

# Regularized Clustering for Documents <sup>\*</sup>

Fei Wang, Changshui Zhang  
 State Key Lab of Intelligent Tech. and Systems  
 Department of Automation, Tsinghua University  
 Beijing, China, 100084  
 feiwang03@gmail.com

Tao Li  
 School of Computer Science  
 Florida International University  
 Miami, FL 33199, U.S.A.  
 taoli@cs.fiu.edu

## ABSTRACT

In recent years, document clustering has been receiving more and more attentions as an important and fundamental technique for unsupervised document organization, automatic topic extraction, and fast information retrieval or filtering. In this paper, we propose a novel method for clustering documents using regularization. Unlike traditional globally regularized clustering methods, our method first construct a local regularized linear label predictor for each document vector, and then combine all those local regularizers with a global smoothness regularizer. So we call our algorithm *Clustering with Local and Global Regularization (CLGR)*. We will show that the cluster memberships of the documents can be achieved by eigenvalue decomposition of a sparse symmetric matrix, which can be efficiently solved by iterative methods. Finally our experimental evaluations on several datasets are presented to show the superiorities of *CLGR* over traditional document clustering methods.

## Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Clustering*; I.2.6 [Artificial Intelligence]: Learning—*Concept Learning*

## General Terms

Algorithms

## Keywords

Document clustering, Regularization

<sup>\*</sup>The work of Fei Wang, Changshui Zhang is supported by the China Natural Science Foundation No. 60675009. The work of Tao Li is partially supported by NSF IIS-0546280 and NIH/NIGMS S06 GM008205.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.  
 Copyright 2007 ACM 978-1-59593-597-7/07/0007 ...\$5.00.

## 1. INTRODUCTION

Document clustering has been receiving more and more attentions as an important and fundamental technique for unsupervised document organization, automatic topic extraction, and fast information retrieval or filtering. A good document clustering approach can assist the computers to automatically organize the document corpus into a meaningful cluster hierarchy for efficient browsing and navigation, which is very valuable for complementing the deficiencies of traditional information retrieval technologies. As pointed out by [8], the information retrieval needs can be expressed by a *spectrum* ranged from narrow keyword-matching based *search* to broad information *browsing* such as what are the major international events in recent months. Traditional document retrieval engines tend to fit well with the *search* end of the spectrum, *i.e.* they usually provide specified search for documents matching the user's query, however, it is hard for them to meet the needs from the rest of the spectrum in which a rather broad or vague information is needed. In such cases, efficient browsing through a good cluster hierarchy will be definitely helpful.

Generally, document clustering methods can be mainly categorized into two classes: hierarchical methods and partitioning methods. The hierarchical methods group the data points into a hierarchical tree structure using bottom-up or top-down approaches. For example, hierarchical agglomerative clustering (*HAC*) [13] is a typical bottom-up hierarchical clustering method. It takes each data point as a single cluster to start off with and then builds bigger and bigger clusters by grouping similar data points together until the entire dataset is encapsulated into one final cluster. On the other hand, partitioning methods decompose the dataset into a number of disjoint clusters which are usually optimal in terms of some predefined criterion functions. For instance, *K-means* [13] is a typical partitioning method which aims to minimize the sum of the squared distance between the data points and their corresponding cluster centers. In this paper, we will focus on the partitioning methods.

As we know that there are two main problems existing in partitioning methods (like *Kmeans* and *Gaussian Mixture Model (GMM)* [16]): (1) the predefined criterion is usually non-convex which causes many local optimal solutions; (2) the iterative procedure (*e.g.* the *Expectation Maximization (EM)* algorithm) for optimizing the criterions usually makes the final solutions heavily depend on the initializations. In the last decades, many methods have been proposed to overcome the above problems of the partitioning methods [19][28].

Recently, another type of partitioning methods based on clustering on data graphs have aroused considerable interests in the machine learning and data mining community. The basic idea behind these methods is to first model the whole dataset as a weighted graph, in which the graph nodes represent the data points, and the weights on the edges correspond to the similarities between pairwise points. Then the cluster assignments of the dataset can be achieved by optimizing some criterions defined on the graph. For example *Spectral Clustering* is one kind of the most representative graph-based clustering approaches, it generally aims to optimize some cut value (e.g. *Normalized Cut* [22], *Ratio Cut* [7], *Min-Max Cut* [11]) defined on an undirected graph. After some relaxations, these criterions can usually be optimized via eigen-decompositions, which is guaranteed to be global optimal. In this way, spectral clustering efficiently avoids the problems of the traditional partitioning methods as we introduced in last paragraph.

In this paper, we propose a novel document clustering algorithm that inherits the superiority of spectral clustering, i.e. the final cluster results can also be obtained by exploit the eigen-structure of a symmetric matrix. However, unlike spectral clustering, which just enforces a smoothness constraint on the data labels over the whole data manifold [2], our method first construct a regularized linear label predictor for each data point from its neighborhood as in [25], and then combine the results of all these local label predictors with a global label smoothness regularizer. So we call our method *Clustering with Local and Global Regularization (CLGR)*. The idea of incorporating both local and global information into label prediction is inspired by the recent works on semi-supervised learning [31], and our experimental evaluations on several real document datasets show that *CLGR* performs better than many state-of-the-art clustering methods.

The rest of this paper is organized as follows: in section 2 we will introduce our *CLGR* algorithm in detail. The experimental results on several datasets are presented in section 3, followed by the conclusions and discussions in section 4.

## 2. THE PROPOSED ALGORITHM

In this section, we will introduce our *Clustering with Local and Global Regularization (CLGR)* algorithm in detail. First let's see the how the documents are represented throughout this paper.

### 2.1 Document Representation

In our work, all the documents are represented by the weighted term-frequency vectors. Let  $\mathcal{W} = \{w_1, w_2, \dots, w_m\}$  be the complete vocabulary set of the document corpus (which is preprocessed by the stopwords removal and words stemming operations). The term-frequency vector  $\mathbf{x}_i$  of document  $d_i$  is defined as

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{im}]^T, \quad x_{ik} = t_{ik} \log \left( \frac{n}{idf_k} \right),$$

where  $t_{ik}$  is the term frequency of  $w_k \in \mathcal{W}$ ,  $n$  is the size of the document corpus,  $idf_k$  is the number of documents that contain word  $w_k$ . In this way,  $\mathbf{x}_i$  is also called the *TF-IDF* representation of document  $d_i$ . Furthermore, we also normalize each  $\mathbf{x}_i$  ( $1 \leq i \leq n$ ) to have a unit length, so that each document is represented by a *normalized TF-IDF* vector.

## 2.2 Local Regularization

As its name suggests, *CLGR* is composed of two parts: *local regularization* and *global regularization*. In this subsection we will introduce the local regularization part in detail.

### 2.2.1 Motivation

As we know that *clustering* is one type of *learning* techniques, it aims to organize the dataset in a reasonable way. Generally speaking, *learning* can be posed as a problem of function estimation, from which we can get a *good* classification function that will assign labels to the training dataset and even the unseen testing dataset with some cost minimized [24]. For example, in the two-class classification scenario<sup>1</sup>(in which we exactly know the label of each document), a linear classifier with least square fit aims to learn a column vector  $\mathbf{w}$  such that the squared cost

$$\mathcal{J} = \frac{1}{n} \sum (\mathbf{w}^T \mathbf{x}_i - y_i)^2 \quad (1)$$

is minimized, where  $y_i \in \{+1, -1\}$  is the label of  $\mathbf{x}_i$ . By taking  $\partial \mathcal{J} / \partial \mathbf{w} = 0$ , we get the solution

$$\mathbf{w}^* = \left( \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^T \right)^{-1} \left( \sum_{i=1}^n \mathbf{x}_i y_i \right), \quad (2)$$

which can further be written in its matrix form as

$$\mathbf{w}^* = \left( \mathbf{X} \mathbf{X}^T \right)^{-1} \mathbf{X} \mathbf{y}, \quad (3)$$

where  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$  is an  $m \times n$  *document matrix*,  $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$  is the *label vector*. Then for a test document  $t$ , we can determine its label by

$$l = \text{sign}(\mathbf{w}^{*T} \mathbf{u}), \quad (4)$$

where  $\text{sign}(\cdot)$  is the sign function.

A natural problem in Eq.(3) is that the matrix  $\mathbf{X} \mathbf{X}^T$  may be singular and thus not invertable (e.g. when  $m \gg n$ ). To avoid such a problem, we can add a regularization term and minimize the following criterion

$$\mathcal{J}' = \frac{1}{n} \sum_{i=1}^n (\mathbf{w}^T \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|^2, \quad (5)$$

where  $\lambda$  is a regularization parameter. Then the optimal solution that minimize  $\mathcal{J}'$  is given by

$$\mathbf{w}^* = \left( \mathbf{X} \mathbf{X}^T + \lambda n \mathbf{I} \right)^{-1} \mathbf{X} \mathbf{y}, \quad (6)$$

where  $\mathbf{I}$  is an  $m \times m$  identity matrix. It has been reported that the regularized linear classifier can achieve very good results on text classification problems [29].

However, despite its empirical success, the regularized linear classifier is on earth a global classifier, i.e.  $\mathbf{w}^*$  is estimated using the whole training set. According to [24], this may not be a smart idea, since a unique  $\mathbf{w}^*$  may not be good enough for predicting the labels of the whole input space. In order to get better predictions, [6] proposed to train classifiers locally and use them to classify the testing points. For example, a testing point will be classified by the local classifier trained using the training points located in the vicinity

<sup>1</sup>In the following discussions we all assume that the documents coming from only two classes. The generalizations of our method to multi-class cases will be discussed in section 2.5.

of it. Although this method seems slow and stupid, it is reported that it can get better performances than using a unique global classifier on certain tasks [6].

### 2.2.2 Constructing the Local Regularized Predictors

Inspired by their success, we proposed to apply the local learning algorithms for clustering. The basic idea is that, for each document vector  $\mathbf{x}_i$  ( $1 \leq i \leq n$ ), we train a local label predictor based on its  $k$ -nearest neighborhood  $\mathcal{N}_i$ , and then use it to predict the label of  $\mathbf{x}_i$ . Finally we will combine all those local predictors by minimizing the sum of their prediction errors. In this subsection we will introduce how to construct those local predictors.

Due to the simplicity and effectiveness of the regularized linear classifier that we have introduced in section 2.2.1, we choose it to be our local label predictor, such that for each document  $\mathbf{x}_i$ , the following criterion is minimized

$$\mathcal{J}_i = \frac{1}{n_i} \sum_{\mathbf{x}_j \in \mathcal{N}_i} \|\mathbf{w}_i^T \mathbf{x}_j - q_j\|^2 + \lambda_i \|\mathbf{w}_i\|^2, \quad (7)$$

where  $n_i = |\mathcal{N}_i|$  is the cardinality of  $\mathcal{N}_i$ , and  $q_j$  is the cluster membership of  $\mathbf{x}_j$ . Then using Eq.(6), we can get the optimal solution is

$$\mathbf{w}_i^* = \left( \mathbf{X}_i \mathbf{X}_i^T + \lambda_i n_i \mathbf{I} \right)^{-1} \mathbf{X}_i \mathbf{q}_i, \quad (8)$$

where  $\mathbf{X}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{in_i}]$ , and we use  $\mathbf{x}_{ik}$  to denote the  $k$ -th nearest neighbor of  $\mathbf{x}_i$ .  $\mathbf{q}_i = [q_{i1}, q_{i2}, \dots, q_{in_i}]^T$  with  $q_{ik}$  representing the cluster assignment of  $\mathbf{x}_{ik}$ . The problem here is that  $\mathbf{X}_i \mathbf{X}_i^T$  is an  $m \times m$  matrix with  $m \gg n_i$ , *i.e.* we should compute the inverse of an  $m \times m$  matrix for every document vector, which is computationally prohibited. Fortunately, we have the following theorem:

**Theorem 1.**  $\mathbf{w}_i^*$  in Eq.(8) can be rewritten as

$$\mathbf{w}_i^* = \mathbf{X}_i \left( \mathbf{X}_i^T \mathbf{X}_i + \lambda_i n_i \mathbf{I}_i \right)^{-1} \mathbf{q}_i, \quad (9)$$

where  $\mathbf{I}_i$  is an  $n_i \times n_i$  identity matrix.

*Proof.* Since

$$\mathbf{w}_i^* = \left( \mathbf{X}_i \mathbf{X}_i^T + \lambda_i n_i \mathbf{I} \right)^{-1} \mathbf{X}_i \mathbf{q}_i,$$

then

$$\begin{aligned} & \left( \mathbf{X}_i \mathbf{X}_i^T + \lambda_i n_i \mathbf{I} \right) \mathbf{w}_i^* = \mathbf{X}_i \mathbf{q}_i \\ \implies & \mathbf{X}_i \mathbf{X}_i^T \mathbf{w}_i^* + \lambda_i n_i \mathbf{w}_i^* = \mathbf{X}_i \mathbf{q}_i \\ \implies & \mathbf{w}_i^* = (\lambda_i n_i)^{-1} \mathbf{X}_i \left( \mathbf{q}_i - \mathbf{X}_i^T \mathbf{w}_i^* \right). \end{aligned}$$

Let

$$\boldsymbol{\beta} = (\lambda_i n_i)^{-1} \left( \mathbf{q}_i - \mathbf{X}_i^T \mathbf{w}_i^* \right),$$

then

$$\begin{aligned} & \mathbf{w}_i^* = \mathbf{X}_i \boldsymbol{\beta} \\ \implies & \lambda_i n_i \boldsymbol{\beta} = \mathbf{q}_i - \mathbf{X}_i^T \mathbf{w}_i^* = \mathbf{q}_i - \mathbf{X}_i^T \mathbf{X}_i \boldsymbol{\beta} \\ \implies & \mathbf{q}_i = \left( \mathbf{X}_i^T \mathbf{X}_i + \lambda_i n_i \mathbf{I}_i \right) \boldsymbol{\beta} \\ \implies & \boldsymbol{\beta} = \left( \mathbf{X}_i^T \mathbf{X}_i + \lambda_i n_i \mathbf{I}_i \right)^{-1} \mathbf{q}_i. \end{aligned}$$

Therefore

$$\mathbf{w}_i^* = \mathbf{X}_i \boldsymbol{\beta} = \mathbf{X}_i \left( \mathbf{X}_i^T \mathbf{X}_i + \lambda_i n_i \mathbf{I}_i \right)^{-1} \mathbf{q}_i \quad \square$$

Using theorem 1, we only need to compute the inverse of an  $n_i \times n_i$  matrix for every document to train a local label predictor. Moreover, for a new testing point  $\mathbf{u}$  that falls into  $\mathcal{N}_i$ , we can classify it by the sign of

$$q_u = \mathbf{w}_i^{*T} \mathbf{u} = \mathbf{u}^T \mathbf{w}_i = \mathbf{u}^T \mathbf{X}_i \left( \mathbf{X}_i^T \mathbf{X}_i + \lambda_i n_i \mathbf{I}_i \right)^{-1} \mathbf{q}_i.$$

This is an attractive expression since we can determine the cluster assignment of  $\mathbf{u}$  by using the inner-products between the points in  $\{\mathbf{u} \cup \mathcal{N}_i\}$ , which suggests that such a local regularizer can easily be *kernelized* [21] as long as we define a proper kernel function.

### 2.2.3 Combining the Local Regularized Predictors

After all the local predictors having been constructed, we will combine them together by minimizing

$$\mathcal{J}_l = \sum_{i=1}^n \left( \mathbf{w}_i^{*T} \mathbf{x}_i - q_i \right)^2, \quad (10)$$

which stands for the sum of the prediction errors for all the local predictors. Combining Eq.(10) with Eq.(6), we can get

$$\begin{aligned} \mathcal{J}_l &= \sum_{i=1}^n \left( \mathbf{w}_i^{*T} \mathbf{x}_i - q_i \right)^2 \\ &= \sum_{i=1}^n \left( \mathbf{x}_i^T \mathbf{X}_i \left( \mathbf{X}_i^T \mathbf{X}_i + \lambda_i n_i \mathbf{I}_i \right)^{-1} \mathbf{q}_i - q_i \right)^2 \\ &= \|\mathbf{P} \mathbf{q} - \mathbf{q}\|^2, \end{aligned} \quad (11)$$

where  $\mathbf{q} = [q_1, q_2, \dots, q_n]^T$ , and the  $\mathbf{P}$  is an  $n \times n$  matrix constructing in the following way. Let

$$\boldsymbol{\alpha}^i = \mathbf{x}_i^T \mathbf{X}_i \left( \mathbf{X}_i^T \mathbf{X}_i + \lambda_i n_i \mathbf{I}_i \right)^{-1},$$

then

$$\mathbf{P}_{ij} = \begin{cases} \boldsymbol{\alpha}^i_j, & \text{if } \mathbf{x}_j \in \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases}, \quad (12)$$

where  $\mathbf{P}_{ij}$  is the  $(i, j)$ -th entry of  $\mathbf{P}$ , and  $\boldsymbol{\alpha}^i_j$  represents the  $j$ -th entry of  $\boldsymbol{\alpha}^i$ .

Till now we can write the criterion of clustering by combining locally regularized linear label predictors  $\mathcal{J}_l$  in an explicit mathematical form, and we can minimize it directly using some standard optimization techniques. However, the results may not be good enough since we only exploit the local informations of the dataset. In the next subsection, we will introduce a global regularization criterion and combine it with  $\mathcal{J}_l$ , which aims to find a good clustering result in a local-global way.

## 2.3 Global Regularization

In data clustering, we usually require that the cluster assignments of the data points should be sufficiently smooth with respect to the underlying data manifold, which implies (1) the nearby points tend to have the same cluster assignments; (2) the points on the same structure (*e.g.* submanifold or cluster) tend to have the same cluster assignments [31].

Without the loss of generality, we assume that the data points reside (roughly) on a low-dimensional manifold  $\mathcal{M}^2$ , and  $q$  is the cluster assignment function defined on  $\mathcal{M}$ , *i.e.*

<sup>2</sup>We believe that the text data are also sampled from some low dimensional manifold, since it is impossible for them to

for  $\forall \mathbf{x} \in \mathcal{M}$ ,  $q(\mathbf{x})$  returns the cluster membership of  $\mathbf{x}$ . The smoothness of  $q$  over  $\mathcal{M}$  can be calculated by the following *Dirichlet integral* [2]

$$D[q] = \frac{1}{2} \int_{\mathcal{M}} \|\nabla q(\mathbf{x})\|^2 d\mathcal{M}, \quad (13)$$

where the gradient  $\nabla q$  is a vector in the tangent space  $\mathcal{T}\mathcal{M}_{\mathbf{x}}$ , and the integral is taken with respect to the standard measure on  $\mathcal{M}$ . If we restrict the scale of  $q$  by  $\langle q, q \rangle_{\mathcal{M}} = 1$  (where  $\langle \cdot, \cdot \rangle_{\mathcal{M}}$  is the inner product induced on  $\mathcal{M}$ ), then it turns out that finding the smoothest function minimizing  $D[q]$  reduces to finding the eigenfunctions of the *Laplace Beltrami* operator  $\mathcal{L}$ , which is defined as

$$\mathcal{L}q \triangleq -\text{div}\nabla q, \quad (14)$$

where *div* is the divergence of a vector field.

Generally, the graph can be viewed as the discretized form of manifold. We can model the dataset as an weighted undirected graph as in spectral clustering [22], where the graph nodes are just the data points, and the weights on the edges represent the similarities between pairwise points. Then it can be shown that minimizing Eq.(13) corresponds to minimizing

$$\mathcal{J}_g = \mathbf{q}^T \mathbf{L} \mathbf{q} = \sum_{i=1}^n (q_i - q_j)^2 w_{ij}, \quad (15)$$

where  $\mathbf{q} = [q_1, q_2, \dots, q_n]^T$  with  $q_i = q(\mathbf{x}_i)$ ,  $\mathbf{L}$  is the *graph Laplacian* with its  $(i, j)$ -th entry

$$L_{ij} = \begin{cases} d_i - w_{ij}, & \text{if } i = j \\ -w_{ij}, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are adjacent} \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

where  $d_i = \sum_j w_{ij}$  is the *degree* of  $\mathbf{x}_i$ ,  $w_{ij}$  is the similarity between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . If  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are adjacent<sup>3</sup>,  $w_{ij}$  is usually computed in the following way

$$w_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}, \quad (17)$$

where  $\sigma$  is a dataset dependent parameter. It is proved that under certain conditions, such a form of  $w_{ij}$  to determine the weights on graph edges leads to the convergence of graph Laplacian to the Laplace Beltrami operator [3][18].

In summary, using Eq.(15) with exponential weights can effectively measure the smoothness of the data assignments with respect to the intrinsic data manifold. Thus we adopt it as a *global regularizer* to punish the smoothness of the predicted data assignments.

## 2.4 Clustering with Local and Global Regularization

Combining the contents we have introduced in section 2.2 and section 2.3 we can derive the clustering criterion is

$$\begin{aligned} \min_{\mathbf{q}} \quad & \mathcal{J} = \mathcal{J}_l + \lambda \mathcal{J}_g = \|\mathbf{P}\mathbf{q} - \mathbf{q}\|^2 + \lambda \mathbf{q}^T \mathbf{L} \mathbf{q} \\ \text{s.t.} \quad & q_i \in \{-1, +1\}, \end{aligned} \quad (18)$$

where  $\mathbf{P}$  is defined as in Eq.(12), and  $\lambda$  is a regularization parameter to trade off  $\mathcal{J}_l$  and  $\mathcal{J}_g$ . However, the discrete

fill in the whole high-dimensional sample space. And it has been shown that the manifold based methods can achieve good results on text classification tasks [31].

<sup>3</sup>In this paper, we define  $\mathbf{x}_i$  and  $\mathbf{x}_j$  to be adjacent if  $\mathbf{x}_i \in \mathcal{N}(\mathbf{x}_j)$  or  $\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)$ .

constraint of  $p_i$  makes the problem an NP hard integer programming problem. A natural way for making the problem solvable is to remove the constraint and relax  $q_i$  to be continuous, then the objective that we aims to minimize becomes

$$\begin{aligned} \mathcal{J} &= \|\mathbf{P}\mathbf{q} - \mathbf{q}\|^2 + \lambda \mathbf{q}^T \mathbf{L} \mathbf{q} \\ &= \mathbf{q}^T (\mathbf{P} - \mathbf{I})^T (\mathbf{P} - \mathbf{I}) \mathbf{q} + \lambda \mathbf{q}^T \mathbf{L} \mathbf{q} \\ &= \mathbf{q}^T \left( (\mathbf{P} - \mathbf{I})^T (\mathbf{P} - \mathbf{I}) + \lambda \mathbf{L} \right) \mathbf{q}, \end{aligned} \quad (19)$$

and we further add a constraint  $\mathbf{q}^T \mathbf{q} = 1$  to restrict the scale of  $\mathbf{q}$ . Then our objective becomes

$$\begin{aligned} \min_{\mathbf{q}} \quad & \mathcal{J} = \mathbf{q}^T \left( (\mathbf{P} - \mathbf{I})^T (\mathbf{P} - \mathbf{I}) + \lambda \mathbf{L} \right) \mathbf{q} \\ \text{s.t.} \quad & \mathbf{q}^T \mathbf{q} = 1 \end{aligned} \quad (20)$$

Using the *Lagrangian* method, we can derive that the optimal solution  $\mathbf{q}$  corresponds to the smallest eigenvector of the matrix  $\mathbf{M} = (\mathbf{P} - \mathbf{I})^T (\mathbf{P} - \mathbf{I}) + \lambda \mathbf{L}$ , and the cluster assignment of  $\mathbf{x}_i$  can be determined by the sign of  $q_i$ , i.e.  $\mathbf{x}_i$  will be classified as class one if  $q_i > 0$ , otherwise it will be classified as class 2.

## 2.5 Multi-Class CLGR

In the above we have introduced the basic framework of *Clustering with Local and Global Regularization (CLGR)* for the two-class clustering problem, and we will extending it to multi-class clustering in this subsection.

First we assume that all the documents belong to  $C$  classes indexed by  $\mathcal{L} = \{1, 2, \dots, C\}$ .  $q^c$  is the classification function for class  $c$  ( $1 \leq c \leq C$ ), such that  $q^c(\mathbf{x}_i)$  returns the *confidence* that  $\mathbf{x}_i$  belongs to class  $c$ . Our goal is to obtain the value of  $q^c(\mathbf{x}_i)$  ( $1 \leq c \leq C$ ,  $1 \leq i \leq n$ ), and the cluster assignment of  $\mathbf{x}_i$  can be determined by  $\{q^c(\mathbf{x}_i)\}_{c=1}^C$  using some proper discretization methods that we will introduce later.

Therefore, in this multi-class case, for each document  $\mathbf{x}_i$  ( $1 \leq i \leq n$ ), we will construct  $C$  locally linear regularized label predictors whose normal vectors are

$$\mathbf{w}_i^{c*} = \mathbf{X}_i \left( \mathbf{X}_i^T \mathbf{X}_i + \lambda_i n_i \mathbf{I}_i \right)^{-1} \mathbf{q}_i^c \quad (1 \leq c \leq C), \quad (21)$$

where  $\mathbf{X}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{in_i}]$  with  $\mathbf{x}_{ik}$  being the  $k$ -th neighbor of  $\mathbf{x}_i$ , and  $\mathbf{q}_i^c = [q_{i1}^c, q_{i2}^c, \dots, q_{in_i}^c]^T$  with  $q_{ik}^c = q^c(\mathbf{x}_{ik})$ . Then  $(\mathbf{w}_i^{c*})^T \mathbf{x}_i$  returns the predicted confidence of  $\mathbf{x}_i$  belonging to class  $c$ . Hence the local prediction error for class  $c$  can be defined as

$$\mathcal{J}_l^c = \sum_{i=1}^n \left( (\mathbf{w}_i^{c*})^T \mathbf{x}_i - q_i^c \right)^2, \quad (22)$$

And the total local prediction error becomes

$$\mathcal{J}_l = \sum_{c=1}^C \mathcal{J}_l^c = \sum_{c=1}^C \sum_{i=1}^n \left( (\mathbf{w}_i^{c*})^T \mathbf{x}_i - q_i^c \right)^2. \quad (23)$$

As in Eq.(11), we can define an  $n \times n$  matrix  $\mathbf{P}$  (see Eq.(12)) and rewrite  $\mathcal{J}_l$  as

$$\mathcal{J}_l = \sum_{c=1}^C \mathcal{J}_l^c = \sum_{c=1}^C \|\mathbf{P}\mathbf{q}^c - \mathbf{q}^c\|^2. \quad (24)$$

Similarly we can define the *global smoothness regularizer*

in multi-class case as

$$\mathcal{J}_g = \sum_{c=1}^C \sum_{i=1}^n (q_i^c - q_j^c)^2 w_{ij} = \sum_{c=1}^C (\mathbf{q}^c)^T \mathbf{L} \mathbf{q}^c. \quad (25)$$

Then the criterion to be minimized for *CLGR* in multi-class case becomes

$$\begin{aligned} \mathcal{J} &= \mathcal{J}_l + \lambda \mathcal{J}_g \\ &= \sum_{c=1}^C \left[ \|\mathbf{P} \mathbf{q}^c - \mathbf{q}^c\|^2 + \lambda (\mathbf{q}^c)^T \mathbf{L} \mathbf{q}^c \right] \\ &= \sum_{c=1}^C \left[ (\mathbf{q}^c)^T \left( (\mathbf{P} - \mathbf{I})^T (\mathbf{P} - \mathbf{I}) + \lambda \mathbf{L} \right) \mathbf{q}^c \right] \\ &= \text{trace} \left[ \mathbf{Q}^T \left( (\mathbf{P} - \mathbf{I})^T (\mathbf{P} - \mathbf{I}) + \lambda \mathbf{L} \right) \mathbf{Q} \right], \quad (26) \end{aligned}$$

where  $\mathbf{Q} = [\mathbf{q}^1, \mathbf{q}^2, \dots, \mathbf{q}^C]$  is an  $n \times C$  matrix, and  $\text{trace}(\cdot)$  returns the trace of a matrix. The same as in Eq.(20), we also add the constraint that  $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$  to restrict the scale of  $\mathbf{Q}$ . Then our optimization problem becomes

$$\begin{aligned} \min_{\mathbf{Q}} \quad & \mathcal{J} = \text{trace} \left[ \mathbf{Q}^T \left( (\mathbf{P} - \mathbf{I})^T (\mathbf{P} - \mathbf{I}) + \lambda \mathbf{L} \right) \mathbf{Q} \right] \\ \text{s.t.} \quad & \mathbf{Q}^T \mathbf{Q} = \mathbf{I}, \quad (27) \end{aligned}$$

From the *Ky Fan* theorem [28], we know the optimal solution of the above problem is

$$\mathbf{Q}^* = [\mathbf{q}_1^*, \mathbf{q}_2^*, \dots, \mathbf{q}_C^*] \mathbf{R}, \quad (28)$$

where  $\mathbf{q}_k^*$  ( $1 \leq k \leq C$ ) is the eigenvector corresponds to the  $k$ -th smallest eigenvalue of matrix  $(\mathbf{P} - \mathbf{I})^T (\mathbf{P} - \mathbf{I}) + \lambda \mathbf{L}$ , and  $\mathbf{R}$  is an arbitrary  $C \times C$  matrix. Since the values of the entries in  $\mathbf{Q}^*$  is continuous, we need to further discretize  $\mathbf{Q}^*$  to get the cluster assignments of all the data points. There are mainly two approaches to achieve this goal:

1. As in [20], we can treat the  $i$ -th row of  $\mathbf{Q}$  as the embedding of  $\mathbf{x}_i$  in a  $C$ -dimensional space, and apply some traditional clustering methods like *kmeans* to clustering these embeddings into  $C$  clusters.
2. Since the optimal  $\mathbf{Q}^*$  is not unique (because of the existence of an arbitrary matrix  $\mathbf{R}$ ), we can pursue an optimal  $\mathbf{R}$  that will *rotate*  $\mathbf{Q}^*$  to an *indication matrix*<sup>4</sup>. The detailed algorithm can be referred to [26].

The detailed algorithm procedure for *CLGR* is summarized in table 1.

### 3. EXPERIMENTS

In this section, experiments are conducted to empirically compare the clustering results of *CLGR* with other 8 representative document clustering algorithms on 5 datasets. First we will introduce the basic informations of those datasets.

#### 3.1 Datasets

We use a variety of datasets, most of which are frequently used in the information retrieval research. Table 2 summarizes the characteristics of the datasets.

<sup>4</sup>Here an *indication matrix*  $\mathbf{T}$  is a  $n \times c$  matrix with its  $(i, j)$ -th entry  $\mathbf{T}_{ij} \in \{0, 1\}$  such that for each row of  $\mathbf{Q}^*$  there is only one 1. Then the  $\mathbf{x}_i$  can be assigned to the  $j$ -th cluster such that  $j = \text{arg}_j \mathbf{Q}_{ij}^* = 1$ .

**Table 1: Clustering with Local and Global Regularization (*CLGR*)**

<b>Input:</b>
1. Dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ ;
2. Number of clusters $C$ ;
3. Size of the neighborhood $K$ ;
4. Local regularization parameters $\{\lambda_i\}_{i=1}^n$ ;
5. Global regularization parameter $\lambda$ ;
<b>Output:</b>
The cluster membership of each data point.
<b>Procedure:</b>
1. Construct the $K$ nearest neighborhoods for each data point;
2. Construct the matrix $\mathbf{P}$ using Eq.(12);
3. Construct the Laplacian matrix $\mathbf{L}$ using Eq.(16);
4. Construct the matrix $\mathbf{M} = (\mathbf{P} - \mathbf{I})^T (\mathbf{P} - \mathbf{I}) + \lambda \mathbf{L}$ ;
5. Do eigenvalue decomposition on $\mathbf{M}$ , and construct the matrix $\mathbf{Q}^*$ according to Eq.(28);
6. Output the cluster assignments of each data point by properly discretize $\mathbf{Q}^*$ .

**Table 2: Descriptions of the document datasets**

Datasets	Number of documents	Number of classes
CSTR	476	4
WebKB4	4199	4
Reuters	2900	10
WebACE	2340	20
Newsgroup4	3970	4

**CSTR.** This is the dataset of the abstracts of technical reports published in the Department of Computer Science at a university. The dataset contained 476 abstracts, which were divided into four research areas: Natural Language Processing(NLP), Robotics/Vision, Systems, and Theory.

**WebKB.** The WebKB dataset contains webpages gathered from university computer science departments. There are about 8280 documents and they are divided into 7 categories: student, faculty, staff, course, project, department and other. The raw text is about 27MB. Among these 7 categories, student, faculty, course and project are four most populous entity-representing categories. The associated subset is typically called WebKB4.

**Reuters.** The Reuters-21578 Text Categorization Test collection contains documents collected from the Reuters newswire in 1987. It is a standard text categorization benchmark and contains 135 categories. In our experiments, we use a subset of the data collection which includes the 10 most frequent categories among the 135 topics and we call it Reuters-top 10.

**WebACE.** The WebACE dataset was from WebACE project and has been used for document clustering [17][5]. The WebACE dataset contains 2340 documents consisting news articles from Reuters new service via the Web in October 1997. These documents are divided into 20 classes.

**News4.** The News4 dataset used in our experiments are selected from the famous 20-newsgroups dataset<sup>5</sup>. The topic *rec* containing *autos*, *motorcycles*, *baseball* and *hockey* was selected from the version 20news-18828. The News4 dataset contains 3970 document vectors.

<sup>5</sup><http://people.csail.mit.edu/jrennie/20Newsgroups/>

To pre-process the datasets, we remove the stop words using a standard stop list, all HTML tags are skipped and all header fields except subject and organization of the posted articles are ignored. In all our experiments, we first select the top 1000 words by mutual information with class labels.

### 3.2 Evaluation Metrics

In the experiments, we set the number of clusters equal to the true number of classes  $C$  for all the clustering algorithms. To evaluate their performance, we compare the clusters generated by these algorithms with the true classes by computing the following two performance measures.

**Clustering Accuracy (Acc).** The first performance measure is the *Clustering Accuracy*, which discovers the one-to-one relationship between clusters and classes and measures the extent to which each cluster contained data points from the corresponding class. It sums up the whole matching degree between all pair class-clusters. Clustering accuracy can be computed as:

$$Acc = \frac{1}{N} \max \left( \sum_{\mathcal{C}_k, \mathcal{L}_m} T(\mathcal{C}_k, \mathcal{L}_m) \right), \quad (29)$$

where  $\mathcal{C}_k$  denotes the  $k$ -th cluster in the final results, and  $\mathcal{L}_m$  is the true  $m$ -th class.  $T(\mathcal{C}_k, \mathcal{L}_m)$  is the number of entities which belong to class  $m$  are assigned to cluster  $k$ . Accuracy computes the maximum sum of  $T(\mathcal{C}_k, \mathcal{L}_m)$  for all pairs of clusters and classes, and these pairs have no overlaps. The greater clustering accuracy means the better clustering performance.

**Normalized Mutual Information (NMI).** Another evaluation metric we adopt here is the *Normalized Mutual Information NMI* [23], which is widely used for determining the quality of clusters. For two random variable  $\mathbf{X}$  and  $\mathbf{Y}$ , the *NMI* is defined as:

$$NMI(\mathbf{X}, \mathbf{Y}) = \frac{I(\mathbf{X}, \mathbf{Y})}{\sqrt{H(\mathbf{X})H(\mathbf{Y})}}, \quad (30)$$

where  $I(\mathbf{X}, \mathbf{Y})$  is the mutual information between  $\mathbf{X}$  and  $\mathbf{Y}$ , while  $H(\mathbf{X})$  and  $H(\mathbf{Y})$  are the entropies of  $\mathbf{X}$  and  $\mathbf{Y}$  respectively. One can see that  $NMI(\mathbf{X}, \mathbf{X}) = 1$ , which is the maximal possible value of *NMI*. Given a clustering result, the *NMI* in Eq.(30) is estimated as

$$NMI = \frac{\sum_{k=1}^C \sum_{m=1}^C n_{k,m} \log \left( \frac{n \cdot n_{k,m}}{n_k \hat{n}_m} \right)}{\sqrt{\left( \sum_{k=1}^C n_k \log \frac{n_k}{n} \right) \left( \sum_{m=1}^C \hat{n}_m \log \frac{\hat{n}_m}{n} \right)}}, \quad (31)$$

where  $n_k$  denotes the number of data contained in the cluster  $\mathcal{C}_k$  ( $1 \leq k \leq C$ ),  $\hat{n}_m$  is the number of data belonging to the  $m$ -th class ( $1 \leq m \leq C$ ), and  $n_{k,m}$  denotes the number of data that are in the intersection between the cluster  $\mathcal{C}_k$  and the  $m$ -th class. The value calculated in Eq.(31) is used as a performance measure for the given clustering result. The larger this value, the better the clustering performance.

### 3.3 Comparisons

We have conducted comprehensive performance evaluations by testing our method and comparing it with 8 other representative data clustering methods using the same data corpora. The algorithms that we evaluated are listed below.

1. Traditional k-means (KM).

2. Spherical k-means (SKM). The implementation is based on [9].
3. Gaussian Mixture Model (GMM). The implementation is based on [16].
4. Spectral Clustering with Normalized Cuts (Ncut). The implementation is based on [26], and the variance of the Gaussian similarity is determined by *Local Scaling* [30]. Note that the criterion that Ncut aims to minimize is just the global regularizer in our *CLGR* algorithm except that Ncut used the normalized Laplacian.
5. Clustering using Pure Local Regularization (CPLR). In this method we just minimize  $\mathcal{J}_l$  (defined in Eq.(24)), and the clustering results can be obtained by doing eigenvalue decomposition on matrix  $(\mathbf{I} - \mathbf{P})^T(\mathbf{I} - \mathbf{P})$  with some proper discretization methods.
6. Adaptive Subspace Iteration (ASI). The implementation is based on [14].
7. Nonnegative Matrix Factorization (NMF). The implementation is based on [27].
8. Tri-Factorization Nonnegative Matrix Factorization (TNMF) [12]. The implementation is based on [15].

For computational efficiency, in the implementation of CPLR and our *CLGR* algorithm, we have set all the local regularization parameters  $\{\lambda_i\}_{i=1}^p$  to be identical, which is set by grid search from  $\{0.1, 1, 10\}$ . The size of the  $k$ -nearest neighborhoods is set by grid search from  $\{20, 40, 80\}$ . For the *CLGR* method, its global regularization parameter is set by grid search from  $\{0.1, 1, 10\}$ . When constructing the global regularizer, we have adopted the *local scaling* method [30] to construct the Laplacian matrix. The final discretization method adopted in these two methods is the same as in [26], since our experiments show that using such method can achieve better results than using *kmeans* based methods as in [20].

### 3.4 Experimental Results

The clustering accuracies comparison results are shown in table 3, and the normalized mutual information comparison results are summarized in table 4. From the two tables we mainly observe that:

1. Our *CLGR* method outperforms all other document clustering methods in most of the datasets;
2. For document clustering, the *Spherical k-means* method usually outperforms the traditional *k-means* clustering method, and the *GMM* method can achieve competitive results compared to the *Spherical k-means* method;
3. The results achieved from the *k-means* and *GMM* type algorithms are usually worse than the results achieved from *Spectral Clustering*. Since *Spectral Clustering* can be viewed as a weighted version of *kernel k-means*, it can obtain good results the data clusters are arbitrarily shaped. This corroborates that the documents vectors are not regularly distributed (spherical or elliptical).
4. The experimental comparisons empirically verify the equivalence between *NMF* and *Spectral Clustering*, which

**Table 3: Clustering accuracies of the various methods**

	CSTR	WebKB4	Reuters	WebACE	News4
KM	0.4256	0.3888	0.4448	0.4001	0.3527
SKM	0.4690	0.4318	0.5025	0.4458	0.3912
GMM	0.4487	0.4271	0.4897	0.4521	0.3844
NMF	0.5713	0.4418	0.4947	0.4761	0.4213
Ncut	0.5435	0.4521	0.4896	0.4513	0.4189
ASI	0.5621	0.4752	0.5235	0.4823	0.4335
TNMF	0.6040	0.4832	<b>0.5541</b>	0.5102	0.4613
CPLR	0.5974	0.5020	0.4832	0.5213	0.4890
CLGR	<b>0.6235</b>	<b>0.5228</b>	0.5341	<b>0.5376</b>	<b>0.5102</b>

**Table 4: Normalized mutual information results of the various methods**

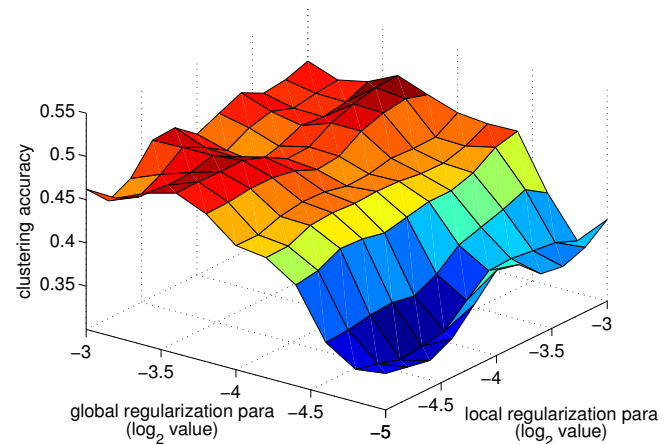
	CSTR	WebKB4	Reuters	WebACE	News4
KM	0.3675	0.3023	0.4012	0.3864	0.3318
SKM	0.4027	0.4155	0.4587	0.4003	0.4085
GMM	0.4034	0.4093	0.4356	0.4209	0.3994
NMF	0.5235	0.4517	0.4402	0.4359	0.4130
Ncut	0.4833	0.4497	0.4392	0.4289	0.4231
ASI	0.5008	0.4833	0.4769	0.4817	0.4503
TNMF	0.5724	0.5011	<b>0.5132</b>	0.5328	0.4749
CPLR	0.5695	0.5231	0.4402	<b>0.5543</b>	0.4690
CLGR	<b>0.6012</b>	<b>0.5434</b>	0.4935	0.5390	<b>0.4908</b>

has been proved theoretically in [10]. It can be observed from the tables that *NMF* and *Spectral Clustering* usually lead to similar clustering results.

- The co-clustering based methods (*TNMF* and *ASI*) can usually achieve better results than traditional purely document vector based methods. Since these methods perform an implicit feature selection at each iteration, provide an adaptive metric for measuring the neighborhood, and thus tend to yield better clustering results.
- The results achieved from *CPLR* are usually better than the results achieved from *Spectral Clustering*, which supports Vapnik's theory [24] that sometimes local learning algorithms can obtain better results than global learning algorithms.

Besides the above comparison experiments, we also test the parameter sensibility of our method. There are mainly two sets of parameters in our *CLGR* algorithm, the local and global regularization parameters ( $\{\lambda_i\}_{i=1}^n$  and  $\lambda$ , as we have said in section 3.3, we have set all  $\lambda_i$ 's to be identical to  $\lambda^*$  in our experiments), and the size of the neighborhoods. Therefore we have also done two sets of experiments:

- Fixing the size of the neighborhoods, and testing the clustering performance with varying  $\lambda^*$  and  $\lambda$ . In this set of experiments, we find that our *CLGR* algorithm can achieve good results when the two regularization parameters are neither too large nor too small. Typically our method can achieve good results when  $\lambda^*$  and  $\lambda$  are around 0.1. Figure 1 shows us such a testing example on the WebACE dataset.
- Fixing the local and global regularization parameters, and testing the clustering performance with different



**Figure 1: Parameter sensibility testing results on the WebACE dataset with the neighborhood size fixed to 20, and the x-axis and y-axis represents the  $\log_2$  value of  $\lambda^*$  and  $\lambda$ .**

sizes of neighborhoods. In this set of experiments, we find that the neighborhood with a too large or too small size will all deteriorate the final clustering results. This can be easily understood since when the neighborhood size is very small, then the data points used for training the local classifiers may not be sufficient; when the neighborhood size is very large, the trained classifiers will tend to be global and cannot capture the typical local characteristics. Figure 2 shows us a testing example on the WebACE dataset.

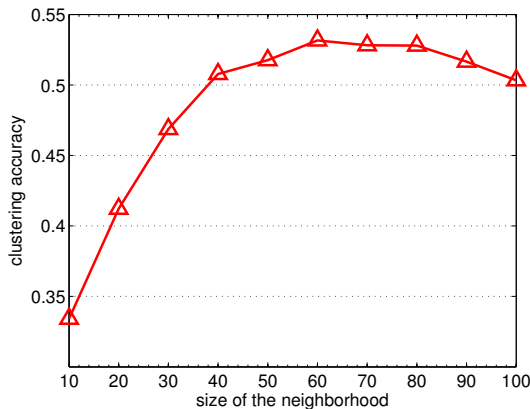
Therefore, we can see that our *CLGR* algorithm (1) can achieve satisfactory results and (2) is not very sensitive to the choice of parameters, which makes it practical in real world applications.

## 4. CONCLUSIONS AND FUTURE WORKS

In this paper, we derived a new clustering algorithm called *clustering with local and global regularization*. Our method preserves the merit of *local learning* algorithms and *spectral clustering*. Our experiments show that the proposed algorithm outperforms most of the state of the art algorithms on many benchmark datasets. In the future, we will focus on the parameter selection and acceleration issues of the *CLGR* algorithm.

## 5. REFERENCES

- [1] L. Baker and A. McCallum. Distributional Clustering of Words for Text Classification. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.
- [2] M. Belkin and P. Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15 (6):1373-1396. June 2003.
- [3] M. Belkin and P. Niyogi. Towards a Theoretical Foundation for Laplacian-Based Manifold Methods. In *Proceedings of the 18th Conference on Learning Theory (COLT)*. 2005.



**Figure 2: Parameter sensibility testing results on the WebACE dataset with the regularization parameters being fixed to 0.1, and the neighborhood size varying from 10 to 100.**

- [4] M. Belkin, P. Niyogi and V. Sindhwani. Manifold Regularization: a Geometric Framework for Learning from Examples. *Journal of Machine Learning Research* 7, 1-48, 2006.
- [5] D. Boley. Principal Direction Divisive Partitioning. *Data mining and knowledge discovery*, 2:325-344, 1998.
- [6] L. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4:888-900, 1992.
- [7] P. K. Chan, D. F. Schlag and J. Y. Zien. Spectral K-way Ratio-Cut Partitioning and Clustering. *IEEE Trans. Computer-Aided Design*, 13:1088-1096, Sep. 1994.
- [8] D. R. Cutting, D. R. Karger, J. O. Pederson and J. W. Tukey. Scatter/Gather: A Cluster-Based Approach to Browsing Large Document Collections. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1992.
- [9] I. S. Dhillon and D. S. Modha. Concept Decompositions for Large Sparse Text Data using Clustering. *Machine Learning*, vol. 42(1), pages 143-175, January 2001.
- [10] C. Ding, X. He, and H. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the SIAM Data Mining Conference*, 2005.
- [11] C. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proc. of the 1st International Conference on Data Mining (ICDM)*, pages 107-114, 2001.
- [12] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal Nonnegative Matrix Tri-Factorizations for Clustering. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.
- [13] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, Inc., 2001.
- [14] T. Li, S. Ma, and M. Ogihara. Document Clustering via Adaptive Subspace Iteration. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004.
- [15] T. Li and C. Ding. The Relationships Among Various Nonnegative Matrix Factorization Methods for Clustering. In *Proceedings of the 6th International Conference on Data Mining (ICDM)*. 2006.
- [16] X. Liu and Y. Gong. Document Clustering with Cluster Refinement and Model Selection Capabilities. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002.
- [17] E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. WebACE: A Web Agent for Document Categorization and Exploration. In *Proceedings of the 2nd International Conference on Autonomous Agents (Agents98)*. ACM Press, 1998.
- [18] M. Hein, J. Y. Audibert, and U. von Luxburg. From Graphs to Manifolds - Weak and Strong Pointwise Consistency of Graph Laplacians. In *Proceedings of the 18th Conference on Learning Theory (COLT)*, 470-485. 2005.
- [19] J. He, M. Lan, C.-L. Tan, S.-Y. Sung, and H.-B. Low. Initialization of Cluster Refinement Algorithms: A Review and Comparative Study. In *Proc. of Inter. Joint Conference on Neural Networks*, 2004.
- [20] A. Y. Ng, M. I. Jordan, Y. Weiss. On Spectral Clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*. 2002.
- [21] B. Schölkopf and A. Smola. *Learning with Kernels*. The MIT Press. Cambridge, Massachusetts. 2002.
- [22] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888-905, 2000.
- [23] A. Strehl and J. Ghosh. Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research*, 3:583-617, 2002.
- [24] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Berlin: Springer-Verlag, 1995.
- [25] Wu, M. and Schölkopf, B. A Local Learning Approach for Clustering. In *Advances in Neural Information Processing Systems 18*. 2006.
- [26] S. X. Yu, J. Shi. Multiclass Spectral Clustering. In *Proceedings of the International Conference on Computer Vision*, 2003.
- [27] W. Xu, X. Liu and Y. Gong. Document Clustering Based On Non-Negative Matrix Factorization. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003.
- [28] H. Zha, X. He, C. Ding, M. Gu and H. Simon. Spectral Relaxation for K-means Clustering. In *NIPS 14*. 2001.
- [29] T. Zhang and F. J. Oles. Text Categorization Based on Regularized Linear Classification Methods. *Journal of Information Retrieval*, 4:5-31, 2001.
- [30] L. Zelnik-Manor and P. Perona. Self-Tuning Spectral Clustering. In *NIPS 17*. 2005.
- [31] D. Zhou, O. Bousquet, T. N. Lal, J. Weston and B. Schölkopf. Learning with Local and Global Consistency. *NIPS 17*, 2005.