Robust Multi-View Spectral Clustering via Low-Rank and Sparse Decomposition

Rongkai Xia, Yan Pan, Lei Du, and Jian Yin
Sun Yat-sen University, Guangzhou, China

Abstract

Multi-view clustering, which seeks a partition of the data in multiple views that often provide complementary information to each other, has received considerable attention in recent years. In real life clustering problems, the data in each view may have considerable noise. However, existing clustering methods blindly combine the information from multi-view data with possibly considerable noise, which often degrades their performance. In this paper, we propose a novel Markov chain method for *Robust Multi-view Spectral Clustering* (RMSC). Our method has a flavor of low-rank and sparse decomposition, where we firstly construct a transition probability matrix from each single view, and then use these matrices to recover a shared low-rank transition probability matrix as a crucial input to the standard Markov chain method for clustering. The optimization problem of RMSC has a low-rank constraint on the transition probability matrix, and simultaneously a probabilistic simplex constraint on each of its rows. To solve this challenging optimization problem, we propose an optimization procedure based on the Augmented Lagrangian Multiplier scheme. Experimental results on various real world datasets show that the proposed method has superior performance over several state-of-the-art methods for multi-view clustering.

Introduction

In many real-life clustering problems, one has access to multiple representations or views of the data. These views often provide complementary information to each other. Multi-view clustering, which seeks to improve the clustering performance by leveraging the information from multiple views, has recently received considerable attention. Many multi-view clustering methods have been proposed (Chaudhuri et al. 2009; Bickel and Scheffer 2004; Kumar and Daumé 2011; Kumar, Rai, and Daumé 2011; Zhou and Burges 2007; Greene and Cunningham 2009).

Among diverse clustering methods, we consider multi-view spectral clustering via Markov chains. Spectral clustering has become one of the most popular clustering methods because it has well-defined mathematical principles, and often has superior performance than traditional clustering methods such as k-means clustering. Spectral clustering has a natural connection to Markov chains (Shi and Malik 2000). For example, in the single view case, spectral clustering is derived from a real relaxation of the combinatorial normalized cut, which leads to the graph Laplacian that can be naturally converted to a transition probability matrix to generate a Markov chain on the graph.

In Markov chain methods, a crucial step is to construct an accurate transition probability matrix. In the context of multi-view clustering, one needs to construct a transition probability matrix by leveraging the information from multiple representations. For example, Zhou *et al.* (Zhou and Burges 2007) proposed a Markov chain method by generalizing the normalized cut from a single view to multiple views, which constructs the transition matrix via a Markov mixture that combines multiple transition matrices defined on different type of representations. [0]

In real world applications of multi-view spectral clustering, the input data may be noisy, which results in the corresponding similarity/transition matrices being corrupted by considerable noise. However, the existing methods for multi-view spectral clustering blindly combine multiple representations of data with possibly considerable noise, which may often degrade the clustering performance.

To address this issue, in this paper, we propose ***R**obust **M**ulti-view **S**pectral **C**lustering* (**RMSC**), a Markov chain method that explicitly handles the possible noise in the transition probability matrices associated with different views. As shown in Figure 1, we firstly use each type of representation to construct a similarity matrix and a transition probability matrix. Then, we propose to learn the final transition probability matrix via low-rank and sparse decomposition. Different from the existing methods for low-rank and sparse decomposition (i.e., (Ye et al. 2012; Pan et al. 2013b)), our formulation introduces a probabilistic simplex constraint on each row of the learned transition probability matrix, resulting in a considerably more challenging optimization problem. We propose an optimization procedure to solve the optimization problem based on the Augmented Lagrangian Multiplier (ALM) scheme (Lin, Chen, and Ma 2010). Experimental results on benchmark datasets of multi-view clustering show that the proposed RMSC outperforms

[0] Corresponding author: Yan Pan (panyan5@mail.sysu.edu.cn)

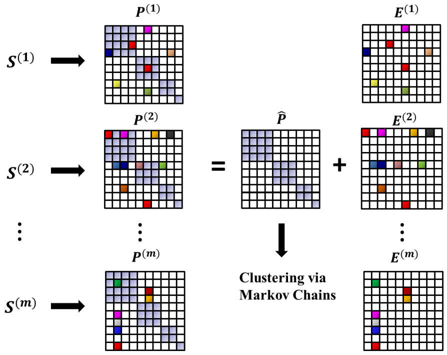

Figure 1: Overview of transition matrix construction. Given n data points with m views, we firstly construct a similarity matrix $S^{(i)}$ $(i = 1, 2, ..., m)$ for each view, and then calculate the corresponding transition probability matrix $P^{(i)}$ by $P^{(i)} = (D^{(i)})^{-1}S^{(i)}$. After that, we recover a low-rank latent transition probability matrix $\hat{P}$ from $P^{(1)}, P^{(2)}, \dots, P^{(m)}$ via low-rank and sparse decomposition, where $\hat{P}$ will be used as input to the standard Markov chain method for (single view) spectral clustering.

several state-of-the-art clustering methods.

Related Work

Clustering is a classical problem in data mining. Recently, with the increasing quantities of data in multiple representations from diverse sources, multi-view clustering algorithms have attracted considerable attention (Chaudhuri et al. 2009; Bickel and Scheffer 2004; Kumar and Daumé 2011; Kumar, Rai, and Daumé 2011; Zhou and Burges 2007; Greene and Cunningham 2009).

Existing methods for multi-view clustering can be roughly categorized into three streams. The methods in the first stream integrate multi-view features into some common representation before (or in) the clustering process (Bickel and Scheffer 2004; Kumar, Rai, and Daumé 2011; Kumar and Daumé 2011; Zhou and Burges 2007). For example, the methods in (Bickel and Scheffer 2004; Kumar, Rai, and Daumé 2011) incorporate multi-view features to construct the loss functions for clustering; the method in (Zhou and Burges 2007) constructs a Markov mixture model for clustering via leveraging multiple transition probability matrices, each of which is exacted from one view. The methods in the second stream firstly project each view of features onto a common low-dimensional subspace, and then conduct clustering in this subspace. A representative method in this stream is CCA for multi-view clustering (Chaudhuri et al. 2009), which uses CCA to project the multi-view high dimensional data onto a low-dimensional subspace. The methods in the third stream (Greene and Cunningham 2009) firstly learn a clustering solution from each single view, and then combine these intermediate outputs to get a final clustering solution. The proposed method in this paper belongs to the first stream.

Our method performs multi-view clustering by building a Markov chain. Recently, some effort has been made in Markov chain methods for multi-view clustering. Zhou *et al.* (Zhou and Burges 2007) proposed a Markov chain method for the generalized normalized cut on multi-views data, which firstly constructs a transition probability matrix on each view, and then combines these matrices via a Markov mixture. However, the real life multi-view data may be noisy, which results in considerable corruptions in the corresponding transition matrix associated with each view. The method in (Zhou and Burges 2007) blindly combines multiple transition matrices with possibly considerable noise, which often degrades its clustering performance.

An integral part of our method is to build an accurate transition probability matrix by combining multiple input matrices via low-rank and sparse decomposition. The idea of explicitly handling the noise in multiple input matrices via low-rank and sparse decomposition is not new. For example, the robust data fusion methods (Ye et al. 2012; Pan et al. 2013b) separate the considerable noise in multiple input matrices via low-rank and sparse decomposition; Pan *et al.* (Pan et al. 2013a) proposed a rank aggregation method to distinguish the noise via low-rank and structured-sparse decomposition. The proposed method in this paper shares some similar features with the previous methods for low-rank and sparse decomposition. However, since our goal is to learn a low-rank transition probability matrix, our formulation introduces a probabilistic simplex constraint on each row of the learned matrix, resulting in a considerably more challenging optimization problem than those in (Ye et al. 2012; Pan et al. 2013b).

Spectral Clustering via Markov Chains

Given a set of data points $\{x_1, \dots, x_n\}$, we define a similarity matrix S where $S_{ij} \geq 0$ denotes the similarity on a pair of data points x_i and x_j. Let $G = (V, E, S)$ be a weighted graph with vertex set V, edge set E, and the corresponding weight/similarity matrix S, where each vertex v_i associates with the data point x_i and each edge $(i, j) \in E$ associates with S_{ij} between x_i and x_j. One popular way is to use Gaussian kernels to define the similarity matrix, i.e., $S_{ij} = exp(-\frac{||x_i - x_j||_2^2}{\sigma^2})$ where $||.||_2$ denotes the ℓ_2 norm and σ^2 denotes the standard deviation (e.g., one can set σ^2 to be the average Euclidean distance over all pairs of data points). The degree of a vertex v_i is defined as $d_i = \sum_{j=1}^{n} S_{ij}$.

Spectral clustering seeks a partition of data points in a weighted graph G. It has been shown that spectral clustering has a natural connection to transition probabilities or random walks of the Markov chains (Shi and Malik 2000). More specifically, spectral clustering can be interpreted as trying to find a partition on G such that the Markov random walk stays long within the same cluster and seldom jumps between clusters. Let P be the transition matrix of a random walk defined on G. One can define P as $P = D^{-1}S$ where D is a diagonal matrix with $D_{ii} = d_i = \sum_{j=1}^{n} S_{ij}$. It is easy to verify that each row of P is a probability distribution, i.e., for all j, $P_{ij} \geq 0$ and $\sum_{j=1}^{n} P_{ij} = 1$. P_{ij} represents the probability of jumping in one step from v_i to v_j. Given a connected and non-bipartite G with the transi-

$$\begin{pmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{pmatrix}$$

Figure 2: An illustration of a low-rank transition probability matrix. There are 7 data points in 3 clusters, i.e., $\{(x_1, x_2, x_3), (x_4, x_5), (x_6, x_7)\}$, where any pair of points in the same cluster is identical (with the maximum similarity 1) and any pair of points in different clusters is different (with the minimum similarity 0). Then the corresponding transition probability matrix is shown in this figure, whose rank is 3, exactly equaling to the number of clusters. The detailed explanation can be found in in the subsection "Transition Matrix Construction".

tion matrix P, there exists a unique stationary distribution π satisfying $\pi = P\pi$.

Here we briefly outline the algorithm of Markov chains for spectral clustering. We refer the readers to (Zhou, Huang, and Schölkopf 2005) for more details of spectral clustering via Markov chains.

- Given a weighted graph $G = (V, E, S)$, define a random walk over G with a transition probability matrix $P = D^{-1}S \in \mathbb{R}^{n\times n}$ such that it has a stationary distribution π satisfying $\pi = P\pi$.
- Let Π denote the diagonal matrix with its ith diagonal elements being the stationary distribution $\pi(i)$. Construct the matrix $L = \Pi - \frac{\Pi P + P^T \Pi}{2}$.
- Obtain the r smallest generalized eigenvectors $u_1, \dots, u_r$ of the generalized eigenproblem $Lu = \lambda \Pi u$.
- Let $U \in \mathbb{R}^{n\times r}$ be the matrix containing the vectors $u_1, \dots, u_r$. Run k-means clustering to cluster the row vectors of U.
- Assign the data point x_i to cluster c if the ith row of U is assigned to cluster c by the k-means algorithm.

Robust Multi-View Spectral Clustering

A crucial step in the Markov chain methods for spectral clustering is constructing an accurate transition probability matrix. In this section, we present the proposed RMSC method which conducts low-rank and sparse decomposition to recover a latent transition probability matrix from multiple views. This recovered matrix is used as input to the standard Markov chain method (see the previous section) to obtain the final clustering solution.

Transition Matrix Construction

In the context of multi-view clustering, we are given a set of data points in m views, in each of which we can construct a similarity matrix $S^{(i)}$ and the corresponding weighted graph $G^{(i)}$ $(i = 1, 2, \cdots, m)$. Let $P^{(1)}, \dots, P^{(m)}$ be the transition matrix associated to $G^{(1)}, \dots, G^{(m)}$, respectively. Figure 1 illustrates the framework of the proposed method for transition matrix construction. The basic assumptions in the proposed method are twofold: (1) The features in each individual view are sufficient to discover most of the clustering information. (2) The features in each individual view might be corrupted by noise, i.e., these noise might result in a small portion of data points being assigned to wrong clusters. Based on these assumptions, each transition probability matrix $P^{(i)}$ associated to an individual view can be naturally decomposed into two parts: a shared latent transition probability matrix $\hat{P}$ that reflects the underlying true clustering information, combined with a deviation error matrix $E^{(i)}$ that encodes the noise in the transition probabilities in each view.

$$\forall i,\ P^{(i)} = \hat{P} + E^{(i)}$$

Once $\hat{P}$ is given, we can simply use $\hat{P}$ as the input transition matrix to the Markov chain method for spectral clustering to obtain the final clustering solution.

A key question arising here is how to model the latent matrix $\hat{P}$ and the error matrices $E^{(i)}$.

For the latent matrix $\hat{P}$, we consider an ideal case with 7 data points in 3 clusters, i.e., $\{(x_1, x_2, x_3), (x_4, x_5), (x_6, x_7)\}$, where any pair of points in the same cluster is identical (with the maximum similarity 1) and any pair of points in different clusters is different (with the minimum similarity 0). Then the resulting transition probability matrix is shown in Figure 2. Since the first three columns are identical, the 4th and 5th columns are identical, and the 6th and 7th columns are identical, it is obvious that the rank of this transition matrix is 3, which equals to the number of clusters. Note that exchanging two columns/rows in a matrix A does not change the rank of A. Hence, in the general case with a set of data points generated by k clusters, we can re-organize the columns/rows of the resulting transition matrix and convert it into a block-diagonal form with k blocks (like the one in Figure 2), and the rank of this block-diagonal matrix is not larger than k. In real world clustering problems, it is still reasonable to assume that the transition probability between any two points within the same cluster is high, and that between any two points in different clusters is low, which results in a matrix that tends to be of low-rank. In summary, these observations motivate us to assume that the transition probability matrix which reflects the underlying true clustering information tends to be of low-rank.

Each error matrix $E^{(i)}$ represents the difference between $P^{(i)}$ and $\hat{P}$. Since we assume the features in each individual view are sufficient to identify most of the clustering structure, it is reasonable to assume that there are only a small fraction of elements in $P^{(i)}$ being significantly different from the corresponding ones in $\hat{P}$. That is, the deviation error matrix $E^{(i)}$ tends to be sparse.

Problem Formulation

Under the low-rank and sparse assumptions, we formulate the transition matrix construction problem as:

$$\min_{\hat{P}, E^{(i)}} rank(\hat{P}) + \lambda \sum_{i=1}^{m} \|E^{(i)}\|_0 \tag{1}$$
$$s.t.\ \ i = 1, 2, ..., m,\ P^{(i)} = \hat{P} + E^{(i)},\ \hat{P} \geq 0, \hat{P}\mathbf{1} = \mathbf{1},$$

where $rank(\hat{P})$ is the rank of $\hat{P}$, the ℓ_0 norm $||E^{(i)}||_0$ represents the number of non-zero elements in $E^{(i)}$, $\mathbf{1}$ denotes the vector with all ones, and λ is a non-negative trade-off parameter. Note that the constraints $\hat{P} \geq 0, \hat{P}\mathbf{1} = \mathbf{1}$ enforce $\hat{P}$ to be a transition probability matrix, i.e., each of its rows is a probability distribution.

It is known that the optimization problem in (1) is NP-hard in general due to the non-convex $rank(\hat{P})$ and $||E^{(i)}||_0$. A popular way is to replace $rank(\hat{P})$ with the trace norm $||\hat{P}||_*$, and $||E^{(i)}||_0$ with the ℓ_1 norm $||E^{(i)}||_1$, resulting in the following convex optimization problem:

$$\begin{gathered}\min_{\hat{P},E^{(i)}} \|\hat{P}\|_* + \lambda \sum_{i=1}^{m} \|E^{(i)}\|_1 \\ s.t.\ \ i = 1,2,...,m,\ P^{(i)} = \hat{P} + E^{(i)},\ \hat{P} \geq 0, \hat{P}\mathbf{1} = \mathbf{1}.\end{gathered} \tag{2}$$

The trace norm $\|\hat{P}\|_*$ is the convex envelope of the rank of $\hat{P}$ over the unit ball of the spectral norm, and minimizing the trace norm often induces the desirable low-rank structure in practice (Fazel, Hindi, and Boyd 2001; Srebro, Rennie, and Jaakkola 2004). The ℓ_1 norm $\|E^{(i)}\|_1 = \sum_{(i,j)} |E_{ij}|$ is well-known to be a convex surrogate of $\|E\|_0$.

Optimization

The optimization problem (2) is still challenging because the matrix $\hat{P}$ has a trace-norm constraint, and simultaneously each of its rows has a probabilistic simplex constraint. In this section, we propose an optimization procedure to solve this problem via the *Augmented Lagrangian Multiplier* (**ALM**) scheme (Lin, Chen, and Ma 2010), which has shown its good balance between efficiency and accuracy in many matrix learning problems.

By introducing an auxiliary variable Q, we convert (2) into the following equivalent form:

$$\begin{gathered}\min_{\hat{P},Q,E^{(i)}} \|Q\|_* + \lambda \sum_{i=1}^{m} \|E^{(i)}\|_1 \\ s.t.\ \ i = 1,2,...,m,\ P^{(i)} = \hat{P} + E^{(i)}, \\ \hat{P} \geq 0, \hat{P}\mathbf{1} = \mathbf{1}, \hat{P} = Q.\end{gathered} \tag{3}$$

The corresponding augmented Lagrange function of (3) is:

$$\begin{aligned}&\mathcal{L}(\hat{P},Q,E^{(i)}) = \|Q\|_* + \lambda \sum_{i=1}^{m} \|E^{(i)}\|_1 \\ &+ \sum_{i=1}^{m} \langle Y^{(i)}, \hat{P} + E^{(i)} - P^{(i)} \rangle + \frac{\mu}{2} \sum_{i=1}^{m} \|\hat{P} + E^{(i)} - P^{(i)}\|_F^2 \\ &+ \langle Z, \hat{P} - Q \rangle + \frac{\mu}{2} \|\hat{P} - Q\|_F^2 \quad s.t.\ \hat{P} \geq 0, \hat{P}\mathbf{1} = \mathbf{1},\end{aligned} \tag{4}$$

where Z, $Y^{(i)}$ represent the Lagrange multipliers, $\langle \cdot, \cdot \rangle$ denotes the inner product of matrices (i.e.,for two matrices A and B, $\langle A, B \rangle = A^T B$), and $\mu > 0$ is an adaptive penalty parameter.

The sketch of the proposed algorithm for transition matrix construction is shown in Algorithm 1. Next we will present the update rules for each of $\hat{P}$, Q and $E^{(i)}$, by minimizing $\mathcal{L}$ in (4) with other variables being fixed. Please refer to Algorithm 1 for the details.

Algorithm 1 Algorithm for transition matrix construction

```
Input: $\lambda, P^{(i)} \in \mathbb{R}^{n \times n} (i = 1, 2, \ldots, m)$
Initialize: $\hat{P} = \mathbf{0}, Q = \mathbf{0}, Z = \mathbf{0}, Y^{(i)} = \mathbf{0}, E^{(i)} = \mathbf{0}, \mu = 10^{-6}, \rho = 1.9, max_\mu = 10^{10}, \epsilon = 10^{-8}$
Repeat
  1. Let $C \leftarrow \frac{1}{m+1}(Q - \frac{Z}{\mu} + \sum_{i=1}^{m}(P^{(i)} - E^{(i)} - \frac{Y^{(i)}}{\mu}))$.
  2. For j=1,2, . . . , n
       Run Algorithm 2 using $C_j$ as input to update $\hat{P}_j$
       where $C_j$/$\hat{P}_j$ is the $j$th row of $C$/$\hat{P}$, respectively.
  3. For i=1,2, . . . , m
       Update $E^{(i)}$ via Eq.(7).
  4. Update $Q$ via Eq.(6).
  5. Set $Z \leftarrow Z + \mu(\hat{P} - Q)$.
  6. For i=1,2, . . . , m
       Set $Y^{(i)} \leftarrow Y^{(i)} + \mu(\hat{P} + E^{(i)} - P^{(i)})$.
  7. Set $\mu \leftarrow \min(\rho\mu, max_\mu)$.
Until $\min(||\hat{P} + E^{(i)} - P^{(i)}||_\infty, ||\hat{P} - Q||_\infty) \leq \epsilon$
Output: $\hat{P}$, $E^{(i)}$ $(i = 1, 2, ...m)$
```

Solving Q

When other variables are fixed, the subproblem w.r.t. Q is

$$\min_{Q} \|Q\|_* + \frac{\mu}{2}\|\hat{P} - Q + \frac{Z}{\mu}\|_F^2, \tag{5}$$

which can be solved by the Singular Value Threshold method (Cai, Candès, and Shen 2010). More specifically, let $U\Sigma V^T$ be the SVD form of $(\hat{P} + \frac{Z}{\mu})$, the solution to (5) is as follows:

$$Q = U\mathcal{S}_{1/\mu}(\Sigma)V^T, \tag{6}$$

where $\mathcal{S}_\delta(\mathbf{X}) = \max(\mathbf{X} - \delta, 0) + \min(\mathbf{X} + \delta, 0)$ is the shrinkage operator (Lin, Chen, and Ma 2010).

Solving $\mathbf{E}^{(i)}$

The subproblem w.r.t. $E^{(i)}$ $(i = 1, 2, ..., m)$ can be simplified as:

$$\min_{E^{(i)}} \lambda\|E^{(i)}\|_1 + \frac{\mu}{2}\|E^{(i)} - (P^{(i)} - \hat{P} - \frac{Y^{(i)}}{\mu})\|_F^2, \tag{7}$$

which has a closed form solution $E^{(i)} = \mathcal{S}_{\lambda/\mu}(P^{(i)} - \hat{P} - \frac{Y^{(i)}}{\mu})$.

Solving $\hat{\mathbf{P}}$

With other variables being fixed, we update $\hat{P}$ by solving

$$\begin{aligned}\hat{P} = \operatorname*{argmin}_{\hat{P}} \frac{\mu}{2} \sum_{i=1}^{m} \|\hat{P} + E^{(i)} - P^{(i)} + \frac{Y^{(i)}}{\mu}\|_F^2 \\ + \frac{\mu}{2}\|\hat{P} - Q + \frac{Z}{\mu}\|_F^2 \quad s.t.\ \hat{P} \geq 0, \hat{P}\mathbf{1} = \mathbf{1}.\end{aligned} \tag{8}$$

For ease of presentation, we define

$$C = \frac{1}{m+1}(Q - \frac{Z}{\mu} + \sum_{i=1}^{m}(P^{(i)} - E^{(i)} - \frac{Y^{(i)}}{\mu})).$$

Then with simple algebra, the problem in (8) can be rewritten as:

$$\begin{aligned}\hat{P} &= \operatorname*{argmin}_{\hat{P}} \frac{1}{2}\|\hat{P} - C\|_F^2, \quad s.t.\ \hat{P} \geq 0, \hat{P}\mathbf{1} = \mathbf{1} \\ &= \operatorname*{argmin}_{\hat{P}_1,\ldots,\hat{P}_n} \frac{1}{2}\sum_{i=1}^{n} \|\hat{P}_i - C_i\|_F^2, \quad s.t.\ \sum_{j=1}^{n} \hat{P}_{ij} = 1, \hat{P}_{ij} \geq 0,\end{aligned} \tag{9}$$

where $\hat{P}_i$/C_i denotes the ith row of the matrix $\hat{P}$/C, respectively. That is, the problem in (9) can be decomposed into n independent subproblems:

$$\min_{\hat{P}_i} \frac{1}{2}\|\hat{P}_i - C_i\|_2^2 \;\; s.t. \;\; \sum_{j=1}^{n} \hat{P}_{ij} = 1, \hat{P}_{ij} \geq 0.$$

Each subproblem is a proximal operator problem with probabilistic simplex constraint, which can be efficiently solved by the projection algorithm in (Duchi et al. 2008). Here we include the algorithm in Algorithm 2 for self-containedness.

Since the objective (2) is convex subject to linear constraints, and all of its subproblems can be solved exactly, based on existing theoretical results (Luo 2012), we have that Algorithm 1 converges to global optima with a linear convergence rate.

Algorithm 2 Algorithm for proximal operator with simplex constraint

```
Input: A vector $C_i \in \mathbb{R}^n$
Sort $C_i$ into $u$: $u_1 \geq u_2 \geq \cdots \geq u_n$
Find $\hat{j} = \max\{j : 1 - \sum_{r=1}^{j}(u_r - u_j) \geq 0\}$
Let $\sigma = \frac{1}{\hat{j}}(\sum_{i=1}^{\hat{j}} u_i - 1)$
Output: $\hat{P}_i$ where $\hat{P}_{ij} = \max(C_{ij} - \sigma, 0), j = 1, 2, \cdots, n$
```

Experiments

In this section, we evaluate and compare the performance of the proposed RMSC method on various real world datasets. We chose the following five multi-view clustering algorithms as baselines: (1)**Best Single View**: Using the individual view which achieves the best spectral clustering performance with a single view of data. (2)**Feature Concatenation**: Concatenating the features of each view, and then performing spectral clustering directly on this concatenated feature representation. (3)**Kernel Addition**: Constructing a kernel matrix from each view, and then averaging these matrices to obtain a single kernel matrix for spectral clustering. (4)**Mixture of Markov Chains (MMC)**: The mixture of Markov chains method proposed in (Zhou and Burges 2007), which is perhaps the most related one to the proposed RMSC. (5)**Co-regularized Spectral clustering (Co-Reg)**: The co-regularization method for spectral clustering (Kumar, Rai, and Daumé 2011). Following the settings in (Kumar, Rai, and Daumé 2011), we use the Gaussian kernel for each view, and the best clustering results are reported with the parameter λ being chosen from 0.01 to 0.05.

dataset	**instances**	**views**	**clusters**
BBC	2225	3	5
BBCSport	737	2	5
WebKB	1051	2	2
UCI Digits	2000	3	10
Flower17	1360	7	17
CCV	9317	3	20

Table 1: Statistics of the real world datasets

Results on Real World Datasets

We report the experimental results on six real-world datasets: BBC and BBCSport[1] for news article clustering, WebKB (Sindhwani, Niyogi, and Belkin 2005) for webpages clustering, UCI digits (Asuncion and Newman 2007) and Flower17[2] for image clustering, and Columbia Consumer Video (CCV) (Jiang et al. 2011) for video event clustering. The statistics of these datasets are summarized in Table 1.

In all the experiments, we use six metrics to measure the clustering performances: precision, recall, F-score, normalized mutual information (NMI), average entropy, and adjusted rand index(Adj-RI) (Manning, Raghavan, and Schütze 2008; Hubert and Arabie 1985). Note that lower values indicate better performance for average entropy, and higher values indicate better performance for the other metrics.

In all the experiments, Gaussian kernels are used to build the similarity matrix for each single view. The standard deviation is set to the median of the pairwise Euclidean distances between every pair of data points for all of the datasets except BBC and BBCSport. For the BBC and BBCSport datasets, we follow (Kumar and Daumé 2011) to set the standard deviation to be 100. For MMC (Zhou and Burges 2007) and the proposed RMSC, the transition probability matrix for each view is constructed by $P = D^{-1}S$, where S is the similarity matrix and D is a diagonal matrix with D_{ii} being the sum of the elements of the ith row in S. In RMSC, the regularization parameter λ is set to be 0.005 [3]. Note that we use the same value of λ in all the views. One can use

	BBCSport			WebKB		
	NMI	Adj-RI	Fscore	NMI	Adj-RI	Fscore
$\lambda = 0.005$	**0.811**	**0.790**	**0.839**	**0.779**	**0.885**	**0.960**
$\lambda = 0.01$	**0.825**	**0.827**	**0.868**	**0.764**	**0.876**	**0.958**
$\lambda = 0.05$	**0.792**	**0.773**	**0.827**	**0.752**	**0.868**	**0.955**
$\lambda = 0.1$	**0.816**	**0.816**	**0.860**	**0.765**	**0.876**	**0.958**
$\lambda = 0.5$	0.785	0.773	0.826	0.718	0.845	0.947
$\lambda = 1$	0.770	0.774	0.827	0.718	0.845	0.947
$\lambda = 5$	0.753	0.763	0.817	0.718	0.845	0.947
$\lambda = 10$	0.733	0.733	0.794	0.718	0.845	0.947
$\lambda = 50$	0.611	0.570	0.672	0.718	0.845	0.947
$\lambda = 100$	0.611	0.554	0.660	0.718	0.845	0.947
the second best baseline	0.718	0.697	0.768	0.718	0.845	0.947

Table 2: Results of RMSC with different values of λ on the BBCSport and WebKB datasets.

[1] http://mlg.ucd.ie/datasets

[2] http://www.robots.ox.ac.uk/ vgg/data/flowers/. We directly use the seven pre-computed kernel matrices included in the dataset as input for clustering. Hence we do not report the results of Feature Concatenation.

[3] Here we set $\lambda = 0.005$ because it works well in all of the datasets. In Section "Parameter Sensitivity", we investigate the effects on the performance of RMSC with different values of the parameter λ, which shows that RMSC has superior performance gains over the baselines as long as λ varying in a suitable range.

dataset	method	F-score	Precision	Recall	Entropy	NMI	Adj-RI
BBC	Best Single View	0.798(0.058)	0.804(0.071)	0.792(0.045)	0.600(0.084)	0.735(0.033)	0.744(0.075)
	Feature Concat	0.828(0.074)	0.822(0.101)	0.838(0.047)	0.498(0.131)	0.784(0.048)	0.780(0.098)
	Kernel Addition	0.828(0.077)	0.818(0.110)	0.842(0.045)	0.503(0.146)	0.783(0.052)	0.779(0.103)
	Co-Reg	0.829(0.049)	0.836(0.057)	0.822(0.042)	0.516(0.076)	0.771(0.031)	0.783(0.063)
	MMC	0.829(0.075)	0.825(0.097)	0.834(0.051)	0.498(0.133)	0.783(0.051)	0.781(0.098)
	Ours	**0.871(0.053)**	**0.879(0.078)**	**0.864(0.046)**	**0.431(0.109)**	**0.808(0.059)**	**0.837(0.087)**
BBCSport	Best Single View	0.768(0.004)	0.781(0.015)	0.756(0.023)	0.616(0.028)	0.715(0.006)	0.697(0.003)
	Feature Concat	0.657(0.015)	0.667(0.019)	0.649(0.030)	0.862(0.041)	0.604(0.016)	0.552(0.018)
	Kernel Addition	0.657(0.020)	0.649(0.007)	0.667(0.044)	0.886(0.034)	0.600(0.022)	0.548(0.022)
	Co-Reg	0.766(0.002)	0.786(0.008)	0.748(0.012)	0.606(0.015)	0.718(0.003)	0.696(0.001)
	MMC	0.657(0.019)	0.658(0.018)	0.658(0.039)	0.877(0.039)	0.601(0.018)	0.550(0.022)
	Ours	**0.869(0.035)**	**0.871(0.022)**	**0.869(0.049)**	**0.405(0.023)**	**0.818(0.017)**	**0.829(0.044)**
WebKB	Best Single View	0.889(0.003)	0.824(0.005)	0.956(0.000)	0.406(0.001)	0.532(0.002)	0.618(0.001)
	Feature Concat	0.947(0.001)	0.947(0.003)	0.947(0.001)	0.214(0.001)	0.718(0.002)	0.845(0.001)
	Kernel Addition	0.947(0.006)	0.947(0.004)	0.947(0.010)	0.214(0.003)	0.718(0.001)	0.845(0.002)
	Co-Reg	0.933(0.002)	0.958(0.003)	0.910(0.001)	0.209(0.002)	0.700(0.008)	0.814(0.005)
	MMC	0.947(0.005)	0.947(0.002)	0.947(0.001)	0.214(0.001)	0.718(0.003)	0.845(0.002)
	Ours	**0.960(0.001)**	**0.965(0.002)**	**0.965(0.001)**	**0.164(0.001)**	**0.779(0.004)**	**0.885(0.002)**
UCI digits	Best Single View	0.591(0.029)	0.582(0.030)	0.601(0.030)	1.195(0.071)	0.642(0.021)	0.545(0.033)
	Feature Concat	0.452(0.019)	0.438(0.024)	0.468(0.015)	1.489(0.078)	0.556(0.022)	0.389(0.022)
	Kernel Addition	0.754(0.020)	0.740(0.035)	0.769(0.011)	0.718(0.033)	0.787(0.007)	0.726(0.023)
	Co-Reg	0.780(0.052)	0.764(0.067)	0.798(0.035)	0.664(0.113)	0.804(0.031)	0.755(0.058)
	MMC	0.762(0.036)	0.740(0.052)	0.787(0.018)	0.687(0.068)	0.799(0.016)	0.735(0.040)
	Ours	**0.811(0.049)**	**0.797(0.065)**	**0.826(0.031)**	**0.601(0.099)**	**0.822(0.026)**	**0.789(0.055)**
CCV	Best Single View	0.119(0.002)	0.126(0.002)	0.114(0.002)	3.466(0.016)	0.177(0.004)	0.069(0.002)
	Feature Concat	0.096(0.001)	0.073(0.001)	0.141(0.007)	3.739(0.009)	0.119(0.002)	0.022(0.002)
	Kernel Addition	0.124(0.002)	0.126(0.004)	0.123(0.003)	3.496(0.017)	0.171(0.004)	0.072(0.003)
	Co-reg	0.119(0.140)	0.125(0.147)	0.113(0.134)	3.473(3.358)	0.176(0.203)	0.068(0.090)
	MMC	0.125(0.001)	0.128(0.003)	0.123(0.002)	3.499(0.019)	0.170(0.004)	0.073(0.002)
	Ours	**0.132(0.002)**	**0.133(0.003)**	**0.129(0.003)**	**3.225(0.017)**	**0.203(0.006)**	**0.082(0.003)**
Flower17	Best Single View	0.206(0.005)	0.191(0.006)	0.225(0.008)	2.820(0.021)	0.315(0.004)	0.153(0.006)
	Kernel Addition	0.090(0.004)	0.063(0.002)	0.165(0.027)	3.862(0.034)	0.064(0.008)	0.007(0.003)
	Co-reg	0.097(0.003)	0.089(0.002)	0.106(0.006)	3.567(0.015)	0.130(0.003)	0.036(0.002)
	MMC	0.369(0.018)	0.362(0.019)	0.376(0.018)	2.015(0.054)	0.509(0.013)	0.329(0.019)
	Ours	**0.423(0.019)**	**0.414(0.015)**	**0.433(0.011)**	**1.813(0.037)**	**0.559(0.009)**	**0.387(0.014)**

Table 3: Comparison results on six datsets. On each dataset, 20 test runs with different random initializations were conducted and the average performance as well as the standard deviation (numbers in parentheses) are reported.

different values of λ for different views if one has certain importance prior of different views, such as in (Zhou and Burges 2007).

The results are shown in Table 3. As can be seen, in all of the six datasets, the proposed RMSC shows superior performance gains over the baselines w.r.t. all the six metrics. Here are some statistics. On Flower17 with seven views, the results of RMSC indicate a relative increase of $\mathbf{14.6}\%$, $\mathbf{9.8}\%$ and $\mathbf{17.6}\%$ w.r.t. F-score, NMI and Adj-RI, respectively, compared to the corresponding second best baseline. On BBCSport with three views, RMSC shows $\mathbf{13.2}\%$, $\mathbf{13.9}\%$ and $\mathbf{18.9}\%$ of relative improvement w.r.t. F-score, NMI and Adj-RI over the corresponding second best baseline, respectively. On WebKB, RMSC indicates a relative increase of $\mathbf{8.5}\%$ and $\mathbf{4.7}\%$ w.r.t NMI and Adj-RI over the corresponding second best baselines, respectively.

Remark We also evaluate RMSC on syntactic datasets to observe the effects with different types of noise. The results can be found in the supplementary material.

Parameter Sensitivity

There is a trade-off parameter λ in RMSC. In unsupervised clustering, one needs to set this parameter empirically. A natural question arising here is whether the performance of RMSC is sensitive to the parameter λ. To answer this question, we conduct experiments on BBCSport and WebKB datasets to observe the effects on clustering performance with different values of λ.

Table 2 lists the results with different values of λ on the BBCSport and WebKB . We can observe that: (1) In both of the two datasets, a relative small λ leads to good performance w.r.t. F-score, NMI and Adj-RI. (2) The performance of RMSC only has small variations as long as λ is chosen in a suitable range, i.e., from 0.005 to 0.1, with the obtained results being consistently better than those of the corresponding second best baseline.

In summary, RMSC is relatively insensitive to its parameter λ as long as the parameter is chosen from a suitable range. This makes RMSC easy to use without much effort for parameter tuning.

Conclusions

In this paper, we developed RMSC, a Markov chain method for robust multi-view spectral clustering, which explicitly handles the possible noise in the multi-view input data and recovers a shared transition probability matrix via low-rank and sparse decomposition. To solve the optimization problem of RMSC, we proposed a procedure based on the ALM scheme. Extensive experiments in various real world datasets for clustering show that RMSC has encouraging performance gains over the state-of-the-arts. RMSC is relatively insensitive to its parameter as long as the parameter is in a suitable range, which makes the algorithm easy to use without much effort for parameter tuning.

Acknowledgment

This work was funded in part by National Science Foundation of China (grant No. 61370021, 61003045, 61033010), Natural Science Foundation of Guangdong Province, China (grant No. S2013010011905). Yan Pan is the corresponding author of this paper (panyan5@mail.sysu.edu.cn).

References

Asuncion, A., and Newman, D. 2007. Uci machine learning repository.

Bickel, S., and Scheffer, T. 2004. Multi-view clustering. In *Proceedings of the IEEE International Conference on Data Mining*, volume 4, 19–26.

Cai, J.-F.; Candès, E. J.; and Shen, Z. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20(4):1956–1982.

Chaudhuri, K.; Kakade, S. M.; Livescu, K.; and Sridharan, K. 2009. Multi-view clustering via canonical correlation analysis. In *Proceedings of the International Conference on Machine Learning*, 129–136.

Duchi, J.; Shalev-Shwartz, S.; Singer, Y.; and Chandra, T. 2008. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proceedings of the International Conference on Machine Learning*, 272–279.

Fazel, M.; Hindi, H.; and Boyd, S. P. 2001. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, volume 6, 4734–4739.

Greene, D., and Cunningham, P. 2009. A matrix factorization approach for integrating multiple data views. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 423–438.

Hubert, L., and Arabie, P. 1985. Comparing partitions. *Journal of Classification* 2(1):193–218.

Jiang, Y.-G.; Ye, G.; Chang, S.-F.; Ellis, D.; and Loui, A. C. 2011. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *Proceedings of the International Conference on Multimedia Retrieval*, 29.

Kumar, A., and Daumé, H. 2011. A co-training approach for multi-view spectral clustering. In *Proceedings of the International Conference on Machine Learning*, 393–400.

Kumar, A.; Rai, P.; and Daumé, H. 2011. Co-regularized multi-view spectral clustering. In *Proceedings of the Advances in Neural Information Processing Systems*, 1413–1421.

Lin, Z.; Chen, M.; and Ma, Y. 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*.

Luo, Z.-Q. 2012. On the linear convergence of the alternating direction method of multipliers. *arXiv preprint arXiv:1208.3922*.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press.

Pan, Y.; Lai, H.; Liu, C.; Tang, Y.; and Yan, S. 2013a. Rank aggregation via low-rank and structured-sparse decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Pan, Y.; Lai, H.; Liu, C.; and Yan, S. 2013b. A divide-and-conquer method for scalable low-rank latent matrix pursuit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 524–531.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8):888–905.

Sindhwani, V.; Niyogi, P.; and Belkin, M. 2005. Beyond the point cloud: from transductive to semi-supervised learning. In *Proceedings of the International Conference on Machine Learning*, 824–831.

Srebro, N.; Rennie, J.; and Jaakkola, T. S. 2004. Maximum-margin matrix factorization. In *Proceedings of the Advances in neural information processing systems*, 1329–1336.

Ye, G.; Liu, D.; Jhuo, I.-H.; and Chang, S.-F. 2012. Robust late fusion with rank minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3021–3028.

Zhou, D., and Burges, C. J. 2007. Spectral clustering and transductive learning with multiple views. In *Proceedings of the International Conference on Machine Learning*, 1159–1166.

Zhou, D.; Huang, J.; and Schölkopf, B. 2005. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the International Conference on Machine Learning*, 1036–1043.