5

A Semi-Supervised Clustering Method Based on Graph Contraction and Spectral Graph Theory

Tetsuya Yoshida
Graduate School of Information Science and Technology, Hokkaido University
Japan

1. Introduction

Semi-supervised learning is a machine learning framework where learning from data is conducted by utilizing a small amount of labeled data as well as a large amount of unlabeled data (Chapelle et al., 2006). It has been intensively studied in data mining and machine learning communities recently. One of the reasons is that, it can alleviate the time-consuming effort to collect "ground truth" labeled data while sustaining relatively high performance by exploiting a large amount of unlabeled data. (Blum & Mitchell, 1998) showed the PAC learnability of semi-supervised learning, especially in classification problem.

On the other hand, data clustering, also called unsupervised learning, is a method of creating groups of objects, or clusters, in such a way that objects in one cluster are very similar and objects in different clusters are quite distinct. Clustering is one of the most frequently performed analysis (Jain et al., 1999). For example, in web activity logs, clusters can indicate navigation patterns of different user groups. Another direct application could be clustering of gene expression data so that genes within a same group evinces similar behavior.

Although labeled data is not required in clustering, sometimes constraints on data assignment might be available as domain knowledge about the data to be clustered. In such a situation, it is desirable to utilize the available constraints as semi-supervised information and to improve the performance of clustering (Basu et al., 2008). By regarding constraints on data assignment as supervised information, various research efforts have been conducted on semi-supervised clustering (Basu et al., 2004; 2008; Li et al., 2008; Tang et al., 2007; Xing et al., 2003). Although various forms of constraints can be considered, based on the previous work (Li et al., 2008; Tang et al., 2007; Wagstaff et al., 2001; Xing et al., 2003), we deal with the following two kinds of pairwise constraints in this paper: must-link constraints and cannot-link constraints. In this chapter, the former is also called as must-links, and the latter as cannot-links.

When similarities among data instances are specified, by connecting each pair of instances with an edge with the corresponding similarity, the entire data instances can be represented as an edge-weighted graph. In this chapter we present our semi-supervised clustering method based on graph contraction in general graph theory and graph Laplacian in spectral graph theory. Graph representation enables to deal with two kinds of pairwise constraints as well as pairwise similarities over a unified representation. Then, the graph is modified by contraction in graph theory (Diestel, 2006) and graph Laplacian in spectral graph theory (Chung, 1997; von Luxburg, 2007) to reflect the pairwise constraints.

Representing the relations (both pairwise constraints and similarities) among instances as an edge-weighted graph and modifying the graph structure based on the specified constraints enable to enhancing semi-supervised clustering. In our approach, the entire data instances are projected onto a subspace which is constructed with respect to the modified graph structure, and clustering is conducted over the projected data representation of instances. Although our approach utilizes graph Laplacian as in (Belkin & Niyogi, 2002), our approach differs from previous ones since pairwise constraints for semi-supervised clustering are also utilized in our approach for constructing the projected data representation (Yoshida, 2010; Yoshida & Okatani, 2010).

We report the performance evaluation of our approach, and compare it with other state-of-the-art semi-supervised clustering methods in terms of accuracy and running time. Extensive experiments are conducted over real-world datasets. The results are encouraging and indicate the effectiveness of our approach. Especially, our approach can leverage small amount of pairwise constraints to increase the performance. We believe that this is a good property in the semi-supervised learning setting.

The rest of this chapter is organized as follows. Section 2 explains the framework of semi-supervised clustering. Section 3 explains the details of our approach for clustering under pairwise constraints. Section 4 reports the performance evaluation over various document datasets. Section 5 discusses the effectiveness of our approach. Section 6 summarizes our contributions and suggests future directions.

2. Semi-supervised clustering

2.1 Preliminaries

Let $\boldsymbol{X}$ be a set of instances. For a set $\boldsymbol{X}$, $|\boldsymbol{X}|$ represents its cardinality.

A graph $G = (\boldsymbol{V}, \boldsymbol{E})$ consists of a finite set of vertices $\boldsymbol{V}$, a set of edges $\boldsymbol{E}$ over $\boldsymbol{V} \times \boldsymbol{V}$. The set $\boldsymbol{E}$ can be interpreted as representing a binary relation over $\boldsymbol{V}$. A pair of vertices (v_i, v_j) is in the binary relation defined by a graph $G = (\boldsymbol{V}, \boldsymbol{E})$ if and only if the pair $(v_i, v_j) \in \boldsymbol{E}$.

An edge-weighted graph $G = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$ is defined as a graph $G = (\boldsymbol{V}, \boldsymbol{E})$ with a weight on each edge in $\boldsymbol{E}$. When $|\boldsymbol{V}| = n$, i.e., the number of vertices in a graph is n, the weights in $\boldsymbol{W}$ can be represented as an n by n matrix $\mathbf{W}$ [1], where w_{ij} in $\mathbf{W}$ stands for the weight on the edge for the pair $(v_i, v_j) \in \boldsymbol{E}$. $\mathbf{W}_{ij}$ also stands for the element w_{ij} in the matrix. We set $w_{ij} = 0$ for pairs $(v_i, v_j) \notin \boldsymbol{E}$. In addition, we assume that $G = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$ is an undirected, simple graph without self-loops. Thus, the weight matrix $\mathbf{W}$ is symmetric and its diagonal elements are zeros.

2.2 Clustering

In general, clustering methods can be divided into two approaches: hierarchical methods and partitioning methods. (Jain et al., 1999). Hierarchical methods construct a cluster hierarchy, or a tree of clusters (called a dendrogram), whose leaves are the data points and whose internal nodes represent nested clusters of various sizes (Guha et al., 1998). Hierarchical methods can be further subdivided into *agglomerative* and *divisive* ones. On the other hand, partitioning methods return a single partition of the entire data under a fixed parameters (number of clusters, thresholds, etc.). Each cluster can be represented by its centroid

[1] A bold italic symbol $\boldsymbol{W}$ denotes a set, while a bold symbol $\mathbf{W}$ denotes a matrix.

(k-means algorithms (Hartigan & Wong, 1979)), or by one of its instances located near its center (k-medoid algorithms (Ng & Han, 2002)). For a recent overview of various clustering methods, please refer to (Jain et al., 1999).

When pairwise similarities among instances are specified, the entire data can be represented as an edge-weighted graph. Various graph-theoretic clustering approaches have been proposed to find subsets of vertices in a graph based on the edges among the vertices. Several methods utilizes graph coloring techniques (Guënoche et al., 1991; Yoshida & Ogino, 2011). Other methods are based on the flow or cut in graph, such as spectral clustering (von Luxburg, 2007). Graph-based spectral approach is also utilized in information-theoretic clustering (Yoshida, 2011).

2.3 Semi-supervised clustering

When the auxiliary or side information for data assignment in clustering is represented as a set of constraints, the *semi-supervised clustering* problem is (informally) described as follows.

Problem 1 (Semi-Supervised Clustering)**.** *For a given set of data $\boldsymbol{X}$ and specified constraints, find a partition (a set of clusters) $\boldsymbol{T} = \{t_1, \ldots, t_k\}$ which satisfies the specified constraints.*

There can be various forms of constraints. Based on the previous work (Li et al., 2008; Tang et al., 2007; Wagstaff et al., 2001; Xing et al., 2003), we consider the following two kinds of constraints defined in (Wagstaff et al., 2001):

Definition 1 (Pairwise Constraints)**.** *For a given data instances $\boldsymbol{X}$ and a partition (a set of clusters) $\boldsymbol{C} = \{c_1, \ldots, c_k\}$, must-link constraints $\boldsymbol{C}_{ML}$ and cannot-link constraints $\boldsymbol{C}_{CL}$ are sets of pairs such that:*

$$\exists (x_i, x_j) \in \boldsymbol{C}_{ML} \Rightarrow \exists c \in \boldsymbol{C}, (x_i \in c \wedge x_j \in c) \tag{1}$$

$$\exists (x_i, x_j) \in \boldsymbol{C}_{ML} \Rightarrow \exists c_a, c_b \in \boldsymbol{C}, c_a \neq c_b, (x_i \in c_a \wedge x_j \in c_b) \tag{2}$$

Intuitively, must-link constraints (also called must-links in this paper) specifies the pairs of instances in the same cluster, and cannot-link constraints (also called cannot-links) specifies the pairs of instances in different clusters.

3. Graph-based semi-supervised clustering

3.1 A graph-based approach

By assuming that some similarity measure for the pairs of instances is specified, we have proposed a graph-based approach for constrained clustering problem (Yoshida, 2010; Yoshida & Okatani, 2010). Based on the similarities, the entire data instances $\boldsymbol{X}$ can be represented as an edge-weighted graph $G = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$ where w_{ij} represents the similarity between a pair (x_i, x_j). In our approach, each data instance $x \in \boldsymbol{X}$ corresponds to a vertex $v \in \boldsymbol{V}$ in G. Thus, we abuse the symbol $\boldsymbol{X}$ to denote the set of vertices in G in the rest of the paper. Also, we assume that all w_{ij} is non-negative.

Definition 1 specifies two kinds of constraints. For must-link constraints, our approach utilizes a method based on graph contraction in general graph theory (Diestel, 2006) and treat it as hard constraints (Sections 3.2); for cannot-link constraints, our approach utilizes a method based on graph Laplacian in spectral graph theory (Chung, 1997; von Luxburg, 2007) and

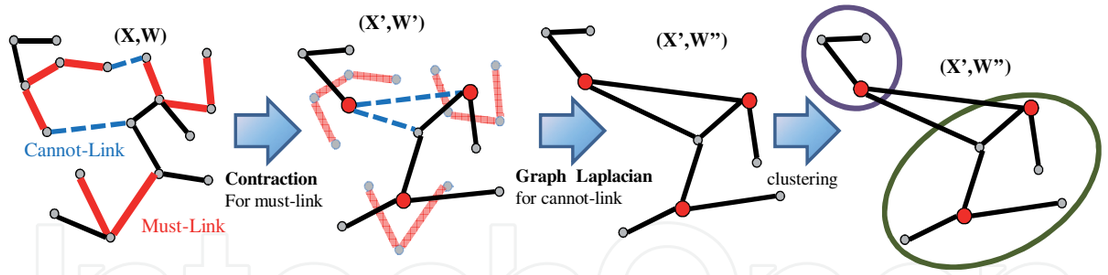


Fig. 1. Overview of our approach.

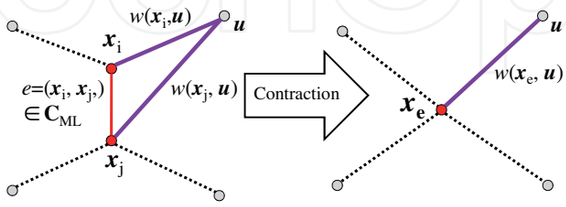


Fig. 2. Contraction for must-link constraints.

treat them as soft constraints under the optimization framework (Section 3.3). The overview of our approach is illustrated in Fig. 1.

3.2 Graph contraction for must-link constraints

When must-link constraints are treated as hard constraints, the transitive law holds among the constraints. This means that, for any two pairs (x_i, x_j) and $(x_j, x_l) \in \boldsymbol{C}_{ML}$, x_i and x_l should also be in the same cluster (however, the cluster label is not known). In order to enforce the transitive law in must-links, we utilize graph contraction in general graph theory (Diestel, 2006) and modify the graph G for a data set $\boldsymbol{X}$ based on the specified must-links.

Definition 2 (Contraction)**.** *Let e=(x_i, x_j) be an edge of a graph $G = (\boldsymbol{X}, \boldsymbol{E})$. define By G/e, we denote the graph $(\boldsymbol{X}', \boldsymbol{E}')$ obtained from G by contracting the edge e into a new vertex x_e, where:*

$$\boldsymbol{X}' = (\boldsymbol{X} \backslash \{x_i, x_j\}) \cup \{x_e\} \tag{3}$$

$$\boldsymbol{E}' = \{(u,v) \in \boldsymbol{E} | \{u,v\} \cap \{x_i, x_j\} = \phi\} \cup \{(x_e, u) | (x_i, u) \in \boldsymbol{E} \backslash \{e\} \text{ or } (x_j, u) \in \boldsymbol{E} \backslash \{e\}\} \tag{4}$$

G/e stands for the graph obtained from G by *contracting* an edge e into a new vertex x_e. The created vertex x_e becomes adjacent to all the former neighbors of x_i and x_j.

By contracting an edge e into a new vertex x_e, the newly created vertex x_e becomes adjacent to all the former neighbors of x_i and x_j. Repeated application of contraction for all the edges (pairs of instance) for must-links guarantees that the transitive law in must-links is sustained in the cluster assignment.

As described above, the entire dataset $\boldsymbol{X}$ is represented as an edge-weighted graph G in our approach. Thus, after contracting an edge e=$(x_i, x_j) \in \boldsymbol{C}_{ML}$ into the newly created vertex x_e, it is necessary to define the weights in the contracted graph G/e. The weights in G represent the similarities among vertices. The original similarities should at least be sustained after contracting an edge in $\boldsymbol{C}_{ML}$, since must-link constraints are for enforcing the similarities, not for reducing.

Based on the above observation, we define the weights in the contracted graph G/e as:

$$w(x_e, u)' = \max(w(x_i, u), w(x_j, u)) \quad if \ \ (x_i, u) \in \boldsymbol{E} \ or (x_j, u) \in \boldsymbol{E} \tag{5}$$

$$w(u, v)' = w(u, v) \quad otherwise \tag{6}$$

where $w(\cdot, \cdot)'$ stands for the weight in the contracted graph G/e. In eq.(5), the function max realizes the above requirement, and guarantees the non-decreasing properties of similarities (weights) after contraction of an edge. On the other hand, the original weight is preserved in eq.(6).

For each pair of edges in must-links, we apply graph contraction and define weights in the contracted graph based on eq.(5) and eq.(6). This results in modifying the original graph G into another graph $G^{'} = (\boldsymbol{X}^{'}, \boldsymbol{E}^{'}, \boldsymbol{W}^{'})$ (as illustrated in Fig. 2). The number of vertices in the contracted graph G' is denoted as $n^{'} = |\boldsymbol{X}^{'}|$. Note that the originally specified cannot-links also need to be modified during graph contraction with respect to must-links. The updated cannot-links over the created graph G' is denoted as $\boldsymbol{C}'_{CL}$.

3.3 Graph Laplacian for cannot-link constraints

3.3.1 Spectral clustering

The objective of clustering is to assign similar instances to the same cluster and dissimilar ones to different clusters. To realize this, we utilize spectral clustering, which is based on the minimum cut of a graph. In spectral clustering (Ng et al., 2001; von Luxburg, 2007), data clustering is realized by seeking a function f: $\boldsymbol{X} \rightarrow \mathcal{R}$ over the dataset $\boldsymbol{X}$ such that the learned function assigns similar values for similar instances and vice versa. The values assigned for the entire dataset can be represented as a vector. By denoting the assigned value for the i-th data instance as f_i, data clustering can be formalized as an optimization problem to find the vector $\boldsymbol{f}$ which minimizes the following objective function :

$$J_0 = \boldsymbol{f}^t \mathbf{L} \boldsymbol{f} \tag{7}$$

where $\boldsymbol{f}^t$ is a transpose of vector $\boldsymbol{f}$, and the matrix $\mathbf{L}$ is defined as:

$$\mathbf{D} = diag(d_1, \ldots, d_n) \quad (d_i = \sum_{j=1}^{n} w_{ij}) \tag{8}$$

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{9}$$

where $diag()$ in eq.(8) represents a diagonal matrix with the specified diagonal elements. The matrix $\mathbf{D}$ in eq.(8) is the degree matrix of a graph, and is calculated based on the weights in the graph. The matrix $\mathbf{L}$ in eq.(9) is called graph Laplacian (Chung, 1997; Ng et al., 2001; von Luxburg, 2007). Some clustering method, such as kmeans (Hartigan & Wong, 1979) or spherical kmeans (skmeans) (Dhillon & Modha, 2001)[2], is applied to the constructed data representation of instances (Ng et al., 2001; von Luxburg, 2007).

3.3.2 Graph Laplacian for cannot-link constraints

We utilized the framework of spectral clustering in Section 3.3.1. Furthermore, to reflect cannot-link constraints in the clustering process, we formalize the clustering under constraints

[2] skmeans is a standard clustering algorithm for high-dimensional sparse data.

as an optimization problem, and consider the minimization of the following objective function:

$$J = \frac{1}{2}\{\sum_{i,j} w_{ij}^{'}||f_i - f_j||^2 - \lambda \sum_{u,v \in C_{CL}^{'}} w_{uv}^{'}||f_u - f_v||^2\} \tag{10}$$

where i and j sum over the vertices in the contracted graph $G^{'}$, and $\boldsymbol{C}_{CL}^{'}$ stands for the cannot-link constraints over $G^{'}$. $\lambda \in [0,1]$ is a hyper-parameter in our approach. The first term corresponds to the smoothness of the assigned values in spectral graph theory, and the second term represents the influence of cannot-links in optimization. Note that by setting $\lambda \in [0,1]$, the objective function in (10) is guaranteed to be a convex function.

From the above objective function in eq.(10), we can derive the following unnormalized graph Laplacian $\mathbf{L}^{''}$ which incorporates cannot-links as:

$$J = \frac{1}{2}\{\sum_{i,j} w_{ij}^{'}||f_i - f_j||^2 - \lambda \sum_{u,v \in C_{CL}^{'}} w_{uv}^{'}||f_u - f_v||^2\} = \boldsymbol{f}^t \mathbf{L}^{''} \boldsymbol{f} \tag{11}$$

The matrix $\mathbf{L}^{''}$ is defined based on the following matrices:

$$(\mathbf{C}^{'})_{uv} = \begin{cases} 1 & (x_u, x_v) \in C_{CL}^{'} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$\mathbf{W}^c = \mathbf{C}^{'} \odot \mathbf{W}^{'}, \quad \mathbf{W}^{''} = \mathbf{W}^{'} - \lambda \mathbf{W}^c \tag{13}$$

$$d_i = \sum_{j=1}^{n^{'}} w_{ij}^{'}, \quad d_i^c = \sum_{j=1}^{n^{'}} w_{ij}^c \tag{14}$$

$$\mathbf{D}^{''} = diag(d_1^{''}, \ldots, d_{n^{'}}^{''}), \quad d_i^{''} = d_i - \lambda d_i^c \tag{15}$$

$$\mathbf{L}^{''} = \mathbf{D}^{''} - \mathbf{W}^{''} \tag{16}$$

where $\odot$ stands for the Hadamard product (element-wise multiplication) of two matrices.

The above process amounts to modifying the representation of the contracted graph $G^{'}$ into another graph $G^{''}$, with the modified weights $\mathbf{W}^{''}$ in eq.(13). Thus, as illustrated in Fig. 1, our approach modifies the original graph G into the contracted graph $G^{'}$ with must-link constraints, and then into another graph $G^{''}$ with cannot-link constraints and similarities.

It is known that some form "balancing" among clusters needs to be considered for obtaining meaningful results (von Luxburg, 2007). Based on eq.(14) and eq.(16), we utilize the following normalized objective function:

$$J_{sym} = \sum_{i,j} w_{ij}^{''}||\frac{f_i}{\sqrt{d_i^{''}}} - \frac{f_j}{\sqrt{d_j^{''}}}||^2 \tag{17}$$

over the graph $G^{''}$. Minimizing J_{sym} in eq.(17) amounts to solving the generalized eigen-problem $\mathbf{L}^{''}\boldsymbol{f} = \alpha \mathbf{D}^{''}\boldsymbol{f}$, where α corresponds to an eigenvalue and $\boldsymbol{f}$ corresponds to the generalized eigenvector with the eigenvalue.

Algorithm 1 graph-based semi-supervised clustering (GBSSC)

Require: $G = (\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{W})$; //*an edge-weighted graph*
Require: $\boldsymbol{C}_{ML}$; //*must-link constraints*
Require: $\boldsymbol{C}_{CL}$; //*cannot-link constraints*
Require: l; //*the number of generalized eigenvectors*
Require: k; //*the number of clusters*

1: **for** each $e \in C_{ML}$ **do**
2: contract e and create the contracted graph G/e;
3: **end for**
// *Let* $G^{'} = (\boldsymbol{X}^{'}, \boldsymbol{E}^{'}, \boldsymbol{W}^{'})$ *be the contracted graph.*
4: create $\mathbf{C}^{'}_{uv}, \mathbf{W}^{c}, \mathbf{W}^{''}, \mathbf{D}^{''}$ as eq.(12) $\sim$ eq.(15).
5: $\mathbf{L}^{''}_{sym} = \mathbf{I} - \mathbf{D}^{''-\frac{1}{2}}\mathbf{W}^{''}\mathbf{D}^{''-\frac{1}{2}}$
6: Find l eigenvectors $\mathbf{F} = \{\boldsymbol{f}^1, \ldots, \boldsymbol{f}^l\}$ for $\mathbf{L}^{''}_{sym}$, with the smallest non-zero eigenvalues.
7: Conduct clustering of data which are represented as $\mathbf{F}$ and construct clusters.
8: **return** clusters

Furthermore, the number of generalized eigenvectors can be extended to more than one. In that case, the generalized eigenvectors with positive eigenvalues are selected with ascending order of eigenvalues. The generalized eigenvectors with respect to the modified graph corresponds to the embeeded representation of the whole data instances.

3.4 Algorithm

The graph-based semi-supervised clustering method (called GBSSC) is summarized in Algorithm 1. The contracted graph $G^{'}$ is constructed from lines 1 to 3 based on the specified must-links. Lines 4 to 6 conduct the minimization of J_{sym} in eq.(17), which is represented as the normalized graph Laplacian $\mathbf{L}^{''}_{sym}$ at line 5.

These correspond to the spectral embedding of the entire data instances $\boldsymbol{X}$ onto the subspace spanned by $\mathbf{F} = \{\boldsymbol{f}^1, \ldots, \boldsymbol{f}^l\}$ (Belkin & Niyogi, 2002). Note that pairwise constraints for semi-supervised clustering are also utilized on the construction of the embedded representation in our approach and thus differs from (Belkin & Niyogi, 2002). Some clustering method is applied to the data at line 7 and the constructed clusters are returned. Currently spherical kmeans (skmeans) (Dhillon & Modha, 2001) is utilized at line 7.

4. Evaluations

4.1 Experimental settings

4.1.1 Datasets

Based on the previous work (Dhillon et al., 2003; Tang et al., 2007), we evaluated our approach on 20 Newsgroup dataset (hereafter, called 20NG) [3] and TREC datasets [4]. Clustering of these datasets corresponds to document clustering, and each document is represented in the standard vector space model based on the occurrences of terms. Since the number of terms are

[3] http://people.csail.mit.edu/~jrennie/20Newsgroups/. (20news-18828 was utilized)
[4] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download

dataset	included groups
Multi5	comp.graphics, rec.motorcycles,rec.sport.baseball, sci.space talk.politics.mideast
Multi10	alt.atheism, comp.sys.mac.hardware,misc.forsale, rec.autos,rec.sport.hockey, sci.crypt,sci.med, sci.electronics,sci.space,talk.politics.guns
Multi15	alt.atheism, comp.graphics, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.guns, talk.politics.mideast, talk.politics.misc

Table 1. Datasets from 20 Newsgroup dataset

dataset	#attributes	#classes	#data
hitech	126372	6	2301
reviews	126372	5	4069
sports	126372	7	8580
la1	31372	6	3204
la2	31372	6	3075
la2	31372	6	6279
k1b	21839	6	2340
ohscal	11465	10	11162
fbis	2000	17	2463

Table 2. TREC datasets (original representation)

huge in general, these are high-dimensional sparse datasets. Please note that our approach is generic and not specific to document clustering.

As in (Dhillon et al., 2003; Tang et al., 2007), 50 documents were sampled from each group (cluster) in order to create a sample for one dataset, and 10 samples were created for each dataset. For each sample, we conducted stemming using porter stemmer [5] and MontyTagger [6], removed stop words, and selected 2,000 words with descending order of mutual information (Cover & Thomas, 2006).

For TREC datasets, we utilized 9 datasets in Table 2. We followed the same procedure in 20NG and created 10 samples for each dataset[7]. Since these datasets are already preprocessed and represented as count data, we did not conduct stemming or tagging.

4.1.2 Evaluation measures

For each dataset, the cluster assignment was evaluated with respect to Normalized Mutual Information (*NMI*) (Strehl & Ghosh, 2002; Tang et al., 2007). Let C, $\hat{C}$ stand for random variables over the true and assigned clusters. *NMI* is defined as

$$NMI = \frac{I(\hat{C};C)}{(H(\hat{C}) + H(C))/2} \tag{18}$$

[5] http://www.tartarus.org/~martin/PorterStemmer
[6] http://web.media.mit.edu/~hugo/montytagger
[7] On fbis, 35 data were sampled for each class.

where $H(\cdot)$ is Shannon Entropy, and $I(\cdot;\cdot)$ is Mutual Information among the random variables C and $\hat{C}$. *NMI* corresponds to the accuracy of assignment. Thus, the larger *NMI* is, the better the cluster assignment is with respect to the "ground-truth" labels in each dataset.

All the compared methods first construct the representation for clustering and then apply some clustering method (e.g., skmeans). The running time (CPU time in second) for representation construction was measured on a computer with Debian/GNU Linux, Intel Xeon W5590, 36 GB memory. All the methods were implemented with R language and R packages.

4.1.3 Comparison

We compared our approach with SCREEN (Tang et al., 2007) and PCP (Li et al., 2008) (details are described in Section 5.2). Since all the compared methods are partitioning based clustering methods, we assume that the number of clusters k in each dataset is available.

SCREEN (Tang et al., 2007) conducts semi-supervised clustering by projecting the given data instances onto the subspace where the covariance with respect to the given data representation is maximized. To realize this, the covariance matrix with respect to the original data representation is constructed and their eigenvectors are utilized for projection. For high-dimensional data such as documents, this process is rather expensive, since the number of attributes (e.g., terms) gets large. To alleviate this problem, PCA (Principal Component Analysis) was first utilized as pre-processing to reduce the number of dimension in the data representation. We followed the same process in (Tang et al., 2007) and pre-processed data by PCA using 100 eigenvectors, and SCREEN was applied to the pre-processed data as in (Tang et al., 2007).

PCP (Li et al., 2008) first conducts metric learning based on the semi-definite programming, and then kernel k-means clustering is conducted over the learned metric. Some package (e.g. Csdp) is utilized to solve the semi-definite programming based on the specified pairwise constraints and similarities.

4.1.4 Parameters

The parameters under the pairwise constraints in Definition 1 are:

1) the number of constraints
2) the pairs of instances for constraints

As for **2)**, pairs of instances were randomly sampled from each dataset to generate the constraints. Thus, the main parameter is **1)**, the number of constraints, for must-links and cannot-links. We set the numbers of these two types of constrains to be the same, and varied the number of constraints.

Each data instance $\boldsymbol{x}$ in a dataset was normalized such that $\boldsymbol{x}^t\boldsymbol{x} = 1$, and Euclidian distance was utilized for SCREEN as in (Tang et al., 2007). With this normalization, cosine similarity, which is widely utilized as the standard similarity measure in document processing, was utilized for GBSSC and PCP, and the initial edge-weighted graph for each dataset was constructed with the similarities. The number of generalized eigenvectors l was set to the number of clusters k. In addition, following the procedure in (Li et al., 2008), m-nearest neighbor graph was constructed for PCP (m was set to 10 in the experiment). The hyper-parameter λ in eq.(10) was set to 0.5, since GBSSC is robust to this value as reported in Section 4.2.

4.1.5 Evaluation procedure

For each number of constraints, the pairwise constraints (must-links and cannot-links) were generated randomly based on the ground-truth labels in the datasets, and clustering was conducted with the generated constraints. Clustering with the same number of constraints was repeated 10 times with different initial configuration in clustering. In addition, the above process was also repeated 10 times for each number of constraints. Thus, for each dataset and the number of constraints, 100 runs were conducted. Furthermore, this process was repeated over 10 samples for each dataset. Thus, the average of 1,000 runs is reported for each dataset.

4.2 Evaluation of graph-based approach

Our approach modifies the data representation in a dataset according to the specified constraints. Especially, the similarities among instances (weights in a graph) are modified. The other possible approach would be to set the weights (similarities) as:

i) each pair $(x_i, x_j) \in C_{ML}$ to the maximum similarity
ii) each pair $(x_i, x_j) \in C_{CL}$ to the minimum similarity

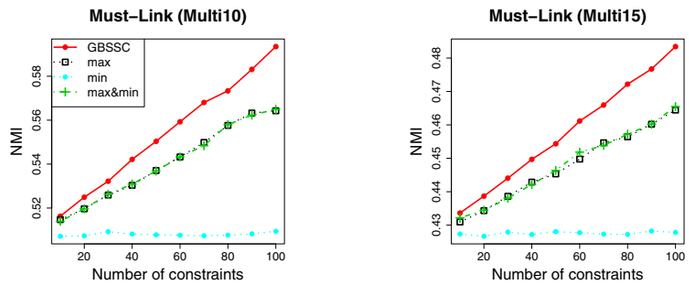


Fig. 3. Weight medication comparison.

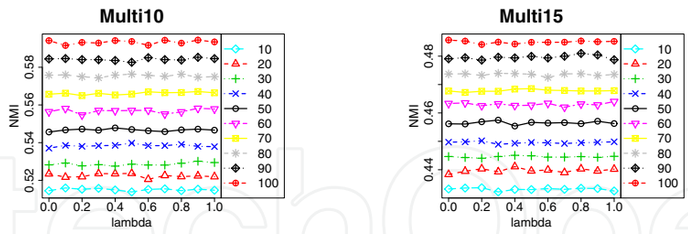


Fig. 4. Influence of λ.

First, we compared our approach for the handling of must-links in Section 3.2 with the above approaches on Multi10 and Multi15 datasets. The results are summarized in Fig. 3. In Fig. 3, horizontal axis corresponds to the number of constraints; vertical one corresponds to *NMI*. In the legend, max (black lines with boxes) stands for i), min (blue dotted lines with circles) stands for ii), and max&min (green dashed lines with crosses) stands for when both i) and ii) are employed. GBSSC (red solid lines with circles) stands for our approach.

The results in Fig. 3 show that GBSSC outperformed others and that it is effective in terms of the weight modification in a graph. One of the reasons for the results in Fig. 3 is that, when i) (max) is utilized, only the instances connected with must-links are affected, and thus they tend to be collected into a smaller "isolated" cluster. Creating rather small clusters makes the

performance degraded. On the other hand, in our approach, instances adjacent to must-links are also affected via contraction.

As for ii) (min), the instances connected with cannot-links are by definition dissimilar with each other and their weights would be small in the original representation. Thus, setting the weights over must-links to the minimal value in the dataset does not affect the overall performance so much. These are illustrated in Fig. 5 and Fig. 6.

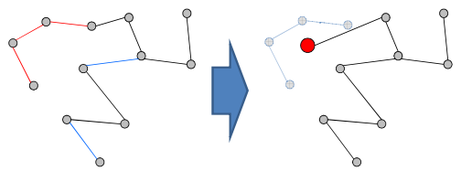


Fig. 5. Contraction of must-link constraints.

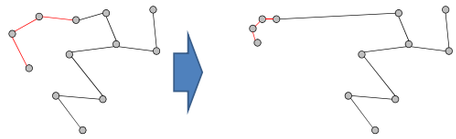


Fig. 6. Weight modification of must-link constraints.

Next, we evaluated the handling of cannot-links in Section 3.3. We varied the value of hyper-parameter λ in eq.(10) and analyzed its influence. The results are summarized in Fig. 4. In Fig. 4, horizontal axis corresponds to the value of λ, and the values in the legend corresponds to the number of pairwise constraints (e.g., 10 corresponds to the situation where the number of pairwise constraints are 10). The performance of GBSSC was not so much affected by the value of λ. Thus, our approach can be said as relatively robust with respect to this parameter. In addition, the accuracy (*NMI*) increased *monotonically* as the number of constraints increased. Thus, it can be concluded that GBSSC reflects the pairwise constraints and improves the performance based on semi-supervised information.

4.3 Evaluation on real world datasets

We report the comparison of our approach with other compared methods. In the reported figures, horizontal axis corresponds to the number of constraints; vertical one corresponds to either *NMI* or CPU time (in sec.).

In the legend in the figures, red lines correspond to our GBSSC, black dotted lines to SCREEN, green lines to PCP. Also, +PCA stands for the case where the dataset was first pre-processed by PCA (using 100 eigenvectors as in (Tang et al., 2007)) and then the corresponding method was applied. GBSSC+PCP (with purple lines) corresponds to the situation where must-links were handled by contraction in Section 3.2 and cannot-links by PCP.

4.3.1 20 Newsgroup datasets

The results for 20NG dataset are summarized in Figs. 7. These are the average of 10 datasets for each set of groups (i.e., average of 1000 runs). The results indicate that our approach outperformed other methods with respect to *NMI* (Fig. 7) when l=k [8]. For Multi5, although the

[8] The number of generalized eigenvectors l was set to the number of clusters k. Note that we did not conduct any tuning for the value of l in these experiments. (Tang et al., 2007) reports that SCREEN could be improved by tuning the number of dimensions.

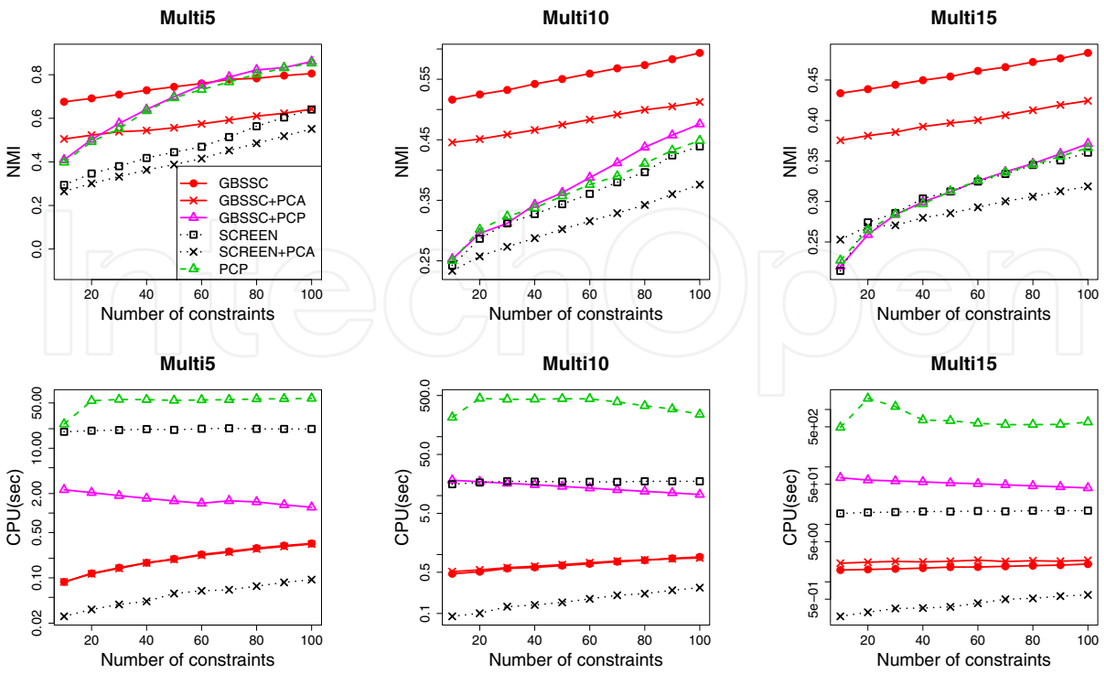


Fig. 7. Results on 20-Newsgroup

performance of PCP got close to that of GBSSC as the number of constraints increased, GBSSC was faster more than two orders of magnitude (100 times faster). Likewise, GBSSC+PCP and PCP were almost the same with respect to *NMI*, but the former was faster with more than one order (10 times faster). Although SCREEN+PCA was two to five times faster than GBSSC, it was inferior with respect to *NMI*. Utilization of PCA as the pre-processing enables this speed-up for SCREEN, in compensation for the accuracy (*NMI*).

Dimensionality reduction with PCA was effective for the speed-up of SCREEN, but it was not for GBSSC. On the other hand, it *deteriorated* their performance with respect to *NMI*. Thus, it is not necessary to utilize pre-processing such as PCA for GBSSC, and still our approach showed better performance.

4.3.2 TREC datasets

The results for TREC datasets are summarized in Fig. 8 and Fig. 9. As shown in Table 2, the number of dimensions (attributes) are huge in TREC datasets. Since calculating the eigenvalues of the covariance matrix with large number of attributes takes too much time, when SCREEN was applied to non-preprocessed data with PCA, it was too slow. Thus, SCREEN was applied only to the pre-processed data in TREC datasets. (shown as SCREEN+PCA).

On the whole, the results were quite similar to those in 20NG. Our approach outperformed SCREEN (in TREC datasets, SCREEN+PCA) with respect to *NMI*. It also outperformed PCP in most datasets, however, as the number of constraints increased, the latter showed better performance for review and sports datasets. In addition, PCP seems to improve the performance as the number of constraints increase. When GBSSC is utilized with PCP

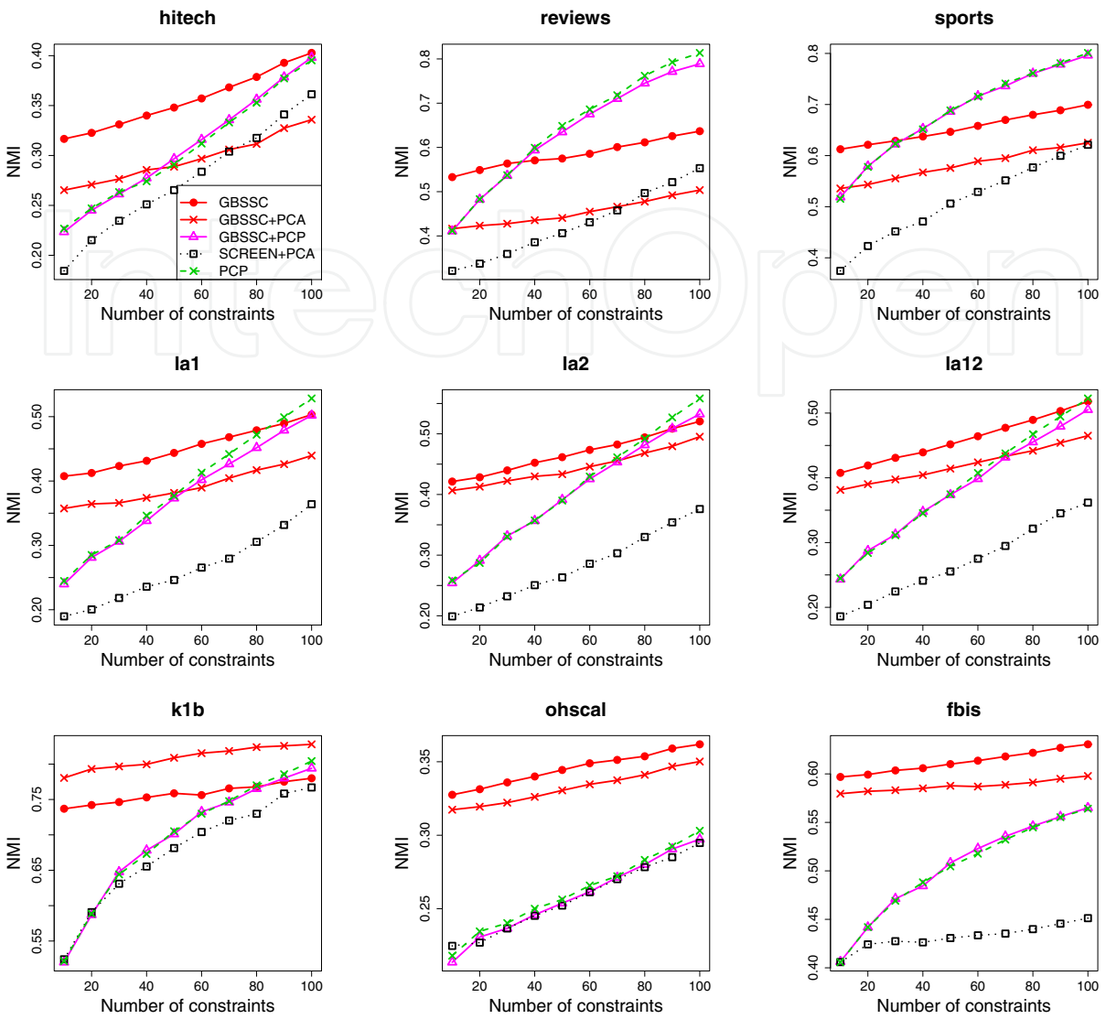


Fig. 8. Results on TREC datasets (*NMI*)

(denoted as GBSSC+PCP in the figure), it showed almost equivalent performance with respect to *NMI*, but the former was faster with more than one order.

5. Discussions

5.1 Effectiveness

The reported results show that our approach is effective in terms of the accuracy of cluster assignment (*NMI*). GBSSC outperformed SCREEN in all the datasets. Although it did not outperformed PCP in some TREC datasets with respect to *NMI*, but it was faster more than two orders of magnitude. Utilization of PCA as data pre-processing for dimensionality reduction enables the speed-up of SCREEN, in compensation for the accuracy of cluster assignment. On the other hand, PCP showed better performance in some datasets with respect to accuracy of cluster assignment, in compensation for the running time. Besides, since SCREEN originally conducts linear dimensionality reduction based on constraints, utilization

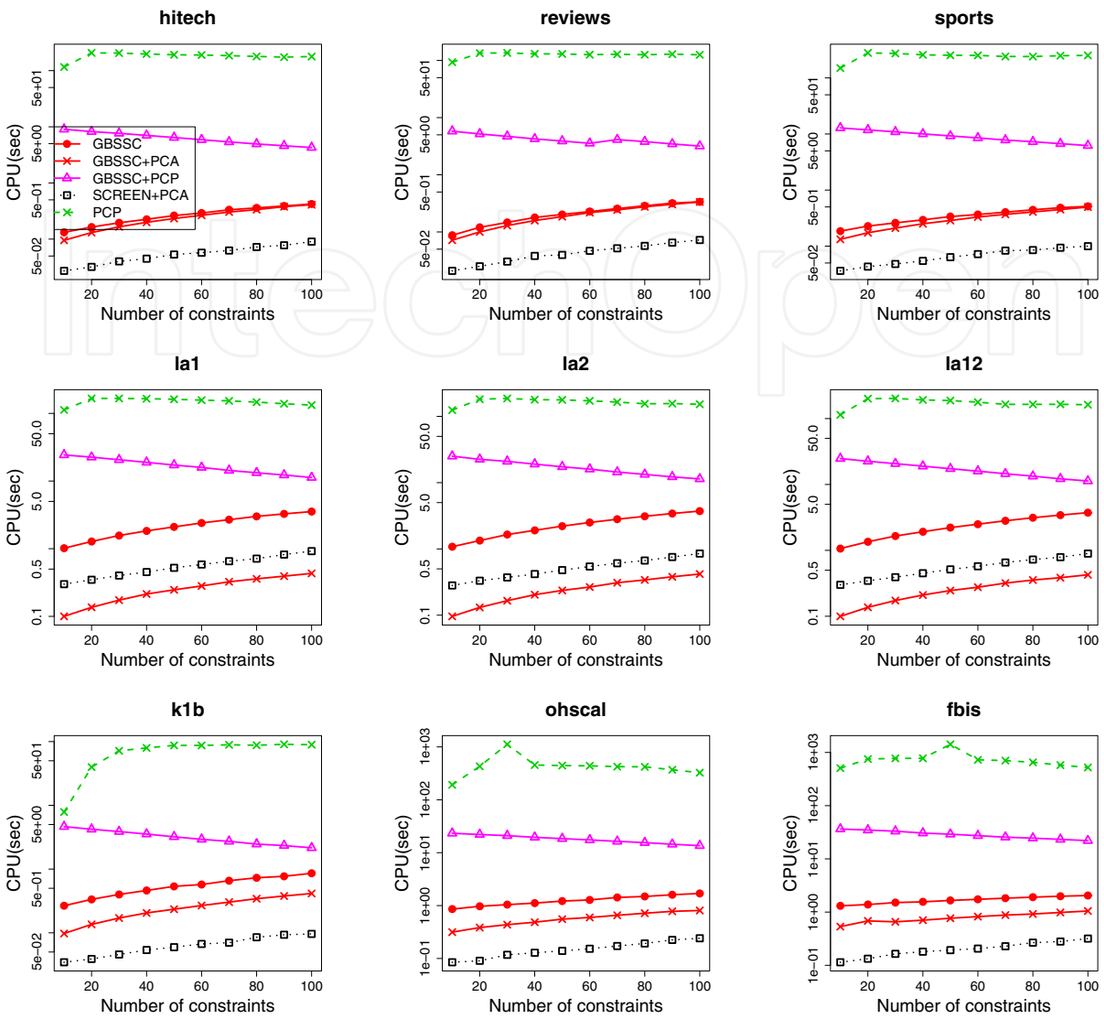


Fig. 9. Results on TREC datasets (CPU time (in seconds))

of *another* linear dimensionality reduction (such as PCA) as pre-processing might obscure its effect.

From these results, our approach can be said as effective in terms of the balance between the accuracy of cluster assignment and running time. Especially, it can leverage small amount of pairwise constraints to increase the performance. We believe that this is a good property in the semi-supervised learning setting.

5.2 Related work

Various approaches have been conducted on semi-supervised clustering. Among them are: constraint-based, distance-based, and hybrid approaches (Tang et al., 2007). The constraint-based approach tries to guide the clustering process with the specified pairwise instance constraints (Wagstaff et al., 2001). The distance-based approach utilizes metric learning techniques to acquire the distance measure during the clustering process based on the

specified pairwise instance constraints (Li et al., 2008; Xing et al., 2003). The hybrid approach combines these two approaches under a probabilistic framework (Basu et al., 2004).

As for the semi-supervised clustering problem, (Wagstaff et al., 2001) proposed a clustering algorithm called COP-kmeans based on the famous kmeans algorithm. When assigning each data item to the cluster with minimum distance as in kmeans, COP-kmeans checks the constraint satisfaction and assigns each data item only to the admissible cluster (which does not violate the constraints).

SCREEN (Tang et al., 2007) first converts the data representation based on must-link constraints and removes the constraints. This process corresponds to contraction in our approach, but the weight definition is different. After that, based on cannot-link constraints, it finds out the linear mapping (linear projection) to a subspace where the variance among the data is maximized. Finally, clustering of the mapped data is conducted on the subspace.

PCP (Li et al., 2008) deals with the semi-supervised clustering problem by finding a mapping onto a space where the specified constraints are reflected. Using the specified constraints, it conducts metric learning based on the semi-definite programming and learn the kernel matrix on the mapped space. Although the explicit representation of the mapping or the data representation on the mapped space is not learned, kernel k-means clustering (Girolami, 2002) is conducted over the learned metric.

6. Conclusion

In this chapter we presented our semi-supervised clustering method based on graph contraction in general graph theory and graph Laplacian in spectral graph theory. Our approach can exploit a small amount of pairwise constraints as well as pairwise relations (similarities) among the data instances. Utilization of graph representation of instances enables to deal with the pairwise constraints as well as pairwise similarities over a unified representation. In order to reflect the pairwise constraints on the clustering process, the graph structure for the entire data instances is modified by graph contraction in general graph theory (Diestel, 2006) and graph Laplacian in spectral graph theory (Chung, 1997; von Luxburg, 2007).

We reported the performance of our approach over two real-world datasets with respect to the type of constraints as well as the number of constraints. We also compared with other state-of-the-art semi-supervised clustering methods in terms of accuracy of cluster assignment and running time. The experimental results indicate that our approach is effective in terms of the balance between the accuracy of cluster assignment and running time. Especially, it could leverage a small amount of pairwise constraints to improve the clustering performance. We plan to continue this line of research and to improve the presented approach in future.

7. Acknowledgments

The author is grateful to Mr. Okatani and Mr. Ogino for their help on implementation.

8. References

Basu, S., Bilenko, M. & Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering, *KDD-04*, pp. 59–68.

Basu, S., Davidson, I. & Wagstaff, K. (eds) (2008). *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, Chapman & Hall/CRC Press.

Belkin, M. & Niyogi, P. (2002). Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Computation* 15: 1373–1396.

Blum, A. & Mitchell, T. (1998). Combining labeled and unlabeled data with to-training, *Proc. 11th Computational Learning Theory*, pp. 92–100.

Chapelle, O., Schölkopf, B. & Zien, A. (eds) (2006). *Semi-Supervised Learning*, MIT Press.

Chung, F. (1997). *Spectral Graph Theory*, American Mathematical Society.

Cover, T. & Thomas, J. (2006). *Elements of Information Theory*, Wiley.

Dhillon, J., Mallela, S. & Modha, D. (2003). Information-theoretic co-clustering, *Proc. KDD'03*, pp. 89–98.

Dhillon, J. & Modha, D. (2001). Concept decompositions for large sparse text data using clustering, *Machine Learning* 42: 143–175.

Diestel, R. (2006). *Graph Theory*, Springer.

Girolami, M. (2002). Mercer kernel-based clustering in feature space, *IEEE Transactions on Neural Networks* 13(3): 780–784.

Guënoche, A., Hansen, P. & Jaumard, B. (1991). Efficient algorithms for divisive hierarchical clustering with the diameter criterion, *J. of Classification* 8: 5–30.

Guha, S., Rastogi, R. & Shim, K. (1998). Cure: An efficient clustering algorithm for large databases, *Proc. the ACM SIGMOD Conference*, pp. 73–84.

Hartigan, J. & Wong, M. (1979). Algorithm AS136: A k-means clustering algorithm, *Journal of Applied Statistics* 28: 100–108.

Jain, A., Murty, M. & P.J., F. (1999). Data clustering: A review, *ACM Computing Surveys* 31: 264–323.

Li, Z., Liu, J. & Tang, X. (2008). Pairwise constraint propagation by semidefinite programming for semi-supervised classification, *ICML-08*, pp. 576–583.

Ng, A. Y., Jordan, M. I. & Weiss, Y. (2001). On Spectral Clustering: Analysis and an algorithm, *Proc. NIPS 14*, pp. 849–856.

Ng, R. & Han, J. (2002). Clarans: a method for clustering objects for spatial data mining, *IEEE Transactions on Knowledge and Data Engineering* 14(5): 1003–1016.

Strehl, A. & Ghosh, J. (2002). Cluster Ensembles -A Knowledge Reuse Framework for Combining Multiple Partitions, *J. Machine Learning Research* 3(3): 583–617.

Tang, W., Xiong, H., Zhong, S. & Wu, J. (2007). Enhancing semi-supervised clustering : A feature projection perspective, *Proc. KDD'07*, pp. 707–716.

von Luxburg, U. (2007). A tutorial on spectral clustering, *Statistics and Computing* 17(4): 395–416.

Wagstaff, K., Cardie, C., Rogers, S. & Schroedl, S. (2001). Constrained k-means clustering with background knowledge, *In ICML01*, pp. 577–584.

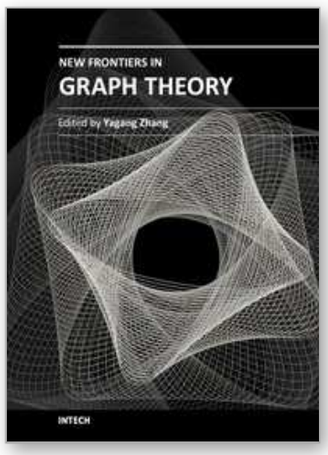
Xing, E. P., Ng, A. Y., Jordan, M. I. & Russell, S. (2003). Distance metric learning, with application to clustering with side-information, *NIPS 15*, pp. 505–512.

Yoshida, T. (2010). Performance Evaluation of Constraints in Graph-based Semi-Supervised Clustering, *Proc. AMT-2010, LNAI 6335*, pp. 138–149.

Yoshida, T. (2011). A graph model for mutual information based clustering, *Journal of Intelligent Information Systems* 37(2): 187–216.

Yoshida, T. & Ogino, H. (2011). A re-coloring approach for graph b-coloring based clustering, *International Journal of Knowledge-Based & Intelligent Engineering Systems* . accepted.

Yoshida, T. & Okatani, K. (2010). A Graph-based projection approach for Semi-Supervised Clustering, *Proc. PKAW-2010, LNAI 6232*, pp. 1–13.



New Frontiers in Graph Theory
Edited by Dr. Yagang Zhang

ISBN 978-953-51-0115-4
Hard cover, 526 pages
Publisher InTech
Published online 02, March, 2012
Published in print edition March, 2012

Nowadays, graph theory is an important analysis tool in mathematics and computer science. Because of the inherent simplicity of graph theory, it can be used to model many different physical and abstract systems such as transportation and communication networks, models for business administration, political science, and psychology and so on. The purpose of this book is not only to present the latest state and development tendencies of graph theory, but to bring the reader far enough along the way to enable him to embark on the research problems of his own. Taking into account the large amount of knowledge about graph theory and practice presented in the book, it has two major parts: theoretical researches and applications. The book is also intended for both graduate and postgraduate students in fields such as mathematics, computer science, system sciences, biology, engineering, cybernetics, and social sciences, and as a reference for software professionals and practitioners.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen