



HAL
open science

Incremental Multi-view Clustering using Barycentric Coordinate Representation

Xiaotong Qian, Guénaél Cabanes, Parisa Rastin, Nistor Grozavu

► **To cite this version:**

Xiaotong Qian, Guénaél Cabanes, Parisa Rastin, Nistor Grozavu. Incremental Multi-view Clustering using Barycentric Coordinate Representation. 2024 International Joint Conference on Neural Networks (IJCNN 2024), Jun 2024, Yokohama, Japan. 10.1109/IJCNN60899.2024.10651114 . hal-04884683

HAL Id: hal-04884683

<https://hal.science/hal-04884683v1>

Submitted on 16 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Incremental Multi-view Clustering using Barycentric Coordinate Representation

1st Xiaotong QIAN
ETIS, UMR 8051

CY Cergy Paris Université
xiaotong.qian@ensea.fr

2nd Guénaël CABANES
LIPN, UMR 7030

Université Sorbonne Paris Nord
LORIA, UMR 7503
Université de Lorraine
guenael.cabanes@lipn.univ-paris13.fr

3rd Parisa RASTIN
LORIA, UMR 7503

Université de Lorraine
parisa.rastin@loria.fr

4th Nistor GROZAVU
ETIS, UMR 8051

CY Cergy Paris Université
nistor.grozavu@cyu.fr

Abstract—Unsupervised learning, particularly clustering, involves systematically categorizing objects based on inherent similarities. As a popular research field of clustering, multi-view data analysis enriches clustering by considering diverse perspectives. Persistent challenges in dynamic multi-view clustering focus on optimizing speed and maintaining quality with an increasing number of views. Motivated by the success of recent researches, we propose in this article an advanced framework of incremental multi-view clustering based on barycentric coordinates representation (IMBC). This framework refines object representation of each view and saves only the first m views as base kernels, then learns a consensus kernel. Updated m base kernels can be obtained via the combination of previous m base kernels and the representation of a new view. The proposed approach also offers the feasibility of incremental multi-modal clustering: its versatility extends to various data types and modalities, enhancing its applicability in diverse scenarios. Through the implementation of various standard clustering algorithms on the consensus kernel, our proposed approaches demonstrate efficient computational processing, achieving comparable clustering performance to existing approaches while incurring lower computational time.

Index Terms—multi-view clustering, incremental multi-view clustering, barycentric coordinate representation

I. INTRODUCTION

Clustering [1] and classification [2] are fundamental concepts in machine learning [3] that both aim to group objects; however, they differ in how the training model incorporates real labels. Classification typically falls under supervised learning [4], where the model groups objects based on predefined labels. In contrast, clustering is a form of unsupervised learning [5], where the model groups different objects based on their physical distance or similarity, relying solely on the characteristics of the objects themselves. In this case, no prior knowledge of real labels is available prior to training. The primary goal of clustering is to group similar objects into the same cluster, while separating dissimilar entities into different clusters. It can be used in various applications, including biological data analysis [6], recommendation systems [7], image processing [8], and more.

This work was funded through the ANR project Pro-Text (project N° ANR-18-CE23-0024-01). More details are available at: <https://pro-text.humnum.fr/le-projet/>

Multi-view clustering [9] is a popular sub-area of clustering that involves examining an object or dataset from multiple perspectives, rather than relying on a single point of view. Individual objects may be characterized by multiple sets of features. Instead of grouping them by a single aspect of observation, considering the complementary information provided by different sets of features has the potential to improve clustering performance. Despite the progress made in multi-view clustering algorithms, there still lacks approaches that address the challenge of dynamic multi-view clustering. In particular, this paper focuses on increasing the speed of the clustering process while maintaining quality in the context of a dynamic increase in the number of views over time.

Motivated by the achievements highlighted in recent research, in particular the work entitled "Incremental Multi-view Spectral Clustering" [10], we present a comparable framework based on the use of the barycentric coordinate (BC) formalism [11]. BC representation offers distinct advantages over alternative representation methods, including computational simplicity, robustness, and stability. The proposed approach integrates object representation while facilitating dynamic learning of a unified representation for current views. It reduces computational complexity in both time and space, overcoming a common problem in dynamic multi-view clustering algorithms. Indeed, since different views correspond to different representation spaces that cannot be directly compared, existing approaches often build a unified representation by computing a similarity matrix between all objects in the views, which can lead to a high level of complexity in terms of time and space, while the new approach creates a much more efficient unified representation space. In addition, since the BC representation depends only on the distance between objects that allows considering multiple similarity measures across different views, thus extending its the applicability of proposed approach to a wide range of data types and modalities.

In the remainder of this paper, we present a brief overview of the barycentric coordinate formalism, clustering, multi-view clustering, and incremental multi-view clustering in Section II, then introduce the framework IMBC and the various proposed approaches in Section III. Finally, We discuss the experimental results in Section IV, before concluding in Section V.

II. RELATED WORK

A. Clustering

Unlike classification, clustering involves grouping objects with the specific definition of similarity or distance between objects varying as needed. Thus, training a clustering model doesn't require real labels. For example, it's possible to group people who are physically close to each other, but it's difficult to summarize the common characteristics of each group using simple descriptions. This could also aid in addressing the challenge of having a restricted set of labels available to describe data in classification models.

Known as one of the most widely used clustering algorithms, KMeans [12] works by iteratively updating k cluster centers using the means of data points within the same cluster, this iterative process ensures continuous refinement and optimization of the cluster assignments. The goal is to minimize the sum of the distances between each point and its associated cluster center. It is simple to implement with low time complexity and its strengths are particularly evident in clusters with spherical shapes.

Meanwhile, Spectral clustering [13] is also famous for well-performed clustering quality, it reduces complex multi-dimensional datasets into k clusters of similar data in rarer dimensions. It does not make strong assumptions about the shape of the clusters, which means it works well on clusters with arbitrary shapes. The detailed structure of spectral clustering could be found in Algo. 1. However, the drawback of this algorithm is that it is relatively slow compared to other clustering algorithms such as KMeans.

Algorithm 1: Spectral clustering algorithm

Data: Dataset $X = \{x_1, \dots, x_n\} \in R^{n \times f}$, k
Result: Clustering result $C \in R^{n \times 1}$
 $S \in R^{n \times n} \leftarrow$ Compute similarity matrix.
 $W \in R^{n \times n} \leftarrow$ Construct sparse adjacency matrix.
 $L \in R^{n \times n} \leftarrow$ Compute Laplacian matrix by using W .
 $\mu \in R^{k \times n} \leftarrow$ Compute the first k eigenvectors of L .
 $C \leftarrow$ KMeans(μ^T, k) to obtain partition result.

KASP [14] is one of the algorithms that accelerates the speed of sampling-based spectral clustering, Algo. 2 shows the structure. This algorithm works by initially applying KMeans to the entire dataset to identify a fixed number ϕ ($\phi > k$) of cluster centers and gather information about the assignment of each data point to its corresponding center. Subsequently, spectral clustering is applied to these cluster centers to group them into k clusters. The final partition matrix for each data point can be determined by identifying the cluster to which its cluster center is assigned. As the number of sub-cluster centers decreases, the complexity of spectral clustering is reduced.

B. Multi-view clustering

Unlike traditional clustering, multi-view clustering groups diverse objects based on multiple sets of features rather than a

Algorithm 2: KASP algorithm

Data: Dataset $X = \{x_1, \dots, x_n\} \in R^{n \times f}$, k , ϕ
Result: Clustering result $C \in R^{n \times 1}$
 $C_K \in R^{n \times 1}, \mu_K \in R^{\phi \times f} \leftarrow$ KMeans(X, ϕ) to obtain partition result and cluster centers.
 $C_S \in R^{\phi \times 1} \leftarrow$ Spectral clustering(μ_K, k).
 $C \leftarrow$ Update C_K by looking at the membership of μ_K in C_S .

single set. These approaches provide more complementary information by incorporating different perspectives. For instance, documents written in various languages, with each language considered as an individual 'view,' could be grouped based on the information provided by different languages. Depending on the modes of collaboration between different views.

Co-training clustering, as described in works such as [15]–[17], involves clustering multi-view data through a co-training strategy, clustering results from different views are combined using prior knowledge or by learning from each other. Through iterative implementation of this strategy, clustering results from all views converge, leading to a comprehensive consensus on the entire dataset. Algorithms employing a co-training style focus on achieving consensus among multiple views, with the goal of maximizing mutual agreement.

Multi-kernel clustering (such as [18]–[20]), originally designed to expand the search space for potential kernel functions and achieve robust generalization, has found extensive application in dealing with multiview data. In this method, predefined kernels corresponding to different views are used and then integrated to improve clustering.

Graphs, or networks, typically represent relationships between objects. In a multi-view context, each graph captures partial data information, assuming a common underlying clustering structure. These graphs reinforce each other's correlations. Multi-view graph-based clustering (such as [21]–[23]) aims to find a fusion graph across all views and applies graph-cut or spectral clustering for the final result.

The key idea of multi-view subspace clustering (such as [24]–[26]) is to learn a common feature representation across all feature subspaces of different views, thus simplifying the clustering process. It involves creating a unified representation for all views, derived from multiple subspaces, to efficiently handle high-dimensional data in clustering models.

C. Incremental multi-view clustering

Although there are many approaches that helps to solve the problem of multi-view clustering, but when it comes to the dynamic aspect, there are few proposed approaches that consider the cluster structure when the number of views increased. However, in real application, this can help to free up memory space during the clustering process, instead of storing all the data information as in static multi-view clustering, it is sufficient to keep only the current model information.

Recently, [10] introduced incremental multi-view spectral clustering (IMSC), that doesn't need to wait for the complete

collection of all views. Instead, it learns consensus clustering results while arriving new views. The basic concept is to define m base kernels (Gaussian kernels) from m initial views, and to obtain a consensus kernel from the first m base kernels. Then, spectral embedding is applied to this consensus kernel to determine a consensus embedding result. When a new view arrives, a new consensus kernel is computed using the m base kernels and the Gaussian kernel of the new view. Then, a consensus spectral embedding is obtained with respect to the spectral embedding of the new consensus kernel and the previous embedding result. The consensus kernel could be used to update the m base kernels for the next incoming view. The clustering result could be obtained by applying KMeans on consensus spectral embedding. To improve efficiency in terms of time and space complexity, instead of computing the Gaussian kernel of each view directly, the authors construct a low-rank approximation of the Gaussian kernels using random Fourier features [27] and perform a low-rank singular value decomposition (SVD) [28] accordingly. The diagram in Fig.1 briefly explains the process.

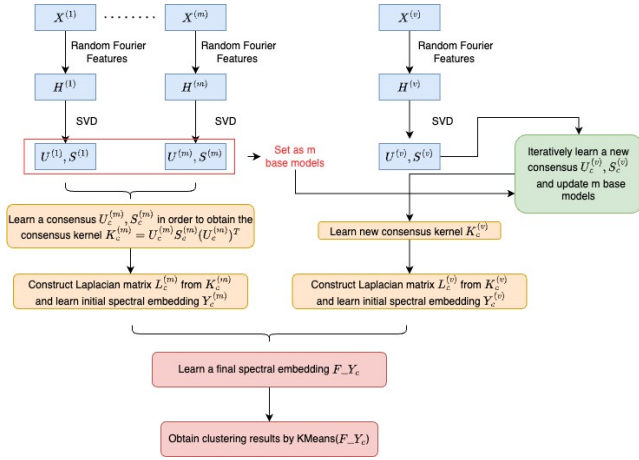


Fig. 1. Diagram IMSC

In 2021, [29] presented an incremental multi-view spectral clustering framework that incorporates Sparse and Connected Graph Learning (SCGL). This method differs from the incremental Gaussian kernel adaptation in IMSC by iteratively updating the sparse similarity matrix upon the arrival of each new view, keeping only a single similarity matrix as base kernel which reduces the complexity in terms of time and memory space. To enhance clustering effectiveness, the model integrates sparse graph learning and connected graph learning, aiming not only to reduce noise but also to preserve accurate connections within clusters. Later in 2022, [30] proposed a continual multi-view clustering (CMVC) approach which works by updating a partition matrix during the incremental process. The process can be outlined as follows: first, a partition matrix is constructed based on the initial view, which is then referred to as the consensus partition matrix. Next, by integrating the incoming new view with the previous consensus partition matrix, a consensus partition matrix is learned. Finally, apply KMeans clustering on this consensus

partition matrix to derive the final clustering results.

D. Barycentric coordinate representation

In geometry, a barycentric coordinate (BC) system [31] is used to specify the location of a point with respect to a simplex. For example, a triangle consists of three vertices (s_1, s_2, s_3) , and for all points $x_i \in X$ in Euclidean space, there exist scalars $\beta_i = [\beta_i^1, \beta_i^2, \beta_i^3]$ that satisfy the relation $x_i = \sum_{j=1}^3 \beta_i^j s_j$ where $\sum_{j=1}^3 \beta_i^j = 1$. And in general, for p -simplexes, $x_i = \sum_{j=1}^p \beta_i^j s_j$ where $\sum_{j=1}^p \beta_i^j = 1$. If there is at least one negative scalar, the point is outside the simplex. Otherwise, if they are all positive, the point is inside. These scalars form a coordinate called the barycentric coordinate (BC) of the corresponding data point, and the vertices are called support points. To ensure that data points are projected into the same barycentric coordinate system, the support points must remain constant for all data points during the projection process. This consistency ensures that the barycentric coordinates of different data points are comparable and can be analyzed in the same coordinate system.

Recent research [11] suggests that the BC representation β_i of a point x_i could be found by following the equation:

$$\beta_i = A^{-1} * M_i, \quad (1)$$

where A is a matrix with size $p \times p$ and $M_i = [d(x_i, s_1) - d(x_i, s_2), \dots, d(x_i, s_1) - d(x_i, s_p), 1]$.

$$A = \begin{pmatrix} d(s_1, s_1) - d(s_2, s_1) & \dots & d(s_1, s_p) - d(s_2, s_p) \\ \vdots & \ddots & \vdots \\ d(s_1, s_1) - d(s_p, s_1) & \dots & d(s_1, s_p) - d(s_p, s_p) \\ 1 & \dots & 1 \end{pmatrix}. \quad (2)$$

As shown in Eq.2, d is a dissimilarity measure that must have properties similar to those of a Euclidean distance.

Since data points are represented by BC, the distances between the data points must be adjusted based on the BC system. This adjustment is crucial for clustering tasks, which are defined as follows:

$$d^2(x_i, x_j) = -\frac{1}{2}(\beta_i - \beta_j)^T * D_s * (\beta_i - \beta_j), \quad (3)$$

where D_s is the dissimilarity matrix between support points.

III. PROPOSED APPROACH

Referring to the framework of recent research, we proposed a similar framework that aims to represent the data points $\{X^{(1)}, \dots, X^{(m)}\}$ of the first m views by the BC representation $\{\beta^{(1)}, \dots, \beta^{(m)}\}$. The m dissimilarity matrices $\{D_s^{(1)}, \dots, D_s^{(m)}\}$ between the support points are computed and stored in memory to be able to compute the distance between points in BC space when needed, and set as base kernels $(\{\beta_b^{(1)}, \dots, \beta_b^{(m)}\}, \{D_b^{(1)}, \dots, D_b^{(m)}\})$. With these m base kernels a consensus kernel $\beta_c^{(m)}$ (or consensus $D_c^{(m)}$) can be computed. When a new view $X^{(v)}$ arrives, computing its BC representation $\beta^{(v)}$ and $D_s^{(v)}$, then compute a consensus kernel of all views $\beta_c^{(v)}$ and $D_c^{(v)}$, updating m base kernels $(\{\beta_b^{(1)}, \dots, \beta_b^{(m)}\}, \{D_b^{(1)}, \dots, D_b^{(m)}\})$ using the new view. Fig. 2 explains the process, and the details of the algorithm

can be found in Algo. 4. The most important thing to know is how to learn a consensus kernel and update m base kernels.

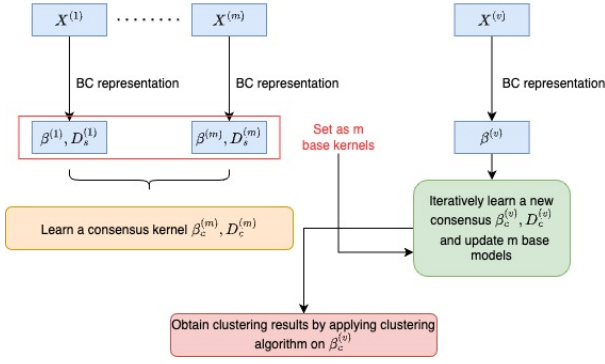


Fig. 2. Diagram IMBC

A. Learn a consensus kernel and update base kernels

To calculate the first initial consensus kernel $\beta_c^{(m)}$ and $D_c^{(m)}$, only need to minimize the following objective function:

$$\min_{\beta_c^{(m)}, D_c^{(m)}} \sum_{i=1}^m d^2(\beta_c^{(m)} - \beta_b^{(i)}) + \sum_{i=1}^m \|D_c^{(m)} - D_s^{(i)}\|_F^2,$$

with partial derivative, the result obtained is :

$$\beta_c^{(m)} = \frac{1}{m} \sum_{i=1}^m \beta_b^{(i)}, D_c^{(m)} = \frac{1}{m} \sum_{i=1}^m D_s^{(i)}, \quad (4)$$

The new consensus kernel $\beta_c^{(v)}$ (or $D_c^{(v)}$) and the updated m base kernels ($\{\beta_b^{(1)}, \dots, \beta_b^{(m)}\}, \{D_b^{(1)}, \dots, D_b^{(m)}\}$) are obtained while arriving a new view by minimizing the following objective function:

$$\begin{aligned} \min_{\substack{\beta_b^{(1)}, \dots, \beta_b^{(m)}, \beta_c^{(v)}, \\ D_b^{(1)}, \dots, D_b^{(m)}, D_c^{(v)}}} & \sum_{i=1}^m d^2(\beta_c^{(v)} - \beta_b^{(i)}) + d^2(\beta_c^{(v)} - \beta^{(v)}) \\ & + \lambda \sum_{i=1}^m d^2(\beta_b^{(i)} - \beta_b^{(i)}) + \sum_{i=1}^m \|D_c^{(v)} - D_b^{(i)}\|_F^2 \\ & + \|D_c^{(v)} - D^{(v)}\|_F^2 + \lambda \sum_{i=1}^m \|D_b^{(i)} - D_b^{(i)}\|_F^2 \end{aligned} \quad (5)$$

where $\|\cdot\|_F^2$ refers to the Frobenius L^2 -norm and as been proposed in [10], λ is a balancing parameter.

- Optimizing $\beta_c^{(v)}$ (resp. $D_b^{(v)}$) by fixing $\beta_b^{(v)}$ (resp. $D_b^{(v)}$).

$$\beta_c^{(v)} = \frac{1}{m+1} \left(\sum_{i=1}^m \beta_b^{(i)} + \beta^{(v)} \right) \quad (6)$$

$$D_c^{(v)} = \frac{1}{m+1} \left(\sum_{i=1}^m D_b^{(i)} + D_s^{(v)} \right)$$

- Optimizing $\beta_b^{(i)}$ (resp. $D_b^{(i)}$) by fixing $\beta_c^{(v)}$ (resp. $D_c^{(v)}$) with $i \in \{1, \dots, m\}$

$$\beta_b^{(i)} = \frac{\lambda \beta_b^{(i)} + \beta_c^{(v)}}{\lambda + 1}, D_b^{(i)} = \frac{\lambda D_b^{(i)} + D_c^{(v)}}{\lambda + 1} \quad (7)$$

The details of this part could be summarized in Algo. 3

Algorithm 3: Learn a consensus kernel and update base kernels

Data: Current base kernels $\beta_b = \{\beta_b^{(1)}, \dots, \beta_b^{(m)}\}$, BC representation of new view $\beta^{(v)}, \lambda$

Result: Consensus kernel $(\beta_c^{(v)}, D_c^{(v)})$, updated base kernels $(\{\beta_b^{(1)}, \dots, \beta_b^{(m)}\}, \{D_b^{(1)}, \dots, D_b^{(m)}\})$

Initialize $\beta_b^{(i)} = \beta_b^{(i)}$ ($1 \leq i \leq m$)

while Not converge **do**

 Calculate $\beta_c^{(v)}$ and $D_b^{(i)}$ by Eq.6

for $t = 1, 2, \dots, m$ **do**

 Update $\beta_b^{(i)}$ and $D_b^{(i)}$ by Eq.7

end

end

Algorithm 4: Framework IMBC

Data: Multi-view dataset $X = \{X^{(1)}, \dots, X^{(v)}\}$, m , k , p , λ

Result: Clustering result $C^{(v)}$ of X

for $t = 1, 2, \dots, m$ **do**

 Construct BC representation $\beta^{(t)}$ and obtain $D_s^{(t)}$ from $X^{(t)}$ using Eq.1.

 Set $(\beta^{(t)}, D_s^{(t)})$ as t -th base kernel $(\beta_b^{(t)}, D_b^{(t)})$.

end

Learn an initial consensus kernel $(\beta_c^{(m)}, D_c^{(m)})$ by Eq.4.

for $t = m+1, m+2, \dots, v$ **do**

 Construct BC representation $\beta^{(t)}$ and obtain $D_s^{(t)}$ from $X^{(t)}$ using Eq.1.

 Learn a consensus kernel $(\beta_c^{(t)}, D_c^{(t)})$ and update m base kernels (details found in Algo.3).

end

Apply different clustering algorithm (adapted to BC representation) on $\beta_c^{(v)}$ to obtain result $C^{(v)}$.

B. Different clustering on consensus kernel $\beta_c^{(v)}$

As been indicated previously, once obtained a consensus kernel $\beta_c^{(t)}$, different clustering algorithms (such as referred in II-A) could be applied on it, only need to modify the algorithms so that they could work on BC system.

BCKMeans as demonstrated in [11], mirrors the KMeans procedure with one important difference. Instead of the conventional distance calculation between each data point and its cluster center, the formula 3 is used.

BCKSpec follows a process similar to spectral clustering. However, when computing the similarity matrix, the pairwise distance between data points, as given by the formula 3, replaces the standard approach.

BCKASP differs from the standard KASP in its approach. Instead of applying KMeans to the entire dataset to derive multiple cluster centers, we randomly select a larger number of data points as cluster centers. Consequently, the distance computation is adjusted using the formula 3.

IV. EXPERIMENTS

In this section, we conduct a comprehensive performance analysis of our proposed framework and relevant approaches on 9 authentic multi-view datasets commonly addressed in multi-view clustering research. The evaluation of clustering performance is based on two external metrics, namely clustering accuracy (ACC) and normalized mutual information (NMI), both of which are crucial for assessing the alignment between predicted clusters and ground-truth labels. A higher metric value indicates better results. In addition, execution time is considered for comprehensive analysis. To ensure the reliability and robustness of our scores, the entire testing process of each approach is repeated 10 times and the mean along with the standard deviation of the evaluation metrics are presented as the final results.

A. Datasets description

The datasets used in our study have been discussed in a variety of previous research and include samples described by different sets of features and generated by different techniques. Detailed descriptions and original sources of these datasets are carefully cited in Tab. I. Before performing any clustering analysis, we first normalize the data for each feature type to ensure that all values fall within the range [0,1].

TABLE I
DATASETS SUMMARY

Real multi-view datasets				
Name	Nb_sampels	Nb_views	Nb_clusters	Nb_features/view
MSRCv1 [32]	210	5	7	[24, 576, 512, 256, 254]
ORL [33]	400	3	40	[4096, 3304, 6750]
100-leaves [34]	1599	3	100	[64, 64, 64]
UCIDigit [35]	2000	6	10	[216, 76, 64, 6, 240, 47]
Caltech101-20 [32]	2386	6	20	[48, 40, 254, 1984, 512, 928]
Out-Scene [36]	2688	4	8	[512, 432, 256, 48]
Reuters [37]	18758	5	6	[24892, 34251, 15506, 11547]
NUSWIDE [38]	30000	5	31	[65, 226, 145, 74, 129]
AwA [39]	30475	6	50	[2690, 2002, 254, 2002, 2002, 2002]

B. Comparison of clustering performance

Our experimental results show our proposed approaches, including IMBCKMeans, IMBCSpec, and IMBCKASP, which integrate adapted clustering algorithms (BCKMeans, BCSpec, BCKASP) discussed in section III-B. These approaches are compared to the existing algorithms IMSC, SCGL, and CMVC. We aim for consistency with the source code of these approaches, minimizing parameter changes and adhering to their described configurations. Recent research [11] indicates that while the number of data points p has a limited impact on quality, it significantly affects time complexity for larger values. Therefore, we limit p to 50, consistent with IMSC. The number of base kernels m is set to 2, following the IMSC approach, and the parameter λ is set to 1, as validated in the IMSC and SCGL source code. For the IMBCKASP approach, we set the number of cluster centers as $5 \times (k + \sqrt{n})$. We uniformly set the number of neighbors for K-Nearest Neighbor to 20 to maintain consistency across approaches. Instead of using multiple kernel KMeans as in CMVC, we opt for standard KMeans to derive a partition matrix, thus ensuring comparability with IMBCKMeans.

TABLE II
EXPERIMENT RESULTS OF DATABASE MSRCV1

Nb_view	Algo	ACC		NMI		Time	
		mean	std	mean	std	mean	std
1	IMSC	0.3605	0.022	0.2552	0.029	0.0264	0.008
	SCGL	0.3514	0.016	0.2194	0.012	0.0534	0.065
	CMVC	0.3638	0.020	0.2585	0.024	0.0074	0.008
	IMBCKMeans	0.3319	0.028	0.2274	0.040	0.0291	0.007
	IMBCSpec	0.3552	0.015	0.2735	0.031	0.1159	0.177
	IMBCKASP	0.3433	0.022	0.2421	0.016	0.0450	0.014
2	IMSC	0.5471	0.041	0.4707	0.040	0.0476	0.004
	SCGL	0.7652	0.074	0.6831	0.044	0.0629	0.017
	CMVC	0.4895	0.058	0.3756	0.062	0.1086	0.034
	IMBCKMeans	0.5457	0.073	0.4905	0.065	0.0268	0.002
	IMBCSpec	0.6638	0.042	0.5958	0.026	0.0419	0.010
	IMBCKASP	0.5905	0.051	0.5146	0.038	0.0380	0.001
3	IMSC	0.6814	0.027	0.5941	0.034	0.5239	0.018
	SCGL	0.7338	0.002	0.6693	0.011	0.0939	0.009
	CMVC	0.6333	0.040	0.5400	0.037	0.2050	0.044
	IMBCKMeans	0.6662	0.061	0.5975	0.047	0.0364	0.019
	IMBCSpec	0.8338	0.022	0.7332	0.031	0.0442	0.005
	IMBCKASP	0.6790	0.048	0.6161	0.044	0.0374	0.001
4	IMSC	0.7181	0.031	0.6280	0.036	0.9345	0.077
	SCGL	0.7633	0.032	0.7079	0.014	0.1320	0.010
	CMVC	0.6586	0.050	0.5616	0.044	0.3477	0.075
	IMBCKMeans	0.5371	0.095	0.5097	0.080	0.0302	0.001
	IMBCSpec	0.8038	0.030	0.7033	0.027	0.0427	0.003
	IMBCKASP	0.7129	0.070	0.6533	0.046	0.0378	0.004
5	IMSC	0.6795	0.041	0.5961	0.047	1.2632	0.036
	SCGL	0.7333	0.000	0.6911	0.000	0.1491	0.007
	CMVC	0.6405	0.067	0.5527	0.052	0.4411	0.071
	IMBCKMeans	0.6338	0.064	0.5813	0.051	0.0346	0.003
	IMBCSpec	0.8200	0.024	0.7221	0.019	0.0429	0.003
	IMBCKASP	0.7200	0.031	0.6304	0.027	0.0373	0.001

TABLE III
EXPERIMENT RESULTS OF DATABASE ORL

Nb_view	Algo	ACC		NMI		Time	
		mean	std	mean	std	mean	std
1	IMSC	0.5440	0.022	0.7265	0.011	0.3336	0.022
	SCGL	0.6318	0.018	0.7919	0.007	0.1119	0.026
	CMVC	0.5148	0.036	0.7165	0.022	0.1013	0.014
	IMBCKMeans	0.4658	0.023	0.6895	0.015	0.2336	0.003
	IMBCSpec	0.5132	0.028	0.7198	0.014	0.1397	0.009
	IMBCKASP	0.3625	0.036	0.5685	0.033	0.1358	0.013
2	IMSC	0.6225	0.039	0.7717	0.018	0.3742	0.022
	SCGL	0.7077	0.024	0.8571	0.008	0.2256	0.009
	CMVC	0.5327	0.034	0.7290	0.018	0.6138	0.023
	IMBCKMeans	0.4822	0.045	0.6831	0.025	0.2399	0.003
	IMBCSpec	0.5488	0.019	0.7347	0.017	0.1515	0.014
	IMBCKASP	0.3848	0.046	0.5798	0.042	0.1364	0.003
3	IMSC	0.6557	0.026	0.7898	0.016	2.0729	0.148
	SCGL	0.6942	0.016	0.8502	0.009	0.3724	0.004
	CMVC	0.5883	0.026	0.7687	0.012	1.2253	0.136
	IMBCKMeans	0.4758	0.035	0.6674	0.021	0.2520	0.005
	IMBCSpec	0.5492	0.023	0.7258	0.014	0.1551	0.005
	IMBCKASP	0.3945	0.038	0.5839	0.029	0.1354	0.005

TABLE IV
EXPERIMENT RESULTS OF DATABASE 100-LEAVES

Nb_view	Algo	ACC		NMI		Time	
		mean	std	mean	std	mean	std
1	IMSC	0.6154	0.013	0.7874	0.008	1.8871	0.194
	SCGL	0.6560	0.009	0.8241	0.004	1.2800	0.099
	CMVC	0.5681	0.022	0.7891	0.007	0.0581	0.004
	IMBCKMeans	0.5402	0.020	0.7748	0.008	2.3002	0.027
	IMBCSpec	0.6038	0.009	0.7942	0.006	1.3782	0.041
	IMBCKASP	0.3572	0.021	0.6248	0.037	1.0630	0.033
2	IMSC	0.7530	0.018	0.8713	0.008	1.5941	0.110
	SCGL	0.5311	0.005	0.7722	0.006	3.2241	0.072
	CMVC	0.4804	0.021	0.7217	0.010	2.5353	0.159
	IMBCKMeans	0.6009	0.026	0.8113	0.010	2.2809	0.034
	IMBCSpec	0.6522	0.011	0.8294	0.005	1.4144	0.130
	IMBCKASP	0.3927	0.012	0.6538	0.021	1.0409	0.018
3	IMSC	0.8592	0.009	0.9162	0.004	8.5384	0.643
	SCGL	0.4529	0.011	0.7064	0.010	5.3927	0.373
	CMVC	0.6124	0.018	0.8036	0.008	5.4342	0.574
	IMBCKMeans	0.6801	0.026	0.8462	0.012	2.2720	0.030
	IMBCSpec	0.7638	0.028	0.8733	0.011	1.3698	0.069
	IMBCKASP	0.4230	0.016	0.6533	0.030	1.0559	0.025

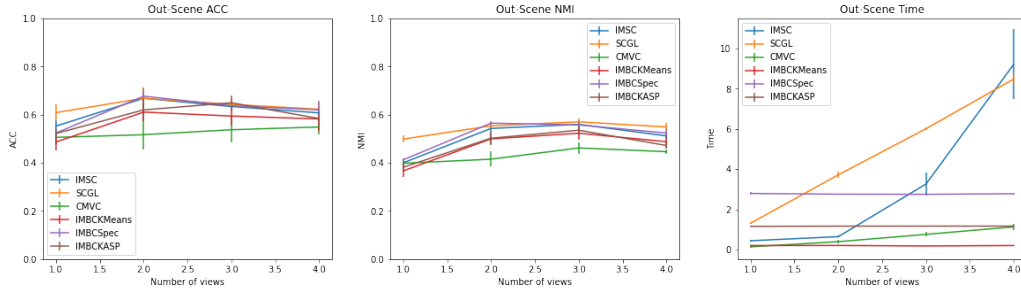


Fig. 3. Experiment results of dataset Out-Scene

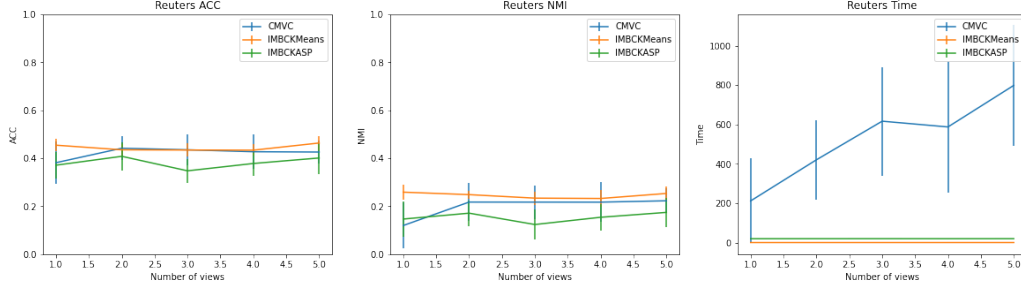


Fig. 4. Experiment results of dataset Reuters

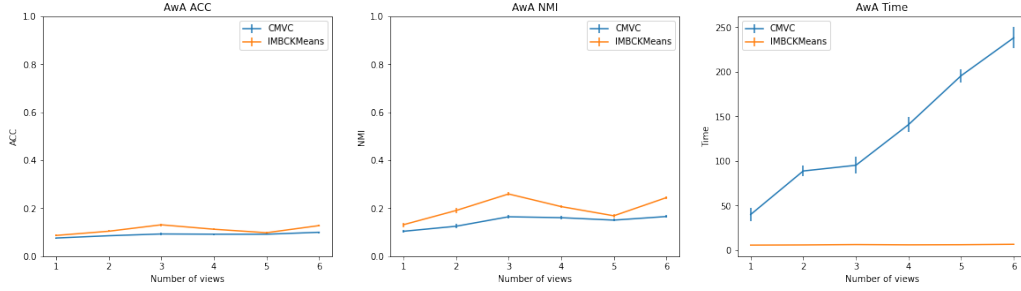


Fig. 5. Experiment results of dataset AWA

the clustering quality evaluated on Reuters dataset (see Fig. VIII) of IMBCKMeans occasionally exceeds that of CMVC.

in terms of clustering quality, as shown in Fig. X. While the

TABLE X
EXPERIMENT RESULTS OF DATABASE AWA

Nb_view	Algo	ACC		NMI		Time	
		mean	std	mean	std	mean	std
1	CMVC	0.0760	0.002	0.1041	0.004	39.8712	7.087
	IMBCKMeans	0.0868	0.003	0.1314	0.009	5.5635	0.287
2	CMVC	0.0856	0.004	0.1250	0.009	88.7627	5.811
	IMBCKMeans	0.1043	0.004	0.1906	0.009	5.7563	0.271
3	CMVC	0.0930	0.004	0.1646	0.009	95.1516	9.310
	IMBCKMeans	0.1311	0.006	0.2597	0.006	6.1558	0.390
4	CMVC	0.0920	0.004	0.1608	0.009	140.7533	8.392
	IMBCKMeans	0.1127	0.003	0.2070	0.007	5.8596	0.107
5	CMVC	0.0919	0.002	0.1506	0.006	195.6748	7.649
	IMBCKMeans	0.0982	0.004	0.1687	0.008	5.9960	0.188
6	CMVC	0.1000	0.003	0.1657	0.006	238.2173	11.870
	IMBCKMeans	0.1279	0.005	0.2445	0.006	6.4769	0.605

Although IMBCKASP has slightly lower performance, it still leads to a lower complexity. Then, we focus on comparing CMVC and IMBCKMeans on larger datasets such as NUSWIDE and AWA, both of which use KMeans-based approaches. IMBCKMeans consistently outperforms CMVC

TABLE XI
ACC RESULTS BY CHANGING DIFFERENT ORDER OF VIEWS

Datasets	Algo	Order 1		Order 2		Order 3	
		mean	std	mean	std	mean	std
MSRCv1	IMBCKMeans	0.6384	0.081	0.5930	0.080	0.6175	0.078
	IMBCKASP	0.6052	0.050	0.5682	0.044	0.5450	0.048
	IMBCKSpec	0.7769	0.032	0.7697	0.025	0.7755	0.030
ORL	IMBCKMeans	0.4876	0.025	0.4860	0.025	0.4833	0.027
	IMBCKASP	0.3995	0.028	0.3831	0.034	0.3870	0.041
	IMBCKSpec	0.5392	0.022	0.5451	0.034	0.5377	0.031
100-leaves	IMBCKMeans	0.6138	0.023	0.5579	0.020	0.6203	0.025
	IMBCKASP	0.3959	0.023	0.3703	0.016	0.3924	0.018
	IMBCKSpec	0.6507	0.013	0.6507	0.017	0.6295	0.014
UCIDigit	IMBCKMeans	0.7008	0.054	0.7414	0.066	0.7170	0.082
	IMBCKASP	0.8162	0.046	0.7611	0.040	0.8434	0.043
	IMBCKSpec	0.8708	0.019	0.7302	0.034	0.8600	0.044
Caltech101-20	IMBCKMeans	0.3408	0.028	0.3519	0.028	0.3705	0.030
	IMBCKASP	0.3288	0.024	0.3405	0.027	0.3455	0.023
	IMBCKSpec	0.3648	0.029	0.3677	0.027	0.3684	0.023
Out-Scene	IMBCKMeans	0.5723	0.033	0.5875	0.042	0.5227	0.035
	IMBCKASP	0.4590	0.030	0.5310	0.025	0.4784	0.031
	IMBCKSpec	0.5461	0.027	0.4984	0.025	0.5365	0.029
Reuters	IMBCKMeans	0.4475	0.024	0.4407	0.020	0.4452	0.029
	IMBCKASP	0.3868	0.061	0.3588	0.051	0.3668	0.053
NUSWIDE	IMBCKMeans	0.1208	0.004	0.1225	0.004	0.1231	0.005
AWA	IMBCKMeans	0.0990	0.003	0.0926	0.003	0.1255	0.004

time required is comparable for a small number of views, there is a significant difference as the number of views increases.

Additionally, we analyzed the effect of varying the order of the views across all datasets by systematically shuffling the views three times. The results in Tab. XI show that clustering quality remains consistent regardless of view order.

V. CONCLUSION

In this paper, we present a novel framework, Incremental Multi-View Clustering using Barycentric Coordinate Representation (IMBC), which facilitates the incremental addition of new views during the learning process. Unlike conventional multi-view methods that aim to achieve clustering on the collection of all views, our approach constructs a unified barycentric coordinate (BC) representation for the current views. Inspired by the IMSC algorithm, IMBC iteratively integrates new views and updates the consensus BC representation without computing full similarity matrices. By applying adapted standard clustering algorithms to the BC system on the final consensus BC representation, our experimental results demonstrate comparable clustering performance to existing methods, while significantly reducing time and memory complexity from quadratic to linear. Future work will explore the application of IMBC in incremental multimodal clustering scenarios, offering a promising approach that goes beyond traditional multi-view clustering by considering different modalities such as text, image, or audio.

REFERENCES

- [1] Xu, D. and Tian, Y., 2015. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2, pp.165-193.
- [2] Cormack, R.M., 1971. A review of classification. *Journal of the Royal Statistical Society: Series A (General)*, 134(3), pp.321-353.
- [3] Jordan, M.I. and Mitchell, T.M., 2015. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), pp.255-260.
- [4] Cunningham, P., Cord, M. and Delany, S.J., 2008. Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval* (pp. 21-49). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [5] Barlow, H.B., 1989. Unsupervised learning. *Neural computation*, 1(3), pp.295-311.
- [6] Madeira, S.C. and Oliveira, A.L., 2004. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM transactions on computational biology and bioinformatics*, 1(1), pp.24-45.
- [7] Ko, H., Lee, S., Park, Y. and Choi, A., 2022. A survey of recommendation systems: recommendation models, techniques, and application fields. *Electronics*, 11(1), p.141.
- [8] Wazarkar, S. and Keshavamurthy, B.N., 2018. A survey on image data analysis through clustering techniques for real world applications. *Journal of Visual Communication and Image Representation*, 55, pp.596-626.
- [9] Yang, Y. and Wang, H., 2018. Multi-view clustering: A survey. *Big Data Mining and Analytics*, 1(2), pp.83-107.
- [10] Zhou, P., Shen, Y.D., Du, L., Ye, F. and Li, X., 2019. Incremental multi-view spectral clustering. *Knowledge-Based Systems*, 174, pp.73-86.
- [11] Rastin, P., Cabanes, G., Matei, B., Bannani, Y. and Marty, J.M., 2019. A new sparse representation learning of complex data: Application to dynamic clustering of web navigation. *Pattern Recognition*, 91, pp.291-307.
- [12] Lloyd, Stuart P. "Least squares quantization in PCM." *Information Theory, IEEE Transactions on* 28.2 (1982): 129-137.
- [13] Von Luxburg, U., 2007. A tutorial on spectral clustering. *Statistics and computing*, 17, pp.395-416.
- [14] Yan, D., Huang, L. and Jordan, M.I., 2009, June. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 907-916).
- [15] Bickel, S. and Scheffer, T., 2004, November. Multi-view clustering. In *ICDM (Vol. 4, No. 2004)*, pp. 19-26.
- [16] Kumar, A. and Daumé, H., 2011. A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 393-400).
- [17] Meng, L., Tan, A.H. and Xu, D., 2013. Semi-supervised heterogeneous fusion for multimedia data co-clustering. *IEEE Transactions on Knowledge and Data Engineering*, 26(9), pp.2293-2306.
- [18] Yu, S., Tranchevent, L., Liu, X., Glanzel, W., Suykens, J.A., De Moor, B. and Moreau, Y., 2011. Optimized data fusion for kernel k-means clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5), pp.1031-1039.
- [19] Tzortzis, G. and Likas, A., 2012, December. Kernel-based weighted multi-view clustering. In *2012 IEEE 12th international conference on data mining* (pp. 675-684). IEEE.
- [20] Zhang, P., Yang, Y., Peng, B. and He, M., 2017, June. Multi-view clustering algorithm based on variable weight and MKL. In *International Joint Conference on Rough Sets* (pp. 599-610). Cham: Springer International Publishing.
- [21] Nie, F., Li, J. and Li, X., 2016, July. Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification. In *IJCAI (Vol. 9)*.
- [22] Nie, F., Li, J. and Li, X., 2017, August. Self-weighted Multiview Clustering with Multiple Graphs. In *IJCAI* (pp. 2564-2570).
- [23] Nie, F., Cai, G. and Li, X., 2017, February. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1)*.
- [24] Cao, X., Zhang, C., Fu, H., Liu, S. and Zhang, H., 2015. Diversity-induced multi-view subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 586-594).
- [25] Zhang, C., Fu, H., Liu, S., Liu, G. and Cao, X., 2015. Low-rank tensor constrained multiview subspace clustering. In *Proceedings of the IEEE international conference on computer vision* (pp. 1582-1590).
- [26] Zhang, C., Hu, Q., Fu, H., Zhu, P. and Cao, X., 2017. Latent multi-view subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4279-4287).
- [27] Rahimi, A. and Recht, B., 2007. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20.
- [28] Stewart, G.W., 1993. On the early history of the singular value decomposition. *SIAM review*, 35(4), pp.551-566.
- [29] Yin, H., Hu, W., Zhang, Z., Lou, J. and Miao, M., 2021. Incremental multi-view spectral clustering with sparse and connected graph learning. *Neural Networks*, 144, pp.260-270.
- [30] Wan, X., Liu, J., Liang, W., Liu, X., Wen, Y. and Zhu, E., 2022, October. Continual multi-view clustering. In *Proceedings of the 30th ACM International Conference on Multimedia* (pp. 3676-3684).
- [31] Hille, E., 2012. *Analytic function theory, Volume I (Vol. 269)*. American Mathematical Soc..
- [32] Cai, X., Nie, F., Huang, H. and Kamangar, F., 2011, June. Heterogeneous image feature integration via multi-modal spectral clustering. In *CVPR 2011* (pp. 1977-1984). IEEE.
- [33] Huang, R., Liu, Q., Lu, H. and Ma, S., 2002, August. Solving the small sample size problem of LDA. In *2002 international conference on pattern recognition (Vol. 3, pp. 29-32)*. IEEE.
- [34] Mallah, C., Cope, J. and Orwell, J., 2013. Plant leaf classification using probabilistic integration of shape, texture and margin features. *Signal Processing, Pattern Recognition and Applications*, 5(1), pp.45-54.
- [35] Duin, Robert. Multiple Features. UCI Machine Learning Repository. <https://doi.org/10.24432/C5HC70>.
- [36] Monadjemi, A., Thomas, B.T. and Mirmehdi, M., 2002. Experiments on high resolution images towards outdoor scene classification.
- [37] Lewis, D.D., Yang, Y., Russell-Rose, T. and Li, F., 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr), pp.361-397.
- [38] Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z. and Zheng, Y., 2009, July. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval* (pp. 1-9).
- [39] Lampert, C.H., Nickisch, H. and Harmeling, S., 2009, June. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 951-958). IEEE.