COMIC: Multi-view Clustering Without Parameter Selection

Xi Peng [1] **Zhenyu Huang** [1] **Jianchen Lv** [1] **Hongyuan Zhu** [2] **Joey Tianyi Zhou** [3]

Abstract

In this paper, we study two challenges in clustering analysis, namely, how to cluster multi-view data and how to perform clustering without parameter selection on cluster size. To this end, we propose a novel objective function to project raw data into one space in which the projection embraces the geometric consistency (GC) and the cluster assignment consistency (CAC). To be specific, the GC aims to learn a connection graph from a projection space wherein the data points are connected if and only if they belong to the same cluster. The CAC aims to minimize the discrepancy of pairwise connection graphs induced from different views based on the view-consensus assumption, *i.e.*, different views could produce the same cluster assignment structure as they are different portraits of the same object. Thanks to the view-consensus derived from the connection graph, our method could achieve promising performance in learning view-specific representation and eliminating the heterogeneous gaps across different views. Furthermore, with the proposed objective, it could learn almost all parameters including the cluster number from data without labor-intensive parameter selection. Extensive experimental results show the promising performance achieved by our method on five datasets comparing with nine state-of-the-art multi-view clustering approaches.

1. Introduction

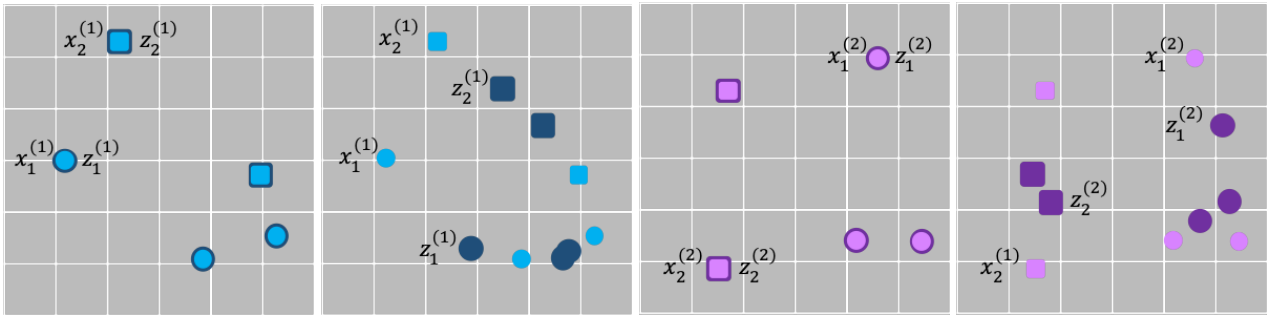
Clustering analysis aims to group unlabeled data into different clusters based on their intrinsic similarities, which is a fundamental task in machine learning. Traditional single-view clustering methods only consider the data from a single source (Hocking et al., 2011; Elhamifar & Vidal, 2013; Liu et al., 2016; Flammarion et al., 2017; Shah & Koltun, 2017; Liu et al., 2019; Peng et al., 2017a; 2018; Liu et al., 2017), which may be less attractive to some scenarios due to the heterogeneous properties in data. To be specific, most real-world data are collected from diverse domains or obtained from various feature extractors (Zhang et al., 2019; Liu & Tsang, 2017; Lu et al., 2018). Each domain or feature extractor is referred to as a particular view, thus leading to the heterogeneous properties of data. In real-world applications, the heterogeneous properties always take in variety of multi-view forms, *e.g.,* 1) text + image + voice, and 2) local binary pattern (LBP) + scale-invariant feature transform (SIFT). Kumar et al. (2011); Xu et al. (2015); Zhang et al. (2017); Yang et al. (2018) have proven that simply applying single-view clustering methods cannot narrow the heterogeneous gap to achieve desirable results because the clusters may largely differ in each of the data views. Therefore, it is highly expected to develop multi-view clustering (MvC) methods which could group similar objects into the same cluster and dissimilar objects into different clusters by utilizing the available multi-view information.

Based on the way to utilize the multi-view information, existing methods could be roughly classified into the following categories, namely, canonical correlation analysis (Chaudhuri et al., 2009), multi-view matrix factorization (Zhang et al., 2018), multi-view subspace clustering/spectral clustering (Kumar et al., 2011; Cao et al., 2015; Lu et al., 2016; Zhang et al., 2017), and deep multi-view clustering (Andrew et al., 2013; Wang et al., 2015; Zhao et al., 2017). The core commonality of these methods is encapsulating the complementary information of different views to learn a shared/common representation followed by a single-view clustering approach. Xu et al. (2013) provided a comprehensive survey and we will introduce these works in related works with more details.

Although numerous works have been conducted and achieved significant progress in multi-view clustering, most of them have suffered from parameter selection issue. In brief, almost all MvC methods have to seek an optimal combination of parameters including expected cluster number so that a desirable data partition is obtained. To seek the optimal parameters, some evaluation metrics such as normalized mutual information (NMI) are used as a performance guid-

[1]College of Computer Science, Sichuan University, Chengdu, China [2]Institute for Infocomm Research, A*STAR, Singapore [3]Institute of Performance Computing, A*STAR, Singapore. Correspondence to: J. T. Zhou <joey.tianyi.zhou@gmail.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.



(a) $t = 0$ on $\mathbf{X}^{(1)}$ (b) $t = 100$ on $\mathbf{X}^{(1)}$ (c) $t = 0$ on $\mathbf{X}^{(2)}$ (d) $t = 100$ on $\mathbf{X}^{(2)}$

Figure 1. Our basic idea. Suppose there are five data points from two views, each view consists of two clusters (rectangle and circle). Light-colored points denote the original data points $\{\mathbf{X}^{(v)}\}_{v=1}^2$ and deep-colored points denote the corresponding representation $\{\mathbf{Z}^{(v)}\}_{v=1}^2$. At the initialization period ($t = 0$), $\mathbf{X}^{(v)} = \mathbf{Z}^{(v)}$. After the model converged, the representation should incorporate the discrimination. Our idea is twofold. On one hand, a good representation could be helpful to address the linearly inseparable issue. On the other hand, the connection graph $\mathbf{S}^{(v)}$ for each single view will be more robust and better than view-specific representation $\mathbf{Z}^{(v)}$ for achieving cross-view consensus. In other words, we enforce the connection graph $\{\mathbf{S}^{(v)}\}_{v=1}^m$ to be as similar as possible. Such a cross-view consensus learning paradigm is remarkably different from existing works which usually enforce $\{\mathbf{Z}^{(v)}\}_{v=1}^m$ to be as similar as possible.

ance, which are based on the label information. In practice, it is a daunting task to obtain either of the label information and the expected cluster number, especially, in big data era.

To overcome the aforementioned challenging issue, we propose a novel multi-view clustering approach, termed as CrOss-view MatchIng Clustering (**COMIC**) which could automatically learn almost all parameters including expected cluster number in a data-driven way. In brief, COMIC projects each data point into a space in which two properties are satisfied, *i.e.*, geometric consistency (GC) and cluster assignment consistency (CAC) which are specifically designed for different goals. More specifically, GC aims to learn a normalized connection graph $\mathbf{S}^{(v)}$ for the v-th view in a learned projection space with the help of local geometrical consistency $\mathbf{W}^{(v)}$. Note that, GC does not learn a compact representation for each data point like existing works did. In contrast, the representation $\mathbf{Z}^{(v)}$ is learned from the ambient space and theoretically $\mathbf{Z}^{(v)}$ will collapse to a small number of landmarks, thus leading to interpretable results. In brief, as the learned representation are with the same dimensionality of the input space, our method enjoys the interpretable data partition and representation. Different from GC, CAC is proposed to handle multi-view data by minimizing the discrepancy of the connection graphs $\{\mathbf{S}^{(v)}\}_{v=1}^m$. In other words, different from most of traditional approaches, our method adopts an end-to-end pipeline to explicitly optimize representations and their relationship which is formulated as a connection graph. By enforcing view consensus on the connection graph instead of the learned representations, our method embraces the following advantages. As shown in Fig. 1, the value of the learning representation $\mathbf{Z}^{(v)}$ may remarkably differ in data views even though a well-established representation learning algorithm is employed. If enforcing the view-specific representation as similar as possible, the optimization may be distorted and useful information probably be lost, thus giving inferior clustering performance. In contrast, by using the connection relationship among $\mathbf{Z}^{(v)}$ as an invariance and enforcing them as close as possible, our method could largely avoid distorting representations, thus boosting data clustering. Such an idea is easily understood. If two objects belong to the same class, their connection relationship will be invariant to different views and different projection spaces.

2. Cross-view Matching Clustering Without Parameter Selection

For a given dataset $\mathbf{X}^{(v)} = [\mathbf{x}_1^{(v)}, \mathbf{x}_2^{(v)}, \cdots, \mathbf{x}_n^{(v)}] \in \mathcal{R}^{D^{(v)} \times n^{(v)}}$, let $\mathbf{X}^{(v)}$ denote the dataset sampled from the v-th view and there are m different views, the proposed objective function is as below:

$$\mathcal{L} = \sum_v \mathcal{L}_1^{(v)} + \mathcal{L}_2, \tag{1}$$

where $\mathcal{L}_1^{(v)}$ and $\mathcal{L}_2$ measure the loss in each single view and cross-views, respectively. To be exact,

$$\mathcal{L}_1^{(v)} = \frac{1}{2}\sum_{i=1}^{n} \underbrace{\|\mathbf{x}_i^{(v)} - \mathbf{z}_i^{(v)}\|_2^2}_{\text{reconstruction loss}} + \underbrace{\frac{\lambda^{(v)}}{2}\sum_{i,j}\mathbf{W}_{ij}^{(v)}\left(\|\mathbf{S}_{ij}^{(v)}\mathbf{z}_i^{(v)} - \mathbf{S}_{ji}^{(v)}\mathbf{z}_j^{(v)}\|_2^2\right) + \mu^{(v)}(\mathbf{S}_{ij}^{(v)} - 1)^2}_{\text{geometric consistency}} \tag{2}$$

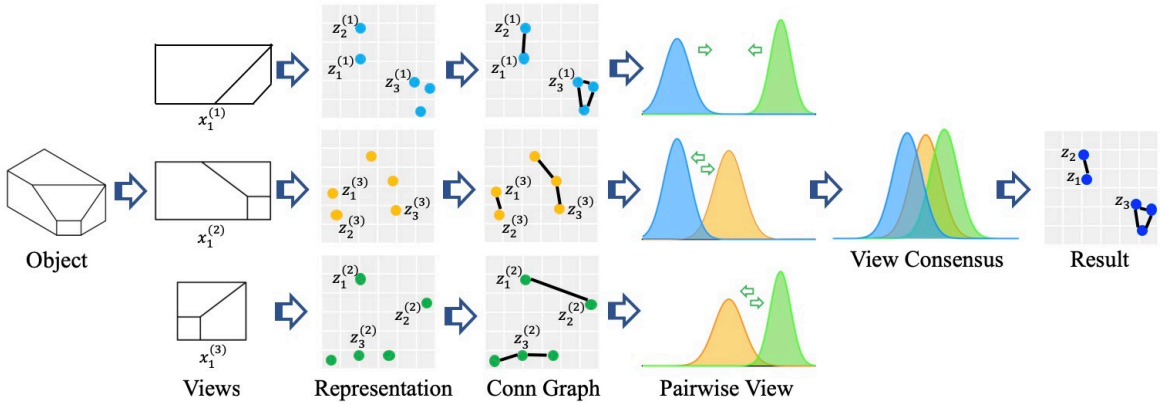


Figure 2. An illustration to the proposed COMIC.

and

$$\mathcal{L}_2 = \frac{1}{2}\sum_{i,j}\sum_{v\neq k}\underbrace{(\mathbf{S}_{ij}^{(v)} - \mathbf{S}_{ij}^{(k)})^2}_{\text{cluster assignment consistency}} \tag{3}$$

where $\mathbf{z}_i^{(v)} \in \mathcal{R}^D$ is the learned representation of $\mathbf{x}_i^{(v)}$, which aims to keep two properties, namely, geometric consistency within a single view and cluster assignment consistency across different views. $\mathbf{W}^{(v)}$ is a precomputed similarity graph to achieve the geometric consistency. In this paper, we employ mutual k-nearest neighbors connectivity (m-kNN) to compute $\mathbf{W}^{(v)}$ and will elaborate it later. The symmetric matrix $\mathbf{S}^{(v)}$ is the learned connection graph of which the connected data points are regarded as belonging to the same cluster. $\lambda^{(v)}$ and $\mu^{(v)}$ are penalty parameters to balance these terms, which are automatically learned from data as presented in the following section.

The terms $\mathcal{L}_1^{(v)}$ and $\mathcal{L}_2$ are designed for different goals. Intuitively, $\mathcal{L}_1^{(v)}$ aims to learn $\mathbf{Z}^{(v)}$ and $\mathbf{S}^{(v)}$ for each single view by embracing the geometric consistency on the manifold. In contrast, $\mathcal{L}_2$ aims to minimize the discrepancy of the connection graphs $\{\mathbf{S}^{(v)}\}_{v=1}^m$ since different views should generate the same connection components. More specifically, $\mathcal{L}_1^{(v)}$ consists of the reconstruction loss and the GC constraint. The reconstruction loss performs like the recent convex clustering (Hocking et al., 2011; Chen et al., 2015; Flammarion et al., 2017; Shah & Koltun, 2017) which learns $\mathbf{Z}^{(v)}$ for $\mathbf{X}^{(v)}$ in the ambient space. The motivation behind of such an idea is that all within-cluster data points are highly encouraged to collapse to the set of a small number of landmarks. To facilitate clustering, we propose the GC constraint to learn the connection graph $\mathbf{S}^{(v)}$ and simultaneously enforce $\mathbf{Z}^{(v)}$ lying onto a manifold charactered by $\mathbf{W}^{(v)}$. Note that, the term $\mathbf{S}_{ij}^{(v)} - 1$ plays three roles. First, it will ignore the connection (i, j) that tends to zero when the connection is established ($\mathbf{S}_{ij}^{(v)} \to 1$) and be a penalty of one when the connection is disestablished ($\mathbf{S}_{ij}^{(v)} \to 0$). Second, the weight of the connection graph is constrained into the range of $[0, 1]$ to avoid the scale variance causing by different views. Third, it could avoid the trivial solutions such as $\mathbf{S}^{(v)} = \mathbf{0}$ and $\mathbf{Z}^{(v)} = \mathbf{X}^{(v)}$.

2.1. Optimization

To optimize $\mathbf{Z}^{(v)}$ and $\mathbf{S}^{(v)}$, we adopt the alternating minimization strategy. To be specific, when $\mathbf{Z}^{(v)}$ is fixed, we compute the derivative of Eq.1 w.r.t. $\mathbf{S}_{ij}^{(v)}$ as below:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{S}_{ij}^{(v)}} =& \lambda^{(v)}\mathbf{W}_{ij}^{(v)}\mathbf{S}_{ij}^{(v)}\|\mathbf{z}_i^{(v)} - \mathbf{z}_j^{(v)}\|_2^2 + \mu^{(v)}(\mathbf{S}_{ij}^{(v)} - 1) \\ &+ \Big((m-1)\mathbf{S}_{ij}^{(v)} - \sum_{k\neq v}\mathbf{S}_{ij}^{(k)}\Big)\end{aligned} \tag{4}$$

Let Eq.4 be zero, then we update $\mathbf{S}^{(v)}$ by

$$\mathbf{S}_{ij}^{(v)} = \frac{\mu^{(v)} + \sum_{k\neq v}\mathbf{S}_{ij}^{(k)}}{\mu^{(v)} + (m-1) + \lambda^{(v)}\mathbf{W}_{ij}^{(v)}\|\mathbf{z}_i^{(v)} - \mathbf{z}_j^{(v)}\|_2^2}. \tag{5}$$

Since $\sum_{k\neq v}\mathbf{S}_{ij}^{(v)} = m - 1$, one could see that if the data points $\mathbf{x}_i^{(v)}$ and $\mathbf{x}_j^{(v)}$ are sufficiently close (*i.e.*, belonging to the same cluster), then $\mathbf{S}_{ij}^{(v)} \to 1$ from Eq.5; Otherwise, $\mathbf{S}_{ij}^{(v)} \to 0$. When $\mathbf{S}^{(v)}$ is fixed, we drop the terms including $\mathbf{S}^{(v)}$ from Eq.3 and rewrite it as below:

$$\begin{aligned}\text{argmin}&\frac{1}{2}\|\mathbf{X}^{(v)} - \mathbf{Z}^{(v)}\|_F^2 \\ &+ \frac{\lambda^{(v)}}{2}\sum_{i,j}\mathbf{W}_{ij}^{(v)}(\mathbf{S}_{ij}^{(v)})^2\|\mathbf{Z}^{(v)}(\mathbf{e}_i^{(v)} - \mathbf{e}_j^{(v)})\|_2^2,\end{aligned} \tag{6}$$

where $\mathbf{e}_i$ is an indicator vector with the i-th entry of 1. This problem can be efficiently solved by

$$\mathbf{Z}^{(v)}\mathbf{M}^{(v)} = \mathbf{X}^{(v)}, \tag{7}$$

where

$$\mathbf{M}^{(v)} = \mathbf{I}^{(v)} + \lambda^{(v)}\mathbf{\Omega}^{(v)}, \tag{8}$$

$\mathbf{\Omega}^{(v)} = \sum_{i,j} \mathbf{W}_{ij}^{(v)}(\mathbf{S}_{ij}^{(v)})^2(\mathbf{e}_i^{(v)} - \mathbf{e}_j^{(v)})(\mathbf{e}_i^{(v)} - \mathbf{e}_j^{(v)})^\top$ and $\mathbf{I}^{(v)}$ is an identity matrix. To efficiently solve the above problem, we need the following theorem.

Theorem 1. $\mathbf{M}^{(v)}$ *defined in Eq.8 is a symmetric and positive semidefinite matrix.*

Proof. As $(\mathbf{e}_i^{(v)} - \mathbf{e}_j^{(v)})(\mathbf{e}_i^{(v)} - \mathbf{e}_j^{(v)})^\top$ is a Laplacian matrix, either of $\mathbf{\Omega}^{(v)}$ and $\mathbf{M}^{(v)}$ is a Laplacian matrix since $\mathbf{W}_{ij}^{(v)} \geq 0$, $\mathbf{S}_{ij}^{(v)} \geq 0$, and the sum of Laplacian matrices is also a Laplacian matrix. In consequence, $\mathbf{M}^{(v)}$ is symmetric and positive semidefinite. Furthermore, one could obtain that $\mathbf{M}^{(v)}$ is symmetric diagonally dominant, *i.e.*, $\mathbf{M}_{ii}^{(v)} \geq \sum_{i \neq j} |\mathbf{M}_{ij}^{(v)}|$. □

With Theorem 1, each row of $\mathbf{Z}^{(v)}$ can be solved independently and in parallel with any one multi-grid solver. In other words, the computational complexity could be reduced from $O(\sum_{v=1}^{m}(n^{(v)})^3)$ to $O(\sum_{v=1}^{m}(\hat{n}^{(v)}\log\hat{n}^{(v)})$, where $\hat{n}^{(v)}$ denotes the number of nonzero entries of the vector of $\mathbf{M}^{(v)}$.

2.2. Initialization and Implementation Details

$\mathbf{Z}^{(v)}$ and $\mathbf{S}^{(v)}$ are initialized by $\mathbf{X}^{(v)}$ and $\mathbf{1}$, respectively. After the model converged, we could obtain the final cluster graph via the following approach. First, we build m view-specific connection graphs in which $\mathbf{z}_i^{(v)}$ and $\mathbf{z}_j^{(v)}$ are connected *iif* $\|\mathbf{z}_i^{(v)} - \mathbf{z}_j^{(v)}\|_2 \leq \epsilon^{(v)}$, where $\epsilon^{(v)}$ is set to the mean length of the shortest 90% edges in $\mathbf{W}^{(v)}$. After that, we obtain the final connection graph by employing the voting strategy. To be specific, any two instances belong to the same cluster *iif* they mutually connect into a half of view-specific connection graphs at least. Note that, although the learned $\mathbf{S}^{(v)}$ could also be directly emerged into the final connection graph, we experimentally found that performance is slightly improved by thresholding the trivial connections as mentioned above.

2.3. Data-driven Parameter Selection

The proposed algorithm includes five parameters, namely, $\lambda^{(v)}$, $\mu^{(v)}$, the connectivity of $\mathbf{W}^{(v)}$, and the threshold parameter $\epsilon^{(v)}$ and δ. All these parameters are automatically learned from data. To be exact, as Eq.7 is a linear least-squares problem, $\lambda^{(v)}$ balances the reconstruction loss and the geometric consistency loss. According to (Shah & Koltun, 2017), we automatically update it as below:

$$\lambda^{(v)} = \frac{\|\mathbf{X}^{(v)}\|_2}{\|\mathbf{\Omega}^{(v)}\|_2}, \tag{9}$$

where $\mathbf{\Omega}^{(v)} = \mathbf{W}_{ij}^{(v)}(\mathbf{S}_{ij}^{(v)})^2(\mathbf{e}_i^{(v)} - \mathbf{e}_j^{(v)})(\mathbf{e}_i^{(v)} - \mathbf{e}_j^{(v)})^\top$, and $\|\cdot\|_2$ denotes the spectral norm of a matrix. Clearly, the update of $\lambda^{(v)}$ only depends on the spectral norm of $\mathbf{X}^{(v)}$ and $\mathbf{\Omega}^{(v)}$. As $\|\mathbf{X}^{(v)}\|_2$ could be computed at the initial stage and reused during training, we only need to compute the largest eigenvalue of $\mathbf{\Omega}^{(v)}$ for each update.

Regarding to $\mu^{(v)}$, we set it to $\mu^{(v)} = (r^{(v)})^2$, where $r^{(v)}$ is the maximal edge length in $\mathbf{W}^{(v)}$. Furthermore, $\epsilon^{(v)}$ is set to be the mean length of the shortest 90% of the edges in $\mathbf{W}^{(v)}$ and δ is the mean of $\{\epsilon^{(v)}\}_{v=1}^{m}$.

Regarding to the precomputed similarity graph $\mathbf{W}^{(v)}$, we set the neighbor size to 10 and adopt the cosine distance as the measurement. In the graph, two nodes are connected *iif* each falls into the neighborhood of the other. To achieve robustness to different views, we set

$$\mathbf{W}_{ij}^{(v)} = \frac{\sum_{k=1}^{n^{(v)}} n_k^{(v)}}{n^{(v)}\sqrt{n_i^{(v)} n_j^{(v)}}}, \tag{10}$$

where $n_i^{(v)}$ is the number of edges connected to $\mathbf{x}_i^{(v)}$ in the graph.

Besides the aforementioned five parameters, our method is capable of automatically determining the cluster number, which is remarkably different from the popular approaches. In fact, to the best of our knowledge, there are few multi-view clustering methods which could escapes from such a parameter selection trap. Benefiting from the cross-view assignment loss $\mathcal{L}_2$, the proposed method directly treats each connected component as a cluster, thus avoiding to specify the value for the parameter and improving the availability in practice.

3. Related Works

In this section, we briefly introduce convex clustering and multi-view clustering, as well as the relationship between our method and them.

3.1. Convex Clustering

Convex clustering (Hocking et al., 2011; Flammarion et al., 2017) projects the data point into another space by incorporating some convex pairwise fusion penalties. Motivated by the success of convex clustering, Flammarion et al. (2017); Chen et al. (2015); Shah & Koltun (2017) have proposed a variety of penalties which result in regularization paths useful for clustering.

Algorithm 1 Cross-view Matching Clustering

Input: A given dataset $\{\mathbf{X}^{(v)}\}_{v=1}^{m}$ from m different views.
1. Normalize each row of $\{\mathbf{X}^{(v)}\}_{v=1}^{m}$ to have a unit of ℓ_2-norm to avoid scale difference caused by different views.
2. Build the similarity graph $\{\mathbf{W}^{(v)}\}_{v=1}^{m}$ via Eq.10 and compute the spectral norm of $\{\mathbf{X}^{(v)}\}_{v=1}^{m}$.
3. Initialize $t = 1$, $\mathbf{Z}^{(v)} = \mathbf{X}^{(v)}$ and $\mathbf{S}^{(v)} = \mathbf{1}$.
4. Initialize $\lambda^{(v)}$, $\mu^{(v)}$, $\epsilon^{(v)}$, δ as indicated in Section 2.3.
while $|\mathcal{L}^{t+1} - \mathcal{L}^{t}| \leq 10^{-8}$ or $t \leq 1000$ **do**
  Update $\mathbf{S}^{(v)}$ using Eq.5.
  Update $\mathbf{Z}^{(v)}$ using Eq.7.
  Update $\lambda^{(v)}$ as indicated in Section 2.3.
  Update t by $t + 1$.
end while
output Obtain the clustering result as elaborated in Section 2.2.

The major differences between existing convex clustering approaches and our COMIC are in three-fold. First, our method does not suffer from the parameter selection issue faced by most of existing convex clustering methods. Second, these methods can only handle singe-view data, whereas COMIC could utilize the multi-view information. Third, they usually enforces some convex penalties such as ℓ_1-norm and the computational complexity is proportional to the cubic of the dataset. In contrast, our objective function could be analytically solved, which is more efficient as analyzed in Section 2.1.

3.2. Multi-view Clustering

The core and commonality of most existing MvC approaches is utilizing the multi-view information to learn representation and applying a single-view clustering approach on the representation. As one of most effective learning paradigms, canonical correlation analysis explores the relationship between two views by finding their linear combinations and maximally correlating them, which has been investigated into shallow (Chaudhuri et al., 2009; Wang & Livescu, 2016) and deep MvC models (Andrew et al., 2013). Multi-view matrix factorization (Zhang et al., 2018) usually decomposes each view into two low rank matrices with some specific constraints and applies k-means to obtain data partitions. Multi-view subspace clustering (Kumar et al., 2011; Cao et al., 2015; Lu et al., 2016; Zhang et al., 2017) utilizes the local/global consistency to learn representation under the framework of manifold learning. In recent, Andrew et al. (2013); Wang et al. (2015); Zhao et al. (2017); Peng et al. (2017b) have devoted to neural network based MvC by deeply learning a shared representation across different views. Moreover, graph weighting (Nie et al., 2017) and binary coding (Zhang et al., 2018) have recently been used for MvC and achieved state-of-the-art performance.

The differences between the aforementioned approaches and this work are in two-fold. On one hand, most of them need to know the number of clusters in advance and have to seek a set of optimal parameters for a better performance. In contrast, our method does not suffer from such a parameter selection issue. On the other hand, these methods usually learn a shared representation for different views and employ a single-view clustering method to achieve data clustering. In other words, they treat representation learning and clustering as two separate steps, which may result in inferior performance. In contrast, our method is a clustering-oriented method, which jointly learn the representation and connection graph. Such an end-to-end pipeline narrows the gap between representation learning and clustering analysis, thus benefitting data partition.

4. Experiments

We carry out experiments on five widely-used multi-view datasets comparing with nine state-of-the-art MvC approaches in terms of two performance evaluation metrics. All the experiments are implemented using MATLAB 2016a/Python 2.7 on a standard Linux Server with an Intel Xeon 2.10 GHz CPU and 32 GB RAM.

4.1. Experimental Setting

We compare our methods with nine MvC approaches, namely, the vanilla *k*-means, the normalized spectral clustering (SC) (Ng et al., 2002), low rank representation (LRR) (Liu et al., 2016), diversity-induced multi-view subspace clustering (DiMSC) (Cao et al., 2015), latent multi-view subspace clustering (Zhang et al., 2017), deep canonical correlation analysis (DCCA) (Andrew et al., 2013), self-weighted multi-view clustering (SwMVC) (Nie et al., 2017), deep canonically correlated autoencoders (DCCA) (Wang et al., 2015), and binary multi-view clustering (BMVC) (Zhang et al., 2018). We use the sklearn code of *k*-means and SC, and implement our COMIC in python. Regarding to other seven tested method, we use the code released by the corresponding authors.

As *k*-means, SC, and LRR are single-view clustering methods, we concatenate all views into a single view and then apply these methods to perform clustering by following the setting in (Zhang et al., 2017). In other words, these three methods transform multi-view datasets into single-view ones. For all the above mentioned methods, we tune their parameters to seek an optimal performance by following the setting in the original works. In brief, we tune the kernel width for SC with the parameter range of $(0.001, 0.01, 0.1)$. For LRR, the value of λ ranges from 0.1 to 6.0 with an interval of

Table 1. Performance comparison with state of the arts in terms of **the NMI score**. The number in bold indicates the best result. The mean score denotes the mean NMI across different datasets.

Method	Caltech101	LandUse-21	Scene-15	Still-DB	mean score
k-means	35.84	28.85	30.86	11.21	26.69
SC (NIPS'02)	37.62	33.52	26.12	11.36	27.16
LRR (TPAMI'13)	14.62	33.70	34.48	7.81	22.65
DCCA (ICML'13)	46.48	26.92	40.21	11.87	31.37
DCCAE (ICML'15)	45.56	26.97	39.90	10.71	30.79
DiMSC (CVPR'15)	29.05	16.65	14.95	13.81	18.62
LMSC (CVPR'17)	63.55	34.01	38.30	14.27	37.53
SwMC (IJCAI'17)	55.92	31.40	34.46	7.79	32.39
BMVC (TPAMI'18)	64.24	28.69	35.55	5.82	33.58
COMIC	**69.25**	**43.29**	**46.59**	**26.03**	**46.29**

Table 2. Performance comparison with state of the arts in terms of **the v-measure score**. The number in bold indicates the best result. The mean score denotes the mean v-measure across different datasets.

Method	Caltech101	LandUse-21	Scene-15	Still-DB	mean score
k-means	35.67	28.82	30.86	11.21	26.64
SC (NIPS'02)	37.50	33.49	26.12	11.36	27.12
LRR (TPAMI'13)	12.26	33.53	34.43	6.67	21.72
DCCA (ICML'13)	46.33	26.62	40.18	11.80	31.23
DCCAE (ICML'15)	45.35	26.60	39.87	10.68	30.63
DiMSC (CVPR'15)	28.87	16.65	14.95	13.81	18.57
LMSC (CVPR'17)	63.26	34.01	38.30	14.27	37.46
SwMC (IJCAI'17)	55.71	29.35	31.39	7.72	31.04
BMVC (TPAMI'18)	64.06	28.62	35.54	5.82	33.51
COMIC	**68.60**	**38.89**	**48.54**	**23.02**	**44.76**

0.5. Regarding to DCCA and DCCAE, we adopt the recommended network structure and parameters. For DiMSC, we seek the optimal λ_s from $(0.001, 0.01, 0.1)$ and the optimal λ_v from $(1, 10, 100)$. For BMVC, we fix the length of code to 128 and refer to the recommended parameter setting. For LMSC, we fix the latent representation dimension to 100 and seek the optimal λ from $(0.01, 0.1, 1, 10, 100)$ as suggested. It should be pointed out that, besides the specific parameters, these baselines need knowing the number of clusters in advance, whereas COMIC does not suffer from the parameter selection issue.

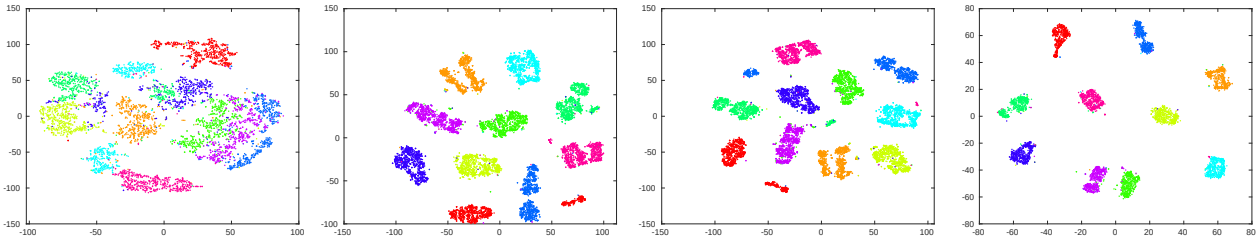
We conduct experiments using five popular datasets, namely, Caltech101, Scene-15, LandUse-21, Still-DB, and MNIST-USPS. To be specific,

- Caltech101 (Li et al., 2015) contains 2,386 images sampled from 20 classes, which is used to construct a multi-view dataset by following the setting in (Zhang et al., 2018). In details, six different features are extracted as views, including 48-dim Gabor feature, 40-dim wavelet moments (WM), 254-dim CENTRIST feature, 1,984-dim HOG feature, 512-dim GIST feature, and 928-dim LBP feature.
- The used Scene-15 multi-view dataset (Zhang et al., 2017) consists of 4485 images distributed over 15 indoor and outdoor scene categories. Three image features are used as views, *i.e.*, GIST, PHOG, and LBP.
- The used LandUse-21 dataset (Zhang et al., 2017) consists of satellite images from 21 categories each of which is with 100 images. The features used are same to Scene-15.
- The used Still-DB multi-view dataset (Zhang et al., 2017) consists of 467 images with six classes of actions. The views consists of three features, *i.e.*, Sift Bow, Color Sift Bow and Shape context Bow.
- For the MNIST-USPS dataset, we treat USPS and MNIST as two different views and randomly select 5,000 samples distributed over 10 digits from each view. The MNIST image is with 784 dimension and the USPS image is with 256 dimension.

We adopt two metrics to evaluate the clustering performance, *i.e.*, Normalized Mutual Information (NMI) and v-measure.

Table 3. Generalization of the COMIC representation across clustering algorithms using the **MNIST-USPS** dataset. RAW, AE, and COMIC denote passing raw data, encoder feature, and COMIC feature through these methods, respectively. The number in bold indicates the best result.

Method	ACC			NMI			ARI			v-measure		
	RAW	AE	COMIC	RAW	AE	COMIC	RAW	AE	COMIC	RAW	AE	COMIC
k-means	47.52	97.26	**98.88**	46.35	93.30	**96.92**	31.09	94.02	**97.53**	46.35	93.30	**96.92**
SC	48.98	94.74	**97.44**	44.28	88.52	**94.83**	29.27	88.80	**94.61**	44.28	88.52	**94.83**
LRR	73.36	97.26	**97.76**	71.59	93.61	**94.55**	64.74	94.04	**95.09**	71.59	93.61	**94.55**
DCCA	97.42	**99.80**	99.24	93.60	**99.39**	97.81	94.35	**99.55**	98.31	93.60	**99.39**	97.81
DCCAE	98.00	**99.82**	99.30	94.70	**99.46**	97.91	95.60	**99.60**	98.45	94.70	**99.46**	97.91
DiMSC	48.34	55.58	**60.12**	36.02	42.34	**52.23**	22.03	27.45	**32.14**	36.01	42.33	**52.21**
LMSC	78.60	96.94	**97.08**	78.49	92.64	**92.97**	70.78	93.36	**93.63**	78.49	92.64	**92.72**
SwMC	99.56	99.36	**99.74**	98.71	98.11	**99.18**	99.02	98.58	**99.42**	98.71	98.11	**99.18**
BMVC	77.60	**84.70**	71.50	78.75	**90.53**	73.36	71.17	**83.56**	59.45	78.74	**90.52**	73.35



(a) $t = 0$ (b) $t = 10$ (c) $t = 100$ (d) $t = 170$ (Converged)

Figure 3. t-sne visualization on the MNIST-USPS dataset with increasing training iteration.

Higher value of these metrics indicates better clustering performance. Note that, we do not employ Accuracy or called Purity score since the metric needs to utilize the ground truth of cluster size, which is inconsistent with our setting since COMIC is a parameter selection free clustering method.

4.2. Comparison with State of The Arts

In this section, we investigate the performance of our method comparing with nine clustering algorithms including three single-view methods (*i.e.*, *k*-means, SC, and LRR), four shallow multi-view clustering approaches (*i.e.*, DiMSC, LMSC, SwMC, and BMVC), and two deep multi-view clustering networks (*i.e.*, DCCA and DCCAE). For ease of comparison, we also compute the mean score in NMI and v-measure across different datasets. The results are reported in Table 1–2 which demonstrates the following observations:

- On the used four datasets, the proposed COMIC remarkably outperforms the other methods by a considerable margin. In terms of NMI, COMIC is at least 5.01%, 9.28%, 6.38%, and 11.76% higher than the second best method. In terms of v-measure, the performance gain over the second best approach is 4.54%, 4.88%, 8.36%, and 8.75%.
- Interestingly, the second best methods are shallow models in most cases. To be specific, BMVC, LMSC, and LMSC achieve the second highest NMI and v-measure score on Caltech101, LandUse-21, and Still-DB, respectively. DCCA performs the second best on Scene-15. The possible reason may be that deep learning based methods always need a large scale dataset whereas Caltech101, LandUse-21, and Still-DB are with relatively small size. Note that, our method is capable of scaling to a larger dataset. However, we do not conduct such an evaluation due to over-high computational cost of LRR, DiMSC, and LMSC.

4.3. Generalization Across Clustering Algorithms

In this section, we evaluate if the representations learned (*i.e.*, $\mathbf{Z}^{(v)}$) by our COMIC generalize to other MvC approaches, as well as whether our method could be beneficial from deep learning. To the end, we conduct experiments on the MNIST-USPS dataset by performing the following three tests. To be specific, the first test is directly feeding the raw data into the tested methods. The second test is passing the features output by a denoising auto-encoder (AE) through the methods. The used denoising auto-encoder is

with the structure of $\hat{m} - 500 - 500 - 2000 - 10 - 2000 - 500 - 500 - \hat{m}$, where $\hat{m}$ denotes the input dimension and the dropout rate is fixed to 0.2. The third test is feeding the above deep features into COMIC and further using the learned representation $\mathbf{Z}^{(v)}$ as features to the tested MvC approaches. In this experiment, as the tested MvC methods need specifying the cluster size, we adopt new metrics to evaluate their performance, namely, Accuracy (ACC) and Adjusted Rand Index (ARI).

The results are shown in Table 3. One could observe that:

- COMIC remarkably improves the clustering performance of *k*-means, SC, LRR, DiMSC, and LMSC. Comparing with the AE features, COMIC improves 3.62% on k-means, 6.31% on SC, 0.94% on LRR, 9.89% on DiMSC, 0.33% on LMSC, and 1.07% on SwMC in terms of NMI.
- Regarding to the other three MvC methods, the COMIC feature slightly degrades their performance due to different reasons. To be specific, DCCA and DCCAE are two deep models and the COMIC feature is also learned deeply, therefore passing the COMIC feature through DCCA and DCCAE means passing raw data into an eight layered encoder network which probably performs bad due to limited data size. Different from the other evaluated algorithms, the learned representation by BMVC is binary, which may be major reason for the performance degrade of using the COMIC feature.

4.4. Visualization Analysis

In this section, we conduct analysis on our method by visualizing the representation and the connection graph learned from the MNIST-USPS dataset with the aforementioned setting.

To visually illustrate learned representation, we concatenate the learned representations from the MNIST and USPS view together and then employ t-sne (Maaten & Hinton, 2008) to reduce the dimensionality to two. As shown in Fig. 3, the learned representation became more compact and discriminative with increasing t.

4.5. Convergence Analysis

In this section, we investigate the convergence analysis of our method by reporting the loss value and the corresponding NMI score with increasing iteration. In this evaluation, we use the Caltech101 dataset and report the result in Fig. 4. One could observe that, at the first several iterations, the loss value remarkably increases and then continuously decrease before $t = 35$. The reason for the increasing loss is that $\mathbf{Z}^{(v)}$ is initialized by $\mathbf{X}^{(v)}$. Regarding to the NMI score, it increases to 75.30% at $t = 17$ and then decreases to 70.0% after $t = 24$. The result tells us that if we exhaustively tune the parameters, COMIC would achieve a better performance than the above reported. In summary, our method achieves convergence quickly, which takes about two seconds for each iteration to clustering the dataset.

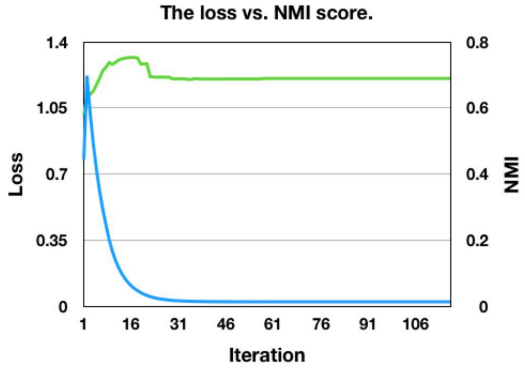


Figure 4. Convergence analysis on the proposed COMIC using the Caltech101 dataset. The left and right y-axis denote the loss value and the corresponding NMI score.

5. Conclusions

The proposed COMIC algorithm could handle two challenging issues in practical applications, namely, clustering multi-view data and clustering without prior of cluster size. One major difference between existing MvC methods and COMIC is that the latter achieves cross-view consensus on view-specific connection graph instead of view-specific representation. Extensive experiments verify the effectiveness of such a learning paradigm. In future, to further facilitate the performance, we plan to investigate the supervised and deep extension of COMIC to utilize the available label and deep neural networks.

Acknowledgment

The authors would like to thank the anonymous reviewers and area chair for their valuable comments and constructive suggestions to improve the quality of this paper. The work was supported in part by the Fundamental Research Funds for the Central Universities under Grant YJ201949 and 2018SCUH0070, in part by the NFSC under Grant 61806135, 61625204, and 61836006, and in part by the grant A1687b0033 and A18A1b0045 from the Singapore government's Research, Innovation and Enterprise 2020 plan (Advanced Manufacturing and Engineering domain).

References

Andrew, G., Arora, R., Bilmes, J., and Livescu, K. Deep canonical correlation analysis. In *Proc Int Conf Mach*

Learn, pp. 1247–1255, 2013.

Cao, X., Zhang, C., Fu, H., Liu, S., and Zhang, H. Diversity-induced multi-view subspace clustering. In *Proc IEEE Conf Comput Vis Pattern Recognit*, pp. 586–594, 2015.

Chaudhuri, K., Kakade, S. M., Livescu, K., and Sridharan, K. Multi-view clustering via canonical correlation analysis. In *Proc Int Conf Mach Learn*, pp. 129–136, New York, New York, USA, Jun 2009. ACM.

Chen, G. K., Chi, E. C., Ranola, J. M. O., and Lange, K. Convex Clustering - An Attractive Alternative to Hierarchical Clustering. *PLoS Computational Biology*, 11(5): e1004228, 2015.

Elhamifar, E. and Vidal, R. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans Pattern Anal Mach Intell*, 35(11):2765–2781, September 2013.

Flammarion, N., Palanisamy, B., and Bach, F. R. Robust Discriminative Clustering with Sparse Regularizers. *J Mach Learn Res*, 18(1):1–50, Aug 2017.

Hocking, T., Vert, J.-P., Bach, F. R., and Joulin, A. Clusterpath - An Algorithm for Clustering using Convex Fusion Penalties. In *Proc Int Conf Mach Learn*, pp. 745–752, Washington, USA, Jul 2011.

Kumar, A., Rai, P., and Daume, H. Co-regularized multi-view spectral clustering. In *Proc Adv Neural Inf Process Syst*, pp. 1413–1421, 2011.

Li, Y., Nie, F., Huang, H., and Huang, J. Large-scale multi-view spectral clustering via bipartite graph. In *Proc Conf AAAI Artif Intell*, pp. 2750–2756, 2015.

Liu, G., Xu, H., Tang, J., Liu, Q., and Yan, S. A deterministic analysis for lrr. *IEEE Trans Pattern Anal Mach Intell*, 38(3):417–430, Mar 2016. ISSN 0162-8828. doi: 10.1109/TPAMI.2015.2453969.

Liu, G., Liu, Q., and Li, P. Blessing of dimensionality: Recovering mixture data via dictionary pursuit. *IEEE Trans Pattern Anal Mach Intell*, 39(1):47–60, Jan 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2539946.

Liu, W. and Tsang, I. W. Making decision trees feasible in ultrahigh feature and label dimensions. *Journal of Machine Learning Research*, 18:81:1–81:36, 2017.

Liu, W., Xu, D., Tsang, I. W., and Zhang, W. Metric learning for multi-output tasks. *IEEE Trans Pattern Anal Mach Intell*, 41(2):408–422, 2019.

Lu, C., Yan, S., and Lin, Z. Convex sparse spectral clustering: Single-view to multi-view. *IEEE Trans Image Process*, 25(6):2833–2843, 2016.

Lu, X., Chen, Y., and Li, X. Hierarchical recurrent neural hashing for image retrieval with hierarchical convolutional features. *IEEE Trans Image Process*, 27(1):106–120, Jan 2018. ISSN 1057-7149. doi: 10.1109/TIP.2017.2755766.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *J Mach Learn Res*, 9(Nov):2579–2605, 2008.

Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *Proc Adv Neural Inf Process Syst*, pp. 849–856, 2002.

Nie, F., Li, J., and Li, X. Self-weighted multiview clustering with multiple graphs. In *Proc Int Joint Conf Artifi Intelli*, pp. 2564–2570, 2017.

Peng, X., Yu, Z., Yi, Z., and Tang, H. Constructing the l2-graph for robust subspace learning and subspace clustering. *IEEE Trans Cybern*, 47(4):1053–1066, Apr. 2017a. ISSN 2168-2267. doi: 10.1109/TCYB.2016.2536752.

Peng, X., Feng, J., Xiao, S., Yau, W. Y., Zhou, J. T., and Yang, S. Structured autoencoders for subspace clustering. *IEEE Trans Image Process*, 27(10):5076–5086, Oct 2018. ISSN 1057-7149. doi: 10.1109/TIP.2018.2848470.

Peng, Y., Qi, J., and Yuan, Y. CM-GANs: Cross-modal Generative Adversarial Networks for Common Representation Learning. *ACM Trans Multimed Comput, Commun, and Appl*, Oct. 2017b.

Shah, S. A. and Koltun, V. Robust continuous clustering. *Proc Natl Acad Sci*, 114(37):9814–9819, Aug 2017.

Wang, W. and Livescu, K. Large-Scale Approximate Kernel Canonical Correlation Analysis. In *Int Conf Learn Rep*, San Juan, Puerto Rico, May 2016.

Wang, W., Arora, R., Livescu, K., and Bilmes, J. On Deep Multi-View Representation Learning. In *Proc Int Conf Mach Learn*, pp. 1083–1092, Lille, France, July 2015.

Xu, C., Tao, D., and Xu, C. A survey on multi-view learning. *arXiv*, April 2013.

Xu, C., Tao, D., and Xu, C. Multi-view learning with incomplete views. *IEEE Trans Image Process*, 24(12): 5812–5825, Dec 2015. ISSN 1057-7149. doi: 10.1109/TIP.2015.2490539.

Yang, E., Deng, C., Li, C., Liu, W., Li, J., and Tao, D. Shared predictive cross-modal deep quantization. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11):5292–5303, 2018.

Zhang, C., Hu, Q., Fu, H., Zhu, P., and Cao, X. Latent multi-view subspace clustering. In *Proc IEEE Conf Comput Vis Pattern Recognit*, pp. 4279–4287, 2017.

Zhang, T., Su, G., Qing, C., Xu, X., Cai, B., and Xing, X. Hierarchical lifelong learning by sharing representations and integrating hypothesis. *IEEE Trans Syst Man Cybern: Syst*, pp. 1–11, 2019. ISSN 2168-2216. doi: 10.1109/TSMC.2018.2884996.

Zhang, Z., Liu, L., Shen, F., Shen, H. T., and Shao, L. Binary multi-view clustering. *IEEE Trans Pattern Anal Mach Intell*, 2018.

Zhao, H., Ding, Z., and Fu, Y. Multi-view clustering via deep matrix factorization. In *Proc Conf AAAI Artif Intell*, pp. 2921–2927, 2017.