Supplementary Material for Deep Incomplete Multi-view Clustering with Cross-view Partial Sample and Prototype Alignment

Jiaqi Jin[1], Siwei Wang[1], Zhibin Dong[1], Xinwang Liu[1,*], En Zhu[1,*]
[1]School of Computer, National University of Defense Technology, Changsha, China
{jinjiaqi, wangsiwei13, dzb20, xinwangliu, enzhu}@nudt.edu.cn

1. Introduction

In this supplementary material, we provide the main notations used in the paper and elaborate the implementation details of CPSPAN in the clustering process in more detail than the original paper. In addition, in order to further verify the effectiveness of the proposed CPSPAN, we conduct some additional experiments and analyses.

2. Notations

In this section, we supplement the main notations used in the whole paper in Table 1 for reference.

Table 1. Basic notations used in the whole paper.

Notation	Meaning
N	number of instances
K	number of clusters
V	number of views
D_v	dimension of instances in v-th view
$\mathbf{X}^{(v)} \in \mathbb{R}^{N \times D_v}$	data matrix for the v-th view
$\tilde{\mathbf{X}}^{(v)} \in \mathbb{R}^{N_v \times D_v}$	complete data for the v-th view
N_v	number of complete instances for the v-th view
$\mathbf{X}_p^{(i,j)}$	paired instances of view i and j
$\mathbf{X}_u^{(i,j)}$	unpaired instances of view i and j
$N_{(i,j)}$	number of instances in $\mathbf{X}_p^{(i,j)}$
d	dimension of embeddings
$\mathbf{H}^{(v)} \in \mathbb{R}^{N_v \times d}$	embedding for the v-th view
$\mathbf{C}^{(v)} \in \mathbb{R}^{K \times d}$	prototype set for the v-th view
$\mathbf{S}^{(i,j)} \in \mathbb{R}^{N_{(i,j)} \times N_{(i,j)}}$	similarity matrix of the paired instances between view i and j
$\mathbf{P}^{(i,j)} \in \mathbb{R}^{K \times K}$	permutation matrix between $\mathbf{C}^{(i)}$ and $\mathbf{C}^{(j)}$
$E_v(\cdot)$	encoder for the v-th view
$D_v(\cdot)$	decoder for the v-th view

*Corresponding author

3. Implementation Details for Clustering

For the proposed CPSPAN, the same autoencoder network structure are adopted for all datasets. Concretely, for the v-th view, the network structure is denoted as $\tilde{\mathbf{X}}^{(v)} \rightarrow \mathrm{FC}_{500} \rightarrow \mathrm{FC}_{500} \rightarrow \mathrm{FC}_{2000} \rightarrow \mathbf{H}^{(v)} \rightarrow \mathrm{FC}_{2000} \rightarrow \mathrm{FC}_{500} \rightarrow \mathrm{FC}_{500} \rightarrow \hat{\mathbf{X}}^{(v)}$. We set the dimensionality of embeddings $\mathbf{H}^{(v)}$ to 10 for all views. ReLU is utilized as the activation function. The training of the model is divided into two stages, namely the pre-training stage and the partial sample and prototype alignment stage. We used Adam as the optimizer for the entire training process. For a fair comparison, the mean values of 10 runs are reported for all datasets. In addition, we implement our approach using the public toolkit of PyTorch 1.7.1 on a desktop with Windows 10 system and RTX 3080 Graphics Processing Units as well as 64GB memory.

4. Additional Experiments

4.1. Comparison with State-of-The-Arts Under Other Missing Rates

In this section, we conduct comparative experiments with the seven state-of-the-art algorithms under the condition that the missing rate are set to 0.2, 0.4, 0.6, 0.8 and 0.9. The experimental results are displayed in Table 2. From these results, we can observe that our proposed CPSPAN can still show advanced performance under other missing rates, especially when the missing rate is 0.8 or even 0.9.

4.2. Structure-filling Method versus Mean-filling Method

In order to verify that our proposed structure embedding filling strategy is more effective than other embedding filling methods, we compare it with the mean-filling strategy when the missing rate is 0.5. From the Table 3, We found that the structure embedding filling strategy is obviously better than the mean embedding filling strategy.

Table 2. The clustering performance comparisons on five benchmark datasets with different missing ratios. The best results are highlighted in bold, while the second best results are marked with underlined numbers.

	Missing rates	0.2			0.4			0.6			0.8			0.9		
	Metrics	ACC	NMI	F-mea	ACC	NMI	F-mea	ACC	NMI	F-mea	ACC	NMI	F-mea	ACC	NMI	F-mea
Caltech101-7	BSV	0.291	0.265	0.304	0.274	0.229	0.290	0.256	0.188	0.274	0.242	0.160	0.263	0.230	0.141	0.260
	PIC	0.715	0.747	0.704	0.619	0.644	0.622	0.785	0.727	0.733	0.764	0.697	0.699	0.756	0.677	0.683
	AWP	0.810	0.720	0.727	0.820	0.713	0.727	0.765	0.691	0.695	0.768	0.698	0.702	0.774	0.698	0.708
	CPM-Nets	0.775	0.645	0.775	0.750	0.631	0.750	0.604	0.481	0.604	0.589	0.468	0.589	0.518	0.426	0.518
	COMPLETER	0.724	0.595	0.724	0.631	0.548	0.631	0.571	0.559	0.596	0.432	0.465	0.484	0.287	0.292	0.343
	DCP	0.655	0.662	0.666	0.451	0.593	0.551	0.466	0.539	0.520	0.160	0.039	0.165	0.181	0.008	0.188
	DSIMVC	0.637	0.560	0.652	0.617	0.528	0.630	0.547	0.460	0.565	0.440	0.354	0.452	0.371	0.291	0.380
	CPSPAN	**0.855**	**0.764**	**0.845**	**0.876**	**0.786**	**0.872**	**0.861**	**0.766**	**0.857**	**0.838**	**0.731**	**0.831**	**0.839**	**0.737**	**0.828**
HandWritten	BSV	0.566	0.586	0.493	0.517	0.535	0.419	0.461	0.477	0.328	0.406	0.423	0.260	0.391	0.398	0.235
	PIC	0.839	0.842	0.791	0.822	0.817	0.760	0.833	0.856	0.815	0.780	0.792	0.751	0.842	0.866	0.825
	AWP	0.659	0.818	0.710	0.778	0.866	0.786	0.773	0.852	0.778	0.826	0.835	0.784	0.828	0.839	0.788
	CPM-Nets	0.863	0.781	0.863	0.780	0.681	0.780	0.673	0.585	0.673	0.635	0.513	0.635	0.565	0.487	0.565
	COMPLETER	0.839	0.755	0.839	0.692	0.723	0.723	0.613	0.644	0.650	0.579	0.618	0.620	0.305	0.346	0.361
	DCP	0.697	0.744	0.725	0.655	0.715	0.685	0.602	0.697	0.649	0.291	0.386	0.332	0.288	0.388	0.333
	DSIMVC	0.777	0.731	0.783	0.752	0.702	0.760	0.709	0.662	0.718	0.589	0.550	0.594	0.410	0.408	0.414
	CPSPAN	**0.938**	**0.882**	**0.938**	**0.940**	**0.884**	**0.940**	**0.931**	**0.873**	**0.931**	**0.922**	**0.858**	**0.921**	**0.919**	**0.854**	**0.919**
ALOI-100	BSV	0.387	0.743	0.239	0.346	0.741	0.237	0.302	0.741	0.236	0.283	0.740	0.236	0.285	0.741	0.237
	PIC	0.699	0.755	0.554	0.711	0.767	0.570	**0.724**	0.776	0.583	**0.698**	0.752	0.549	0.690	0.742	0.532
	AWP	0.682	0.796	**0.693**	0.679	0.784	0.568	0.674	0.757	0.538	0.660	0.737	0.525	0.656	0.726	0.505
	CPM-Nets	0.187	0.457	0.205	0.097	0.290	0.114	0.082	0.242	0.094	0.067	0.213	0.079	0.034	0.112	0.042
	COMPLETER	0.189	0.430	0.212	0.164	0.421	0.207	0.152	0.417	0.201	0.157	0.415	0.196	0.150	0.411	0.192
	DCP	0.246	0.492	0.279	0.241	0.480	0.266	0.207	0.454	0.231	0.207	0.454	0.231	0.200	0.439	0.225
	DSIMVC	0.306	0.599	0.322	0.299	0.572	0.313	0.274	0.542	0.287	0.257	0.517	0.269	0.235	0.496	0.246
	CPSPAN	**0.701**	**0.856**	0.675	**0.680**	**0.848**	**0.659**	0.693	**0.845**	**0.667**	0.690	**0.848**	**0.662**	**0.715**	**0.855**	**0.695**

Table 3. Comparison with mean embedding filling method. We select ACC as the evaluation metric. (The missing rate is 0.5)

	Structure Filling	Mean Filling
Caltech101-7	0.861	0.522
HandWritten	0.936	0.530
ALOI-100	0.709	0.316

4.3. Influence of Embedding Dimension

In the last step of our CPSPAN, we first adopt the structure embedding filling method to concate the embedding of each view, and finally use kmeans to cluster to get the final result. Therefore, in this section, we study the impact of the embedding dimension of each view on the clustering results. To assess the impact of dimensions, we change the dimensionality of the representation in the range of 32, 64, 128, 256. The missing rate is fixed to 0.5. The results in Table 4 verifies that in most cases, too large or too small dimensionality can cause performance degradation. Our analysis shows that small dimensionality will lose information, while too large dimensionality will contain a lot of redundant information.

Table 4. Impact of embedding dimension on clustering results. The best results are highlighted in bold. (The missing rate is 0.5)

Dataset	Dimension	ACC	NMI	F-mea
Caltech101-7	8	0.851	0.754	0.844
	10	0.861	0.775	0.856
	16	**0.895**	**0.810**	**0.892**
	32	0.875	0.787	0.871
	64	0.867	0.787	0.871
	128	0.891	0.809	0.887
HandWritten	8	0.904	0.863	0.901
	10	**0.936**	**0.879**	**0.935**
	16	0.880	0.853	0.875
	32	0.930	0.870	0.930
	64	0.842	0.832	0.835
	128	0.853	0.841	0.844
ALOI-100	8	0.703	0.854	0.678
	10	0.709	**0.864**	0.679
	16	**0.716**	**0.864**	**0.685**
	32	0.680	0.854	0.651
	64	0.692	0.856	0.665
	128	0.707	**0.864**	0.678