforcement clustering of multi-type interrelated data objects. In *SIGIR '03*, pages 274–281, 2003.

[30] H.-J. Zeng, Z. Chen, and W.-Y. Ma. A unified framework for clustering heterogeneous web objects. In *WISE '02*, pages 161–172, 2002.

[31] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. *Advances in Neural Information Processing Systems*, 14, 2002.