Robust Subspace Clustering for Multi-view Data by Exploiting Correlation Consensus

Yang Wang[1] Xuemin Lin[1] Lin Wu[2,3] Wenjie Zhang[1]
Xiaodi Huang[4] Qing Zhang[5]

[1] The University of New South Wales, Sydney, Australia
{wangy,lxue,zhangw}@cse.unsw.edu.au
[2] The University of Adelaide, Australia; [3] Australian Centre for Robotic Vision
lin.wu@adelaide.edu.au
[4] Charles Sturt University, Australia
xhuang@csu.edu.au
[5] Australian E-health Research Centre, Brisbane, Australia
qing.zhang@csiro.edu.au

Technical Report
UNSW-CSE-TR-201511
July 2015

THE UNIVERSITY OF NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

More often than not, a multimedia data described by multiple features, such as color and shape features, can be naturally decomposed of multi-views. Since multi-views provide complementary information to each other, great endeavors have been dedicated by leveraging multiple views instead of a single view to achieve the better clustering performance. To effectively exploit data correlation consensus among multi-views, in this paper we study subspace clustering for multi-view data while keeping individual views well encapsulated. For characterizing data correlations, we generate a similarity matrix in a way that high affinity values are assigned to data objects within the same subspace across views, while the correlations among data objects from distinct subspaces are minimized. Before generating this matrix, however, we should consider that multi-view data in practice might be corrupted by noise. The corrupted data will significantly downgrade clustering results.

We firstly present a novel objective function coupled with an angular based regularizer. By minimizing this function, multiple sparse vectors are obtained for each data object as its multiple representations. In fact, these sparse vectors result from reaching data correlation consensus on all views. For tackling noise corruption, we present a sparsity based approach that refines the angular based data correlation. By using this approach, a more ideal data similarity matrix is generated for multi-view data. Spectral clustering is then applied to the similarity matrix to obtain the final subspace clustering. Extensive experiments have been conducted to validate the effectiveness of our proposed approach.

1 Introduction

It is widely known that many high dimensional data can be seen as a set of samples drawn from a *union* of multiple low-dimensional subspaces. Subspace clustering refers to clustering the data into their original subspaces so as to uncover their underlying structures. Subspace clustering has attracted considerable attentions in computer vision and machine learning communities, with numerous applications including motion segmentation [1], and face clustering [2, 3]. Recent work on sparse representation (SSC) [4, 5, 6], low rank representation (LRR) [3, 7, 2], least square regression (LSR) [8], and their extensions have attracted much attention due to their effectiveness in clustering and robustness to noise. The essence of these approaches lies in constructing an affinity matrix, which is close to a block diagonal matrix with nonzero entries corresponding to the pairs of data points from the same subspace. They differ in the objective functions with different regularization, i.e., either ℓ_1-minimization (SSC), rank minimization (LRR) or ℓ_2-regularization (LSR). The success of SSC, LRR, and LSR supports the fact that if data are sufficiently sampled from independent subspaces, a block diagonal solution can be achieved provided that their objective functions satisfy the Enforced Block Diagonal (EBD) conditions [8].

However, the above methods either encourage sparsity a lot for data selection but lack of grouping effect (SSC), or exhibit strong grouping effect but are short in subset selection (LRR and LSR). It has been observed that both sparsity and grouping effect are important to subspace segmentation. A method of correlation adaptive subspace segmentation by using trace lasso is presented [9], which is able to simultaneously perform data selection and correlated data grouping. Moreover, the authors theoretically prove that trace lasso can also lead to a block sparse solution if the objective function satisfies the conditions of Enforced Block Sparse (EBS).

The nature of visual data in practice is multi-view, *e.g.*, an image can be described by a color view or a shape view. These multiple views often encode compatible and complementary information [10, 11]. This fact naturally motivates one to either leverage all views or simply concatenate them into a monolithic one, in order to improve the performance achieved by a single view. Given data objects with high dimensions that lie in a mixture of subspaces and viewed by multiple views, we attempt to segment data into proper clusters that are consistent among all views by taking advantage of complementary properties of different views. As pointed out by existing multi-view based research [12, 13, 10, 11, 14, 15, 16, 17], the critical point to well leverage the complementary information from different views is to exploit the consensus information among all views, which motivates us to achieve the correlation consensus over subspace clustering for multi-view data objects.

Numerous approaches [17, 18, 19, 20, 21, 22] of multi-view subspace clustering are already available. However, they may either fail to produce the similarity matrix that can characterize the data objects within the same subspace [17, 18], or rely on a rigid data initialization such as Gaussian distribution [19], or even requires the dimensions of projected subspace to be highly parameterized, rather automatically learned. They may not effectively explore the complementary information from multi-views, as they simply follow one-combo-fits-all fashion, *e.g.*, [20, 21], by concatenating all features into one long feature vector, to perform subspace clustering. This, nevertheless, will disregard the local (neighborhood) structure of each view, downgrading the performance of subspace clustering for multi-view data.

To overcome the above-mentioned limitations, we aim to achieve the correlation or similarity consensus among all views, while the data objects within the same subspace

should encode a large similarity and small similarity for data objects within the distinct subspaces for each view. Our approach is based on the fact that one data point for each view in a union of subspaces has a sparse representation with respect to a set of basis vectors formed by all other data points. This inspires us to construct a data similarity matrix for multi-views, from which the subspace clustering for multi-view data objects can be obtained through spectral clustering.

Towards these ends, we propose a novel technique based on trace lasso norm [23] as shown in Eq. 3.1 for each view *e.g.,* i^th view, which learns the sparse coefficients vector *e.g.,* s_k^i, of each data object *e.g.,* x_k over the entire data set. One nice property found in [9] regarding s_k^i is:

Lemma 1 *[9]* ***Trace lasso has the grouping effect,*** **i.e.,** ***the sparse coefficients of a group of correlated data objects within the same subspace are approximately equal. Meanwhile, the sparse coefficients of non-correlated data objects are very small.***

After learning the sparse representation vector for each data object featured with the property indicated by lemma 1 against any individual view, we can trivially get the similarity between any pair-wise data objects via their corresponding sparse representation values for each view. The remaining challenge is how to achieve the consensus of the similarities from all views so as to perform the subspace clustering for multi-view data. To resolve this, we propose a novel angular similarity based regularizer to regularize the sparse codes for the same data from all views to achieve the consensus.

One may wonder why proposing angular based similarity rather than Euclidean distance to coordinate the sparse vectors from all views so as to achieve consensus?

We show an example below to penetrate the illustration:

Example 1 ***Suppose*** $X = \{x_1, x_2, x_3, x_4\}$***, and we have learned the sparse representations for*** x_4 ***from three views e.g.,*** i^{th}***,*** j^{th} ***and*** m^{th} ***views as:*** $s_4^i = [1, 1, 0]$***,*** $s_4^j = [3, 3, 0]$ ***and*** $s_4^m = [0, 0, 1]$ ***via Eq.*** (3.1) ***for each view, the same coefficient is formed by using trace lasso as per Lemma 1.***

The above example indicates s_4^i characterize the same correlations with s_4^j, since x_4 has the similar correlations with other three data objects for both i^{th} and j^{th} views. Specifically, x_4 has the large correlations with x_1 and x_2, but no(small) correlations with x_3. If we evaluate the similarity according to Euclidean distances, then the similarity between s_4^i and s_4^j is smaller than s_4^k, implied by large Euclidean distances. That apparently violates the fact. Therefore, to address such issue, we propose the angular similarity metric, leading to the small angular between s_4^i and s_4^j, meanwhile lead to large angular for s_4^m.

In practice, there may be disturbing noises, missing values and corruptions available for view-specific feature representations. To achieve the robustness and correlations consensus, we propose to decompose the sparse representation vector of each data object into two parts for all views.

- The first part is the latent consensus sparse representation shared by all views. We propose to learn such latent consensus sparse representation for each data object by minimizing the angular similarity between each view-specific sparse representations learned via Eq.(3.1) and it, so as to achieve the data correlation consensus encoded in sparse representations among all views.

- The second part describes the possible noise corruptions for each view-specific feature representations and the view-specific sparse representations for each individual view, leading to non-precise and non-consistent data correlations encoded

in sparse representations. As observed, such noise sparse representations are sparsely distributed, therefore, we propose to model it via ℓ_1 norm.

For our method, a novel objective function is proposed by leveraging our angular similarity based regularizer and ℓ_1 norm sparse representation to address the possible noise corruptions for each view-specific feature representations. The final consensus sparse representations and noise term are yielded by minimizing the proposed objective function. The consensus sparse representations are further utilized for subspace clustering over multi-view data. For simplicity, we illustrate our overall framework in Fig.1.1 from two views, which can be naturally extended to multi-view scenario as our proposed technique later.

Please note that the above decomposition model must be upon the following critical claim: two distinct sparse codes for the same data object across views encode similar values in their entries, motivated by common assumption for the multi-view clustering [22]: the same data object set under different views should reveal the similar correlations.

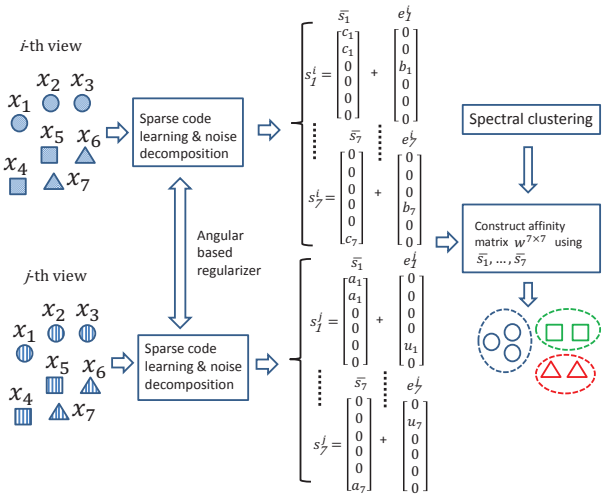


Figure 1.1: Overview of our framework. For each data x_k in a given set of data objects with two views $X = \{x_k\}_{k=1}^{7}$ that might be corrupted by noise, we learn its robust sparse representation with respect to other data points in the same subspace as x_k. This is achieved by using trace lasso as a sparsity promoter, which can automatically seek sparse coefficients, regularized by an angular-based regularizer to reach consensus on all views. By modeling noise via sparse decomposition, *e.g.*, e_k^i, we can recover latent shared sparse vectors ($\overline{s_k}$) from which the affinity matrix is constructed for subsequent subspace clustering.

Our major contributions are summarized as follows.

- To the best of our knowledge, this is the first work on applying trace lasso into multi-view data for subspace clustering.
- To exploit the data correlation consensus on views, we propose a novel angular based regularizer over the data sparse codes in multi-views. The objective function is minimized under the regularization of this new regularizer.

This paper is an extension of [11] with additionally constructive contributions below.

- To cope with input data that might be corrupted by noise, we develop an approach that can effectively recover a shared latent sparse representation from multiple views, which well reflects the true clustering information.
- More extensive experiments have been conducted on real-world image datasets, which demonstrate the effectiveness of exploiting the correlation consensus among sparse codes of data objects across views for multi-view subspace clustering.

2 Related Work

In this section, we briefly review existing typical work related to multi-view subspace clustering.

Using a co-training based method[24, 10], Kumar *et al.* [17] constructs a compatible multi-view similarity matrix in eigen-subspaces spanned by Laplacian matrix, such that the similarity matrix in one view is affected by that in another view. However, they simply calculate the similarity matrix in a K-nearest neighbors manner. This degrades the performance if data points are nearby the intersection of two distinct subspaces. That is, the neighborhoods of a data point may cover data points from different subspaces. The same problem exists in [18] as well. In [19], the multi-view data are projected into one common subspace, then the clustering algorithm, *e.g.*, K-means, is applied to yield the subspace clustering results. Such a method, however, is sensitive to data initialization. Specifically, it requires that the data initialization should strictly follow the Gaussian distribution while keeping different groups of data objects separated. Besides, the number of dimensions for the projected subspace needs to be known in advance. Matrix factorization is also utilized to perform subspace clustering for multi-view data, such as [20, 21]. Its essential idea is that the features of heterogeneous views are first concatenated into a single-long feature, then non-negative matrix factorization is applied to obtain subspace clustering results. One limitation of such a one-combo-fits-all strategy is that the data correlation information in each of original view-specific feature space is not well exploited. To overcome this limitation, [22] proposes a joint non-negative matrix factorization on each individual view to compute distinct coefficient matrices, which are then regularized towards a common consensus that represents the clustering structure shared by all views. This method, however, suffers from the drawback that the dimension number of latent reduced subspace needs to be manually parameterized, rather than automatically determined.

3 Proposed Technique

In this section, we first formalize the problem of subspace clustering on multi-view data, then model data correlations in a single view, followed by a non-trivial extension towards correlation consensus on multiple views. After that, we present a novel technique for noise decomposition in multi-view data.

3.1 Notations and Problem Definition

Let $X = \{x_k\}_{k=1}^n$ be a set of data points with n data instances. Suppose that each data object has V views. Without loss of generality, for the i-th view, we have $X^i = \{x_k^i\}_{k=1}^n$, $(i = 1, \ldots, V)$, where x_k^i is the feature representation of x_k under the i-th view. We denote s_k^i as the sparse representation vector of x_k^i based on X^i. The trace lasso [23] is defined as $\|X^i Diag(s_k^i)\|_*$, where $Diag(s_k^i)$ is the diagonal matrix with its i-th diagonal element corresponding to the i-th entry of s_k^i, and $\|A\|_*$ is the nuclear norm (the summation of all the singular value) of a matrix A. The norms of $\|a\|_1$, $\|a\|_2$ and $\|a\|_\infty$ denote the ℓ_1 (sum of absolute value of each entry), ℓ_2 norm of a vector a and ℓ_∞ (maximum value of entry). Considering that multi-view data objects are possibly corrupted by noise, *e.g.*, possible corrupted or missing values for any feature representations, we model the view-specific noise by a sparse decomposition e_k^i for each s_k^i.

With the notations defined above, we aim to learn the latent robust sparse representations $\overline{s}_k$ for each x_k^i $(i = 1, \ldots, V)$ shared by all views of possible noisy multi-view data objects. The representations are then utilized to construct a compatible similarity matrix W for multi-view data objects. The subspace clustering result is achieved by applying spectral clustering upon W.

3.2 Modeling Data Correlations in single view

The challenge of modeling data correlations in each individual view is to ensure high correlations for data points within the same subspace, while eliminating connections among data objects from distinct subspaces. To achieve this goal, we employ the trace lasso norm to learn the sparse representations for each data object.

As shown in [9], trace lasso is more adaptive than ℓ_1 or ℓ_2 norm, and it is equal to ℓ_1-norm or ℓ_2-norm if data points are uncorrelated (orthogonal) or highly correlated. Thereby, we have $\|s_k^i\|_2 \leqslant \|X^i Diag(s_k^i)\|_* \leqslant \|s_k^i\|_1$. The sparse representation s_k^i of x_k^i can well reflect the correlation between x_k^i and other data points under the i-th view. In particular, if we normalize each column of X to one, the problem of learning sparse code vectors of each data point in the i-th view can be formulated below:

$$\min_{s_k^i} \frac{1}{2}\|x_k^i - X_k^i s_k^i\|_2^2 + \lambda \|X_k^i Diag(s_k^i)\|_*, \tag{3.1}$$

where X_k^i represents the data set excluding x_k^i. The parameter λ controls the effect of the trace lasso term. Through trace lasso, s_k^i is composed of the approximately equal yet large coefficients on a few data objects, implying their strong correlations with respect to x_k^i. Meanwhile, the coefficients of data having no (weak) correlation to x_k^i are set to 0. This conclusion also holds for the $j(j \neq i)$-th view.

The convex optimization problem (3.1) can be solved by using the Alternating Direction Method (ADM) [9], which can converge globally. The work in [23] indeed introduces an iteratively reweighted least squares algorithm for estimating the vector s_k^i, however, the solution is not necessarily globally optimal due to an additional term to avoid non-invertible. To apply the ADM method, we first convert the problem (3.1) into its equivalent formulation as follows,

$$\min_{s_k^i, M_k^i} \frac{1}{2}\|x_k^i - X_k^i s_k^i\|_2^2 + \lambda \|M_k^i\|_*, \, s.t. M_k^i = X_k^i Diag(s_k^i). \tag{3.2}$$

Then, problem (3.2) can be solved by ADM, which works on the following augmented

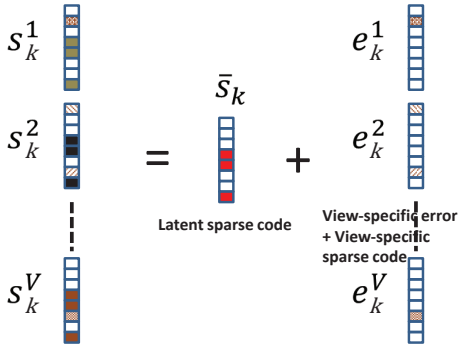


Figure 3.1: For any data object x_k, $(k = 1, \ldots, n)$, its sparse code s_k^i associated with an individual view i $(i = 1, \ldots, V)$ can be naturally decomposed into two parts: a shared latent sparse code $\bar{s}_k$ that reflects the true clustering information, and e_k^i encoding a view-specific deviation error vector that encodes the noise in sparse codes in each view, together with a view-specific sparse representation for each view.

Lagrangian function:

$$\begin{aligned} L(M_k^i, s_k^i) &= \frac{1}{2}\|x_k^i - X_k^i s_k^i\|_2^2 + \lambda\|M_k^i\|_* \\ &+ Tr((Y_k^i)^T(M_k^i - X_k^i Diag(s_k^i))) + \frac{\alpha}{2}\|M_k^i - X_k^i Diag(s_k^i)\|_F^2, \end{aligned} \tag{3.3}$$

where $Y_k^i \in \mathbb{R}^{d\times n}$ is the Lagrange multiplier, and $\alpha > 0$ is the penalty parameter for the violation of linear constraint. $L(M_k^i, s_k^i)$ is separable to two subproblems with regard to M_k^i and s_k^i, respectively. Hence, s_k^i can be updated with a closed form solution, that is, for iteration t, $s_k^i = A_k^i\left((X_k^i)^T x_k^i + diag((X_k^i)^T((Y_k^i)^t + \alpha^t(M_k^i)^{t+1}))\right)$ where $A_k^i = \left((X_k^i)^T X_k^i + \alpha^t Diag(diag((X_k^i)^T X_k^i))\right)^{-1}$.

3.3 Exploiting Correlation Consensus in Multiple Views

It is non-trivial to learn sparse representations that characterize the correlation consensus on all views by considering the subspaces from which they come. This is because many multi-view learning methods, *e.g.*, [25], rely on common label spaces across views. Without label information, it thus becomes more challenging to exploit their consensus property shared by views.

The principle of multi-view clustering [22] is that the true underlying clustering would assign corresponding data objects across views to the same cluster. According to this principle, we propose to exploit the data correlation consensus among all views, which further determines the subspace clustering on multi-view data objects.

Basic idea: Angular based similarity. We attempt to effectively exploit the correlation consensus on multi-views while keeping their individuality well-encapsulated. One natural question is how to quantify the similarities among the sparse representations of the same data object with different views. One may consider a distance metric, *e.g.,* Euclidean distance. However, as aforementioned, a small Euclidean distance cannot indicate a similar data correlation shared by two sparse representations, illustrated by Example 1 in Section 1. Fortunately, the sparse code of any data object, *e.g.*, s_k^i for x_k^i, can well reflect the correlation between x_k^i and other data points under the i-th

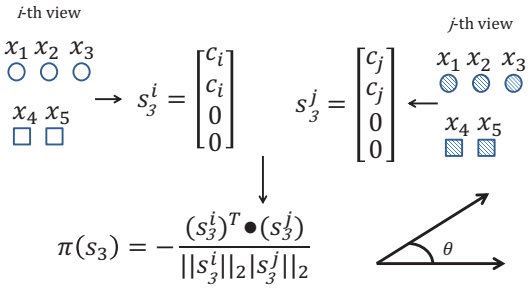


Figure 3.2: An example of illustrating the angular based similarity. The same shape indicates that data objects are in the same cluster, and the same filled pattern means that data objects come from the same view space. The sparse code of x_3 yielded by trace lasso is s_3^i (s_3^j) under the i-th (j) view. In the case of noise free, the coefficients in s_3^i (s_3^j) can correctly indicates the similarity between x_3 and $\{x_j\}_{j\neq 3}$ in the i-th (j) view. Thus, minimizing $\pi(s_3)$ is equivalent to making sparse codes across views reach consensus in terms of their similarities.

view. Another observation is that sparse codes for different views have the same dimension. The two observations motivate us to quantify the **angular based similarity** over sparse codes of multi-view data objects. This is equivalent to quantifying the data correlations regarding data object set X under various views. As a result, we propose a novel regularizer term, $\pi(s_k)$, to encode the cosine similarity among s_k^i regarding x_k^i in the i-th view.

Mathematically, $\pi(s_k)$ can be defined as

$$\pi(s_k) = -\sum_{i,j\neq i}^{V} \frac{(s_k^i)^T \cdot s_k^j}{\|s_k^i\|_2 \|s_k^j\|_2}. \tag{3.4}$$

The objective function in Eq. (3.7) needs to be minimized. Thus, we add minus sign "$-$" in Eq. (3.4) to encourage the large value of $\sum_{i,j\neq i}^{V} \frac{(s_k^i)^T \cdot s_k^j}{\|s_k^i\|_2 \|s_k^j\|_2}$, as well as the small angle among various vectors of view-specific sparse representation for s_k.

Intuition of angular similarity to achieving consensus. Minimization of $\pi(s_k)$ can yield the correlation consensus on data objects across views. An intuitive example is shown in Fig.3.2 where we have a set of five data objects $\{x_k\}_{k=1}^5$ with two views. For each data object such as x_3, we calculate its sparse representation s_3^i (s_3^j) under the i-th view by Eq.(3.1) based on the dictionary composed of all five data objects. Owing to its grouping effect, trace lasso can well capture the correlations among data objects. The (nearly) equal positive coefficients c_i (c_j) in s_3^i (s_3^j) are generated to encode the strong correlation between x_1 and x_2, and 0 between x_4 and x_5 under both i-th and j-th views. It is straightforward that s_3^i and s_3^j have a large cosine similarity value. This value in turn measures the relations between two views by capturing the data correlation consensus between multi-view sparse codes.

The above observations are based on the assumption that input data objects are noise-free. However, such an ideal case is almost impossible in practice where either data objects may be noisy, or feature values are corrupted. This may result in inaccurate coefficients of sparse vectors. In what follows, we will tackle the problem of the noise of subspace clustering for multi-view data by recovering a latent sparse representation. Such a representation reflects the true data correlations shared by all views.

To this end, we recover a common sparse representation of each data, which is consistent across views by effectively decomposing noise in each view. As suggested in [4, 26], one can fill in missing entries or correct errors using sparse decomposition since visual features contain sufficient clustering information, whilst features in each view might have a small portion of information corrupted by noise. In what follows, we show that our method can also cluster data points with corrupted entries in multi-view data objects by recovering and modeling the error matrix by sparse decomposition. Formally, let s_k^i be the sparse code for x_k^i regarding the i-th view that may be corrupted . Then, we decompose s_k^i into a shared latent sparse vector $\overline{s}_k$, and e_k^i encoding a noise corruption for feature representation and specific sparse representations for the i-th view; then, we have:

$$\forall i, \forall k, s_k^i = \overline{s}_k + e_k^i. \tag{3.5}$$

Eq.(3.5) is based on the basic assumption that the sparse representation in each individual view is robust enough to contain most of the clustering information, although noise might lead to the small number of data points assigned to wrong clusters. Consequently, $\overline{s}_k$ is the consensus correlation of a data point with respect to other points across all views. We illustrate this intuition in Fig.3.1.

Now we are ready to present how to extend the learning technique from a single view to multiple views.

$$\pi(\overline{s}_k) = -\sum_i^V \frac{(s_k^i)^T \cdot \overline{s}_k}{\|s_k^i\|_2 \|\overline{s}_k\|_2}, s.t.\ \ s_k^i = \overline{s}_k + e_k^i. \tag{3.6}$$

Intuitively, Eq.(3.6) recovers a shared latent sparse representation for multiple views, which is regularized by the angular based similarity.

Considering Eq.(3.5) and Eq.(3.6), we formulate the problem of learning robust sparse representation for multi-view data objects as follows:

$$\begin{aligned} &\min_{\overline{s}_k, e_k^i} \sum_i^V \beta_i \left(\frac{1}{2} \|x_k^i - X_k^i s_k^i\|_2^2 + \lambda \|X_k^i Diag(s_k^i)\|_* + \eta \|E^i\|_1 \right) + \gamma \pi(\overline{s}_k), \\ &s.t.\ \ s_k^i = \overline{s}_k + e_k^i, E^i = [e_1^i, \ldots, e_n^i], \overline{s}_k \geqslant 0, \end{aligned} \tag{3.7}$$

where β_i balances the contribution from the i-th view, $\sum_{i=1}^V \beta_i = 1$, and γ is a weight parameter on $\pi(\overline{s}_k)$ to regulate the correlation consensus over multi-views. E^i represents the difference collection between s_k^i and $\overline{s}_k$ for the i-th view, where $k = 1, \ldots, n$ and η is a non-negative balance parameter.

Since we assume that sparse representation in each view is robust and has enough information to identify most of the clustering structure, it is reasonable to hypothesize that there is only a small fraction of elements in s_k^i being apparently different from the corresponding ones in $\overline{s}_k$. Thus, the deviation error matrix and view-specific sparse representation E^i tends to be sparse. In summary, we aim to learn sparse codes $\overline{s}_k$ of x_k^i for each view, by optimizing Eq. (3.7) for subspace clustering on noisy multi-view data. Specifically, for each data object, we attempt to recover a shared latent sparse representation that reflects the underlying true data correlation with other data objects. These shared sparse codes are regarded as a crucial input to compute a more accurate affinity matrix on data objects, which can be subsequently combined with some clustering strategies, *e.g.*, spectral clustering, to disclose clusters.

4 Optimization Strategy

The difficulty of optimizing Eq. (3.7) lies in its non-joint-convex for $M_k^i = X_k^i Diag(s_k^i)$ and s_k^i, along with non-smoothness for trace lasso. We alternatively optimize each variable by fixing others. Each s_k^i is initialized by optimizing Eq. (3.7) via Alternative Direction Method (ADM) [9] with the parameter λ of 0.15. We derive an equivalent variational formulation of the trace norm [27]. Assume $M \in \mathbb{R}^{n\times m}$, the trace norm of M equals

$$\|M\|_* = \frac{1}{2}\inf_{S\geq 0} tr(M^T S^{-1} M) + tr(S), \tag{4.1}$$

where the infimum is achieved when we have $S = (MM^T)^{1/2}$. Then we recast Eq. (3.7) as

$$\begin{aligned}
&\min_{\overline{s}_k, e_k^i} \inf_{M_k^i \geq 0} \sum_i^V \frac{\beta_i}{2}\|x_k^i - X_k^i s_k^i\|_2^2 \\
&+ \frac{\lambda\beta_i}{2}\left(tr\left((S_k^i)^2 (X_k^i)^T (M_k^i)^{-1} X_k^i\right) + tr(M_k^i)\right) \\
&+ \beta_i \eta \|E^i\|_1 + \gamma\pi(\overline{s}_k), \\
&s.t.\ \ s_k^i = \overline{s}_k + e_k^i, E^i = [e_1^i, \ldots, e_n^i], \overline{s}_k \geqslant 0,
\end{aligned} \tag{4.2}$$

where $S_k^i = Diag(s_k^i)$.

4.1 Solving M_k^i

With other variables fixed, the optimization problem in Eq. (4.2) is convex with respect to M_k^i. We conduct a coordinate descent procedure to optimize M_k^i for each x_k^i. Given s_k^i, we enjoy the closed form solution of M_k^i as

$$M_k^i = (X_k^i (S_k^i)^2 (X_k^i)^T)^{1/2}. \tag{4.3}$$

4.2 Solving $\overline{s}_k$ and e_k^i

To optimize $\overline{s}_k$ and e_k^i, we introduce an optimization procedure to solve this problem via the Augmented Lagrangian Multiplier (**ALM**) scheme [28]. For ease of the representation, we define $A^{(i)} = \inf \frac{\beta_i}{2}\|x_k^i - X_k^i s_k^i\|_2^2 + \frac{\lambda\beta_i}{2}\left(tr\left((S_k^i)^2 (X_k^i)^T (M_k^i)^{-1} X_k^i\right) + tr(M_k^i)\right)$. By introducing an auxiliary variable p_k, we convert problem (3.7) into the following form:

$$\min_{p_k, e_k^i, \overline{s}_k} \gamma\pi(p_k) + \sum_{i=1}^V (A^{(i)} + \beta_i \eta \|E^i\|_1) \tag{4.4}$$

The corresponding augmented Lagrange function is:

$$\begin{aligned}
&\mathcal{L}(\overline{s}_k, p_k, e_k^i) = \gamma\pi(p_k) + \sum_{i=1}^V (A^{(i)} + \langle y_k^i, \overline{s}_k + e_k^i - s_k^i\rangle \\
&+ \frac{\mu}{2}\|\overline{s}_k + e_k^i - s_k^i\|_F^2) + \langle z_k, \overline{s}_k - p_k\rangle + \frac{\mu}{2}\|\overline{s}_k - p_k\|_F^2. \\
&s.t.\ \ \overline{s}_k \geqslant 0,
\end{aligned} \tag{4.5}$$

where z_k and y_k^i are Lagrange multipliers, $\langle\cdot,\cdot\rangle$ denotes the inner product of matrices, and $\mu > 0$ is an adaptive penalty parameter. Next, we will present the update rules for each p_k, $\overline{s}_k$, and e_k^i, by minimizing $\mathcal{L}$ in Eq.(4.5) with other variables being fixed.

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

$$\min_{p_k} \gamma\pi(p_k) + \frac{\mu}{2}\|\overline{s}_k - p_k + \frac{z_k}{\mu}\|_F^2, \tag{4.6}$$

Considering that p_k encodes the latent structure of each sparse code across all views, we thus have $p_k = [s_k^i - e_k^i, \ldots, s_k^V - e_k^V]$. Thereafter, we obtain a matrix $P = [p_1, p_2, \ldots, p_n]$, then Eq. (4.6) can be rewritten as

$$\min_{P} \gamma\|P\|_* + \frac{\mu}{2}\|\overline{S} - P + \frac{Z}{\mu}\|_F^2, \tag{4.7}$$

where $\overline{S} = [\overline{s}_1, \overline{s}_2, \ldots, \overline{s}_n]$ and $Z = [z_1, z_2, \ldots, z_n]$. Eq. (4.7) can be solved by Singular Value Threshold method [29]. More specifically, let $U\Sigma V^T$ be the SVD form of $(\overline{S} + \frac{Z}{\mu})$. The solution to Eq.(4.7) is $P = U\mathcal{S}_{1/\mu}(\Sigma)V^T$, where $\mathcal{S}_\delta(\mathbf{X}) = \max(X - \delta, 0) + \min(X + \delta, 0)$ is the shrinkage operator [28].

Solving e_k^i. The subproblem w.r.t. $\overline{e}_k^i$ can be simplified as:

$$\min_{e_k^i} \beta_i\|e_k^i\|_1 + \frac{\mu}{2}\|e_k^i - (s_k^i - \overline{s}_k - \frac{y_k^i}{\mu})\|_F^2, \tag{4.8}$$

which has a closed form solution $e_k^i = \mathcal{S}_{\beta_i/\mu}(s_k^i - \overline{s}_k - \frac{y_k^i}{\mu})$. $\mathcal{S}_\theta(\mathbf{X}) = \max(\mathbf{X} - \theta, 0) + \min(\mathbf{X} + \theta, 0)$ is the shrinkage operator [28].

Solving $\overline{s}_k$. With other variables being fixed, we update $\overline{s}_k$ by solving

$$\begin{aligned}\overline{s}_k = \arg\min_{\overline{s}_k} \frac{\mu}{2}\sum_{i=1}^{V}\|\overline{s}_k + e_k^i - s_k^i + \frac{y_k^i}{\mu}\|_F^2 \\ + \frac{\mu}{2}\|\overline{s}_k - p_k + \frac{z_k}{\mu}\|_F^2 . s.t.\ \ \overline{s}_k \geqslant 0.\end{aligned} \tag{4.9}$$

For convenience of the representation, we define $\mathbf{c} = \frac{1}{V+1}\left(p_k - \frac{z_k}{\mu} + \sum_{i=1}^{V}(s_k^i - e_k^i - \frac{y_k^i}{\mu})\right)$. With simple algebraic manipulation, the problem (4.9) can be rewritten as

$$\overline{s}_k = \arg\min_{\overline{s}_k} \frac{1}{2}\|\overline{s}_k - \mathbf{c}\|_F^2, s.t.\ \ \overline{s}_k \geqslant 0. \tag{4.10}$$

The Lagrangian of the problem in Eq.(4.10) is

$$\mathcal{L}(\overline{s}_k, \zeta) = \frac{1}{2}\|\overline{s}_k - \mathbf{c}\|_2^2 - \zeta \cdot \overline{s}_k,$$

where $\zeta \in \mathbb{R}_+^{n-1}$ is a vector of non-negative Lagrange multipliers. Differentiating with respect to $\overline{s}_k[i]$ and comparing to zero gives the optimality condition, $\frac{\partial\mathcal{L}}{\partial\overline{s}_k[i]} = \overline{s}_k[i] - c[i] - \zeta[i] = 0$. The complementary slackness KKT condition implies that whenever $\overline{s}_k[i] > 0$, we must have $\zeta[i] = 0$. Thus, if $\overline{s}_k[i] > 0$, we have

$$\overline{s}_k[i] = c[i].$$

All the non-negative elements of the vector $\overline{s}_k$ are tied via a single variable, thus, identifying the indices of these elements yields a much simpler problem.

Lemma 2 *[30].* **Let $\overline{s}_k$ be the optimal solution to Eq.**(4.10)**, i and j be two indices such that $c[i] > c[j]$, if $\overline{s}_k[i] = 0$ then $\overline{s}_k[j]$ must be zero as well.**

Algorithm 1: Algorithm for solving problem (4.10).

Input: A vector $\mathbf{c} \in \mathbb{R}^{n-1}$.
Output: $\overline{s}_k$.
Sort $\mathbf{c}$ into $\mathbf{b}$: $b[1] \geqslant b[2] \geqslant \ldots \geqslant b[n-1]$;
Find $\hat{j} = \max\{j \in [n-1] : b[j] > 0\}$;
for $i = 1, \ldots, \hat{j}$ **do**
  $\overline{s}_k[i] = b[i]$;
for $i = \hat{j}+1, \ldots, n-1$ **do**
  $\overline{s}_k[i] = 0$;
return $\overline{s}_k$

Denote by I the set of indices of non-zero components of the sorted optimal solution, the Lemma 2 implies that $I = [\varrho]$ for some $1 \leqslant \varrho \leqslant n-1$. We can find the optimal ϱ by the following lemma once we sort $\mathbf{c}$ in descending order.

Lemma 3 Let $\overline{s}_k$ be the optimal solution to the minimization problem in Eq.(4.10)**. Let $\mathbf{b}$ denote the vector obtained by sorting $\mathbf{c}$ in a descending order. Then, the number of strictly positive elements in $\overline{s}_k$ is $\varrho(\mathbf{b}) = \max\{j \in [n-1] : b[j] > 0\}$.**

The pseudo-code describing the procedure for solving problem (4.10) is given in Algorithm 1. Overall, we summarize the process of recovering the sparse representation of multi-view data from corruption in Algorithm 2. The Algorithm 2 is repeatedly performed until it meets the stopping criteria. This stopping criteria works by the follow: For any data point x_k within any view i, if the minimum value of the maximum entry of the difference between its latent sparse vector $\overline{s}_k$ and auxiliary variable p_k is less than the threshold ϵ or the minimum value of maximum entries of the difference of each point's latent sparse vector $\overline{s}_k$, together with e_k^i and original sparse vector s_k^i is less than the threshold ϵ.

Algorithm 2: Robust subspace clustering on multi-view data.

Input: $X = \{x_k\}_{k=1}^n$, V views, initialize $\{s_k^i\}(k = 1, \ldots, n; i = 1, \ldots, V)$ by optimizing Eq.(3.1), λ, η, β_i, γ.
Output: $\overline{s}_k$, e_k^i, $(k = 1, \ldots, n; i = 1, \ldots, V)$.
Initialize: $\overline{s}_k$=$\mathbf{0}$, p_k=$\mathbf{0}$, z_k=$\mathbf{0}$, y_k^i=$\mathbf{0}$, e_k^i=$\mathbf{0}$, $\mu = 10^{-6}$, $\rho = 1.9$, $\max_\mu = 10^{10}$, $\epsilon = 10^{-3}$.
repeat
  $\mathbf{c} = \frac{1}{V+1}\left(p_k - \frac{z_k}{\mu} + \sum_{i=1}^{V}(s_k^i - e_k^i - \frac{y_k^i}{\mu})\right)$;
  for $k = 1, \ldots, n$ **do**
    Form X_k by eliminating x_k;
    Run Algorithm 1 using $\mathbf{c}$ as input to update $\overline{s}_k$;
    for $i = 1, \ldots, V$ **do**
      Update e_k^i via Eq.(4.8);
    Update p_k via Eq.(4.6);
    $z_k \leftarrow z_k + \mu(\overline{s}_k - p_k)$;
    for $i = 1, \ldots, V$ **do**
      $y_k^i \leftarrow y_k^i + \mu(\overline{s}_k + e_k^i - s_k^i)$;
    $\mu \leftarrow \min(\mu\rho, \max_\mu)$;
until $\min\left(\min_{k,i} \|\overline{s}_k + e_k^i - s_k^i\|_\infty, \min_k \|\overline{s}_k - p_k\|_\infty\right) \leq \epsilon$;

4.3 Subspace Clustering for Multi-view Data

The convergence for optimizing Eq. (3.7) is determined by optimizing $\overline{s}_k$, as M_k^i enjoys a closed form at each step. On the other hand, the convergence study on updating $\overline{s}_k$ is demonstrated in experimental part indicating that our algorithm is fast to reach convergence.

So far, we have presented how to use the latent sparse representation for clustering multi-view data objects from multiple subspaces. The optimal solution to Eq.(3.7), $\overline{s}_k \in \mathbb{R}^{n-1}$, is a vector whose nonzero entries correspond to points in X_k with the same subspace as x_k. Thus, by inserting a zero entry at the i-th entry of $\overline{s}_k$, we derive an n-dimensional vector, $\hat{s}_k \in \mathbb{R}^n$, whose nonzero entries correspond to points in X that lie in the same subspace as x_k. After solving Eq.(3.7) at each point $x_k, (k = 1, \ldots, n)$, we obtain a matrix of coefficients $C = [\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_n] \in \mathbb{R}^{n \times n}$. The similarity between data objects i and j is then calculated as $W(i, j) = \frac{C_{ij}+C_{ji}}{2}$. The final affinity matrix regarding all views is obtained as W. It is ready to perform subspace clustering.

5 Experimental Results

In this section, we comprehensively evaluate the performance of our approach by comparing with state-of-the-art baselines.

Table 5.1: Clustering performance on three real-world image datasets.

Method	Accuracy(%)			Normalized Mutual Information(%)		
	UCI	CMU PIE	PASCAL	UCI	CMU PIE	PASCAL
BSV-SSC	69.028±0.010	70.255±0.013	62.381±0.009	52.622±0.010	51.457±0.009	56.403±0.009
ConcatSSC	70.832±0.007	65.735±0.011	60.228±0.006	54.055±0.011	48.829±0.008	51.520±0.009
CCA	76.114±0.004	77.482±0.006	70.010±0.006	61.671±0.007	60.227±0.006	58.731±0.008
Co-train-SC	84.642±0.002	83.070±0.007	76.125±0.004	77.011±0.005	71.420±0.004	66.592±0.006
MultiNMF	88.014±0.003	86.807±0.004	80.227±0.007	80.257±0.010	77.561±0.011	69.882±0.011
SSC-Con	93.149±0.001	90.006±0.003	87.540±0.004	86.553±0.009	82.716±0.010	74.660±0.010
Ours	**95.203±0.003**	**91.120±0.006**	**89.720±0.011**	**89.153±0.012**	**86.026±0.013**	**79.333±0.012**

5.1 Datasets and Competitors

Datasets. Three real-world image datasets are used in our experiments.

• **UCI Handwritten Digit Dataset** [1]: This handwritten digits (0-9) database consists of 2,000 examples. There are 10 subspace clustering in total and for each digit we select its first 50 samples. All algorithms are performed on each digit, and the mean and standard deviation of errors are reported. We construct two views, with the first view being 76 Fourier coefficients and the second view being 240 pixel averages in 2×3 windows.

• **CMU PIE Face Database** [2]: This database contains 68 subjects with 41,368 face images. We construct four subspace clustering tasks based on randomly selecting 5, 8, 10, 15 subjects face images of this database. These images are first projected into 6-dimensional subspace by PCA, respectively. Each image is 32×32, and we use three kinds of features as three views: LBP (256-dim), HOG (100-dim), and grey levels (128-dim).

• **PASCAL VOC 2010 Database**: This dataset contains 10,103 images from 20 classes, and thus there are 20 clustering problems in total. We first use PCA to project the data into a 12-dimensional subspaces, and then all algorithms are performed on each class to calculate mean and standard deviation errors. We adopt four types of

[1] http://archive.ics.uci.edu/ml/datasets.html

[2] http://www.ri.cmu.edu/projects/project_418.html

Table 5.2: Statistics of the three datasets.

Dataset	Instance	# of view	# of cluster
UCI digit	2,000	2	10
CMU PIE face	41,368	3	68
PASCAL VOC 2010	10,103	4	20

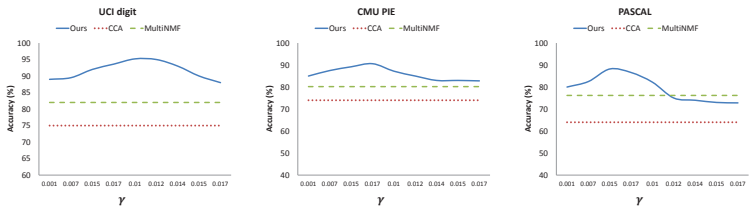
features as four views: color moments (255-dim), color histogram (64-dim), edge distribution (73-dim), and wavelet texture (128-dim).

The statistics of the datasets are summarized in Table 5.2.

Competitors. We choose the followings algorithms as competitors in our experiments.

- The best single view on Sparse Subspace Clustering (**BSV-SSC**): use the individual view which achieves the best subspace clustering performance with a single view of data. The sparse representation is obtained by ℓ_1-minimization.
- Concatenating features of each view on Sparse Subspace Clustering (**ConcatSSC**): stitch features from all views to be a feature vector to perform Eq. (3.1), and then directly perform subspace clustering on the obtained sparse representations.
- Multi-view clustering via Canonical Correlation Analysis (**CCA**) [19]: construct the projections from high-dimensional data into low-dimensional subspaces by using multiple views of the data via the canonical correlation analysis.
- Co-training based multi-view Spectral Clustering (**Co-train-SC**) [17]: develop a multi-view spectral clustering that has a favor of co-training to reach an agreement on clustering assignment.
- Multi-view Nonnegative Matrix Factorization (**MultiNMF**) [22]: use NMF-based multi-view clustering to search for a factorization that gives compatible clustering solutions across multiple views.
- Multi-view Sparse Subspace Clustering via correlation Consensus (**SSC-Con**) [11]: develop a multi-view subspace clustering that advocates adaptive data correlation and reaches consensus across views by angular based regularizer.
- **SSC**-ℓ_1 [4]: Replace the trace lasso minimization in our objective function to be ℓ_1-minimization.
- **LSR** [8]: Replace the trace lasso term in our objective function to be Least Squares Regression, which works by using ℓ_2-minimization for subspace clustering.
- **LRR** [7]: Replace the trace lasso term in our objective function to be Low-Rank Representation, which works to group correlated data together by rank minimization.

The clustering results are evaluated by comparing the obtained label of each data point with the label provided by the datasets.



(a) (b) (c)

Figure 5.1: The study on parameter γ over three image databases.

5.2 Experimental Settings

Evaluation Metrics. Two widely used metrics, clustering accuracy and normalized mutual information, are used to measure the clustering performance [17]. The clustering accuracy discovers the one-to-one relationship between clusters and classes measures to which extent each cluster contains data points from the corresponding class. Clustering accuracy is defined as follows:

$$Accuracy = \frac{\sum_{i=1}^{n} \delta(map(r_i), l_i)}{n}, \tag{5.1}$$

where r_i denotes the cluster label of x_i, and l_i denotes the true class label, n is the total number of images, $\delta(x, y)$ is the function that equals one if $x = y$ and equals zero otherwise, and $map(r_i)$ is the permutation mapping function that maps each cluster label r_i to the equivalent label from the database. The Normalized Mutual Information (NMI) is used to determine the quality of clusters, which can be estimated by

$$NMI = \frac{\sum_{i=1}^{c}\sum_{j=1}^{c} n_{i,j} \log \frac{n_{i,j}}{n_i \hat{n}_j}}{\sqrt{(\sum_{i=1}^{c} n_i \log \frac{n_i}{n})(\sum_{j=1}^{c} \hat{n}_j \log \frac{\hat{n}_j}{n})}}, \tag{5.2}$$

where n_i denotes the number of images contained in the cluster C_i ($1 \leqslant i \leqslant c$), $\hat{n}_j$ is the number of images belonging to the class L_j ($1 \leqslant j \leqslant c$), and $n_{i,j}$ denotes the number of images that are in the intersection between cluster C_i and class L_j.

The larger the NMI is, the better the clustering results will be.

To evaluate the quality of clusters $\{U_l\}_{l=1}^{K}$, we use Davies-Bouldin Index (DBI) to measure the uniqueness of clusters w.r.t. the unified similarity measure.

$$DBI(\{U_l\}_{l=1}^{K}) = \frac{1}{K}\sum_{i=1}^{K} \max_{j \neq i} \frac{d(c_i, c_j)}{\sigma_i + \sigma_j}, \tag{5.3}$$

where c_x is the centroid of U_x, $d(c_i, c_j)$ is the similarity between c_i and c_j, and σ_x denotes the average similarity of vertices in U_x to c_x. The smaller the DBI is, the better the quality of clusters will be.

Parameters. In our algorithm, the sparsity parameter λ is set to 0.15 in all experiments by cross-validation, and β_i for each modality is equally set as $\frac{1}{V}$. Parameter η that controls the error sparsity is set by the grid search in $\{0.1, \ldots, 0.9\}$. Finally, we study the parameter γ, the controller of the angular based regularization term, to examine its impact on the performance of our approach. The analysis on the impact of γ is presented in Section 5.3.

In accordance with **CCA**, the lower dimensionality is set to 40. Each cluster results from 5 runs of K-means with the lowest scores reported as the final cluster assignment. In **MultiNMF**, the parameter λ_v that tunes the relative weight among different views is empirically set to 0.1 for all views and datasets. **Co-train-SC** requires the number of clusters specified in advance to obtain eigenvectors. Thus, we set the same value as the category number in each dataset.

Noise Setting. We use Gaussian noise with the zero mean and unit variance to corrupt the data. The noise can be modified by varying magnitudes and we provide results for each magnitude of noise. We report the noise using Peak Signal-to-Noise Ratio (PSNR), which is commonly to measure the quality of image reconstruction. Given a noise-free $m \times n$ image S and its noisy approximation B, the PSNR is defined as

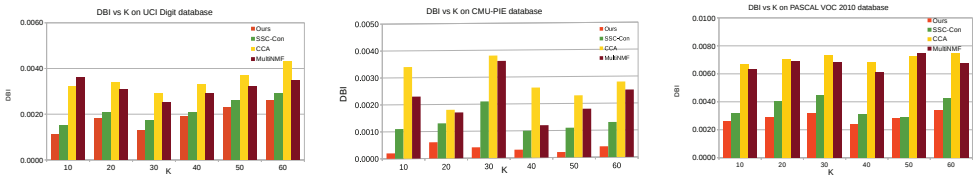
$$PSNR = 10 \log_{10} \left(\frac{b^2}{\frac{1}{mn} \sum_i^m \sum_j^n (S_{ij} - B_{ij})^2} \right), \quad s.t. \ \|S - B\|_1 = \xi, \tag{5.4}$$

where b is the maximum possible pixel value of the image S. For color images with three RGB values per pixel, the definition of PSNR is the same except the denominator $(\frac{1}{mn} \sum_i^m \sum_j^n (S_{ij} - B_{ij})^2)$ is the sum over all squared value differences divided by image size and by three. Decreasing values of PSNR means the increasing amount of noise. $\|S - B\|_1 = \xi$ is used to ensure the sparsity of noise. Note that the denominator of PSNR is 0 in a case of noise-free. The PSNR values are reported as rounded averages in our experiments.

5.3 Parameter Study

The regularization parameter γ in the angular based difference is critical to our algorithm. It manages the extent to which each modality makes consensus. A larger γ focuses on reaching agreement across views. When γ is 0, the problem reduces to do subspace clustering for each modality separately; when γ goes to infinity, sparse codes from different views share the same value. Fig.5.1 shows how the accuracy of our approach on the three datasets varies with changes in parameter γ. As we can see, our algorithm performs relatively stable on our dataset when γ is around 0.015, which is the default value in our experiments. Moreover, our method in most time still outperforms baselines when γ takes various values.

5.4 Evaluation on Clustering Performance



(a) (b) (c)

Figure 5.2: The evaluation on cluster qualities of different methods over three real-world benchmarks.

Table 5.1 reports the clustering performances of different algorithms on the three datasets. We use PSNR = 48 as the noise setting to corrupt each pixel of an image. Without other specified, PSNR = 48 is used by default. It can be observed that our

Table 5.3: The clustering accuracy (%) without/with angular based regularizer.

	UCI	CMU-PIE	PASCAL
Averaged affinity	74.286	69.365	64.287
Angular regularizer	94.255	90.267	88.092

algorithm without noise term, referred to be **SSC-Con**, outperforms the second best counterpart **MultiNMF** by a margin of 5.8%(7.8%) on UCI, 3.7%(6.6%) on CMU PIE face, and 9.1%(6.8%) on PASCAL, in terms of accuracy(NMI). One reason for this is that **SSC-Con** can automatically learn a good similarity matrix by aggregating consensus data correlations across views, rather than manually set the dimension number of reduced subspaces. Moreover, the improved gains are significantly high between **SSC-Con** and other alternatives of **CCA**, **Co-train-SC**, **ConcatSSC**, and **BSV-SSC**. Meanwhile, our approach with explicit noise model is able to achieve better clustering results than **SSC-Con** does, in terms of three benchmarks. This is mainly attributed to the noise term which helps to recover latent sparse representation shared by multiple views.

We demonstrate the stableness and robustness of our approach by comparing with **CCA**, and **MultiNMF** in terms of self-adaptive data correlation discovery. Both of the compared approaches need to manually set the dimension number of reduced subspaces, *i.e.*, number of clusters. Fig.5.2 shows the DBI comparison on three real-world databases with various K values. It can be seen that our method has the lowest DBI, while competitors have higher DBIs. This indicates that our parameter-free algorithm can discover true clusters more robustly through adaptively correlating data consensus across views. For comparison, the manually assigned values of K make **CCA** and **MultiNMF** sensitive to parameters and less effective in clustering quality.

5.5 Contribution of Angular based Regularizer

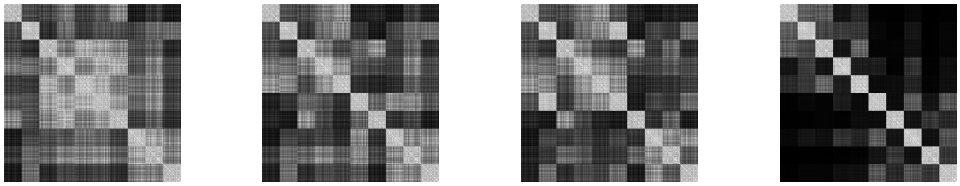
In the experiment, we study the contribution of angular based regularizer in terms of achieving consensus sparse vectors among multi-views. As a comparison, we need to calculate the averaged affinity matrix on each database by optimizing Eq.(3.1) w.r.t. individual views, and average the similarity between data objects from different views. In Table 5.3, we show the clustering accuracy over the averaged affinity matrix as well as the results obtained by imposing angular based regularizer on sparse representations. In this experiment, we keep the data uncorrupted, and report the averaged clustering accuracy values over three benchmarks. It can be seen that the angular regularization across multi-views outperforms average affinity values over individual views on the three databases. This verifies the necessity of imposing angular base regularization which is able to achieve consensus in sparse representations while each individual view only describes one aspect of visual data and their averaged similarities cannot encode their complementary properly.

5.6 Evaluation on Trace Lasso

To understand the effect of trace lasso in modeling data correlation, we evaluate the performance of the best single view with trace lasso as well as another three penalty norms: **SSC**-ℓ_1, **LRR**, and **LSR**. Table 5.4 shows the clustering results on CMU-PIE face database where there are three subspace clustering problems on the first 5, 10, and 15 subjects. We can see that the best single view with trace lasso outperforms all

Table 5.4: The clustering accuracy (%) with different penalty norms on CMU-PIE database.

Method	Accuracy(%)		
	5 subjects	10 subjects	15 subjects
SSC-ℓ_1	80.315±0.010	54.190±0.017	37.905±0.026
LRR	86.741±0.008	65.526±0.013	51.468±0.018
LSR	91.352±0.009	74.195±0.012	59.370±0.019
Ours	**94.333±0.004**	**84.720±0.010**	**73.851±0.015**



(a) **SSC**-ℓ_1 (b) **LRR** (c) **LSR** (d) **Trace lasso**

Figure 5.3: The affinity matrix derived by (a) **SSC**-ℓ_1 (b) **LRR** (c) **LSR**, and (d) best single view with trace lasso on CMU-PIE face database.

competitors on all these three clustering tasks. Both **LRR** and **LSR** perform better than **SSC**-ℓ_1, which is a result of strong grouping effect of the two methods. However, **LRR** and **LSR** lack the ability of subset selection, and thus may group some data between clusters together. By contrast, trace lasso not only preserves the grouping effect of within cluster but also encourages sparsity between clusters. To illustrate this issue, we provide an intuitive comparison of the four methods in Fig. 5.3.

5.7 Evaluation on Noise Term

In this experiment, we validate the feature of our approach that is able to recover a latent sparse representation from multi-view corrupted data objects. We conduct the validation on three databases. In CMU-PIE face database, for the computational convenience, we use 20 out of 68 subjects and randomly select 50 images from each subject to form the data collection, $X = \{X_1, \ldots, X_{20}\}$. Three views are used: LBP, HOG, and grey levels. Our purpose is to correctly segment the data into 20 clusters. After corrupting the data from CMU PIE face with various levels of Gaussian noise, we evaluate the clustering performance of our approach against competitors. Likewise, in UCI handwritten digit database and PASCAL VOC 2010, for each subject, we randomly select 50 samples to form data collection, and we aim to discover 10 and 20 clusters, respectively. Results are shown in Fig.5.4. We observe that at all levels of noise, our method outperforms every competitor by effectively recovering a latent sparse representation from corruptions.

Another interesting observation is that the gap between the proposed method and **SSC-Con** is generally maintained with different levels of noises in Fig.5.4. This is mainly because the performance of **SSC-Con** will be largely pulled down when the noise corruptions tend to uniformly distributed for all view-specific feature representations. That is, the noise distributions for all views can simultaneously reduce or increase the correlations between the same data objects. Otherwise, the sparse repre-

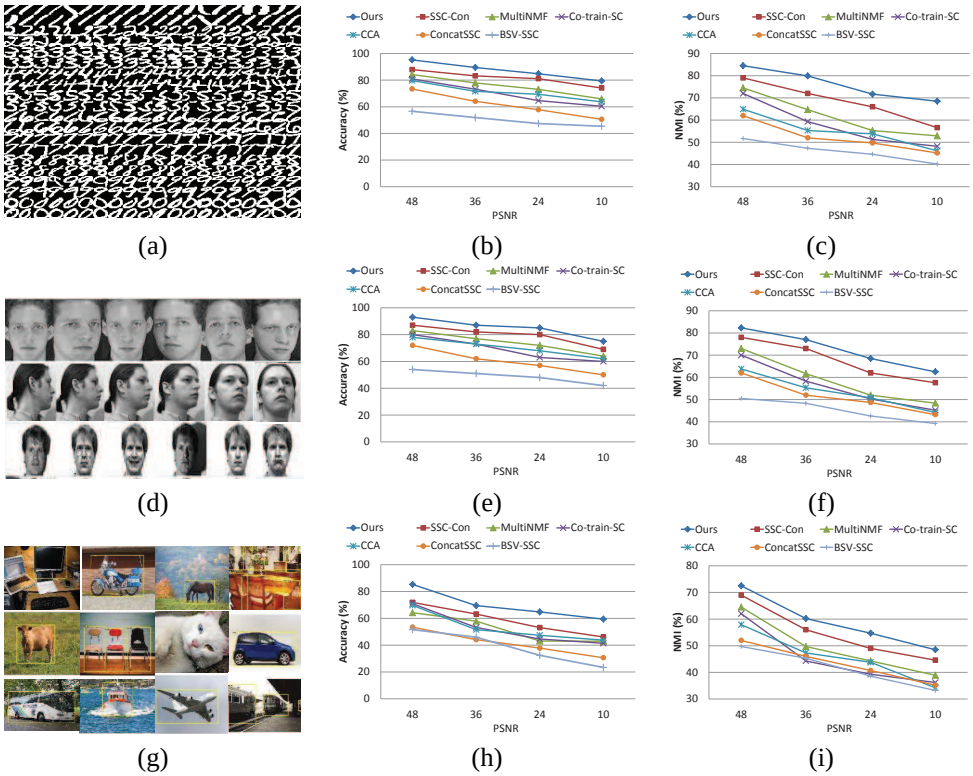


Figure 5.4: The robustness study on three real-world databases. (a)-(c) Examples of UCI handwritten digits, and performance curve w.r.t. increasing magnitudes of noises. (d)-(f) Examples of CMU PIE Face dataset, and performance curve w.r.t. increasing magnitudes of noises. (g)-(i) Examples of PASCAL-VOC 2010 dataset, and performance curve w.r.t. increasing magnitudes of noises.

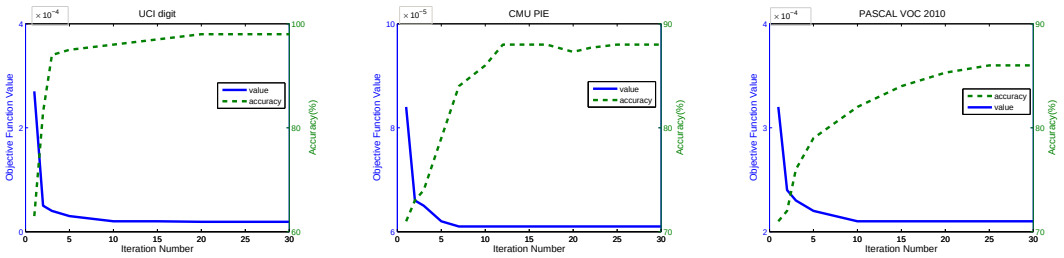
sentations for any heavily noised view may be recovered by other views via the angular regulation minimization. Such case may not be frequently met in our random noise generation for all view specific feature representation, thus, lead to maintained gap between two methods.

Nevertheless, the proposed technique in our paper can effectively tackle the noise issue by recovering the common consensus sparse representations for all views, therefore, it always outperforms **SSC-Con**.

5.8 Convergence and Multi-view Consensus Study

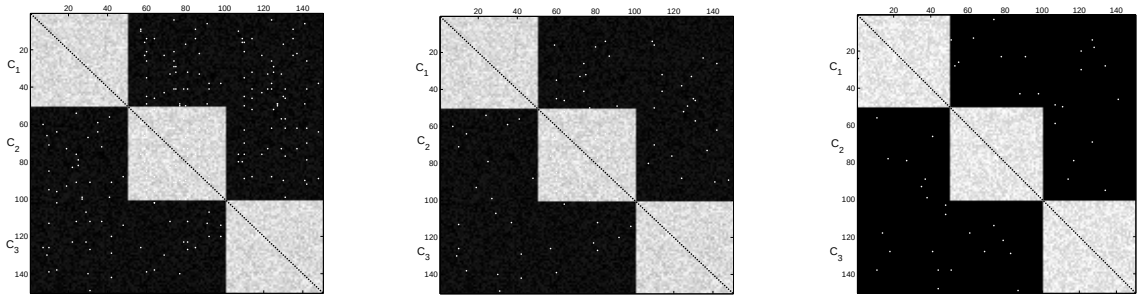
The updating rules make the minimization of the objective function in an essentially iterative way. In this section, we empirically show that the updating scheme of our method leads to the convergence. In Fig.5.5, we plot the convergence curve, together with its clustering performance on the three benchmarks. The solid line shows the value of our objective function at each iteration, while the dash line indicates the clustering accuracy accordingly. It can be seen that the algorithm always becomes convergent after around less than 20 iterations.

To show the correlation consensus propertied by angular term $\pi(s_k)$, we conduct



(a) (b) (c)

Figure 5.5: The study on the convergence and corresponding performance curve on three real-world benchmarks. (a) UCI digit. (b) CMU PIE face. (c) PASCAL VOC 2010.



(a) Iteration 10 (b) Iteration 15 (c) Iteration 20

Figure 5.6: The plot of coefficient matrices on CMU PIE face dataset. Without loss of generality, we put the data points in the same subject category together, which form a block diagonal matrix. The lighter color means the closer correlation and higher coefficients.

another experiment on PASCAL VOC 2010 where the training set X is composed of a number of randomly selected images from 50 to 450. The rest are taken as the test samples. The consensus threshold is T, whose value is discretely set from 0.73 to 0.94. We employ the consensus ratio as the evaluation metric, defined as the number of test samples whose values of Eq. (3.4) are larger or equal to T. The results are shown in Fig.5.7. It can be seen that the more number of training samples are used, the larger value of the consensus ratio is obtained. This implies that learned sparse codes are more consensus. However, the consensus ratios naturally decrease more when cosine values become higher, resulting in a more restrict consensus. Overall, our method preforms well even with a relatively small training set and large values of T.

Overall, experimental studies have demonstrated the superiority of our approach over state-of-the-art algorithms in terms of multi-modal data clustering. Moreover, our method is robust to noise corruption, insensitive to parameter settings, and fast in convergence.

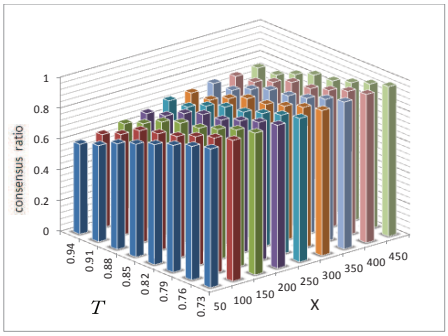


Figure 5.7: The values of consensus ratios versus the number of training data (X) and levels of thresholds (T).

6 Conclusions and Future Work

In this paper, we presented a novel approach towards subspace clustering over multi-view data. A novel angular based regularizer is proposed to achieve the data correlation consensus on multi-views. Based on that, we further propose a novel sparse decomposition based method to generate the refined data correlation consensus with the scenario that the considerable noise is available for each-view specific representations. The extensive experiments are conducted on real-world datasets to validate the effectiveness of our technique by exploiting the correlations consensus.

One future direction may consider learning the dictionary atoms for sparse representations for multi-view data, while develop novel techniques to achieve the similarity consensus based on such sparse representations for subspace clustering. Another future work is to exploit the consensus information among the cross-view data (*e.g.,* the heterogeneous data sources captured by a day camera, infrared camera and X-Ray sensors) instead of multi-view data objects for subspace clustering.

Bibliography

[1] K. Kanatani, "Motion segmentation by subspace separation and model selection," in *ICCV*, 2001.

[2] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *ICCV*, 2011, pp. 1615–1622.

[3] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, 2013.

[4] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *CVPR*, 2009.

[5] M. Tan, I. W. Tsang, and L. Wang, "Matching pursuit lasso part i: Sparse recovery over big dictionary," *IEEE Transactions on Signal Processing*, vol. 63, no. 3, 2015.

[6] M. Tan, I. Tsang, and L. Wang, "Matching pursuit lasso part ii: Applications and sparse recovery over batch signals," *IEEE Transactions on Signal Processing*, vol. 63, no. 3, 2015.

[7] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *ICML*, 2010.

[8] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least square regression," in *ECCV*, 2012.

[9] C.-Y. Lu, J. Feng, Z. Lin, and S. Yan, "Correlation adaptive subspace segmentation by trace lasso," in *ICCV*, 2013.

[10] Y. Wang, X. Lin, and Q. Zhang, "Towards metric fusion on multi-view data: a cross-view based graph random walk approach." in *ACM CIKM*, 2013.

[11] Y. Wang, X. Lin, L. Wu, and et al, "Exploiting correlations consensus: Towards subspace clustering for multi-modal data," in *ACM Multimedia*, 2014.

[12] Y. Wang, X. Lin, L. Wu, W. Zhang, and Q. Zhang, "Lbmch: Learning bridging mapping for cross-modal hashing." in *ACM SIGIR*, 2015.

[13] Y. Wang, X. Lin, L. Wu, Q. Zhang, and W. Zhang, "Shifting multi-hypergraphs via collaborative probabilistic voting," *Knowledge and Information Systems, DOI 10.1007/s10115-015-0833-8*, 2015.

[14] L. Wu, Y. Wang, and J. Shepherd, "Efficient image and tag co-ranking: a bregman divergence optimization method." in *ACM Multimedia*, 2013.

[15] Y. Wang, M. A. Cheema, X. Lin, and Q. Zhang, "Multi-manifold ranking: Using multiple features for better image retrieval." in *PAKDD*, 2013.

[16] Y. Wang, J. Pei, X. Lin, Q. Zhang, and W. Zhang, "An iterative fusion approach to graph-based semi-supervised learning from multiple views." in *PAKDD*, 2014.

[17] A. Kumar and H. Daum, "A co-training approach for multi-view spectral clustering," in *ICML*, 2011.

[18] A. Kumar, P. Rai, and H. Daum, "Co-regularized multi-view spectral clustering," in *NIPS*, 2011.

[19] K. Chaudhuri, S. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *ICML*, 2009.

[20] Z. Akata, C. Thurau, and C. Bauckhage, "Non-negative matrix factorization in multimodality data for segmentation and label prediction," in *ICCV Winter-workshop*, 2011.

[21] D. Greene and P. Cunningham, "A matrix factorization approach for integrating multiple data views," in *ECML/PKDD*, 2009.

[22] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *SDM*, 2013.

[23] E. Grave, G. Obozinski, and F. Bach, "Trace lasso: a trace norm regularization for correlated designs," in *NIPS*, 2011.

[24] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT*, 1998.

[25] H. Wang, F. Nie, H. Huang, and C. Ding, "Heterogeneous visual feature fusion via sparse multimodal machine," in *CVPR*, 2013.

[26] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *AAAI*, 2013.

[27] A. Argyriou, T.Evgeniou, and M. Pontil, "Multi-task feature learning," in *NIPS*, 2007.

[28] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," UIUC Technical Report UILU-ENG-09-2215, Tech. Rep., 2009.

[29] J. Cai, E. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[30] J. Duchi and S. Shalev-Shwartz, "Efficient projection onto the ℓ_1-ball for learning in high dimensions," in *ICML*, 2008.