Multi-view Latent Subspace Clustering based on both Global and Local Structure

Honghan Zhou  
Weiling Cai  
Le Xu  
Ming Yang  
HHANZHOU@OUTLOOK.COM  
CAIWL@NJNU.EDU.CN  
NJNUMASTERXU91@163.COM  
MYANG@NJNU.EDU.CN  

School of Computer and Electronic Information Nanjing Normal University Nanjing, China

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

Most existing multi-view clustering methods focus on the global structure or local structure among samples, and few methods focus on the two structures at the same time. In this paper, we propose a Multi-view Latent subspace Clustering based on both Global and Local structure (MLCGL). In this method, a latent embedding representation is learned by exploring the complementary information from different views. In the latent space, not only the global reconstruction relationship but also the local geometric structure among the latent variables are discovered. In this way, a unified affinity graph matrix is constructed in the latent space for different views, which indicates a clear between-class relationship. Meanwhile, a rank constraint is introduced on the Laplacian graph to facilitate the division of samples into the required clusters. In MLCGL, the affinity graph also provides positive feedback to optimize the learned latent representation and contribute to divided it into reasonable clusters. Moreover, we present an alternating iterative optimization scheme to optimize objective functions. Compared with the state-of-art algorithms, MLCGL has achieved excellent experimental performance on several real-world datasets.

Keywords: multi-view, latent represent, global and local structure, rank constraint

1. Introduction

With the continuous development of science and technology, the same data is usually comprised of several different views (Kang et al. (2020); Huang et al. (2020); Lin et al. (2021)). This is referred to as multi-view data, where each individual view constitutes a given task but each view also has its biases. The multi-view learning technology aims to integrate complementary information from different views to reduce the complexity of given task. And how to effectively use multi-view data has become a critical challenge faced by researchers.

Clustering which assigns the data into different groups with similarity, is a fundamental and important topic in machine learning. For multi-view scenarios, the traditional clustering algorithms usually process the view separately and choose the best result, or connect the features of all views as single-view data. But these algorithms fail to utilize the complementary information of different views and may cause excessively high dimensions of the data to be processed. To solve these problems, Kumar et al. (2011) introduced spectral clustering into multiple views application, which co-regularized the clustering indicator to achieve the same cluster membership on different views. And Peng and Cai (2021) captured the com-
compatible intrinsic information across different views through low-rank sparse representation. Besides, (Xia et al. (2014); Zong et al. (2018); Nie et al. (2018)) explored the complementarity between views from the three aspects of Markov Chain, Spectral Perturbation, and Procrustes Analysis, which enhanced the finally clustering performance.

How to explore the similarity relationships among samples is also a research hotspot in multi-view clustering. In recent years, Huang et al. (2015) has pointed out that the similarity among samples can be represented in the form of global relationship or local relationship. The global relationship can be obtained by various manners, such as calculating the distance between every two samples Liu et al. (2013), computing the self-representation coefficient on training samples Ren and Sun (2020). The local relationship can be obtained by computing k-nearest neighbors, the local geometric structure, and so on. Some subspace clustering algorithms like Brbić and Kopriva (2018) perform the clustering by exploiting the global similarity among samples, it constructs the self-representation coefficient matrix of each view to indicate the global structure, and then introduces a regularization term to balance the global structure discrepancy between views. By contrast, the graph-based clustering algorithms (Nie et al. (2017); Wang et al. (2019)) usually utilize the local structure to build the affinity graphs and then perform clustering. They integrate the complementary information to build the shared graph, as a result, the clustering performance largely depends on the quality of the graph.

These multi-view clustering algorithms mentioned above mostly perform learning in the original space, but the original space usually has higher dimensions. To avoid this problem, Wang et al. (2020) project the multi-view data into a common subspace, then learn the common similarity graph from the local structure in the subspace. But, when the original space data is projected into the subspace, the noise will also be projected inevitably. To resist the influence of noise, Zhang et al. (2018) assumes that different views are all originated from one underlying latent representation, and then through the self-representation, the graph is learned to reflect the global structure of latent representation. Inspired by Zhang et al. (2018), Chen et al. (2020) introduced rank constraints on the Laplacian graph to implement the assignment of the sample points into designated clusters. However, (Zhang et al. (2018); Chen et al. (2020)) did not consider the local neighbor relationship in the latent space. Besides, (Huang et al. (2019a,b)) uses kernel tricks and neural network to learning nonlinear relationships.

In summary, in most existing multi-view clustering algorithms, few methods simultaneously mine the local and global similarity of data while avoiding the curse of dimensionality. To solve this issue, this paper proposes a novel algorithm named Multi-view Latent subspace Clustering based on both Global and Local structure (MLCGL). In MLCGL, a common latent representation is learned to capture the complementary information from different views. In the latent space, the global self-representation and local neighbor relationships are combined to construct a common affinity graph. In this way, the obtained affinity graph can more clearly reflect the within-class and the between-class relation. Based on this, a rank constraint is introduced on the Laplacian graph to group the data points in the latent space into a specified number of clusters. The main contributions of this paper are:

1) By learning the global and local structural relationships on the latent representation, the within-class and between-class relation in the affinity graph will be clarified, which promote the latent representation to be divided into more reasonable clusters. 2) The common
latent representation which fuses the complementary information can suppress the possible
noise while avoiding the curse of dimensionality. 3) An iterative optimization scheme is pre-
sented to solve our proposed model. According to the experimental results, our algorithm
can achieve the objective function convergence in a few iterations.

2. Related Work

2.1. Latent Multi-view Subspace Clustering (LMSC)

In LMSC Zhang et al. (2018), it is assumed that the multi-view data $X^v$ is converted from
a shared latent representation $Y$, that is $X^v = W^vY + E^v_y$, where $E^v_y$ is the reconstruction
error, and $W^v$ is the transform matrix. Then, LMSC reconstructs the latent representation
via self-representation: $Y = YZ + E_r$. Besides, LMSC introduced kernel norm constraint
coefficient matrix $Z$ to capture the global structure of the latent representation. Therefore,
the overall objective function is:

$$\min_{W,Y,Z,E} ||E_y||_2,1 + \lambda_1 ||E_r||_2,1 + \lambda_2 ||Z||_*$$

$$s.t. X = WY + E_y, Y = YZ + E_r, WW^T = I$$

is the latent representation reconstruction error. $\lambda_1, \lambda_2$ are the hyper-parameters which are
used to balance the different parts of the objective function. $|| \cdot ||_{2,1}, || \cdot ||_*$ denote $l_{2,1}$ norm
and nuclear norm, respectively.

2.2. Multi-view Clustering and Semi-supervised Classification with Adaptive
Neighbours (MLAN)

Since high-dimensional data contains low-dimensional manifold structure Nie et al. (2016),
MLAN Nie et al. (2017) introduces adaptive local structure to learn the inter-sample rela-
tionship for single view:

$$\min_{s_i} \sum_{i,j} ||x_i - x_j||_2^2 s_{ij} + \alpha ||S||_F^2$$

$$s.t. \forall i, s_i^T 1 = 1, 0 \leq s_{ij} \leq 1$$

where $s_i$ is a vector with $j$-th element as $s_{ij}$ in similarity matrix $S$. The regular term $||S||_F^2$
used to obtain an effective solution where $|| \cdot ||_2$ and $|| \cdot ||_F$ denote $l_2$-norm and the Frobenius
norm respectively. And $\alpha$ is a hyper-parameter to balance the two terms. In addition,
MLAN promotes the division of sample points into reasonable clusters by introducing rank
constraints on the Laplacian graph $L_s$. Besides, MLAN believes that the quality of each
view in the original space is different. Therefore, the objective function in MLAN can be
formulated as:

$$\min_{s_i} \sum_v (w_v \sum_{i,j} ||x_i^v - x_j^v||_2^2 s_{ij} + \gamma ||w_v||_2^2) + \alpha ||S||_F^2$$

$$s.t. \forall i, s_i^T 1 = 1, 0 \leq s_{ij} \leq 1, w_v^T 1 = 1, 0 \leq w_v \leq 1, rank(L_s) = n - c$$
where $L_s = D - ((S^T + S))/2$, $D_{ij} = \sum_j (s_{ij} + s_{ji})/2$. Also, MLAN uses an optimization trick to turn the view weight into an adaptive parameter, and the final objective function is:

$$\min_{s_i} \sum_v \sum_{i,j} \left| \left| x^v_i - x^v_j \right| \right|^2 s_{ij} + \alpha ||S||^2_F$$

subject to $\forall i, s^T_i 1 = 1$, $0 \leq s_{ij} \leq 1$, $\text{rank}(L_S) = n - c$.

where $w_v = 1/\sqrt{\sum_{i,j} \left| \left| x^v_i - x^v_j \right| \right|^2 s_{ij}}$.

3. Proposed

Denote $\{X^1, \cdots, X^V\} \in \mathbb{R}^{d_v \times n}$ as multi-view data containing $V$ views, each view has $n$ samples, and $d_v$ is the feature dimension of the $v$-th view. Denote $\{W^1, \cdots, W^V\}$ as the corresponding transform matrices, and $Y = \{y_i\}_{i=1}^n$ as the shared latent representation. And $\text{Tr}(\cdot)$ denote the trace of the matrix.

3.1. Latent Representation

We represent multi-view data by different mappings of the latent representation: $X^v = W^vY + E^v$, which $E^v$ is the reconstruction error. In this way, the intrinsic relation between views can be explored, and meanwhile the dimensionality of multi-view data can also be reduced. Besides, the possible noise may also be suppressed White et al. (2012). By minimizing the reconstruction error, the latent representation $Y$ is encouraged to fully represent multi-view data, namely:

$$\min_{W^v} \sum_{v=1}^V \left| \left| X^v - W^vY \right| \right|^2_F \quad s.t. W^vT W^v = I$$

3.2. Local and Global Structure Learning

Generally, most graph-based and subspace-based clustering algorithms only learn the similarity relationship separately from the local structure or the global structure, where cause the learned similarity relationship cannot fully indicate the relationship between samples. To this end, this paper will learn both local and global structure on the latent representation simultaneously to obtain a more comprehensive and robust affinity graph.

To mine the local structure hidden in the latent space, our model constructs the objective function as (6), meaning that the larger the Euclidean distance $||y_i - y_j||_2$ between two points, the smaller the corresponding probability $s_{ij}$:

$$\min_S \sum_{i,j} ||y_i - y_j||_2^2 s_{ij} \quad s.t. 1^T s_i = 1, s_{ij} > 0$$

where $S$ indicates the local similarity between samples, $s_i$ is the column vector of $S$, and $s_{ij}$ is the $j$-th element of $s_i$. 

4.
To further discover the global construction relationship in the latent space, our model adopts the self-representation and assumes that $Y = YZ + EY$. By minimizing the Frobenius norm of the reconstruction error $EY$, the reconstruction coefficient matrix $Z$ can capture the global structure for the latent representation:

$$\min_Z \| Y - YZ \|_F^2 \quad s.t. 1^T z_i = 1, z_{ij} > 0$$ (7)

where $Z$ indicates the global similarity between samples, $z_i$ is the column vector of $Z$, and $z_{ij}$ is the $j$-th element of $z_i$.

Therefore, we formulate the objective function (8) to learn the local and the global structure on the latent representation:

$$\min_S \sum_{i,j} \| y_i - y_j \|_2^2 s_{ij} + \| Y - YS \|_F^2 \quad s.t. 1^T s_i = 1, s_{ij} > 0$$ (8)

In the iterative optimization process, the learned affinity graph will also contribute the latent representation to be divided into more reasonable clusters. The constructed graph considers the strengthens of the global structure and the local structure simultaneously, which achieves a balance of the two structure attributes. From the perspective of granularity, the structural information embodied by $S$ tends to be cluster granularity. In this way, the unified graph induced by the similarity has a fuller diagonal block structure, which is also verified by the experimental results section. In this paper, the learned graph $S$ reflects clearer distribution information between classes, so it is more conducive to subsequent clustering.

### 3.3. Rank Constraint

Ideally, the number of connected components in $S$ is equal to the cluster number. But in fact, (8) usually cannot achieve this goal. In order to facilitate the learned affinity graph to be divided into connected components of a specified number, according to Theorem 1 Mohar et al. (1991), a rank constraint is introduced on the Laplacian graph $LS$. That is, if rank $rank(L_S) = n - c$ holds, the corresponding graph $S$ consists of $c$ connected components.

**Theorem 1**

The multiplicity $c$ of the eigenvalue 0 of the Laplacian matrix $L_S$ of $S$ is equal to the number of connected components in the graph with similarity matrix $S$.

Therefore, by introducing rank constraint, equation (8) is reformulated as:

$$\min_S \sum_{i,j} \| y_i - y_j \|_2^2 s_{ij} + \| Y - YS \|_F^2 \quad s.t. 1^T s_i = 1, s_{ij} > 0, rank(L_S) = n - c$$ (9)

### 3.4. Full Objective Function

Combining the aforementioned insights into analysis, we write the objective function of MLCGL as:

$$\min_{W_v,Y,S,F} \sum_{v=1}^V \| X^v - W^vY \|_F^2 + \alpha \sum_{i,j} \| y_i - y_j \|_2^2 s_{ij} + \beta \| Y - YS \|_F^2 \quad s.t. 1^T s_i = 1, s_{ij} > 0, W^vT W^v = I, rank(L_S) = n - c$$ (10)
where $\alpha$ and $\beta$ are the hyper-parameters used to control the weight of the local structure term and the global structure term, respectively. Because of the rank constraint, it is difficult for us to solve the objective function (10). Define $\sigma_i$ as the $i$-th smallest eigenvalue on $L_S$, since Laplace matrix $L_S$ is a positive semi-definite matrix, namely $\sigma_i > 0$, $\text{rank}(L_S) = n - c$ is equivalent to $\sum_{i=1}^{c} \sigma_i = 0$. According to Ky Fan’s theorem (Fan 1949):

$$\sum_{i=1}^{c} \sigma_i(L_S) = \min_{F^TF=I} \text{Tr}(F^TL_SF)$$

where $F$ is the low-dimensional embedding matrix.

Thus the final objective function of MLCGL can be formulated as:

$$\min_{W^v, Y, S, F} \sum_{v=1}^{V} \frac{||X^v - W^vY||_F^2}{c} + \alpha \sum_{i,j} ||y_i - y_j||_2^2 s_{ij} + \beta \frac{||Y - YS||_F^2}{c} + \gamma \text{Tr}(F^TL_SF)$$

$$\text{s.t.} \ a_1^T s_i = 1, s_{ij} > 0, W^v^TW^v = I, F^TF = I$$

where $\gamma$ is used as a hyper-parameter to balance the proportion of rank constraints. When $\gamma$ is large enough, the optimization objective function (12) will make $\text{rank}(L_s) = n - c$ hold.

### 3.5. Optimization

To address the objective function (12), we introduce an alternating optimization scheme to solve this problem.

**Update $F$**

When $W^v$, $Y$ and $S$ are all fixed, the problem of objective function (12) is equivalent to the following:

$$\min_{F} \text{Tr}(F^TL_SF) \quad \text{s.t.} F^TF = I$$

The optimal solution $F$ can be obtained by $c$ eigenvectors of $L_S$ corresponding to the $c$ smallest eigenvalues.

**Update $W$**

When the other variables are fixed, the problem of objective function (12) is equivalent to the following:

$$\min_{W^v} \sum_{v=1}^{V} \frac{||X^v - W^vY||_F^2}{c} \quad \text{s.t.} W^v^TW^v = I$$

Let $X = [(X^1)^T, \cdots, (X^V)^T]^T, W = [(W^1)^T, \cdots, (W^V)^T]^T$, then (14) can be rewritten as:

$$\min_{W} ||X - WY||_F^2 \quad \text{s.t.} W^Tw = I$$

Since $||A||_F = ||A^T||_F$, so (15) is equivalent to:

$$\min_{W} ||X^T - Y^TW^T||_F^2 \quad \text{s.t.} W^T W = I$$

(16)
According to Theorem 2 Wahba (1965), the constraint condition needs to be \( R^T R = R R^T = I \), but in Zhang et al. (2018), it is mentioned that relaxing the constraint to \( R^T R = I \) can also obtain an optimal solution. From this, we can solve \( W \) by Theorem 2.

**Theorem 2**

Given the objective function

\[
\min_R \| Q - GR \|_F^2, \text{s.t.} R^T R = R R^T = I,
\]

the optimal solution is \( R = U V^T \), where \( U \) and \( V \) are left and right singular values of SVD decomposition of \( G^T Q \).

**Update Y**

When the other variables are fixed, the problem of objective function (12) is equivalent to the following:

\[
\min_Y \left\| X - WY \right\|_F^2 + \alpha \sum \left\| y_i - y_j \right\|_2^2 s_{ij} + \beta \left\| Y - YS \right\|_F^2
\]

And \( \sum_{i,j} \left\| y_i - y_j \right\|_2^2 s_{ij} = Tr(Y L_S Y^T) \), so (17) can be rewritten as:

\[
\min_Y \left\| X - WY \right\|_F^2 + \alpha Tr(Y L_S Y^T) + \beta \left\| Y - YS \right\|_F^2
\]

Take the derivative with respect to \( Y \) and set it to zero:

\[
W^T WY - W^T X + \alpha Y(L_S + L_S^T) + \beta(Y(I - S)(I - S)^T) = 0
\]

Because of \( L_S = L_S^T \), Equation (19) is rewritten as:

\[
W^T WY + Y \times (2\alpha L_S + \beta(I - S)(I - S)^T) = W^T X
\]

Equation (20) is a standard Sylvester equation which has a unique solution by the Bartels-Stewart algorithm and there has a similar solution in Hu et al. (2014) or solved by MATLAB toolkit.

**Update S**

When the other variables are fixed, the problem of objective function (12) is equivalent to the following:

\[
\min_S \alpha \sum_{i,j} \left\| y_i - y_j \right\|_2^2 s_{ij} + \beta \left\| Y - YS \right\|_F^2 + \gamma Tr(F^T L_S F)
\]

\[
\text{s.t.} 1^T s_i = 1, s_{ij} > 0
\]

Equation (21) is rewritten as:

\[
\min_S \alpha Tr(Y L_S Y^T) + \beta Tr(K - 2KS + S^T KS) + \gamma Tr(F^T L_S F)
\]

\[
\text{s.t.} 1^T s_i = 1, s_{ij} > 0
\]

where \( K = Y^T Y \), and (22) can be rewritten in a column-wise manner as:

\[
\min_{S_{i,i}} b_i^T S_{i,i} + \beta(K_{ii} - 2K_{i,i}S_{i,i} + S_{i,i}^T KS_{i,i}) \quad \text{s.t.} 1^T s_i = 1, s_{ij} > 0
\]
where \( b_{ij} = \alpha \| y_{i,:} - y_{j,:} \|^2 + \gamma \| f_{i,:} - f_{j,:} \|^2 \). Further, (23) is equivalent to:

\[
\min_{S_{i,:}} \beta S_{i,:}^T K S_{i,:} + (b_i^T - 2\beta K_{i,:}) S_{i,:} \quad s.t. 1^T S_i = 1, s_{ij} > 0
\]  

(24)

Equation (24) is a quadratic programming problem, which can be solved by the related algorithm in Kang et al. (2017) or MATLAB toolkit.

### 3.6. Complexity Analysis

**Algorithm 1 MLCGL**

**Input:** multi-view data \( \{X^1, \cdots, X^V\} \), the dimension \( d \) of latent representation \( Y \), the cluster number \( c \), and the hyper-parameters \( \alpha, \beta, \gamma \).

**Initialize:** Initialize \( S, W = 0 \), and \( Y \) is a matrix with random values.

**While** not convergence, **do**:

1: update \( F \) according to (13).
2: update \( W \) according to (16).
3: update \( Y \) according to (20).
4: For each column vector of \( S \), update according to (24).

**End**

**Output:** \( S, W, Y, F \).

The whole iteration process of MLCGL is shown in Algorithm 1. In every single iteration, the complexity of update \( F \) is \( O(cn^2) \), the complexity of update \( W \) is \( O(d^2 \sum d_v + (\sum d_v)^3) \). The complexity of solving the Sylvester equation when updating \( Y \) is \( O(d^3) \). The complexity of quadratic programming is \( O(n^5) \), so updating \( S \) requires \( O(n^4) \). Because of \( d \ll \sum d_v, c \ll n \), the total complexity of a single iteration is about \( O(n^4 + (\sum d_v)^3) \). The complexity to initialize \( S \) is \( O(Vncd) \), \( d = \max(d_1, d_v) \). Suppose total iterations is \( t \), the total complexity is about \( O(tn^4 + t(\sum d_v)^3 + Vncd) \).

### 4. Experiment

In this section, we evaluate the performance of MLCGL on five real-world datasets. And Table 1 describes the details of the datasets. We compare the proposed MLCGL with the multiple state-of-art algorithms. And we use five evaluation metrics: F-score, Precision, Recall, NMI (normalized mutual information), and ARI (adjusted rand index) to measure the clustering performance.

#### 4.1. Datasets Description

**MSRC-v1** Winn and Jojic (2005): A picture dataset, which contains 210 samples in 7 categories such as trees, buildings, and airplanes. In our experiment, four views are used, which are CM feature, GIST feature, LBP feature, and GENT feature.
Table 1: Dataset Describe

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Numbers</th>
<th>c</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d_3$</th>
<th>$d_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRC-v1</td>
<td>210</td>
<td>7</td>
<td>24</td>
<td>512</td>
<td>256</td>
<td>254</td>
</tr>
<tr>
<td>Caltech101-5</td>
<td>240</td>
<td>5</td>
<td>782</td>
<td>144</td>
<td>213</td>
<td></td>
</tr>
<tr>
<td>3Sources</td>
<td>169</td>
<td>6</td>
<td>3560</td>
<td>3631</td>
<td>3068</td>
<td></td>
</tr>
<tr>
<td>Prokaryotic</td>
<td>551</td>
<td>4</td>
<td>438</td>
<td>3</td>
<td>393</td>
<td></td>
</tr>
<tr>
<td>WebKB</td>
<td>203</td>
<td>4</td>
<td>1703</td>
<td>230</td>
<td>230</td>
<td></td>
</tr>
</tbody>
</table>

Caltech101-5: A picture dataset, which contains 101 types of objects such as faces, animals, and landscapes. This experiment uses a subset of the original dataset, which contains 240 samples in 5 categories. The used views are Gray feature, HOG feature, and LBP feature.

3Sources: A news articles dataset collected from three online sources: BBC, Reuters, and The Guardian. This experiment used 169 news of the original 948 articles. All articles are in the bag-of-words representation.

Prokaryotic: This data set contains 551 samples in 4 categories. All samples are described as heterogeneous, including data and different genomic representations. Three views used in this experiment are textual description, proteome composition, and an indication denoting the presence/absence of gene families in a genome.

WebKB: This dataset contains 203 web pages in 4 categories. Each web-page is described by the content of the page, the anchor text of the hyper-link, and the text in its title.

4.2. Compare Method
Co-Regularized Spectral Clustering (Co-reg): It assumed the cluster membership among views are consistent, and co-regularizes the membership to obtain the common clustering indicator.

Robust Multi-View Spectral Clustering (RMSC): This algorithm introduces the standard Markov chain to clustering.

Multi-View Clustering with Adaptive Neighbors (MLAN): Adaptively learn view weights and dynamically learn the local structure of the data.

Latent Multi-view Subspace Clustering (LMSC): Assume that multiview data are all mapped from the shared latent representation.

Multi-view Low-Rank Sparse Subspace Clustering (MLRSC): Learn the global structure via self-representation on each view, and weigh the discrepancy of different views through a regularization term.

Weighted Multi-View Spectral Clustering (WMSC): Dynamically learn the weights of different views through the spectral perturbation theory. And according to the view’s weight, the clustering results of spectral clustering tend to be consistently on different views.

3. https://linqs.soe.ucsc.edu/data
Graph-based Multi-View Clustering (GMC)\textsuperscript{1}Wang et al. (2019): Dynamically learn the local structure and the view weight of different views, then integrate the local structure into a shared similarity graph according to the weight.

4.3. Clustering Results Comparison

Table 2 to Table 6 show the clustering comparison performance respectively yielded by eight different methods on five datasets. For each metric, higher values indicate better performance. And the best result values are \textbf{bold}.

From the clustering experiment results in Table 2 to Table 6, it can be seen that comparing the multiple algorithms proposed recently, MLCGL can always achieve excellent performance on different datasets. Specifically, on the MSRC-v1 and Caltech101-5 datasets, MLCGL achieves the best performance in all clustering performance metrics. In particular, on MSRC-v1, MCLGL’s performance improvements over the second-best method more than 5%, while on Caltech101-5, the increase has reached about 10%. On 3 Sources and Prokaryotic, only the recall of MCLGL failed to achieve the optimal performance, while the remaining performance metrics all greatly exceed the comparison algorithms. On WebKB, MCLGL achieves the optimal performance in the three clustering performance indicators, while the other two metrics are close to the best.

Table 2: MSRC-v1

<table>
<thead>
<tr>
<th>Method</th>
<th>F-score</th>
<th>Precision</th>
<th>Recall</th>
<th>NMI</th>
<th>ARI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-reg</td>
<td>0.481±0.033</td>
<td>0.453±0.037</td>
<td>0.514±0.033</td>
<td>0.515±0.031</td>
<td>0.392±0.040</td>
</tr>
<tr>
<td>RMSC</td>
<td>0.674±0.033</td>
<td>0.656±0.035</td>
<td>0.693±0.031</td>
<td>0.705±0.026</td>
<td>0.705±0.039</td>
</tr>
<tr>
<td>MLAN</td>
<td>0.698±0.001</td>
<td>0.638±0.001</td>
<td>0.771±0.001</td>
<td>0.752±0.001</td>
<td>0.644±0.001</td>
</tr>
<tr>
<td>LMSC</td>
<td>0.739±0.022</td>
<td>0.730±0.031</td>
<td>0.748±0.025</td>
<td>0.765±0.030</td>
<td>0.696±0.041</td>
</tr>
<tr>
<td>MLRSCC</td>
<td>0.528±0.026</td>
<td>0.518±0.027</td>
<td>0.537±0.017</td>
<td>0.566±0.014</td>
<td>0.450±0.026</td>
</tr>
<tr>
<td>WMSC</td>
<td>0.663±0.014</td>
<td>0.640±0.034</td>
<td>0.687±0.011</td>
<td>0.693±0.012</td>
<td>0.606±0.013</td>
</tr>
<tr>
<td>GMC</td>
<td>0.675±0</td>
<td>0.612±0</td>
<td>0.753±0</td>
<td>0.714±0</td>
<td>0.616±0</td>
</tr>
<tr>
<td>MLCGL</td>
<td>\textbf{0.806±0.005}</td>
<td>\textbf{0.792±0.005}</td>
<td>\textbf{0.820±0.006}</td>
<td>\textbf{0.820±0.006}</td>
<td>\textbf{0.774±0.006}</td>
</tr>
</tbody>
</table>

Table 3: Caltech101-5

<table>
<thead>
<tr>
<th>Method</th>
<th>F-score</th>
<th>Precision</th>
<th>Recall</th>
<th>NMI</th>
<th>ARI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-reg</td>
<td>0.630±0.013</td>
<td>0.615±0.017</td>
<td>0.647±0.010</td>
<td>0.618±0.013</td>
<td>0.535±0.017</td>
</tr>
<tr>
<td>RMSC</td>
<td>0.681±0.011</td>
<td>0.646±0.008</td>
<td>0.718±0.021</td>
<td>0.702±0.012</td>
<td>0.596±0.011</td>
</tr>
<tr>
<td>MLAN</td>
<td>0.609±0</td>
<td>0.521±0</td>
<td>0.732±0</td>
<td>0.617±0</td>
<td>0.490±0</td>
</tr>
<tr>
<td>LMSC</td>
<td>0.745±0.002</td>
<td>0.748±0.002</td>
<td>0.742±0.002</td>
<td>0.736±0.004</td>
<td>0.682±0.002</td>
</tr>
<tr>
<td>MLRSCC</td>
<td>0.718±0.006</td>
<td>0.657±0.019</td>
<td>0.794±0.013</td>
<td>0.744±0.007</td>
<td>0.640±0.009</td>
</tr>
<tr>
<td>WMSC</td>
<td>0.662±0.013</td>
<td>0.638±0.011</td>
<td>0.687±0.021</td>
<td>0.691±0.011</td>
<td>0.574±0.010</td>
</tr>
<tr>
<td>GMC</td>
<td>0.688±0</td>
<td>0.619±0</td>
<td>0.776±0</td>
<td>0.718±0</td>
<td>0.599±0</td>
</tr>
<tr>
<td>MLCGL</td>
<td>\textbf{0.851±0.012}</td>
<td>\textbf{0.847±0.013}</td>
<td>\textbf{0.855±0.010}</td>
<td>\textbf{0.832±0.012}</td>
<td>\textbf{0.813±0.015}</td>
</tr>
</tbody>
</table>
Table 4: 3Sources

<table>
<thead>
<tr>
<th>Method</th>
<th>F-score ±</th>
<th>Precision ±</th>
<th>Recall ±</th>
<th>NMI ±</th>
<th>ARI ±</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-reg</td>
<td>0.490±0.041</td>
<td>0.534±0.064</td>
<td>0.454±0.032</td>
<td>0.518±0.038</td>
<td>0.351±0.057</td>
</tr>
<tr>
<td>RMSC</td>
<td>0.393±0.011</td>
<td>0.417±0.022</td>
<td>0.372±0.017</td>
<td>0.465±0.009</td>
<td>0.223±0.025</td>
</tr>
<tr>
<td>MLAN</td>
<td>0.683±0</td>
<td>0.609±0</td>
<td>0.777±0</td>
<td>0.689±0</td>
<td>0.571±0</td>
</tr>
<tr>
<td>LMSC</td>
<td>0.410±0.003</td>
<td>0.494±0.002</td>
<td>0.350±0.003</td>
<td>0.387±0.005</td>
<td>0.269±0.003</td>
</tr>
<tr>
<td>MLRSSC</td>
<td>0.572±0.024</td>
<td>0.656±0.030</td>
<td>0.508±0.028</td>
<td>0.620±0.017</td>
<td>0.463±0.029</td>
</tr>
<tr>
<td>WMSC</td>
<td>0.430±0.016</td>
<td>0.424±0.014</td>
<td>0.435±0.020</td>
<td>0.471±0.015</td>
<td>0.254±0.017</td>
</tr>
<tr>
<td>GMC</td>
<td>0.605±0</td>
<td>0.484±0</td>
<td>0.805±0</td>
<td>0.621±0</td>
<td>0.443±0</td>
</tr>
<tr>
<td>MLCGL</td>
<td>0.771±0.019</td>
<td>0.767±0.049</td>
<td>0.778±0.030</td>
<td>0.747±0.008</td>
<td>0.700±0.028</td>
</tr>
</tbody>
</table>

Table 5: Prokaryotic

<table>
<thead>
<tr>
<th>Method</th>
<th>F-score ±</th>
<th>Precision ±</th>
<th>Recall ±</th>
<th>NMI ±</th>
<th>ARI ±</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-reg</td>
<td>0.452±0.007</td>
<td>0.552±0.009</td>
<td>0.383±0.007</td>
<td>0.284±0.013</td>
<td>0.191±0.011</td>
</tr>
<tr>
<td>RMSC</td>
<td>0.562±0.019</td>
<td>0.660±0.011</td>
<td>0.489±0.016</td>
<td>0.394±0.021</td>
<td>0.341±0.020</td>
</tr>
<tr>
<td>MLAN</td>
<td>0.577±0.005</td>
<td>0.444±0.009</td>
<td><strong>0.827±0.008</strong></td>
<td>0.239±0.010</td>
<td>0.132±0.006</td>
</tr>
<tr>
<td>LMSC</td>
<td>0.605±0.022</td>
<td>0.673±0.027</td>
<td>0.533±0.031</td>
<td>0.434±0.017</td>
<td>0.389±0.033</td>
</tr>
<tr>
<td>MLRSSC</td>
<td>0.591±0.056</td>
<td>0.725±0.068</td>
<td>0.499±0.048</td>
<td>0.437±0.039</td>
<td>0.389±0.082</td>
</tr>
<tr>
<td>WMSC</td>
<td>0.500±0.031</td>
<td>0.585±0.027</td>
<td>0.436±0.019</td>
<td>0.388±0.030</td>
<td>0.245±0.011</td>
</tr>
<tr>
<td>GMC</td>
<td>0.461±0</td>
<td>0.447±0</td>
<td>0.476±0</td>
<td>0.193±0</td>
<td>0.091±0</td>
</tr>
<tr>
<td>MLCGL</td>
<td><strong>0.671±0.020</strong></td>
<td><strong>0.757±0.022</strong></td>
<td>0.603±0.013</td>
<td><strong>0.487±0.015</strong></td>
<td><strong>0.494±0.035</strong></td>
</tr>
</tbody>
</table>

Table 6: WebKb

<table>
<thead>
<tr>
<th>Method</th>
<th>F-score ±</th>
<th>Precision ±</th>
<th>Recall ±</th>
<th>NMI ±</th>
<th>ARI ±</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-reg</td>
<td>0.604±0.056</td>
<td>0.640±0.040</td>
<td>0.579±0.087</td>
<td>0.343±0.034</td>
<td>0.373±0.072</td>
</tr>
<tr>
<td>RMSC</td>
<td>0.591±0.014</td>
<td>0.607±0.017</td>
<td>0.576±0.010</td>
<td>0.281±0.012</td>
<td>0.338±0.025</td>
</tr>
<tr>
<td>MLAN</td>
<td>0.668±0</td>
<td>0.559±0</td>
<td>0.831±0</td>
<td>0.402±0</td>
<td>0.373±0</td>
</tr>
<tr>
<td>LMSC</td>
<td>0.683±0.006</td>
<td><strong>0.729±0.006</strong></td>
<td>0.641±0.006</td>
<td>0.473±0.007</td>
<td>0.497±0.010</td>
</tr>
<tr>
<td>MLRSSC</td>
<td>0.714±0.028</td>
<td>0.648±0.046</td>
<td>0.792±0.037</td>
<td>0.468±0.056</td>
<td>0.490±0.064</td>
</tr>
<tr>
<td>WMSC</td>
<td>0.474±0.014</td>
<td>0.574±0.011</td>
<td>0.403±0.017</td>
<td>0.287±0.015</td>
<td>0.221±0.013</td>
</tr>
<tr>
<td>GMC</td>
<td>0.688±0</td>
<td>0.582±0</td>
<td><strong>0.840±0</strong></td>
<td>0.413±0</td>
<td>0.417±0</td>
</tr>
<tr>
<td>MLCGL</td>
<td><strong>0.723±0.014</strong></td>
<td>0.670±0.005</td>
<td>0.790±0.052</td>
<td><strong>0.513±0.011</strong></td>
<td><strong>0.518±0.031</strong></td>
</tr>
</tbody>
</table>

4.4. Parameter Analysis

Table 7 shows the optimal parameters of MLCGL on each dataset. Fig. 1 presents the relationship between the dimension of the latent representation Y and the clustering performance metric NMI. From the experimental results of the above datasets, the dimensionality of the latent representation has a great influence on the algorithm clustering performance.
Compared with the high dimensionality of the original space, the dimensionality of the latent representation is basically between 10 and 50, which indicated that the original dimensionality may contain too many redundancy and possible noise. After learned the latent representation, the dimensionality of the multi-view data is significantly reduced and the noise is effectively suppressed.

Table 7: Best parameters of each datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>d</th>
<th>α</th>
<th>β</th>
<th>γ</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRC-v1</td>
<td>50</td>
<td>0.5</td>
<td>0.5</td>
<td>10</td>
</tr>
<tr>
<td>Caltech101-5</td>
<td>40</td>
<td>1</td>
<td>0.5</td>
<td>10</td>
</tr>
<tr>
<td>3Sources</td>
<td>20</td>
<td>10</td>
<td>0.01</td>
<td>0.5</td>
</tr>
<tr>
<td>Prokaryotic</td>
<td>10</td>
<td>0.005</td>
<td>0.005</td>
<td>10</td>
</tr>
<tr>
<td>WebKB</td>
<td>10</td>
<td>0.01</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 1: Dimensionality of latent $Y$ vs NMI.

Figure 2 shows the relationship among hyper-parameters $\alpha$, $\beta$, and clustering metric NMI, where $\alpha$ affects the preservation of the local structure, and $\beta$ affects the preservation of the global structure. From the experiments on different datasets, we can see that on Caltech101-5 and 3Sources datasets, the clustering performance is greatly affected by the local structure. For WebKB, the performance is greatly affected by the global structure, and for MSRC-v1 and Prokaryotic, the global structures and local structures have similar influence on clustering performance.

Figure 3 gives the relationship between the hyper-parameter $\gamma$ and the clustering performance metric NMI, where $\gamma$ denotes the influence of the rank constraint introduced on the Laplacian graph on MLCGL. From the experimental results shown in Fig. 3, in the existing parameter range, different hyper-parameters $\gamma$ have little effect on the clustering.
performance. Then, if we extend the parameter search range, another hyper-parameter $\gamma$ may be searched to obtain better clustering performance.

![Graphs showing hyper-parameter $\alpha$ and $\beta$ vs NMI.](image1)

![Graphs showing hyper-parameter $\gamma$ vs NMI.](image2)

**4.5. Similarity Graph Comparison**

In Fig. 4, MLAN trends to learn the local structure in the original space, LMSC learns the global structure in the latent space, and MLRSSC learns the global structure in the original space. Observing the similarity graph in Fig. 4, the diagonal blocks of MLCGL are denser than others, indicating that our method MLCGL encourages us to lean a denser and distinct diagonal blocks with $c$ connected components.
4.6. Convergence Analysis

Figure 5 shows the convergence curves of the objective function in different datasets. Obviously, MLCGL can quickly converge within the first 10 iterations.

From the experimental results shown above, it’s concluded that MLCGL can learn the local structure and global structure on the latent representation simultaneously. By introducing the rank constraint in the Laplacian graph, MLCGL can further make the clustering performance obtained on the latent representations more reasonable. The convergence curve also presents the efficiency of the algorithm solution.

5. Conclusion

In this paper, we proposed an algorithm named Multi-view Latent subspace Clustering based on both Global and Local structure. MLCGL learns latent representation, similarity graph, and cluster indicator simultaneously in a single model. Experimental results on multiple real-world datasets also prove the excellent performance of MLCGL. But for large-scale datasets, we do not have an effective solution. In the following work, we will try to obtain a robust low-dimensional representation in the framework of Contrastive Learning.
References


