electronics

MDPI

Article

Distributed Ensemble Clustering in Networked Multi-Agent Systems

Nemanja Ilić [1,2,*] and Marija Punt [3]

[1] Department of Information Technologies, College of Applied Technical Sciences, 37000 Kruševac, Serbia
[2] School of Computing, Union University, 11000 Belgrade, Serbia
[3] School of Electrical Engineering, University of Belgrade, 11000 Belgrade, Serbia; marija.punt@etf.rs
* Correspondence: nilic@raf.rs

Abstract: Ensemble clustering, a paradigm that deals with combining the results of multiple clusterings into a single solution, has been widely studied in recent years. The goal of this study is to propose a novel distributed ensemble clustering method that is applicable for use in networked multi-agent systems. The adopted setting supports both object-distributed and feature-distributed clusterings. It is not limited to specific types of algorithms used for obtaining local data labels. The method assumes local processing of local data by the individual agents and neighbor-wise communication of the processed information between the neighboring agents in the network. Using the proposed communication scheme, all agents are able to achieve reliable global results in a fully decentralized way. The network communication design is based on the multi-agent consensus averaging algorithm applied to clustering similarity matrices. It provably results in the fastest convergence to the desired asymptotic values. Several simulation examples illustrate the performance of the proposed distributed solution in different scenarios, including diverse datasets, networks, and applications within the multimedia domain. They show that the obtained performance is very close to that of the corresponding centralized solution.

Keywords: ensemble clustering; multi-agent systems; decentralized algorithm; consensus communication scheme; multimedia applications

check for updates

Citation: Ilić, N.; Punt, M. Distributed Ensemble Clustering in Networked Multi-Agent Systems. *Electronics* **2023**, *12*, 4558. https://doi.org/10.3390/electronics12224558

Academic Editor: Miin-shen Yang

Received: 16 September 2023
Revised: 18 October 2023
Accepted: 31 October 2023
Published: 7 November 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Clustering represents a prototypical unsupervised learning task within the fields of artificial intelligence and machine learning and has numerous and diverse applications. Its goal is to partition unlabeled data into groups or clusters so that data points in the same group are similar. This is typically achieved by introducing some measure of similarity, which is then optimized in a predefined way.

In view of the adopted clustering task, there are many situations where reiterating the clustering process and combining the obtained individual results is desirable or needed. Separate clustering iterations can result from different settings, starting from different initializations that may yield different labelings or from different numbers of target clusters [1,2]. Furthermore, individual clustering results may be obtained using different clustering algorithms. Moreover, individual clusterings can be associated with different datasets, e.g., obtained from the original dataset via random sampling with or without replacement [2,3]. This falls within the category of object-distributed clustering [4], which also encompasses more general cases when different individual datasets occur naturally as a consequence of the adopted clustering problem. Another category is feature-distributed clustering [4], where individual clusterings are based on particular aspects/views of data, i.e., specific sets of features or attributes for each data point.

Combining individual clustering results into a single solution has the potential to improve the quality, robustness, and stability of the overall results. The main motivation

for this approach lies in the known variable nature of the clustering processes; the more diverse the individual results are, the more benefits of combining them can be expected. There are even some theoretical results showing that it is not possible to make decisions regarding the clustering process, which would result in revealing the natural clusterings in the considered dataset, in advance [5]. Namely, it is provably impossible to design a clusterer that would be, at the same time, scale-invariant (not sensitive to scale changes affecting the distances between data points), rich (able to output any possible partitioning of the dataset), and consistent (able to output the same result when distances between points inside a cluster get shrunk and distances between points in different clusters get expanded). Therefore, since it is not possible to find a single optimal clustering solution, reiterating the clustering process might be the only solution.

1.1. Review of Related Ensemble Clustering Approaches

Reiterating and combining clusterings has been adopted within several similar, albeit slightly different, paradigms. The authors in [4] refer to an individual clustering algorithm with a specific view of the data as a clusterer, and define the *cluster ensemble* task as combining clusterer results without accessing the original data features, but only cluster labels. They also point to the fact that combining individual clustering results represents an inherently more difficult problem than combining classification results, as the label correspondence problem should be solved. On the other hand, the authors in [3] introduce the notion of *consensus clustering*, where the results across multiple runs of a clustering algorithm, applied to the resampled data of the same original dataset, are combined. Their focus is on determining the number of clusters and the corresponding confidence levels. The term consensus is used only to refer to the agreement of the individual results, and not in the context of consensus algorithms, such as those known in the multi-agent literature [6,7]. *Consensus aggregation* represents another similar concept [8], where different clusterings of the same dataset are combined in a way that minimizes the total number of induced disagreements between the individual clusterings and the final solution.

Classical clustering solutions perform data collection, preprocessing, and clustering within a single location. This centralized setting might not be suitable in situations where data are inherently being acquired and stored in different locations. Furthermore, the computational and bandwidth costs of centralized schemes might represent an issue. In today's world of big data, cloud storage, the Internet of Things, etc., it is clear that obtaining decentralized/distributed solutions represents an imperative. Distributed methods should allow for the efficient handling of large volumes of data. If properly designed, they should also provide means for addressing the critical issues of data security, privacy, access rights, and access to heterogeneous data. In this way, the corresponding solutions also fit well within the decentralized federated learning framework, e.g., [9]. The fitting is naturally achieved when using the cluster ensemble setting [4], as it assumes that data are processed in situ, and only compressed information is used for obtaining the final results. Decentralized methods represent fault-tolerant solutions since they do not rely on a centralized entity, which is a potential focal vulnerability point of the system.

The existing decentralized and other relevant cluster ensemble algorithms typically adopt specific restrictive assumptions or have a relatively limited scope. Several papers focus only on the expectation–maximization algorithm; some propose novel consensus-based distributed schemes [10,11], while [12] proposes a solution based on a majority voting scheme. The authors in [13] propose a geometric approach where individual clusterers observe the same set of objects. A scaling solution for very large datasets represented by a centroid-based ensemble merging algorithm has been proposed in [14]. It relies on the centralized processing of centroid information. The authors in [15] assume object-distributed clustering, the communication of prototypes of local clusters, and clustering validity indices to avoid conflicts.

1.2. Proposed Solution

In this study, the cluster ensemble formulation [4] is adopted, as it represents the most general setting, allowing for different algorithms, as well as for both object and feature-distributed clustering schemes. The goal is to propose a novel distributed and decentralized ensemble clustering algorithm based on a consensus scheme that belongs to a wider class of consensus algorithms that are well studied in the multi-agent literature [6,7,16]. This scheme will serve as a model for the communication protocol, where individual clusterers exchange their clustering results in a neighbor-wise fashion. The setting will not assume all-to-all communications [15]. The focus will be on the so-called similarity [4] (also known as connectivity [3] or co-association [17]) matrices—square matrices with binary entries equal to 1 if two data points, indexing their rows and columns, fall within the same cluster, and to 0 otherwise. Specifically, it will be demonstrated how the corresponding cluster ensemble similarity matrix, representing typically an average (or weighted average) of the individual similarity matrices, can be obtained in a distributed manner. In some papers, e.g., [2,3], the cluster ensemble similarity matrix is also called a consensus matrix. Herein, this denomination will not be used, so as to avoid confusion, since the term consensus matrix will be used in reference to the communication scheme in the multi-agent network. In general, in this paper, consensus will be referred to in the context of a communication scheme between individual clusterers. This is usual within the literature in the field of multi-agent systems.

The proposed algorithm can serve as a framework for various solutions—it neither requires a specific clusterer design nor a specific algorithm that yields the final clustering results. Its only limitation is that it cannot be used in cluster ensembles that combine local results based on variables other than similarity matrices. The potential scaling problems with respect to the data size can be alleviated using different strategies, similarly as in, e.g., [8,17,18]. It should be noted that comparative studies [2] show that the best approaches for cluster ensembles use similarity matrices for obtaining the final clustering results. Similarity matrices remain central points of many more recent ensemble clustering studies [18–20].

Due to its general setting supporting feature-distributed clustering, the proposed solution is especially important for multimedia applications, with multi-view and multi-modal data, ubiquitous in the modern world. The corresponding datasets might originate from video data sources and include speech transcripts and results of image processing, or from image data sources and include text description features along with the features obtained by image processing. In this domain, parallel hierarchical architectures based on divide-and-conquer strategies have been proposed [21]. These solutions yield final clustering results faster than the original centralized solutions, but not in an inherently decentralized way.

The structure of this paper is as follows: Section 2 introduces the considered ensemble clustering problem and the used similarity matrices. It also includes an illustrative example and describes the corresponding centralized solution. In Section 3, the proposed distributed algorithm is presented, focusing on its structural design and emphasizing the multi-agent communication scheme and its optimal configuration. The results of the numerical experiments are given in Section 4, with three examples illustrating the properties of the proposed algorithm and its performance. Section 5 gives concluding remarks and discusses topics for further research.

2. Problem Formulation

2.1. Ensemble Clustering

Let the clustering problem at hand deal with the dataset corpus of N objects or data points: $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$. Its goal is to partition the considered dataset into a set of clusters, exhaustive and non-overlapping. A partitioning into K clusters can be represented as a set of K sets of objects: $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$, such that $\cup_{k=1}^{K} \mathcal{C}_k = \mathcal{X}$ and $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$, for all i and j, such that $i \neq j$. Alternative notation assumes a label vector

$\lambda \in \mathbb{N}^N$, of which the entries corresponding to data points from the same cluster share the same value or label. The clustering function Φ represents a mapping of an input set of objects into an output set of labels: $\Phi : \mathcal{X} \to \lambda$. For example, a partitioning of a set of $N = 7$ objects $\mathcal{X} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ into $K = 3$ clusters may be represented as $\{\mathcal{C}_1 = \{x_1, x_2, x_3\}, \mathcal{C}_2 = \{x_4, x_5\}, \mathcal{C}_3 = \{x_6, x_7\}\}$, but also as $\lambda = (1, 1, 1, 2, 2, 3, 3)^T$. The cluster labels themselves hold no specific meaning: for example, an equivalent partitioning to the one given above can be $\{\mathcal{C}_1 = \{x_6, x_7\}, \mathcal{C}_2 = \{x_1, x_2, x_3\}, \mathcal{C}_3 = \{x_4, x_5\}\}$, or in alternative notation $\lambda = (2, 2, 2, 3, 3, 1, 1)^T$.

It is assumed that within the considered ensemble clustering setting, the clustering process is performed or reiterated M times, i.e., we are dealing with the set of M clusterers $\{\Phi^{(1)}, \Phi^{(2)}, \ldots, \Phi^{(M)}\}$. These M clusterers are dispatched to solve a common clustering problem at hand, regarding the same observed phenomenon. A general setting, allowing for both object and feature-distributed clustering, is adopted. More specifically, each clusterer's local dataset, denoted as $\mathcal{X}^{(m)}$, $m = 1, \ldots, M$, is assumed to represent a subset of the original dataset $\mathcal{X}$, with a possibly different set of features or attributes. Each clusterer performs its own clustering algorithm, allowing also for possibly different initial conditions, as well as a different number of target clusters. Therefore, we are dealing with the set of M mappings: $\Phi^{(m)} : \mathcal{X}^{(m)} \to \lambda^{(m)}$, $m = 1, \ldots, M$, where $\lambda^{(m)}$ denotes a label vector of the m-th clusterer. The task of ensemble clustering is then to combine or aggregate these M label vectors onto a single label vector λ in an optimal way, according to some predefined criterion, which can be defined in different ways [4,8,17].

2.2. Similarity Matrices

A common intermediary step toward combining individual clustering results $\lambda^{(m)}$, used in several of the best performing ensemble clustering approaches [2], is to map $\lambda^{(m)}$ onto the so-called similarity (or connectivity or co-association) matrices. Then, the aggregation of these matrices, typically obtained via entry-wise simple or weighted averaging, is used to infer the final clustering results represented by the label vector λ.

Similarity matrices in this context are usually based on the reasoning that two objects have a similarity of 1 if they are in the same cluster and a similarity of 0 if they are in different clusters. Consequently, an $N \times N$ similarity matrix can be created for each clustering. More formally, let the similarity matrix associated with $\lambda^{(m)}$ be denoted as $S^{(m)}$. Its entries can be obtained by

$$S^{(m)}(i, j) = \begin{cases} 1, & \text{if } \lambda^{(m)}(i) = \lambda^{(m)}(j) \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where $\lambda^{(m)}(i)$ denotes the i-th entry of the label vector $\lambda^{(m)}$. The straightforward way to aggregate individual similarity matrices into the resulting ensemble clustering similarity matrix S is to find their average:

$$S(i, j) = \frac{1}{M} \sum_{m=1}^{M} S^{(m)}(i, j). \tag{2}$$

For missing values, the simplest way is to adopt $S^{(m)}(i, j) = 0$ for all cases, except for $i = j$ when the corresponding value is 1. This definition of S has been used in [4].

The simplified way of treating missing values might be problematic, especially when the number of missing values cannot be neglected. Therefore, a better and more natural option would be not to take into account the missing values when calculating S. To this end, auxiliary indicator matrices need to be introduced. For each clusterer m, they are represented by $N \times N$ matrices whose entry at location (i, j) corresponds to the binary indicator of the presence of both data points x_i and x_j from the original dataset $\mathcal{X}$ in the clusterer's local dataset $\mathcal{X}^{(m)}$:

$$I^{(m)}(i,j) = \begin{cases} 1, & \text{if } \{x_i, x_j\} \in \mathcal{X}^{(m)} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Here, the similarity matrix S can be defined as [3]:

$$S(i,j) = \frac{\sum_{m=1}^{M} I^{(m)}(i,j) S^{(m)}(i,j)}{\sum_{m=1}^{M} I^{(m)}(i,j)}. \tag{4}$$

The term $I^{(m)}(i,j)$ is put in the nominator as well as in the denominator in (4) to make sure that the equation holds regardless of how missing values are treated in the similarity matrices, i.e., which values are given to the corresponding entries. Furthermore, the introduction of indicator matrices $I^{(m)}$ in the way it has been introduced in (4) allows for very important extensions of the algorithm. Namely, instead of binary entries, $I^{(m)}$ can include continuous scalar values that serve as starting points in the refinements of the clusterers' base results. These should be based on the level of confidence that the observed pair of data points belongs/does not belong to the same cluster, e.g., depending on the cluster size and dimensions of data points [22]. Other locally weighted schemes, e.g., aimed at weakening the effects caused by the outlier data points [18], are also possible, as well as weighted schemes where all entries of $I^{(m)}$ are set in the same way [23]. Finally, ensemble-driven locally weighted schemes [19], or maybe even schemes based on the adaptive graph filters [24], can be taken into account, although obtaining their distributed versions represents an additional research topic.

2.3. Illustrative Example

In order to clarify the existing subtleties and differences in calculating the average of the individual similarity matrices, reflected in two different approaches defined by (2) and (4), an appropriate illustrative example is given below, following [4], which would hopefully serve as a reference point for the subsequent discussion.

Let the dataset $\mathcal{X}$ consist of $N = 7$ objects, and let the following label vectors represent the clusterings of $M = 4$ clusterers:

$$\begin{aligned} \lambda^{(1)} &= (1, \ 1, \ 1, \ 2, \ 2, \ 3, \ 3)^T \\ \lambda^{(2)} &= (2, \ 2, \ 2, \ 3, \ 3, \ 1, \ 1)^T \\ \lambda^{(3)} &= (1, \ 1, \ 2, \ 2, \ 2, \ 3, \ 3)^T \\ \lambda^{(4)} &= (1, \ 2, \ ?, \ 1, \ 2, \ ?, \ ?)^T. \end{aligned} \tag{5}$$

Clusterers 1 to 3 have access to all the objects, i.e., $\mathcal{X}^{(i)} = \mathcal{X}$, $i = 1, 2, 3$, while clusterer 4 has access to $\mathcal{X}^{(4)} = \{x_1, x_2, x_4, x_5\}$. It can be seen that $\lambda^{(1)}$ and $\lambda^{(2)}$ are essentially the same clusterings, just with different label notations; $\lambda^{(3)}$ is similar but not the same, while $\lambda^{(4)}$ is not that similar and has three missing data points (denoted by question marks). Corresponding similarity matrices are illustrated in Figure 1. It can be seen that these visualizations align well with the manual inspection of the correspondence between individual clusterings; they can also be used as a tool for determining the number of clusters [3]. Figure 1 also shows that similarity matrices are naturally robust to different label notations used by different clusterers.

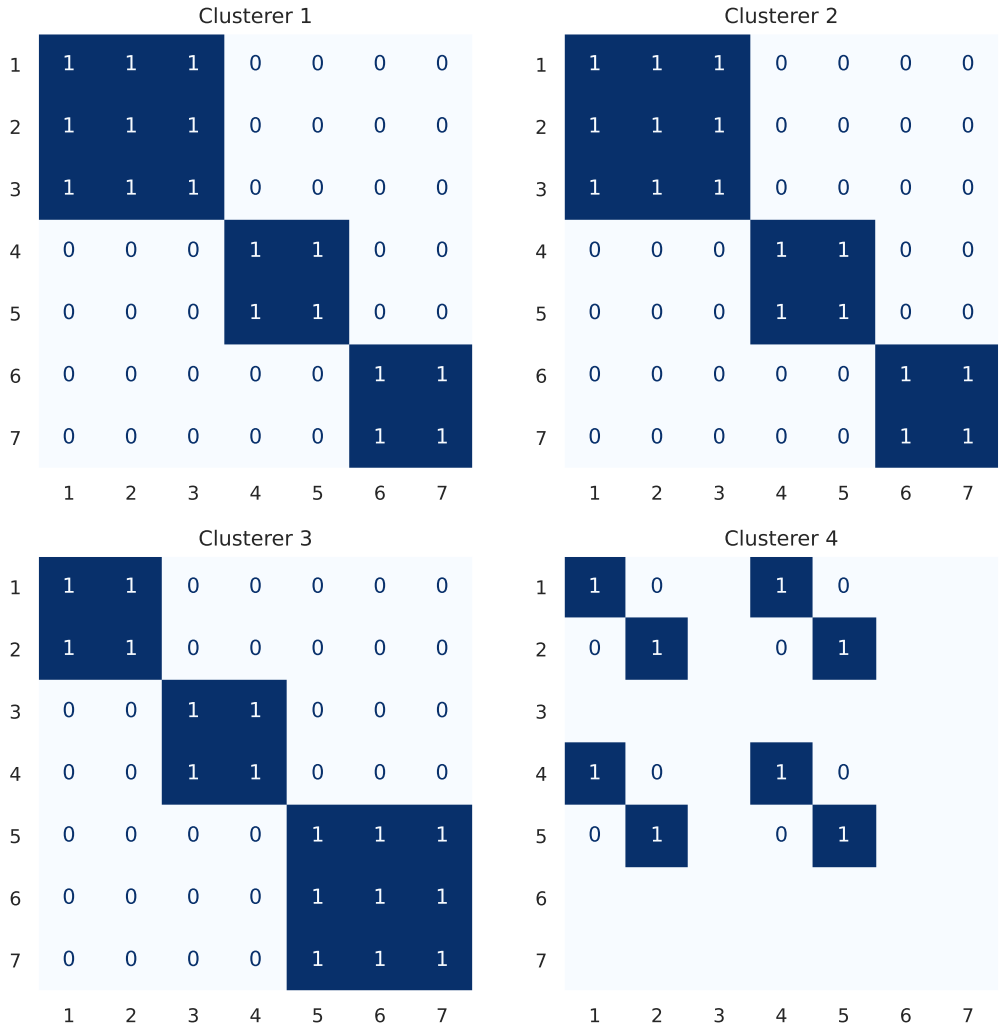

Figure 1. Four similarity matrices corresponding to four label vectors shown in (5). The entries are highlighted in different shades of blue, with the amount of color proportional to their values.

For label vectors in (5) and similarity matrices in Figure 1, the corresponding ensemble clustering similarity matrix S when calculated by (2) is illustrated in Figure 2 (left), while S calculated by (4) is illustrated in Figure 2 (right). Looking at, for example, data points 1 and 3 in Figures 1 and 2 (left), it can be seen that the corresponding entry in S is equal to 0.5, as these two objects occur in the same cluster for $m = 1, 2$, in different clusters for $m = 3$, while for $m = 4$, the information is not available, but the objects are treated as belonging to different clusters. However, looking at Figure 2 (right), it can be seen that, for the aforementioned data points 1 and 3, the corresponding value is 0.66, since, among the three clusterers that had observed these two points, two have put them in the same cluster. This is the main difference between (2) and (4).

In this study, two multi-agent distributed algorithms for calculating (2) and (4) will be proposed. Using (2) represents a simpler approach, while using (4) is an approach more appropriate for cases when clusterers have a relatively high number of missing values with respect to the original data corpus $\mathcal{X}$, which is typical for object-distributed clustering scenarios.

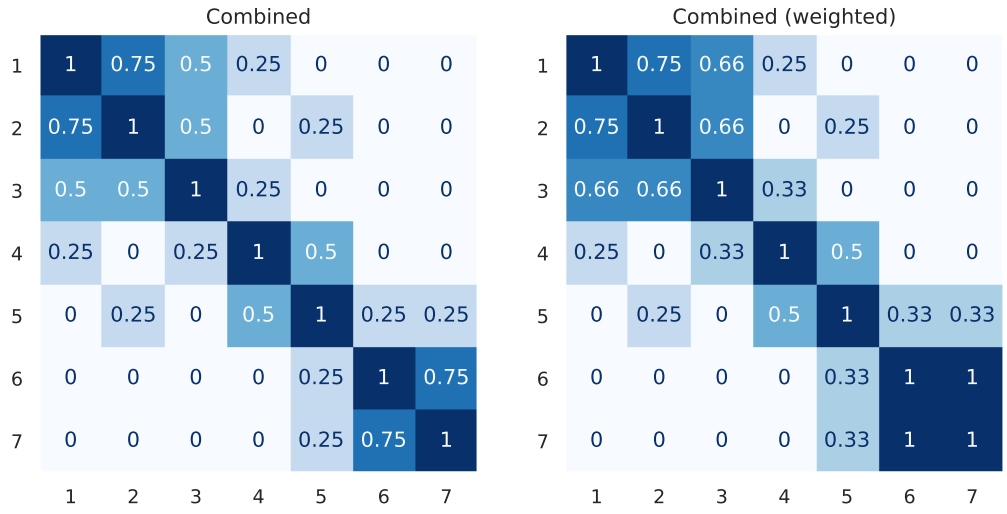

Figure 2. Aggregated similarity matrices corresponding to Figure 1. The entries are highlighted in different shades of blue, with the amount of color proportional to their values. (**Left**): simple averaging based on (2). (**Right**): weighted averaging based on (4).

2.4. Centralized Algorithm

Now that the notions of individual clusterers, their label vectors, corresponding similarity matrices, and their aggregation schemes are introduced, the process of ensemble clustering can be summarized in an appropriate way. When this whole process is being performed by a single entity, it represents a centralized algorithm, the flowchart of which is illustrated in Figure 3.

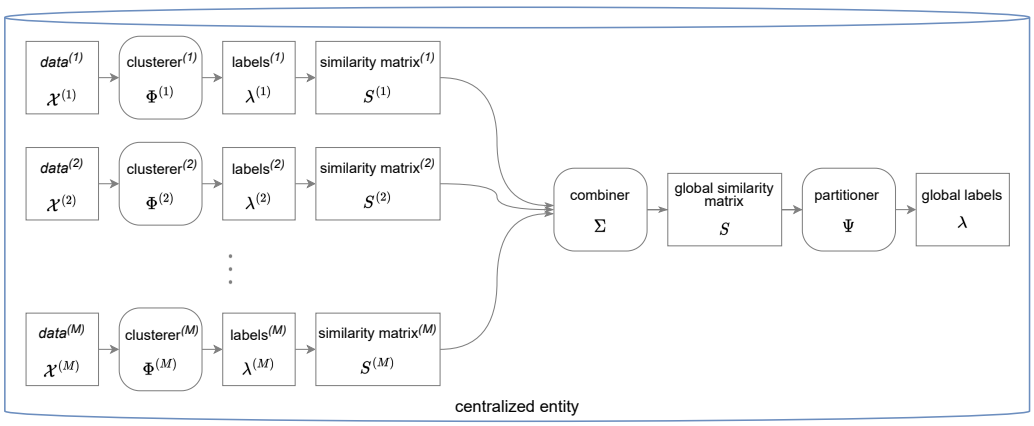

Figure 3. Flowchart of the centralized ensemble clustering process.

It can be seen that, within a single entity, a set of M datasets $\{\mathcal{X}^{(m)}, \ldots, \mathcal{X}^{(M)}\}$ has been collected and clustered by the corresponding clusterers $\{\Phi^{(1)}, \ldots, \Phi^{(M)}\}$. The obtained label vectors $\{\lambda^{(1)}, \ldots, \lambda^{(M)}\}$ are transformed into corresponding similarity matrices $\{S^{(1)}, \ldots, S^{(M)}\}$ using (1). The individual similarity matrices are combined into a single global similarity matrix S, using either (2) or (4); this combination process is denoted in the flowchart using the operator Σ. What remains is to obtain final ensemble clustering labels λ using the now available global similarity matrix S. This task is delegated to a block that shall be referred to as a partitioner (denoted in the flowchart as Ψ).

There are numerous algorithms documented in the literature for addressing the final partitioning task. These algorithms differ in the way they use the obtained similarity matrix S. In [4], within the proposed cluster-based similarity partitioning algorithm (CSPA), S serves as a basis for the induced similarity graph where objects represent vertices, while similarity values represent edge weights. This graph is then partitioned using the algorithm

based on the results from [25]. The authors in [2], in one of their approaches, use $1 - S$ as a distance matrix associated with the objects, and input it to a single linkage clustering algorithm, the output of which represents the set of final ensemble partitionings. Interestingly, they obtain the best performance when interpreting the matrix S as a data matrix, with N objects and N features, i.e., the i-th object's j-th feature is represented by $S(i, j)$. They apply several clustering algorithms on S as a data matrix: single linkage, mean linkage, and k-means [2]. Using similarities as feature values has been demonstrated to perform well in some other applications as well [26].

3. Distributed Algorithm

3.1. Structural Design

This study proposes a novel algorithm aimed at providing all agents in the considered multi-agent network, addressing the adopted ensemble clustering problem, with reliable labels for the whole dataset of interest. These global labels are to be obtained in a completely decentralized and distributed manner while being quantitatively very close to the result of a corresponding centralized algorithm.

To this end, each clusterer is associated with an agent within the adopted multi-agent setting. A flowchart of the proposed distributed ensemble clustering system is shown in Figure 4. Each agent, in general, represents an entity with its own data, processing, and communication capabilities. More specifically, the m-th agent, based on its local dataset $\mathcal{X}^{(m)}$, performs clustering by the use of a local clusterer $\Phi^{(m)}$, and obtains a local label vector $\lambda^{(m)}$, which it then transforms into a local similarity matrix $S^{(m)}$. Based on a communication scheme incorporating the entries of similarity matrices, which will be proposed in the following subsection, each agent obtains its own estimate of the global similarity matrix S (which should be close to the corresponding centralized solution). This similarity matrix can then be processed locally, using an appropriate partitioner Ψ, to come to the final label vector λ. In this way, every agent is provided with the global label vector λ in a fully decentralized way.

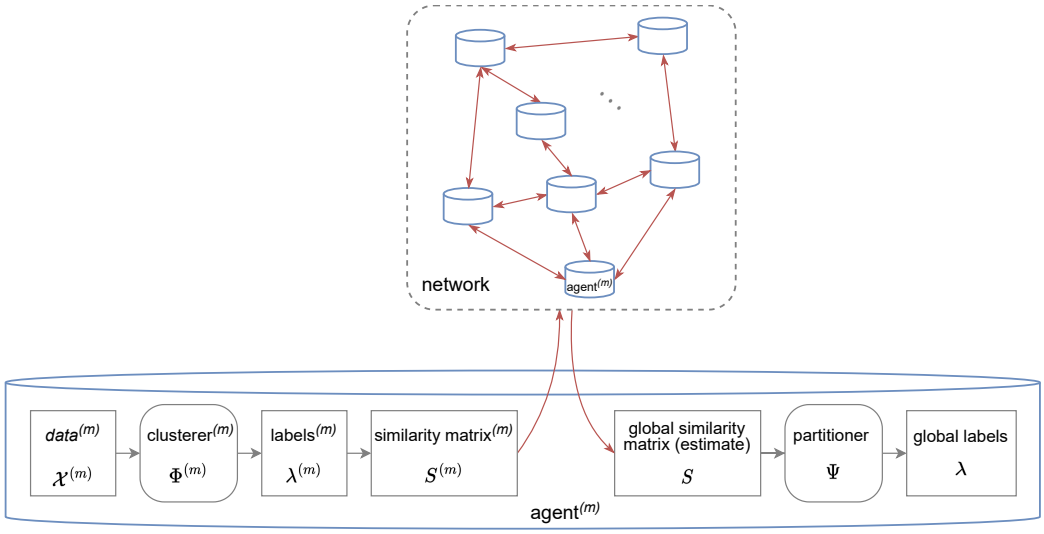

Figure 4. Flowchart of the proposed distributed ensemble clustering process.

It should be emphasized that the goal of this study is not to propose a specific ensemble clustering solution with specific choices and designs for the local datasets, clusterers, or partitioners. Instead, our aim is to propose a general distributed solution, which would, regardless of the aforementioned specifics, yield a result close to the result of a corresponding centralized algorithm. To the best of the authors' knowledge, a distributed ensemble clustering solution of this kind has not yet been proposed in the literature. The higher the

clustering performance of the centralized solution is, the higher the performance of the proposed distributed solution can be achieved.

Another point that should be noted, in order to allow for fair evaluations, is that the proposed solution has the potential limitation in that it is exclusively based on similarity matrices. However, this fact should not represent a conceptual problem, since the ensemble clustering methods based on similarity matrices are widely used [18–20], and exhibit the best performance in comparative studies [2]. Having in mind that the dimensions of similarity matrices correspond to the dataset size, in the case of extremely large datasets, potential scaling problems should be circumvented by the use of an appropriate scaling strategy. For example, only a certain number of nearest neighbors for each data point can be retained, which would, with the additional implied technicalities, reduce the effective size of S [17]. Subsampling strategies can also be applied [8]. Additionally, the size of S can be reduced by taking into account the so-called must-link information between data points, regarding the object pairs consistently labeled across all clusterers, which can then be excluded from S [18]. Making the distributed scheme compatible with spectral ensemble clustering approaches [27], which also enables good scaling properties, represents one of the tasks for future research.

3.2. Communication Scheme

In order to propose a novel communication scheme in the adopted multi-agent setting, which represents the core contribution of the paper, some formal notations need to be introduced. Each agent from the set of M agents mentioned in the previous subsection is assumed to be able to communicate the needed information (e.g., entries of the corresponding similarity matrix) with neighboring agents via a communication network. This inter-agent network is represented by a connected graph $\mathcal{G} = (\mathcal{M}, \mathcal{E})$, where $\mathcal{M} = \{1, \ldots, M\}$ is the set of nodes and $\mathcal{E}$ the set of edges; each edge is an unordered pair of distinct nodes $\{i, j\}$. The set of neighboring nodes of node i is denoted as $\mathcal{M}_i$, and $\mathcal{J}_i$ is defined as $\mathcal{M}_i \cup \{i\}$.

The inter-agent communication is modeled by the $M \times M$ consensus matrix C, whose entries represent the communication weights used as multiplicative factors for the exchanged quantities between different nodes. The sparsity pattern of C follows the network graph topology, i.e., $C(i, j) = 0$ if $j \notin \mathcal{J}_i$. In order to obtain results that are close to the corresponding centralized solution, the neighbor-wise communications are to be reiterated through the network multiple times, which are referred to as consensus steps, the total number of which is denoted as L.

In the following, two distributed algorithms are proposed, corresponding to distributed schemes for obtaining (2) and (4), respectively.

3.2.1. Algorithm 1

Obtaining a distributed variant of (2) is straightforward. We deal with $S^{(m)}(i, j)$, which denotes the i-th row and j-th column entry of the m-th agent's similarity matrix $S^{(m)}$, $m = 1, \ldots, M$, and $i, j = 1, \ldots, N$. It is assumed that the agents exchange information on the entries of similarity matrices in multiple iterations; these iterations of the communication scheme are denoted by a current consensus step number l, where $l = 1, \ldots, L$. The similarity matrix obtained after l steps of consensus is denoted as $S_l^{(m)}(i, j)$; the corresponding initial value is set as $S_0^{(m)}(i, j) = S^{(m)}(i, j)$. Furthermore, for compact representation, entries of different agents are concatenated as $\mathcal{S}_l(i, j) = (S_l^{(1)}(i, j), \ldots, S_l^{(M)}(i, j))^T$. Here, it can be written as

$$\mathcal{S}_l(i, j) = C\mathcal{S}_{l-1}(i, j), \tag{6}$$

for $l = 1, \ldots, L$. From the single agent point of view, the consensus communication scheme corresponds to

$$S_l^{(m)}(i, j) = \sum_{m' \in \mathcal{J}_m} C(m, m') S_{l-1}^{(m')}(i, j). \tag{7}$$

Namely, all agents exchange information with their neighbors on the entries of their similarity matrices for a total of L iterations. Obviously, the following asymptotic relation must be satisfied:

$$\lim_{L\to\infty} S_L^{(m)}(i,j) = \frac{1}{M}\sum_{m'=1}^{M} S^{(m')}(i,j), \tag{8}$$

for all $m = 1,\ldots,M$. Furthermore, the algorithm needs to yield as good results as possible for a finite, practically acceptable, number of consensus steps L. Coming back to (6), by reiterating for $l = 1,\ldots,L$, it can be written

$$\mathcal{S}_L(i,j) = C^L \mathcal{S}_0(i,j). \tag{9}$$

Here, it is easy to see that in order to achieve (8), consensus matrices must satisfy

$$\lim_{L\to\infty} C^L = \frac{\mathbf{1}\mathbf{1}^T}{M}, \tag{10}$$

where $\mathbf{1}$ denotes a column vector with all entries equal to 1. This equation holds if and only if [28]:

$$\mathbf{1}^T C = \mathbf{1}^T, \quad C\mathbf{1} = \mathbf{1}, \quad \text{and} \quad \rho(C - \mathbf{1}\mathbf{1}^T/M) < 1, \tag{11}$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix.

If one is to obtain the fastest convergence to the desired asymptotic values in (10), the problem of finding such a consensus matrix C that minimizes $\rho(C - \mathbf{1}\mathbf{1}^T/M)$ must be solved [28]. This represents an optimization problem with many degrees of freedom; the simplest reduction in the degrees of freedom is to set all non-zero non-diagonal entries of the consensus matrix equal to a constant scalar α. In this case, it can be shown that the optimal value of α, which results in the fastest consensus scheme, is given by [28]:

$$\alpha^* = \frac{2}{\sigma_{\tilde{L}}(1) + \sigma_{\tilde{L}}(M-1)}, \tag{12}$$

where $\tilde{L}$ denotes the Laplacian matrix of the graph $\mathcal{G}$, and $\sigma_{\tilde{L}}(i)$ its i-th largest eigenvalue. The Laplacian matrix $\tilde{L}$ is in the same form as the adjacency matrix, but with non-zero elements equal to -1, and with diagonal elements set in a way that ensures each row of $\tilde{L}$ sums to 0. An illustrative example of designing the fastest consensus scheme will be given in the simulation section. Randomizing the consensus communication scheme and distributing the optimization process itself represent possible extensions of the proposed algorithm, in line with the results from [29].

3.2.2. Algorithm 2

The task of obtaining the distributed solution for (4) is slightly more involved. It is assumed that the agents exchange information based on the entries of similarity and indicator matrices in multiple iterations. The entry in the agent m's indicator matrix obtained after l steps of consensus is denoted as $I_l^{(m)}(i,j)$, starting from $I_0^{(m)}(i,j) = I^{(m)}(i,j)$. For compact representation, the entries of different agents are concatenated as $\mathcal{I}_l(i,j) = (I_l^{(1)}(i,j),\ldots,I_l^{(M)}(i,j))^T$. These variables are needed in relation to the denominator of (4). For the nominator, a novel variable $Z_l^{(m)}(i,j)$ is introduced, which starts from $Z_0^{(m)}(i,j) = I^{(m)}(i,j)S^{(m)}(i,j)$, and its entries are concatenated as $\mathcal{Z}_l(i,j) = (Z_l^{(1)}(i,j),\ldots,Z_l^{(M)}(i,j))^T$. Here, it is possible to write

$$\begin{aligned} \mathcal{I}_l(i,j) &= C\mathcal{I}_{l-1}(i,j), \\ \mathcal{Z}_l(i,j) &= C\mathcal{Z}_{l-1}(i,j), \end{aligned} \tag{13}$$

for $l = 1, \ldots, L$, which can be reiterated as

$$\begin{aligned} \mathcal{I}_L(i,j) &= C^L \mathcal{I}_0(i,j), \\ \mathcal{Z}_L(i,j) &= C^L \mathcal{Z}_0(i,j). \end{aligned} \tag{14}$$

Consequently, one can obtain

$$\mathcal{S}_L(i,j) = \frac{\mathcal{Z}_L(i,j)}{\mathcal{I}_L(i,j)}, \tag{15}$$

where the division operates in an element-wise manner. In this algorithm, from the single agent point of view, the consensus communication scheme corresponds to

$$S_l^{(m)}(i,j) = \frac{\sum_{m' \in \mathcal{J}_m} C(m,m') Z_{l-1}^{(m')}(i,j)}{\sum_{m' \in \mathcal{J}_m} C(m,m') I_{l-1}^{(m')}(i,j)}. \tag{16}$$

It has greater communication requirements than (7), as it assumes that all agents exchange information with their neighbors on the entries of both the indicator matrices and the matrices initially obtained by element-wise multiplication of the indicator and similarity matrices. The algorithm structurally resembles the Two Parallel Passes of the Agreement Algorithm from [16]. It is based on adaptive consensus-based approaches for distributed estimation [30,31]. For all elements $S_L^{(m)}(i,j)$ of $\mathcal{S}_L(i,j)$ to satisfy

$$\lim_{L \to \infty} S_L^{(m)}(i,j) = \frac{\sum_{m'=1}^{M} I^{(m')}(i,j) S^{(m')}(i,j)}{\sum_{m'=1}^{M} I^{(m')}(i,j)}, \tag{17}$$

similar to before, conditions (10) and (11) must be met. For the fastest communication scheme with all equal communication weights, the weights are to be set based on (12).

4. Experiments

In this section, three sets of numerical experiments are conducted. The first is focused on illustrating the design principles underlying the proposed communication scheme, the second is focused on demonstrating the performance of the proposed distributed solution in both object-distributed and feature-distributed settings, and the third is focused on a specific application scenario within the multimedia domain.

4.1. Communication Scheme Design

As an appropriate starting point for the analysis of the proposed ensemble clustering solution based on consensus, some classical results regarding the consensus communication schemes [28] are revisited. An agent network with $M = 10$ nodes is assumed, where the nodes are randomly spatially distributed and connected according to a distance criterion (corresponding to the so-called random geometric graph topology). Connected nodes are able to communicate with each other bidirectionally. In this example, the nodes are distributed within a square area and connected if their distance is less than half of the side of the square, making sure that the corresponding graph is connected. One example of such a network is illustrated in Figure 5.

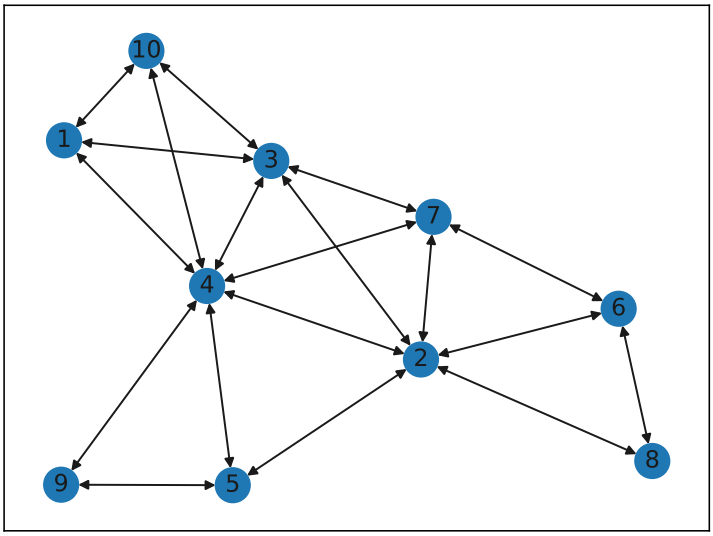

Figure 5. An example of the used network graph. The nodes are labeled with integers, so that the correspondence to the Laplacian and consensus matrices can be established (labels correspond to column and row indices of the respective matrices).

The corresponding Laplacian matrix is:

$$\tilde{L} = \begin{bmatrix} 3 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 6 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 \\ -1 & -1 & 5 & -1 & 0 & 0 & -1 & 0 & 0 & -1 \\ -1 & -1 & -1 & 7 & -1 & 0 & -1 & 0 & -1 & -1 \\ 0 & -1 & 0 & -1 & 3 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 3 & -1 & -1 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 & -1 & 4 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 2 & 0 \\ -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}. \tag{18}$$

The set of all eigenvalues of $\tilde{L}$, i.e., the spectrum of $\tilde{L}$, can be calculated as $\sigma_{\tilde{L}} = \{8.1, 7.09, 5.92, 4.21, 4, 3.44, 2.77, 1.38, 1.04, 0\}$ (in descending order). Here, the optimal value of the parameter α can be obtained, according to (12), as $\alpha^* = 2/(8.1 + 1.04) = 0.22$. Subsequently, the entries of the optimal consensus communication matrix, ensuring the fastest agreement between the nodes, can be calculated. Their values, rounded to two decimals, are:

$$C = \begin{bmatrix} 0.35 & 0 & 0.22 & 0.22 & 0 & 0 & 0 & 0 & 0 & 0.22 \\ 0 & -0.31 & 0.22 & 0.22 & 0.22 & 0.22 & 0.22 & 0.22 & 0 & 0 \\ 0.22 & 0.22 & -0.09 & 0.22 & 0 & 0 & 0.22 & 0 & 0 & 0.22 \\ 0.22 & 0.22 & 0.22 & -0.52 & 0.22 & 0 & 0.22 & 0 & 0.22 & 0.22 \\ 0 & 0.22 & 0 & 0.22 & 0.35 & 0 & 0 & 0 & 0.22 & 0 \\ 0 & 0.22 & 0 & 0 & 0 & 0.35 & 0.22 & 0.22 & 0 & 0 \\ 0 & 0.22 & 0.22 & 0.22 & 0 & 0.22 & 0.13 & 0 & 0 & 0 \\ 0 & 0.22 & 0 & 0 & 0 & 0.22 & 0 & 0.56 & 0 & 0 \\ 0 & 0 & 0 & 0.22 & 0.22 & 0 & 0 & 0 & 0.56 & 0 \\ 0.22 & 0 & 0.22 & 0.22 & 0 & 0 & 0 & 0 & 0 & 0.35 \end{bmatrix}. \tag{19}$$

For confirmation, the spectral radius of $C - \mathbf{1}\mathbf{1}^T/M$, connected to different consensus matrices C obtained for different values of α, is calculated and illustrated in Figure 6. It can be seen that the lowest value, ensuring the fastest convergence of the communication scheme, indeed results from using α^*.

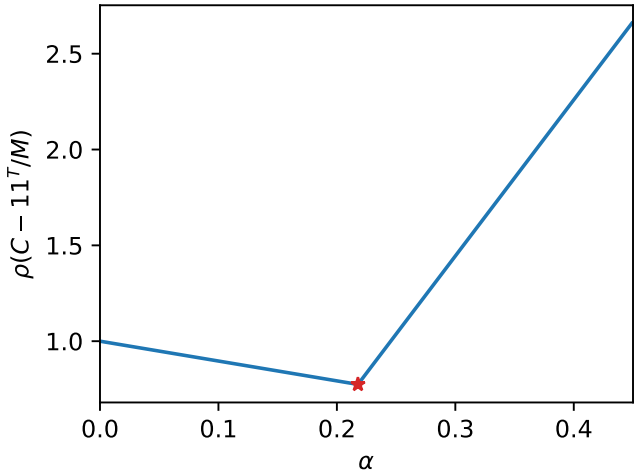

Figure 6. Spectral radius of the consensus communication matrices for different values of the parameter α. The optimal value is illustrated with a star, corresponding to the α^* obtained from (12).

Now that it is demonstrated how the fastest consensus communication scheme can be designed and obtained, the effect of the network topology and the number of consensus steps on the performance of the distributed algorithm is further explored. Several network topologies that arise from different real-world scenarios and practical applications are considered. Specifically, the performance of ring, star, and mesh topologies, which serve as prototypical examples of decentralized federated learning schemes, is investigated. Furthermore, the examples of random geometric and Erdős–Rényi networks, which represent some of the most popular network topologies in the domain of networked multi-agent systems, are studied. For the described five networks, as shown in the left plots in Figure 7, the norm of $C - \mathbf{1}\mathbf{1}^T/M$, as a measure of the disagreement between the nodes, is calculated with respect to the different numbers of consensus steps. The resulting curves are shown in the right plots in Figure 7. It can be seen, as expected, that the agreement is achieved more quickly as the number of communication links (loosely speaking) and the number of consensus steps increase. This analysis also provides an indicator for the number of communication interactions between the nodes needed to obtain the wanted level of performance for a given practical scenario.

4.2. Distributed Ensemble Clustering

The above numerical analysis shows that the proposed distributed algorithms will, for a sufficient number of consensus steps, achieve performance very close to the performance of the corresponding centralized solutions. This conclusion holds regardless of the data splitting scenario (object and feature-wise), and regardless of the specific clusterers and partitioners used. However, to obtain a more complete picture, the proposed distributed solutions are tested on the Optical Recognition of Handwritten Digits dataset [32]. This dataset allows for a clear and simple illustration of the expected performance of the proposed algorithms in various scenarios. The numerical experiments are conducted using *Python* programming language, and its well-known *scikit-learn* machine learning library [33]. The main task, in addition to the design of the consensus communication scheme, represents the design of the experimental setup itself, which should provide an assessment of the performance of multiple algorithms in both object and feature-distributed scenarios.

The considered dataset consists of 1797 objects with 64 features, divided across known 10 classes. The network from Figure 5 and the corresponding consensus matrix from (19) are used. Each node is associated with its own k-means clusterer with 10 target clusters, and its own dataset, obtained by random sampling of the original dataset. Both object and feature-distributed clusterings are taken into account by using a range of sizes obtained as different fractions of the number of objects and the number of features of the original

dataset. For the local partitioners, *k*-means clustering on the similarity matrices used as data matrices is applied. For better controllability, parameters of all *k*-means clusterers and partitioners are uniformly configured.

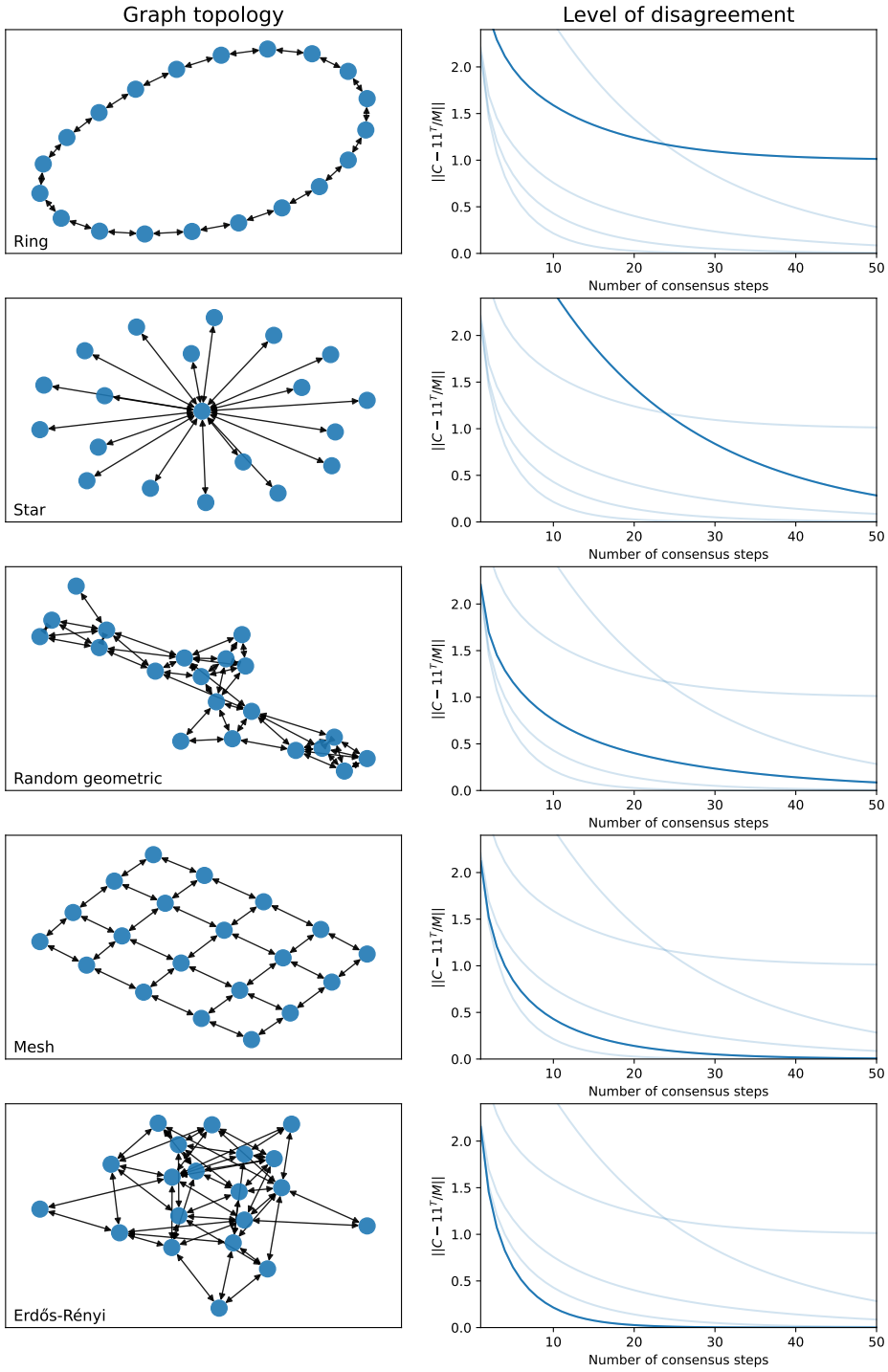

Figure 7. (**Left**): five used network communication topologies. (**Right**): corresponding levels of disagreement between the network nodes for a different number of consensus steps. For better comparison, each plot contains curves for all five networks; the curve corresponding to the network on the left side of it is shown in deeper blue.

Both centralized and distributed ensemble clustering algorithms are simulated for both Algorithms 1 and 2. In addition, a representative of ensemble clustering schemes relying

on the exchange of information on the prototypes of local clusters (i.e., cluster centroids) is considered. Following [15], it assumes an all-to-all communication scheme and introduces a suitable centroid similarity function aimed at solving the label correspondence problem. Herein, a softmax function on the negative distances between the cluster centroids has been used within the soft centroid-to-centroid assignment scheme. It should be emphasized that the corresponding solution, named Algorithm 3, utilizes local clusterers, which all have to deal with the same set of features. This is due to the fact that the communication is based on cluster centroids, which, therefore, must be embedded within the same feature space. Notwithstanding its inherent limitations, analyzing Algorithm 3 provides an indicator for the performance of a whole class of ensemble clustering solutions whose focus is on the prototypes of local clusters.

Figure 8 illustrates the corresponding performances in terms of the Normalized Mutual Information (NMI) score of the obtained labelings with respect to the available true labelings. NMI has been one of the standard metrics used in this context [18,19,21]. For distributed algorithms, average values across all nodes are shown. As expected, it can be seen that the performances are better for higher number of objects and higher number of features each clusterer has access to. Both Algorithms 1 and 2 behave similarly when the relative numbers of objects that are available from the original dataset are high; for lower relative numbers, Algorithm 2 is obviously a better option. It can be seen that the proposed distributed solutions achieve performance very close to that of their centralized counterparts. Algorithm 3 underperforms the proposed Algorithms 1 and 2 in all cases except in the case of a low number of objects available to local clusterers. This indicates the potential of Algorithm 3 to serve as a basis for analogous consensus-based object-distributed approaches with a low number of objects if its inherent limitations do not represent an obstacle.

4.3. Multi-Modal Example

To examine the potential of using the proposed algorithm in the domain of multimedia tools and applications, it is tested on the well-known multi-modal Corel 5k dataset [34], which has been extensively used in this field, e.g., [21,35]. The dataset consists of a collection of 5000 images from 50 classes, with 100 images each. There are two modalities of features for each image. The first modality is visual, where segments of the images are preprocessed and clustered according to their features so that each image is associated with up to 10 blobs (centroids of clustered segments). The total number of blobs is 500, which can be thought of as a set of 500 binary features for each image (with a maximum of 10 of them equal to one). The second modality is the image caption, represented by a set of up to five words. The total number of words is 375, which can be thought of as a set of additional 375 binary features for each image (with a maximum of 5 of them equal to one). This example deals with demonstrating a particular practical use of the proposed algorithm, as feature distributions arise naturally from the multi-modal character of the dataset, and may be dispersed across multiple distinct entities. Furthermore, individual clusterers can adopt different feature space designs [21], e.g., they can perform dimensionality reduction algorithms, which is also the subject of analysis.

Half of the agents are associated with visual features and half with text features. Principal Component Analysis (PCA) on the resulting feature sets is performed and different numbers of principal components are kept for each agent. Since all objects in the individual datasets are considered, Algorithms 1 and 2 will behave the same. Figure 9 plots the performance metrics of both centralized and the proposed distributed ensemble clustering solutions, based on the PCA features of unimodal datasets, together with the centralized algorithm based on all features from both data modalities. It can be seen, as before, that the proposed distributed algorithm achieves performance close to that of its centralized unimodal counterpart. As the number of the used PCA features increases, this performance aligns relatively well with the performance of the centralized algorithm having access

to both data modalities (with a total of 875 features), confirming the effectiveness of the distributed unimodal ensemble clustering approach.

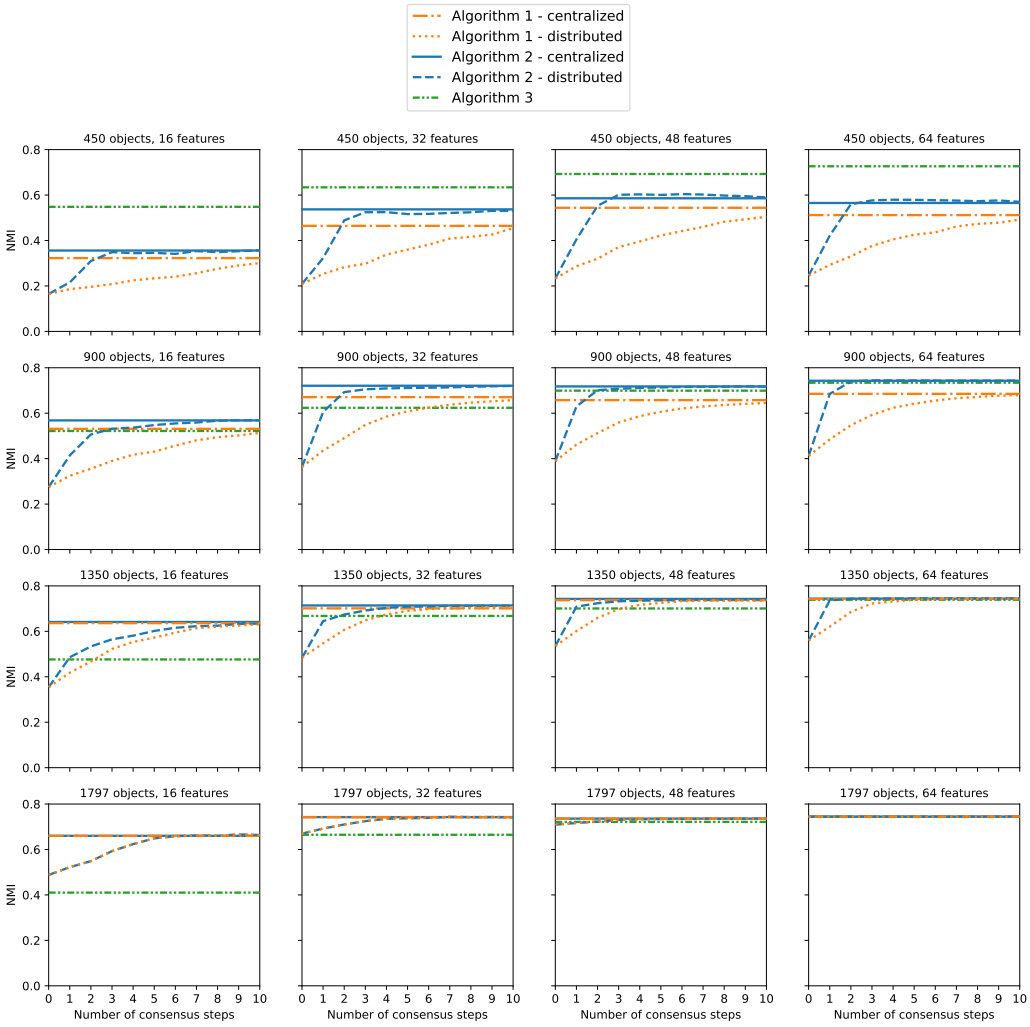

Figure 8. The obtained Normalized Mutual Information (NMI) scores versus the number of consensus steps for different ensemble clustering algorithms—object and feature-distributed examples. Each plot assumes that the individual clusterers have access to different numbers of objects and their features, which is indicated by the plot title. Algorithms 1 and 2 utilize different feature sets, and Algorithm 3 the same feature set, for different clusterers.

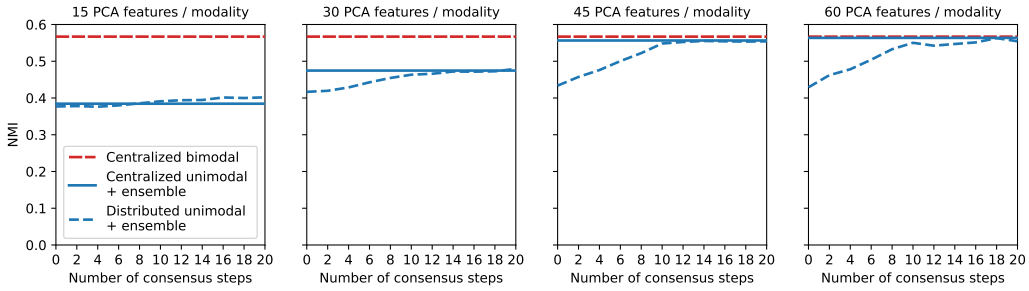

Figure 9. The obtained Normalized Mutual Information (NMI) scores versus the number of consensus steps for different ensemble clustering algorithms—multimodal example. The individual unimodal clusterers have access to different numbers of principal components of the unimodal features; these are indicated by the titles of the respective plots.

5. Conclusions

In this study, a novel distributed ensemble clustering scheme for networked multi-agent systems is proposed. It supports various choices and designs for the local datasets, clusterers, and partitioners. The individual datasets can be both object-distributed and feature-distributed, and clusterers can be of any kind. The proposed communication scheme, inherently decentralized, is based on the exchange of information on the local similarity matrices between the neighboring agents. It results in solutions whose performance closely aligns with that of the analogous centralized solutions, in a provably fastest way. The proposed distributed scheme can serve as a basis or framework for obtaining decentralized algorithms in diverse application scenarios, encompassing various combinations of local datasets, clusterers, and partitioners.

The presented results open up several directions for further research. In the case of large (object-wise) datasets, the available scaling techniques [8,17,18] should be tested and compared in order to come to an even more practically efficient solution. Furthermore, the potential of the proposed distributed scheme (namely, Algorithm 2) to be used in conjunction with the algorithms that refine the individual clusterers' base results, such as [18,19,22–24], is worth investigating. Additionally, exploring the possible relationships of the proposed distributed solutions with the spectral ensemble clustering approach [27], or with the approaches reducing the granularity of the exchanged information between the nodes [14,15], represents very interesting field for future endeavors.

Author Contributions: Conceptualization, N.I. and M.P.; methodology, N.I. and M.P.; software, N.I.; validation, N.I. and M.P.; formal analysis, N.I.; investigation, M.P.; resources, M.P.; data curation, M.P.; writing—original draft preparation, N.I.; writing—review and editing, M.P.; visualization, N.I.; supervision, M.P.; project administration, M.P.; funding acquisition, M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC were funded by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia, grant number 451-03-47/2023-01/200103.

Data Availability Statement: The publicly archived datasets analyzed in the paper can be found at https://archive.ics.uci.edu/dataset/80/optical+recognition+of+handwritten+digits (accessed on 16 September 2023) and http://kobus.ca/research/data/eccv_2002/ (accessed on 16 September 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Murphy, K. *Machine Learning: A Probabilistic Perspective*; MIT Press: Cambridge, MA, USA, 2021.
2. Kuncheva, L.; Hadjitodorov, S.; Todorova, L. Experimental Comparison of Cluster Ensemble Methods. In Proceedings of the 2006 9th International Conference on Information Fusion, Florence, Italy, 10–13 July 2006; pp. 1–7. [CrossRef]
3. Monti, S.; Tamayo, P.; Mesirov, J.; Golub, T. Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. *Mach. Learn.* **2003**, *52*, 91–118. [CrossRef]
4. Strehl, A.; Ghosh, J. Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583–617. [CrossRef]
5. Kleinberg, J.M. An Impossibility Theorem for Clustering. *Adv. Neural Inf. Process. Syst. (NIPS)* **2002**, *15*, 463–470.
6. Ren, W.; Beard, R.W.; Atkins, E.M. A survey of consensus problems in multi-agent coordination. In Proceedings of the American Control Conference, Portland, OR, USA, 8–10 June 2005; pp. 1859–1864. [CrossRef]
7. Olfati-Saber, R.; Fax, A.; Murray, R. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **2007**, *95*, 215–233. [CrossRef]
8. Gionis, A.; Mannila, H.; Tsaparas, P. Clustering aggregation. In Proceedings of the 21st International Conference on Data Engineering (ICDE'05), Tokyo, Japan, 5–8 April 2005; pp. 341–352. [CrossRef]
9. Liu, S.; Liu, Z.; Xu, Z.; Liu, W.; Tian, J. Hierarchical Decentralized Federated Learning Framework with Adaptive Clustering: Bloom-Filter-Based Companions Choice for Learning Non-IID Data in IoV. *Electronics* **2023**, *12*, 3811. [CrossRef]
10. Rosa, A.; Di Lorenzo, P.; Panella, M. Distributed Data Clustering over Networks. *Pattern Recognit.* **2019**, *93*, 603–620. [CrossRef]
11. Gu, D. Distributed EM Algorithm for Gaussian Mixtures in Sensor Networks. *IEEE Trans. Neural Netw.* **2008**, *19*, 1154–1166. [CrossRef]
12. Katselis, D.; Beck, C.L.; van der Schaar, M. Ensemble Online Clustering through Decentralized Observations. In Proceedings of the 53rd IEEE Conference on Decision and Control, Los Angeles, CA, USA, 15–17 December 2014; pp. 910–915. [CrossRef]

13. Ding, H.; Su, L.; Xu, J. Towards Distributed Ensemble Clustering for Networked Sensing Systems: A Novel Geometric Approach. In Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing, New York, NY, USA, 10–14 July 2016; MobiHoc '16; pp. 1–10. [CrossRef]
14. Hore, P.; Hall, L.; Goldgof, D. A Scalable Framework For Cluster Ensembles. *Pattern Recognit.* **2009**, *42*, 676–688. [CrossRef]
15. Rosato, A.; Rosa, A.; Panella, M. A Decentralized Algorithm for Distributed Ensemble Clustering. *Inf. Sci.* **2021**, *578*, 669–677. [CrossRef]
16. Olshevsky, A.; Tsitsiklis, J.N. Convergence Speed in Distributed Consensus and Averaging. *SIAM Rev.* **2011**, *53*, 747–772. [CrossRef]
17. Fred, A.; Jain, A. Combining Multiple Clusterings Using Evidence Accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 835–850. [CrossRef] [PubMed]
18. Zhou, J.; Zheng, H.; Pan, L. Ensemble Clustering based on Dense Representation. *Neurocomputing* **2019**, *357*, 66–76. [CrossRef]
19. Huang, D.; Wang, C.D.; Lai, J.H. Locally Weighted Ensemble Clustering. *IEEE Trans. Cybern.* **2018**, *48*, 1460–1473. [CrossRef] [PubMed]
20. Chu, X.; Tan, X.; Zeng, W. A Clustering Ensemble Method of Aircraft Trajectory Based on the Similarity Matrix. *Aerospace* **2022**, *9*, 269. [CrossRef]
21. Sevillano, X.; Carrié, J.C.; Alías-Pujol, F. Parallel Hierarchical Architectures for Efficient Consensus Clustering on Big Multimedia Cluster Ensembles. *Inf. Sci.* **2019**, *511*, 212–228. [CrossRef]
22. Wang, X.; Yang, C.; Zhou, J. Clustering aggregation by probability accumulation. *Pattern Recognit.* **2009**, *42*, 668–675. [CrossRef]
23. Li, T.; Ding, C. Weighted Consensus Clustering. In Proceedings of the SIAM International Conference on Data Mining, SDM, Atlanta, GA, USA, 24–26 April 2008; Volume 2, pp. 798–809. [CrossRef]
24. Zhou, P.; Du, L.; Li, X. Adaptive Consensus Clustering for Multiple K-means via Base Results Refining. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 10251–10264. [CrossRef]
25. Karypis, G.; Kumar, V. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *Siam J. Sci. Comput.* **1999**, *20*, 359–392. [CrossRef]
26. Pekalska, E.; Duin, R. *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*; Vol. Series in Machine Perception and Artificial Intelligence; World Scientific Publishing: Singapore, 2005. [CrossRef]
27. Liu, H.; Liu, T.; Wu, J.; Tao, D.; Fu, Y. Spectral Ensemble Clustering. In Proceedings of the KDD'15: The 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 715–724. [CrossRef]
28. Xiao, L.; Boyd, S. Fast linear iterations for distributed averaging. *Syst. Control Lett.* **2004**, *53*, 65–78. [CrossRef]
29. Boyd, S.; Ghosh, A.; Prabhakar, B.; Shah, D. Randomized Gossip Algorithms. *IEEE Trans. Inf. Theory* **2006**, *52*, 2508–2530. [CrossRef]
30. Ilić, N.; Stanković, M.S.; Stanković, S.S. Adaptive Consensus-Based Distributed Target Tracking in Sensor Networks with Limited Sensing Range. *IEEE Trans. Control Syst. Technol.* **2014**, *22*, 778–785. [CrossRef]
31. Stanković, S.S.; Ilić, N.; Stanković, M.S. Adaptive Consensus-Based Distributed System for Multisensor Multitarget Tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 2164–2179. [CrossRef]
32. Alpaydin, E.; Kaynak, C. Optical Recognition of Handwritten Digits. UCI Machine Learning Repository. 1998. Available online: https://archive.ics.uci.edu/dataset/80/optical+recognition+of+handwritten+digits (accessed on 16 September 2023).
33. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
34. Duygulu, P.; Barnard, K.; Freitas, J.; Forsyth, D. *Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary*; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2353, pp. 349–354. [CrossRef]
35. Bekkerman, R.; Jeon, J. Multi-modal Clustering for Multimedia Collections. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.