Fusion of Centroid-Based Clustering With Graph Clustering: An Expectation-Maximization-Based Hybrid Clustering

Zekeriya Uykan, *Senior Member, IEEE*

***Abstract*— This article extends the expectation-maximization (EM) formulation for the Gaussian mixture model (GMM) with a novel weighted dissimilarity loss. This extension results in the fusion of two different clustering methods, namely, centroid-based clustering and graph clustering in the same framework in order to leverage their advantages. The fusion of centroid-based clustering and graph clustering results in a simple "soft" asynchronous hybrid clustering method. The proposed algorithm may start as a pure centroid-based clustering algorithm (e.g., *k*-means), and as the time evolves, it may eventually and gradually turn into a pure graph clustering algorithm [e.g., basic greedy asynchronous distributed interference avoidance (GADIA) (Babadi and Tarokh, 2010)] as the algorithm converges and vice versa. The "hard" version of the proposed hybrid algorithm includes the standard Hopfield neural networks (and, thus, Bruck's Ln algorithm by (Bruck, 1990) and the Ising model in statistical mechanics), Babadi and Tarokh's basic GADIA in 2010, and the standard *k*-means (Steinhaus, 1956), (MacQueen, 1967) [i.e., the Lloyd algorithm (Lloyd, 1957, 1982)] as its special cases. We call the "hard version" of the proposed clustering as "hybrid–nongreedy asynchronous clustering (H-NAC)." We apply the H-NAC to various clustering problems using well-known benchmark datasets. The computer simulations confirm the superior performance of the H-NAC compared to the *k*-means clustering, *k*-GADIA, spectral clustering, and a very recent clustering algorithm structured graph learning (SGL) by Kang *et al.* (2021), which represents one of the state-of-the-art clustering algorithms.**

***Index Terms*— Bruck's Ln algorithm, centroid-based clustering, expectation-maximization (EM), greedy asynchronous distributed interference avoidance (GADIA), graph clustering, hybrid–nongreedy asynchronous clustering (H-NAC), *k*-means clustering.**

I. Introduction and Motivation

Clustering is a fundamental mechanism in data processing and machine learning applications, and thus, it is a fundamental research area. It is the task of grouping a set of objects so that objects in the same group are more similar to each other than to those in other groups, and thus, it helps in understanding and discovering the natural grouping in a dataset. The clustering algorithms in the literature can be put into different categories depending on their features and mathematical models, such as centroid-based clustering, graph clustering, spectral clustering, distribution-based clustering, hierarchical clustering, graph-based subspace clustering, and multiview clustering. There is not a magical clustering method that solves all different types of challenging real-life clustering problems with the best performance. Naturally, there are potential shortcomings for all existing clustering techniques depending on their mathematical models and the characteristics of datasets. For example, generally speaking, the *k*-means clustering cannot find nonconvex clusters. Distribution-based clustering suffers from an overfitting problem although it has a strong theoretical foundation. Another prominent method, the Gaussian mixture model (GMM), assumes Gaussian distributions, which is a rather strong assumption for various real-life datasets. The spectral clustering-based methods usually have high time complexity due to the high time cost of eigenvector calculations of a Laplacian matrix [49]–[51]. Eventually, an algorithm that is designed for a specific kind of dataset model may fail on a different type of dataset model in practice. Thus, for a given particular clustering problem and its datasets, we often end up determining the most appropriate clustering algorithm experimentally.

Spectral clustering [50], [51], [53]–[56] aims at exploring the local nonlinear manifold (spectral graph) structure [57], [58] and has attracted great attention in recent years. Spectral clustering methods usually perform eigendecomposition of the similarity matrix first and then use *k*-means to obtain the final clustering results from eigenvectors. Some popular spectral clustering algorithms are the ratio cut [86], normalized cut [50], multiclass spectral clustering [87], spectral embedded clustering [56], constrained Laplacian rank [88], and self-balanced min cut [89], among many other methods [57], [82], [83]. They have shown their power in many applications, such as image clustering [89], image segmentation [87], [90], gene expression data [91], and face recognition [48]. A novel large-scale balanced min cut (LABIN) algorithm LABIN is presented in [81].

Subspace clustering [84] has recently been one of the most common methods used to recover the low-dimensional representations of high-dimensional data with some theoretical guarantees [92], [93]. The subspace clustering has successfully been applied to numerous areas, such as image segmentation, human motion segmentation [94], sequential data clustering [95], medical imaging [96], and bioinformatics [97]. For further references, see [49].

Various authors have, in recent years, reported that, among multiple approaches, graph-based subspace clustering often generates the best performance [98], [99]. As a result,

Manuscript received December 21, 2020; revised July 2, 2021 and September 29, 2021; accepted October 13, 2021.

The author is with the College of Engineering and Technology, American University of the Middle East, Egaila, Kuwait (e-mail: zekeriya.uykan@aum.edu.kw).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2021.3121224.

Digital Object Identifier 10.1109/TNNLS.2021.3121224

graph-based subspace clustering methods have been a hot research topic very recently. Its goal is to find relevant dimensions spanning a subspace for each cluster [84], [85]. Kang *et al.* [46] propose a novel graph-based subspace clustering framework, which is called structured graph learning (SGL). Specifically, the SGL [46] is based on the ideas of anchor points and bipartite graphs. Kang *et al.* [46] establish a theoretical connection between the SGL and the k-means clustering, and present extensive experiments demonstrating the efficiency and effectiveness of the SGL with respect to many state-of-the-art clustering methods in [46]. Wang *et al.* [47] have recently introduced an efficient method for graph clustering, which performs spectral embedding and spectral rotation simultaneously.

Furthermore, the idea of multiview spectral clustering [59], [60] has been another hot research topic in recent years as well due to the fact that combining information from multiview feature spaces generally outperforms the single-view clustering [52]. These views may be obtained from multiple sources or different feature subsets. Multiview spectral clustering examines local manifold structures by seeking eigenvalue–eigenvector decompositions [52]. A great many methods for multiview data clustering have been proposed by a wide spectrum of applications, such as similarity search [100] and human action recognition [101], among others: Some very recent methods are multiview subspace clustering [61], multiview spectral clustering [64], [65], graph-based multiview clustering [62], multiview clustering with graph learning [63], and other multiview spectral clustering algorithms [64]–[70].

Among all the clustering methods mentioned above, there is a very simple yet important algorithm that has been "shining" for more than 50 years: the k-means clustering [3]–[6], [12]: Indeed, if we made to a list of the "top-ten clustering algorithms in the history," most of the researchers and experts would probably give a place to the k-means clustering in this top list. The reason why we say "shining" for the k-means is twofold.

1) The k-means has never lost its popularity since the 1980s.
2) The k-means has taken important roles in not only "old fashion" clustering methods [71] but also the very latest state-of-the-art clustering methods.

Consequently, it would be fair to say that one of the most popular and efficient clustering methods is the k-means method (e.g., [33], [42]). The Lloyd algorithm [5], [6] for the k-means clustering seeks to find k clusters that minimize the sum of squared Euclidean distances between each observation and its respective centroid (cluster mean). In its simplest form, the k-means clustering iteratively alternates between two steps: 1) for a given set of cluster centers, assign each observation to the cluster with the nearest center and 2) for a given assignment of observations to clusters, update each centroid as the sample means of all points in that cluster.

The k-means and its variants have successfully been used in many different areas since the 1980s: market segmentation [40], computer vision (e.g., [37] and [28]), astronomy (e.g., [41]), image segmentation (e.g., [26]–[28] and [37]), computer graphics, social network analysis (e.g., [35] and [36]), social media analysis [32], marketing and sales [38], spam message identification [39], image recognition (e.g., [43]), text document clustering (e.g., [29] and [30]), tweet clustering [31], semantic analysis (e.g., [34]), feature extraction (e.g., [44]), feature selection (e.g., [45]), feature learning step in (semi-) supervised learning or unsupervised learning (e.g., [33]), and so on, just to name a few. In short, the k-means has been one of the fundamental clustering methods in signal processing since the 1980s in spite of its limitations. For further references, see e.g., [13]. Various authors reported that using k-means clustering as the unsupervised learning module in "feature learning" stages can lead to excellent results (e.g., [13] and [14]). A relaxed solution of k-means clustering gives principal component analysis (PCA) [23], [25]. If $k = 2$, then the line connecting the two centroids of the k-means approximates the first PCA eigenvector (i.e., the best 1-D space on which the data projections yield the maximum variance). If $k = 3$, the 2-D plane spanned by the three cluster centroids approximates the plane defined by the first two PCA eigenvectors.

The reasons why the k-means clustering has been so popular in various applications are given as follows.

1) The k-means (Lloyd's algorithm [5], [6]) can be used in many different machine learning problems when we need to choose k different but prototypical samples from a large dataset.
2) The computational cost is low, and it is very fast and easily implemented at an even large scale.
3) It is often used as a preprocessing step (and sometimes as a postprocessing step) for many other algorithms.

On the other hand, the theory of graphs has been well established since the 19th century, and the associated applications from this theory have extended to mathematics, physics, chemistry, linguistics, biology, computer science, and social sciences. Graph clustering has recently broadened its areas of application to social network analysis, big data networks, and biological (protein interaction and gene coexpression) networks, and wireless systems, among others. For the graph clustering (partitioning) theory literature, see [11]. For some recent applications of graph clustering to wireless systems, see [16]–[18].

In this article, we extend the well-known expectation-maximization (EM) formulation for the GMM with a novel weighted dissimilarity loss. This extension yields the fusion of two different clustering methods, namely, centroid-based clustering and graph clustering, in the same framework in order to leverage their advantages. Our investigations on the fusion of centroid-based clustering and graph clustering in this article yield a novel hybrid clustering method, which touches various multidisciplinary subjects, such as Bruck's Ln algorithm [1], the Hopfield neural network (HNN) [7], [8] in the artificial neural networks literature, thus, the Ising model in statistical mechanics, Babadi and Tarokh's pioneering algorithm greedy asynchronous distributed interference avoidance (GADIA) in [2] in wireless communications, the k-means clustering, and graph clustering in machine learning.

HNN [7], [8] has been one of the important milestones in the history of artificial neural networks. HNN's applications varied from associative memory systems design problems to mobile *ad hoc* network routing problems, combinatorial optimization to image restoration, constrained optimization problems to analog-to-digital conversion, system identification to channel allocation in wireless systems, and so on. For references related to the real HNN and complex HNN, see [9] and [10], respectively. Bruck presented the so-called Ln algorithm in his 1990 article [1] in order to show an alternative proof for the convergence of the *discrete-time* real HNN. A recent article [9] has further investigated the behavior of any neuron in not only *discrete-time* HNN but also *continuous-time* HNN when the corresponding energy function is minimized during an optimization process. In this article, we further examine Bruck's Ln algorithm [1] and extend it from a two-cluster case to an arbitrary k-cluster case. This yields a graph clustering (partitioning) algorithm which we will call the k-Ln algorithm in this article due to Bruck's Ln algorithm in [1].

A. Contributions of This Article

In this article, using the well-known EM formulation, we unite the two commonly used but different types of clustering methods, namely, the centroid-based clustering and the graph clustering, in the same framework in order to design a hybrid clustering method with more merits than each of the two. Our investigations yield various novel results; some of which are given as follows.

1) This article extends the EM formulation for the GMM with a novel weighted dissimilarity loss. This extension results in the fusion of two different clustering methods, namely, centroid-based clustering and graph clustering. The proposed algorithm may start as a pure centroid-based clustering algorithm (e.g., k-means), and as the time evolves, it may eventually and gradually turn into a pure graph clustering algorithm (e.g., basic GADIA [2]) as the algorithm converges and vice versa.
2) The "hard" version of the proposed method, called hybrid–nongreedy asynchronous clustering (H-NAC) algorithm, includes the standard HNNs (and, thus, Bruck's Ln algorithm in [1] and the Ising model in statistical mechanics), Babadi and Tarokh's basic GADIA in [2], and the standard k-means [3], [4] (and the Lloyd [5], [6] algorithm) as its special cases.
3) We compare the clustering performance of the proposed algorithm H-NAC with those of the k-means, k-GADIA, spectral clustering, and the SGL in [46], which represents a state-of-the-art clustering algorithm as applied to various well-known benchmark datasets. The computer simulations confirm the superior performance of the H-NAC compared to these algorithms.
4) Furthermore, we also extend Bruck's Ln algorithm [1] from two-cluster case to arbitrary k-cluster case, which will be called the k-Ln algorithm in this article. The extended k-Ln clustering algorithm turns out to be equivalent to the basic version of the pioneering algorithm GADIA of Babadi and Tarokh [2].

The article is arranged as follows. In Section II, we summarize the standard EM algorithm for the GMM. Sections III and IV present the soft version and the hard version of the proposed hybrid clustering method, respectively. Computer simulations are presented in Section V, followed by the conclusions in Section VI.

II. Related Work: EM Algorithm and EM for GMM

A. EM Algorithm

Let us denote L-dimensional data as $\boldsymbol{x} = \{x_i\}_{i=1}^N \in \Re^L$. In the problem definition, the cluster label for each data point is not known. This unobserved/hidden label is called latent variable, which is represented as $z = \{z_j\}_{j=1}^K$ in this article. The latent variable of sample i, represented as $z_{j(i)}$, is multinomial and indicates which distribution a given sample x_i belongs to. Thus, the model is probability distribution function $p(\boldsymbol{x}, z; \theta)$ in which only $\boldsymbol{x}$ is observed. Assuming independent data samples $\{\boldsymbol{x}_i\}_{i=1}^N \in \Re^L$, our aim is to maximize the probability $l(\theta) = \prod_{i=1}^N p(\boldsymbol{x}_i; \theta)$. However, for the sake of mathematical simplicity, instead of the probability $l(\theta)$, it is common to maximize its log-likelihood

$$L(\theta) = \sum_{i=1}^{N} \ln\{p(\boldsymbol{x}_i; \theta)\}. \tag{1}$$

Incorporating the latent variable $z_{j(i)}$ into (1) and using Jensen's inequality, one obtains

$$L(\theta) \geq \underbrace{\sum_{i=1}^{N}\sum_{k=1}^{K} q_i(z_i = k) \ln\left\{\frac{p(\boldsymbol{x}_i, z_i = k; \theta)}{q_i(z_i = k)}\right\}}_{\text{Lower Bound: } Q_{\text{LB}}(\theta)} \tag{2}$$

where $q_i(z_i = k)$ represents the probability of z_i taking the value k under the distribution of q_i. The details of the derivations can be found in any related textbook or lecture notes. Let us represent the lower bound in (2) as $Q_{\text{LB}}(\theta) = \sum_{i=1}^N \sum_{k=1}^K q_i(z_i = k)\ln\{(p(\boldsymbol{x}_i, z_i = k; \theta))/(q_i(z_i = k))\}$. It is well-known that: 1) (2) holds for any distribution $q_i(z_i = k)$ and 2) if and only if the distribution $q_i(z_i = k)$ is chosen as

$$q_i(z_i = k) = p(z_i = k | x_i; \theta) \tag{3}$$

then $L(\theta) = Q_{\text{LB}}(\theta)$, i.e., the lower bound $Q_{\text{LB}}(\theta)$ becomes exactly equal to the log-likelihood $L(\theta)$ in (1) for the given parameters θ.

Let us represent the posterior as $\gamma_{z_i}^{(k)} = p(z_i = k | \boldsymbol{x}_i; \theta)$. In other words, $p(z_i | \boldsymbol{x}_i; \theta)$ is the posterior distribution of z_i given the observation x_i and the parameter θ. Similarly, the prior is $p(\boldsymbol{x}_i | z_i = j; \theta)$, which represents the probability of x_i given that the data sample has been drawn from cluster j for a fixed parameter set θ, and $\varphi_j = p(z_i = j; \theta)$, which represents the probability of cluster j for a given parameter set θ. Finally, the posterior probability, i.e., after we observe a specific observation (data sample) $\boldsymbol{x}_i$, is obtained by the Bayes rule

$$\gamma_{z_i}^{(k)} = p(z_i = j | \boldsymbol{x}_i; \theta) = \frac{p(\boldsymbol{x}_i | z_i = k; \theta) p(z_i = k; \theta)}{\sum_{j=1}^K p(\boldsymbol{x}_i | z_i = j; \theta) p(z_i = j; \theta)} \tag{4}$$

and

$$p(\boldsymbol{x}_i; \theta) = \sum_{j=1}^{K} p(\boldsymbol{x}_i, z_i = j; \theta) = \sum_{j=1}^{K} p(\boldsymbol{x}_i | z_i = j, \theta). \tag{5}$$

The EM algorithm will make the log-likelihood increase monotonically converging to a locally optimum solution. Thus, the likelihood that the given dataset $\boldsymbol{x} = \{\boldsymbol{x}_i\}_{i=1}^N$ is withdrawn from the distribution is maximized by the EM algorithm. Further details and proofs can be found in any related textbook. Table I presents the notation used in Sections II and III for convenience and clarity. Table II summarizes the standard EM algorithm.

B. EM for GMM

In this section, we summarize the standard EM algorithm for the well-known GMM. The GMM model with K different Gaussian distributions (K mixtures) has the following parameters.

$\boldsymbol{\mu}_k$: The mean of the k'th Gaussian distribution.

$\mathfrak{C}_k$: The covariance matrix of the k'th Gaussian distribution.

TABLE I
NOTATION TABLE

Variable	Meaning
$\{z_k\}_{k=1}^{K}$	latent variables (unobserved/hidden labels)
$q_i(z_i = k)$	the probability of z_i taking the value k under a distribution q_i
$p(\boldsymbol{x}_i \vert z_i = k; \theta)$	the prior probability, (the probability of $\boldsymbol{x}_i$ given the cluster k and parameter set θ)
$\gamma_{z_i}^{(k)} = p(z_i = k \vert \boldsymbol{x}_i; \theta)$	the posterior probability (i.e. after we observe a particular data sample $\boldsymbol{x}_i$, the probability that this observed data sample $\boldsymbol{x}_i$ is drawn from the cluster k for a given parameter set θ)
$\varphi_k = p(z_i = k; \theta)$	the probability of cluster k for a fixed parameter set θ
$\Delta_k^i(t)$ and $\Delta_k^i(t-1)$	sum of weighted dissimilarities between the sample $\boldsymbol{x}_i$ and cluster k at current step and previous step, respectively
$\boldsymbol{\mu}_k$ and $\boldsymbol{C}_k$	the mean and covariance matrix of the k'th Gaussian distribution
w_{ij}	dissimilarity between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

TABLE II
BRIEF EXPLANATION OF THE STANDARD EM ALGORITHM

Initialize the parameters θ.
Loop { t=t+1 1) Expectation Step (E-Step): Keeping the parameters $\theta(t)$ constant, calculate the posterior probabilities of cluster assignments $q_i(z_i) = \gamma_{z_i}^{(k)}(t)$ by (4) , $k = 1,2,,\ldots,K$, for each data sample x_i. (This choice maximizes the lower bound $Q(\theta)$ in (2) and makes it equal to the log-likelihood, i.e. $L(\theta) = Q_{LB}(\theta)$). 2) Maximization Step (M-Step): Keeping the posteriors $\gamma_{z_i}^{(k)}$ in step (1) constant, update the parameters $\theta(t)$ which maximizes the $Q_{LB}(\theta)$: $\theta(t) = arg\ max\ Q_{LB}(\theta)$ (6) }
Continue the loop until $Q_{LB}(\theta)$ (or equivalently parameter set $\theta(t)$ converges, reaching a local minima.

Here, $k = 1, 2, \ldots, K$. Each Gaussian distribution is given by

$$p_{\text{GMM}}(\boldsymbol{x}_i | z_i = k; \theta) = N(\boldsymbol{x}_i, \boldsymbol{\mu}_k, \mathfrak{C}_k)$$
$$= \frac{1}{\sqrt{(2\pi)^L |\mathfrak{C}_k|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \mathfrak{C}_{\boldsymbol{k}}^{-\mathbf{1}} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right). \quad (7)$$

From (4), the posterior probability (i.e., after we observe a particular data sample $\boldsymbol{x}_i$, the probability that this observed data sample $\boldsymbol{x}_i$ is drawn from the cluster k) in the GMM is obtained by

$$\gamma_{z_i}^{(k)} = \frac{N(\boldsymbol{x}_i, \boldsymbol{\mu}_k, \mathfrak{C}_k)\varphi_k}{\sum_{j=1}^{K} N(\boldsymbol{x}_i, \boldsymbol{\mu}_j, \mathfrak{C}_j)\varphi_j}. \quad (8)$$

The goal is to maximize the log-likelihood in (1) for the GMM model, i.e., $\max_{\mu,\mathfrak{C},\varphi} \sum_{i=1}^{N} \ln\{p(\boldsymbol{x}_i; \boldsymbol{\mu}, \mathfrak{C}, \varphi)\}$. Instead, the lower bound in (2) is maximized, denoted as $\text{LB}_{\text{GMM}}(\boldsymbol{\mu}_k, \mathfrak{C}_k, \varphi_k)$ for the GMM case

$$\text{LB}_{\text{GMM}}(\boldsymbol{\mu}_k, \mathfrak{C}_k, \varphi_k) = \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{z_i}^{(k)} \ln\left\{\frac{N(\boldsymbol{x}_i, \boldsymbol{\mu}_k, \mathfrak{C}_k)\varphi_k}{\gamma_{z_i}^{(k)}}\right\} \quad (9)$$

where $\varphi_j = p(z_i = j; \theta)$ and $\sum_{j=1}^{K} \varphi_j = 1$. From (3), data sample x_i is assigned to cluster k with the following probability:

$$q_i(z_i = k) = \gamma_{z_i}^{(k)}. \quad (10)$$

Applying the Lagrange multipliers' optimization approach yields the optimum parameters for the M-step of the EM algorithm for the GMM. The optimum GMM parameters (for the M-step) are obtained as follows, which can be found in any related textbook or lecture notes (see [19]):

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} \gamma_{z_i}^{(k)} \boldsymbol{x}_i}{\sum_{i=1}^{N} \gamma_{z_i}^{(k)}} \quad (11)$$

and

$$\mathfrak{C}_k = \frac{\sum_{i=1}^{N} \gamma_{z_i}^{(k)} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T}{\sum_{i=1}^{N} \gamma_{z_i}^{(k)}} \quad (12)$$

and

$$\varphi_k = p(z_i = j; \theta) = \frac{1}{N}\sum_{i=1}^{N} \gamma_{z_i}^{(k)}. \quad (13)$$

Equation (13) is obtained by the constraints $\sum_{j=1}^{K} \varphi_j = 1$ and $\sum_{j=1}^{K} \gamma_{z_i}^{(j)} = 1$ (for the details of the derivations by the Lagrange multipliers, see [19]).

III. FUSION OF THE GMM AND THE GRAPH CLUSTERING BY THE EM

In this section, we extend the EM formulation for the GMM with a novel weighted dissimilarity loss, as explained in the following.

A. Soft Version

In this section, we propose and analyze the "soft version" of our hybrid clustering algorithm. The same log-likelihood in (1) is maximized: $\max L(\theta) = \sum_{i=1}^{N} \ln\{p(x_i; \theta)\}$. First, let w_{ij} denote the dissimilarity between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Here, we define a new concept called "sum of weighted dissimilarities between the sample $\boldsymbol{x}_i$ and cluster k" as follows.

Definition: The sum of weighted dissimilarities between the sample $\boldsymbol{x}_i$ and cluster k at time (t) is defined as

$$\Delta_i^k(t) = \sum_{j=1}^{N} w_{ij} \gamma_{z_j}^{(k)}(t-1) \quad (14)$$

where $\gamma_{z_i}^{(k)}(t-1)$ represents the probability that the sample $\boldsymbol{x}_i$ is assigned to cluster k (assuming each component stands for a cluster) in the Expectation (E)-step at time $(t-1)$.

We introduce α and Δ_i^k into the argument of the Gaussian distribution in the GMM and split the argument of each Gaussian distribution into two parts as follows:

$$\begin{aligned} &p_H(\boldsymbol{x}_i|z_i = k;\theta)\\ &= N\big(\boldsymbol{x}_i, \boldsymbol{\mu}_k, \mathfrak{C}_k, \Delta_i^k, \alpha\big) = \frac{1}{\sqrt{(2\pi)^L|\mathfrak{C}_k|}}\\ &\times \exp\left(\begin{array}{c} -\dfrac{1}{2}\underbrace{(1-\alpha)\big(\boldsymbol{x}_i-\boldsymbol{\mu}_k\big)^T\mathfrak{C}_{\boldsymbol{k}}^{-\boldsymbol{1}}\big(\boldsymbol{x}_i-\boldsymbol{\mu}_k\big)}_{\text{“centroid-based” clustering part}}\\ -\dfrac{1}{2}\underbrace{\alpha\,\Delta_i^k}_{\text{“graph clustering” part}} \end{array}\right) \end{aligned} \tag{15}$$

where $0 \le \alpha \le 1$. If $\alpha = 0$ in (15), then the proposed priori is exactly equal to the standard GMM. Similar to (8), the posterior probability (after observing a specific sample $\boldsymbol{x}_i$), the probability that the sample $\boldsymbol{x}_i$ has been drawn from mixture component (cluster) k of the proposed hybrid clustering, is obtained by the Bayes rule

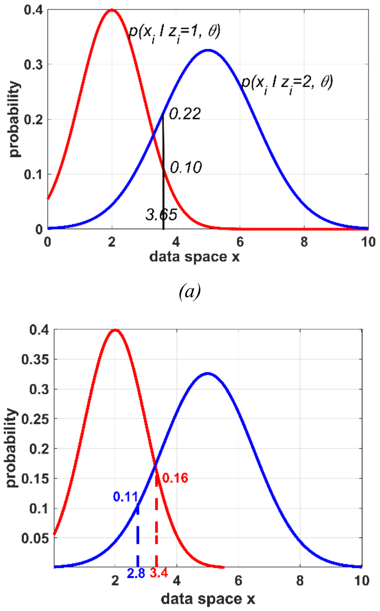
$$\gamma_{z_i}^{(k)} = q_i(z_i = k) = \frac{N\big(\boldsymbol{x}_i, \boldsymbol{\mu}_k, \mathfrak{C}_k, \Delta_i^k, \alpha\big)\varphi_k}{\sum_{j=1}^K N\big(\boldsymbol{x}_i, \boldsymbol{\mu}_k, \mathfrak{C}_k, \Delta_i^k, \alpha\big)\varphi_j} \tag{16}$$

where $N(\boldsymbol{x}_i, \boldsymbol{\mu}_k, \mathfrak{C}_k, \Delta_i^k, \alpha)$ is defined in (15). In other words, assuming that each mixture component represents a cluster, the posterior $\gamma_{z_i}^{(k)}$ represents the probability that this specific sample $\boldsymbol{x}_i$ is assigned to the cluster (mixture component) k for a given parameter set $\{\boldsymbol{x}_i, \boldsymbol{\mu}_k, \mathfrak{C}_k, \varphi_k, \alpha\}$ at time t.

The goal of our proposed hybrid model is to maximize the log-likelihood in (1), i.e., $\max L(\theta) = \sum_{i=1}^N \ln\{p(x_i;\theta)\}$. As in the GMM case in (9), we maximize its lower bound for the proposed hybrid model, denoted as $\mathrm{LB}_H(\boldsymbol{\mu}_k, \mathfrak{C}_k, \varphi_k, \Delta_i^k, \alpha)$

$$\begin{aligned} &\mathrm{LB}_H\big(\boldsymbol{\mu}_k, \mathfrak{C}_k, \varphi_k, \Delta_i^k, \alpha\big)\\ &\quad = \sum_{i=1}^N\sum_{k=1}^K \gamma_{z_i}^{(k)} \ln\left\{\frac{N\big(\boldsymbol{x}_i, \boldsymbol{\mu}_k, \mathfrak{C}_k, \Delta_i^k, \alpha\big)\varphi_k}{\gamma_{z_i}^{(k)}}\right\} \end{aligned} \tag{17}$$

where $\varphi_k = p(z_i = k;\theta)$ and $\sum_{j=1}^K \varphi_j = 1$. Applying the Lagrange multipliers' optimization approach and following the derivation of the standard GMM yield the same optimum parameters in (11)–(13) for the M-step simply because the terms Δ_i^k and α are constants as far as $\boldsymbol{\mu}_k$, $\mathfrak{C}_k$, and φ_k are concerned. The main difference between the standard GMM and the proposed hybrid model is in determining the posterior probability $\gamma_{z_i}^{(k)} = p(z_i = k|x_i;\theta)$ as seen by comparing (8) and (16). $\gamma_{z_i}^{(k)} = p(z_i = k|\boldsymbol{x}_i;\theta)$ shows the probability that a specific sample $\boldsymbol{x}_i$ has been drawn from mixture component (cluster) k, after observing the sample $\boldsymbol{x}_i$. It is obvious that the value of $\gamma_{z_i}^{(k)}$ plays a critical role in the clustering performance of either of the two models.



(a)

(b)

Fig. 1. Snaphot of the prior probabilities $p(\boldsymbol{x}_i|z_i = j;\theta)$, with $\sigma_1 = 1$, $\mu_1 = 2$ and $\sigma_2 = 1.5, \mu_2 = 5$, $j = 1, 2$, for a data sample $x_i = 3.65$: (a) standard GMM case and (b) proposed hybrid clustering where $\Delta_1^i = 1$ and $\Delta_2^i = 4$. (a) $\alpha = 0$ (standard GMM). (b) $\alpha = 0.5$ for the proposed hybrid algorithm.

In order to illustrate how the prior and posterior probabilities of the proposed hybrid clustering are completely changed as compared to the standard GMM case, we compare an arbitrary snapshot of $p_{\mathrm{GMM}}(\boldsymbol{x}_i|z_i = k;\theta)$ with $p_H(\boldsymbol{x}_i|z_i = k;\theta)$ in Fig. 1. For $x_i = 3.65$ in Fig. 1(a), the standard GMM clearly gives $\gamma_{z_i}^{(2)} > \gamma_{z_i}^{(1)}$. For the "soft" GMM, the probability that the sample $x_i = 3.65$ is assigned to cluster 2 is much higher than that of cluster 1. Therefore, for the "hard" GMM, the sample $x_i = 3.65$ is assigned to cluster 2 at this step. Thus, the "winning cluster" of the standard GMM is cluster 2 in this case, and any other choice is not possible. However, for the proposed hybrid algorithm, the "winning cluster" can be either of the two clusters: The sample $x_i = 3.65$ in Fig. 1 may be assigned to cluster 2 or cluster 1 depending on the value of α and Δ_1^i and Δ_2^i. As an example, if $\Delta_1^i = 1$ and $\Delta_2^i = 4$, then the "the winning cluster" is 1, i.e., the proposed hybrid algorithm assigns the same sample $x_i = 3.65$ to cluster 1, as shown in Fig. 1(b), which is the opposite of the GMM.

In summary, for the snapshot in Fig. 1, while the standard GMM dictates that the "winning" cluster is 2, the proposed hybrid algorithm allows cluster 1 to be the "winner" depending on α and Δ_{i1} in the E-step, as seen in Fig. 1(b).

The soft version of the proposed hybrid clustering is summarized in Table III.

B. Hard Version

In order to convert a soft clustering into a hard clustering, each data sample is often assigned, with probability 1, to the Gaussian distribution to which they most likely belong. Here, we will follow the same approach. First, we calculate the sum of weighted dissimilarities $(\Delta_i^k(t))$ between the sample $\boldsymbol{x}_i$ and cluster k using (14). Then, let us define

$$\beta_i^{(k)}(t) = \frac{N\big(\boldsymbol{x}_i, \boldsymbol{\mu}_k, \mathfrak{C}_k, \Delta_i^k(t), \alpha\big)\varphi_k}{\sum_{j=1}^K N\big(\boldsymbol{x}_i, \boldsymbol{\mu}_k, \mathfrak{C}_k, \Delta_i^k(t), \alpha\big)\varphi_j} \tag{18}$$

where $j = 1, 2, \ldots, K$. Then, the posterior probability in (16) in the hard clustering turns into the following:

$$\begin{aligned} \gamma_{z_i}^{(k)}(t) &= q_i(z_i = k)\\ &= \begin{cases} 1, & \text{if } k = \operatorname{index}_j\ \max\Big\{\beta_i^{(k)}(t)\Big\}_{j=1}^K\\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{19}$$

TABLE III

SOFT VERSION OF THE PROPOSED HYBRID CLUSTERING

```
Initialize the parameters $\boldsymbol{\mu}_k, \boldsymbol{C}_k, \varphi_k$.

Loop
{  t=t+1
    1) E-Step: Keeping the parameters $\boldsymbol{\mu}_k, \boldsymbol{C}_k, \varphi_k$ constant, calculate
        • the sum of weighted dissimilarities $\Delta_i^k$ in (14),
        • the posterior probabilities of cluster assignments $q_i(z_i) = \gamma_{z_i}^{(k)}(t)$ by (16).

    2) M-Step: Keeping the posteriors $\gamma_{z_i}^{(k)}$ in the E-step constant, update the parameters $\boldsymbol{\mu}_k, \boldsymbol{C}_k, \varphi_k$ by (11), (12) and (13).
}
Continue the loop until $LB_H(\boldsymbol{\mu}_k, \boldsymbol{C}_k, \varphi_k)$ (or equivalently parameter set $\boldsymbol{\mu}_k, \boldsymbol{C}_k, \varphi_k$ converges), reaching a local minima.
```

TABLE IV

HARD VERSION OF THE PROPOSED HYBRID CLUSTERING

```
Initialize the parameters $\boldsymbol{\mu}_k, \boldsymbol{C}_k, \varphi_k$.

Loop
{ t=t+1
    1) E-Step: Keeping the parameters $\boldsymbol{\mu}_k, \boldsymbol{C}_k, \varphi_k$ constant, calculate
        • the sum of weighted dissimilarities $\Delta_i^k$ in (21),
        • the posterior probabilities of cluster assignments $q_i(z_i) = \gamma_{z_i}^{(k)}(t)$ by (19). This dictates the cluster sets $S_j(t)$ in (20).

    2) M-Step: Keeping the posteriors $\gamma_{z_i}^{(k)}$ in the E-step constant, update the parameters $\boldsymbol{\mu}_k, \boldsymbol{C}_k, \varphi_k$ by (22),(23) and (24).
}
Continue the loop until $LB_H(\boldsymbol{\mu}_k, \boldsymbol{C}_k, \varphi_k)$ (or equivalently parameter set $\boldsymbol{\mu}_k, \boldsymbol{C}_k, \varphi_k$ converges), reaching a local minima.
```

where $k = 1, 2, \ldots, K$. Thus, in the hard version, a specific data sample $\boldsymbol{x}_i$ is assigned to only one cluster with probability 1 for a given parameter set $\boldsymbol{\mu}_k, \mathfrak{C}_k, \varphi_k, \alpha$ during the E-step. Thus, (19) yields K solid sets $\{S_j(t)\}_{j=1}^K$; each of which includes the indices of those data samples that are assigned to cluster k with probability 1 at time t in the E-step

$$S_j(t) = \{i\} \ni \gamma_{z_i}^{(j)} = 1 \tag{20}$$

where $j = 1, 2, \ldots, K$. Note that $\gamma_{z_i}^{(j)} = 1$ denotes that the data sample $\boldsymbol{x}_i$ is assigned to cluster (component) j with probability 1 (in the hard version) in the corresponding E-step, and $\sum_j^K \gamma_{z_i}^{(j)} = 1$, for any data sample $\boldsymbol{x}_i$. Writing (19) and (20) into (14), the dissimilarity $\Delta_i^k(t)$ in (14) turns into the following equation:

$$\Delta_i^k(t) = \sum_{j \in S_k(t-1)} w_{ij} \tag{21}$$

where $S_k(t-1)$ is the set of indices of data samples that were assigned to cluster k in the previous step at time $(t-1)$.

On the other hand, in the M-step, we maximize the lower bound $\mathrm{LB}_H(\boldsymbol{\mu}_k, \mathfrak{C}_k, \varphi_k, \alpha)$. The optimum parameters $\boldsymbol{\mu}_k, \mathfrak{C}_k$, and φ_k in the M-step of the *hard* version are found by writing (19)–(21) into (11)–(13), which yields

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{j \in S_k} \boldsymbol{x}_j \tag{22}$$

and

$$\mathfrak{C}_k = \frac{\sum_{j \in S_k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T}{N_k} \tag{23}$$

and

$$\varphi_k = p(z_i = j; \theta) = \frac{N_k}{N} \tag{24}$$

where N_k represents the number of data samples assigned to cluster k with probability 1 in the E-step at time t.

The hard version of the proposed hybrid clustering is summarized in Table IV.

IV. H-NAC: H-NAC

It is well-known that, if the covariance matrix in (12) is chosen as the unit matrix, then the standard *hard* GMM yields the standard k-means clustering (see [24]). Because the soft version of the proposed hybrid clustering in Table III is reduced to the standard GMM setting $\alpha = 0$, it implies that, if the covariance matrices in (23) are chosen as unit matrices and $\alpha = 0$, then the hard version of the proposed hybrid algorithm in Table IV results in the standard k-means clustering. If $0 < \alpha < 1$, then the proposed clustering in Table III (soft version) and Table IV (hard version) is a hybrid algorithm. In order to analyze the proposed algorithm, we need first to summarize some definitions from the graph theory [15].

Definition: Cut of a Simple Graph: A cut means a partition of the vertices of the graph into two sets.

Definition: Size of a Cut: The size of a cut is the sum of the edges cutting the graph into two groups.

Definition: Max-Cut of a Graph: A maximum cut is a cut whose size is not smaller than the size of any other cut.

Definition: k-Cut of a Graph: A k-cut is defined as the sum edges cutting the graph into k groups.

Definition: Max k-Cut of a Graph: A max k-cut is a k-cut whose size is not smaller than the size of any other k-cut.

Definition: Dissimilarity between any two data samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, represented as w_{ij}, represents how dissimilar they are, and it can be defined by different measures, such as spatial distance or correlation. Let $f(\boldsymbol{x}_i, \boldsymbol{x}_j)$ represent the dissimilarity function between sample $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Thus, $w_{ij} = f(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

Definition: I_k: Sum of intracluster dissimilarities for cluster k, $k = 1, 2, \ldots, K$, is defined as

$$I_k(S_k(n)) = \sum_{j \in S_k} \sum_{i \in S_k} w_{ij} \tag{25}$$

where $w_{ij} \geq 0$ represents the dissimilarity between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

Definition: vol(**W**): Volume of *a* matrix **W**, where $w_{ij} \geq 0$, is defined as

$$\mathrm{vol}(W) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}. \tag{26}$$

Definition: $Q_G(t)$: Sum of intracluster dissimilarities over all clusters is defined as follows:

$$Q_G\left(\{S_k(t)\}_{k=1}^{K}\right) = \sum_{k=1}^{K} I_k(S_k(t)). \tag{27}$$

Definition: $Q_{km}(t)$: *Sum of Intracluster Error Squares:* Given a dataset $\{\boldsymbol{x}_i\}_{i=1}^{N} \in \Re^L$ in a deterministic setting, the sum of intracluster error squares is given by

$$Q_{km}\left(\{S_k(t)\}_{s=1}^{L}\right) = \sum_{k=1}^{K}\left(\sum_{i\in S_k(t)} \|\boldsymbol{x}_i - \boldsymbol{c}_k\|_2^2\right) \tag{28}$$

where set $S_k(t)$ represents the indices of data samples in cluster k at time t. Note that the k-means clustering (Lloyd's algorithm) minimizes the sum of intracluster error squares (i.e., the variances) in (28).

Now, we are ready to present one of our findings as follows. In this section, without loss of generality and for the sake of simplicity, we define the dissimilarity w_{ij} between arbitrary samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as

$$w_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2. \tag{29}$$

We present our hybrid clustering algorithm called H-NAC as follows:

Definition: $Q_H(\{C_s\}_{s=1}^{L})$: H-NAC cost function at time t is defined as follows:

$$Q_H(t) = (1-\alpha)Q_{km}(t) + \alpha Q_G(t) \tag{30}$$

where $0 \leq \alpha \leq 1$, $Q_{km}(t)$ is the k-means cost function (quantization error) defined by (28), and $Q_G(t)$ is the sum of intracluster dissimilarities defined by (27).

Let N_k represent the number of samples in cluster k at a particular time. Using $\|\boldsymbol{x}_i - \boldsymbol{c}_k\|_2^2 = \|\boldsymbol{x}_i\|_2^2 + \|\boldsymbol{c}_k\|_2^2 - 2\langle \boldsymbol{x}_i, \boldsymbol{c}_k\rangle$, and $\boldsymbol{c}_k = (1/N_k)\sum_{i\in S_k} \boldsymbol{x}_i$, it is straightforward to obtain the following:

$$\sum_{i\in S_k} \|\boldsymbol{x}_i - \boldsymbol{c}_k\|_2^2 = \sum_{i\in S_k} \|\boldsymbol{x}_i\|_2^2 - \frac{1}{N_k}\sum_{m\in S_k}\sum_{n\in S_k} \langle \boldsymbol{x}_m, \boldsymbol{x}_n\rangle \tag{31}$$

and

$$2N_k \sum_{i\in S_k} \|\boldsymbol{x}_i - \boldsymbol{c}_k\|_2^2 = \sum_{i\in S_k}\sum_{j\in S_k} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2. \tag{32}$$

Using (31) and (32) in (30) yields the following H-NAC cost function:

$$Q_H(t) = \sum_{k=1}^{K}\left(\sum_{i\in S_k} \|\boldsymbol{x}_i - \boldsymbol{c}_k\|_2^2(\alpha(2N_k - 1) + 1)\right). \tag{33}$$

Proposition 1: Given a data sample set $\{\boldsymbol{x}_i\}_{i=1}^{N} \in \Re^L$ in a deterministic setting and the cluster number K, if

1) $0 \leq \alpha \leq 1$.
2) Priori cluster probability $\varphi_j = (1/K)$.
3) The covariance matrix $\mathfrak{C}_j$ is chosen as the identity matrix $\boldsymbol{I}_L$.
4) The dissimilarity w_{ij} is chosen as (29), then the hard version of the proposed hybrid clustering in Table IV above locally *minimizes* the hybrid cost function $Q_{HN}(t)$ in (30). [Note that the H-NAC cost function $Q_H(t)$ is given by (33) if w_{ij} is defined by (29)].

Proof:

1) The E-step of the hard version in Table IV, $\gamma_{z_i}^{(k)}(t)$ in (19) yields the sets $\{S_k(t)\}_{k=1}^{K}$ in (20) at time t. Thus, $S_j(t)$ becomes the set of indices of data samples that are assigned to cluster j with probability 1 at time t. From (19) to (20) and the definitions above, we write

$$\mathrm{vol}(\boldsymbol{W}) = \underbrace{\sum_{k=1}^{K}\left(\sum_{j\in S_k(t)}\sum_{i\in S_k(t)} w_{ij}\right)}_{\text{sum of inter-cluster weights}} + \underbrace{\sum_{k=1}^{K} I_k(t)}_{\text{sum of intra-cluster weights}} \tag{34}$$

where $\bar{S}_k(t)$ represents the set of indices of all other data samples except those in cluster k, and $I_k(t)$ is the sum of intracluster dissimilarities defined by (25). Let us denote the "winner cluster index" for data sample $\boldsymbol{x}_i$ at time *(t + 1)* as z_i^w so that $\gamma_{z_i}^{(w)} = 1$. In other words, using (19)

$$z_i^w = \mathrm{index}_k \max\left\{\gamma_{z_i}^{(k)}(t+1)\right\}_{k=1.}^{k=K} \tag{35}$$

Furthermore, let us denote the "winner cluster index" for data sample $\boldsymbol{x}_i$ at the previous time (t) as z_i^p, which means that

$$z_i^p = \mathrm{index}_k \max\left\{\gamma_{z_i}^{(k)}(t)\right\}_{k=1.}^{k=K} \tag{36}$$

From (19), (20), (35), and (36), we have $\boldsymbol{x}_j \in S_{z_j^p}(t)$ and $\boldsymbol{x}_j \in S_{z_i^w}(t+1)$ i.e., the sample $\boldsymbol{x}_i$ is moved from cluster z_i^p to cluster z_i^w at time *(t + 1)*. Form (19), (20), and (25), we have

$$I_k(t) = \underbrace{\Delta_i^{z_i^p}(t)}_{\text{related to } \boldsymbol{x}_i} + \underbrace{I_k^{\text{others}}(t)}_{\text{without } \boldsymbol{x}_i}. \tag{37}$$

Let $d_k^i(t)$ represent the squared distance between sample $\boldsymbol{x}_i$ and centroid $\boldsymbol{c}_k$ at time t

$$d_i^k(t) = \|\boldsymbol{x}_i - \boldsymbol{c}_k(t)\|_2^2 \tag{38}$$

where $k\epsilon\{1, 2, \ldots, K\}$. We also define the following "hybrid dissimilarity" $(\tau_i^k(t))$ between the sample $\boldsymbol{x}_i$ and cluster k at time t as follows:

$$\tau_i^k = (1-\alpha)d_i^k + \alpha\Delta_i^k \tag{39}$$

where d_i^k is given by (38), Δ_k^i is in (21), and $0 \leq \alpha_t \leq 1$. From (15), (18)–(20), (35), and (39), we obtain that

$$z_i^w = \mathrm{index}_k \min\left\{\tau_i^1, \tau_i^2, \ldots, \tau_i^K\right\}. \tag{40}$$

In other words, (40) is obtained because of the fact that the smallest $\tau_i^k(t)$ in (39) yields the highest $\beta_i^{(k)}(t)$ in (18) if $\varphi_j = 1/K$ is the same for each cluster. Using (27), (28), (39), and (40), the E-step by (19) and (20) results in the following H-NAC cost function in (30) at time (t) and time *(t + 1)*, respectively:

$$Q_H(t) = \underbrace{\alpha\Delta_i^{z_i^p}}_{\text{related to } \boldsymbol{x}_i} + \underbrace{\alpha Q_{G,i}^{\text{others}}(t)}_{\text{without } \boldsymbol{x}_i} + \underbrace{(1-\alpha)d_i^{z_i^p}(t)}_{\text{related to } \boldsymbol{x}_i} + \underbrace{(1-\alpha)Q_{km,i}^{\text{others}}(t)}_{\text{without } \boldsymbol{x}_i} \tag{41}$$

TABLE V

H-NAC: H-NAC ALGORITHM ($\mathfrak{C}_k = \boldsymbol{I}_L, w_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2, \varphi_k = 1/K$)

Initialize the parameters $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$.
1) E- Step: Keeping the parameters $\boldsymbol{\mu}_k$ constant, calculate: the sum of weighted dissimilarities Δ_i^k in (21), the posterior probabilities of cluster assignments $q_i(z_i) = \gamma_{z_i}^{(k)}(t)$ by (19). *(Alternatively, equivalently, from Proposition 1 and eq.(39), we find the “winning cluster” z_i^w in (40) for sample $\boldsymbol{x}_i$. This dictates the two cluster sets $S_{z_i^p}(t+1)$ and $S_{z_i^w}(t+1)$ to be updated accordingly by (20)).*
2) M-Step: Keeping the posteriors $\gamma_{z_i}^{(k)}$ in the E-step constant, update the parameters $\boldsymbol{\mu}_k$ by (22).
Continue the loop until $LB_H(\boldsymbol{\mu}_k)$ (or equivalently until parameter set $\boldsymbol{\mu}_k$ converges, reaching a local minima.

and

$$\begin{aligned} Q_H(t+1) = \ & \underbrace{\alpha \Delta_i^{z_i^w}}_{\text{elated to } \boldsymbol{x}_i} + \underbrace{Q_{G,i}^{\text{others}}(t+1)}_{\text{without } \boldsymbol{x}_i} \\ & + \underbrace{(1-\alpha) d_i^{z_i^w}(t)}_{\text{related to } \boldsymbol{x}_i} \\ & + \underbrace{(1-\alpha) Q_{km,i}^{\text{others}}(t+1)}_{\text{without } \boldsymbol{x}_i} \end{aligned} \tag{42}$$

where Δ_i^k is given by (21), $Q_{G,i}^{\text{others}}$ represents the sum of all terms without $\boldsymbol{x}_i$ in (27), $d_i^k(t)$ is given by (38), and $Q_{km,i}^{\text{others}}$ represents the sum of all terms without $\boldsymbol{x}_i$ in (28). Using (40)–(42) and the fact that $Q_{G,i}^{\text{others}}(t) = Q_{G,i}^{\text{others}}(t+1)$ and $Q_{km,i}^{\text{others}}(t) = Q_{km,i}^{\text{others}}(t+1)$, we conclude that

$$Q_H(t+1) \begin{cases} = Q_H(t), & \text{if } z_j^w = z_j^p \\ < Q_H(t), & \text{if } z_j^w \neq z_j^p. \end{cases} \tag{43}$$

In the M-step, keeping the posterior $\gamma_{z_i}^{(k)}(t)$ constant that is found in the E-step, we update the optimum cluster H-NAC centroids $\boldsymbol{\mu}_k$ by (22). Applying the E-step and the M-step in the next time step, $Q_H(t)$ is further decreased due to (43). Because the global minimum of the *H-NAC cost function* $Q_H(t)$ in (33) is finite, the *H-NAC* algorithm converges to local minima after a finite number of steps. This completes the proof.

The H-NAC algorithm in Proposition 1 (i.e., $0 \leq \alpha_t \leq 1$, $\mathfrak{C}_k = \boldsymbol{I}_L, \varphi_k = 1/K$, and $w_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$ in Table IV) is simplified in Table V.

In the H-NAC in Table V, the update of centroid $\boldsymbol{\mu}_k$ can be done either after every $q_i(z_i)$ is calculated by (19) (i.e., sample-based update) or after all the $\{q_i(z_i)\}_{i=1}^N$ are calculated (i.e., epoch-based update). Here, we denote the latter one as H-NAC-e.

Proposition 2: The H-NAC algorithm in Proposition 1 (i.e., $0 \leq \alpha_t \leq 1$, $\varphi_k = (1/K), \mathfrak{C}_k = \boldsymbol{I}_L$, and $w_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$) includes the k-means clustering (Lloyd algorithm [5], [6]) in computer science, the standard HNN in electrical engineering, the basic GADIA [2] in wireless communications, and the Ising model in statistical mechanics as its special cases. To be precise, the H-NAC in Table V (in Proposition 1) is reduced to

1) The k-means clustering (Lloyd algorithm [5], [6]) if $\alpha = 0$.
2) The basic GADIA in [2] if $\alpha = 1$ maximizing the K-cut of the dissimilarity matrix $\boldsymbol{W}$ (graph clustering).
3) The HNN [7], [8], equivalently, Bruck's Ln algorithm in [1], if $K = 2$ and $\alpha = 1$.
4) The Ising model in statistical mechanics if $K = 2$ and $\alpha = 1$.

Proof:

1) Taking $\alpha = 0$ in (39) and (40) yields the “winning cluster index” for the data sample $\boldsymbol{x}_i$ in the E-step as

$$z_i^w = \text{index}_k \min \left\{d_i^1, d_i^2, \ldots, d_i^K\right\}. \tag{44}$$

The E-step in (44) and the M-step (22) finding the cluster centroid $\boldsymbol{\mu}_k$ yield the standard k-means clustering.

2) By definition, the K-cut of matrix $\mathbf{W}$ is equal to

$$K - \text{cut}(\boldsymbol{W}) = \sum_{k=1}^{K} \left(\sum_{j \in S_k} \sum_{i \in \bar{S}_k} w_{ij} \right). \tag{45}$$

Taking $\alpha = 1$ in (39) and (40) yields $\min Q_{HN}(t) = \min Q_G(t)$ and the “winning cluster index” for the data sample $\boldsymbol{x}_i$ in the E-step as

$$z_i^w = \text{index}_k \min \left\{\Delta_i^1, \Delta_i^2, \ldots, \Delta_i^K\right\}. \tag{46}$$

Using (21), (27), (30), (34), and (46), and the fact that $\text{vol}(\boldsymbol{W})$ is constant, we conclude that, under the light of Corollary 2 and Proposition 6 of [9], the $K - \text{cut}(\boldsymbol{W})$ in (45) strictly increases each time an arbitrary $\boldsymbol{x}_i$ is assigned from z_i^p to z_i^w by (46) in the E-step of the H-NAC. Because the maximum of $K - \text{cut}(\boldsymbol{W})$ is finite, the H-NAC algorithm converges to local maxima of the $K - \text{cut}(\boldsymbol{W})$ after a finite number of steps.

Comparing the cluster update of the H-NAC algorithm in (46) with the channel allocation update of the basic version of the pioneering algorithm GADIA in [2], we observe that they are equivalent: while Δ_i^k represents the sum of pairwise dissimilarities between sample $\boldsymbol{x}_i$ and cluster k in the proposed H-NAC, it actually corresponds to the sum of cochannel interference experienced by the transmitter i in channel k in the basic GADIA in [2]. It is proven in [2] that $\sum_{k=1}^K I_k(t+1) \leq \sum_{k=1}^K I_k(t)$, where $I_k(t)$ is defined in (25) above. This shows that the H-NAC algorithm in Table V is equivalent to the basic GADIA in [2] (although their problems, motivations, and formulations are completely different) if $\alpha = 1$.

3) Bruck [1] and Uykan [9] further investigate the working principle of the HNN and show that maximizing the $K - \text{cut}(\boldsymbol{W})$ in (45) is equivalent to minimizing the HNN's energy Lyapunov) function

$$L(k) = -\boldsymbol{y}^T(k)\boldsymbol{W}\boldsymbol{y}(k) + \boldsymbol{b}^T y(k) \tag{47}$$

where $y_i(k) \epsilon \{-1, +1\}$ is the state of neuron i, b_i is the bias of neuron i, $i = 1, 2, \ldots, N$, and w_{ij} is defined in (29). Uykan [9] proves that the HNN [7], [8] is a special case of the basic GADIA in [2]. This shows that the H-NAC algorithm in Table V is equivalent to the

TABLE VI

COMPARISON OF THE H-NAC WITH THE k-MEANS CLUSTERING, HNNS, AND THE BASIC GADIA

K	α	**H-NAC :** $Q_H(n)$ in (30) (or (33))
= 2	1	H-NAC becomes equal to the Hopfield Neural Network: $Q_H(t) = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} y_i y_j$ where $y_i, y_j \in$ {-1, + 1} and w_{ij} is in (29).
≥ 2	0	H-NAC becomes equal to the k-means clustering: $Q_H(t) = \sum_{k=1}^{K} \left(\sum_{i \in S_k} \lVert \boldsymbol{x}_i - \boldsymbol{c}_k \rVert_2^2 \right)$
≥ 2	1	H-NAC becomes equal to the basic GADIA [2]: $Q_H(t) = \sum_{k=1}^{K} I_k(t)$ where $I_k(t)$ is given by (25)
≥ 2	$0 < \alpha < 1$	H-NAC $Q_H(t)$ is given by (30) (or (33))

HNN and, thus, Bruck's Ln algorithm in [1] if $\alpha = 1$ and $K = 2$.

4) It is well-known that the HNN's energy function in (47) corresponds to the Hamiltonian function of the classical Ising model in statistical mechanics. The Ising model is a mathematical model of ferromagnetism in classical statistical mechanics and consists of discrete variables that can be in one of two states $\{-1, +1\}$ that represent magnetic dipole moments of atomic spins. Proposition 2 above and (47) imply that the H-NAC includes the Ising model as its special case, which completes the proof.

The findings in Proposition 2 are summarized in Table VI. We extend Bruck's Ln algorithm [1] from a two-cluster case to an arbitrary k-cluster case in Appendix A, which will be called the k-Ln algorithm in this article. A connection between the k-Ln algorithm and the spectral clustering formulation is presented in Appendix B, relaxing the discrete-solution constraint. Furthermore, the extended k-Ln clustering algorithm turns out to be equivalent to the basic version of the pioneering algorithm GADIA of Babadi and Tarokh [2]. We call the basic GADIA in [2] when applied to the data clustering problem as k-GADIA.

A. Increasing/Decreasing the Parameter α_t

The parameter $\alpha_t (0 \leq \alpha_t \leq 1)$ determines how much the proposed hybrid algorithm looks like a centroid-based clustering (the k-means algorithm) and how much it looks like a graph clustering (the k-Ln) algorithm. The smaller the α_t is, the more it looks like k-means, and the less it looks like the k-Ln, and vice versa. In the proposed H-NAC in Table V, the parameter α is a constant. It is straightforward to show that the parameter α_t at time t $(0 \leq \alpha_t \leq 1)$ may gradually be increased all the way up to 1 or alternatively may gradually be decreased all the way down to 0 as follows: let $\epsilon > 0$, a small positive number.

1) Gradually increasing α_t to 1 (i.e., gradually turning the H-NAC to a graph-clustering algorithm, such as the basic k-GADIA): If $d_i^{s_j} > \Delta_i^{s_j}$, then α_t is gradually increased, i.e., $\alpha_{t+1} = \min\{\alpha_t + \epsilon, 1\}$; otherwise, $\alpha_{t+1} = \alpha_t$. This satisfies $Q_H(t+1) \leq Q_H(t)$.
2) Gradually decreasing α_t to 0 (i.e., gradually turning the H-NAC to the pure standard k-means clustering, which is centroid-based clustering): If $d_i^{s_j} < \Delta_i^{s_j}$, then α_t is gradually decreased, i.e., $\alpha_{t+1} = \max\{\alpha_t - \epsilon, 0\}$; otherwise, $\alpha_{t+1} = \alpha_t$. This satisfies $Q_H(t+1) \leq Q_H(t)$.

B. How to Choose the Parameter α

If the parameter α of the H-NAC is kept as a constant, then, as a rule of thumb, α is first chosen in the range of $1/(N/K)$, where N is the number of data points and K is the number of clusters, as will be seen in most of the simulations in Section V. This is because of the following reason: The assignment of $\boldsymbol{x}_i$ to cluster k depends on the "hybrid dissimilarity" $\tau_i^k(t)$ in (39), which consists of d_i^k in (38) (centroid-based clustering part) and Δ_k^i in (21) (graph clustering part). Because d_i^k is between the data sample $\boldsymbol{x}_i$ and the centroid of cluster k, while Δ_i^k is between the data sample $\boldsymbol{x}_i$ and all other data samples in cluster k, very roughly speaking, the magnitude of Δ_k^i is in the order of (N/K) times the magnitude of d_i^k, assuming that the number of samples in cluster k is around (N/K). In other words, in the hybrid dissimilarity $\tau_i^k = (1-\alpha)d_i^k + \alpha \Delta_i^k$, there are two terms "competing" with each other; however, one term (Δ_i^k) is, roughly speaking, about (N/K) times bigger than the other term (d_i^k). In order to balance these two terms in τ_i^k, as a rule of thump, α is first chosen in the range of $1/(N/K)$. This choice of α gives priority to neither the "centroid-based clustering part" nor the "graph clustering part." Otherwise, for example, if $\alpha \gg 1/(N/K)$, then the H-NAC looks more like a graph clustering algorithm, and if $\alpha \ll 1/(N/K)$, then it looks more like a centroid-based clustering algorithm.

C. Discussion About the H-NAC, the k-Means, and Graph Clustering

As explained in Sections II and III, the proposed H-NAC is a hybrid clustering touching both the k-means and the k-GADIA (graph clustering) simultaneously. Our starting point in Section II was the EM formulation as applied to the GMM whose argument has been extended by the dissimilarities of the samples (in the feature space). Unlike the spectral clustering and most of the spectral clustering-based graph clustering, the proposed H-NAC does not calculate any eigenvector, nor does it apply an eigendecomposition to a Laplacian matrix (although the k-Ln and k-GADIA are also related to the spectral clustering formulation, as shown in Appendix B). Instead, the H-NAC is implemented like the k-means clustering by taking also the pairwise dissimilarities into account at the cost of an additional computational burden. Because the dissimilarities are precalculated offline and are read from a lookup table, the H-NAC's computational cost is slightly higher than the k-means clustering. Therefore, as far as the implementation is concerned, it would be fair to say that the H-NAC resembles more like the k-means than any graph clustering algorithm. That is why we also provided detailed literature on the k-means clustering in Section I. It is well-known that the k-means minimizes the intracluster distances (dissimilarities), hoping that the intercluster distances (dissimilarities) are also maximized. In other words, the k-means does not explicitly take the intercluster dissimilarities into account during optimization. On the other hand, the proposed H-NAC simultaneously takes both into account: the H-NAC simultaneously both minimizes the sum of intracluster distances (dissimilarities) and maximizes the sum of intercluster dissimilarities during optimization (see Corollary 1 in Appendix A and (51) in Appendix B). What is more, the H-NAC also performs graph clustering due to the weighted dissimilarity loss added to the argument. As explained in Section II, we do not impose any assumption on how the dissimilarity between samples (features) is defined. For the sake of brevity and clarity, we define the dissimilarity as the squared Euclidean norm in this article.

V. Simulation Results

In this section, we compare the clustering performance of the proposed H-NAC with those of k-means, the k-GADIA, spectral clustering, and the SGL in [46], which represents one of the state-of-the-art clustering algorithms. The reason why we have chosen the SGL in [46] as a reference algorithm in our article is simply because the SGL [46] has shown superior performance compared to many state-of-the-art clustering methods, such as the accelerated low-rank representation (ALRR) published in 2018 [72], the K-multiple-means (KMMs) in 2019 [73], the efficient sparse subspace clustering (ESSC) (an ESSC algorithm) in 2020 [74], the fast normalized cut (FNC) (normalized-cut-based clustering method) in 2018 [75], and the sparse subspace clustering-orthogonal matching pursuit (SSC-OMP) (a popular sparse subspace clustering (SSC) algorithm) [76]. In [46], the SGL's multiview version (MSGL) is also compared to the state-of-the-art multiview clustering algorithms recently published in [77]–[80]. Furthermore, we also examine the performance of the traditional spectral clustering [53], [54] algorithm in our simulation campaigns for comparison reasons. All the algorithms (the k-means, the H-NAC, spectral clustering, and the SGL [46]) run under the same initial conditions using the same features for the sake of fairness. Toy examples are presented in the supplementary material.

Here, we use the clustering accuracy (ACC), the normalized mutual information (NMI), and the purity to evaluate the clustering performance, such as in many other articles (e.g., [46]). We conduct all our experiments on a computer with a 2-GHz Intel Core i7 CPU and 16-GB RAM MATLAB R2016a.

A. Example 1: 20NewsGroup Dataset[1]

The k-means clustering has widely been applied to the text document clustering, which is an important and challenging problem (see [29] and [30]). We run a large simulation campaign using the well-known 20Newsgroups dataset [20].

Each document is represented as a 26 214-D normalized *tf-idf* [21], [22] sparse vector [20]. The best parameter α if chosen over the following range as a rule of thumb in all the simulations below without any further tuning or optimization, whatsoever $\alpha \in (1/N_k)[0, 0.5, 1, 2, 3, 4, 5, N_k]$. The initial conditions are always the same for all clustering algorithms k-means, the basic GADIA (i.e., k-Ln algorithm), the spectral clustering [53], and the SGL [46] for the sake of fairness.

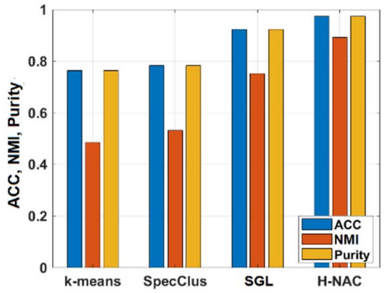


Fig. 2. Clustering performance (ACC, NMI, and purity) of the algorithms for a subset of the 20NewsGroup dataset with three clusters and 100 samples in each cluster.

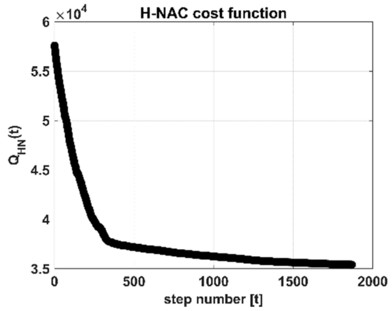


Fig. 3. Evolution of the H-NAC cost function $Q_H(t)$ in (33) for a subset of the 20NewsGroup dataset with three clusters and 100 samples in each cluster.

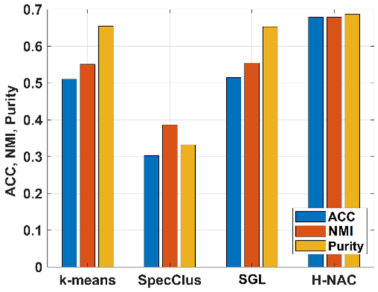


Fig. 4. Clustering performance (ACC, NMI, and purity) of the algorithms for the 20NewsGroup dataset with 20 clusters and 100 samples in each cluster.

Clustering performance (ACC, NMI, and purity) of the algorithms for a subset of the 20NewsGroup dataset with three clusters and 100 samples in each cluster is shown in Fig. 2 and for the 20NewsGroup dataset with 20 clusters and 100 samples in each cluster is shown in Fig. 4, respectively. The evolution of the H-NAC cost function $Q_H(t)$ in (33) for the three-cluster case is shown in Fig. 3, which confirms Proposition 1. As seen from Figs. 2 and 4, the best performance is achieved by the proposed H-NAC in terms of the ACC, NMI, and purity.

B. Example 2: MNIST Dataset[2]

Each sample in the well-known Mixed National Institute of Standards and Technology (MNIST) database of handwritten digits has 784 features. The number of clusters is 10, and we take 300 samples from each cluster. In Fig. 5, we examine how sensitive the clustering performance is to the parameter α over $\alpha \in (1/300)[0, 0.5, 1, 2, 3, 4, 5, 300]$. The figure shows that the best performance of the H-NAC is very stable with respect to a large range of α values for the MNIST dataset. From Fig. 5, the H-NAC parameter α is chosen as $\alpha = 0.01$.

[1] http://qwone.com/~jason/20Newsgroups/

[2] http://yann.lecun.com/exdb/mnist/

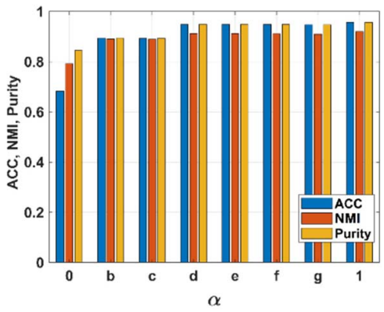


Fig. 5. Clustering performance (ACC, NMI, and purity) of the H-NAC with respect to parameter α for the subset of the MNIST dataset.

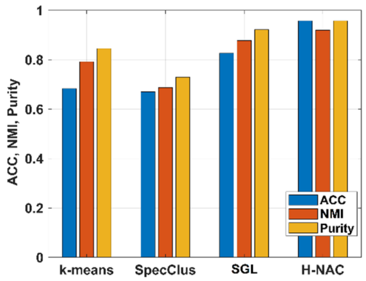


Fig. 6. Clustering performance (ACC, NMI, and purity) of the algorithms for the subset of the MNIST dataset (ten clusters and 300 samples in each cluster).

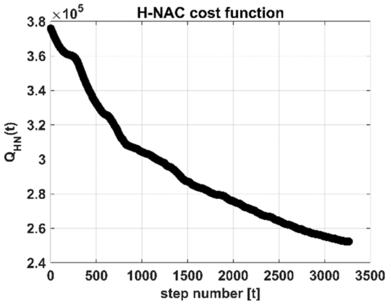


Fig. 7. Evolution of the H-NAC cost function $Q_H(t)$ in (33) for the MNIST dataset with ten clusters and 100 samples in each cluster.

The clustering performance (ACC, NMI, and purity) of the algorithms for the MNIST dataset (with ten clusters and 300 samples in each cluster) is shown in Fig. 6. The best performance of the SGL [46] is obtained by $m = 50$, $\beta = 0.01$, and $\alpha = 0.1$. The evolution of the H-NAC cost function $Q_H(t)$ in (33) is shown in Fig. 7, which confirms Proposition 1. As seen from Fig. 6, the best performance is achieved by the proposed H-NAC in terms of the ACC, NMI, and purity for this subset of the MNIST dataset.

C. Example 3: TR45 Dataset[3]

Each sample in the well-known TR45 dataset has 8261 features. The number of clusters is 10, and we take 14 samples from each cluster because there existed only 14 samples in one of the clusters in the TR45 dataset (in this article, we assume that the samples are distributed to the clusters evenly). In Fig. 8, we examine how sensitive the clustering performance is to the parameter α over $\alpha \in [0, 0.0357, 0.0714, 0.1071, 0.1429, 0.1786, 0.2, 1]$. From the figure, the H-NAC parameter α is chosen as $\alpha = 0.1071$. On the other hand, the best performance of the SGL [46] is obtained by $m = 50$, $\beta = 1$, and $\alpha = 10$ for this particular dataset.

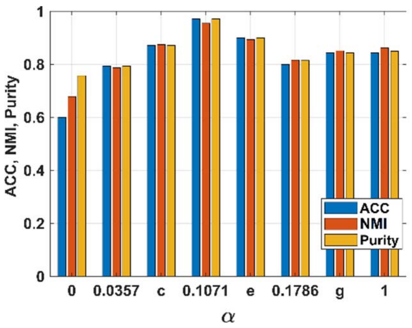


Fig. 8. Clustering performance (ACC, NMI, and purity) of the H-NAC with respect to parameter α for the TR45 dataset.

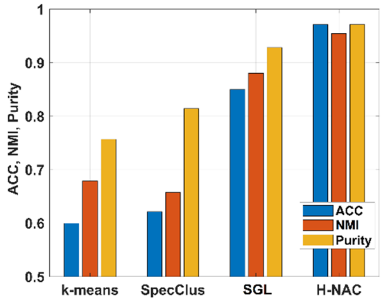


Fig. 9. Clustering performance (ACC, NMI, and purity) of the algorithms for the subset of the TR45 dataset (ten clusters and 14 samples in each cluster).

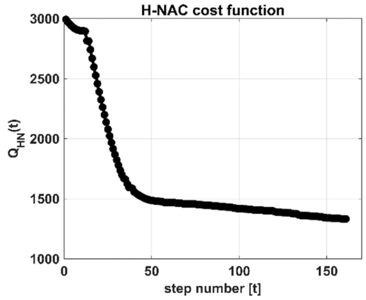


Fig. 10. Evolution of the H-NAC cost function $Q_H(t)$ in (33) for the subset of the TR45 dataset (ten clusters and 14 samples in each cluster).

The clustering performance (ACC, NMI, and purity) of the algorithms for the TR45 dataset (with ten clusters and 14 samples in each cluster) is shown in Fig. 9. The evolution of the H-NAC cost function $Q_H(t)$ in (33) is shown in Fig. 10, which confirms Proposition 1. As seen from Fig. 9, the best performance is achieved by the proposed H-NAC in terms of the ACC, NMI, and purity for the subset of the TR45 dataset.

D. Example 4: Caltech Dataset[4]

The Caltech-7 is a multiview object recognition dataset. Because we focus on the single-view clustering in this article, we take only one view that is the Gabor features of the dimension of 48. (We call it Caltech-Gabor dataset.) The number of clusters is 7, and we take 34 samples from each cluster because there existed only 34 samples in one of the clusters in this single-view Caltech-Gabor dataset (in this

[3] http://www-users.cs.umn.edu/han/data/tmdata.tar.gz

[4] http://www.vision.caltech.edu/ImageDatasets/Caltech101/

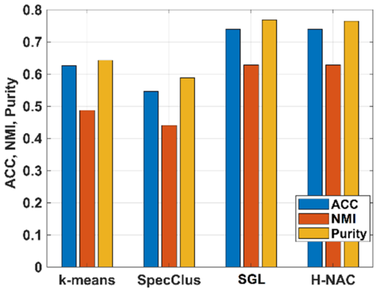


Fig. 11. Clustering performance (ACC, NMI, and purity) of the algorithms for the Caltech-Gabor dataset (seven clusters and 34 samples in each cluster).

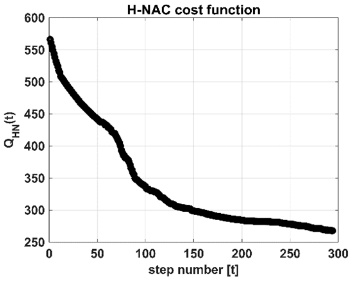


Fig. 12. Evolution of the H-NAC cost function $Q_H(t)$ in (33) for the Caltech-Gabor dataset (seven clusters and 34 samples in each cluster).

article, we assume that the samples are distributed to the clusters evenly). The best H-NAC performance is obtained by $\alpha = 0.03$ out of eight different values. On the other hand, the best performance of the SGL [46] is obtained by $m = 25$, $\beta = 0.01$, and $\alpha = 10$ out of 24 different combinations.

The clustering performance (ACC, NMI, and purity) of the algorithms for the Caltech-Gabor dataset (with seven clusters and 34 samples in each cluster) is shown in Fig. 11. The evolution of the H-NAC cost function $Q_H(t)$ in (33) is shown in Fig. 12, which confirms Proposition 1. As seen from Fig. 11, the proposed H-NAC gives comparable performance to the SGL [46] and better performance than the other algorithms for the subset of the TR45 dataset.

VI. Conclusion and Future Work

In this section, we summarize the main conclusions of our article as follows.

1) This article extends the EM formulation for the GMM with a novel weighted dissimilarity loss. This extension results in the fusion of two different clustering methods, namely, centroid-based clustering and graph clustering, resulting in a simple "soft" asynchronous hybrid clustering method. The proposed algorithm may start as a pure centroid-based clustering algorithm (e.g., k-means), and as the time evolves, it may eventually and gradually turn into a pure graph clustering algorithm (e.g., basic GADIA [2]) as the algorithm converges and vice versa.
2) The proposed H-NAC takes both the intracluster distances (dissimilarities) and the intercluster dissimilarities chosen by the user simultaneously and explicitly into account during optimization. In short, the H-NAC simultaneously both minimizes the intracluster distances and maximizes the intercluster dissimilarities (chosen by the user) during optimization.
3) The "hard" version of the proposed algorithm is called H-NAC and includes the standard HNNs (and, thus, Bruck's Ln algorithm in [1] and the Ising model in statistical mechanics), Babadi and Tarokh's basic GADIA in [2], and the standard k-means [3], [4] (and the Lloyd [5], [6] algorithm) as its special cases.
4) We apply the proposed H-NAC algorithm to various using well-known benchmark datasets. The computer simulations results confirm the findings and the superior performance of the H-NAC compared to the k-means clustering, k-GADIA, spectral clustering, and one of the very-recent state-of-the-art clustering algorithm SGL in [46].
5) We extend Bruck's Ln algorithm [1] from a two-cluster case to an arbitrary k-cluster case, which is called the k-Ln algorithm in this article. The extended k-Ln clustering algorithm turns out to be equivalent to the basic version of the pioneering algorithm GADIA of Babadi and Tarokh [2].

As far as the implementation of the proposed H-NAC is concerned, it is actually like the k-means clustering by taking also the pairwise dissimilarities into account at the cost of an additional computational burden. Because the dissimilarities are precalculated offline and are read from a lookup table, the H-NAC's computational cost is slightly higher than the k-means clustering. This implies that the possible imminent applications of the proposed H-NAC are abundant in signal processing, computer vision, astronomy, image processing, machine learning, and so on. Due to the multidisciplinary nature of this article's research topic touching HNNs and optimization in computer science, Ising model in statistical mechanics, interference avoidance in wireless communications systems, k-means clustering, graph clustering, and feature extraction in machine learning, the presented results in this article allow us to build new connections among all these different areas. In this article, we present the simulation results only for the hard version of the proposed hybrid clustering, called the H-NAC. Our current and near-future research subjects include examining the performance of the soft version of the proposed hybrid clustering (presented in Section III). Last but not least, in this article, we focus on the basic versions of the k-means clustering. However, the variants of k-means and variants of the k-GADIA (i.e., k-Ln) can also be combined in the same way by the proposed hybrid model.

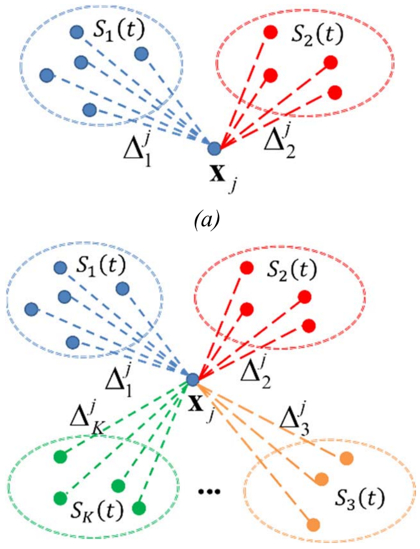
Appendix A
Extension of Bruck's Ln Algorithm in [1]

In what follows, we extend Bruck's Ln algorithm in [1] to arbitrary k-cluster case, as shown in Fig. 13.

Let us assign N data samples to K clusters, and let the sets $\{S_k\}_{k=1}^{K} = \{S_1, S_2, \ldots, S_K\}$ represent the cluster sets, i.e., the indices of data samples in each cluster. The cost function to be minimized by the k-Ln clustering is $Q_G(\{S_k(t)\}_{k=1}^{K})$ in (27), that is, the sum of intracluster *dissimilarities* over all clusters.

Bruck's Ln algorithm in his 1990 article [1] showed an alternative proof for the convergence of the *discrete-time* real HNN. Its working principle is sketched in Fig. 13(a). Because the HNN has two possible output values (−1 and +1), it represents a two-cluster case, which yields the following cluster update:

$$s_j = \text{index min}\left\{\Delta_1^j, \Delta_2^j\right\}. \tag{48}$$



(a)

(b)

Fig. 13. (a) Illustration of Bruck's Ln algorithm in [1] and (b) its extension to an arbitrary k-cluster case (i.e., the k-Ln algorithm) for an arbitrary data sample j at an arbitrary time t.

TABLE VII

k-Ln Algorithm (Simple GADIA [2])

Initialize the cluster assignments randomly. For an arbitrary sample $\boldsymbol{x}_j$ at an arbitrary time t: Let the cluster of sample $\boldsymbol{x}_j$be s_j^p.
Step (1) E-Step of k-Ln: Calculate the local dissimilarities $\Delta_k^j(t)$ for sample $\boldsymbol{x}_j$ by (21): $\{\Delta_1^j(t), \Delta_2^j(t), \cdots, \Delta_k^j(t)\}$.
Step (2): M-Step: The cluster index of sample j is updated at time *(t+1)* according to the following GADIA-like simple rule: $s_j = index_k \, min \, \{\Delta_k^j\}_{k=1}^{k=K} \quad (49)$ where s_j^p is the cluster index in the previous step. If $s_j \neq s_j^p$, (i.e. if the sample j is assigned to a different cluster s_j^p), then the corresponding two cluster sets $S_{s_j^p}(t+1)$ and $S_{s_j}(t+1)$ in (20) are updated accordingly (and all other cluster sets remain the same).
Step (3): Continue step (1) to (2) *asynchronously* for each sample one by one until there is no cluster change any more, reaching a local minima.

The extension of the Ln algorithm in [1] to an arbitrary k-cluster case is called the k-Ln algorithm and is sketched in Fig. 13(b) for an arbitrary data sample j at an arbitrary time t. The steps of the k-Ln algorithm are given in Table VII.

Corollary 1: Given a dataset $\{\boldsymbol{x}_i\}_{i=1}^N \in \Re^L$ in a deterministic setting, the k-Ln algorithm in Table VII above

1) Locally $\underline{minimizes}$ the sum of intracluster dissimilarities $\underline{Q_G(t) \text{ in } (27)}$.
2) Locally $\underline{maximizes}$ the k-cut for the graph $\mathbf{W}$.
3) Is equivalent to the basic version of the pioneering algorithm GADIA of Babadi and Tarokh [2].

The proof of Corollary 1 is obvious in the light of the findings in Section III.

Appendix B
Connection Between the k-Ln Algorithm and Spectral Clustering

Defining the cluster indicator matrix $\mathbf{H}$ as follows:

$$\mathbf{H} = [h_{ij}] = \begin{cases} 1, & \text{if sample } i \text{ is in } C_j \\ 0, & \text{otherwise} \end{cases} \tag{50}$$

where $i = 1, \ldots, N$, and $j = 1, \ldots, k$, from [16] and [53], and it is straightforward to obtain the following:

$$\max\text{Cut}_{C_1,\ldots,C_L}\{\mathbf{W}\} = \max\{\text{Tr}(\boldsymbol{H}^T \boldsymbol{L} \boldsymbol{H})\} \tag{51}$$

where $\text{Tr}(\cdot)$ represents trace operation for a matrix and $\mathbf{L}$ is the unnormalized Laplacian matrix of the dissimilarity matrix $\mathbf{W}$ (for further details, see [16] and [53]). Equation (51) is a standard trace maximization problem, and relaxing the discrete-solution constraint of (50), it is well-known from linear algebra that the optimum solution is given by choosing $\boldsymbol{H}$ as the matrix, which contains the greatest k eigenvectors of the unnormalized Laplacian matrix $\boldsymbol{L}$ as columns (e.g., [53]). The fact that the k-Ln algorithm partitions the graph $\mathbf{W}$ maximizing the sum of the intercluster dissimilarities (as shown in Corollary 1 in Appendix A) implies a connection between the k-Ln (i.e., k-GADIA) and the spectral clustering formulation.

References

[1] J. Bruck, "On the convergence properties of the Hopfield model," *Proc. IEEE*, vol. 78, no. 10, pp. 1579–1585, Oct. 1990.

[2] B. Babadi and V. Tarokh, "GADIA: A greedy asynchronous distributed interference avoidance algorithm," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6228–6252, Dec. 2010.

[3] H. Steinhaus, "Sur la division des corps matériels en parties," *Bull. Acad. Polonaise Sci.*, vol. 4, no. 12, pp. 801–804, 1956.

[4] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1, 1967, pp. 281–297.

[5] S. P. Lloyd's, "Least square quantization in PCM," Bell Lab., Dept. Math. Res., Holmdel, NJ, USA, Tech. Rep., Jul. 1957, p. 136.

[6] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[7] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, 1982.

[8] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 55, no. 3, pp. 141–146, 1985.

[9] Z. Uykan, "On the working principle of the Hopfield neural networks and its equivalence to the GADIA in optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3294–3304, Sep. 2020.

[10] Z. Uykan, "Shadow-cuts minimization/maximization and complex Hopfield neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1096–1109, Mar. 2021.

[11] S. E. Schaeffer, *Survey: Graph Clustering, Computer Science Review I*. Amsterdam, The Netherlands: Elsevier, 2007, pp. 27–64.

[12] E. W. Forgy, "Cluster analysis of multivariate data: Efficiency versus interpretability of classifications," *Biometrics*, vol. 21, no. 3, pp. 768–769, 1965.

[13] A. Coates and A. Y. Ng, "Learning feature representations with k-means," in *Neural Networks: Tricks Trade* (Lecture Notes in Computer Science), vol. 7700, G. Montavon, G. B. Orr, and K.-R. Muller, Eds., 2nd ed. Berlin, Germany: Springer, 2012.

[14] A. Coates, H. Lee, and A. Ng, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.

[15] N. Biggs, E. Lloyd, and R. Wilson, *Graph Theory*. New York, NY, USA: Oxford Univ. Press, 1986.

[16] Z. Uykan, "N-GAIR: Non-greedy asynchronous interference reduction algorithm in wireless networks," *Ad Hoc Sensor Wireless Netw.*, vol. 23, nos. 1–2, pp. 93–116, 2014.

[17] Z. Uykan and R. Jantti, "Transmission-order optimization for bidirectional device-to-device (D2D) communications underlaying cellular TDD networks—A graph theoretic approach," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 1–14, Jan. 2016.

[18] Z. Uykan and R. Jantti, "Joint optimization of transmission-order selection and channel allocation for bidirectional wireless links—Part II: Algorithms," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3991–4002, Jul. 2014.

[19] T. Ma and A. Ng, *CS229 Machine Learning Lecture Notes*. Stanford, CA, USA: Stanford Univ., 2019.

[20] *The 20 Newsgroups Data Set*. (Jun. 2021). [Online]. Available: http://qwone.com/jason/20Newsgroups/

[21] Z. Uykan and M. C. Ganiz, "Application of the spechybrid algorithm to text document clustering problem," in *Proc. Int. Symp. Innov. Intell. Syst. Appl. (INISTA)*, Istanbul, Turkey, Jun. 2011, pp. 118–122.

[22] Z. Uykan, M. C. Ganiz, and Ç. Şahinli, "Discrete-time Hopfield neural network based text clustering algorithm," in *Neural Information Processing* (Lecture Notes in Computer Science), vol. 7663. Berlin, Germany: Springer, 2012, pp. 551–559.

[23] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2004, pp. 225–232.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Gaussian mixture models and k-means clustering," in *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. New York, NY, USA: Cambridge Univ. Press, 2007, sec. 16.1.

[25] X.-D. Wang, R.-C. Chen, F. Yan, Z.-Q. Zeng, and C.-Q. Hong, "Fast adaptive K-means subspace clustering for high-dimensional data," *IEEE Access*, vol. 7, pp. 42639–42651, 2019, doi: 10.1109/ACCESS.2019.2907043.

[26] T.-W. Chen and S.-Y. Chien, "Bandwidth adaptive hardware architecture of K-means clustering for video analysis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 6, pp. 957–966, Jun. 2010.

[27] T.-W. Chen and S.-Y. Chien, "Flexible hardware architecture of hierarchical K-means clustering for large cluster number," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 8, pp. 1336–1345, Aug. 2011.

[28] H. Rong, A. Ramirez-Serrano, L. Guan, and Y. Gao, "Image object extraction based on semantic detection and improved K-means algorithm," *IEEE Access*, vol. 8, pp. 171129–171139, 2020.

[29] M. Alhawarat and M. Hegazi, "Revisiting K-means and topic modeling, a comparison study to cluster Arabic documents," *IEEE Access*, vol. 6, pp. 42740–42749, 2018.

[30] X. Wang, Y. Li, M. Wang, Z. Yang, and H. Dong, "An improved K-means algorithm for document clustering based on knowledge graphs," in *Proc. 11th Int. Congr. Image Signal Process., Biomed. Eng. Informat.*, Beijing, China, Oct. 2018, pp. 1–5.

[31] N. Garg and R. Rani, "Analysis and visualization of Twitter data using K-means clustering," in *Proc. Int. Conf. Intell. Comput. Control Syst.*, Madurai, India, Jun. 2017, pp. 670–675.

[32] A. Alsayat and H. El-Sayed, "Social media analysis using optimized K-means clustering," in *Proc. IEEE 14th Int. Conf. Softw. Eng. Res., Manage. Appl.*, Towson, MD, USA, Jun. 2016, pp. 61–66.

[33] M.-S. Yang and K. P. Sinaga, "A feature-reduction multi-view K-means clustering algorithm," *IEEE Access*, vol. 7, pp. 114472–114486, 2019, doi: 10.1109/ACCESS.2019.2934179.

[34] Y. Jing and J. Wang, "Tag clustering algorithm LMMSK: Improved K-means algorithm based on latent semantic analysis," *J. Syst. Eng. Electron.*, vol. 28, no. 2, pp. 374–384, Apr. 2017.

[35] X. Yang, Y. Wang, and W. Qiao, "Social network analysis on Sina Weibo based on K-means algorithm," in *Proc. IEEE Int. Conf. Cloud Comput. Big Data Anal.*, Chengdu, China, Jul. 2016, pp. 127–132.

[36] X. Yang, Y. Wang, D. Wu, and A. Ma, "K-means based clustering on mobile usage for social network analysis purpose," in *Proc. 6th Int. Conf. Adv. Inf. Manage. Service*, Seoul, South Korea, 2010, pp. 223–228.

[37] M. Mignotte, "Segmentation by fusion of histogram-based K-means clusters in different color spaces," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 780–787, May 2008.

[38] Y. Donghong, Y. Dongyang, and L. Hui, "The evaluation on oil company marketing force based on K-means cluster," in *Proc. Int. Conf. Inf. Manage., Innov. Manage. Ind. Eng.*, Taipei, Taiwan, Dec. 2008, pp. 403–406.

[39] K. Jain and S. Agrawal, "A hybrid approach for spam filtering using local concentration based K-means clustering," in *Proc. 5th Int. Conf. Confluence Next Gener. Inf. Technol. Summit*, Noida, India, Sep. 2014, pp. 194–199.

[40] S. H. Shihab, S. Afroge, and S. Z. Mishu, "RFM based market segmentation approach using advanced K-means and agglomerative clustering: A comparative study," in *Proc. Int. Conf. Electr., Comput. Commun. Eng.*, Cox's Bazar, Bangladesh, Feb. 2019, pp. 1–4.

[41] J. S. Almeida, J. A. L. Aguerri, C. M. Tuñón, and A. de Vicente, "Automatic unsupervised classification of all sloan digital sky survey data release 7 galaxy spectra," *Astrophys. J.*, vol. 714, no. 1, p. 487, 2010.

[42] T. Zhang, F. Ma, D. Yue, C. Peng, and G. M. P. ÒHare, "Interval type-2 fuzzy local enhancement based rough K-means clustering considering imbalanced clusters," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 9, pp. 1925–1939, Sep. 2020.

[43] I. V. Pustokhina *et al.*, "Automatic vehicle license plate recognition using optimal K-means with convolutional neural network for intelligent transportation systems," *IEEE Access*, vol. 8, pp. 92907–92917, 2020.

[44] M. Capo, A. Perez, and J. A. Lozano, "A cheap feature selection approach for the K-means algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 2195–2208, May 2021, doi: 10.1109/TNNLS.2020.3002576.

[45] C. Boutsidis and M. Magdon-Ismail, "Deterministic feature selection for K-means clustering," *IEEE Trans. Inf. Theory*, vol. 59, no. 9, pp. 6099–6110, Sep. 2013.

[46] Z. Kang, Z. Lin, X. Zhu, and W. Xu, "Structured graph learning for scalable subspace clustering: From single view to multiview," *IEEE Trans. Cybern.*, early access, Mar. 17, 2021, doi: 10.1109/TCYB.2021.3061660.

[47] Z. Wang, Z. Li, R. Wang, F. Nie, and X. Li, "Large graph clustering with simultaneous spectral embedding and discretization," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jun. 15, 2020, doi: 10.1109/TPAMI.2020.3002587.

[48] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[49] J. Li, H. Liu, Z. Tao, H. Zhao, and Y. Fu, "Learnable subspace clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 14, 2020, doi: 10.1109/TNNLS.2020.3040379.

[50] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 849–856.

[51] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1669–1680, Aug. 2015.

[52] Y. Wang, L. Wu, X. Lin, and J. Gao, "Multiview spectral clustering via structured low-rank matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4833–4843, Oct. 2018, doi: 10.1109/TNNLS.2017.2777489.

[53] U. V. Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[54] U. V. Luxburg, M. Belkin, and O. Bousquet, "Consistency of spectral clustering," *Ann. Statist.*, vol. 36, pp. 555–586, Apr. 2008.

[55] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1601–1608.

[56] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1796–1808, Nov. 2011.

[57] C. Hou, F. Nie, D. Yi, and D. Tao, "Discriminative embedded clustering: A framework for grouping high-dimensional data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1287–1299, Jun. 2015.

[58] H. Tao, C. Hou, F. Nie, Y. Jiao, and D. Yi, "Effective discriminative feature selection with nontrivial solution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 796–808, Apr. 2016.

[59] X. Cai, F. Nie, H. Huang, and F. Kamangar, "Heterogeneous image feature integration via multi-modal spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1977–1984.

[60] X. Chen, X. Xu, J. Z. Huang, and Y. Ye, "TW-k-means: Automated two-level variable weighting clustering algorithm for multiview data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 932–944, Apr. 2013.

[61] A. Khan and P. Maji, "Multi-manifold optimization for multi-view subspace clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 19, 2021, doi: 10.1109/TNNLS.2021.3054789.

[62] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1116–1129, Jun. 2020.

[63] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2887–2895, Oct. 2018.

[64] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2149–2155.

[65] Y. Chen, X. Xiao, Z. Hua, and Y. Zhou, "Adaptive transition probability matrix learning for multiview spectral clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 2, 2021, doi: 10.1109/TNNLS.2021.3059874.

[66] J. Wu, Z. Lin, and H. Zha, "Essential tensor learning for multi-view spectral clustering," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5910–5922, Dec. 2019.

[67] Z. Kang, H. Pan, S. C. H. Hoi, and Z. Xu, "Robust graph learning from noisy data," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1833–1843, May 2020.

[68] C. Zhang *et al.*, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Jan. 2020.

[69] X. Zhang, H. Sun, Z. Liu, Z. Ren, Q. Cui, and Y. Li, "Robust low-rank kernel multi-view subspace clustering based on the schatten p-norm and correntropy," *Inf. Sci.*, vol. 477, pp. 430–447, Mar. 2019.

[70] Y. Chen, X. Xiao, and Y. Zhou, "Jointly learning kernel representation tensor and affinity matrix for multi-view clustering," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 1985–1997, Aug. 2020.

[71] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[72] J. Fan, Z. Tian, M. Zhao, and T. W. S. Chow, "Accelerated low-rank representation for subspace clustering and semi-supervised classification on large-scale data," *Neural Netw.*, vol. 100, pp. 39–48, Apr. 2018.

[73] F. Nie, C.-L. Wang, and X. Li, "K-multiple-means: A multiple-means clustering method with specified k clusters," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 959–967.

[74] F. Pourkamali-Anaraki, J. Folberth, and S. Becker, "Efficient solvers for sparse subspace clustering," *Signal Process.*, vol. 172, Jul. 2020, Art. no. 107548.

[75] X. Chen, W. Hong, F. Nie, D. He, M. Yang, and J. Z. Huang, "Spectral clustering of large-scale data by directly solving normalized cut," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1206–1215.

[76] C. You, D. P. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3918–3927.

[77] F. Nie, J. Li, and X. Li, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *Proc. IJCAI*, 2016, pp. 1881–1887.

[78] M. Brbić and I. Kopriva, "Multi-view low-rank sparse subspace clustering," *Pattern Recognit.*, vol. 73, pp. 247–258, Jan. 2018.

[79] X. Wang, Z. Lei, X. Guo, C. Zhang, H. Shi, and S. Z. Li, "Multi-view subspace clustering with intactness-aware similarity," *Pattern Recognit.*, vol. 88, pp. 50–63, Apr. 2019.

[80] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, pp. 4412–4419.

[81] X. Chen, R. Chen, Q. Wu, Y. Fang, F. Nie, and J. Z. Huang, "LABIN: Balanced min cut for large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 3, pp. 725–736, Mar. 2020.

[82] Q. Wang, Z. Qin, F. Nie, and X. Li, "Spectral embedded adaptive neighbors clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1265–1271, Apr. 2018.

[83] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, and Y. Yang, "Rank-constrained spectral clustering with flexible embedding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6073–6082, Dec. 2018.

[84] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[85] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.

[86] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 11, no. 9, pp. 1074–1085, Sep. 1992.

[87] S. X. Yu and J. Shi, "Multiclass spectral clustering," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 1, Oct. 2003, pp. 313–319.

[88] F. Nie, X. Wang, M. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1969–1976.

[89] X. Chen, J. Z. Huang, F. Nie, R. Chen, and Q. Wu, "A self-balanced min-cut algorithm for image clustering," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2061–2069.

[90] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[91] M. C. P. de Souto, I. G. Costa, D. S. A. de Araujo, T. B. Ludermir, and A. Schliep, "Clustering cancer gene expression data: A comparative study," *BMC Bioinform.*, vol. 9, no. 1, p. 497, 2008.

[92] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 1, pp. 1–37, 2009.

[93] M. Soltanolkotabi, E. Elhamifar, and E. J. Candès, "Robust subspace clustering," *Ann. Statist.*, vol. 42, no. 2, pp. 669–699, 2014.

[94] S. Li, K. Li, and Y. Fu, "Temporal subspace clustering for human motion segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4453–4461.

[95] S. Tierney, J. Gao, and Y. Guo, "Subspace clustering for sequential data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1019–1026.

[96] H. E. Cetingul, M. J. Wright, P. M. Thompson, and R. Vidal, "Segmentation of high angular resolution diffusion MRI using sparse Riemannian manifold clustering," *IEEE Trans. Med. Imag.*, vol. 33, no. 2, pp. 301–317, Feb. 2014.

[97] T. Wallace, A. Sekmen, and X. Wang, "Application of subspace clustering in DNA sequence analysis," *J. Comput. Biol.*, vol. 22, no. 10, pp. 940–952, Oct. 2015.

[98] G. Liu, Z. Zhang, Q. Liu, and H. Xiong, "Robust subspace clustering with compressed data," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5161–5170, Oct. 2019.

[99] C.-G. Li, C. You, and R. Vidal, "Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2988–3001, Jun. 2017.

[100] L. Liu, M. Yu, and L. Shao, "Multiview alignment hashing for efficient image search," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 956–966, Mar. 2015.

[101] L. Shao, L. Liu, and M. Yu, "Kernelized multiview projection for robust action recognition," *Int. J. Comput. Vis.*, vol. 118, no. 2, pp. 115–129, 2016.

Zekeriya Uykan (Senior Member, IEEE) received the B.S. and M.S. degrees (Hons.) in electrical engineering from Istanbul Technical University (ITU), Istanbul, Turkey, in 1993 and 1996, respectively, and the Lic.Tech. (Hons.) and Dr.Tech. degrees in electrical engineering from the Helsinki University of Technology (HUT, now Aalto University), Espoo, Finland, in 1998 and 2001, respectively.

He was a Research and Teaching Assistant with ITU, from April 1994 to September 1996, and also a Research Scientist with HUT, from September 1996 to February 2001. He was with the NOKIA Research Center (now Nokia-Bell Labs), Radio Communications Laboratory, Helsinki, Finland, from July 2000 to April 2007, and Nokia Siemens Networks (now Nokia Solutions and Networks), Espoo, from April 2007 to December 2010. He was a Visiting Researcher with Sabanci University, Istanbul, from May 2000 to July 2000; a Visiting Scholar with Stanford University, Stanford, CA, USA, from January 2002 to December 2002; a Visiting Scientist with Harvard University, Cambridge, MA, USA, from September 2008 to June 2009; and a Visiting Associate Professor with the COMNET Department, School of Electrical Engineering, Aalto University, Espoo, from August 2012 to September 2012 and from February 2013 to August 2013. He was with Dogus University, Istanbul, from September 2009 to September 2015. He has been with the American University of the Middle East, Egaila, Kuwait, since September 2014. His current research interests include deep learning, artificial intelligence in wireless communications, quantum mechanics, and quantum communication.