

Multi-view Clustering using Barycentric Coordinate Representation

1st Xiaotong QIAN
ETIS, UMR 8051
CY Cergy Paris Université
Cergy, France
xiaotong.qian@ensea.fr

2nd Lili JIN
LIPN, UMR 7030
Université Sorbonne Paris Nord
Villetaneuse, France
lili.jin@edu.univ-paris13.fr

3rd Guénaél CABANES
LIPN, UMR 7030
Université Sorbonne Paris Nord
Villetaneuse, France
guenael.cabanes@lipn.univ-paris13.fr

4th Parisa RASTIN
LORIA, UMR 7503
Université de lorraine
Vandoeuvre-lès-Nancy, France
parisa.rastin@loria.fr

5th Nistor GROZAVU
ETIS, UMR 8051
CY Cergy Paris Université
Cergy, France
nistor.grozavu@cyu.fr

Abstract—We consider clustering issues where the available attributes can be divided into various independent groups that frequently offer complimentary information. We concentrate on real-world applications in this paper where a single instance can be represented by a number of heterogeneous features. As was performed successfully in the prior work of clustering on a single view dataset by using barycentric coordinate(BC) representation, and also a recent KMeans-based multi-view clustering RMKMC which proposed that the weights of views can be auto-updated by introducing a hyperparameter γ , we further propose a novel approach of multi-view clustering BCMvlearn by combining these two approaches to reduce complexity without sacrificing clustering quality. In addition, the vector form of the original dataset not being absolutely necessary due to the distance-based property of the BC representation, a variant application of multi-modal clustering is also achievable.

Index Terms—multi-view clustering, barycentric coordinate

I. INTRODUCTION

In the field of unsupervised machine learning, clustering is the process of dividing objects into several groups (clusters) based on their similarities, without prior knowledge of group assignments. The goal of clustering is to group similar objects into the same cluster and separate dissimilar objects into different clusters. The main applications are exploratory data analysis, pattern recognition, and dimensionality reduction [1]. Over the past years, a number of clustering approaches have been proposed (see [2], [3] for a review of the main approaches). Among them, the adaptations of KMeans [4], such as Fuzzy clustering [5], Genetic K-means [6] and the adaptations of spectral clustering [7] such as SSC [8], LRR [9] are very popular. These approaches have proven to be effective in different fields, inspiring researchers to explore the possibility of extending them to a wider range of tasks, such as multi-view data analysis.

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 refers to the problem of grouping objects based on multiple sets of features to integrate these different views to uncover the intrinsic clustering structure [10]. Imagine an object in front of you that can be observed from different perspectives, each providing a different set of information (a view). Another example could be news articles described in multiple languages, where each language represents a different view. By analyzing multi-view data, we can obtain more complementary information about the object and facilitate a better understanding of ensemble diversity [11]. Traditional unsupervised learning, which relies on a single view, may miss out on important information that can be used to optimize the clustering effect through multi-view clustering. Recent surveys in the field of multi-view clustering have noted the emergence of various approaches [12]–[14]. However, despite the advancements in multi-view clustering algorithms, the majority of existing methods struggle with either scalability for large-scale multi-view datasets or limitations in handling a significant number of views.

Inspired by the success of clustering on single-view datasets using the barycentric coordinate representation [15], we propose a novel approach to multi-view clustering by extending this method. The barycentric coordinate formalism simplifies the representation of objects by using weighted averages of a subset of these objects. The weights are computed from the similarities between objects, which uniquely determine the location of the objects within the representation space. This reduces the computational complexity in time and space, a bottleneck of many popular multi-view clustering algorithms. Barycentric coordinates offer several advantages over other forms of point representation, including ease of computation, robustness and stability [16]. In addition, we adapt the framework of RMKMC [17] to automatically update the importance of each view, which is often not addressed in other approaches, or they only use a user-defined weight as a hyperparameter. The proposed approach allows multimodal clustering scenarios

without the need for the original dataset to be in vector form, as the barycentric coordinate formalism only requires similarities between objects. This approach also allows for the consideration of multiple similarity measures across different views, increasing its applicability to a wide range of data types and modalities.

In the remainder of this paper, we introduce the barycentric coordinate formalism and present a short review on multi-view clustering in Section II. We then present the proposed approach, BCMvlearn, in Section III. We detail and discuss the experimental results and application to multi-modal clustering in Section IV, before concluding in Section V.

II. RELATED WORK

A. Barycentric coordinate (BC) representation

Barycentric coordinates [16] of a point in a simplex, such as a triangle or a tetrahedron, are a set of values $(\lambda_1, \dots, \lambda_n)$ that represents the point's relative position within the simplex. These coordinates can be used to define any point within the simplex in terms of a linear combination of the simplex's vertices (P_1, \dots, P_n) . According to a proposal in [15], a representation space can be defined by selecting a subset of P "support points" from a dataset (X) as a virtual space of dimension $P-1$. Any item in the dataset can be expressed using its Barycentric Coordinates (BCs) in this space, represented by the equation:

$$x^n = \sum_{p=1}^P \beta_p^n s^p \text{ with } \sum_{p=1}^P \beta_p = 1. \quad (1)$$

Here, s^p stands for the vector representation of support point p . However, in real-world applications, it may not always be possible to express the data, including the support points, in vector form. Nevertheless, this vector representation is not essential for computing the coordinates β_p^n . Instead, a dissimilarity measure that computes the distance between each data item and the support points can be used. This measure d defines the vector space of the data, whether known or unknown, and must possess properties similar to those of an euclidean distance. In order to obtain the coordinates β_p^n of an object x^n , with respect to the system of support points, considering the following $P \times P$ matrix: $A = (A_{i,j}), 1 \leq i, j \leq P$, which contains the differences between the pairwise dissimilarity of all the support points and the first support point only:

$$A = \begin{pmatrix} d(s^1, s^1) - d(s^2, s^1) & \dots & d(s^1, s^P) - d(s^2, s^P) \\ \dots & \dots & \dots \\ d(s^1, s^1) - d(s^P, s^1) & \dots & d(s^1, s^P) - d(s^P, s^P) \\ 1 & \dots & 1 \end{pmatrix} \quad (2)$$

The barycentric coordinates β^n with size $P \times 1$ is as solution of the following linear system:

$$A * \beta^n = M^n \Rightarrow \beta^n = A^{-1} * M^n \quad (3)$$

$$\text{with } M^n = \begin{pmatrix} d(x^n, s^1) - d(x^n, s^2) \\ \dots \\ d(x^n, s^1) - d(x^n, s^P) \\ 1 \end{pmatrix}$$

Based on this formalism, a relational K-means clustering algorithm using the BC representation has been proposed in [15]. The equation for computing the distance between an object x^n and a cluster center μ^k by using their respective coordinates is as follows:

$$d^2(x^n, \mu^k) = -\frac{1}{2}(\beta^n - \beta^k)^T * D_s * (\beta^n - \beta^k), \quad (4)$$

where D_s is the dissimilarity matrix between the support points. The center updates are computed as the arithmetic mean of the barycentric coordinates of the cluster's members. The experiment conducted on single-view datasets showed that this algorithm has a linear time and memory complexity, without a significant loss of quality when compared to state-of-the-art algorithms.

B. Multi-view clustering

There are several well-known multi-view clustering algorithms, such as the Co-EM based framework for multi-view clustering, mvKMeans [18]. This algorithm adapts the traditional K-Means clustering algorithm to handle two conditionally independent views of a dataset. As a result, multi-view K-Means clustering outperforms its single-view counterpart when applied to a concatenated version of the two views of data. It works by iteratively performing the maximization and expectation steps of a traditional EM algorithm in one view and then using the computed hidden variables as input for the maximization step in another view. Another similar algorithm, mvSphKMeans, also computes the objective optimization and makes assignments using cosine distance rather than Euclidean distance. This method is effective when the various data views are conditionally independent and when the collection contains properties of two different data types. However, it is limited to only two views. The Co-training Method for Multi-View Spectral Clustering, mvSpecClus [19] is a multi-view clustering technique that applies the concept of co-training to multi-view spectral clustering. It uses spectral clustering on different views to obtain the discriminative eigenvectors of each view, which are then used to iteratively modify the graph structure of each view. The goal is to ensure consistency in the relationships between pairs of points across all views. This means that if two points are assigned to the same cluster in one view, they should in the same cluster in all views. A similar version of this method, Co-regularized Multi-View Spectral Clustering, mvCRSpecClus [20] employs a pairwise co-regularization scheme to encourage similarities of examples in the new format to be similar across all perspectives. However, both methods are based on spectral clustering and may result in significant computational time loss.

Previous algorithms do not account for the effect of the view's weight α on the minimization objective function. This motivates the search for multi-view clustering methods that take into account the weight's influence on the clustering results. There are several methods that address this issue, such as Collaborative Fuzzy Clustering [21], where the weight α represents the collaboration between two views, and the goal is to find a structure that is common to all views. It

works by analyzing different subsets and sharing knowledge of local partition matrices. Similarly, Collaborative Non-negative Matrix Factorization (NMF) [22] divides the objective function into two parts. The first part minimizes the approximation error between the original data and the reconstructed matrix based on the cluster structure, and the second part minimizes the distance between the unit of each view of the dataset and the cluster centers of each local NMF view. Fuzzy Clustering in Parallel Universes [23] also aims to use the weight α to denote the contribution of each view, with a smaller value indicating less influence on the objective function minimization. These methods are preferable in real-world applications as they do not treat all views as equally important.

In the previously mentioned methods, the value of α is user-specified, but in unsupervised learning contexts, determining the importance of a view or the degree of collaboration between multiple views can be challenging. To address this issue, the Robust Multi-View K-means Clustering (RMKMC) [17] method has been developed. It automatically updates the weight α and is regulated by a parameter γ to control the weights distribution. The objective function changes when the clustering indicator matrix is used, it now consists of minimizing the weighted sum of the distance between each data item and the cluster center. Given that there are four self-parameters that need to be optimized, the method can determine the optimal value for the final parameter at each iteration until convergence. The outcome shows that it performs well in comparison to other multi-view clustering methods. The proposed BC system-based algorithm aims to achieve an equivalent or better result with less time and memory use by using the framework of this method.

III. PROPOSED APPROACH

We propose a new approach for multi-view clustering using the barycentric clustering (BC) representation. In this method, each data item is represented as a BC (as defined in Equation 1) and the distances between the cluster centers and data items are computed with Equation 2. The objective function for this approach is defined as follows:

$$\begin{aligned} J &= \min_{\mu^{(v)}, G, \alpha^{(v)}} \sum_{v=1}^V (\alpha^{(v)})^\gamma \sum_{n=1}^N d^2((x^n)^{(v)}, \mu^{(v)} G_n^T) \\ &= \min_{(\beta^\mu)^{(v)}, G, \alpha^{(v)}} -\frac{1}{2} \sum_{v=1}^V (\alpha^{(v)})^\gamma H^{(v)}, \end{aligned} \quad (5)$$

where $H^{(v)} = \sum_{n=1}^N \Phi_n^T D_s^{(v)} \Phi_n$ with $\Phi_n = (\beta^n)^{(v)} - (\beta^\mu)^{(v)} G_n^T$ and $\beta^\mu \in R^{P \times K}$ represents the BC of all the centers μ . With respect to that, $\sum_{v=1}^V \alpha^{(v)} = 1$ and $\sum_{k=1}^K G_{nk} = 1$, where $G \in R^{N \times K}$ is a binary cluster indicator matrix which indicate the membership of each data point to a specific cluster. Based on the idea of [17], the optimization of objective function could be divided into 3 parts. At each step, we suppose that the two other parts are already optimized.

A. Optimization

- 1) Updating $(\beta^\mu)^{(v)}$ by calculating the partial derivative of J with respect to $C^{(v)}$, then zeroing the derivative result.

$$(\beta^\mu)^{(v)} = \sum_{n=1}^N (\beta^n)^{(v)} \left(\sum_{n=1}^N G_n^T G_n \right)^{-1} \quad (6)$$

- 2) Updating G , once we obtained $(\beta^\mu)^{(v)}$, we only need to calculate the distance matrix of the weighted sum of each view

$$i = \underset{k \in \{1, \dots, K\}}{\operatorname{argmin}} \left\{ -\frac{1}{2} \sum_{v=1}^V (\alpha^{(v)})^\gamma \Psi_k^T D_s^{(v)} \Psi_k \right\} \quad (7)$$

With $\Psi_k = (\beta^n)^{(v)} - (\beta^\mu)_k^{(v)}$ the difference between data item n and cluster center k in view v , so that $\forall n \in \{1, \dots, N\}$, $G_{ni} = 1$ and $G_{nj} = 0$ where $j \in \{1, \dots, k\}$, $i \neq j$

- 3) Updating $\alpha^{(v)}$, we use Lagrange multiplier λ to solve the following function :

$$\mathcal{L}(\alpha^{(v)}, \lambda) = \sum_{v=1}^V (\alpha^{(v)})^\gamma H^{(v)} - \lambda \left(\sum_{v=1}^V (\alpha^{(v)})^\gamma - 1 \right)$$

Computing the partial derivative of \mathcal{L} with respect to $\alpha^{(v)}$ and λ , then zeroing the derivative result.

$$\alpha^{(v)} = \frac{(\gamma H^{(v)})^{\frac{1}{1-\gamma}}}{\sum_{v=1}^V (\gamma H^{(v)})^{\frac{1}{1-\gamma}}} \quad (8)$$

B. Algorithm

It is now straightforward to implement the proposed approach. The process begins by projecting each data item into the Barycentric Coordinate (BC) subspace, which is created using a set of support points selected from the data. Next, the BC β^k of each cluster center μ^k are initialized randomly. Then, the cluster indicator matrix G is computed and each view is given equal weight. These three components are then iteratively updated through optimization until the objective function converges. A detailed description of the algorithm can be found in Algorithm 1.

To further improve the proposed approach, we incorporate the KMeans++ algorithm [24] which upgrades the K-Means initialization process by strategically selecting initial centers. It selects the first center at random from data items, then chooses subsequent centers using a weighted probability distribution based on the distance to the nearest center already chosen. This process continues until K centers are selected.

In our implementation, we modify the KMeans++ algorithm by using the barycentric distance computation to determine the closest data point. We refer to this modified method as BCmvllearn++. By combining the benefits of the KMeans++ initialization with the barycentric distance computation, we expect to achieve improved clustering results.

Algorithm 1: BCMvlearn

Data: Subsets of patterns $X^1, \dots, X^{(v)}, P, K$
Liste of dissimilarity metrics $[d^1, \dots, d^{(v)}]$
Result: Global cluster indicator matrix G
Choose randomly P support points (always the same data item in each view)
for i in $1, \dots, v$ **do**
 Use $d^{(i)}$ to calculate dissimilarity matrix $O^{(i)}$ between data items and support points.
 Extract dissimilarity matrix between support points $D_s^{(i)}$ from $O^{(i)}$.
 Calculate new representation $\beta^{(i)}$ of $X^{(i)}$ in BC system by (3).
end
Initialize K centers as $\mu^{(v)}$ from $\beta^{(i)}$
Initialize $\alpha^{(i)} = \frac{1}{v}, \forall i \in [1, \dots, v]$.
Set common cluster indicator matrix G by $\mu^{(v)}$.
 $nb_iter \leftarrow 1$
while *Not converge* **do**
 Update cluster centers $\mu^{(i)}$ of each view by (6), $\forall i \in [1, \dots, v]$.
 Update cluster indicator G with help of (7)
 Calculate $H^{(i)}, \forall i \in [1, \dots, v]$.
 Update $\alpha^{(i)}$ by (8), $\forall i \in [1, \dots, v]$.
 $nb_iter \leftarrow nb_iter + 1$
end

IV. EXPERIMENT

To mitigate the impact of redundant variables on the experimental results, we will follow the same setup as outlined in [17]. We standardize the dataset for each type of feature by ensuring that all values fall within a specific range of $[-1, 1]$. To evaluate the quality of the results, we use the Adjusted Rand Index (ARI) [25] and Normalized Mutual Information (NMI) [26] as external indices and the Davies Bouldin index (DBI) [27] and Silhouette index (SI) [28] as internal indices. Additionally, we also consider the time complexity of the algorithms. A higher value of all the metrics, except DBI, indicates a better result. Since the initialization of the algorithms is crucial for their performance, we run each technique 20 times and report the average performance.

A. Dataset description

We used 10 real-world multi-view datasets in our experiments. For a summary of each dataset, refer to Tab. I.

- **MSRCv1** [29]: The 240 images and 9 object classes in the MSRCv1 dataset from Microsoft Research in Cambridge have coarse pixel-wise labels.
- **SensITVehicle** [30]: Dataset collected from package LIB-SVM, which contains 78,823 objects, the first 50 features are acoustic, while the rest are seismic. Extract randomly 300 data items (100 per class) following [17].
- **Movies** [31]: Comprising 617 movies across 17 labels with keywords and actors of the same movies.

- **Caltech101-7/Caltech101-20** [32]: Caltech-101 is composed of image objects belonging to 101 classes, by extracting different features (GABOR, WM, CENTRIST, HOG, GIST, LBP) of images into 6 views. And extract the 7/20 more popular classes for test.
- **100-leaves** [33]: 100 categories of leaves images (16 items per class) are described by 3 features. For each feature, a 64 element vector is given per data item of leaf. These vectors are taken as a contiguous descriptors (for shape) or histograms (for texture and margin).
- **Handwritten** [33]: Handwritten dataset is made up of characteristics of handwritten numbers (from "0" to "9") that were taken from a set of Dutch utility maps. A total of 2,000 patterns have been scanned as binary pictures, 200 patterns per class. There are totally 6 characteristic features, such as FOU, FAC, KAR, PIX, ZER, MOR.
- **Waveform** [33]: 3 classes of waves with 40 attributes, the latter 19 attributes are all noise attributes.
- **Small-NUS/Small-Reuters** : The Reuters [33] dataset includes 18758 items containing features of documents written in English and their translations in 5 languages, across 6 categories. The NUS-WIDE [34] dataset includes 30000 images with six types of low-level features extracted from them. To reduce their size, we use subsets of these datasets already used in a previous study. [35]

TABLE I
DATASETS SUMMARY

Datasets name	Nb data	Nb cluster	Nb view	Nb feature per view
MSRCv1	210	7	5	[24, 576, 512, 256, 254]
SensITVehicle	300	3	2	[50, 50]
Movies	617	17	2	[1878, 1398]
Caltech101-7	1474	7	6	[48, 40, 254, 1984, 512, 928]
Small-Reuters	1500	6	5	[21531, 24892, 34251, 15506, 11547]
100-leaves	1599	100	3	[64, 64, 64]
Handwritten	2000	10	6	[76, 216, 64, 240, 47, 6]
Small-NUS	2000	31	5	[65, 226, 145, 74, 129]
Caltech101-20	2386	20	6	[48, 40, 254, 1984, 512, 928]
Waveform	5000	3	2	[21, 19]

B. Parameter analysis

In our approach, there are two hyper-parameters: the number of support points, denoted as nb_SP , and γ which controls the weight of each view. Due to space limitations, we only present the parameter analysis results of BCMvlearn, which are similar as with BCMvlearn++. In theory, the optimal choice of nb_SP for a single-view dataset is the number of dimensions plus one in the representation space. However, according to a study [15], the internal and external quality indices are not significantly affected by nb_SP on single-view datasets. Based on this result, we conducted a similar experiment on multi-view datasets by varying nb_SP from 10 to 200. As shown in Fig. 1, the influence of the number of support points is similar for both single-view and multi-view datasets: increasing the number of support points usually does not increase significantly the quality of the output, but does increase the computation time. For γ , we varied it from 0.2 to 2, increasing by 0.2 at each step. According to the results

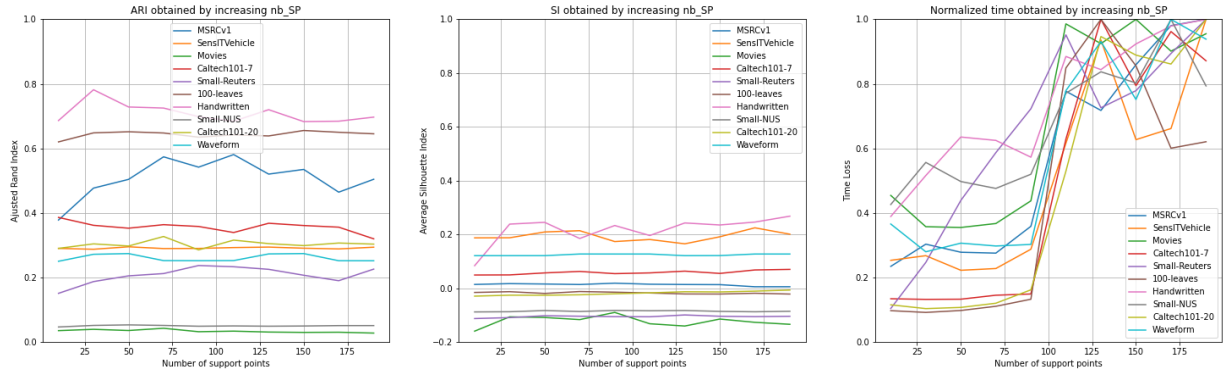


Fig. 1. Effect of the number of support points on the clustering quality and computational time in BCMvlearn.

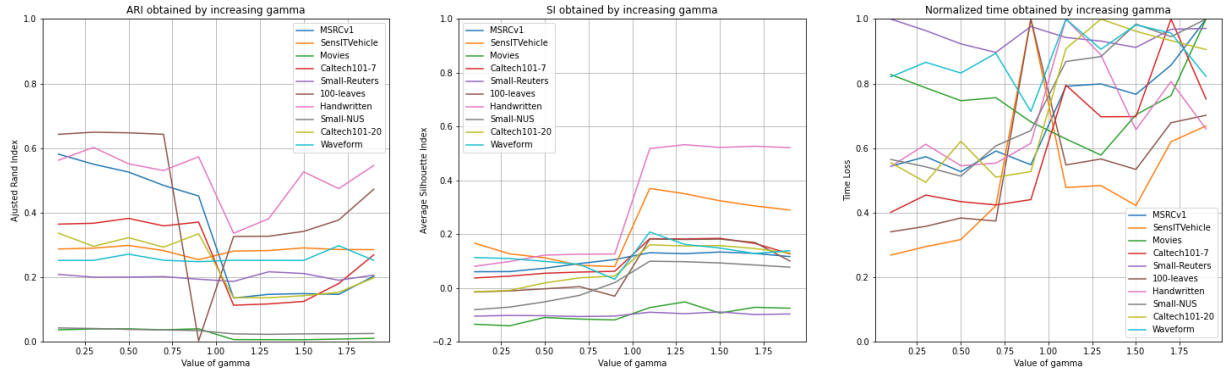


Fig. 2. Effect of the γ values on the clustering quality and computational time in BCMvlearn.

in Fig. 2, we found that when γ varies from 0.9 to 1.1, the external index decreases while the internal index increases. However, there is not much change in other intervals. Considering that the decrease of the external index in most datasets is relatively larger than the increase of the internal index, we set in the following experiments the parameter values to $nb_SP = 30$ and $\gamma = 0.5$ to balance both aspects as much as possible.

C. Clustering Algorithm Performance

In this experiment, we aim to evaluate the overall performance of BCMvlearn and BCMvlearn++ against multi-view clustering algorithms as outlined in Section II-B. Detailed results for each dataset can be found in Tab. II to Tab. VI.

The experimental results show that BCMvlearn and BCMvlearn++ exhibit superior performance to mvKMeans and mvSphKMeans in terms of mean external quality, as shown in Figure 3. In addition, these approaches are comparable to the performance of RMKMC, mvSpecClus and mvCRSpecClus. While BCMvlearn++ may be slightly less efficient than BCMvlearn in terms of external indices, it performs well in terms of average internal quality, as shown in Figure 4. In fact, it outperforms both BCMvlearn and RMKMC in this respect, although it may not be as robust as mvKMeans or mvSphKMeans. However, it is still much better than mvSpecClus and mvCRSpecClus.

Furthermore, the proposed approaches offer significant computational advantages over RMKMC, mvSpecClus and mvCRSpecClus, as shown in Figure 5, due to the BC representation. Although mvKMeans and mvSphKMeans are fast algorithms, they can only handle two views and were therefore only tested on three datasets, unlike the other algorithms. To further assess the performance of the proposed algorithms, we performed a Nemenyi statistical test [36] on the average rank of the effect of each algorithm on different datasets. The results show that BCMvlearn and BCMvlearn++ consistently rank comparable to the other algorithms.

In summary, the experimental results show that the proposed approaches, BCMvlearn and BCMvlearn++, have comparable or superior performance to RMKMC, mvSpecClus and mvCRSpecClus, while offering faster processing times. In addition, they outperform mvKMeans and mvSphKMeans in terms of external indices and are not limited by the number of views. These results suggest that BCMvlearn and BCMvlearn++ could be effective tools for solving multi-view clustering problems.

D. Application on multi-modal clustering

One of the key advantages of our approach, with the BC representation, is its ability is to handle not only vectorial data but also data with unknown vectorial representations such as images, text, and more. As long as the distance between data items can be computed, they can be projected into the BC

TABLE II
ARI RESULTS OF EACH ALGORITHM ON DIFFERENT DATASETS

Dataset	BCmvllearn		BCmvllearn++		RMKMC		mvKMeans		mvSphKMeans		mvSpecClus		mvCRSpecClus	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MSRCv1	0.54	0.05	0.525	0.05	0.5949	0.04	-	-	-	-	0.6216	0.03	0.4591	0.04
SensITVehicle	0.2923	0.01	0.285	0.02	0.3003	0.03	0.3098	0.0	0.3349	0.02	0.2577	0.0	0.3551	0.0
Movies	0.0404	0.01	0.0005	0.0	0.037	0.01	0.001	0.0	0.0009	0.0	0.1143	0.01	0.0771	0.01
Caltech101-7	0.375	0.03	0.4284	0.09	0.3902	0.04	-	-	-	-	0.3212	0.05	0.2737	0.0
Small-Reuters	0.1974	0.04	-0.0007	0.0	0.2362	0.04	-	-	-	-	0.2737	0.01	0.2336	0.0
100-leaves	0.6491	0.03	0.2667	0.11	0.3225	0.13	-	-	-	-	0.7548	0.02	0.713	0.02
Handwritten	0.5908	0.06	0.5494	0.04	0.6838	0.08	-	-	-	-	0.8496	0.0	0.7235	0.0
Small-NUS	0.0372	0.0	0.0366	0.0	0.0435	0.0	-	-	-	-	0.049	0.0	0.0404	0.0
Caltech101-20	0.2938	0.02	0.4	0.05	0.3907	0.05	-	-	-	-	0.2923	0.01	0.2738	0.01
Waveform	0.2658	0.05	0.2515	0.0	0.262	0.04	0.2599	0.01	0.2627	0.01	0.1243	0.0	0.25	0.0
Mean	0.3282	0.03	0.2742	0.04	0.3261	0.05	0.1902	0.0	0.1995	0.01	0.3658	0.01	0.3399	0.01

TABLE III
NMI RESULTS OF EACH ALGORITHM ON DIFFERENT DATASETS

Dataset	BCmvllearn		BCmvllearn++		RMKMC		mvKMeans		mvSphKMeans		mvSpecClus		mvCRSpecClus	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MSRCv1	0.6453	0.04	0.6288	0.04	0.6793	0.03	-	-	-	-	0.7006	0.02	0.5941	0.02
SensITVehicle	0.2966	0.03	0.2868	0.02	0.3047	0.01	0.2963	0.01	0.3281	0.01	0.2497	0.0	0.3325	0.0
Movies	0.1918	0.02	0.0519	0.01	0.1678	0.02	0.0241	0.03	0.0345	0.02	0.2759	0.01	0.253	0.01
Caltech101-7	0.5363	0.02	0.5506	0.04	0.5684	0.03	-	-	-	-	0.5043	0.02	0.4626	0.01
Small-Reuters	0.2581	0.03	0.0058	0.0	0.2922	0.03	-	-	-	-	0.3061	0.01	0.282	0.0
100-leaves	0.8924	0.01	0.7313	0.12	0.7499	0.1	-	-	-	-	0.9251	0.01	0.8983	0.01
Handwritten	0.7025	0.04	0.6774	0.02	0.7686	0.05	-	-	-	-	0.8588	0.0	0.7896	0.0
Small-NUS	0.1826	0.0	0.1787	0.0	0.189	0.0	-	-	-	-	0.19	0.0	0.1753	0.0
Caltech101-20	0.5769	0.01	0.5976	0.01	0.6454	0.02	-	-	-	-	0.6017	0.01	0.559	0.01
Waveform	0.3714	0.03	0.3639	0.0	0.3648	0.01	0.3649	0.0	0.3735	0.0	0.1308	0.0	0.368	0.0
Mean	0.4654	0.02	0.4073	0.03	0.473	0.03	0.2284	0.01	0.2454	0.01	0.4743	0.01	0.4714	0.01

TABLE IV
TIME LOSS OF EACH ALGORITHM ON DIFFERENT DATASETS

Dataset	BCmvllearn		BCmvllearn++		RMKMC		mvKMeans		mvSphKMeans		mvSpecClus		mvCRSpecClus	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MSRCv1	0.3149	0.08	0.3784	0.06	1.2784	0.6	-	-	-	-	3.0422	0.76	0.5734	0.08
SensITVehicle	0.1	0.03	0.1143	0.03	0.2115	0.07	0.0078	0.0	0.0083	0.0	0.5901	0.17	0.1647	0.01
Movies	1.4289	0.32	0.431	0.1	6.0721	1.47	1.6289	0.09	3.0745	1.06	18.4848	5.2	3.2225	0.43
Caltech101-7	4.4249	1.64	5.6106	1.77	21.0537	4.26	-	-	-	-	127.852	5.07	22.0369	0.6
Small-Reuters	9.2546	1.19	9.0282	3.57	1207.1476	243.29	-	-	-	-	944.6798	71.76	1093.8278	62.82
100-leaves	35.6309	5.91	51.3801	23.76	350.3398	573.56	-	-	-	-	145.8762	18.1	37.5597	15.28
Handwritten	15.3088	4.85	19.4362	7.52	17.3569	5.98	-	-	-	-	184.7781	4.43	12.6909	0.19
Small-NUS	35.1765	11.32	37.7945	9.83	73.3589	19.06	-	-	-	-	211.3438	113.3	23.0451	5.76
Caltech101-20	32.944	6.57	35.8318	10.29	135.8155	43.39	-	-	-	-	457.246	17.62	66.9648	4.38
Waveform	2.8044	0.69	2.1922	0.61	10.9507	5.31	0.0403	0.0	0.1669	0.15	394.1557	19.76	37.2147	3.5
Mean	13.7388	3.26	16.2197	5.75	182.3585	89.7	0.559	0.03	1.0832	0.4	248.8049	25.62	129.73	9.3

TABLE V
DBI RESULTS OF EACH ALGORITHM ON DIFFERENT DATASETS

Dataset	BCmvllearn		BCmvllearn++		RMKMC		mvKMeans		mvSphKMeans		mvSpecClus		mvCRSpecClus	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MSRCv1	2.5285	0.15	2.5174	0.13	2.6613	0.11	-	-	-	-	2.7412	0.07	2.8363	0.05
SensITVehicle	3.4565	0.37	3.4609	0.3	3.1228	0.26	2.6621	0.04	2.7684	0.06	6.106	0.0	3.1559	0.04
Movies	5.2645	0.26	0.9252	0.33	7.1921	0.19	2.1065	0.54	2.4949	0.84	6.7152	0.07	7.0752	0.07
Caltech101-7	3.379	0.22	3.3332	0.25	3.4424	0.33	-	-	-	-	3.6727	0.04	3.6878	0.04
Small-Reuters	8.1208	1.05	0.8006	0.05	8.8358	0.8	-	-	-	-	8.6107	0.35	9.1911	0.01
100-leaves	2.7526	0.05	2.0742	0.12	2.8396	0.19	-	-	-	-	2.8385	0.06	3.2164	0.09
Handwritten	2.4176	0.06	2.4273	0.06	2.785	0.09	-	-	-	-	3.2954	0.0	2.8141	0.0
Small-NUS	5.2833	0.17	4.9599	0.23	5.3473	0.16	-	-	-	-	6.1699	0.16	6.7317	0.12
Caltech101-20	3.7223	0.15	3.3995	0.21	3.5603	0.15	-	-	-	-	3.859	0.06	4.0833	0.14
Waveform	29.366	1.79	28.7558	0.11	27.8142	0.81	17.5773	0.47	16.6202	0.28	38.6312	0.42	28.5766	0.05
Mean	6.6291	0.43	5.2654	0.18	6.7601	0.31	7.4486	0.35	7.2945	0.39	8.264	0.12	7.1368	0.06

TABLE VI
SI RESULTS OF EACH ALGORITHM ON DIFFERENT DATASETS

Dataset	BCmlearn		BCmlearn++		RMKMC		mvKMeans		mvSphKMeans		mvSpecClus		mvCRSpecClus	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MSRCv1	0.0718	0.01	0.07	0.01	0.0729	0.01	-	-	-	-	0.0586	0.0	0.0477	0.0
SensITVehicle	0.1158	0.02	0.1126	0.02	0.1508	0.02	0.1558	0.0	0.1207	0.02	0.0969	0.0	0.1185	0.0
Movies	-0.1106	0.06	-0.0583	0.06	-0.0864	0.01	0.1038	0.02	0.0637	0.01	-0.1033	0.01	-0.0889	0.01
Caltech101-7	0.0537	0.01	0.0625	0.02	0.0531	0.01	-	-	-	-	0.0328	0.02	0.0201	0.0
Small-Reuters	-0.1024	0.01	0.0938	0.02	-0.1012	0.0	-	-	-	-	-0.0993	0.0	-0.1007	0.0
100-leaves	-0.0071	0.01	-0.1433	0.09	-0.0186	0.04	-	-	-	-	0.0041	0.0	-0.0196	0.01
Handwritten	0.126	0.01	0.125	0.01	0.1151	0.01	-	-	-	-	0.1123	0.0	0.1079	0.0
Small-NUS	-0.0527	0.0	-0.0695	0.0	-0.0784	0.01	-	-	-	-	-0.0946	0.0	-0.0816	0.0
Caltech101-20	0.0156	0.01	0.0203	0.01	0.0315	0.01	-	-	-	-	-0.0207	0.01	-0.0185	0.0
Waveform	0.0999	0.01	0.1025	0.0	0.1049	0.01	0.1104	0.0	0.1101	0.0	-0.0102	0.0	0.1133	0.0
Mean	0.021	0.02	0.0316	0.02	0.0244	0.01	0.1233	0.01	0.0982	0.01	-0.0023	0.0	0.0098	0.0

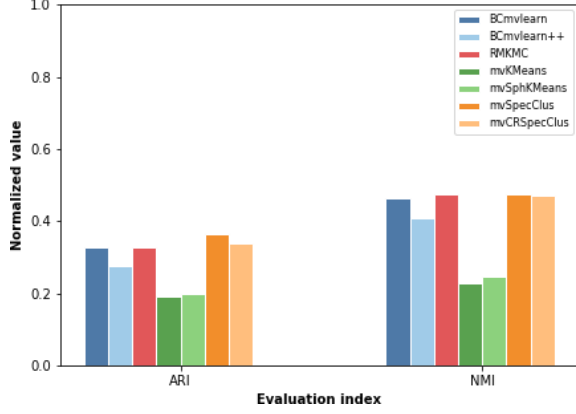


Fig. 3. Mean external quality (ARI and NMI) for each algorithm.

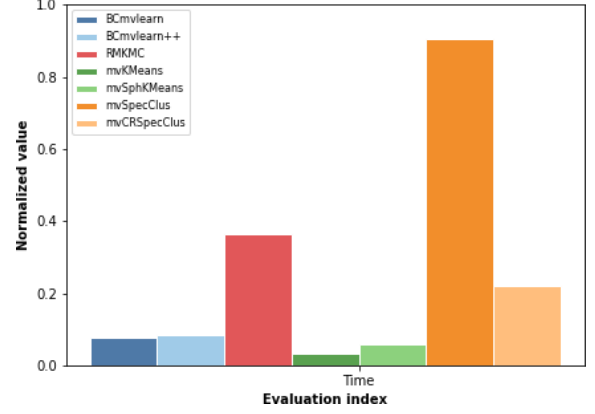


Fig. 5. Mean normalized time loss for each algorithm.

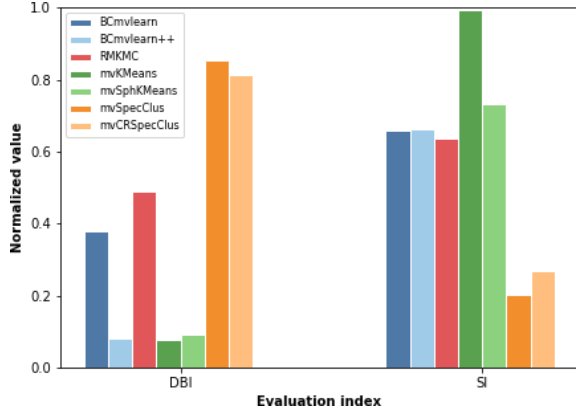


Fig. 4. Mean normalized internal quality (DBI and SI) for each algorithm. Note that a small values of DBI indicates a good clustering result.

space. This flexibility allows our approach to work on multi-modal datasets as well.

As an example of a dataset that can be analysed with our approach, **IEMOCAP** [37] consists of 151 videos of recorded dialogues, with 2 speakers per session for a total of 302 videos across the dataset. Each segment is annotated for the presence of 9 emotions (angry, excited, fear, sad, surprised, frustrated, happy, disappointed and neutral). With different feature extractors, three modalities could be obtained to represent the textual, audio and visual information.

Different distance metrics can be used to compare the data items: the Euclidean distance (ED) for the audio information and the Image Euclidean distance (IMED) for the visual information. For textual information, the cosine distance (CosD) is a popular choice, but it doesn't satisfy the triangle inequality. We therefore propose the use of another distance (CosED) which aims to convert CosD to ED [38], where CosED is equal to $\sqrt{2 - 2\text{CosD}}$.

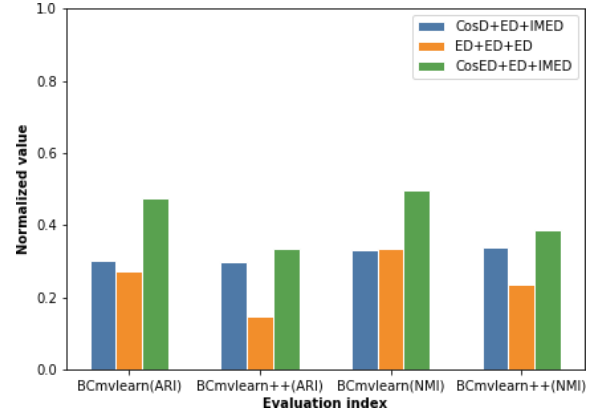


Fig. 6. External quality of BCmlearn and BCmlearn++ on IEMOCAP datasets (multi-modal) by using different combinations of distance metrics.

To showcase the impact of different distance measures, we

applied CosED and CosD on the textual modality and IMED on the images, in addition to using ED on all modalities as a comparison. Fig. 6 shows that our approach performs better on multi-modal datasets when adapted distance metrics are used on different modalities, indicating the effectiveness of our approach with different similarity measures.

V. CONCLUSION

In conclusion, this paper presents a novel multi-view clustering approach called BCMvlearn, which uses the barycentric coordinate representation. By projecting the original data points into a barycentric space and optimising the prototypes of each cluster based on a common cluster indicator, this approach reduces dimensionality and discovers a consensus model, enabling multi-view clustering. The results show that the proposed approach is stable, converges to a solution in a reasonable number of iterations, and is competitive with other state-of-the-art multi-view algorithms in terms of quality and time complexity. The slight discrepancy between the external and internal quality of BCMvlearn and BCMvlearn++ suggests that BCMvlearn++ is more effective for the computation of homogeneous clusters, while BCMvlearn is more suitable when the clustering results need to match external labels. Furthermore, the proposed approach can accommodate different types of distance metrics, making it a promising tool for multi-modal clustering applications. In the future, our goal is to extend the approach to handle dynamic multi-view data streams for even greater real-world applicability.

REFERENCES

- [1] Diday, E. and Simon, J.C., 1976. Clustering analysis. *Digital pattern recognition*, pp.47-94.
- [2] Jain, A.K., Murty, M.N. and Flynn, P.J., 1999. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3), pp.264-323.
- [3] Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O.P., Tiwari, A., Er, M.J., Ding, W. and Lin, C.T., 2017. A review of clustering techniques and developments. *Neurocomputing*, 267, pp.664-681.
- [4] Lloyd, Stuart P. "Least squares quantization in PCM." *Information Theory, IEEE Transactions on* 28.2 (1982): 129-137.
- [5] Dunn, J.C., 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters.
- [6] Krishna, K. and Murty, M.N., 1999. Genetic K-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3), pp.433-439.
- [7] Ng, A., Jordan, M. and Weiss, Y., 2001. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14.
- [8] Elhamifar, E. and Vidal, R., 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11), pp.2765-2781.
- [9] Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y. and Ma, Y., 2012. Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35(1), pp.171-184.
- [10] Chen, M.S., Lin, J.Q., Li, X.L., Liu, B.Y., Wang, C.D., Huang, D. and Lai, J.H., 2022. Representation learning in multi-view clustering: A literature review. *Data Science and Engineering*, 7(3), pp.225-241.
- [11] Luo, H., Kong, F. and Li, Y., 2007. An analysis of diversity measures in clustering ensembles. *CHINESE JOURNAL OF COMPUTERS-CHINESE EDITION-*, 30(8), p.1315.
- [12] Chao, G., Sun, S. and Bi, J., 2017. A survey on multi-view clustering. *arXiv preprint arXiv:1712.06246*.
- [13] Yang, Y. and Wang, H., 2018. Multi-view clustering: A survey. *Big Data Mining and Analytics*, 1(2), pp.83-107.
- [14] Fu, L., Lin, P., Vasilakos, A.V. and Wang, S., 2020. An overview of recent multi-view clustering. *Neurocomputing*, 402, pp.148-161.
- [15] Rastin, P., Cabanes, G., Matei, B., Bennani, 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.
- [16] Hille, E., 2002. *Analytic function theory (Vol. 2)*. American Mathematical Soc.
- [17] Cai, X., Nie, F. and Huang, H., 2013, June. Multi-view k-means clustering on big data. In *Twenty-Third International Joint conference on artificial intelligence*.
- [18] Bickel, S. and Scheffer, T., 2004, November. Multi-view clustering. In *ICDM (Vol. 4, No. 2004)*, pp. 19-26.
- [19] 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).
- [20] Kumar, A., Rai, P. and Daume, H., 2011. Co-regularized multi-view spectral clustering. *Advances in neural information processing systems*, 24.
- [21] Pedrycz, W., 2002. Collaborative fuzzy clustering. *Pattern Recognition Letters*, 23(14), pp.1675-1686.
- [22] Benlamine, K., Grozavu, N., Bennani, Y. and Matei, B., 2019, September. Collaborative non-negative matrix factorization. In *International Conference on Artificial Neural Networks* (pp. 655-666). Springer, Cham.
- [23] Wiswedel, B. and Berthold, M.R., 2007. Fuzzy clustering in parallel universes. *International Journal of Approximate Reasoning*, 45(3), pp.439-454.
- [24] Arthur, D. and Vassilvitskii, S., 2006. k-means++: The advantages of careful seeding. *Stanford*.
- [25] Hubert, L. and Arabie, P., 1985. Comparing partitions. *Journal of classification*, 2, pp.193-218.
- [26] Vinh, N.X., Epps, J. and Bailey, J., 2009, June. Information theoretic measures for clusterings comparison: is a correction for chance necessary?. In *Proceedings of the 26th annual international conference on machine learning* (pp. 1073-1080).
- [27] Davies, D.L. and Bouldin, D.W., 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2), pp.224-227.
- [28] Rousseeuw, P.J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20, pp.53-65.
- [29] Winn, J. and Jojic, N., 2005, October. Locus: Learning object classes with unsupervised segmentation. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 (Vol. 1)*, pp. 756-763). IEEE.
- [30] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011
- [31] Bisson, G. and Grimal, C., 2012, December. Co-clustering of multi-view datasets: a parallelizable approach. In *2012 IEEE 12th International Conference on Data Mining* (pp. 828-833). IEEE.
- [32] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative-Model Based Vision*, 2004.
- [33] Dua, D. and Graff, C. (2019). *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
- [34] 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).
- [35] Liang, Y., Huang, D. and Wang, C.D., 2019, November. Consistency meets inconsistency: A unified graph learning framework for multi-view clustering. In *2019 IEEE International Conference on Data Mining (ICDM)* (pp. 1204-1209). IEEE.
- [36] Hollander, M., Wolfe, D.A. and Chicken, E., 2013. *Nonparametric statistical methods*. John Wiley & Sons.
- [37] Busso, C., Bulut, M., Lee, CC. et al. IEMOCAP: interactive emotional dyadic motion capture database. *Lang Resources & Evaluation* 42, 335–359 (2008).
- [38] Schubert, E., 2021, September. A triangle inequality for cosine similarity. In *International Conference on Similarity Search and Applications* (pp. 32-44). Springer, Cham.