This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Multi-View Subspace Clustering

Hongchang Gao
Computer Science and Engineering
University of Texas at Arlington
Arlington, TX, 76019, USA
hongchanggao@gmail.com

Feiping Nie
Computer Science and Engineering
University of Texas at Arlington
Arlington, TX, 76019, USA
feipingnie@gmail.com

Xuelong Li
Center for OPTIMAL, XIOPM
Chinese Academy of Sciences
Xi'an, 710119, China
xuelong_li@opt.ac.cn

Heng Huang*
Computer Science and Engineering
University of Texas at Arlington
Arlington, TX, 76019, USA
heng@uta.edu

Abstract

For many computer vision applications, the data sets distribute on certain low-dimensional subspaces. Subspace clustering is to find such underlying subspaces and cluster the data points correctly. In this paper, we propose a novel multi-view subspace clustering method. The proposed method performs clustering on the subspace representation of each view simultaneously. Meanwhile, we propose to use a common cluster structure to guarantee the consistence among different views. In addition, an efficient algorithm is proposed to solve the problem. Experiments on four benchmark data sets have been performed to validate our proposed method. The promising results demonstrate the effectiveness of our method.

1. Introduction

In recent years, subspace clustering has been explored extensively. It assumes that the data points are drawn from multiple low-dimensional subspaces. Therefore, many subspace clustering models have been proposed to uncover such underlying subspaces such that all data points can be segmented correctly and each group fits into one of the low-dimensional subspaces.

A number of subspace clustering approaches have been developed in recent years. For instance, the iteration based methods such as [21] and [11], the factorization based methods such as [5] and [12], statistical approaches such as [20], and spectral clustering based approaches such as [25] and [10]. Besides, sparse subspace clustering (SSC) has been proposed in [8] to find a sparse representation corresponding to the data points from the same subspace. After obtaining the representation of the subspace, the spectral clustering can be performed on such new representation. The low-rank subspace segmentation (LRR) was proposed in [16] to find the subspace structure with a low-rank representation. Additionally, [18] proposed another subspace discovery method, which is to discover the number of the subspace, its dimension and the data points in each subspace. However, these methods mostly focus on the features from single source rather than multiple ones. In this paper, we will apply the subspace clustering on the data set with multi-view features to uncover the subspace structure of the data set and perform clustering on it.

Many problems in computer vision are concerned with the data set represented by multiple distinct feature sets. Different feature sets characterize different and partly independent information about the data set. For instance, an image can be described by the color, texture, shapes and so on. As another example, in multi-lingual information retrieval, a document is simultaneously described by several different languages. These different features can provide useful information from different views so as to improve the clustering performance.

The multi-view clustering is to integrate these multiple feature sets together to perform clustering. Much progress has been made in developing effective multi-view clustering method. The multi-view spectral clustering model was proposed in [3] to integrate heterogeneous visual descriptors for image categorizations. The co-regularized multi-view spectral clustering was introduced in [13] to

*Corresponding Author. HG, FN, HH were supported by U.S. NSF-IIS 1117965, NSF-IIS 1302675, NSF-IIS 1344152, NSF-DBI 1356628, NIH 1R01AG049371-01A1.

perform clustering on different views simultaneously with a co-regularization constraint. To solve the large-scale multi-view clustering problems, [2] proposed multi-view K-means clustering method. However, these methods only focus on the clustering directly, rather than mining the structure of the features. The other structured sparse learning based multi-view clustering method was proposed in [22] to perform feature selection and multi-view clustering simultaneously.

In this paper, we propose a novel multi-view subspace clustering model. Unlike the method in [4] where performing clustering on a common view, we perform subspace clustering on each view simultaneously, meanwhile guarantee the consistence of the clustering structure among different views. Specifically, we perform clustering on the subspace representation of each view simultaneously. To make sure the consistence among different views, we adopt a common indicator to guarantee the common cluster structure. That is to enforce the points in different views to be classified into the same cluster. We also propose an efficient algorithm to solve our novel optimization problem. At last, extensive experiments on four benchmark data sets show the effectiveness of our method.

2. Multi-View Subspace Clustering

In this section, we will introduce the standard subspace clustering method, that is to obtain the subspace structure of the original data set and perform clustering on such subspace representation of the data set. After that, we will give the motivation of our multi-view subspace clustering.

2.1. Subspace Clustering

For a data set, it usually lies in an underlying low-dimension subspace rather than distribute uniformly in the entire space. Thus, the data points can be represented by a low-dimension subspace. After obtaining the subspace structure of the data set, we can perform clustering based on the subspace rather than the entire space.

Given n data points $X = \{x_1, x_2, ..., x_n\} \in \Re^{d\times n}$, the subspace clustering uses the self-expression property [8] of the data set to represent itself as:

$$X = XZ + E\,, \tag{1}$$

where $Z = \{z_1, z_2, ..., z_n\} \in \Re^{n\times n}$ is the subspace representation matrix, and each z_i is the representation of the original data point x_i based on the subspace. $E \in \Re^{d\times n}$ is the error matrix.

The subspace clustering needs solve the following optimization problem:

$$\min_{Z} ||X - XZ||_F^2 \quad s.t. \quad Z(i,i) = 0, Z^T\mathbf{1} = \mathbf{1}\,. \tag{2}$$

The constraint $Z^T\mathbf{1} = \mathbf{1}$ denotes that the data point lies in a union of affine subspaces, rather than linear subspaces. The constraint $Z(i,i) = 0$ eliminates the case that a data point is repesented as a combination of itself, which means that each data point x_i can only be represented as the combination of other points $x_j (j \neq i)$. Solving the optimization problem (2), we will get the representation z_i for each data point x_i. The nonzero elements of z_i correspond to points from the same subspace [8]. Thus, we can get the subspace structure Z of the original data set X.

After obtaining the subspace structure, we will get the affinity matrix $W = \frac{|Z|+|Z^T|}{2}$ for the data set. Thus, we can perform spectral clustering on such a subspace affinity matrix:

$$\min_{F} Tr(F^T L F) \quad s.t. \quad F^T F = I\,, \tag{3}$$

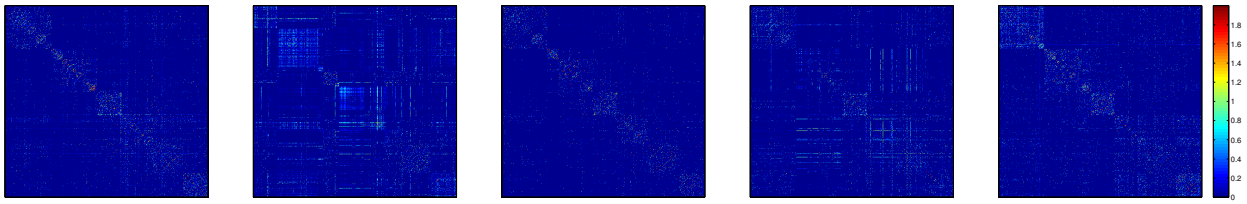
where F is the cluster indicator matrix, and $L = D - W$ where D is a diagonal matrix whose diagonal elements are defined as $d_{ii} = \Sigma_j w_{ij}$.

2.2. Multi-View Subspace Clustering

In this section, we will introduce our novel multi-view subspace clustering model. Given the data set $X_v \in \Re^{d_v\times n}$, which denotes the features in the v-th view ($v = 1, 2, 3, ..., k$, totally we have k views). If we perform the subspace learning on each single view, we can get the subspace representation Z_v for the v-th view. The nonzero elements in Z_v correspond to the data points from the same subspace. In fact, how to combine multi-view features in subspace clustering is challenging. The naive method is to concatenate all the features together and perform clustering on the concatenated features. However, in such a method, the more informative view and the less informative one will be treated equally. Thus, the solution is not optimal. In fact, it is better to perform the clustering on individual view perspectively and then unify them together. In order to combine the multi-view subspace clustering results, we can perform the subspace learning on different views simultaneously by simply solving:

$$\begin{aligned} &\min_{Z_v, Z} \sum_v ||X_v - X_v Z_v||_F^2 + \lambda \sum_v ||Z - Z_v|| \\ &s.t. \quad Z_v^T\mathbf{1} = \mathbf{1}, Z_v(i,i) = 0\,, \end{aligned} \tag{4}$$

where Z is the unified subspace representation result and then spectral clustering can be performed on Z. The above multi-view feature learning strategy has been used in previous computer vision research, but it doesn't work for subspace clustering. Although the data block structures in different Z_v are similar, the magnitude of element values in Z_v can be dramatically different, just as shown in Fig. 1. Thus, how to integrate subspace clustering using multi-view features is not trivial.



(a) CENSTRIST (b) CMT (c) LBP (d) HOG (e) SIFT

Figure 1. By solving Eq. (2) on Caltech101-7 dataset, we get the structure of subspace representation Z for five views respectively. Although different Z shows similar block structure, the magnitude of element values are different.

Instead of computing a common Z to unify different Z_v, we propose to integrate the clustering results using different Z_v such that the data block structures can be jointly learned by multi-view visual features. Meanwhile, we consider the existing subspace clustering methods used a separated spectral clustering post-processing step, which may lead to sub-optimal results. We propose a novel multi-view subspace clustering method to address these challenges by solving:

$$\begin{aligned} &\min_{Z_v,F} \sum_v ||X_v - X_v Z_v||_F^2 \\ &+ \lambda \sum_v Tr(F^T(D_v - W_v)F) \\ &s.t. \quad Z_v^T \mathbf{1} = \mathbf{1}, Z_v(i,i) = 0, F^T F = I \ , \end{aligned} \tag{5}$$

where $W_v = \frac{|Z_v| + |Z_v^T|}{2}$, Z_v is the subspace representation matrix of the v-th view, F is the cluster indicator matrix, and D_v is a diagonal matrix with diagonal elements defined as $d_{v_{ii}} = \sum_j w_{v_{ij}}$.

In our new objective, we use the same indicator matrix F for all of the views, thus the clustering results will be consistent for all of the views, which means that the corresponding points in different view will be in the same cluster.

Note that, in Eq. (5) we do not adopt the low rank subspace representation for multi-view features, such as [16]. Because the result of this method is bad for the low dimension features, such as the Color Moment feature shown in Figure 2, due to the rigor restriction for the rank.

Moreover, for the real-world data set, the data point does not perfectly lie in a subspace, it is usually corrupted by the outlying entries due to some inevitable reasons. Thus, to make our method be more robust to different data set, we generalize the above objective to our final objective as follows:

$$\begin{aligned} &\min_{Z_v,E_v,F} \sum_v ||X_v - X_v Z_v - E_v||_F^2 \\ &+ \lambda_1 \sum_v Tr(F^T(D_v - W_v)F) + \lambda_2 \sum_v ||E_v||_1 \\ &s.t. \quad Z_v^T \mathbf{1} = \mathbf{1}, Z_v(i,i) = 0, F^T F = I \ , \end{aligned} \tag{6}$$

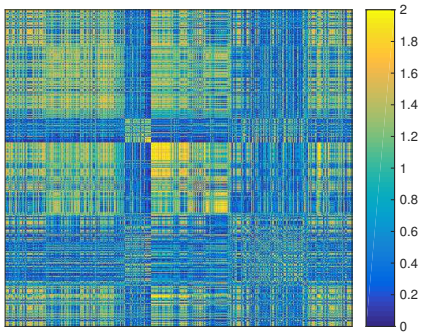


Figure 2. By solving the low rank subspace problem as [16], we get the subspace representation Z for Color Moment feature whose dimension is far smaller than the number of data points.

where $E_v = \{e_{v_1}, e_{v_2}, ..., e_{v_n}\}$ is the outlying entries matrix whose column e_{v_i} is the outlying entry for data point x_{v_i}. Due to the number of outlying entries not very much, we use the ℓ_1-norm regularization term to guarantee the sparsity of the outlying entries matrix.

3. Optimization Algorithm

It is difficult to solve the constrained problem in Eq. (6). In this section, we propose an alternative algorithm to solve this optimization problem efficiently.

3.1. Update cluster indicator matrix F

The first step is fixing Z_v and E_v, updating the clustering indicator matrix F. When Z_v and E_v are fixed, the problem in Eq. (6) can be rewritten as the following problem with respect to F:

$$\begin{aligned} &\min_F \sum_v Tr(F^T(D_v - \frac{|Z_v| + |Z_v|^T}{2})F) \\ &s.t. \quad F^T F = I \ . \end{aligned} \tag{7}$$

Furthermore, it can be rewritten as the following:

$$\begin{aligned} &\min_{F} Tr(F^T M F) \\ &s.t. \quad F^T F = I \ , \end{aligned} \tag{8}$$

where $M = \sum_v (D_v - \frac{|Z_v| + |Z_v|^T}{2})$. The solution of such a problem (8) are the eigenvectors corresponding to the smallest k eigenvalue of the Laplacian matrix M.

3.2. Update subspace representation matrix Z_v

The second step is fixing F and E_v, updating the subspace representation matrix Z_v. When F and E_v are fixed, we have

$$\begin{aligned} &\min_{Z_v} \sum_v ||X_v - X_v Z_v - E_v||_F^2 \\ &+ \lambda \sum_v Tr(F^T (D_v - \frac{|Z_v| + |Z_v|^T}{2}) F) \\ &s.t. \quad Z_v^T \mathbf{1} = \mathbf{1}, Z_v(i,i) = 0 \ . \end{aligned} \tag{9}$$

Specifically, Z_v can be solved separately for each view v as follows:

$$\begin{aligned} &\min_{Z_v} ||X_v - X_v Z_v - E_v||_F^2 \\ &+ \lambda Tr(F^T (D_v - \frac{|Z_v| + |Z_v|^T}{2}) F) \\ &s.t. \quad Z_v^T \mathbf{1} = \mathbf{1}, Z_v(i,i) = 0 \ . \end{aligned} \tag{10}$$

For convenience, ignoring the subscript tentatively, we get

$$\begin{aligned} &\min_{Z} ||X - XZ - E||_F^2 \\ &+ \lambda Tr(F^T (D - \frac{|Z| + |Z|^T}{2}) F) \\ &s.t. \quad Z^T \mathbf{1} = \mathbf{1}, Z(i,i) = 0 \ . \end{aligned} \tag{11}$$

When we replace X with $[X^T, \alpha * \mathbf{1}]^T$ where α approaches to infinity and E with $[E^T, \mathbf{0}]^T$, it is equivalent to the following problem:

$$\begin{aligned} &\min_{Z} ||X - XZ - E||_F^2 \\ &+ \lambda Tr(F^T (D - \frac{|Z| + |Z|^T}{2}) F) \\ &s.t. \quad Z(i,i) = 0 \ . \end{aligned} \tag{12}$$

We can prove problem (11) is equivalent to (12) by expanding the problem (12) as following:

$$\begin{aligned} &||X - XZ - E||_F^2 \\ &= ||X_{original} - X_{original} Z - E_{original}||_F^2 \\ &+ \alpha ||\mathbf{1}^T - \mathbf{1}^T Z - \mathbf{0}^T||_F^2 \ , \end{aligned} \tag{13}$$

where $X_{original}$ and $E_{original}$ are the original X and E. When α approaches to infinity, $Z^T \mathbf{1}$ approaches to $\mathbf{1}$. Thus, problem (11) is equivalent to problem (12). To reformulate the problem (12), we need the following theorem.

Theorem 1. *For Laplacian matrix $L \in \Re^{n \times n}$ and the matrix $F \in R^{n \times c}$, we have*

$$Tr(F^T L F) = \frac{1}{2} Tr(WP)$$

where $P_{ij} = ||f^i - f^j||_2^2$, f^i is the i-th row of matrix F.

It was proved in [1]. From theorem 1, it is easy to reformulate problem (12) as following:

$$\begin{aligned} &\min_{Z} ||X - XZ - E||_F^2 + \frac{\lambda}{2} Tr(|Z|^T P) \\ &s.t. \quad Z(i,i) = 0 \ , \end{aligned} \tag{14}$$

where $P_{ij} = ||f^i - f^j||_2^2$. Then, we can use alternative optimization strategy to solve problem (14). When all the rows except the i-th row are fixed, we can update the i-th row of Z by solving the following problem:

$$\begin{aligned} &\min_{z} ||X_1 - xz^T||_F^2 + \frac{\lambda}{2} |z|^T p \\ &s.t. \quad z_i = 0 \ , \end{aligned} \tag{15}$$

where z^T is the i-th row of Z, and p is the i-th column of P, and $X_1 = X - (XZ - xz^T) - E$. We can easily verify that problem (15) is equivalent to problem (16), since their objectives differ only by a constant.

$$\begin{aligned} &\min_{z} x^T x z^T z - 2 z^T X_1^T + x + \frac{\lambda}{2} |z|^T p \\ &s.t. \quad z_i = 0 \ . \end{aligned} \tag{16}$$

Also, the objective in problem (16) differs with the following problem only by a constant:

$$\begin{aligned} &\min_{z} ||z - v||_2^2 + \frac{\lambda}{2} |z|^T p \\ &s.t. \quad z_i = 0 \ , \end{aligned} \tag{17}$$

where $v = \frac{X_1^T x}{x^T x}$. Thus, problem (15) has the same solution with problem (17) which has closed form solution. In detail, if $k = i$, then $z_k = 0$. If $k \neq i$, we can solve the following problem:

$$\min_{z_k} \frac{1}{2} (z_k - v_k)^2 + \frac{\lambda p_k}{4} |z_k| \ . \tag{18}$$

The solution of problem (18) is as following:

$$\begin{aligned} z_k &= sign(v_k)(|v_k| - \frac{\lambda p_k}{4})_+ \\ &= \begin{cases} v_k - \frac{\lambda p_k}{4}, & if \quad v_k > \frac{\lambda p_k}{4} \\ v_k + \frac{\lambda p_k}{4}, & if \quad v_k < -\frac{\lambda p_k}{4} \\ 0, & otherwise \end{cases} \end{aligned} \tag{19}$$

3.3. Update the outlying entries matrix E_v

The third step is fixing Z_v and F, updating the outlying entries matrix E_v. When Z_v and F are fixed, we have

$$\min_{E_v} \sum_v ||X_v - X_v Z_v - E_v||_F^2 + \lambda \sum_v ||E_v||_1 \ . \quad (20)$$

Specifically, E_v can be solved separately for each view v. For convenience, we ignore the subscript as follows:

$$\min_E \frac{1}{2}||E - (X - XZ)||_F^2 + \frac{\lambda}{2}||E||_1 \ . \quad (21)$$

We solve problem (21) column-wise as follows:

$$\min_{e_i} \frac{1}{2}||e_i - (X - XZ)_i||_2^2 + \frac{\lambda}{2}||e_i||_1 \ , \quad (22)$$

where e_i is the i-th column of E. For problem (22), it has closed form solution just as equation (19). Thus, the solution of problem (21) is as follows:

$$\begin{aligned} E_{ij} &= sign(X - XZ)_{ij}(abs(X - XZ)_{ij} - \frac{\lambda}{2})_+ \\ &= \begin{cases} (X - XZ)_{ij} - \frac{\lambda}{2}, & if \quad (X - XZ)_{ij} > \frac{\lambda}{2} \\ (X - XZ)_{ij} + \frac{\lambda}{2}, & if \quad (X - XZ)_{ij} < -\frac{\lambda}{2} \\ 0, & otherwise \end{cases} \end{aligned} \quad (23)$$

By the above three steps, we alternatively update the Z_v, E_v, F and repeat the process again and again until the objective function approaches to convergence. We summarize the above algorithm in Alg. 1

Algorithm 1 Algorithm to solve the problem in Eq. (6).

Input:
$X = \{X_1, X_2, ..., X_k\}$, $X_v \in R^{d_v \times n}$
Output: $F \in R^{n \times c}$
Initialize $Z_v = 0$, $E_v = 0$, F.
repeat
Update F by Eq. (8):

$$\min_{F^T F = I} Tr(F^T M F)$$

Solve it by eigenvalue decomposition of M;
Update the i-th row of Z_v by Eq. (19):
For $k = i$, $z_k = 0$;
For $k \neq i$,

$$z_k = \begin{cases} v_k - \frac{\lambda p_k}{4}, & if \quad v_k > \frac{\lambda p_k}{4} \\ v_k + \frac{\lambda p_k}{4}, & if \quad v_k < -\frac{\lambda p_k}{4} \\ 0, & otherwise \end{cases}$$

Update E_v by Eq. (23)

$$E_{ij} = \begin{cases} (X - XZ)_{ij} - \frac{\lambda}{2}, & if \quad (X - XZ)_{ij} > \frac{\lambda}{2} \\ (X - XZ)_{ij} + \frac{\lambda}{2}, & if \quad (X - XZ)_{ij} < -\frac{\lambda}{2} \\ 0, & otherwise \end{cases}$$

until Converges

4. Experiment

In this section, we have evaluated our algorithm on three widely used benchmark data sets. That is Caltech-101 [9], Microsoft Research Cambridge Volume [23] and the ETH Zurich ETH-80 [15].

4.1. Feature Description

For the three image dataset, we extract different features to construct a multi-view visual features. The features adopted in this paper is shown as the following.

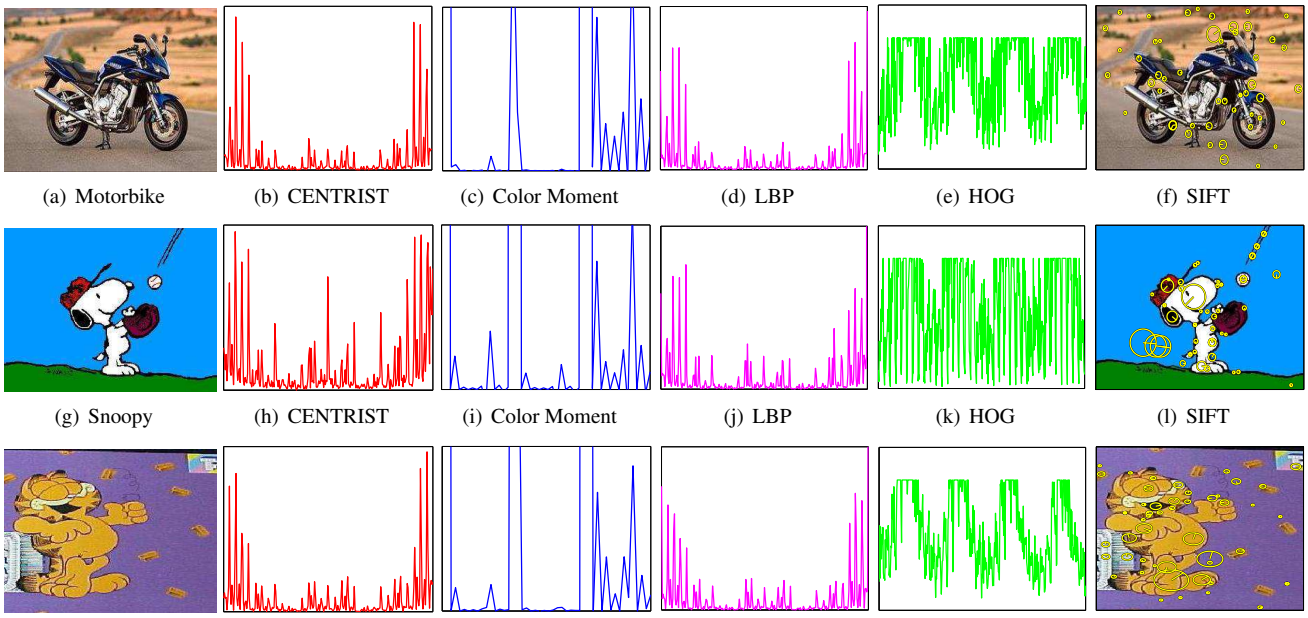
- **CENTRIST** [24] is a holistic representation for the image. It can capture structural properties such as rectangular shapes, flat surfaces and so on. Thus, we can use it to capture the geometrical information based on the histogram of the gradients, especially for the global shape structure within an image.
- **Color Moment (CMT)** [26] calculates the first and second moments as the representation of the image pixel distribution, describing the local photometrical and spatial information based on pixel values. With it we can obtain the distribution of the color within an image.
- **HOG** [6] is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid. It can capture edge or gradient structure that is very characteristic of local shape. We can extract the local object appearance and shape within an image.
- **LBP** [19] computes the histogram of local binary patterns in an encoded image to capture the textures. It is a texture descriptor, we can use it to extract the texture information within the image.
- **SIFT** [17] extracts key points from the image at first, and then keypoint descriptor is calculated as the representation that allows for significant levels of local shape distortion and change in illumination. We use LIBPMK [1] tookit to build the similarity matrix.

We utilize these features to construct multiple feature sets. In Fig. 3, we show the visual patterns extracted from three sample images of Caltech101-7 with CENTRIST, ColorMoment, LBP, HOG and SIFT .

4.2. Data Set Descriptions

The detailed information about these data sets are shown as the following.

[1] http://people.csail.mit.edu/jjl/libpmk/



(a) Motorbike (b) CENTRIST (c) Color Moment (d) LBP (e) HOG (f) SIFT

(g) Snoopy (h) CENTRIST (i) Color Moment (j) LBP (k) HOG (l) SIFT

(m) Garfield (n) CENTRIST (o) Color Moment (p) LBP (q) HOG (r) SIFT

Figure 3. The visual patterns of descriptors CENTRIST, ColorMoment, LBP, HOG and SIFT on three sample images from Caltech101-7.

- **Caltech101-7** [7] is a widely used subset of the image dataset Caltech101 which contains 101 categories. In this subset, there are 7 categories, including Dolla-Bill, Faces, Garfield, Motorbikes,Snoopy, Stop-Sign and Windsor-Chair, and there are 441 images totally selected from the 7 categories. For each image, we extract the above 5 visual features. In detail, the dimension of CENTRIST feature is 1302, the diension of Color Moment (CMT) is 48, the dimension of HOG is 100, the dimension of LBP is 256 and SIFT is 128.
- **MSRCV1** data set consists of 240 images and 8 object classes. Just as [14], we select 7 classes, including tree, building, airplane, cow, face, car and bicycle. For each class, we randomly sample 30 images. Just as Caltech101 data set, we extract the same features from each image to construct different view features.
- **ETH-80** data set contains 8 categories, including apples, pears, tomatoes, cows, dogs, horses, cups and cars. We sample 50 images from each categories, and extract different features, including LBP, HOG, CENTRIST and CMT, to obtain the multiple view representation for the data set.
- **Caltech101-20** data set also comes from Caltech101 data set. However, 20 categories have been selected to evaluate the performance of our algorithm on the complex data set. They are binocular, brain, camera, car-side, faces, ferry, garfield, hedgehog, leopards, motobike, pagoda, rhino, snoopy,stapler, stop-sign, waterlily, windsor, chair, wrench and yin-yang. There are 1230 images totally. Just as Caltech101-7 data set, we extract the same kinds of visual features to construct different view representations for the data set.

4.3. Experiment Setup

To evaluate the perfomance of our method, we have compared our method with each single-view counterpart. We have also compared with methods on the concatenated features. Besides, we compare with other state-of-the-art methods, including centroid co-regularized multi-modal spectral clustering (CC-MSC) [13], and pair-wised co-regularized multi-modal spectral clustering (PC-MSC) [13]. To evaluate the performance quantitatively, the Clustering Accuracy and Normal Mutual Information are resorted to measure the performance of multi-view subspace clustering. The detailed information about the comparison is as the following:

- **Single View with SSC:** For each single view feature, at first we use the subspace learning method proposed in [8], to get the its subspace representation, and then we run spectral clustering method on the subspace representation.
- **Single Modality with LRR:** We run LRR [16] on each view features to get the low-rank subspace representation, and then run spectral clustering on such representations.
- **MLRR_Con and MSSC_Con:** We concatenate all features together and run LLR [16] and SSC [8] respectively to get the subspace representation of the

data set. After that, following [16] we perform spectral clustering on the subspace representation.

- **MLRR_Add and MSSC_Add:** Different with MLRR_Con method, MLRR_Add method is to run LLR [16] on each view features to get the subspace representation of each view, and then sum these representations together and perform spectral clustering on it. The same procedure is performed by MSSC_Add method except that we run SSC [8] method on each view features.

- **CC-MSC [13] :** This method enforces the corresponding point in different modality to be in the same cluster by a centroid-based co-regularization term, which makes different views to be same to a common one.

- **PC-MSC [13] :** This method is similar to CC-MSC, other than a pair-wised co-regularization term, which makes different views to be same to each other.

When we implement the single view method, for each single view features, we run the subspace learning method to get its subspace representation Z_v at first, and then we use $(|Z_v| + |Z_v^T|)/2$ as the affinity matrix to perform spectral clustering. In addition, in our experiments, we initialize F in our method with the result of spectral clustering.

Table 1. Clustering results on Caltech101-7 data set

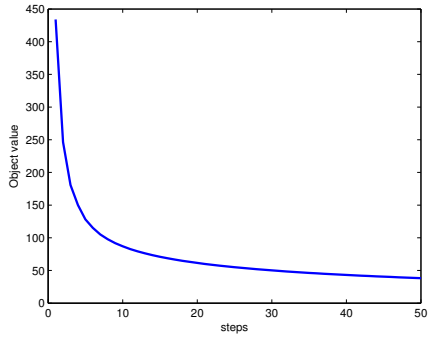
Methods	ACC	NMI
CEN_SSC	0.6390(±0.0257)	0.5592(±0.0219)
CMT_SSC	0.3730(±0.0108)	0.2510(±0.0081)
HOG_SSC	0.6286(±0.0125)	0.5503(±0.0087)
LBP_SSC	0.4113(±0.0061)	0.2922(±0.0053)
SIFT_SSC	0.6735(±0.0001)	0.6418(±0.0054)
CEN_LRR	0.5764(±0.0014)	0.4052(±0.0038)
CMT_LRR	0.3451(±0.0049)	0.2510(±0.0081)
HOG_LRR	0.6417(±0.0001)	0.4686(±0.0028)
LBP_LRR	0.4138(±0.0031)	0.2914(±0.0039)
SIFT_LRR	0.6667(±0.0095)	0.5777(±0.0077)
MLRR_Con	0.3610(±0.0014)	0.2498(±0.0023)
MSSC_Con	0.5605(±0.0018)	0.3786(±0.0071)
MLRR_Add	0.7168(±0.0007)	0.5926(±0.0027)
MSSC_Add	0.7102(±0.0009)	0.5645(±0.0011)
PC-MSC	0.6599(±0.0401)	0.6499(±0.0208)
CC-MSC	0.7188(±0.0420)	0.6768(±0.0170)
MVSC	**0.7415**(±0.0532)	**0.7153**(±0.0151)

4.4. Experiment Results

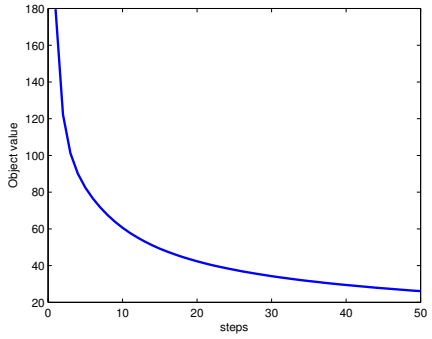
The experiment results of the four dataset are shown in Tables 1, 2, 3 and 4. As shown in these tables, we can see that some individual view features are more discriminative for performing clusters, such as HOG descriptor, while other views are less discriminative, such as CMT descriptor. It is consistent with the Fig. 1 where the block structures of CENTRIST, HOG and SIFT are more clearer than the other

Table 2. Clustering results on MSRCV1 data set

Method	ACC	NMI
CEN_SSC	0.6262(±0.0406)	0.5643(±0.0092)
CMT_SSC	0.3429(±0.0214)	0.2181(±0.0242)
HOG_SSC	0.6190(±0.0441)	0.5268(±0.0236)
LBP_SSC	0.6233(±0.0450)	0.5046(±0.0388)
SIFT_SSC	0.3967(±0.0129)	0.2404(±0.0094)
CEN_LRR	0.4543(±0.0056)	0.3602(±0.0069)
CMT_LRR	0.3305(±0.0056)	0.1212(±0.0045)
HOG_LRR	0.6062(±0.0023)	0.4971(±0.0066)
LBP_LRR	0.5990(±0.0049)	0.4582(±0.0067)
SIFT_LRR	0.4748(±0.0055)	0.2924(±0.0076)
MLRR_Con	0.5071(±0.0075)	0.3942(±0.0060)
MSSC_Con	0.5714(±0.0000)	0.4954(±0.0000)
MLRR_Add	0.6252(±0.0055)	0.5384(±0.0135)
MSSC_Add	0.6343(±0.0129)	0.5415(±0.0118)
PC-MSC	0.6667(±0.0055)	0.5745(±0.0071)
CC-MSC	0.6567(±0.0420)	0.5645(±0.0264)
MVSC	**0.7047**(±0.0102)	**0.5814**(±0.0430)



(a) Caltech101-7



(b) MSRCV1

Figure 4. The convergence curve of our MVSC on Caltech101-7 and MSRCV1 data set

two. In addition, almost all of the multi-view methods outperform the single view method. In addition, our proposed MVSC method can improve the clustering performance apparently compared with single view clustering. At the same time, our MVSC can also outperform the other multi-view methods. Comparing table 1, which has 7 classes, with Table 4, which has 20 classes, our proposed MVSC still has good performance on large scale data set, outperforming the other methods apparently.

To show the advantage of combining multi-view features, we apply our method to increasing-view features. Due to the limitation of the space, we only show the result of MSRCV1 dataset. The result is show in Table 5. Each result is the average of all the corresponding number of views. Apparently, the performance becomes better when increasing the number of views. Additionally, from the deviation we can see that the performance of different views is different dramatically. Some have better performance than others. Thus, combining multiple views is better than using only a certain view features.

Table 3. Clustering results on ETH-80 data set

Method	ACC	NMI
CEN_SSC	0.4990(±0.0139)	0.4832(±0.0197)
CMT_SSC	0.5182(±0.0314)	0.5021(±0.0144)
HOG_SSC	0.5640(±0.5048)	0.6408(±0.0434)
LBP_SSC	0.5048(±0.0425)	0.6059(±0.0276)
SIFT_SSC	0.4832(±0.0392)	0.4681(±0.0128)
CEN_LRR	0.3635(±0.0032)	0.3001(±0.0020)
CMT_LRR	0.2310(±0.0126)	0.1212(±0.0043)
HOG_LRR	0.5540(±0.0021)	0.5952(±0.0015)
LBP_LRR	0.4902(±0.0087)	0.4613(±0.0083)
SIFT_LRR	0.3200(±0.0001)	0.3410(±0.0054)
MLRR_Con	0.3980(±0.0153)	0.3410(±0.0271)
MSSC_Con	0.4672(±0.0036)	0.4213(±0.0052)
MLRR_Add	0.5127(±0.0038)	0.5200(±0.0098)
MSSC_Add	0.5247(±0.0053)	0.5143(±0.0110)
PC-MSC	0.5556(±0.0465)	0.5843(±0.0355)
CC-MSC	0.5512(±0.0619)	0.6147(±0.0335)
MVSC	**0.5650**(±0.0012)	**0.6459**(±0.0139)

In our experiment, the stop criteria is defined as following:

$$\frac{|f^{(t+1)} - f^{(t)}|}{f^{(t)}} < 10^{-2}\,, \tag{24}$$

where $f^{(t)}$ is the objective value in the t-th iteration. In Fig. 4, we show the convergence of our proposed MVSC method. Due to the limitation of the space, we only report the results of Caltech101-7 and MSRCV1 data set. As shown in Fig. 4, the algorithm approaches to convergence quickly.

Table 4. Clustering results on Caltech101-20 data set

Method	ACC	NMI
CEN_SSC	0.5074(±0.0202)	0.5654(±0.0087)
CMT_SSC	0.2481(±0.0042)	0.2840(±0.0054)
HOG_SSC	0.3120(±0.0091)	0.2966(±0.0049)
LBP_SSC	0.2941(±0.0043)	0.3367(±0.0046)
SIFT_SSC	0.2450(±0.0006)	0.2658(±0.0047)
CEN_LRR	0.5158(±0.0092)	0.4732(±0.0094)
CMT_LRR	0.2939(±0.0070)	0.2887(±0.0040)
HOG_LRR	0.3223(±0.0091)	0.2603(±0.0080)
LBP_LRR	0.2904(±0.0052)	0.3058(±0.0056)
SIFT_LRR	0.3312(±0.0039)	0.2974(±0.0038)
MLRR_Con	0.2520(±0.0050)	0.2978(±0.0051)
MSSC_Con	0.3102(±0.0050)	0.3288(±0.0033)
MLRR_Add	0.4617(±0.0079)	0.4457(±0.0052)
MSSC_Add	0.5527(±0.0069)	0.5151(±0.0057)
PC-MSC	0.5714(±0.0182)	0.6164(±0.0121)
CC-MSC	0.5225(±0.0318)	0.5897(±0.0131)
MVSC	**0.6130**(±0.0068)	**0.6532**(±0.0197)

Table 5. Increasing views on MSRCV1 dataset

No.of Views	ACC	NM
One	0.4445(±0.1057)	0.2879(±0.1190)
Two	0.4902(±0.1231)	0.3296(±0.1174)
Three	0.5446(±0.1185)	0.3864(±0.1124)
Four	0.6161(±0.1005)	0.4494(±0.0850)
Five	0.6767(±0.0060)	0.5289(±0.0050)

5. Conclusions

In this paper, we have proposed a novel multi-modal subspace clustering model. To utilize the different modal features, we perform subspace clustering on individual modality respectively and then unify them. Since the magnitude of the subspace representation for different modalities is different, we unify them with a common indicator matrix rather than a common subspace representation. Thus, the proposed method can guarantee the data points in different modalities to be classified in the same cluster. The experiments show that our algorithm outperform other state-of-the-art algorithm apparently.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.

[2] X. Cai, F. Nie, and H. Huang. Multi-view k-means clustering on big data. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2598–2604. AAAI Press, 2013.

[3] X. Cai, F. Nie, H. Huang, and F. Kamangar. Heterogeneous image features integration via multi-modal spectral clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, pages 1977–1984, 2001.

[4] N. Chen, J. Zhu, and E. P. Xing. Predictive subspace learning for multi-view data: a large margin approach. In *Advances in neural information processing systems*, pages 361–369, 2010.

[5] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[7] D. Dueck and B. J. Frey. Non-metric affinity propagation for unsupervised image categorization. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[8] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2765–2781, 2013.

[9] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[10] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007.

[11] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–11. IEEE, 2003.

[12] K. Kanatani. Motion segmentation by subspace separation and model selection. *IEEE International Conference on Computer Vision*, 1:1, 2001.

[13] A. Kumar, P. Rai, and H. Daume. Co-regularized multi-view spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1413–1421, 2011.

[14] Y. J. Lee and K. Grauman. Foreground focus: Unsupervised learning from partially matching images. *International Journal of Computer Vision*, 85(2):143–166, 2009.

[15] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–409. IEEE, 2003.

[16] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):171–184, 2013.

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[18] D. Luo, F. Nie, C. Ding, and H. Huang. Multi-subspace representation and discovery. In *European Conference, ECML PKDD 2011*, pages 405–420. Springer, 2011.

[19] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[20] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482, 1999.

[21] P. Tseng. Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105(1):249–252, 2000.

[22] H. Wang, F. Nie, and H. Huang. Multi-view clustering and feature learning via structured sparsity. *The 30th International Conference on Machine Learning (ICML 2013), Journal of Machine Learning Research, W&CP*, 28(3):352–360, 2013.

[23] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 756–763. IEEE, 2005.

[24] J. Wu and J. M. Rehg. Centrist: A visual descriptor for scene categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1489–1501, 2011.

[25] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Computer Vision–ECCV 2006*, pages 94–106. Springer, 2006.

[26] H. Yu, M. Li, H.-J. Zhang, and J. Feng. Color texture moments for content-based image retrieval. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 3, pages 929–932. IEEE, 2002.