From Ensemble Clustering to Multi-View Clustering

Zhiqiang Tao\(^1\), Hongfu Liu\(^1\), Sheng Li\(^1\), Zhengming Ding\(^1\) and Yun Fu\(^{1,2}\)

\(^1\)Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA
\(^2\)College of Computer and Information Science, Northeastern University, Boston, MA 02115, USA

\{zqtao, hfliu, shengli, allanding, yunfu\}@ece.neu.edu

Abstract

Multi-View Clustering (MVC) aims to find the cluster structure shared by multiple views of a particular dataset. Existing MVC methods mainly integrate the raw data from different views, while ignoring the high-level information. Thus, their performance may degrade due to the conflict between heterogeneous features and the noises existing in each individual view. To overcome this problem, we propose a novel Multi-View Ensemble Clustering (MVEC) framework to solve MVC in an Ensemble Clustering (EC) way, which generates Basic Partitions (BPs) for each view individually and seeks for a consensus partition among all the BPs. By this means, we naturally leverage the complementary information of multi-view data in the same partition space. Instead of directly fusing BPs, we employ the low-rank and sparse decomposition to explicitly consider the connection between different views and detect the noises in each view. Moreover, the spectral ensemble clustering task is also involved by our framework with a carefully designed constraint, making MVEC a unified optimization framework to achieve the final consensus partition. Experimental results on six real-world datasets show the efficacy of our approach compared with both MVC and EC methods.

1 Introduction

Multi-View learning benefits from leveraging the complementary information from multiple views of a particular dataset, where these views can be obtained by various sensors or represented with different descriptors. For example, we may capture human activity from RGB video cameras, depth cameras, and on-body sensors [Li et al., 2016]; for vision tasks, images could be encoded by host of handcrafted and deep features. Some interesting multi-view problems include subspace learning [Ding and Fu, 2014], outlier detection [Li et al., 2015a; Zhao and Fu, 2015], cross-domain learning [Wang et al., 2017], and incomplete multi-view case [Zhao et al., 2016]. In this paper, we focus on the Multi-View Clustering (MVC) problem. Considerable research efforts have been made to solve the MVC problem, such as optimizing certain loss function with concatenated multi-view features [Bickel and Scheffer, 2004; Kumar and III, 2011], conducting traditional clustering on a common low-dimensional latent subspace [Chaudhuri et al., 2009; Xia et al., 2014; Wang et al., 2016; Zhao et al., 2017], and late fusion approaches [Greene and Cunningham, 2009; Bruno and Marchand-Maillet, 2009].

The key problem for MVC is how to integrate the complementary information from multiple views. For instance, [Kumar et al., 2011] enhanced the similarity of eigenvectors learnt from different views and integrated it within a spectral clustering framework; [Liu et al., 2013b] designed a joint matrix factorization method and sought for a compatible solution of multi-view data. However, the noises of each single-view data can seriously degrade the performance of these pioneering works. In light of this, [Xia et al., 2014] learned a consensus affinity graph among multiple views and handled the view-specific noises via a low-rank and sparse decomposition framework. Along this line, [Wang et al., 2016] captured the local manifold structure in each view and achieved the clustering agreement by minimizing the difference across views. All these MVC methods directly integrate the raw multi-view data, which, however, is not an easy task due to the distinct gap among heterogeneous feature spaces. Thus, one promising way (i.e. late fusion methods) for solving MVC is to first transform multi-view data into the same partition space [Greene and Cunningham, 2009; Bruno and Marchand-Maillet, 2009], and then obtain the clustering result in an ensemble clustering manner.

Ensemble Clustering (EC) [Strehl and Ghosh, 2003; Fred and Jain, 2005] methods take as input a set of Basic Partitions (BPs) and integrate multiple BPs into a consensus one. Thus, it naturally has the ability of leveraging complementary information from heterogeneous sources [Wu et al., 2015]. Nevertheless, ensemble clustering draws little attention in the field of multi-view clustering, and it neglects to explore the connection among different views. Moreover, most existing EC methods also suffer from the noises in the multi-view BPs, which could be caused by the intra-view or inter-view disagreement among all the BPs [Tao et al., 2016].

To address the above challenges, we propose a novel Multi-View Ensemble Clustering (MVEC) algorithm in this paper. Specifically, we first generate a group of BPs for each single view, and summarize each group as a view-specific co-
association matrix \cite{Fred and Jain, 2005}, which works as a pairwise affinity matrix upon the categorical data. After that, we employ low-rank and sparse decomposition to seek for the consensus affinity matrix shared by all the views and compensate the disagreement among all the BPs. Meanwhile, the spectral ensemble clustering task \cite{Liu et al., 2015, Liu et al., 2017} is also involved by our MVEC framework with a carefully designed constraint, leading to a unified optimization framework to jointly integrate multi-view information and find the consensus partition.

MVEC is inspired by two previous works \cite{Xia et al., 2014, Tao et al., 2016}. Compared with \cite{Xia et al., 2014}, we solve MVC in a partition space rather than using raw features, since it is more reasonable to uncover the cluster structure shared by multi-view co-association matrices. Moreover, different from \cite{Xia et al., 2014} that only learns a low-rank matrix, our approach simultaneously performs the tasks of low-rank representation learning and spectral graph partitioning. On the other side, \cite{Tao et al., 2016} focuses on the single-view case and thus cannot make full use of the multi-view data. In contrast, our MVEC explicitly utilizes the connection between different views and employs the learned consensus partition to iteratively enhance the cluster structure of each view.

The contributions of this work are summarized as follows: (1) A general Multi-View Ensemble Clustering (MVEC) framework is proposed, which effectively exploits the multi-view basic partitions to solve the MVC problem. (2) We jointly learn the view-consensus affinity matrix and find the consensus partition within a unified optimization framework. (3) A novel self-boost constraint is designed to iteratively improve the clustering performance.

2 Multi-View Ensemble Clustering

2.1 Problem Formulation

Given a set of \( n \) data points with \( m \) views \( i.e., \) feature representations or modalities, we denote the dataset of each view as \( \mathcal{X}^{(v)} = \{x_1^{(v)}, \ldots, x_n^{(v)}\} \), \( 1 \leq v \leq m \). For \( \forall v \), we assume \( \mathcal{X}^{(v)} \) is sampled from \( K \) crispy clusters, denoted as \( \mathcal{C} = \{C_1, \ldots, C_K\} \). Let \( \Pi^{(v)} = \{\pi_1^{(v)}, \ldots, \pi_r^{(v)}\} \) be a group of \( r \) basic partitions \( \text{BPs} \) for \( \mathcal{X}^{(v)} \), where each BP \( \pi_i^{(v)} \) partitions \( \mathcal{X}^{(v)} \) into \( K_i \) clusters, \( i.e., \pi_i^{(v)} = \{\pi_i^{(v)}(x_1^{(v)}), \ldots, \pi_i^{(v)}(x_n^{(v)})\} \) is a set of categorical data, \( 1 \leq \pi_i^{(v)}(x_j^{(v)}) \leq K_i \), \( 1 \leq i \leq r \), and \( 1 \leq j \leq n \). For each view, we obtain \( \Pi^{(v)} \) by using the Random Parameter Selection \( \text{RPS} \) strategy \cite{Fred and Jain, 2005, Wu et al., 2015}, which performs K-means algorithm on \( \mathcal{X}^{(v)} \) \( r \) times with varying cluster number from \( K \) to \( \sqrt{n} \). It has been shown that, RPS can capture various cluster structures existing in the real-world datasets \cite{Fred and Jain, 2005}.

In this paper, we aim to solve multi-view clustering in an ensemble clustering way. By following \cite{Fred and Jain, 2005, Liu et al., 2015}, each \( \Pi^{(v)} \) is summarized as a co-association matrix \( S^{(v)} \in \mathbb{R}^{n \times n} \):

\[
S^{(v)}(x_p^{(v)}, x_q^{(v)}) = \frac{1}{r} \sum_{i=1}^{r} \delta(\pi_i^{(v)}(x_p^{(v)}), \pi_i^{(v)}(x_q^{(v)})),
\]

where \( x_p^{(v)}, x_q^{(v)} \in \mathcal{X}^{(v)} \) and \( \delta(a, b) = 1 \) if \( a = b \); 0 otherwise. By this means, we actually compute a pairwise affinity graph for each view upon the partition space, and thus, we can naturally conduct graph partitioning on \( S^{(v)} \) to find a consensus clustering result. In particular, each view could be solved by spectral ensemble clustering \cite{Liu et al., 2015}:

\[
\min_{H} \text{tr}(H^T L_z^{(v)} H) \text{ s.t. } H^T H = I, \tag{2}
\]

where \( H \in \mathbb{R}^{n \times K} \) is the scaled partition matrix that represents the cluster membership of all the data points: \( H_{jk} = 1/\sqrt{|C_k|} \) if \( x_j \in C_k \); 0 otherwise, and \( L_z^{(v)} \) is the normalized Laplacian matrix \cite{Ng et al., 2001} of the co-association matrix in the \( v \)-th view.

However, Eq. (2) only tackles each view individually, yet without utilizing the complementary information between different views. There are still two challenging problems of employing the co-association matrices from multiple views for the clustering aim: (1) How do we seek a consensus co-association matrix to identify the underlying cluster structure shared by multi-view data? (2) How can we capture the disagreements among multiple BPs of intra-view or inter-view?

To address the above challenges, we propose to learn a common representation shared by multi-view co-association matrices via low-rank and sparse matrix decomposition \cite{Ye et al., 2012, Xia et al., 2014}. Moreover, we jointly perform the tasks of spectral graph partitioning and low-rank representation learning. Our Multi-View Ensemble Clustering \( \text{MVEC} \) algorithm is formulated as:

\[
\min_{H, Z, E^{(v)}} \text{tr}(H^T L_z^{(v)} H) + \lambda_1 \|Z\|_1 + \lambda_2 \sum_{v=1}^{m} \|E^{(v)}\|_1 \text{ s.t. } H^T H = I, \forall v, S^{(v)} + H H^T = S^{(v)} Z + E^{(v)}, \tag{3}
\]

where \( H \in \mathbb{R}^{n \times K} \) represents the consensus partition, \( Z \in \mathbb{R}^{n \times n} \) is the low-rank representation, \( E^{(v)} \in \mathbb{R}^{n \times n} \) captures noises for the \( v \)-th view, \( I \in \mathbb{R}^{n \times n} \) is the vector of all ones, and \( \lambda_1, \lambda_2 > 0 \) are two balancing parameters. \( L_z = D_z - Z \) is the Laplacian matrix \cite{Ng et al., 2001} of \( Z \), and \( D_z \) is a diagonal matrix consisting of the sum of each row in \( Z \). By following \cite{Liu et al., 2013a}, we employ the nuclear norm \( \|Z\|_1 \) to measure the matrix rank, and the \( \ell_1 \) norm \( \|E^{(v)}\|_1 \) to characterize the sparseness.

Considering that, the co-association matrices of all the views share with the same cluster structure, and the rank of \( S^{(v)} \) ideally equals to the cluster number \( K (K \ll n) \), we seek for a low-rank representation \( Z \) to reveal the membership between data points through all the views. Meanwhile, to alleviate the “conflict” among different views and handle the outliers existing in co-association matrix, we learn a sparse error matrix \( E^{(v)} \) for each single view. Hence, we may take \( Z \) as a consensus pairwise affinity matrix, and find the consensus partition \( H \) on \( Z \) by spectral clustering.

Taking a close look at Eq. (3), we develop a self-boost constraint to iteratively enhance the original co-association matrix of each view, \( i.e., S^{(v)} + H H^T = S^{(v)} Z + E^{(v)} \), where \( H H^T \) enjoys a clear cluster structure. By using this carefully designed constraint, we first find a high quality consensus partition \( H \) from \( Z \), and then in return, \( H \) is leveraged...
to better guide the learning of $Z$. Besides, we also add the probabilistic simplex constraint ($Z \geq 0$, $Z_1 = 1$) [Duchi et al., 2008] to guarantee the probability property of $Z$.

### 2.2 Optimization

A unified optimization framework of three variable groups is provided by Eq. (3), which is convex with respect to each variable by keeping the others fixed. Thus, we can divide it into several subproblems, and solve them in an iterative way. In details, we apply the Augmented Lagrange Multiplier (ALM) algorithm [Lin et al., 2011] to address our MVEC problem. To facilitate the optimization process, we introduce an auxiliary variable $J \in \mathbb{R}^{n \times n}$ with $Z = J$ to make Eq. (3) separable. Then, our problem could be equivalently converted as:

$$\min_{H, Z, J, E(v)} \text{tr}(H^T L_v H) + \lambda_1 \|J\|_* + \lambda_2 \sum_{v=1}^m \|E(v)\|_1$$

s.t. $\forall v, S(v) + HHT = S(v)Z + E(v), Z = J, Z \geq 0, Z_1 = 1$. (4)

The augmented Lagrangian function of Eq. (4) is written as:

$$\mathcal{L} = \text{tr}(H^T L_v H) + \lambda_1 \|J\|_* + \lambda_2 \sum_{v=1}^m \|E(v)\|_1$$

$$+ \sum_{v=1}^m \Phi(S(v) + HHT - S(v)Z - E(v), Y(v))$$

$$+ \Phi(Z - J, A) + \langle Z_1 - 1, 1 \rangle + \mu \|Z_1 - 1\|_2^2,$$

where $Z \geq 0, \forall v Y(v) \in \mathbb{R}^{n \times n}, A \in \mathbb{R}^{n \times n}$, and $w \in \mathbb{R}^n$ are Lagrange multipliers, $\Phi(A, B) \equiv \|A - B\|_F^2 + \frac{\mu}{2} \|A\|_F^2$, and $\mu > 0$ is a penalty parameter. In the next, we will give the details of iteratively solving $J, Z, E(v)$ and $H$ in sequence.

#### Subproblem of $J$

Solving $\mathcal{L}$ w.r.t $J$ is equivalent to:

$$\min_{J} \lambda_1 \|J\|_* + \frac{1}{2} \|J - (Z(t) + \frac{\Lambda(t)}{\mu(t)})\|_F^2.$$ (5)

Following the previous work [Liu et al., 2013a], Eq. (6) could be effectively solved by a closed-form solution:

$$J(t+1) = S_{\frac{1}{\mu(t)}}(Z(t) + \frac{\Lambda(t)}{\mu(t)}).$$ (7)

where $S(\cdot)$ is the Singular Value Threshold (SVT) operator [Cai et al., 2010].

#### Subproblem of $Z$

Generally, we obtain the solution of $Z(t+1)$ by taking derivate of $\mathcal{L}$ w.r.t. $Z$ and setting it as zero. However, it is not straightforward to solve $\text{tr}(H^T L_v H)$ in a matrix form, since $L_v = D_v - Z$ and $D_v$ is the degree matrix of $Z$. Thus, to simplify this term, we introduce an auxiliary matrix $P \in \mathbb{R}^{n \times n}$ as:

$$P = [P_1 \ldots P_j \ldots P_n], P_j = \begin{bmatrix} \|H_1 - H_j\|_2^2 \\ \vdots \\ \|H_n - H_j\|_2^2 \end{bmatrix},$$ (8)

where $H_j$ is the $j$th row vector in $H$. By utilizing the property of Laplacian matrix and Eq. (8), we have the following simple deduction:

$$\text{tr}(H^T L_v H) = \frac{1}{2} \sum_{i,j} \|H_i - H_j\|^2 Z_{ij} = \frac{1}{2} \text{tr}(P^T Z).$$

Then, we can solve $Z(t+1)$ via the following equivalent formulation as:

$$\min_{Z \geq 0} \frac{1}{2\mu(t)} \text{tr}(P^T(t) Z) + \frac{1}{2} \sum_{v=1}^m \|S(v) + H(t)H(t)^T - S(v)Z - E(v)\|_F^2$$

$$- \|E(v) + Y(v)/\mu(t)\|_F^2 + \frac{1}{2} \|Z - J(t) + \Lambda(t)/\mu(t)\|_F^2$$

$$+ \frac{1}{2} \|Z_1 - 1 + w(t)/\mu(t)\|_F^2.$$ (9)

Inspired by [Zhuang et al., 2016; Lin et al., 2011], we linearize the Eq. (9) at $Z(t)$ as:

$$\min_{Z \geq 0} \langle Z - Z(t), F(t) \rangle + \frac{1}{2\eta(t)} \|Z - Z(t)\|_F^2,$$ (10)

where $\eta(t) = \|P(t)\|_2^2 + \sum_v \|S(v)\|_2^2 + 1 + \|1\|_2^2$, and $F(t)$ equals:

$$\frac{P(t)}{2\mu(t)} + \sum_{v=1}^m S(v)^T Z(t) + E(v) - S(v) - H(t)H(t)^T -$$

$$Y(v)/\mu(t) + \langle Z(t) - J(t+1) + \Lambda(t)/\mu(t) + (Z(t)1 - 1 + w(t)/\mu(t))1^T.$$ (11)

Then, the solution of $Z(t+1)$ is given by:

$$Z(t+1) = \arg \min_{Z \geq 0} \|Z - Z(t) - \frac{1}{\eta(t)} F(t)\|_F^2$$

$$= (Z(t) - \frac{1}{\eta(t)} F(t))_+.$$ (11)

#### Subproblem of $E(v)$

For each view $v$, we update $E(v)$ by:

$$\min_{E(v)} \frac{\lambda_2}{\mu(t)} \|E(v)\|_1 + \frac{1}{2} \|E(v) - (S(v) + H(t)H(t)^T - S(v)Z(t+1) + Y(v)/\mu(t))\|_F^2.$$ (12)

Following [Lin et al., 2011], we have:

$$E(v) = D_{\frac{\lambda_2}{\mu(t)}} \left( S(v) + H(t)H(t)^T - S(v)Z(t+1) + Y(v)/\mu(t) \right).$$ (13)
Subproblem of $H$. Clearly, there are two parts w.r.t $H$ in Eq. (5), which are corresponding to spectral ensemble clustering and our self-boost constraint, respectively. Recall that, we devise this constraint to involve an interaction between learning $Z$ and finding $H$. As shown by Eq. (11), the term of $HH^T$ can effectively hold the cluster structure of $Z_{t+1}$. However, when computing $H$ from $Z$, considering this term will take some unnecessary “noises” induced by $E^{(e)}$ and $Y^{(e)}$ into the clustering process. Hence, we omit the part of $L$ that contains $HH^T$, and recast the subproblem of $H$ as:

$$H_{t+1} = \arg \min_{H} \text{tr}(H^T L_t H),$$

where $L_t$ is updated by $Z_{t+1}$. As suggested by [Zha et al., 2002; Dhillon et al., 2004], Eq. (14) is solved by directly setting $H_{t+1}$ as the $K$ smallest eigenvectors of $L_t$.

Multipliers. Totally, we have $m + 2$ multipliers, which can be updated by:

$$\Delta^{(e)}_{t+1} = \mathcal{S}^{(e)} + H_{t+1}H^T_{t+1} - S^{(e)}Z_{t+1} - E^{(e)}_{t+1},$$

$$Y^{(e)}_{t+1} = Y^{(e)} + \mu_{(e)} \Delta^{(e)}, \forall v = 1, \ldots, m,$$

$$\Lambda_{t+1} = \Lambda_{t} + \mu_{(e)}(Z_{t+1} - J_{t+1}),$$

$$w_{t+1} = w_{t} + \mu_{(e)}(Z(t+1) - 1),$$

where $\Delta^{(e)}$ is introduced for the conciseness. The entire solution for MVEC is summarized in Algorithm 1.

2.3 Discussion

Convergence Analysis. As shown by Algorithm 1, the proposed MVEC can be divided into four subproblems, each of which is convex w.r.t one variable and solved with a closed-form solution. Thus, by finding the optimal solution for each subproblem alternatively, our algorithm at least converges to a local minimum solution.

Complexity Analysis. The major computation of Algorithm 1 lies at step 3-5 and step 7. In details, step 3 costs $O(n^3)$ due to the SVD operation. Step 4 involves several matrix multiplications, leading to a complexity of $O((ln)^3)$, where $l$ is the number of multiplications. For each view, Eq. (13) takes $O(n^2)$, thus the complexity of step 5 is $O(mn^3)$. The eigenvalue decomposition in step 7 costs $O(n^3)$. Hence, the total cost of Algorithm 1 is $O(T(mn^2 + (l + 2)n^3))$, where $T$ is the iteration number. To make our algorithm scalable for large-scale datasets, several off-the-shelf acceleration methods could be used, such as divide-and-conquer [Talwalkar et al., 2013] and the skinny SVD based ones [Zhang et al., 2014; Xiao et al., 2015].

3 Experiment

3.1 Experimental Setting

Datasets. Six real-world datasets are used in the experiment, which cover three text-type ones, i.e., the 3-Sources\(^3\) dataset, the 4-Areas\(^2\) dataset, and the BBCSport dataset provided by [Xia et al., 2014]; and three image-type ones, i.e.,

\(^2\)http://web.cs.ucla.edu/~yzsun/data/four_area.zip

Table 1: Dataset Details

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Instance</th>
<th>#View</th>
<th>#Class</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-Sources</td>
<td>169</td>
<td>3</td>
<td>6</td>
<td>text</td>
</tr>
<tr>
<td>4-Areas</td>
<td>4236</td>
<td>2</td>
<td>4</td>
<td>text</td>
</tr>
<tr>
<td>BBCSport</td>
<td>544</td>
<td>2</td>
<td>5</td>
<td>text</td>
</tr>
<tr>
<td>Caltech101-20</td>
<td>2386</td>
<td>6</td>
<td>20</td>
<td>image</td>
</tr>
<tr>
<td>Digit</td>
<td>2000</td>
<td>2</td>
<td>10</td>
<td>image</td>
</tr>
<tr>
<td>Notting-Hill</td>
<td>550</td>
<td>3</td>
<td>5</td>
<td>image</td>
</tr>
</tbody>
</table>

3-Source\(^3\) (RMVSC) [Xia et al., 2013b] and Robust Multi-View Spectral Clustering (RSEC) [Tao et al., 2016]. SEC\(_{BSV}\) (RSEC\(_{BSV}\)) represents the best ensemble clustering result of each single view, whilst SEC\(_{SUM}\) (RSEC\(_{SUM}\)) denotes the result with the averaged co-association matrices of all the views. Note that, since KCC directly finds consensus clustering among multiple BPs, we only report the KCC\(_{BSV}\).

Clustering Tools. Following [Wu et al., 2015; Liu et al., 2016; Tao et al., 2017], we generate a set of $r = 100$ basic partitions for each individual view by using the Random Parameter Selection (RPS) strategy, which performs K-means algorithm with cosine distance and various cluster numbers. We feed these basic partitions sets as the default input to all the EC methods. For traditional clustering methods, we directly take the raw data as input and follow their preprocessing steps. Two widely-used clustering validation criteria are used to evaluate the clustering performance of all the methods, which are Normalized Mutual Information (NMI) and Normalized Rand Index (Rn) [Wu et al., 2009]. Both these two metrics are positive measures and ranged from 0 to 1, where NMI will drop to zero for the random partition and $Rn$ might be negative to the extremely poor clustering result.
Table 2: Clustering performance on six real-world datasets by NMI (%)  

<table>
<thead>
<tr>
<th>Datasets</th>
<th>3-Sources</th>
<th>4-Areas</th>
<th>BBCSport</th>
<th>Caltech101-20</th>
<th>Digit</th>
<th>Notting-Hill</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpectralBSV</td>
<td>47.14±2.45</td>
<td>43.27±8.53</td>
<td>71.76±0.32</td>
<td>59.65±1.35</td>
<td>63.92±2.40</td>
<td>71.58±2.90</td>
<td>4.49</td>
</tr>
<tr>
<td>SpectralCON</td>
<td>51.66±2.22</td>
<td>54.77±1.52</td>
<td>52.65±3.84</td>
<td>85.96±3.47</td>
<td>76.60±3.07</td>
<td>77.59±4.68</td>
<td>4.38</td>
</tr>
<tr>
<td>SpectralSUM</td>
<td>46.40±4.09</td>
<td>37.06±10.45</td>
<td>60.56±2.48</td>
<td>52.65±1.58</td>
<td>76.60±3.07</td>
<td>77.59±4.68</td>
<td>4.38</td>
</tr>
<tr>
<td>CRSC</td>
<td>51.35±2.93</td>
<td>32.97±2.48</td>
<td>71.71±2.48</td>
<td>56.29±1.36</td>
<td>72.23±1.05</td>
<td>75.20±4.74</td>
<td>4.38</td>
</tr>
<tr>
<td>MultiNMF</td>
<td>39.69±3.95</td>
<td>9.55±1.92</td>
<td>22.60±3.43</td>
<td>59.23±1.43</td>
<td>72.81±2.55</td>
<td>78.53±2.90</td>
<td>3.60</td>
</tr>
<tr>
<td>RMVSC</td>
<td>37.66±4.20</td>
<td>50.81±0.04</td>
<td>80.26±2.91</td>
<td>54.52±1.47</td>
<td>76.71±1.95</td>
<td>67.17±4.93</td>
<td>4.57</td>
</tr>
<tr>
<td>KCCBSV</td>
<td>60.42±2.81</td>
<td>49.03±6.18</td>
<td>84.25±2.33</td>
<td>60.67±1.10</td>
<td>74.59±2.78</td>
<td>77.08±7.10</td>
<td>5.08</td>
</tr>
<tr>
<td>SECBSV</td>
<td>55.23±1.87</td>
<td>55.22±5.09</td>
<td>87.26±2.08</td>
<td>61.91±1.07</td>
<td>76.73±1.09</td>
<td>73.80±1.42</td>
<td>5.13</td>
</tr>
<tr>
<td>SECBSUM</td>
<td>59.21±3.50</td>
<td>74.06±0.00</td>
<td>86.86±4.47</td>
<td>59.56±1.28</td>
<td>66.90±2.89</td>
<td>69.73±0.04</td>
<td>5.22</td>
</tr>
<tr>
<td>RSECBSV</td>
<td>63.09±0.00</td>
<td>43.09±4.42</td>
<td>64.95±0.00</td>
<td>62.13±1.27</td>
<td>88.20±1.68</td>
<td>78.61±0.00</td>
<td>5.03</td>
</tr>
<tr>
<td>RSECBSUM</td>
<td>41.65±0.00</td>
<td>46.67±0.00</td>
<td>62.24±0.00</td>
<td>52.92±1.24</td>
<td>85.72±1.68</td>
<td>80.50±0.20</td>
<td>4.69</td>
</tr>
<tr>
<td>MVEC</td>
<td>75.69±1.00</td>
<td>77.66±1.00</td>
<td>90.83±1.00</td>
<td>62.87±0.94</td>
<td>84.17±0.91</td>
<td>5.98</td>
<td></td>
</tr>
</tbody>
</table>

The top NMI value is highlighted by red bold font and the second best by blue italic. * indicates the difference between MVEC and runner-up is statistically significant on this dataset.

For all the compared methods, we use the true cluster number for fair comparison, and run the codes provided by authors with the parameters as suggested in their work. We test each method 20 times, and report the average result as well as the standard deviation (std). To validate the statistic significance of comparison results, the $p$-value is also calculated with $t$-test. In the experiment, we set $\lambda_1 = 1$ and $\lambda_2 = 0.01$ as the default setting for our MVEC method.

3.2 Clustering Performance

Table 2 and Table 3 summarize the clustering results of the proposed MVEC and other methods in terms of NMI and $R_n$, respectively. As can be seen, our approach generally performs best on all the datasets by both metrics. In details, we improve around 12% NMI (8% $R_n$) and 4% NMI (6% $R_n$) over the runner-up on 3-Sources and 4-Areas, which clearly shows the effectiveness of our approach. To further evaluate the performance, we compute a measurement score by following [Wu et al., 2015]: $\text{score}(A_i) = \sum_j \frac{f(A_i, D_j)}{\max_j f(A_i, D_j)}$, where $f(A_i, D_j)$ indicates the NMI or $R_n$ value of $A_i$ method on the $D_j$ dataset. This score gives a overall evaluation on all the datasets, which shows our approach outperforms the other methods substantially. Besides, according to $p$-value, our model outperforms the second best method with a statistically significant level in most cases.

**Ensemble Clustering vs Multi-View Clustering.** Based on the input data, we may divide all the methods in Table 2 and Table 3 as two categories, such as ensemble clustering (EC) methods (e.g., KCCBSV, SECBSV and KCCSUM) that employ basic partitions, and multi-view clustering methods (e.g., SpectralSUM, CRSC and RMVC) that directly use the multi-view data. As can be seen, EC methods generally outperform the multi-view clustering ones, and even the EC method of single-view performs much better than the traditional ones. This demonstrates the significant superiority and great potentiality of using ensemble clustering methods to solve the multi-view clustering problem.

**Single-View vs Multi-View.** As shown by Table 2 and 3, traditional multi-view clustering methods perform slightly better than the single-view ones. For example, RMVSC only improves 0.08 (0.04) overall score over SpectralBSV by NMI ($R_n$). The similar observation appears at the ensemble clustering case, where the EC methods integrating BPs from all the views generally outperform the BSV ones with a little improvement. Specifically, SECBSUM is only 0.09 (0.21) higher than SECBSV in Table 2 (3). These two observations indicate that: (i) Traditional multi-view clustering methods sometimes cannot improve the performance compared with the single-view ones, since they may suffer from the conflict among different views; (ii) Existing EC methods neglect to fully exploit the multi-view information. In contrast, our MVEC method not only improves the robustness of co-association matrix to the multi-view BPs, but also explicitly considers the connection between different views. Thus, we achieve better perfor-
In this paper, we presented a novel Multi-View Ensemble Clustering (MVEC) framework, which employed co-association matrix to characterize the pairwise affinity of each sing-view data upon multiple basic partitions. A unified optimization framework with the self-boost constraint was provided to jointly learn a consensus affinity matrix shared by multiple views and find the final clustering result. Experiments on six real-world datasets showed the effectiveness of the proposed method compared with both ensemble and multi-view clustering methods.

Acknowledgments

This work is supported in part by the NSF IIS award 1651902, ONR Young Investigator Award N00014-14-1-0484 and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.
References


[Li et al., 2015a] Sheng Li, Ming Shao, and Yun Fu. Multi-view low-rank analysis for outlier detection. In *SDM*, 2015.


[Liu et al., 2013b] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix factorization. In *SDM*, 2013.


