COMPLETER: Incomplete Multi-view Clustering via Contrastive Prediction

Yijie Lin¹, Yuanbiao Gou¹, Zitao Liu², Boyun Li¹, Jiancheng Lv¹, Xi Peng¹∗
¹ College of Computer Science, Sichuan University, China.
² TAL Education Group, China.
{lxyijie@gm, gouyuanbao, zitao.jerry.liu, liboyun@gm, pengx@gm}@gmail.com; lvjiancheng@scu.edu.cn

Abstract

In this paper, we study two challenging problems in incomplete multi-view clustering analysis, namely, i) how to learn an informative and consistent representation among different views without the help of labels and ii) how to recover the missing views from data. To this end, we propose a novel objective that incorporates representation learning and data recovery into a unified framework from the view of information theory. To be specific, the informative and consistent representation is learned by maximizing the mutual information across different views through contrastive learning, and the missing views are recovered by minimizing the conditional entropy of different views through dual prediction. To the best of our knowledge, this could be the first work to provide a theoretical framework that unifies the consistent representation learning and cross-view data recovery. Extensive experimental results show the proposed method remarkably outperforms 10 competitive multi-view clustering methods on four challenging datasets. The code is available at https://pengxi.me.

1. Introduction

In the real world, multi-view data, which often exhibit heterogeneous properties, is collected from diverse sensors or obtained from various feature extractors. As one of the most important unsupervised multi-view methods, multi-view clustering (MVC) aims to separate data points into different clusters in an unsupervised fashion [11, 17, 20, 29, 40, 54]. To achieve the end, the key is exploring the consistency across different views so that a common/shared representation is learned [5, 12, 14, 21, 33, 47]. Behind the consistency learning, the implicit assumption is that the views are complete, i.e., all data points will present in all possible views.

In practice, however, some views of data points might be missing due to the complexity in data collection and transmission, leading to so-called incomplete multi-view problem (IMP). For example, in online meetings, some video frames might lose the visual or audio signal due to the breakdown of sensors. To solve IMP, some incomplete multi-view clustering algorithms (IMC) have been proposed by employing numerous data recovery methods to complete the missing data, e.g., matrix factorization based methods [10, 22, 35, 46, 53] and generative adversarial networks based methods [16, 41, 45]. These works have attempted to overcome the following two challenges: i) how to learn informative and consistent representations across different views? and ii) how to eliminate the influence of the miss-
ing views? Although some promising results have been achieved, almost all existing works treat these two challenges as two independent problems and a unified theoretical understanding is still lacking.

Different from existing IMC studies, we theoretically show that cross-view consistency learning and data recovery could be treated as two sides of one coin and these two challenging tasks could mutually boost. Our motivation comes from [38], as shown in Fig. 1. It should be pointed out that, [38] utilizes predictive learning to enhance the performance of contrastive learning, while we aim at recovering the missing data through dual prediction. Moreover, another difference lies on our theoretical result, i.e., the data recovery and consistency learning could mutually boost through contrastive learning and dual prediction.

Based on our observations and theoretical results, we propose a novel incomplete multi-view clustering method, termed inCOMPlete muLi-view clustEring via conTrastivE pRediction (COMPLETER). In detail, COMPLETER projects a given dataset into a feature space wherein information consistency and data restorability are guaranteed using three jointly learning objectives. More specifically, a within-view reconstruction loss is used to learn a view-specific representation so that the trivial solution is avoided. In the latent feature space, a contrastive loss is introduced to learn the cross-view consistency by maximizing mutual information \( I(Z^1, Z^2) \), and a dual prediction loss is used to recover the missing view by minimizing conditional entropy \( H(Z^1 | Z^2) \) and \( H(Z^2 | Z^1) \). It should be pointed out that the data recovery referred in this paper is task-oriented, i.e., only the shared instead of all information would be recovered to facilitate the downstream tasks like MVC. To summarize:

- We provide a novel insight to the community, i.e., the data recovery and consistency learning of incomplete multi-view clustering are with intrinsic connections, which could be elegantly unified into the framework of information theory. Such a theoretical view is remarkably different from existing works which treat consistency learning and data recovery as two separate problems.

- The proposed COMPLETER method is with a novel loss function which achieves the information consistency and data restorability using a contrastive loss and a dual prediction loss. Extensive experiments verify the effectiveness of the proposed loss function.

2.1. Incomplete Multi-view Clustering

Based on the way of utilizing the multi-view information, most existing IMC methods could be roughly classified into three categories, i.e., matrix factorization (MF) based IMC [10,22,35,53], spectral clustering based IMC [39], and kernel learning based IMC [26]. In brief, MF based methods project the incomplete data into a common subspace by utilizing the low-rankness. For example, DAIMC [10] establishes a consensus basis matrix with the help of \( \ell_{2,1} \)-norm and IMG [53] utilizes the \( \ell_F \)-norm to reduce the influence of missing data. As a typical spectral clustering based method, PIC [39] learns the common representation using a consistent Laplacian graph constructed from complete views. EERIMVC [26] proposes using a multi-kernel method to achieve IMC in an iterative optimization manner. Besides, the methods like [16,41] utilize cycleGAN [55] to generate the missing view from the complete views and CDIMC-net [44] incorporates the view-specific encoders and the graph embedding strategy to handle the incomplete multi-view data.

The differences between this study and existing works are given below. First, we aim to infer the missing data rather than the missing similarity, thus enjoying higher interpretability [26]. Second, our method is a deep rather than shallow model [10, 19, 22, 26, 35, 39, 53], thus naturally embracing the capacity of handling complex and large-scale dataset. Third, almost all existing IMC methods [10,16,22,26,35,39,41,53] treat data recovery and consistency learning as two independent problems/steps, while lacking a theoretical understanding. In contrast, we proposed that data recovery and consistency learning could be unified into the framework of information theory [36]. Both data recovery and consistency learning could be of benefit to learning the common representation.

2.2. Contrastive Learning

As one of most effective unsupervised learning paradigms, contrastive learning [2, 4, 8, 23, 28, 30, 37, 38] has achieved state-of-the-art performance in representation learning. The basic idea of contrastive learning is learning a feature space from raw data by maximizing the similarity between positive pairs while minimizing that of negative pairs. In recent, some studies show that the success of contrastive learning could attribute to the maximization of mutual information. For example, MoCo [9] and CPC [30] minimize the InfoNCE loss that can be regarded as maximizing a lower bound on mutual information, i.e., \( I(Z^1, Z^2) \geq \log(N) - \mathcal{L}_{\text{NCE}} \), where \( N \) is the number of negative pairs, \( Z^1 \) and \( Z^2 \) are the latent representations of multi-view data \( X^1 \) and \( X^2 \), respectively.

The differences between this work and existing contrastive learning studies are as below. First, most existing contrastive learning methods [2, 8, 9, 28] aim to handle
3. The Proposed Method

In this section, we propose a deep multi-view clustering method, termed inCOMPlete muLit-i-view clustEr ing via conTrastivE pRediction (COMPLETER) for learning the representations with a set of incomplete multi-view samples. As illustrated in Fig. 2, COMPLETER consists of three jointly learning objectives, namely, within-view reconstruction, cross-view contrastive learning, and cross-view dual prediction. For clarity, we will first introduce the proposed loss function and then elaborate on each objective.

3.1. The Objective Function

Without loss of generality, we take bi-view data as an example. Given a dataset \( \mathbf{X} = \{ \mathbf{X}^{1,2}, \mathbf{X}^1, \mathbf{X}^2 \} \) of \( n \) instances, where \( \mathbf{X}^{1,2}, \mathbf{X}^1, \) and \( \mathbf{X}^2 \) denote the examples presented in both views, the first view only, and the second view only, respectively. Let \( m \) be the data size of complete examples \( \mathbf{X}^{1,2} \) and \( \mathbf{X}^v \) be the \( v \)-th view of \( \mathbf{X}^{1,2} \), then \( \mathbf{X}^{1,2} = \{ \mathbf{X}^1, \mathbf{X}^2 \} \).

With the above definitions, we propose the following objective function:

\[
\mathcal{L} = \mathcal{L}_{cl} + \lambda_1 \mathcal{L}_{pre} + \lambda_2 \mathcal{L}_{rec},
\]

where \( \mathcal{L}_{cl}, \mathcal{L}_{pre}, \) and \( \mathcal{L}_{rec} \) are cross-view contrastive loss, dual prediction loss, and within-view reconstruction loss, respectively. The parameters \( \lambda_1 \) and \( \lambda_2 \) are the balanced factors on \( \mathcal{L}_{pre} \) and \( \mathcal{L}_{rec} \), respectively. In our experiments, we simply fix these two parameters to 0.1.

**Within-view Reconstruction:** For each view, we pass it through an autoencoder to learn the latent representation \( \mathbf{Z}^v \) by minimizing

\[
\mathcal{L}_{rec} = \sum_{v=1}^{2} \sum_{t=1}^{m} \left\| \mathbf{X}^v_t - g^{(v)} \left( f^{(v)} (\mathbf{X}^v_t) \right) \right\|_2^2,
\]

where \( \mathbf{X}^v_t \) denotes the \( t \)-th sample of \( \mathbf{X}^v \). \( f^{(v)} \) and \( g^{(v)} \) denote the encoder and decoder for the \( v \)-th view, respectively. Hence, the representation of \( t \)-th sample in \( v \)-th view is given by

\[
\mathbf{Z}^v_t = f^{(v)} (\mathbf{X}^v_t),
\]

where \( \mathbf{Z}^v \) denotes the representations of \( \mathbf{X}^v \) and \( v \in \{ 1, 2 \} \).

It should be pointed out that the autoencoder structure is helpful to avoid the trivial solution.

**Cross-view Contrastive Learning:** In the latent space parameterized by \( \mathcal{L}_{rec} \), we conduct contrastive learning to learn a common representation shared across different views. Unlike most existing contrastive learning studies [9, 30] which maximize the consistency between the
learned representations $Z^1$ and $Z^2$ by maximizing the lower bound of mutual information, we directly maximize the mutual information between the representations of different views. Mathematically,

$$L_{cl} = - \sum_{i=1}^{m} (I(Z_i^1, Z_i^2) + \alpha (H(Z_i^1) + H(Z_i^2))) ,$$

where $I$ denotes the mutual information, $H$ is the information entropy, and parameter $\alpha$ is set as 9 to regularize the entropy in our experiments. We design this objective with the following goals. On the one hand, from information theory, information entropy is the average amount of information conveyed by an event [3]. Hence a larger entropy $H(Z^i)$ denotes a more informative representation $Z^i$. On the other hand, the maximization of $H(Z^1)$ and $H(Z^2)$ will avoid the trivial solution of assigning all samples to the same cluster.

To formulate $I(Z^1, Z^2)$, we first define the joint probability distribution $P(z, z')$ of variable $z$ and $z'$. As a softmax function is stacked at the last layer of the encoder, each element of $Z^1$ and $Z^2$ could be regarded as an over-cluster class probability like $[13, 15, 34]$. In other words, $Z^1$ and $Z^2$ could be understood as the distribution of two discrete cluster assignment variables $z$ and $z'$ over $D$ “classes”, where $D$ is the dimension of $Z^1$ and $Z^2$. As a result, $P(z, z')$ is defined as $P \in R^{D \times D}$, i.e.,

$$P = \frac{1}{m} \sum_{i=1}^{m} Z_i^1 (Z_i^2)^\top .$$

Let $P_d$ and $P'_d$ denote the marginal probability distributions $P(z = d)$ and $P(z' = d')$, they could be obtained by summing over the $d$-th rows and $d'$-th columns of joint probability distribution matrix $P$. Expecting $z$ and $z'$ are with equal importance, $P$ is further calculated by $(P + P')/2$. For discrete distributions, Eq. (4) is given as below:

$$L_{cl} = - \sum_{d=1}^{D} \sum_{d'=1}^{D} P_{dd'} \ln \frac{P_{dd'}}{P_{d+1}.P_{d'+1}},$$

where $P_{dd'}$ is the element at the $d$-th row and $d'$-th column of $P$ and $\alpha$ is a balance parameter of entropy as defined in Eq. (4). The details from Eq. (4) to Eq. (6) are presented in supplementary material.

**Cross-view Dual Prediction:** To infer the missing views, we propose a dual prediction mechanism as shown in Fig. 2. To be specific, in a latent space parameterized by a neural network, the view-specific representation will be predicted by another through minimizing the entropy $H(Z|Z')$, where $i = 1, j = 2$ or $i = 2, j = 1$. Such a dual prediction mechanism is with theoretical explanation as elaborated in Fig. 1. In short, $Z^1$ is fully determined by $Z^j$ if and only if the conditional entropy $H(Z^i|Z^j) = -\mathbb{E}_{P_{Z^i, Z^j}}[\log P(Z^i|Z^j)] = 0$. To solve this objective, a common approximative approach is introducing a variational distribution $Q(Z^i|Z^j)$ and maximizing $\mathbb{E}_{P_{Z^i, Z^j}}[\log Q(Z^i|Z^j)]$ which is the lower bound of $\mathbb{E}_{P_{Z^i, Z^j}}[\log P(Z^i|Z^j)]$, i.e.,

$$\mathbb{E}_{P_{Z^i, Z^j}}[\log P(Z^i|Z^j)] = \mathbb{E}_{P_{Z^i, Z^j}}[\log Q(Z^i|Z^j)] + D_{KL}(P(Z^i|Z^j) \parallel Q(Z^i|Z^j)) .$$

Such a variational distribution $Q$ can be any types like Gaussian [7] and Laplacian distribution [55]. In practice, we simply assume the distribution $Q$ as a Gaussian distribution $\mathcal{N}(Z^i|G^{(j)}(Z^j), \sigma I)$, where $G^{(j)}(\cdot)$ could be a parameterized model which maps $Z^j$ to $Z^i$ and $\sigma I$ is the variance matrix. By ignoring the constants derived from the Gaussian distribution, maximizing $\mathbb{E}_{P_{Z^i, Z^j}}[\log Q(Z^i|Z^j)]$ is equivalent to

$$\min \mathbb{E}_{P_{Z^i, Z^j}} \|Z^i - G^{(j)}(Z^j)\|_2^2 .$$

For a given bi-view dataset, we further have

$$L_{pre} = \|G^{(1)}(Z^1) - Z^2\|_2^2 + \|G^{(2)}(Z^2) - Z^1\|_2^2 .$$

It should be pointed out that the above loss may lead to trivial solutions without the within-view reconstruction loss, i.e., $Z^1$ and $Z^2$ are equivalent to the same constant.

After the model converged, it is easy to predict the missing representation $Z^j$ from $Z^i$ through the above dual mapping, i.e.,

$$Z^j = G^{(j)}(Z^i) = G^{(j)}(f^{(j)}(X^j)) ,$$

where $\hat{X}^j$ denotes the representations of $X^j$.

### 3.2. Implementation Details

As shown in Fig. 2, COMPLETER consists of two training modules, i.e., two view-specific autoencoders and two cross-view prediction networks. For these two modules, we simply adopt a fully-connected network where each layer is followed by a batch normalization layer and a ReLU layer. The softmax activation function is used at the last layer of the encoder and prediction module. In the supplementary material, all details of our model have been presented.

In the training stage, we use the complete data $\hat{X}^{1,2}$ to train COMPLETER in an end-to-end manner. Specifically, we train the autoencoders by $L_{cl}$ and $L_{rec}$ in the first 400 epochs to stabilize the training of the dual prediction. Then, we train the whole networks with $L$ for 400 epochs. Once the network converged, we feed the whole dataset into the network to obtain the representations for all views including
Table 1. The clustering performance comparisons on four challenging datasets. “−” indicates unavailable results due to out of memory. The 1st/2nd best results are indicated in red/blue.

<table>
<thead>
<tr>
<th>Missing Type</th>
<th>Method</th>
<th>Caltech101-20</th>
<th>LandUse-21</th>
<th>Scene-15</th>
<th>Noisy MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC</td>
<td>NMI</td>
<td>ARI</td>
<td>ACC</td>
<td>NMI</td>
</tr>
<tr>
<td>Incomplete</td>
<td>AE$^2$Nets [51]</td>
<td>33.61</td>
<td>49.20</td>
<td>24.99</td>
<td>19.22</td>
</tr>
<tr>
<td></td>
<td>IMG [53]</td>
<td>42.29</td>
<td>58.26</td>
<td>33.69</td>
<td>15.52</td>
</tr>
<tr>
<td></td>
<td>UEAf [43]</td>
<td>47.35</td>
<td>56.71</td>
<td>37.08</td>
<td>16.38</td>
</tr>
<tr>
<td></td>
<td>DAIMC [10]</td>
<td>44.63</td>
<td>59.53</td>
<td>32.70</td>
<td>19.30</td>
</tr>
<tr>
<td></td>
<td>EERIMVC [26]</td>
<td>40.66</td>
<td>51.38</td>
<td>27.91</td>
<td>24.35</td>
</tr>
<tr>
<td></td>
<td>DCCA [42]</td>
<td>40.01</td>
<td>52.88</td>
<td>30.00</td>
<td>14.94</td>
</tr>
<tr>
<td></td>
<td>PVC [22]</td>
<td>41.42</td>
<td>56.53</td>
<td>31.00</td>
<td>21.33</td>
</tr>
<tr>
<td></td>
<td>BMVC [52]</td>
<td>32.13</td>
<td>40.58</td>
<td>12.20</td>
<td>18.76</td>
</tr>
<tr>
<td></td>
<td>DCCA [1]</td>
<td>38.59</td>
<td>52.51</td>
<td>29.81</td>
<td>14.08</td>
</tr>
<tr>
<td></td>
<td>PIC [39]</td>
<td>57.53</td>
<td>64.32</td>
<td>45.22</td>
<td>23.60</td>
</tr>
<tr>
<td></td>
<td>COMPLETER</td>
<td>68.44</td>
<td>67.39</td>
<td>75.44</td>
<td>22.16</td>
</tr>
<tr>
<td>Complete</td>
<td>AE$^2$Nets [51]</td>
<td>49.10</td>
<td>65.38</td>
<td>35.66</td>
<td>24.79</td>
</tr>
<tr>
<td></td>
<td>IMG [53]</td>
<td>44.51</td>
<td>61.35</td>
<td>35.74</td>
<td>16.40</td>
</tr>
<tr>
<td></td>
<td>UEAf [43]</td>
<td>47.40</td>
<td>57.90</td>
<td>38.98</td>
<td>23.00</td>
</tr>
<tr>
<td></td>
<td>DAIMC [10]</td>
<td>45.48</td>
<td>61.79</td>
<td>32.40</td>
<td>24.35</td>
</tr>
<tr>
<td></td>
<td>EERIMVC [26]</td>
<td>43.28</td>
<td>55.04</td>
<td>30.42</td>
<td>24.92</td>
</tr>
<tr>
<td></td>
<td>DCCA [42]</td>
<td>44.05</td>
<td>59.12</td>
<td>34.56</td>
<td>15.62</td>
</tr>
<tr>
<td></td>
<td>PVC [22]</td>
<td>44.91</td>
<td>62.13</td>
<td>35.77</td>
<td>25.22</td>
</tr>
<tr>
<td></td>
<td>BMVC [52]</td>
<td>42.55</td>
<td>63.63</td>
<td>32.33</td>
<td>25.34</td>
</tr>
<tr>
<td></td>
<td>DCCA [1]</td>
<td>41.89</td>
<td>59.14</td>
<td>33.39</td>
<td>15.51</td>
</tr>
<tr>
<td></td>
<td>PIC [39]</td>
<td>62.27</td>
<td>67.93</td>
<td>51.56</td>
<td>24.86</td>
</tr>
<tr>
<td></td>
<td>COMPLETER</td>
<td>70.18</td>
<td>68.06</td>
<td>77.88</td>
<td>25.63</td>
</tr>
</tbody>
</table>

the missing ones. After that, the common representation, which is obtained by simply concatenating all view-specific representations together, is further fed into $k$-means to get the clustering results like the traditional fashion [1, 10, 22, 25, 26, 32, 39, 42, 43, 48, 49, 53].

4. Experiments

In this section, we evaluate the proposed COMPLETER method on four widely-used multi-view datasets with the comparisons of 10 multi-view clustering approaches.

4.1. Experimental Settings

Four widely-used datasets are used in our experiments. In brief, Caltech101-20 [24] consists of 2,386 images of 20 subjects with the views of HOG and GIST features. Scene-15 [6], which consists of 4,485 images distributed over 15 scene categories, is with PHOG and GIST features. LandUse-21 [50] consists of 2100 satellite images from 21 categories with PHOG and LBP features. Noisy MNIST [42] uses the original 70k MNIST images as view 1 and randomly selects within-class images with white Gaussian noise as view 2. As most of the baselines cannot handle such a large dataset, we could only use a subset of Noisy MNIST consisting of 10k validation images and 10k testing images.

To evaluate the performance of handling incomplete multi-view data, we randomly select some instances as incomplete data by randomly removing one view. The missing rate $\eta$ is defined as $\eta = (n - m)/n$, where $m$ is the number of complete examples, and $n$ is the number of the whole dataset.

For a comprehensive analysis, three widely-used clustering metrics including Normalized Mutual Information (NMI), Accuracy (ACC), and Adjusted Rand Index (ARI) are used. A higher value of these metrics indicates a better clustering performance.

We implement our COMPLETER in PyTorch 1.2 [31] and carry all evaluations on a standard Ubuntu-18.04 OS with an NVIDIA 2080Ti GPU. We use Adam optimizer [18] with the default parameters to train our model and set the initial learning rate as 0.0001. The batch size is set to 256 and the maximal training epoch is fixed to 500 on all datasets. The entropy parameter $\alpha$ is fixed to $9$ and trade-off hyper-parameters $\lambda_1$ and $\lambda_2$ are fixed to 0.1 for all datasets. In our implementation environment, COMPLETER takes about 60 seconds to train a model on Caltech101-20, 80 seconds on Scene-15, 50 seconds on LandUse-15, and 500 seconds on NoisyMNIST.
4.2. Comparisons with State of the Arts

We compare COMPLETER with 10 multi-view clustering approaches including Deep Canonically Correlated Analysis (DCCA) [1], Deep Canonically Correlated Autoencoders (DCCAE) [42], Binary Multi-view Clustering (BMVC) [52], Autoencoder in Autoencoder Networks (AE^2-Nets) [51], Partial Multi-View Clustering (PVC) [22], Efficient and Effective Regularized Incomplete Multi-view Clustering (EERIMVC) [26], Doubly Aligned Incomplete Multi-view Clustering (DAIMC) [10], Incomplete Multi-Modal Visual Data Grouping (IMG) [53], Unified Embedding Alignment Framework (UEAF) [43], and Perturbation-oriented Incomplete Multi-view Clustering (PIC) [39]. The first four methods could only handle complete multi-view data and thus we fill the missing data with the mean values of the same view. For all methods, we use the recommended network structure and parameters for fair comparisons. In brief, for CCA-based methods (i.e., DCCA and DCCAE), we fix the hidden representation dimension to 10. For BMVC, we fix the length of binary code to 128. For EERIMVC, we exploit the “Gauss kernel” to construct the kernel matrices and seek the optimal $\lambda$ from $2^{-15}$ to $2^{15}$ with an interval of $2^3$. We test all methods in two settings, i.e., missing rate $\eta = 0.5$ (denoted by Incomplete) and $\eta = 0$ (denoted by Complete). The average clustering results are obtained by repeating each method with five random initializations and dataset partitions.

As shown in Table 1, COMPLETER significantly outperforms these state-of-the-art baselines by a large performance margin on all four datasets. In the Incomplete setting, COMPLETER surpasses the best baseline by 3.07% on Caltech101-20, 4.37% on Scene-15, and 14.68% on NoisyMNIST in terms of NMI. Moreover, COMPLETER achieves more than 50% performance improvements over the best baseline on Caltech101-20 and NoisyMNIST in terms of ARI. In the Complete setting, COMPLETER also remarkably outperforms almost all baselines. The encouraging performance demonstrates the promising representability of COMPLETER thanks to our unified theoretical framework of contrastive learning and dual prediction.

4.3. Performance with Different Missing Rates

To further investigate the effectiveness of our method, we conduct experiments by varying the missing rate $\eta$ from 0 to 0.9 with a gap of 0.1 on Caltech101-20. When the missing rate is 0.9, the size of the whole training data is smaller than that of a data batch, and thus we reduce the batch size to 128. From the results in Fig. 3, one could observe that: i) COMPLETER significantly outperforms all
the tested baselines in all missing rates setting; ii) with increasing the missing rate, the performance degradations of the compared methods are much larger than that of ours. For example, COMPLETER and PIC achieve the NMI of 0.6806 and 0.6793 with $\eta = 0$, respectively, while with the increase of the missing rate, COMPLETER is remarkably superior to PIC.

4.4. Parameter Analysis and Ablation Studies

In this section, we analyze COMPLETER on the Caltech101-20 dataset from two perspectives, i.e., parameter sensitivity analysis and ablation studies. In the evaluations, the missing rate $\eta$ is fixed to 0.5.

Our method contains three user-specified parameters, i.e., the entropy parameter $\alpha$, the prediction trade-off parameter $\lambda_1$, and the reconstruction trade-off parameter $\lambda_2$. In the following studies, we first investigate the relation among $\alpha$, information entropy of representations $H(Z^i)$, and clustering performance by fixing $\lambda_1$ and $\lambda_2$ to 0.1 and changing the value of $\alpha$. As shown in Fig. 5, the information entropy grows in step with $\alpha$. Specifically, with the increase of the information entropy (from left to right), the clustering performance (ACC, NMI, and ARI) improves first and then degrades. The reason may due to the following aspects. On the one hand, the increased entropy (information contained in the representation) will enlarge the mutual information which further boosts the clustering performance. On the other hand, with the increase of $\alpha$, an over-informative representation will suppress the mutual information term in Eq. (4) and then the consistency is reduced.

To evaluate the influence of $\lambda_1$ and $\lambda_2$, we change their value in the range of $\{0.01, 0.1, 1, 10, 100\}$. As shown in Fig. 4, our method is robust to the choice of $\lambda_1$. In addition, a good choice of $\lambda_2$ will remarkably improve the performance of COMPLETER.

To further verify the importance of each module in COMPLETER, we conduct the following ablation study. In detail, the following seven experiments are designed to isolate the effect of the contrastive loss $L_{cl}$, the reconstruction loss $L_{rec}$, and the dual prediction loss $L_{pre}$. As shown in Table 2, all loss terms play indispensable roles in COMPLETER. It should be pointed out that optimizing the dual prediction loss $L_{pre}$ alone may lead to trivial solutions. To solve this problem, we add a batch normalization layer to each fully-connected layer and report the corresponding results.

4.5. Visualization Verification on Our Theoretical Results

In this section, we carry out experiments to verify our theoretical results presented in Fig. 1. The experiments are conducted on Noisy MNIST dataset by visualizing the recovered views and the common representations. In the experiments, the missing rate $\eta$ is fixed to 0.5.

Different from most existing incomplete multi-view methods, COMPLETER could explicitly infer the representation of the missing views. As a result, the corresponding reconstruction in the original space could be obtained through the decoder. To show the recoverability of COMPLETER, Fig. 6 visually illustrates some recovered exam-
ples from Noisy MNIST. From the results, one could have the following observations. In the top three rows, the recovered images (Line 3) are much similar to the complete ones (Line 1), while being with a clean background like the missing view (Line 2). In the bottom three rows, a similar observation could also be obtained even though COMPLETER recovered the missing images from the images with a clean rather than noisy background. In short, COMPLETER could recover the important information while discarding the indistinct characteristics like noises in this example.

It should be noticed that the semantic information and the noisy background in this example could be regarded as consistency and inconsistency of two views. Therefore, the reasons for the above observations are two-fold. On the one hand, the recovered views will contain the shared information (semantic information instead of noise) of two available views thanks to the maximization of the mutual information. On the other hand, the minimization of conditional entropy designed for data recovery could subtly discard the inconsistent information across different views. As a result, the noise in the missing views will be suppressed during recovery. This verifies the effectiveness of our theory.

Besides the above visualizations, we also demonstrate the t-sne [27] visualizations of the learned common representations. As shown in Fig. 7, the learned representations become more compact and discriminative with the increase of the epoch.

4.6. Convergence Analysis

In this section, we investigate the convergence of COMPLETER by reporting the loss value and the corresponding clustering performance with increasing epochs. As shown in Fig. 8, one could observe that the loss remarkably decreases in the first 200 epochs, and meanwhile ACC, NMI, and ARI continuously increase.

5. Conclusion

To learn common representations from a given multi-view data wherein some views are missing, this paper proposed COMPLETER which embraces the rigid mathematical motivation and explanation from information theory. In short, we treat consistency learning and view completing as two sides of one coin rather than two separate problems. Such a unified framework would provide novel insight to the community on understanding consistency learning and data recovery. In the future, we plan to further explore the potential of our theoretical framework in other multi-view learning tasks, e.g., object ReId. Moreover, it is also promising to extend it to handle the image translation tasks.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2020AAA0104500; in part by the Fundamental Research Funds for the Central Universities under Grant YJ201949; in part by NSFC under Grant U19A2081, 61625204, 61836006, U19A2078; in part by the Fund of Sichuan University Tomorrow Advancing Life; and in part by Beijing Nova Program (Z201100006820068) from Beijing Municipal Science & Technology Commission.
References


[34] Xi Peng, Hongyuan Zhu, Jiashi Feng, Chunhua Shen, Haixian Zhang, and Joey Tianyi Zhou. Deep clustering with sample-assignment invariance prior. IEEE Transactions on...
Weixiang Shao, Lifang He, and S Yu Philip. Multiple incomplete views clustering via weighted nonnegative matrix factorization with $l_{2,1}$ regularization. In ECML PKDD, pages 318–334, 2015. 1, 2


Qi Wang, Mulin Chen, Feiping Nie, and Xuelong Li. Detecting coherent groups in crowd scenes by multiview clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(1):46–58, 2018. 1

Qianqian Wang, Zhengming Ding, Zhiqiang Tao, Quanxue Gao, and Yun Fu. Partial multi-view clustering via consistent gan. In ICDM, pages 1290–1295, 2018. 1, 2


Handong Zhao, Hongfu Liu, and Yun Fu. Incomplete multi-modal visual data grouping. In IJCAI, pages 2392–2398, 2016. 1, 2, 5, 6


1. Introduction

In this supplementary material, we provide additional information including mathematical notations, mathematical derivation of our loss, network architectures, and implementation details. To investigate the effectiveness of our method, we also conduct some additional experimental analysis.

2. Notations and Definitions.

In this section, we summarize the mathematical notations used throughout the manuscript in Table 1 for a clear reference. The bi-view dataset $\bar{X}$ consists of three parts, $\bar{X}^{1,2}$, $\bar{X}^1$, and $\bar{X}^2$, where $\bar{X}^{1,2}$, $\bar{X}^1$, and $\bar{X}^2$ denote the examples presented in all views, the first view only, and the second view only, respectively. $n$ and $m$ denote the number of the data points of the whole dataset $\bar{X}$ and $\bar{X}^{1,2}$, respectively. More specifically, Fig. 1 visually illustrates our setting and some notations by taking a dataset as a showcase.

![Image of incomplete bi-view dataset](image)

Figure 1. Illustrations of the incomplete bi-view dataset $\bar{X}$.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{X}$</td>
<td>Incomplete bi-view dataset where $\bar{X} = {\bar{X}^{1,2}, \bar{X}^1, \bar{X}^2}$</td>
</tr>
<tr>
<td>$\bar{X}^{1,2}$</td>
<td>Set of examples presented in both views.</td>
</tr>
<tr>
<td>$\bar{X}^v$</td>
<td>Set of examples only presented in view $v$.</td>
</tr>
<tr>
<td>$X^v$</td>
<td>The $v$-th view of complete samples $\bar{X}^{1,2}$, i.e., $\bar{X}^{1,2} = {X^1, X^2}$.</td>
</tr>
<tr>
<td>$Z^v$</td>
<td>The representations of $X^v$.</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of examples presented in $\bar{X}$.</td>
</tr>
<tr>
<td>$m$</td>
<td>Number of examples presented in $\bar{X}^{1,2}$.</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Trade-off parameter of information entropy.</td>
</tr>
<tr>
<td>$\lambda_1$</td>
<td>Trade-off parameter of dual prediction loss.</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>Trade-off parameter of reconstruction loss.</td>
</tr>
<tr>
<td>$f(v)$</td>
<td>Encoder of $v$-th view.</td>
</tr>
<tr>
<td>$g(v)$</td>
<td>Decoder of $v$-th view.</td>
</tr>
<tr>
<td>$G(i)$</td>
<td>Predictor which recovers missing representation $Z^j$ from complete one $Z^i$.</td>
</tr>
</tbody>
</table>

Table 1. Mathematical notations in the manuscript.

3. Theoretical Derivation on Our Loss

In this section, we elaborate on the derivations of our loss that are omitted in the manuscript due to the space limitation.

3.1. Cross-view Contrastive Loss

We have imported the information entropy to the standard definitions of mutual information in the manuscript. Mathematically, the contrastive loss is defined as,

$$L_{cl} = -\sum_t m \left(I \left(Z^1_t, Z^2_t\right) + \alpha \left(H \left(Z^1_t\right) + H \left(Z^2_t\right)\right)\right). \ (1)$$

As illustrated in the manuscript, the representations of $t$-th sample $Z^1_t \in \mathbb{R}^D$ and $Z^2_t \in \mathbb{R}^D$ can be interpreted as the distribution of discrete random variables $z$ and $z'$ over $D$ classes, respectively. In other words, the probability distributions $P(z = d)$ is the $d$-th element of $Z^1_t$ where...
1 \leq d \leq D$. Hence, considering the case of a data set or batch, the joint probability distribution $P(z, z')$ of variable $z$ and $z'$ could be defined by $P \in \mathcal{R}^{D \times D}$, i.e.,
\[
P = \frac{1}{m} \sum_{i=1}^{m} Z_i \left( Z_i' \right) \top.
\] (2)

Thus, for discrete distributions, the mutual information and information entropy are given as below:
\[
I(Z^1, Z^2) = \sum_{z=1}^{D} \sum_{z'=1}^{D} P(z, z') \log \left( \frac{P(z, z')}{P(z)P(z')} \right)
\] (3)
\[
= \sum_{d=1}^{D} \sum_{d'=1}^{D} P_{dd'} \ln \frac{P_{dd'}}{P_d \cdot P_{d'}}
\]
\[
H(Z^1) = - \sum_{z=1}^{D} P(z) \log P(z) = - \sum_{d=1}^{D} P_d \ln P_d,
\] (4)
\[
H(Z^2) = - \sum_{z'=1}^{D} P(z') \log P(z') = - \sum_{d'=1}^{D} P_{d'} \ln P_{d'},
\] (5)
where $P_d$ and $P_{d'}$ denote the marginal probability distributions $P(z = d)$ and $P(z' = d')$ which could be obtained by summing over the $d$-th row and $d'$-th column of $P$, respectively. By substituting Eq. (3), Eq. (4), and Eq. (5) into Eq. (1), we could obtain the final form of our cross-view contrastive loss as below:
\[
L_{cl} = - (I(Z^1, Z^2) + \alpha (H(Z^1) + H(Z^2)))
\]
\[
= - \sum_{d=1}^{D} \sum_{d'=1}^{D} P_{dd'} \ln \frac{P_{dd'}}{P_d \cdot P_{d'}} - \alpha \left( \sum_{d=1}^{D} P_d \ln P_d + \sum_{d'=1}^{D} P_{d'} \ln P_{d'} \right)
\]
\[
= - \sum_{d=1}^{D} \sum_{d'=1}^{D} P_{dd'} \ln \frac{P_{dd'}}{P_d \cdot P_{d'}} + \alpha \left( \sum_{d=1}^{D} P_d \ln \frac{1}{P_d} + \sum_{d'=1}^{D} \sum_{d'=1}^{D} P_{dd'} \ln \frac{1}{P_{d'}} \right)
\]
\[
= - \sum_{d=1}^{D} \sum_{d'=1}^{D} P_{dd'} \ln \frac{P_{dd'}}{P_d \cdot P_{d'}} + \alpha \left( \ln \frac{1}{P_d} + \ln \frac{1}{P_{d'}} \right)
\]
\[
= - \sum_{d=1}^{D} \sum_{d'=1}^{D} P_{dd'} \ln \frac{P_{dd'}}{P_d \cdot P_{d'}} + \alpha \left( \frac{1}{P_d} + \frac{1}{P_{d'}} \right)
\]
\[
= - \sum_{d=1}^{D} \sum_{d'=1}^{D} P_{dd'} \ln \frac{P_{dd'}}{P_d \cdot P_{d'}} + \alpha \left( \frac{1}{P_d} \cdot \frac{1}{P_{d'}} \right)
\] (6)

3.2. Cross-view Dual Prediction Loss

To infer the missing views $Z^i$ from $Z^j$, we propose to minimize the conditional entropy $H(Z^i | Z^j) = -E_{P_{Z^i, Z^j}}[\log P(Z^i | Z^j)]$. As it is intractable to solve such a problem, we introduce a variational distribution $Q(Z^i | Z^j)$ and further maximize the lower bound of $E_{P_{Z^i, Z^j}}[\log P(Z^i | Z^j)]$, i.e., $E_{P_{Z^i, Z^j}}[\log Q(Z^i | Z^j)]$. To be specific, we assume $Q$ as a Gaussian distribution $N(Z^i | G(j)(Z^j), \sigma I)$, where $G(j)(\cdot)$ is the predictor which maps $Z^j$ to $Z^i$ and $\sigma I$ is a variance matrix. As a result, we have
\[
\max E_{P_{Z^i, Z^j}} \left[ \log Q \left( Z^i | Z^j \right) \right] = E_{P_{Z^i, Z^j}} \left[ \log \left( \frac{1}{\sqrt{\sigma I} \sqrt{2\pi}} e^{-\frac{(Z^i - G(j)(Z^j))^2}{2\sigma I}} \right) \right]
\] (7)
which could be equivalent to the following optimization problem:
\[
\max E_{P_{Z^i, Z^j}} \left[ - \left( Z^i - G(j)(Z^j) \right)^2 \right] + \log \frac{1}{\sqrt{2\pi\sigma I}}
\] (8)

By ignoring the constant $\log \frac{1}{\sqrt{2\pi\sigma I}}$ and scaling factor $2\sigma I$, we could obtain the prediction loss as below,
\[
\max -E_{P_{Z^i, Z^j}} \left[ \left( Z^i - G(j)(Z^j) \right)^2 \right].
\] (9)

Hence, for a given bi-view dataset, the dual prediction loss is accordingly defined as
\[
L_{pre} = \left\| G^{(1)}(Z^1) - Z^2 \right\|^2_2 + \left\| G^{(2)}(Z^2) - Z^1 \right\|^2_2.
\] (10)

4. Experiment Details

In this section, we elaborate on the implementation details of our method and the experimental settings.

4.1. Network Architectures of COMPLER

The proposed method contains two training modules, i.e., view-specific auto-encoders and cross-view prediction networks. Table 2 and 3 have presented the details of the network architectures in these two training modules, respectively. For these two modules, we simply adopt a dense (i.e., fully-connected) network where each layer is followed by a batch normalization layer and a ReLU layer. The softmax activation function is used at the last layer of the encoders and prediction modules. The structures of auto-encoders $f^{(1)}$, $g^{(1)}$, and predictors $G^{(j)}$ for different views are the same. Specifically, the size of the output of the encoder and predictor should be the same and set to 64 or 128 according to the dataset.

4.2. Implementation Details for Clustering

To perform clustering, we adopt $k$-means to compute the cluster assignments on the common representation which is
Table 2. The architecture of the autoencoders in COMPLETER.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Encoder</th>
<th>Decoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caltech101-20</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
</tr>
<tr>
<td></td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
</tr>
<tr>
<td></td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
</tr>
<tr>
<td></td>
<td>Dense (Softmax, size = 128)</td>
<td>Dense (BatchNorm1d, ReLU, size = input)</td>
</tr>
<tr>
<td>LandUse-21</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
</tr>
<tr>
<td></td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
</tr>
<tr>
<td></td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
</tr>
<tr>
<td></td>
<td>Dense (Softmax, size = 64)</td>
<td>Dense (BatchNorm1d, ReLU, size = input)</td>
</tr>
<tr>
<td>Scene-15</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
</tr>
<tr>
<td></td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
</tr>
<tr>
<td></td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
</tr>
<tr>
<td></td>
<td>Dense (Softmax, size = 128)</td>
<td>Dense (BatchNorm1d, ReLU, size = input)</td>
</tr>
<tr>
<td>Noisy MNIST</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
</tr>
<tr>
<td></td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
</tr>
<tr>
<td></td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
<td>Dense (BatchNorm1d, ReLU, size = 1024)</td>
</tr>
<tr>
<td></td>
<td>Dense (Softmax, size = 64)</td>
<td>Dense (BatchNorm1d, ReLU, size = input)</td>
</tr>
</tbody>
</table>

Table 3. The architecture of dual prediction in COMPLETER.

<table>
<thead>
<tr>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense (BatchNorm1d, ReLU, size = 128)</td>
</tr>
<tr>
<td>Dense (BatchNorm1d, ReLU, size = 256)</td>
</tr>
<tr>
<td>Dense (BatchNorm1d, ReLU, size = 128)</td>
</tr>
<tr>
<td>Dense (BatchNorm1d, ReLU, size = 256)</td>
</tr>
<tr>
<td>Dense (BatchNorm1d, ReLU, size = 128)</td>
</tr>
<tr>
<td>Dense (Softmax, size = input)</td>
</tr>
</tbody>
</table>

Table 4. Influence of dimensionality.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Dimension</th>
<th>ACC</th>
<th>NMI</th>
<th>ARI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caltech101-20</td>
<td>32</td>
<td>43.48</td>
<td>60.31</td>
<td>41.50</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>51.99</td>
<td>62.88</td>
<td>47.91</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>68.44</td>
<td>67.39</td>
<td>75.44</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>69.56</td>
<td>65.63</td>
<td>74.54</td>
</tr>
<tr>
<td>Scene-15</td>
<td>32</td>
<td>37.30</td>
<td>40.79</td>
<td>21.41</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>37.60</td>
<td>41.01</td>
<td>20.55</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>39.50</td>
<td>42.35</td>
<td>23.51</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>36.37</td>
<td>41.87</td>
<td>22.10</td>
</tr>
</tbody>
</table>

obtained by simply concatenating all view-specific representations together. Specifically, we use the $k$-means contained in the Scikit-Learn package [3] with the default configuration. For a fair comparison, we run all the used methods five times with different initializations and data partitions to obtain the common representations. For each run, we further conduct $k$-means 10 times on the representation to obtain the clustering results. In all experiments, we adopt three evaluation metrics implemented by Scikit-Learn to evaluate the clustering performance, namely, ACC, NMI, and ARI.

5. Additional Experiments

This section presents two experimental studies including: i) the influence of dimensionality of latent representations and ii) clustering performance on the full datasets.

5.1. Influence of Dimensionality

In the proposed method, we treat each element of the representation as an over-cluster class probability like [1, 2, 4]. To evaluate the effectiveness of such over-clustering strategy, we change the dimensionality of the representation in the range of {32, 64, 128, 256}. The missing rate $\eta$ is fixed to 0.5 and the results are shown in Table 4. The results demonstrate that a too large or too small dimensionality will cause performance degradation. The former is of insufficient representability and the latter may have some redundant information.

5.2. Experiment on the Full Datasets

In the main body of the manuscript, we only report the results on the 20k subsets of the Noisy MNIST dataset because most of the baselines are inefficient to handle large scale datasets. In this evaluation, we carry out clustering experiments on the whole Noisy MNIST dataset and report the results compared with scalable methods including DCCA [5], DCCAE [5], BMVC [7], and AE2Nets [6]. Similarly, we also test these methods in both Incomplete ($\eta = 0.5$) and Complete ($\eta = 0$) settings. As shown in Table 5, COMPLETER still outperforms all baselines.
Table 5. Performance comparisons on full Noisy MNIST.

<table>
<thead>
<tr>
<th>Missing Type</th>
<th>Method</th>
<th>ACC</th>
<th>NMI</th>
<th>ARI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incomplete</td>
<td>DCCA</td>
<td>45.32</td>
<td>48.73</td>
<td>25.70</td>
</tr>
<tr>
<td></td>
<td>DCCAE</td>
<td>49.44</td>
<td>48.49</td>
<td>25.31</td>
</tr>
<tr>
<td></td>
<td>AE2Nets</td>
<td>37.76</td>
<td>35.53</td>
<td>20.57</td>
</tr>
<tr>
<td></td>
<td>BMVC</td>
<td>46.42</td>
<td>36.23</td>
<td>22.34</td>
</tr>
<tr>
<td></td>
<td>COMPLETE</td>
<td>94.28</td>
<td>87.39</td>
<td>88.12</td>
</tr>
<tr>
<td>Complete</td>
<td>DCCA</td>
<td>89.29</td>
<td>91.35</td>
<td>87.04</td>
</tr>
<tr>
<td></td>
<td>DCCAE</td>
<td>89.03</td>
<td>91.40</td>
<td>87.77</td>
</tr>
<tr>
<td></td>
<td>AE2Nets</td>
<td>50.70</td>
<td>53.26</td>
<td>40.49</td>
</tr>
<tr>
<td></td>
<td>BMVC</td>
<td>91.57</td>
<td>83.55</td>
<td>83.83</td>
</tr>
<tr>
<td></td>
<td>COMPLETE</td>
<td>97.17</td>
<td>94.19</td>
<td>93.58</td>
</tr>
</tbody>
</table>

References