Abstract

Subspace clustering is to find underlying low-dimensional subspaces and cluster the data points correctly. In this paper, we propose a novel multi-view subspace clustering method. Most existing methods suffer from two critical issues. First, they usually adopt a two-stage framework and isolate the processes of affinity learning, multi-view information fusion and clustering. Second, they assume the data lies in a linear subspace which may fail in practice as most real-world datasets may have non-linearity structures. To address the above issues, in this paper we propose a novel Enriched Robust Multi-View Kernel Subspace Clustering framework where the consensus affinity matrix is learned from both multi-view data and spectral clustering. Due to the objective and constraints which is difficult to optimize, we propose an iterative optimization method which is easy to implement and can yield closed solution in each step. Extensive experiments have validated the superiority of our method over state-of-the-art clustering methods.

1. Introduction

In machine learning, high-dimensional data are ubiquitous. For example, images may consist of thousands of pixels and text data may have tens of features. High dimensionality requires demanding computational time and memory, and moreover, noise in the data can bring adversely influence on performance. Fortunately, recent research shows that high-dimensional data often lies in low-dimensional structures. For instance, the set of face images under all possible illumination conditions can be well approximated by a 9-dimensional linear subspace \[2\]. Recovering the low-dimensional structures of data can not only save computational cost, but also will improve the accuracy and effectiveness of learning methods. For data samples lie in low-dimensional subspaces instead of being uniformly distributed across ambient space, subspace clustering is to separate data according to their underlying subspaces and the basis for each subspace \[45\]. For the past decade, subspace clustering has been explored actively and applied in many applications such as image/motion/video segmentation \[16, 50, 51\], image representation \[25, 60\], etc.

Subspace clustering approaches have been developed and studied extensively, and among them are: iteration-based methods such as \[44, 58\] which alternates cluster assignment and subspace fitting; factorization-based algebraic approaches such as \[34, 46\] which hypothesizes that the subspaces are independent; statistical approaches such as Multi-stage Learning \[17\], Mixtures of Probabilistic PCA \[43\] which alternates between clustering and subspace estimation via Expectation Maximization; spectral clustering based approaches such as Local Subspace Affinity \[50\], Locally Linear Manifold Clustering \[16\] where data segmentation is obtained from spectral clustering. More recently, sparse subspace clustering (SSC) has been proposed \[11, 37, 38\] to find a sparse representation corresponding to the data points from the same subspace.

In the big-data era, many computer vision problems are fed with the dataset represented by multiple feature sets, which is so called ‘multi-view’ data. Different descriptors characterize various and independent information from different perspectives. For instance, an image can be described by color, texture, histogram of oriented gradients (HOG), local binary pattern (LBP), etc. These different features can provide useful information from different views to improve clustering performance \[32\]. Multi-view clustering is to integrate these multiple feature sets together to perform reliable clustering. Most existing multi-view subspace clustering methods integrate multi-view information in similarity or representation by merging multiple graphs or representation matrices into a shared one. For example, \[18, 42\] learn a shared sparse subspace representation by performing matrix factorization. Similarly, centroid-based multi-view low-rank sparse subspace clustering methods \[5, 33, 55\] induce low-rank and sparsity constraints on the shared affinity matrix across different views. Instead of obtaining a shared representation directly, Hilbert-Schmidt Independence Criterion (HSIC) and Markov chain are introduced to learn complementary subspace representations, followed by adding them together appropriately \[7, 48\].
2.1. Sparse Subspace Clustering

Given \( n \) data points \( X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^{d \times n} \), subspace clustering assumes that each data point can be approximated by a linear combination of dataset samples \([11]\):

\[
X = XC + E,
\]

where \( C = \{c_1, c_2, \ldots, c_n\} \in \mathbb{R}^{k \times n} \) is the subspace representation matrix, with each \( c_i \) representing the original data point \( x_i \) based on the subspace. \( E \in \mathbb{R}^{d \times n} \) is the error matrix.

Sparse subspace clustering formulates the objective as:

\[
\min_C \|X-XC\|_F^2 + \theta \|C\|_1, \quad s.t. \ diag(C) = 0, \ C^T 1 = 1,
\]

where \( \| \cdot \|_F \) denotes the Frobenius norm while \( \|C\|_1 = \sum_{i,j} |C_{ij}| \). The constraint \( C^T 1 = 1 \) indicates that the data point lies in a union of affine subspaces while the constraint \( diag(C) = 0 \) rules out the case that a data point is represented by itself, which hints that each data point \( x_i \) can only be represented by a combination of other points \( x_j (j \neq i) \).

Solving the optimization problem in Eq. (2), we will get the representation \( c_i \) for each data point \( x_i \).

After obtaining the subspace structure, we construct the affinity matrix by setting \( W = \frac{|C^T C|}{2} \). Therefore, we can perform spectral clustering on subspace affinity matrix:

\[
\min_F \text{tr}(F^T LF), \quad s.t. \ F^T F = I,
\]

where \( F \) is the cluster indicator matrix, \( L := D - W \) where \( D \) is a diagonal matrix given by \( D(i,i) = \sum_j W(i,j) \).

2.2. Robust Multi-View Kernel Subspace Clustering

Given the \( v \)-view dataset \( X^{(v)} \in \mathbb{R}^{d_v \times n} \), if we perform the subspace learning on each single view, we can get the subspace representation \( C^{(v)} \) for the \( v \)-th view. The fundamental challenge boils down to combine multi-view features in subspace clustering. An intuitive and naive method is to concatenate all the features together and perform clustering on the concatenated features, where the more informative view and the less informative one will be treated equally. Therefore, the solution is inevitably not optimal in many scenarios. In contrast, one can perform the clustering on each single view followed by fusing them together. In order to combine multi-view sparse subspace clustering results, we can perform the subspace learning on different views simultaneously by solving:

\[
\min_{C^{(v)}, C^*} \sum_v \left( \|X^{(v)} - X^{(v)}C^{(v)}\|_F^2 + \theta \|C^{(v)}\|_1 \right) + \lambda \|C^* - C^*\|_F^2, \quad s.t. \ diag(C^{(v)}) = 0, \ C^{(v)}^T 1 = 1
\]

where \( C^* \) is the consensus affinity matrix across multiple views and spectral clustering will be performed based on it.
However, experiments have demonstrated that $C^{(v)}$ can be significantly different. One can see that if a certain view $C^{(v)}$ is not learned well, to minimize the objective, $C^*$ will deviate from optimal solution due to the squared Frobenius norm which is known to be sensitive to noise/outliers. In contrast, the robust formulation: $\min_{C^{(v)}, C^*} \sum_v \|X^{(v)} - X^{(v)}C^{(v)}\|_F^2 + \theta \|C^{(v)}\|_1 + \lambda \|C^{(v)} - C^*\|_1$, s.t. $diag(C^{(v)}) = 0, C^{(v)T}1 = 1$. (5)

On the other hand, kernel tricks have been played in various classification/algorithmic such as PCA [35], SVM [41], K-means [9], etc. Those kernel version methods yield very promising results especially when the data in original space is not well separable, but can be separated by projecting into a higher dimensional space via $\Phi(\cdot)$ where $\Phi: \mathbb{R}^d \rightarrow \mathbb{R}^m (m > d)$ [14]. Therefore, we introduce the robust multi-view kernel subspace clustering by optimizing:

$$\min_{C^{(v)}, C^*} \sum_v \|\Phi(X^{(v)}) - \Phi(X^{(v)})C^{(v)}\|_F^2 + \theta \|C^{(v)}\|_1 + \lambda \|C^{(v)} - C^*\|_1, \text{s.t. } diag(C^{(v)}) = 0, C^{(v)T}1 = 1.$$ (6)

### 2.3. Enriched Multi-View Subspace Clustering

Most existing multi-view subspace learning will do spectral clustering after obtaining $C^*$ which ignores the potential connection between the two stages. As a contribution of this paper, we propose an enriched procedure by combining the learning with clustering stage via:

$$\min_{C^{(v)}, C^*, F} \sum_v \|\Phi(X^{(v)}) - \Phi(X^{(v)})C^{(v)}\|_F^2 + \theta \|C^{(v)}\|_1 + \lambda \|C^{(v)} - C^*\|_1 + \gamma \text{tr}(F^T LF), \text{s.t. } F^T F = I, \text{diag}(C^{(v)}) = 0, C^{(v)T}1 = 1.$$ (7)

It is worth noting that here $L$ is constructed based on the affinity matrix $W$ from $C^*$ instead of $C^{(v)}$ in each view. Therefore, different from Eq. (6) which learns $C^*$ from each view, Eq. (7) also learns $C^*$ from spectral clustering.

The proposed optimization model consists of two parts. The first part is the intra-view structure learning, which aims to learn the subspace structure in each view. The second part is the inter-view consistency learning, which measures the correlation across different views. By exploring both the view-specific property and view-consistency across multiview data, our unified model can learn both the intra-view subspace structure and common cluster structure simultaneously. In this way, the proposed method can achieve the optimal consensus affinity matrix across multiple views that produces promising clustering results. Fig. 2 shows the main framework of the proposed method.

### 3. Optimization

Considering the constraints and non-differential property of the above objective for $C^{(v)}$, we propose an updating algorithm based on Alternating Direction Method of Multipliers (ADMM) [3, 19, 28]. By introducing $A^{(v)} = C^{(v)} \in \mathbb{R}^{d \times k}$, we have: $\Phi(X^{(v)}) = X^{(v)}C^{(v)} \in \mathbb{R}^{m \times k}$ and $\Phi(X^{(v)}) = X^{(v)}C^{(v)} \in \mathbb{R}^{m \times k}$.

Figure 2. The framework of our proposed method.
\[ \mathbb{R}^{n \times n}, \text{we reformulate the objective as:} \]
\[ \min_{A^{(v)},C^{(v)},C^*,F} \sum_v \| \Phi(X^{(v)}) - \Phi(X^{(v)})C^{(v)} \|_F^2 + \theta \| A^{(v)} \|_1 \]
\[ + \lambda \| A^{(v)} - C^* \|_1 + \gamma \text{tr}(F^TLF), \]
\[ \text{s.t. } F^TF = I, \text{diag}(C^{(v)}) = 0, C^{(v)T}1 = 1, A^{(v)} = C^{(v)}. \tag{8} \]

The corresponding augmented Lagrangian function is:
\[ L(A^{(v)},C^{(v)},C^*,F,\delta^{(v)},\Sigma^{(v)}) = \sum_v \| \Phi(X^{(v)}) - \Phi(X^{(v)})C^{(v)} \|_F^2 + \lambda \| A^{(v)} - C^* \|_1 \]
\[ + \frac{\rho}{2} \| C^{(v)T}1 - 1 \|_2^2 + \langle \delta^{(v)}, C^{(v)T}1 - 1 \rangle + \theta \| A^{(v)} \|_1 \]
\[ + \frac{\rho}{2} \| C^{(v)} - A^{(v)} + \text{diag}(A^{(v)}) \|_F^2 + \gamma \text{tr}(F^TLF) \]
\[ + \langle \Sigma^{(v)}, C^{(v)} - A^{(v)} + \text{diag}(A^{(v)}) \rangle, \text{ s.t. } F^TF = I, \tag{9} \]

where \( \delta \in \mathbb{R}^n, \Sigma \in \mathbb{R}^{n \times n} \) are the Lagrangian Multipliers.

3.1. Updating \( F \)

When fixing \( C^* \), \( L \) is fixed and \( F \) can be optimized via:
\[ \min_F \text{tr}(F^TLF), \text{ s.t. } F^TF = I, \tag{10} \]
where \( L = D - W \) and \( W = [C^* + |C^*|^T]^T \). Apparently the solutions are the eigenvectors corresponding to the smallest \( k \) eigenvalues of the Laplacian matrix \( L \) where \( k \) is the number of clusters [29].

3.2. Updating \( A^{(v)} \)

For optimized \( C^* \) in each view, it is obtained via:
\[ \min_j \beta \| J \|_1 + \alpha \| J - C^* \|_1 + \frac{1}{2} \| J - Y \|_F^2, \tag{11} \]
with \( A = J - \text{diag}(J) \), where \( Y = C + \Sigma, \beta = \frac{\rho}{2}, \alpha = \lambda \). Apparently, \( \text{diag}(A) = 0 \) and the above equation can be solved through element-wise optimization:
\[ \min_j \beta |j| + \alpha |j - c^*| + \frac{1}{2} (j - y)^2. \tag{12} \]

Due to space limit, we provide the closed solution for \( c^* \neq 0 \) (otherwise, it degenerates into well known standard soft-thresholding with \( j^+ = \text{sgn}(y) \max\{|y| - \alpha + \beta, 0| \} \) by leaving the details to supplemental file:
\[ j^* = \begin{cases} y - \alpha + \beta, & \text{if } c^* > 0 \land y \geq \alpha + \beta + c^*; \\
\alpha + \beta, & \text{if } c^* > 0 \land y < \alpha + \beta - c^*; \\
y + \alpha + \beta, & \text{if } c^* > 0 \land y \geq y + \alpha + \beta; \\
y - \alpha - \beta, & \text{if } c^* < 0 \land y \geq \alpha + \beta; \\
y - \alpha + \beta, & \text{if } c^* < 0 \land y < \alpha - \beta + c^*; \\
y + \alpha + \beta, & \text{if } c^* < 0 \land y \geq y + \alpha + \beta; \\
0, & \text{else,} \end{cases} \tag{13} \]

where ‘\( \land \)’ denotes logical conjunction.

3.3. Updating \( C^{(v)} \)

For optimized \( C \) in each view, it can be obtained via (by skipping \( \text{diag}(A) \) as it is 0 aforementioned):
\[ \min_{C^v} \| \Phi(X) - \Phi(X)C^v \|_F^2 + \frac{\gamma}{2} \| C^vT1 - 1 \|_2^2 + \frac{\rho}{2} \| C - A + \Sigma \|_F^2. \tag{14} \]

By taking the derivative and set it to 0, we have:
\[ C = (\mathcal{K} + \rho I + \rho 11^T)^{-1} (\mathcal{K} + \rho 11^T - 1\delta^T + \rho A - \Sigma), \tag{15} \]
where \( \mathcal{K} = \Phi(X)^T \Phi(X) \). One can see that with different kernel chosen, \( \mathcal{K} \) is different but always is computationally efficient. For example, when polynomial kernel is applied, then \( \mathcal{K}(i,j) = \langle (x_i, x_j) + c \rangle^d \).

3.4. Updating \( C^* \)

As \( C^* \) is enriched, which is related with 2 terms, it can be optimized via:
\[ \min_{C^*} \gamma \text{tr}(F^TLF) + \sum_v \lambda \| A^{(v)} - C^* \|_1. \tag{16} \]

Before we optimize the above equation, we first introduce a useful lemma which is critical for \( C^* \):

\textbf{Lemma 1.} For Laplacian matrix \( L \) and the matrix \( F \), we have [6]:
\[ \text{tr}(F^TLF) = \frac{1}{2} \sum_{i,j} W(i,j) \| f_i - f_j \|_2^2. \tag{17} \]

We turn to optimize \( C^* \) by noticing the above equation can be written in a more compact formulation: \( \text{tr}(F^TLF) = \frac{1}{2} \langle |C^*|, Q \rangle \), where \( Q \) is symmetric and \( Q(i,j) = \| f_i - f_j \|_2^2 \). On the other hand, by definition \( W = |C^*| + |C^*|^T \), by simple algebraic operation we have:
\[ \text{tr}(F^TLF) = \frac{1}{2} \langle |C^*|, Q \rangle. \tag{18} \]

Therefore, \( C^* \) can be optimized by:
\[ \min_{\frac{C^*}{2}} \| C^* \|, Q \sum_v \lambda \| A^{(v)} - C^* \|_1. \tag{19} \]

Similar to \( A \), we can optimize \( C^* \) by element-wise:
\[ \min_{\frac{C^*}{2}} \gamma |c^*| + \sum_v 2\lambda |a^{(v)} - c^*|. \tag{20} \]

---

1We note that for Linear Kernel case, which is \( \Phi(X) = X \in \mathbb{R}^{d \times n} \) and \( \mathcal{K}(X,X) = X^T X \). When \( n \gg d \), we have accelerated updating algorithm for inversion calculation. First we denote \( X^T X + \rho I + \rho 11^T = Z^T Z + \rho I, \) where \( Z = [X; \sqrt{\rho} 1^T] \in \mathbb{R}^{(d+1) \times n} \). Then by matrix inversion lemma (aka Sherman-Morrison-Woodbury Formula), \( (Z^T Z + \rho I)^{-1} = \rho^{-1} I_n - \rho^{-1} Z^T (I_{d+1} + \rho ZZ^T)^{-1} Z \), the complexity can be reduced from \( \mathcal{O}(n^3) \) to \( \mathcal{O}(d^3 + dn^2) \) which is a significant improvement for \( n \gg d \).
Algorithm 1 Algorithm for Enriched Robust Multi-View Kernel Subspace Clustering to solve Eq. (7).

Input: data $X^{(v)} \in \mathbb{R}^{d \times n}$, number of clusters $k$, regularization parameters $\lambda, \gamma, \theta$, number of iterations $T$.
Initialization: $C^{(v)}, \Sigma^{(v)}, A^{(v)}, C^* \in \mathbb{R}^{n \times n}, \delta^{(v)} \in \mathbb{R}^n, \rho = 0.2, t = 1$.

while $t \leq T$ do
  Optimize $F$ by solving Eq. (10);
  Optimize $A$ in each view by solving Eq. (11);
  Optimize $C$ in each view by solving Eq. (14);
  Optimize each $C^*$ by solving Eq. (19);
  Update $\delta, \Sigma$ in each view as Eq. (22);
  Update $\rho = 1.2\rho$;
  $t = t + 1$.
end while

Output: $F_v$, based on which $K$-means will be conducted after row normalization.

Without loss of generality, we sort $[a^{(1)}, a^{(2)}, \ldots, a^{(v)}]$ in non-decreasing order as $[a_1, a_2, \ldots, a_v]$ and none is zero (or it can be transferred into this case by simple operation). Due to space limit, we leave the derivative details to supplemental file and directly give the solution:

$$
c^* = \begin{cases} 
    a \left[ \frac{2\lambda - 2\gamma}{\lambda + \gamma} \right], & \text{if } 2\nu\lambda > \gamma q \wedge a \left[ \frac{2\lambda - 2\gamma}{\lambda + \gamma} \right] > 0; \\
    a \left[ \frac{2\lambda + 2\gamma}{\lambda - \gamma} \right], & \text{if } 2\nu\lambda > \gamma q \wedge a \left[ \frac{2\lambda + 2\gamma}{\lambda - \gamma} \right] < 0; \\
    0, & \text{else},
\end{cases}
$$

(21)

where $[\cdot]$ denotes the ceiling function.

3.5. Updating Lagrangian Multipliers in Each View

Following ADMM framework [3], we can simply update Lagrangian Multipliers by gradient ascent:

$$
\delta^{(v)} = \delta^{(v)} + \rho(C^{(v)}T - 1),
$$

$$
\Sigma^{(v)} = \Sigma^{(v)} + \rho(C^{(v)} - A^{(v)}).
$$

(22)

We summarize the above algorithm in Alg. 1.

4. Experiments

In this section, we will evaluate our proposed algorithm on several widely used benchmark datasets to illustrate its potential in multi-view clustering.

Six benchmark datasets are used in the experiment, including MSRC-v1, UCI Handwritten digits [10], Caltech101-7 [13], Caltech101-20 [13], ORL [40] and Yale [1]. For each dataset, multiple feature sets are available to describe the images from various aspects. The detailed information is summarized in Table 3.

Throughout the experiments, we use Matlab R2019a on a laptop with 1.4 GHz QuadCore Intel Core i5 processor. The clustering quality is measured by clustering accuracy (ACC), which is the percentage of items correctly clustered with the maximum bipartite matching [49], and normalized mutual information (NMI) [23, 30]. We repeat each experiment 10 times and report the average performance with the standard deviation.

4.1. Feature Descriptions

Features adopted in this paper is shown as the following, each feature captures quite different information from images:

1. CENTRIST [47] stands for census transform histogram, is a holistic representation of images, which can be applied to capture the structural and textural properties from images.

2. HOG [8] is based on oriented gradients, so it has great power to capture edge structures in images naturally.

3. Color moment (CMT) [53] represents the color distribution in images. The mathematical basis of this descriptor is that the color distribution can be represented efficiently by some low-order moments.

4. Local binary pattern captures the texture information from an image by computing the histogram of local binary patterns [12].

5. GIST [39] is a global image feature. Gist features represent scene information from images well. It relates to the gradient information for different parts in an image, including scales and orientations.

6. Gabor, which is extracted by a Gabor filter and can do texture analysis and object detection in images.

7. Intensity (IT). Pixel intensity is the primary information stored within pixels, represents the densities of a certain pixel.

4.2. Experiment Setup

To evaluate the performance of our method, we compare our method with two subspace learning algorithms applied on single view: spectral clustering (SC) [36] and lower rank representation (LRR) [26], and five state-of-the-art multi-view methods including: pairwise co-regularized multi-view spectral clustering (P-CoReg) [22], centroid co-regularized multi-view spectral clustering (C-CoReg) [22], robust multi-view spectral clustering via low-rank and sparse decomposition (RMSC) [48], multi-view consensus
graph clustering (MCGC) [54], and multi-view clustering via joint nonnegative matrix factorization (MNMF) [27]. Detailed description about the methods mentioned above and the experiment process is as the following:

1. Single view with SC. We run spectral clustering on each view-specific affinity matrix independently to get clustering results based on different features.

2. Single view with LRR. We run LRR on each single feature set to get the low-rank subspace representation first, and then apply spectral clustering on each such
Table 2. Subspace clustering results on benchmark datasets

<table>
<thead>
<tr>
<th>Method</th>
<th>Yale</th>
<th>ORL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC</td>
<td>NMI</td>
</tr>
<tr>
<td>IT-SC</td>
<td>0.271±0.026</td>
<td>0.256±0.031</td>
</tr>
<tr>
<td>LBP-SC</td>
<td>0.617±0.036</td>
<td>0.637±0.017</td>
</tr>
<tr>
<td>Gabor-SC</td>
<td>0.653±0.012</td>
<td>0.643±0.011</td>
</tr>
<tr>
<td>IT-LRR</td>
<td>0.153±0.022</td>
<td>0.101±0.007</td>
</tr>
<tr>
<td>LBP-LRR</td>
<td>0.631±0.019</td>
<td>0.602±0.022</td>
</tr>
<tr>
<td>Gabor-LRR</td>
<td>0.625±0.023</td>
<td>0.637±0.018</td>
</tr>
<tr>
<td>P-CoReg</td>
<td>0.635±0.015</td>
<td>0.667±0.037</td>
</tr>
<tr>
<td>C-CoReg</td>
<td>0.655±0.009</td>
<td>0.637±0.011</td>
</tr>
<tr>
<td>RMSC</td>
<td>0.630±0.012</td>
<td>0.644±0.014</td>
</tr>
<tr>
<td>MCGC</td>
<td>0.649±0.035</td>
<td>0.533±0.071</td>
</tr>
<tr>
<td>MNMF</td>
<td>0.564±0.031</td>
<td>0.571±0.027</td>
</tr>
<tr>
<td>Our</td>
<td><strong>0.668±0.012</strong></td>
<td><strong>0.672±0.009</strong></td>
</tr>
</tbody>
</table>

3. P-CoReg, which makes the eigenvector matrix in standard spectral clustering method related to different views be close to each other, by employing pair-wise co-regularizers in the objective function.

4. C-CoReg, similar to P-CoReg, it regularizes the eigenvectors related to each specific view feature towards a consensus set.

5. RMSC, which is based on Markov chain method for clustering. A shared low-rank transition probability matrix is used as a crucial input to the standard Markov chain method for clustering.

6. MCGC, where a consensus graph structure is learned by minimizing disagreement between diverse views and constraining the rank of the Laplacian matrix, it’s able to obtain the cluster assignment directly from the consensus graph without any post-processing steps.

7. MNMF, a joint nonnegative matrix factorization algorithm to regularize coefficient matrices learnt from different views towards a consensus, followed by K-means on the consensus matrix.

4.3. Experiment Results

Fig. 3 shows the subspace representation matrices \( C \) obtained by different feature descriptors and the final consensus \( C^* \) of the MSRC-v1 dataset. A good \( C \) should have a clear block diagonal structure, since the data in the MSRC-v1 dataset is grouped by object classes. In Fig. 3, view-specific \( C \) vary a lot from each other since they are capturing different characteristic from images. And for some of
Table 3. Datasets information and available feature sets

<table>
<thead>
<tr>
<th>Dataset</th>
<th># images</th>
<th># classes</th>
<th>HOG</th>
<th>CENTRIST</th>
<th>Color Moment</th>
<th>LBP</th>
<th>GIST</th>
<th>Intensity</th>
<th>Gabor</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRC-v1</td>
<td>210</td>
<td>7</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Handwritten</td>
<td>2000</td>
<td>10</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Caltech101-7</td>
<td>1474</td>
<td>7</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Caltech101-20</td>
<td>2386</td>
<td>20</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>ORL</td>
<td>400</td>
<td>40</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Yale</td>
<td>165</td>
<td>15</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

them, there is no obvious block structure, noise exists over the whole matrix. It is apparent that only relying on one single view-specific \( C \) has a high chance to achieve poor result. But for the consensus \( C^* \), the block diagonal structure is well-established and there is almost no noise off the diagonal blocks, which means each sample is well represented by the remaining data from the same object class, thus a great clustering result can be obtained based on it. We also utilize polynomial kernels to further improve the subspace learning performance. Fig. 5 shows the consensus \( C^* \) obtained with linear kernel and polynomial kernel for the MSRC-v1 dataset, please zoom in to observe the details and differences. The block diagonal structure gets more recognizable with an appropriate polynomial kernel. Thus on complex datasets where nonlinear relationships exist, polynomial kernels can have superior performance compared to simple linear kernel. And from Fig. 6 we can see that the objective of the ADMM solver is converging as the iteration increases.

Experiment results on six datasets are shown in Table 1 and 2, highest ACC and NMI for each dataset is highlighted. From the results it’s not hard to conclude that certain view-specific \( C \) cannot have a satisfying clustering performance; this may be caused by the fact that images from different clusters have great similarity in the characteristic captured by that view, for example, the color moment feature doesn’t work well on the Handwritten dataset. But with multi-view clustering methods, independent feature sets are combined together to construct a view-consistent \( C^* \), the clustering performance is greatly improved. What’s more, in our proposed method, the consensus \( C^* \) will not deviate from the optimal solution when a view-specific \( C \) is not well learned, so our proposed method can achieve best clustering performance in most cases over the comparison methods.

In addition, we investigate the performance of our method with varying parameter settings. There are three important parameters in our method: \( \lambda, \gamma, \) and \( \theta \). We explore the effect of two parameters by fixing another. We present the results on MSRC-v1 dataset as Fig. 4 demonstrates. From the figure we see that with the setting of \( \lambda \) in the range of \((0.001, 10)\), \( \gamma \) in the range of \((0.001, 10)\) and \( \theta \) in the range of \((0.001, 1)\), promising performance can be achieved. To show the advantage of combining multi-view feature sets in subspace clustering, we run our proposed method with increasing number of views on four datasets. The result is averaged on all the possible combinations of views, and for each combination we run the experiment 5 times. Comparison is shown in Fig. 7. Apparently subspace clustering performance is improved significantly as number of views increases for all the datasets, since the data is described in a more comprehensive and extensive way.

**5. Conclusion**

In this paper, we propose an *Enriched Robust Multi-View Kernel Subspace Clustering* model. Different from most existing multi-view clustering methods, our method obtains an enriched consensus affinity matrix from both the learning and clustering stages. Besides, the proposed method extends linear space to kernel space to capture the nonlinear structure hidden in the multi-view data. To optimize the objective with various constraints, we propose ADMM to obtain the optimal solution where in each step a closed solution is provided. Extensive experimental results on six benchmark datasets have demonstrated the superiority of our method over several SOTA clustering methods.
References


[6] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.


[34] Yi Ma, Allen Y Yang, Harm Derksen, and Robert Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. SIAM review, 50(3):413–458, 2008. 1


[38] Xi Peng, Lei Zhang, and Zhang Yi. Scalable sparse subspace clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 430–437, 2013. 1


[57] Pei Zhang, Xinwang Liu, Jian Xiong, Sihang Zhou, Wentao Zhao, En Zhu, and Zhiping Cai. Consensus one-step multi-view subspace clustering. IEEE Transactions on Knowledge and Data Engineering, 2020. 2

