Clustering with Local and Global Regularization

Fei Wang[1], **Changshui Zhang**[1], **Tao Li**[2]
[1]State Key Laboratory of Intelligent Technologies and Systems
Department of Automation, Tsinghua University, Beijing, China. 100084.
[2]School of Computer Science, Florida International University, Miami, FL 33199, U.S.A.

Abstract

Clustering is an old research topic in data mining and machine learning communities. Most of the traditional clustering methods can be categorized local or global ones. In this paper, a novel clustering method that can explore both the local and global information in the dataset is proposed. The method, *Clustering with Local and Global Consistency* (*CLGR*), aims to minimize a cost function that properly trades off the local and global costs. We will show that such an optimization problem can be solved by the eigenvalue decomposition of a sparse symmetric matrix, which can be done efficiently by some iterative methods. Finally the experimental results on several datasets are presented to show the effectiveness of our method.

Introduction

Clustering (Jain & Dubes, 1988) is one of the most fundamental research topics in both data mining and machine learning communities. It aims to divide data into groups of similar objects, *i.e. clusters*. From a machine learning perspective, what clustering does is to learn the *hidden patterns* of the dataset in an unsupervised way, and these patterns are usually referred to as *data concepts*. From a practical perspective clustering plays an outstanding role in data mining applications such as scientific information retrieval and text mining, Web analysis, marketing, computational biology, and many others (Han & Kamber, 2001).

Many clustering methods have been proposed till now, among which *K-means* (Duda *et al*, 2001) is one of the most famous algorithms, it aims to minimize the sum of the squared distance between the data points and their corresponding cluster centers. However, it is well known that there are some problems existing in the *K-means* algorithm: (1) the predefined criterion is usually non-convex which causes many local optimal solutions; (2) the iterative procedure (*e.g.* the for optimizing the criterion usually makes the final solutions heavily depend on the initializations. In the last decades, many methods (He *et al*, 2004; Zha *et al*, 2001) have been proposed to overcome the above problems.

Recently, another type of methods, which are 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* (Shi & Malik, 2000), *Ratio Cut* (Chan *et al*, 1994), *Min-Max Cut* (Ding *et al*, 2001)) defined on an undirected graph. After some relaxations, these criterions can usually be optimized via eigendecompositions, which is guaranteed to be global optimal. In this way, spectral clustering efficiently avoids the problems of the traditional *K-means* method.

In this paper, we propose a novel clustering algorithm that inherits the superiority of spectral clustering, *i.e.* the final cluster results can also be obtained by exploit the eigenstructure of a symmetric matrix. However, unlike spectral clustering, which just enforces a smoothness constraint on the data labels over the whole data manifold (Belkin & Niyogi, 2003), our method first construct a regularized linear label predictor for each data point from its neighborhood, 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 (Zhou *et al*, 2004), 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.

The Proposed Algorithm

In this section, we will introduce our *Clustering with Local and Global Regularization* (*CLGR*) algorithm in detail. First let's introduce the notations and problem statement.

Table 1: Frequently used notations

n	The total number of data
$\mathbf{X}$	The data matrix, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$
$\mathcal{N}_i$	The neighborhood of $\mathbf{x}_i$
n_i	The cardinality of $\mathcal{N}_i$
$\mathbf{X}_i$	The matrix composed of $\mathcal{N}_i$
$\mathbf{L}$	The graph Laplacian constructed on $\mathcal{X}$

Notations and Problem Statement

In a clustering problem, we are given n data points, $\mathbf{x}_1, \cdots, \mathbf{x}_n$, and a positive integer C. The goal is to partition the given dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ ($\mathbf{x}_i \in \mathbb{R}^m$) into C clusters, such that different clusters are in some sense "distinct" from each other.

Mathematically, the result of a clustering algorithm can be represented by a *cluster assignment indication matrix* $\mathbf{P}_{n\times C}$, such that $\mathbf{P}_{ij} = 1$ if $\mathbf{x}_i$ belongs to cluster j, and $\mathbf{P}_{ij} = 0$ otherwise. That is, there is only one 1 for each row of matrix $\mathbf{P}$, and the rest of the elements are all zero.

The same as in (Yu & Shi, 2003), we will not solve for the matrix $\mathbf{P}$ directly. What we will solve in this paper is a scaled *cluster assignment indication matrix* $\mathbf{Q}_{n\times C}$, such that $\mathbf{Q}_{ij} = \mathbf{P}_{ij}/\sqrt{n_j}$, then

$$\mathbf{Q} = \mathbf{P}\left(\mathbf{P}^T\mathbf{P}\right)^{-1/2}. \quad (1)$$

Therefore $\mathbf{Q}$ is a *semi-orthogonal matrix* in that

$$\mathbf{Q}^T\mathbf{Q} = \left(\mathbf{P}^T\mathbf{P}\right)^{-1/2}\left(\mathbf{P}^T\mathbf{P}\right)\left(\mathbf{P}^T\mathbf{P}\right)^{-1/2} = \mathbf{I}, \quad (2)$$

where $\mathbf{I}$ is an $n \times n$ identity matrix. In the following we will write $\mathbf{Q}$ as

$$\mathbf{Q} = \left[\mathbf{q}^1, \mathbf{q}^2, \cdots, \mathbf{q}^C\right], \quad (3)$$

where $\mathbf{q}^i$ ($1 \leqslant i \leqslant C$) corresponds to the i-th row of $\mathbf{Q}$, and q_{ij} can be regarded as the *confidence* that $\mathbf{x}_i$ belongs to cluster j.

Table 1 shows some symbols and notations that will be frequently used throughout the paper.

Regularized Linear Classifier Revisited

The traditional machine learning methods can be categorized into two main types: *supervised learning* and *unsupervised learning*. For unsupervised learning, what we face with are a data set with no labels and our goal is to organize them in a reasonable way (such as *clustering*), while supervised learning can be posed as a problem of function estimation, in which we aims to get a *good* classification function from the labeled training data set that can predict the labels for the unseen testing data set with some cost minimized (Vapnik, 1995). The linear classifier with least square fit is one of the simplest supervised learning methods, which aims to learn a column vector $\mathbf{w}$ such that the squared cost

$$\mathcal{J}' = \frac{1}{n}\sum_i (\mathbf{w}^T\mathbf{x}_i - y_i)^2 \quad (4)$$

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

$$\mathbf{w}^* = \left(\mathbf{X}\mathbf{X}^T\right)^{-1}\mathbf{X}\mathbf{y}, \quad (5)$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ is an $m \times n$ *data matrix*, $\mathbf{y} = [y_1, y_2, \cdots, y_n]^T$ is the *label vector*. For the two-class problem, $y_i \in \{+1, -1\}$, and we can determine the label of a test point $\mathbf{x}_u$ by

$$l = sign(\mathbf{w}^{*T}\mathbf{x}_u), \quad (6)$$

where $sign(\cdot)$ is the sign function. For the multi-class (say C-class) problem, we can adopt a similar way as we have introduced in last subsection, *i.e.* we can construct one classifier for each class by minimizing

$$\mathcal{J}^{c'} = \frac{1}{n}\sum_i \left((\mathbf{w}^c)^T\mathbf{x}_i - (\mathbf{y}^c)_i\right)^2, \quad (7)$$

where $1 \leqslant c \leqslant C$, $(y^c)_i = 1$ if $\mathbf{x}_i$ belongs to class c, $(\mathbf{y}^c)_i = 0$ otherwise. Then the normal vector for the classifier of the c-th class becomes

$$\mathbf{w}^{c*} = \left(\mathbf{X}\mathbf{X}^T\right)^{-1}\mathbf{X}\mathbf{y}^c, \quad (8)$$

and the label of a test point $\mathbf{x}_u$ can be determined by

$$c = argmax_c\left((\mathbf{w}^{c*})^T\mathbf{x}_u\right) \quad (9)$$

To avoid the singularity of $\mathbf{X}\mathbf{X}^T$(*e.g.* when $m \gg n$), we can add a regularization term and minimize the following criterion for the c-th class

$$\mathcal{J}^c = \frac{1}{n}\sum_{i=1}^n \left((\mathbf{w}^c)^T\mathbf{x}_i - y_i\right)^2 + \lambda_c\|\mathbf{w}^c\|^2, \quad (10)$$

where λ_c is a regularization parameter. Then the optimal solution that minimize $\mathcal{J}'_c$ becomes

$$\mathbf{w}^{c*} = \left(\mathbf{X}\mathbf{X}^T + \lambda n\mathbf{I}\right)^{-1}\mathbf{X}\mathbf{y}^c, \quad (11)$$

where $\mathbf{I}$ is an $m \times m$ identity matrix. This is what we usually called *regularized linear classifier*.

Like most of the supervised learning methods (*e.g.* SVM, decision trees), regularized linear classifier is one kind of global classifiers, *i.e.* it uses the whole training set for training the classifier. However, as pointed out by (Vapnik, 1995), sometimes it may be hard to find a classifier that is good enough for predicting the labels of the whole input space. In order to get better predictions, (Bottou & Vapnik, 1992) found that for certain tasks, locally trained classifiers could get better performances for predicting the labels of the test data.

Local Regularization

Inspired by the work of (Bottou & Vapnik, 1992) and (Wu & Schölkopf, 2006), we applied the local learning algorithms for clustering. The basic idea is that, we train a local label predictor for each data point $\mathbf{x}_i$ ($1 \leqslant i \leqslant n$) based on its neighborhood $\mathcal{N}_i$ (*k-nearest neighborhood* or *ε neighborhood*), and use it to predict the label of $\mathbf{x}_i$. Then all these local predictors will be combined together by minimizing the sum of their prediction errors.

Due to its simplicity and effectiveness, we choose the regularized linear classifier as our local label predictor, *i.e.* for each datum $\mathbf{x}_i$, we aims to get a $\mathbf{w}_i$ that minimizes

$$\mathcal{J}_i^c = \frac{1}{n_i} \sum_{\mathbf{x}_j \in \mathcal{N}_i} \| \left(\mathbf{w}_i^c\right)^T \mathbf{x}_j - (\mathbf{q}^c)_j \|^2 + \lambda_i \|\mathbf{w}_i^c\|^2, \quad (12)$$

where $n_i = |\mathcal{N}_i|$ is the cardinality of $\mathcal{N}_i$, and $(\mathbf{q}^c)_j$ is the confidence that $\mathbf{x}_j$ belongs to cluster c. From Eq.(11) we can get the optimal solution

$$\mathbf{w}_i^{c*} = \left(\mathbf{X}_i \mathbf{X}_i^T + \lambda_i n_i \mathbf{I}\right)^{-1} \mathbf{X}_i \mathbf{q}_i^c \ (1 \leqslant c \leqslant C), \quad (13)$$

where $\mathbf{X}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \cdots, \mathbf{x}_{in_i}]$ with $\mathbf{x}_{ik}$ being the k-th neighbor of $\mathbf{x}_i$, and $\mathbf{q}_i^c = [q_{i1}^c, q_{i2}^c, \cdots, q_{in_i}^c]^T$ with $q_{ik}^c = q^c(\mathbf{x}_{ik})$. It can be easily shown that Eq.(13) can be further transformed to

$$\mathbf{w}_i^{c*} = \mathbf{X}_i \left(\mathbf{X}_i^T \mathbf{X}_i + \lambda_i n_i \mathbf{I}_i\right)^{-1} \mathbf{q}_i^c \ (1 \leqslant c \leqslant C), \quad (14)$$

Then for a new testing point $\mathbf{u}$ that falls into $\mathcal{N}_i$, we can predict the confidence of it belonging to class c by

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

Note that the above expression can be easily kernelized (Schölkopf & Smola, 2002) as in (Wu & Schölkopf, 2006) since it only involves the computation of inner products.

After all the local predictors having been constructed, we will combine them together by minimizing the sum of their prediction errors

$$\mathcal{J}_l = \sum_{c=1}^{C} \sum_{i=1}^{n} \left(\left(\mathbf{w}_i^{c*}\right)^T \mathbf{x}_i - q_i^c \right)^2. \quad (15)$$

Combining Eq.(15) and Eq.(11), we can get

$$\begin{aligned} \mathcal{J}_l &= \sum_{c=1}^{C} \sum_{i=1}^{n} \left(\left(\mathbf{w}_i^{c*}\right)^T \mathbf{x}_i - q_i \right)^2 \\ &= \sum_{c=1}^{C} \sum_{i=1}^{n} \left(\mathbf{x}_i^T \mathbf{X}_i \left(\mathbf{X}_i^T \mathbf{X}_i + \lambda_i n_i \mathbf{I}_i\right)^{-1} \mathbf{q}_i^c - q_i^c \right)^2 \\ &= \sum_{c=1}^{C} \|\mathbf{G}\mathbf{q}^c - \mathbf{q}^c\|^2, \end{aligned} \quad (16)$$

where $\mathbf{q}^c = [q_1^c, q_2^c, \cdots, q_n^c]^T$, and the $\mathbf{G}$ is an $n \times n$ matrix with its (i,j)-th entry

$$\mathbf{G}_{ij} = \begin{cases} \boldsymbol{\alpha}_j^i, & if\ \mathbf{x}_j \in \mathcal{N}_i \\ 0, & otherwise \end{cases}, \quad (17)$$

where $\boldsymbol{\alpha}_j^i$ represents the j-th entry of

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

One may argue that the local approach we used here is similar to *Locally Linear Embedding*(Roweis & Saul, 2000), which assumes that each data point can be linearly reconstructed from its neighborhood. More concretely, for each data point $\mathbf{x}_i$, it minimizes

$$\begin{aligned} \varepsilon_i &= \left\| \mathbf{x}_i - \sum\nolimits_{\mathbf{x}_i \in \mathcal{N}(\mathbf{x}_i)} w_{ij} \mathbf{x}_i \right\|^2 \\ s.t. &\quad \sum\nolimits_j w_{ij} = 1. \end{aligned} \quad (18)$$

Comparing ε_i with the local loss $\mathcal{J}_i$ shown in Eq.(12), we can find that the *LLE* approach focuses on linear relationships from pure neighborhood points, and no label information is needed, while in our local regularization step we aims to construct linear classifiers from the neighborhood points. Therefore it is conceptually different from our local regularization method.

Till now we construct all the locally regularized linear label predictors and combine them in a cost function that can be written in an explicit mathematical form, which can be efficiently minimized directly using 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.

Global Regularization

A common assumption that can guide the learning process is the *cluster assumption* (Zhou *et al*, 2004), which states

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.

In other words, the cluster assumption implies that the labels of the data set should vary smoothly with respect to the intrinsic data structure. According to (Belkin & Niyogi, 2003), the smoothness of the data label (or cluster assignment) vector $\mathbf{q}$ can be measured by

$$\mathcal{J}_g = \sum_{c=1}^{C} (\mathbf{q}^c)^T \mathbf{L} \mathbf{q}^c = \sum_{c=1}^{C} \sum_{i=1}^{n} (q_i^c - q_j^c)^2 w_{ij}, \quad (19)$$

where $\mathbf{L}$ is an $n \times n$ matrix with its (i,j)-th entry

$$L_{ij} = \begin{cases} d_i - w_{ii}, & if\ i = j \\ -w_{ij}, & otherwise \end{cases}, \quad (20)$$

$d_i = \sum_j w_{ij}$, and w_{ij} is the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. There have been many methods to compute w_{ij}, some of the representative ones are listed below

1. *Unweighted k-Nearest Neighborhood Similarity* (Belkin & Niyogi, 2004): The similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ is 1 if $\mathbf{x}_i$ is in the k-nearest neighborhood of $\mathbf{x}_j$ or $\mathbf{x}_j$ is in the k-nearest neighborhood of $\mathbf{x}_i$, and 0 otherwise. k is the only hyperparameter that controls this similarity. As noted by (Zhu *et al*, 2003), this similarity has the nice property of "adaptive scales", since the similarities between pairwise points are the same in low and high density regions.
2. *Unweighted ϵ-Ball Neighborhood Similarity*: The similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ is 1 if for some distance function $d(\cdot)$, $d(\mathbf{x}_i, \mathbf{x}_j) \leqslant \epsilon$. ϵ is the only hyperparameter controlling this similarity, which is continuous.
3. *Weighted* $\tanh$ *Similarity* (Zhu *et al*, 2003): Let d_{ij} be the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, then the tanh similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ can be computed by

$$w_{ij} = \frac{1}{2}(\tanh(\alpha_1(d_{ij} - \alpha_2)) + 1)$$

The intuition is to create a soft cutoff around length α_2, so that similar examples (presumably from the same class) have higher similarities and dissimilar examples (presumably from different classes) have lower similarities. The hyperparameters α_1 and α_2 controls the slope and cutoff values of the tanh similarity respectively.

4. *Weighted Exponential Similarity* (Shi & Malik, 2000; Belkin & Niyogi, 2003; Zhu *et al*, 2003): Let d_{ij} be the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, then the tanh similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ can be computed by

$$w_{ij} = \exp\left(-\frac{d_{ij}^2}{\sigma}\right), \tag{21}$$

which is also a continuous weighting scheme with σ controlling the decay rate.

In this paper we have preferred the weighted exponential similarity because (1) it is simple and has widely been applied in many fields; (2) 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 (Belkin & Niyogi, 2005; Hein *et al*., 2005), and the *Euclidean distance* is selected as the method for computing d_{ij}.

Clustering with Local and Global Regularization

Combining the local and global regularization criterions introduced above, we can derive the clustering criterion as

$$\begin{aligned} \min_q \quad & \mathcal{J} = \mathcal{J}_l + \lambda\mathcal{J}_g = \sum_{c=1}^{C}\left(\|\mathbf{G}\mathbf{q}^c - \mathbf{q}^c\|^2 + \lambda(\mathbf{q}^c)^T\mathbf{L}\mathbf{q}^c\right) \\ s.t. \quad & \mathbf{Q}^T\mathbf{Q} = \mathbf{I}, \end{aligned} \tag{22}$$

where $\mathbf{G}$ is defined as in Eq.(17), and λ is a positive real-valued parameter to tradeoff $\mathcal{J}_l$ and $\mathcal{J}_g$, $\mathbf{Q} = \left[\mathbf{q}^1, \mathbf{q}^2, \cdots, \mathbf{q}^C\right]$. Note that we have relaxed the constraints on $\mathbf{Q}$ such that it only needs to satisfy the semi-orthogonality constraint. Then the objective that we aims to minimize becomes

$$\begin{aligned} \mathcal{J} &= \mathcal{J}_l + \lambda\mathcal{J}_g \\ &= \sum_{c=1}^{C}\left[\|\mathbf{G}\mathbf{q}^c - \mathbf{q}^c\|^2 + \lambda(\mathbf{q}^c)^T\mathbf{L}\mathbf{q}^c\right] \\ &= \sum_{c=1}^{C}\left[(\mathbf{q}^c)^T\left((\mathbf{G}-\mathbf{I})^T(\mathbf{G}-\mathbf{I}) + \lambda\mathbf{L}\right)\mathbf{q}^c\right] \\ &= trace\left[\mathbf{Q}^T\left((\mathbf{G}-\mathbf{I})^T(\mathbf{G}-\mathbf{I}) + \lambda\mathbf{L}\right)\mathbf{Q}\right], \end{aligned} \tag{23}$$

Therefore we should solve the following optimization problem

$$\begin{aligned} \min_{\mathbf{Q}} \quad & \mathcal{J} = trace\left[\mathbf{Q}^T\left((\mathbf{G}-\mathbf{I})^T(\mathbf{G}-\mathbf{I}) + \lambda\mathbf{L}\right)\mathbf{Q}\right] \\ s.t. \quad & \mathbf{Q}^T\mathbf{Q} = \mathbf{I}, \end{aligned} \tag{24}$$

From the *Ky Fan* theorem (Zha *et al*, 2001), we know the optimal solution of the above problem is

$$\mathbf{Q}^* = [\mathbf{q}_1^*, \mathbf{q}_2^*, \cdots, \mathbf{q}_C^*]\mathbf{R}, \tag{25}$$

Table 2: Clustering with Local and Global Regularization

Input:
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 λ;

Output:
The cluster membership of each data point.

Procedure:
1. Construct the K *nearest neighborhoods* for each data point;
2. Construct the matrix $\mathbf{P}$ using Eq.(17);
3. Construct the Laplacian matrix $\mathbf{L}$ using Eq.(20);
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.(25);
6. Output the cluster assignments of each data point by properly discretize $\mathbf{Q}^*$.

where $\mathbf{q}_k^*$ $(1 \leqslant k \leqslant C)$ is the eigenvector corresponds to the k-th smallest eigenvalue of matrix $(\mathbf{G}-\mathbf{I})^T(\mathbf{G}-\mathbf{I}) + \lambda\mathbf{L}$, and $\mathbf{R}$ is an arbitrary $C \times C$ matrix. Hence the optimal solution to the above optimization problem is not unique, it is a subspace of matrices which is usually referred to as *Grassman manifold*. Then what we should really find is a *scaled cluster assignment indication* matrix $\mathbf{Q}^*$ together with a *rotation matrix* $\mathbf{R}$ such that $\mathbf{Q}^*\mathbf{R}$ is close to a true *discrete* scaled cluster assignment indication matrix, in that way, the resultant *cluster assignment matrix* $\mathbf{P}$ will be close to the true *discrete* cluster assignment indication matrix. To achieve this goal, we will adopt the method proposed in (Yu & Shi, 2003) in our experiments.

From another point of view, what *CLGR* do is just clustering with a hybrid of different types of regularizations. The feasibility of such kind of methods has been discussed by (Zhu & Goldberg, 2007) and attempted by (Chapelle *et al*, 2006) in the semi-supervised learning fields. However, as far as we know there is little work towards such direction in the unsupervised learning field.

The algorithm flowchart of *CLGR* is summarized in table 2.

Experiments

In this section, experiments are conducted to empirically compare the clustering results of *CLGR* with some other clustering algorithms on 4 datasets. First we will briefly introduce the basic information of those datasets.

Datasets

We use four real world datasets to evaluate the performances of the methods. Table 3 summarizes the characteristics of the datasets.

The *UMIST* dataset contains the face images of 20 different persons. The *USPS* dataset contains a subset of the famous *USPS* handwritten digits dataset, which contains the

Table 3: Descriptions of the datasets

Datasets	Sizes	Classes	Dimensions
UMIST	575	20	1024
USPS	3874	4	256
Newsgroup	3970	4	1000
WebACE	2340	20	1000

image samples of digits 1,2,3,4. The *Newsgroup* dataset is the classes *autos*, *motorcycles*, *baseball* and *hockey* of the Newsgroup20 dataset and the *WebACE* dataset contains 2340 documents consisting news articles from Reuters new service via the Web in October 1997. For the last two text datasets, we have selected the top 1000 words by mutual information with class labels.

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:

$$Acc = \frac{1}{N} \max \left(\sum_{\mathcal{C}_k, \mathcal{L}_m} T(\mathcal{C}_k, \mathcal{L}_m) \right), \qquad (26)$$

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***). An other evaluation metric we adopt here is the *Normalized Mutual Information NMI* (Strehl & Ghosh, 2002), which is widely used for determining the quality of clusters. For two random variable $\mathbf{X}$ and $\mathbf{Y}$, the *NMI* is defined as:

$$NMI(\mathbf{X}, \mathbf{Y}) = \frac{I(\mathbf{X}, \mathbf{Y})}{\sqrt{H(\mathbf{X})H(\mathbf{Y})}}, \qquad (27)$$

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 X and 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.(27) is estimated as

$$NMI = \frac{\sum_{k=1}^{C} \sum_{m=1}^{C} n_{k,m} log\left(\frac{n \cdot n_{k,m}}{n_k \hat{n}_m}\right)}{\sqrt{\left(\sum_{k=1}^{C} n_k log\frac{n_k}{n}\right)\left(\sum_{m=1}^{C} \hat{n}_m log\frac{\hat{n}_m}{n}\right)}}, \qquad (28)$$

where n_k denotes the number of data contained in the cluster $\mathcal{C}_k$ ($1 \leqslant k \leqslant C$), $\hat{n}_m$ is the number of data belonging to the m-th class ($1 \leqslant m \leqslant C$), 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.(28) is used as a performance measure for the given clustering result. The larger this value, the better the clustering performance.

Table 4: Clustering accuracy results

	UMIST	USPS	Newsgroup	WebACE
KM	0.4365	0.7423	0.3228	0.3120
SC	0.6433	0.9342	0.5235	0.4561
CPLR	0.6897	0.9330	0.5425	0.5531
CLGR	**0.7124**	**0.9553**	**0.5796**	**0.5831**

Table 5: Normalized mutual information results

	UMIST	USPS	Newsgroup	WebACE
KM	0.6479	0.8523	0.2014	0.1445
SC	0.7620	0.9716	0.4978	0.3887
CPLR	0.7963	0.9649	0.5012	0.4776
CLGR	**0.8003**	**0.9801**	**0.5231**	**0.5074**

Comparisons and Parameter Settings

We have compared the performances of our method with three other clustering approaches, namely *K-means* (*KM*), *Spectral Clustering* (*SC*) (Shi & Malik, 2000), and *Clustering with Pure Local Regularization* (*CPLR*), *i.e.*, clustering just by minimize $\mathcal{J}_l$ in Eq.(15).

For *CLGR* and *SC*, the weights on data graph edges are computed by Gaussian functions, and the variance of which is determined by *local scaling*(Zelnik-Manor & Perona, 2005). All local regularization parameters $\{\lambda_i\}_{i=1}^{n}$ are set to the same in *CPLR* and *CLGR*, which is determined by searching the grid $\{0.1, 1, 10\}$, and the neighborhood size is set by searching the grid $\{20, 40, 80\}$. The global regularization parameter λ in *CLGR* is set by searching the grid $\{0.1, 1, 10\}$. For *SC, CPLR, CLGR*, we adopt the same discretization method as in (Yu & Shi, 2003) since it shows better empirical results.

Experimental Results

The final clustering results are shown in table 4 and table 5, from which we can see that *CLGR* outperforms all other three clustering methods on these four datasets, which supports the assertion that combining both local and global information in clustering can improve the clustering results.

Conclusions

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 some 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.

Acknowledgement

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.

References

Belkin, M. and Niyogi, P. (2003) Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15 (6):1373-1396.

Belkin, M., Niyogi, P. (2004). Semi-Supervised Learning on Riemannian Manifolds. *Machine Learning* 56: 209-239.

Belkin, M. and Niyogi, P. (2005). Towards a Theoretical Foundation for Laplacian-Based Manifold Methods. In *Proceedings of the 18th Conference on Learning Theory* (*COLT*).

Bottou, L. and Vapnik, V. (1992). Local learning algorithms. *Neural Computation*, 4:888-900.

Chan, P. K., Schlag, D. F. and Zien, J. Y. (1994). Spectral K-way Ratio-Cut Partitioning and Clustering. *IEEE Trans. Computer-Aided Design*, 13:1088-1096.

Chapelle, O., Chi, M. and Zien, A. (2006). A Continuation Method for Semi-Supervised SVMs. *Proceedings of the 23rd International Conference on Machine Learning*, 185-192.

Ding, C., He, X., Zha, H., Gu, M., and Simon, H. D. (2001). A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of the 1st International Conference on Data Mining* (*ICDM*), pages 107-114.

Duda, R. O., Hart, P. E. and Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons, Inc.

Han, J. and Kamber, M. (2001). *Data Mining*. Morgan Kaufmann Publishers.

Hein, M., Audibert, J. Y. and Luxburg, U. von. (2005). From Graphs to Manifolds - Weak and Strong Pointwise Consistency of Graph Laplacians. In *Proceedings of the 18th Conference on Learning Theory* (*COLT*), 470-485.

He, J., Lan, M., Tan, C.-L., Sung, S.-Y., and Low. H.-B. (2004). Initialization of Cluster Refinement Algorithms: A Review and Comparative Study. In *Proceedings of International Joint Conference on Neural Networks*.

Jain, A. and Dubes, R. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs, NJ.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002). On Spectral Clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*.

Roweis, S. T. and Saul, L. K. (2000). Noninear Dimensionality Reduction by Locally Linear Embedding. *Science*: vol. 290, 2323-2326.

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels*. The MIT Press. Cambridge, Massachusetts.

Shi, J. and Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888-905.

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

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Berlin: Springer-Verlag.

Wang, F., Zhang, C., and Li, T. (2007). Regularized Clustering for Documents. In *Proceedings of the ACM SIGIR 2007 Conference*.

Wu, M. and Schölkopf, B. (2006). A Local Learning Approach for Clustering. In *Advances in Neural Information Processing Systems 18*.

Yu, S. X., and Shi, J. (2003). Multiclass Spectral Clustering. In *Proceedings of the International Conference on Computer Vision*.

Zelnik-Manor, L. and Perona, P. (2005). Self-Tuning Spectral Clustering. In *Advances in Neural Information Processing Systems 17*.

Zha, H., He, X., Ding, C., Gu, M. and Simon, H. (2001). Spectral Relaxation for K-means Clustering. In *Advances in Neural Information Processing Systems 14*.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J. and Schölkopf, B. (2004). Learning with Local and Global Consistency. *Advances in Neural Information Processing Systems 16*.

Zhu, X., Lafferty, J. and Ghahramani, Z. (2003). Semi-Supervised Learning: From Gaussian Fields to Gaussian Process. *Computer Science Technical Report*, Carnegie Mellon University, CMU-CS-03-175.

Zhu, X. and Goldberg, A. (2007). Kernel Regression with Order Preferences. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI)*.