MPC: Multi-view Probabilistic Clustering

Junjie Liu[1,2,*], Junlong Liu[2], Shaotian Yan[1,2], Rongxin Jiang[1,4,†],
Xiang Tian[1,4], Boxuan Gu[1,3], Yaowu Chen[1,3], Chen Shen[2], Jianqiang Huang[2]

[1]Zhejiang University [2]Alibaba Cloud Computing Ltd.
[3] Zhejiang University Embedded System Engineering Research Center, Ministry of Education of China
[4] Zhejiang Provincial Key Laboratory for Network Multimedia Technologies
{jumptoliujj, yanshaotian}@gmail.com, {pingwu.ljl, jason.sc, jianqiang.hjq}@alibaba-inc.com

Abstract

Despite the promising progress having been made, the two challenges of multi-view clustering (MVC) are still waiting for better solutions: i) Most existing methods are either not qualified or require additional steps for incomplete multi-view clustering and ii) noise or outliers might significantly degrade the overall clustering performance. In this paper, we propose a novel unified framework for incomplete and complete MVC named multi-view probabilistic clustering (MPC). MPC equivalently transforms multi-view pairwise posterior matching probability into composition of each view's individual distribution, which tolerates data missing and might extend to any number of views. Then graph-context-aware refinement with path propagation and co-neighbor propagation is used to refine pairwise probability, which alleviates the impact of noise and outliers. Finally, MPC also equivalently transforms probabilistic clustering's objective to avoid complete pairwise computation and adjusts clustering assignments by maximizing joint probability iteratively. Extensive experiments on multiple benchmarks for incomplete and complete MVC show that MPC significantly outperforms previous state-of-the-art methods in both effectiveness and efficiency.

1. Introduction

Multi-view clustering (MVC) [36], a task aiming at exploiting both correlated and complementary information implied in multi-view data and dividing samples into various clusters in an unsupervised manner, has become a hot spot in the area of computer vision, due to its superiority over single-view clustering in performance. With the explosion of multi-source and multi-modal data, a great deal of effort has been put into MVC. Co-EM [23] intends to maximize the mutual agreement across all views by learning knowledge from each other. SwMC [22] constructs a unified similarity graph from multiple views and then partition this graph to obtain the clustering result. GMC [29] weights each data graph matrix to derive the unified graph matrix. SMSC [27] integrates anchor learning and graph construction into a unified framework. MKKM [16] seeks to optimally combine the predefined kernels with matrix-induced regularization in order to improve clustering performance.

Despite the progress having been made, MVC methods still face various challenges: 1) Absence of partial views among data points [17, 35] might frequently take place in practice, while existing methods are either not qualified [27, 29] or require specific additional steps [30, 33] for these cases. 2) K-means [18] and spectral [25] clustering algorithm are commonly employed as the last step of MVC. Both of them are sensitive to the quality of the common representations or similarity matrices, which might be significantly degraded by the noise or outliers contained in multi-view data [11, 28, 42, 43] due to the complexity in data collection. Moreover, the performance of K-means and spectral clustering relies on the selection of total cluster number which is usually unavailable in real world cases.

To address these issues, we propose a novel unified framework for incomplete and complete MVC named multi-view probabilistic clustering (MPC). Instead of learning or calculating a common similarity matrix, we utilize posterior probability to directly reflect the pairwise matching possibility between samples. To obtain the posterior probability matrix, we mathematically decompose it into the formulas of each views' distribution, which exhibits enhanced tolerance to the partial missing of views and has the benefit of easily extending to any number of views. Then, MPC performs graph-context-aware probability refinement with path propagation and co-neighbor propagation, which can effectively alleviate the impact of noise and

[*] This work was done during research intern at Alibaba.
[†] Corresponding authors.

outliers. Finally, clusters are generated using fast probabilistic clustering algorithm, which is more robustness to noise and not requires the prior knowledge of cluster numbers. To avoid complete pairwise computation, we equivalently transforms probabilistic clustering's objective and adjusts clustering assignments by maximizing joint probability iteratively. Extensive experiments demonstrate that MPC significantly outperforms state-of-the-art methods in both clustering performance and computational efficiency.

In summary, the main novelties of this paper are as follows:

- The proposed MPC framework equivalently transforms the multi-view pairwise posterior matching probability into composition of each view's individual distribution, which tolerates data missing and might extend to any number of views.
- The proposed graph-context-aware refinement effectively alleviates the impact of noise and outliers.
- The proposed fast probabilistic clustering algorithm cuts computational complexity by a large margin and does not require any prior knowledge.

2. Related Work

2.1. Multi-view Clustering

Based on the mechanisms and principles used in integrating multiple views, existing MVC algorithms can be grouped into the following categories. The first category is based on graph clustering [22, 29, 30, 43]. As a typical graph clustering method, PIC [30] seeks to learn a consensus representation using a consistent graph matrix constructed from all views and then use spectral clustering algorithm on the learned consensus graph to generate clustering result. The second one is based on matrix factorization [12, 15, 24, 31]. This category seeks to learn a consensus representation by performing low rank matrix factorization on the data matrix to achieve clustering. For example, MIC [24] optimizes a learning consensus matrix based on weighted non-negative matrix factorization and $L_{2,1}$-norm regularization. The third one is multiple kernel learning [32, 39–41]. In brief, this category seeks to find a fused graph using a group of predefined kernels and extract a common cluster structure. For example, OSLF [39] proposes to cluster each independent similarity matrix to learn a consensus clustering partition matrix. Besides, the methods like [4, 14, 38] are based on deep multi-view clustering and MCDCF [4] simultaneously integrates MVC and deep matrix factorization into a unified framework to learn a common consensus representation matrix from the hierarchical information.

We propose a novel method to adaptively estimate the posterior matching probability from multiple views without complicated hyper-parameters fine-tuning. Besides, category information is not required in our method, which severely affects the clustering performance in some methods [30, 39].

2.2. Unsupervised Clustering

K-means clustering [18], spectral clustering [25], hierarchical clustering [26] and some other traditional clustering algorithms [9, 10] are usually used for clustering tasks. K-means [18] minimizes the total intra-cluster variance with a given number of clusters. Spectral [25] performs the graph cut based on the affinity matrix. The clustering performance of these algorithms is affected by the optimization parameters and the number of clusters. As one of effective clustering algorithms, probabilistic clustering algorithms [19, 20] are pioneered to incorporate pairwise relations and have achieved state-of-the-art performance in clustering tasks. The basic idea of probabilistic clustering is to maximize the intra-cluster similarities and minimize the inter-cluster similarities among the objects. Empirical functions are usually used to handle pairwise similarities in these methods, which limits the final clustering performance. Moreover, the matching probability of all pairwise relations are taken into consideration resulting in high computational complexity.

Thus, we propose a fast and parameter-free probabilistic clustering algorithm, which has no optimization parameters and can generate clustering result with a linear computational complexity.

3. The Proposed Method

In this section, we discuss the details of the proposed MPC. As illustrated in Figure 1, the proposed method consists of three phases. First, during the probability estimation phase, given the data matrix of each view, a multi-view pairwise posterior matching probability matrix is generated from the composition of each view's individual distribution using the consistency information and complementary information of all views. We aim to make full use of the information of each view. And then graph-context-aware probability refinement with path propagation and co-neighbor propagation is introduced to refine the pairwise posterior matching probability and alleviate the impact of noise and outliers. In the final phase, fast probabilistic clustering algorithm is introduced to generate clustering result in an efficient and robust way based on the refined multi-view pairwise posterior matching probability matrix.

3.1. Probability Estimation

Given a multi-view dataset of N samples with M views $S = \{V^{(1)}, V^{(2)}, ..., V^{(M)}\}$. $V^{(m)} \in R^{d^{(m)}*N}$ denotes the feature matrix in m-th view, where $d^{(m)}$ is the feature dimension of the m-th view. Let $W^{(m)} \in R^{N*N}$ calculated by $V^{(m)}$ using cosine similarity denotes the similarity

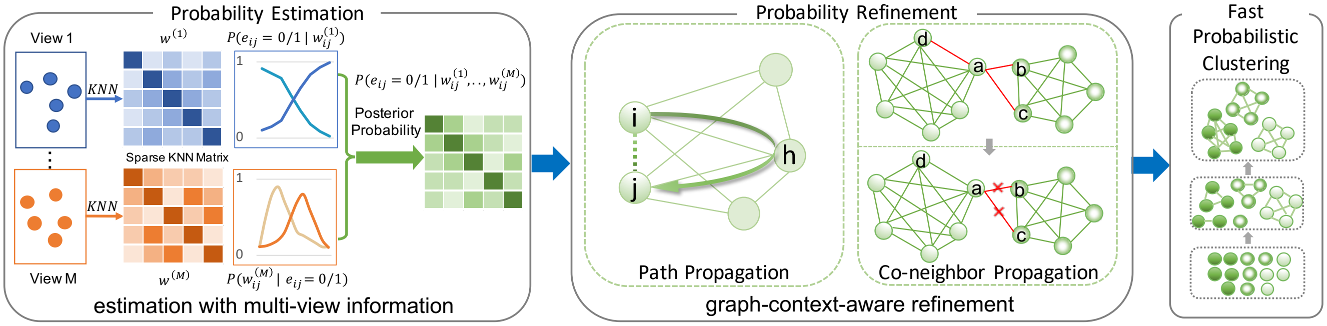
Probability Estimation
View 1
$w^{(1)}$
$P(e_{ij}=0/1 \mid w_{ij}^{(1)})$
KNN
1
0
Sparse KNN Matrix
$P(e_{ij}=0/1 \mid w_{ij}^{(1)},..,w_{ij}^{(M)})$
Posterior Probability
KNN
View M
$w^{(M)}$
$P(w_{ij}^{(M)} \mid e_{ij}=0/1)$
estimation with multi-view information
Probability Refinement
i
j
h
d
a
b
c
Path Propagation
Co-neighbor Propagation
graph-context-aware refinement
Fast Probabilistic Clustering

Figure 1. Overview of the proposed MPC. The proposed method consists of three phases. In the probability estimation phase, a multi-view pairwise posterior matching probability matrix is generated from the composition of each view's individual distribution using the consistency information and complementary information of all views. In the probability refinement phase, path propagation and co-neighbor propagation are introduced to fine-tune the posterior matching probability. As shown in path propagation, taken probability consistency information into consideration, h sets up the probability path between i and j and the probability between i and j can be enhanced by finding the path with the maximum probability. Besides, in co-neighbor propagation, b and c are the noise in k-nearest-neighbors of a. Based on the number of common neighbours and the proportion of the common probabilities, co-neighbor propagation refinement adjusts the probability between a and b and the probability between a and c to a small value and the small value indicates that they are not linked. The probability between a and d can be further adjusted and enhanced. Next, the refined pairwise posterior matching probability is used for clustering. As shown in fast probabilistic clustering procedure, each sample is assigned to its own clustering set at the beginning and each sample is moved to the neighbour clustering set in random seqential order by maximizing joint probability iteratively. Finally, a good clustering result can be generated in a convergent way.

matrix of the m-th view. The similarity matrix of different views may vary from each other even though they generate similar clustering results. Hence, we propose to estimate the pairwise posterior probability based on the similarity matrix of all views instead of simply combining similarity matrices into one common similarity matrix. The pairwise posterior probability of sample i and j can be expressed as:

$$P(i,j) = P(e_{ij}=1|w_{ij}^{(1)}, w_{ij}^{(2)}, ..., w_{ij}^{(M)}) \qquad (1)$$

where $e_{ij}=1$ indicates that the two samples belong to the same class and $w_{ij}^{(m)}$ denotes the similarity of the two samples in m-th view. Assuming that all views are conditionally independent similar to previous works [1, 3, 5, 6, 34], based on Bayesian formula and conditional independence, the above formula can be expressed as:

$$P(i,j) = \frac{(\prod_{m=2}^{M} P(w_{ij}^{(m)}|e_{ij}=1))P(e_{ij}=1|w_{ij}^{(1)})}{\sum_{l\in\{0,1\}}(\prod_{m=2}^{M} P(w_{ij}^{(m)}|e_{ij}=l))P(e_{ij}=l|w_{ij}^{(1)})} \qquad (2)$$

The detailed derivation of Eq. (2) is presented in supplementary material.

As shown in Eq. (2), the similarity information of all views can be considered into the formula. The formula is designed with the following goals. On the one hand, the formula can independently use the similarity information of each view. Accordingly, a larger $P(w_{ij}^{(m)}|e_{ij}=1)$ or $P(e_{ij}=1|w_{ij}^{(m)})$ denotes a larger pairwise probability in the m-th view and can be reflected to the multi-view pairwise probability. On the other hand, the formula can fuse the probability information of all views. When the similarity information of all views is consistent (large $P(w_{ij}^{(m)}|e_{ij}=1)$ and $P(e_{ij}=1|w_{ij}^{(m)})$ in all views or small $P(w_{ij}^{(m)}|e_{ij}=1)$ and $P(e_{ij}=1|w_{ij}^{(m)})$ in all views), the formula can reflect the consistency information. When the similarity information of some views is fuzzy, the formula can reflect the complementary information which is obtained in other views. Different from learning a weighting parameter to fuse multiple similarity matrices in previous works [29, 30], Eq. (2) can adaptively estimate the posterior matching probability from multiple views.

To estimate the $P(w_{ij}^{(m)}|e_{ij}=1/0)$ and $P(e_{ij}=1|w_{ij}^{(m)})$, we use clustering algorithm to generate pseudo labels on each view and generate some pairwise relations using k-nearest-neighbors of each sample. There are many methods that can generate pseudo labels and we use our proposed clustering algorithm. We label these paired samples as 0/1 using pseudo labels to indicate whether the two samples in pairwise relations belong to the same class. Then, we use simple isotonic regression and histogram statistics to estimate $P(w_{ij}^{(m)}|e_{ij}=1/0)$ and $P(e_{ij}=1|w_{ij}^{(m)})$ respectively, which are preprocessing and only need to be estimated once. All observed data of each view is used for calculation, no matter it is a complete view or a missing view. We aim to make full use of the information of each view including the unique infromation in incomplete views and the consistency information and complementary information in complete views. The estimated value using

$P(w_{ij}^{(m)}|e_{ij}=1/0)$ and $P(e_{ij}=1|w_{ij}^{(m)})$ with pseudo labels only need to be approximately correct and it can be discussed from three aspects. First, due to the formula in Eq. (2) composed of each view's individual distribution, the multi-view pairwise posterior matching probability can not be affected by a special view and can adaptively scale the estimation value to eliminate the disturbance and enhance the robustness. Second, the pairwise probability is used for the proposed fast probabilistic clustering algorithm (will be introduced in Section 3.3), and the proposed fast probabilistic clustering algorithm can successfully classify clusters with dense probabilities. Accordingly, the pairwise probability only needs to be able to roughly represent the pairwise relationship. Third, the probability refinement is introduced in next section to further fine-tune the pairwise probability. Given the $P(w_{ij}^{(m)}|e_{ij}=1/0)$ and $P(e_{ij}=1|w_{ij}^{(m)})$ of all views, a multi-view pairwise posterior matching probability matrix can be generated and used to generate clustering result instead of $W^{(m)}$.

3.2. Graph-context-aware Refinement

The probability estimation is calculated based on the aspect of sample relationship, ignoring the aspect of graph context which contains rich information. Thus, we perform graph-context-aware refinement with path propagation and co-neighbor propagation.

Path Propagation. Due to the data perturbation of each view, there exists a few outliers in dataset which may affect the clustering performance in the final step. The probability estimation of outliers can not be calculated accurately by using Eq. (2), we therefore try to fine-tune them with path propagation.

Inspired by the message passing, where the information among nodes is transmissible, the proposed path propagation (**PP**) passes probabilities between samples like follows:

$$P(i,j)=\max\left(P(i,j),P(i,h)\times P(h,j)\right) \tag{3}$$

where $j\in knn_i$, $h\in knn_{ij}$, $knn_i=\{\cup knn_i^m\}$, $knn_j=\{\cup knn_j^m\}$, $knn_{ij}=\{knn_i\cap knn_j\}$ and $knn_i^m\in R^k$ is the k-nearest-neighbors of sample i in m-th view. In Eq. (3), sample h sets up the path between sample i and sample j and the probability between sample i and sample j can be enhanced by finding the path with the maximum probability. Using path propagation, the probability consistency information between the outliers and their neighbors is taken into consideration, in which the outliers can be detected and the pairwise probabilities between the outliers and their neighbors can be enhanced.

Co-neighbor Propagation. The probability estimation is calculated in Euclidean space while the visual features usually lie in low-dimensional manifolds [7]. Only using the information in Euclidean space, overlooking the graph context, may result in inaccuracy of the actual pairwise posterior probabilities between samples. To take advantage of the graph context, the co-neighbor propagation (**CP**) is defined as:

$$P(i,j)=\frac{\sum_{h\in knn_{ij}}(P(i,h)+P(j,h))}{\sum_{h_i\in knn_i}P(i,h_i)+\sum_{h_j\in knn_j}P(j,h_j)} \tag{4}$$

where $knn_i\in R^k$ is the k-nearest-neighbors of sample i calculated by $P(i,j)$ and $knn_{ij}=\{knn_i\cap knn_j\}$. In the formula, the local graph is constructed by the k-nearest-neighbors of two samples. We take both the number of common neighbours and the proportion of the common probabilities into consideration to further refine the probability based on the local graph. As shown in Eq. (4), the available graph-based probability information can be mined to dig out as much manifold-like distribution information as possible. Using co-neighbor propagation, the noise in k-nearest-neighbors can be detected and the outliers can be further enhanced in an efficient way.

3.3. Fast Probabilistic Clustering

In this phase, the fast probabilistic clustering algorithm is introduced to generate clustering result. Given N samples with the clustering set $\pi:[z_1,z_2,...,z_N]$, the optimization goal of fast probabilistic clustering (**FPC**) can be mathematically expressed as:

$$\begin{aligned}\pi_{opt}&=argmax_{\pi}P(X|\pi)=\frac{P(X,\pi)}{P(\pi)}\\ s.t.\ P(X,\pi)&=\frac{\prod_{i,j}(\frac{P(e_{ij}=1)}{P(e_{ij}=0)})^{\delta(z_i,z_j)}P(e_{ij}=0)}{\Omega}\end{aligned} \tag{5}$$

where δ is the Kronecker function and Ω is the normalization parameter. With the above definitions, the objective optimization function $L=-logP(X|\pi)$ can be expressed as:

$$L=\sum_{i,j}(\delta(z_i,z_j)(logP(e_{ij}=0)-logP(e_{ij}=1)))+c \tag{6}$$

where $c=-\sum_{i,j}(logP(e_{ij}=0))-logP(\pi)-log\Omega$ is a constant. Only the probabilities within the class need to be calculated in Eq. (6), which reduces the computational complexity. The whole probabilistic clustering optimization procedure is outlined in Algorithm 1. In the first step, k-nearest-neighbors is constructed using refined multi-view pairwise posterior matching probability. In the second step, each sample is assigned to its own clustering set. Then, in random seqential order, each sample is moved to the neighbour clustering set that results in the minimum value using Eq. (6). The moving procedure is repeated for every sample until no moving steps. The visualization of the probabilistic clustering process is shown in Figure 1. With this algorithm, a good clustering result can be generated in a convergent way.

In incomplete multi-view clustering, we first generate clustering result using the above algorithm on the samples with complete views. And then for incomplete samples, k-nearest-neighbors is constructed on the samples that have the complete views. We utilize co-neighbor propagation to

Algorithm 1: FPC Optimization Procedure

```
Input: $P(e_{ij} = 1)$ and $P(e_{ij} = 0)$;
Construct KNN $nbrs \in R^{n*k}$ by $P(e_{ij} = 1)$;
Initialization: $listn = [1, 2, ..., n]$, $it = 0$,
  $maxiter = 20$, $z = [z_1, z_2, ..., z_n] = [1, 2, .., n]$;
while $it < maxiter$ do
    $count = 0$
    random shuffle $listn$
    for $i$ in $listn$ do
        find $z_{find}$ in $z[nbrs[i]]$ with minimum objective
          value denoted by Eq. (6)
        if $z_i\ != z_{find}$ then
            update $z_i = z_{find}$
            $count = count + 1$
        end
    end
    if $count == 0$ then
        break
    end
    $it = it + 1$
end
Output: $z$;
```

refine the pairwise probability between the incomplete sample and it's k-nearest-neighbors. Finally, we find the maximum probability in k-nearest-neighbors and merge the incomplete sample to the neighbour clustering set. Besides, there still exists some cases in which all incomplete samples have two common views and we can also utilize the complete multi-view clustering procedure to generate the clustering result.

4. Experiments

4.1. Experimental Settings

Datasets. The experimental comparisons are experimentally evaluated on three multi-view datasets. **(1) Handwritten** [8] contains 2000 samples of 10 digits (i.e., digits '0-9'), covering four kinds of features, which are average pixels features, Fourier coefficient features, Zernike moments features and Karhunen-Love coefficient features. **(2) 100Leaves** [21] contains 1600 samples from 100 plant species. For each sample, a shape descriptor and texture histogram are given. **(3) Humbi240**, a subset of Humbi [37] dataset, contains 13440 samples of 240 persons covering face features extracted by face recognition model[1] and body features extracted by person reID model[2]. The datasets are summarized in Table 1. To evaluate the clustering perfomance on incomplete data, we select $c\%$ ($c = 90, 70, 50, 30$) samples as the paired samples that have full views. For the remaining samples, half of them miss the first view, while the second view of the other half is removed. The missing rate is defined as $\eta = 1 - c$.

Table 1. Summary of the datasets. $\{M, C, N, d^{(m)}\}$ denotes the number of {views, clusters, samples, features} in each view, respectively.

Datasets	M	C	N	$d^{(m)}(m = 1, ..., M)$
Handwritten	4	10	2000	240,76,47,64
100Leaves	2	100	1600	64,64
Humbi240	2	240	13440	256,256

Evaluation Metrics. In the experiments, several widely-used clustering metrics including BCubed Fmeasure, Pairwise Fmeasure [2], Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) are used as the evaluation metrics. A higher value of these metrics indicates a better clustering performance.

Implementation Details. We implement our MPC with Python 3.8 and perform all evaluations on a standard Linux OS with 16 2.50GHz Intel Xeon Platinum 8163 CPUs. For all methods, we use an appropriate K to construct k-nearest-neighbors for fair comparisons. And K is set to 200, 20 and 120 on Handwritten, 100Leaves and Humbi240, respectively.

4.2. Compared Methods

We compare our method with SOTA multi-view clustering algorithms. SMSC [27] integrates anchor learning and graph construction into a unified framework. GMC [29] weights each data graph matrix to derive the unified graph matrix. MCDCF [4] brings deep concept factorization to MVC for learning the hierarchical information. SFMC [13] presents a scalable and parameter-free graph fusion framework for MVC. PIC [30] learns the common representation using a fusion graph constructed from incomplete views. OSLF [39] allows the imputation of the base partition matrices to help the learning of the consensus partition matrix. EEIMC [17] proposes using a multi-kernel method to impute the incomplete base clustering matrices. UEAF [33] simultaneously reconstructs the missing views and learns the common representation of multiple views. IMCCP [14] learns representations with contrastive prediction and missing data recovery. 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 download their released codes and tune the hyper-parameters by grid search to generate the best possible results on each dataset. In brief, for PIC, we seek the optimal β from 1e-4 to 1e4 with an interval of 10. For EEIMC, we exploit the "Gauss kernel" to construct the kernel matrices and seek the optimal λ from 2^{-15} to 2^{15} with an interval of 2^3. For OSLF, we exploit the "Gauss kernel" to construct the kernel matrices and seek the optimal λ from 2^{-15} to 2^{15} with an interval of 2. For UEAF, we

[1] https://github.com/XiaohangZhan/face_recognition_framework
[2] https://github.com/layumi/Person_reID_baseline_pytorch

Table 2. The clustering performance comparisons on three datasets. The 1st/2nd best results are indicated in red/blue. MVC indicates complete multi-view clustering; IMVC indicates incomplete multi-view clustering with 0.5 missing rate.

	Methods	Handwritten				100Leaves				Humbi240			
		F_P	F_B	NMI	ARI	F_P	F_B	NMI	ARI	F_P	F_B	NMI	ARI
MVC	MCDCF [4]	54.92	59.32	64.90	49.45	51.04	58.14	82.20	50.52	53.16	67.99	88.91	52.91
	SMSC [27]	67.48	69.20	72.54	63.83	25.88	42.12	72.59	24.77	26.59	44.37	74.09	26.13
	SFMC [13]	72.70	73.72	77.35	69.66	29.97	61.31	80.97	28.94	51.78	91.19	95.47	51.50
	IMCCP [14]	76.56	80.96	83.86	73.73	22.91	36.20	69.94	21.78	49.68	58.43	88.42	49.37
	GMC [29]	74.84	80.47	82.20	71.75	36.40	78.98	88.75	35.47	87.99	96.05	98.57	87.94
	OSLF [39]	78.24	78.55	79.32	75.82	65.55	69.59	87.68	65.20	90.35	93.62	98.20	90.31
	EEIMC [17]	78.86	79.13	80.80	76.51	74.10	77.53	91.18	73.84	91.45	94.45	98.54	91.41
	UEAF [33]	80.61	80.92	81.43	78.46	64.54	72.81	89.18	64.16	86.36	90.36	97.11	86.30
	PIC [30]	76.61	77.88	80.23	73.94	78.04	81.49	92.76	77.82	94.34	96.29	98.95	94.32
	MPC	84.57	84.45	85.60	83.04	84.18	85.65	94.40	84.04	95.49	97.03	99.07	95.47
IMVC	MCDCF [4]	20.84	22.99	25.38	11.38	23.84	30.61	68.36	23.06	29.91	41.78	71.44	29.53
	SMSC [27]	62.83	63.26	65.65	58.65	17.51	30.59	63.26	16.27	18.69	31.59	64.42	18.17
	SFMC [13]	54.81	67.30	71.99	47.53	22.67	51.94	73.81	21.50	7.61	71.73	81.66	6.88
	IMCCP [14]	58.52	71.10	72.68	52.71	17.08	24.75	60.84	15.99	37.20	42.66	80.93	36.84
	GMC [29]	53.56	73.19	73.56	46.05	3.55	47.35	56.76	1.76	2.55	52.86	65.28	1.75
	OSLF [39]	53.86	54.06	58.51	48.73	33.86	39.04	71.84	33.19	70.72	73.40	89.41	70.59
	EEIMC [17]	68.80	69.48	70.26	65.33	52.65	56.74	81.11	52.18	80.94	86.24	94.84	80.86
	UEAF [33]	68.94	69.48	72.55	65.48	38.47	45.87	75.62	37.82	86.04	89.96	96.81	85.98
	PIC [30]	75.65	76.03	76.67	72.95	50.79	55.61	80.72	50.30	83.30	85.74	94.64	83.23
	MPC	77.44	77.65	78.52	75.13	58.31	61.19	83.39	57.94	90.10	91.56	96.53	90.06

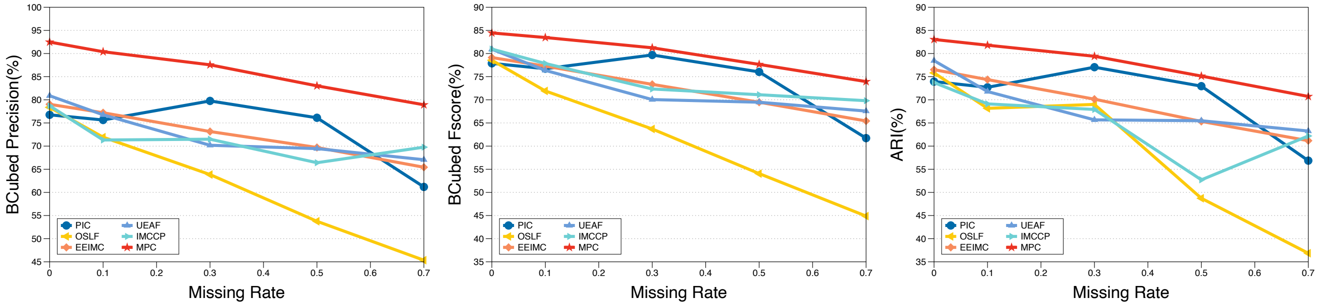

Figure 2. The clustering performance comparisons on Handwritten with different missing rates.

exploit the grid search approach to find the optimal penalty parameters $\lambda_1, \lambda_2, \lambda_3$ from 1e-5 to 1e5 with an interval of 10. For SMSC, we fine-tune the number of anchors.

Performance Comparison with Two Views. Table 2 lists the experimental results of different methods on three datasets. In the complete cases, our method achieves the best performance and surpasses the best baseline by 4.58% on Handwritten, 6.58% on 100Leaves and 1.15% on Humbi240 in terms of ARI. Moreover, in the incomplete cases, our method surpasses the SOTA by 2.18% on Handwritten, 5.76% on 100Leaves and 4.08% on Humbi240 in terms of ARI. GMC and SMFC achieve inferior performance in comparison with the other methods on Humbi240 with 0.5 missing rate. This illustrates that simply filling in the missing views with the average vector is harmful to clustering. Furthermore, the incomplete multi-view clustering performance with different missing rates on Hnadwritten is shown in Figure 2. From these experimental results, we can observe the following points: (1) our method outperforms all the tested baselines with different missing rates, which demonstrates the MPC's capacity of tolerating data missing; (2) our method achieves the best precision which further proves the accuracy of multi-view pairwise posterior matching probability in our proposed MPC.

Performance Comparison with Four Views. For the Handwritten dataset, additional incomplete case is constructed in which all samples have two complete views (the fitst view and the second view) and half of them miss the third view, while the other half of the samples remove the fourth view. As shown in Table 3, MPC significantly outperforms these state-of-the-art methods and MPC surpasses the best baseline by 9.24% and 8.67% in terms of ARI in complete case and incomplete case, respectively. The encouraging performance demonstrates our method's capacity of tolerating data missing and extending to multiple views. Specially compared with complete (first view and second view are complete) performance in Table 2 for OSLF and EEIMC, the clustering performance is unstable and de-

Table 3. The clustering performance comparisons on Handwritten with 4 views. View 1 and view 2 are complete and view 3 and view 4 are 50% missing in the incomplete cases. The 1st/2nd best results are indicated in red/blue.

	Methods	Pairwise Fmeasure			BCubed Fmeasure			NMI	ARI
		Precision	Recall	Fscore	Precision	Recall	Fscore		
Complete	OSLF [39]	76.23	76.58	76.40	76.28	76.70	76.49	76.51	73.79
	EEIMC [17]	75.33	76.39	75.86	76.53	76.51	76.52	78.28	73.17
	PIC [30]	80.76	80.91	80.84	81.28	81.01	81.14	83.26	78.72
	UEAF [33]	81.59	82.25	81.92	82.57	82.34	82.45	83.00	79.91
	MPC	95.85	85.12	90.17	94.89	85.19	89.78	89.77	89.15
Incomplete	OSLF [39]	62.25	67.05	64.56	64.61	67.21	65.88	69.75	60.48
	EEIMC [17]	73.93	78.60	78.26	78.88	78.71	78.79	79.53	75.85
	PIC [30]	77.24	79.72	78.46	78.83	79.82	79.32	81.34	76.04
	UEAF [33]	81.31	81.77	81.54	81.90	81.86	81.88	82.39	79.49
	MPC	95.42	83.84	89.26	94.09	83.93	88.72	88.70	88.16

crease about 15.34% and 0.66% in terms of ARI, respectively. Moreover, the Pairwise precision of our method is about 14% higher than that of UEAF, demonstrating MPC's capacity of multi-view pairwise posterior matching probability in multi-view information excavation.

Table 4. Running time comparison on Handwritten and Humbi240.

Methods	MCDCF [4]	PIC [30]	UEAF [33]	IMCCP [14]	MPC
Handwritten	20min	150s	5h	80s	**45s**
Humbi240	20h	7.5h	288h	280s	**180s**

Table 5. Running time of MPC's components on Handwritten and Humbi240.

Components	PE	PP	CP	FPC	Total
Handwritten	5s	10s	21s	9s	45s
Humbi240	14s	48s	80s	38s	180s

4.3. Computational Complexity Analysis

The computational complexity of MPC is composed of the cost of three phases. In the probability estimation (**PE**) phase, the computational complexity is less than $O(NVK)$, where $K(\ll N)$ is used to generate k-nearest-neighbors and $V(\ll N)$ is the number of views. In the probability refinement (**PP,CP**) phase, the computational complexity is $O(NK)$. According to the fast probabilistic clustering (**FPC**) optimization procedure, the computational complexity is $O(NKL)$, where $L(\ll N)$ denotes the iteration number. Consequently, the computational complexity of our proposed MPC is $O(NK(V+1+L)) = O(NK^*)$ linear to the number of samples, where $K^* \ll N$. For running time comparison in Table 4, the tested baselines cannot balance the clustering performance and computational complexity. For example, the running time of MCDCF is up to 20 hours on Humbi240. And UEAF suffers from a large number of hyper-parameters to be fine-tuned and a higher computational complexity. Compared with these methods, our proposed MPC can achieve a good clustering performance with an appropriate linear running time. The detailed running time of MPC components is shown in the Table 5.

4.4. Ablation Studies

In this section, we conduct some studies on Handwritten and Humbi240 in the following.

Ablation on Probability Estimation. In the probability estimation, we use Eq. (2) to fuse the probability information of each view. In Table 6, we compare the formula with different aggregation functions on Handwriten with two views and four views. And the aggregation function is expressed as: $P(i,j) = Aggregation(P(e_{ij} = 1|w^{(1)}), P(e_{ij} = 1|w^{(2)}), ..., P(e_{ij} = 1|w^{(M)}))$, where aggregation functions include mean, max and min. The aggregation functions treat multiple views as equally important and cannot generate good clustering result. Compared with the naive max function, using the formula in Eq. (2) can significantly boost the ARI from 76.17 to 89.15 on handwritten with four views. It further proves that Eq. (2) can adaptively estimate the posterior matching probability from multiple views. From the perspective of multi-view probability fusion, we compare our method on single view and multi views in Table 7. For single view, we use $P(e_{ij} = 1|w^{(m)})$ as the probability estimation. As shown in Table 7, the performance of probability estimation on single view is about 2% higher than that of origin similarity on single view on Handwritten in terms of ARI. And the performance of multi-view pairwise posterior matching probability based on our method surpasses the best single view clustering performance by 18.20% on Handwritten and 20.68% on Humbi240 in terms of ARI. These experimental results prove that our formula proposed in Eq. (2) can adaptively fuse multi-view probability information in an efficient way, which plays a major role in performance improvement.

Ablation on Refinement Components. The refinement contains two steps: Path Propagation and Co-neighbor Propagation. As we analyze in Section 3.2, two steps are indispensable. In Table 8, we compare the evaluation met-

Table 6. Ablation study of our method. Comparison between the formula and the different aggregation functions on Handwritten.

	Methods	F_P	F_B	NMI	ARI
view 1-2	mean	81.75	81.83	83.99	79.99
	min	80.03	79.74	81.53	78.08
	max	73.46	74.54	79.39	70.88
	formula	**84.57**	**84.45**	**85.60**	**83.04**
view 1-4	mean	86.70	86.65	86.98	85.34
	min	84.30	84.13	84.49	82.71
	max	78.39	79.14	82.49	76.17
	formula	**90.17**	**89.78**	**89.77**	**89.15**

Table 7. The clustering performance comparisons of our method with single view and multiple views on Handwritten and Humbi240. V1 and V2 indicate the origin similarity matrix in the first view and second view; MPC-V1 and MPC-V2 indicate our proposed method in the first view and second view; MPC indicates our proposed method in multiple views.

	Methods	F_P	F_B	NMI	ARI
Handwritten	V1	65.24	65.45	75.65	62.52
	V2	52.92	53.70	64.36	49.04
	MPC-V1	67.48	67.21	76.11	64.84
	MPC-V2	56.36	56.62	66.40	52.64
	MPC	**84.57**	**84.45**	**85.60**	**83.04**
Humbi240	V1	54.72	56.65	87.57	54.61
	V2	53.57	58.07	85.02	53.42
	MPC-V1	74.87	76.52	92.82	74.78
	MPC-V2	63.94	66.99	87.79	63.81
	MPC	**95.49**	**97.03**	**99.07**	**95.47**

Table 8. Ablation study of different refinement components on Handwritten.

Refinement Components	F_P	F_B	NMI	ARI
Only Co-neighbor Propagation	78.72	78.42	81.95	76.75
Only Path Propagation	80.96	80.61	83.68	79.18
Both	**84.57**	**84.45**	**85.60**	**83.04**

Table 9. The clustering performance comparisons of our method with probabilistic clustering algorithm and traditional clustering algorithms on Handwritten and Humbi240. K-means indicates K-means clustering algorithm. Spectral indicates spectral clustering algorithm. FPC indicates our proposed fast probabilistic clustering algorithm.

	Methods	F_P	F_B	NMI	ARI
Handwritten	K-means	76.75	76.89	82.63	74.56
	Spectral	73.76	73.35	81.91	71.45
	FPC	**84.57**	**84.45**	**85.60**	**83.04**
Humbi240	K-means	82.46	85.45	95.66	82.39
	Spectral	90.05	90.77	97.52	90.02
	FPC	**95.49**	**97.03**	**99.07**	**95.47**

rics under different refinement components. As shown in Table 8, clustering result with single refinement component achieves poor performance. Equipped with two refinement components, the clustering performance gains a significant further improvement, demonstrating their efficiency in detecting noise and enhancing outliers.

Ablation on Clustering Methods. FPC is introcuded in MPC to generate clustering result. The number of clusters used for K-means and Spectral clustering is generated by FPC, which is 16 and 320 on Handwritten and Humbi240 respectively. As shown in Table 9, clustering result with K-means and Spectral based on the refined pairwise posterior matching probability achieve poor performance and are severely affected by the number of clusters. Equipped with FPC, the clustering performance gains a significant further improvement, demonstrating FPC's success in clustering.

4.5. Limitations

MPC equivalently transforms multi-view pairwise posterior matching probability into composition of each view's individual distribution and Eq. (2) is proposed based on the assumption that each view is conditionally independent, which is consistent with previous works [1,3,5,6,34]. Based on conditional independence assumption, the individual distribution estimation may be somewhat different from the actual distribution. However, $P(w_{ij}^{(m)}|e_{ij} = 1/0)$ can still roughly represent the distribution in the positive and negative pairwise relations. As we discussed in Section 3.1, the individual distribution estimation only needs to be able to roughly represent the pairwise relationship. Hence, our proposed MPC based on conditional independence assumption can still generate robust performance.

5. Conclusion

In this paper, we propose a novel multi-view probabilistic clustering (MPC) framework to tackle the challenges: i) lack of unified frameworks for incomplete and complete MVC, ii) the performance penalty caused by noise and outliers and iii) the over-complication of existing methods. The proposed MPC equivalently transforms multi-view pairwise posterior matching probability into composition of each view's individual distribution, which tolerates incomplete views. Equipped with graph-context-aware probability refinement and fast probabilistic clustering, MPC exhibits excellent efficiency as well as superior robustness to noise and outliers. Extensive experiments on various multi-view datasets show that our method performs markedly better than SOTA methods while requiring shortest running time, demonstrating the effectiveness of MPC.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2020AAA0103901; in part by the Fundamental Research Funds for the Central Universities; in part by Alibaba Group through Alibaba Research Intern Program; and in part by Alibaba-NTU Singapore Joint Research Institute.

References

[1] Steven Abney. Bootstrapping. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, page 360–367, 2002. 3, 8

[2] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009. 5

[3] Steffen Bickel and Tobias Scheffer. Multi-view clustering. In *ICDM*, volume 4, pages 19–26, 2004. 3, 8

[4] Shuai Chang, Jie Hu, Tianrui Li, Hao Wang, and Bo Peng. Multi-view clustering via deep concept factorization. *Knowledge-Based Systems*, 217:106807, 2021. 2, 5, 6, 7

[5] Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th annual international conference on machine learning*, pages 129–136, 2009. 3, 8

[6] Ning Chen, Jun Zhu, and Eric P Xing. Predictive subspace learning for multi-view data: a large margin approach. In *NIPS*, pages 361–369, 2010. 3, 8

[7] Uri Cohen, SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *Nature communications*, 11(1):1–13, 2020. 4

[8] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. Available: http://archive.ics.uci.edu/ml. 5

[9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996. 2

[10] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007. 2

[11] Abhishek Kumar and Hal Daumé. A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 393–400, 2011. 1

[12] D. D. Lee and H. S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401(7), 1999. 2

[13] Xuelong Li, Han Zhang, Rong Wang, and Feiping Nie. Multi-view clustering: A scalable and parameter-free bipartite graph fusion method. *PAMI*, pages 1–1, 2020. 5, 6

[14] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. Completer: Incomplete multi-view clustering via contrastive prediction. In *CVPR*, pages 11174–11183, June 2021. 2, 5, 6, 7

[15] J. Liu, W. Chi, G. Jing, and J. Han. *Multi-view clustering via joint nonnegative matrix factorization*. Proceedings of the 2013 SIAM International Conference on Data Mining, 2013. 2

[16] Xinwang Liu, Yong Dou, Jianping Yin, Lei Wang, and En Zhu. Multiple kernel k-means clustering with matrix-induced regularization. In *IJCAI*, page 1888–1894, 2016. 1

[17] Xinwang Liu, Miaomiao Li, Chang Tang, Jingyuan Xia, Jian Xiong, Li Liu, Marius Kloft, and En Zhu. Efficient and effective regularized incomplete multi-view clustering. *PAMI*, 43(8):2634–2646, 2021. 1, 5, 6, 7

[18] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 1, 2

[19] Z. Lu and T. K. Leen. Semi-supervised learning with penalized probabilistic clustering. In *NIPS*, pages 849–856, 2004. 2

[20] Zhengdong Lu and Todd K. Leen. Penalized probabilistic clustering. *Neural Computation*, 19(6):1528–1567, 2007. 2

[21] Charles Mallah, James Cope, James Orwell, et al. Plant leaf classification using probabilistic integration of shape, texture and margin features. *Signal Processing, Pattern Recognition and Applications*, 5(1), 2013. 5

[22] Feiping Nie, Jing Li, and Xuelong Li. Self-weighted multiview clustering with multiple graphs. In *IJCAI*, page 2564–2570, 2017. 1, 2

[23] Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, page 86–93, 2000. 1

[24] Weixiang Shao, Lifang He, and Philip S. Yu. Multiple incomplete views clustering via weighted nonnegative matrix factorization with $l_{2,1}$ regularization. In *Machine Learning and Knowledge Discovery in Databases*, pages 318–334, 2015. 2

[25] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000. 1, 2

[26] R. Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 01 1973. 2

[27] Mengjing Sun, Pei Zhang, Siwei Wang, Sihang Zhou, Wenxuan Tu, Xinwang Liu, En Zhu, and Changjian Wang. Scalable multi-view subspace clustering with unified anchors. In *ACMMM*, page 3528–3536, 2021. 1, 5, 6

[28] Grigorios Tzortzis and Aristidis Likas. Kernel-based weighted multi-view clustering. In *2012 IEEE 12th International Conference on Data Mining*, pages 675–684, 2012. 1

[29] Hao Wang, Yan Yang, and Bing Liu. Gmc: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1116–1129, 2020. 1, 2, 3, 5, 6

[30] Hao Wang, Linlin Zong, Bing Liu, Yan Yang, and Wei Zhou. Spectral perturbation meets incomplete multi-view data. In *IJCAI*, pages 3677–3683, 7 2019. 1, 2, 3, 5, 6, 7

[31] Jing Wang, Feng Tian, Hongchuan Yu, Chang Hong Liu, Kun Zhan, and Xiao Wang. Diverse non-negative matrix factorization for multiview data representation. *IEEE Transactions on Cybernetics*, 48(9):2620–2632, 2018. 2

[32] S. Wang, X. Liu, E. Zhu, C. Tang, and J. Yin. Multi-view clustering via late fusion alignment maximization. In *IJCAI*, pages 3778–3784, 7 2019. 2

[33] Jie Wen, Zheng Zhang, Yong Xu, Bob Zhang, Lunke Fei, and Hong Liu. Unified embedding alignment with missing views inferring for incomplete multi-view clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5393–5400, Jul. 2019. 1, 5, 6, 7

[34] Martha White, Yaoliang Yu, Xinhua Zhang, and Dale Schuurmans. Convex multi-view subspace learning. In *NIPS*, pages 1682–1690, 2012. 3, 8

[35] Shuo Xiang, Lei Yuan, Wei Fan, Yalin Wang, Paul M. Thompson, and Jieping Ye. Multi-source learning with block-wise missing data for alzheimer's disease prediction. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 185–193, 2013. 1

[36] Yan Yang and Hao Wang. Multi-view clustering: A survey. *Big Data Mining and Analytics*, 1(2):83–107, 2018. 1

[37] Zhixuan Yu, Jae Shin Yoon, In Kyu Lee, Prashanth Venkatesh, Jaesik Park, Jihun Yu, and Hyun Soo Park. Humbi: A large multiview dataset of human body expressions. In *CVPR*, pages 2987–2997, June 2020. 5

[38] Changqing Zhang, Yeqing Liu, and Huazhu Fu. Ae2-nets: Autoencoder in autoencoder networks. In *CVPR*, pages 2572–2580, 2019. 2

[39] Yi Zhang, Xinwang Liu, Siwei Wang, Jiyuan Liu, Sisi Dai, and En Zhu. One-stage incomplete multi-view clustering via late fusion. In *ACMMM*, page 2717–2725, 2021. 2, 5, 6, 7

[40] B. Zhao, J. T. Kwok, and C. Zhang. Multiple kernel clustering. In *Proceedings of the SIAM International Conference on Data Mining*, pages 638–649, 2009. 2

[41] Sihang Zhou, Xinwang Liu, Miaomiao Li, En Zhu, Li Liu, Changwang Zhang, and Jianping Yin. Multiple kernel clustering with neighbor-kernel subspace segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4):1351–1362, 2020. 2

[42] Xiaofeng Zhu, Xuelong Li, Shichao Zhang, Zongben Xu, Litao Yu, and Can Wang. Graph pca hashing for similarity search. *IEEE Transactions on Multimedia*, 19(9):2033–2044, 2017. 1

[43] Xiaofeng Zhu, Shichao Zhang, Wei He, Rongyao Hu, Cong Lei, and Pengfei Zhu. One-step multi-view spectral clustering. *IEEE Transactions on Knowledge and Data Engineering*, 31(10):2022–2034, 2019. 1, 2