Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 753102, 13 pages
http://dx.doi.org/10.1155/2015/753102

Research Article

Distributed Data Clustering via Opinion Dynamics

Gabriele Oliva,[1] Damiano La Manna,[2] Adriano Fagiolini,[2] and Roberto Setola[1]

[1] *University Campus Bio-Medico of Rome, Via A. del Portillo 21, 00128 Rome, Italy*
[2] *Dipartimento di Energia, Ingegneria dell'Informazione e Modelli Matematici (DEIM), University of Palermo, Viale delle Scienze, Edificio 10, 90128 Palermo, Italy*

Correspondence should be addressed to Gabriele Oliva; g.oliva@unicampus.it

Received 27 November 2014; Accepted 5 February 2015

Academic Editor: Jianshe Wu

We provide a distributed method to partition a large set of data in clusters, characterized by small in-group and large out-group distances. We assume a wireless sensors network in which each sensor is given a large set of data and the objective is to provide a way to group the sensors in homogeneous clusters by information type. In previous literature, the desired number of clusters must be specified a priori by the user. In our approach, the clusters are constrained to have centroids with a distance at least ε between them and the number of desired clusters is not specified. Although traditional algorithms fail to solve the problem with this constraint, it can help obtain a better clustering. In this paper, a solution based on the Hegselmann-Krause opinion dynamics model is proposed to find an admissible, although suboptimal, solution. The Hegselmann-Krause model is a centralized algorithm; here we provide a distributed implementation, based on a combination of distributed consensus algorithms. A comparison with k-means algorithm concludes the paper.

1. Introduction

The problem of grouping large amounts of data into a small number of subsets with some common features among the elements (often referred to as the *data clustering problem*) has attracted the work of several researchers in different fields, ranging from statistics to imagine analysis and bioinformatics [1–3].

Data clustering techniques are developed to partition an initial set of observation data into collections with small in-group distances and big out-group distances.

Among the existing techniques, one of the most used is the k-means algorithm or its successive extensions (e.g., fuzzy c-means [4], mixture of Gaussians algorithms [5]). Given a set of initial observation data and a number k of desired clusters, the k-means algorithm computes a suboptimal placement of k cluster centroids and assigns the observations to such centroids, alternating between an assignment phase, where each observation point is assigned to its nearest centroid, and refinement phase, where each centroid position is updated as the center of mass of all observations belonging to that centroid.

A well-known limitation of data clustering algorithms, such as the k-means algorithm, is that the number of clusters has to be specified beforehand, based for example, on subjective evaluations or a priori analysis. Since this assumption is typically not feasible in practice, a typical solution consists of running several times the algorithm with a different number of clusters and then deciding the best obtained solution based on a posteriori evaluation [6]. Another issue of traditional algorithms is that there is no guarantee that the clusters are sufficiently far from each other. To this respect, distance-constrained data clustering approaches have been devised in the literature: in [7, 8] the considered constraints are the so-called *must-links* (i.e., an observation i must belong to a cluster j) and *cannot-link* (i.e., an observation i can not belong to a cluster j); in [9] the feasibility of a constrained problem involving the so-called δ-*constraints* (i.e., any two observations must have a distance greater than δ) and the ε-*constraints* (i.e., for any observation i in cluster j there must be at least another observation h in cluster j such that the distance between i and j is less than ε) is given. To the best of our knowledge, nowadays, there is no methodology to specify a constraint on the distance between

cluster centroids, while this class of constraints might help finding a choice for the number k when such value is a priori unknown.

This problem has a particular relevance in a distributed setting, where a network of sensors has to classify information provided by several sensors without a central authority but using only local data exchange among neighbors in the network. Differently from community clustering in wireless networks (e.g., see [10]), here the nodes are clustered depending on the data they sense, with no particular dependency on the network topology.

In this paper, based on the preliminary results in [11], a novel approach to solve the data clustering problem with a distance constraint among cluster centroids is provided that does not require the specification of the initial number k of clusters and that is based on an extension of the Hegselmann-Krause (HK) opinion dynamics model [12, 13], which handles scalar data, in order to handle data in $\mathbb{R}^d$. Such a model, similar to consensus [14, 15], represents how a set of agents interacts in order to reach a local agreement, but the agents may split in several clusters depending on their "opinions."

Within the proposed approach, rather than striving for a complete agreement among the agents, we exploit the peculiarity of HK model to generate several clusters, with the aim to map a large set of measurement data into few values (i.e, the opinion clusters). In this view, as it will be discussed in the following, HK models can be seen as a powerful methodology to determine the number of clusters while respecting the constraints on the distance among cluster centroids. The HK model, in its original setting, is a centralized algorithm [13]; hence we devise a distributed implementation based on a combination of consensus algorithms, which are an effective way to distribute complex operations, for example, time synchronization [16].

The outline of the paper is as follows: after some preliminaries, in Section 2, we provide a formulation of the problem at hand and in Section 3 we analyze the formulation of the standard data clustering problem and the k-means algorithm; then in Section 4 the HK opinion dynamics model is reviewed, while in Section 5 the distributed consensus algorithms are examined; Section 6 is devoted to outline the proposed approach to solve the distance-constrained clustering problem, while Section 7 addresses the distributed implementation of the HK opinion dynamics model; Section 8 contains some numeric examples to show the potentialities of this method, while Section 9 contains some conclusive remarks.

Preliminaries. Let $G = \{V, E\}$ be a *graph* with n nodes $v_i \in V$ and e_i edges $(v_i, v_j) \in E$.

A graph is said to be *undirected* if $(v_i, v_j) \in E$ whenever $(v_j, v_i) \in E$ and is said to be *directed* otherwise. Let the neighborhood $\mathcal{N}_i$ of a node i be the set of nodes v_j such that $(v_j, v_i) \in E$, and let the *degree* $d_i = |\mathcal{N}_i|$. A graph G is *connected* if for any $v_i, v_j \in V$ there is a path whose endpoints are in v_i and v_j.

2. Problem Statement

Consider a collection of n sensors, each equipped with a d-dimensional measurement or piece of information $x_1, \ldots, x_n$, with $x_i \in \mathbb{R}^d$. We want to select a number k of groups or *clusters*, $C_1, \ldots, C_k$, $k \leq n$. Every cluster C_j is assigned to a cluster centroid $c_j \in \mathbb{R}^d$, which represents the centroid of the observations allocated to that cluster. A data point $x_i \in \mathbb{R}^d$ is said to belong to cluster C_j if its distance from the centroid c_h, with $h \neq j$, of every other cluster C_h is larger than its distance from c_j. If x_i belongs to C_j we can set a binary assignment variable $r_{ij} \in \mathbb{B}$ to 1 and to 0 otherwise. The solution of a data clustering problem with distance constraints involves the computation of the optimal choice of cluster centroids c_j, for $j = 1, \ldots, k$, that minimizes the distance of every measurement data point x_i from the cluster it belongs to. More formally, we need to solve the following.

Problem 1 (distance-constrained data clustering). We want to find the number of clusters k, the cluster centroids, $c_j \in \mathbb{R}^d$, and measurement data assignments, $r_{ij} \in \mathbb{R}$ that minimize the index

$$D = \sum_{i=1}^{n} \sum_{j=1}^{k} r_{ij} \left\| x_i - c_j \right\|^2, \tag{1}$$

subject to the constraints

$$\begin{aligned} &\text{(I)} \quad \sum_{j=1}^{k} r_{ij} = 1 \quad \forall i = 1, \ldots, n \\ &\text{(II)} \quad \left\| c_h - c_j \right\|^2 \geq \varepsilon \quad \forall i, j = 1, \ldots, k;\ h \neq j \\ &\text{(III)} \quad r_{ij} \in \mathbb{B} \quad \forall i = 1, \ldots, n;\ \forall j = 1, \ldots, k. \end{aligned} \tag{2}$$

The first set of constraints (I) along with the third one implies that each observation is assigned exactly to one cluster; the constraints (II) imply that ε is a lower bound for the distance between any pair of centroids; finally constraints (III) imply that r_{ij} are binary decision variables.

This problem is novel, since in the literature, the set of constraints (II) is typically not considered. Including such constraints, however, is quite useful, since it may improve the algorithm's ability to detect homogeneous clusters. The problem is very hard to solve exactly and traditional methods as the k-means algorithm [17] may fail even at finding an admissible solution, as it will be shown in the next section.

On the other side, the proposed algorithm uses an algorithm based on the Hegselmann-Krause model to choose an admissible, although suboptimal, solution, and it has a modest increase in computational complexity with respect to the k-means algorithm, while allowing several advantages (further discussed later in the paper):

(i) the algorithm provided in this paper always finds an admissible, although suboptimal, solution to the problem by means of the HK model, while the k-means algorithm may fail;

(ii) the proposed approach does not require the user to define a priori the number of clusters, but it finds automatically a suitable number of clusters based on the parameter ε;

(iii) outliers can be automatically isolated, without any a priori data processing; this feature can be obtained by dropping out the clusters whose cardinality is significantly less than the others;

(iv) the solution provided in this paper is deterministic, that is, for fixed ε and fixed observations, the result is always the same, while the k-means algorithm depends on the initial random choice of the centroids;

(v) while traditional approaches are very computationally expensive when applied in a decentralized and distributed setting [18] (i.e., for a sensors network), this method can be distributed with a modest increase in computational complexity [19].

2.1. Distributed Setting. In this paper we want to provide a solution to Problem 1, where a set of n agents solves the above problem in a distributed fashion, that is, by means of local information exchange among the agents instead of resorting to a centralized processing unit. Let $G = \{V, E\}$ be a graph involving all the agents and without any loss of generality; suppose G is connected, undirected, and fixed. Moreover, it is assumed that each agent i in the network

(1) has an associated observation $x_i \in \mathbb{R}^d$;

(2) knows $\varepsilon \in [0, 1]$ which is the same for all agents;

(3) has a unique identifier;

(4) can exchange directly information only with neighbors over G;

(5) can act synchronously.

In the next section we briefly review the traditional data clustering problem and the k-means algorithm.

3. Data Clustering

Let us discuss the following problem.

Problem 2 (standard data clustering problem). Given $k \leq n$, the data clustering problem consists in finding $r_{ij} \in \mathbb{B}$, $c_j \in \mathbb{R}^d$, $i, j = 1, \ldots, n$, that minimize

$$D = \sum_{i=1}^{n} \sum_{j=1}^{k} r_{ij} \left\| x_i - c_j \right\|^2$$

Subject to

$$\text{(I)} \quad \sum_{j=1}^{n} r_{ij} = 1 \quad \forall i = 1, \ldots, n$$

$$\text{(II)} \quad r_{ij} \in \{0, 1\} \quad \forall i = 1, \ldots, n; \ \forall j = 1, \ldots, k. \tag{3}$$

Problem (3) is hard to solve, and in the literature, several iterative algorithms have been devised. Among the others, the *k-means algorithm* [17] proved its effectiveness.

Specifically, starting with a random set of k centroids $c_1(0), \ldots, c_k(0)$, the algorithm alternates for each step an *assignment* and a *refinement* phase.

During the assignment phase, each observation x_i is assigned to the set characterized by the nearest centroid, that is,

$$r_{ih}(t) = \begin{cases} 1, & \text{if } h = \underset{j}{\operatorname{argmin}} \left\| x_i - c_j(t) \right\|, \\ 0, & \text{else.} \end{cases} \tag{4}$$

Within the refinement phase each centroid c_j is updated as the centroid of the observations associated with the cluster $C_j(t)$, that is,

$$c_j(t+1) = \frac{\sum_{i=1}^{n} r_{ij}(t)\, x_i}{\sum_{i=1}^{n} r_{ij}(t)}. \tag{5}$$

The two steps are iterated until convergence or up to a maximum of M iterations.

Figure 1 reports a simulation run of the algorithm for a set of $n = 12$ observations in $\mathbb{R}^2$ and for $k = 3$. Specifically Figure 1(a) shows with circles the initial centroids, Figures 1(b) and 1(c) report the assignment and refinement phases for the first step, while Figure 1(d) depicts the assignment phase for the second step.

The k-means algorithm is granted to converge to a local optimum value, while there is no guarantee to converge to the global optimum [17, 20].

Since there is a strong dependency on the initial choice of the centroids, a common practice is to execute the algorithm several times and select the best solution. The algorithm, moreover, is extremely sensitive to outliers, which can significantly alter the results; to cope with this issue, the outliers have to be identified and excluded prior to the execution of the algorithm.

Note that, for each step, each of the n observations and for each of the d components of the observations, the algorithm calculates the difference with each of the k centers; hence the computational complexity is $O(d\ k\ n\ M)$, where M are the total number of iterations [20]. Notice that in [18] a distributed implementation of the k-means algorithm has been provided, with a computational complexity of $O(d\ k\ n^2\ M)$ for each agent.

Note further that, unfortunately, the k-means algorithm is not able to solve Problem 1; hence we need to seek for other solutions.

In the following we discuss an alternative method based on the HK opinion dynamics model, which is granted to provide an admissible solution to Problem 1. One of the peculiar characteristics of this algorithm is that the number of clusters k has not to be imposed a priori but becomes an output of the algorithm. The HK opinion dynamics model is reviewed in the next section, while the proposed approach is outlined in Section 6.

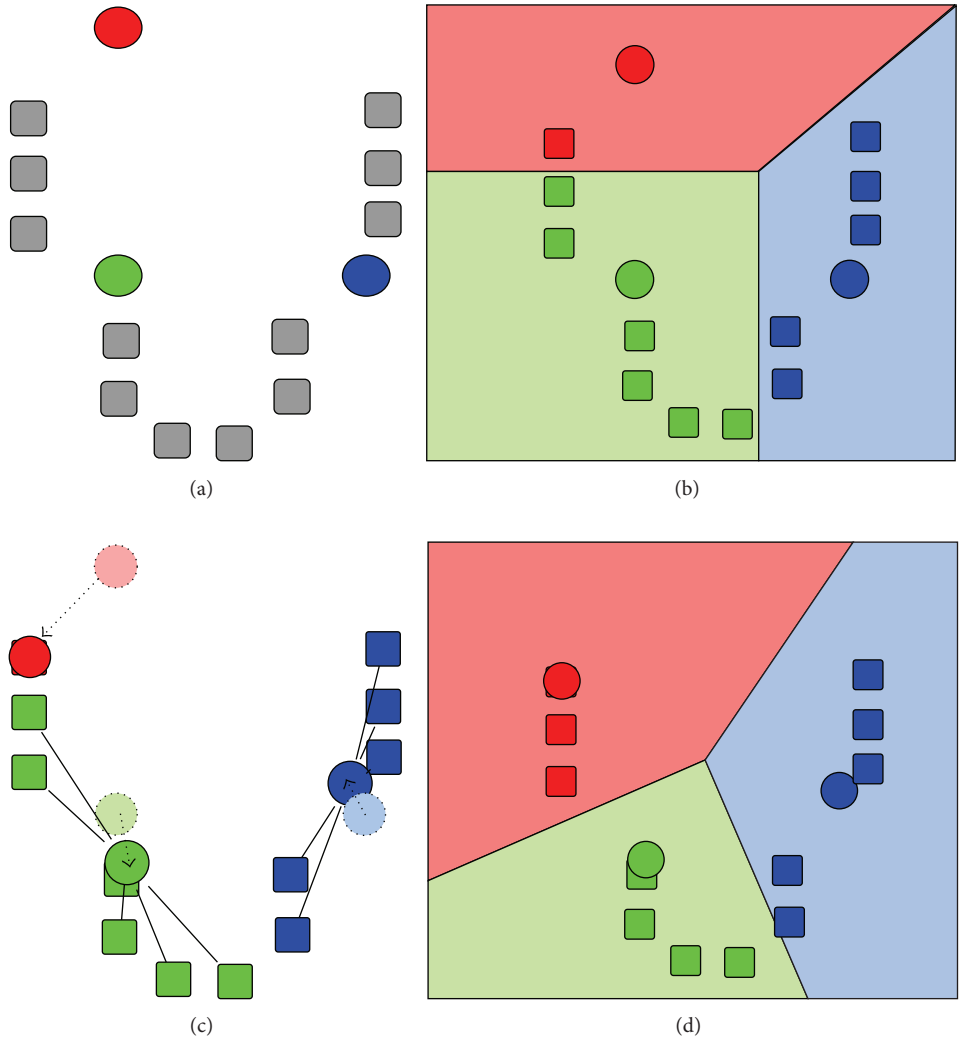

Figure 1: Example of execution of k-means algorithm (source: Wikimedia Commons available under GNU Free Documentation License v. 1.2).

4. Hegselmann-Krause Opinion Dynamics Model

In this section we recall the so-called HK opinion dynamics model and revise some of the main theoretical results. Consider a group of n agents, each characterized by a discrete time dynamic equation. The opinions, represented by the state associated with each agent.

Specifically, each agent is provided with a scalar initial opinion $x_i(0) \in \mathbb{R}$, and the opinion of each agent varies depending on the opinion of the others.

The key idea of the HK model is that agents with completely different opinions do not influence each other, while some sort of mediation occurs among agents whose opinions are close enough.

Let $x_i(t) \in \mathbb{R}$ be the opinion of ith agent at time step t and let $x(t) = [x_1(t), \ldots, x_n(t)]^T$ be the vector of the opinions of all the agents. The ith agent is influenced by opinions that differ from his own less than a given confidence level $\varepsilon \geq 0$. Hence the *neighborhood* of an agent for each time step t can be defined as

$$\mathcal{N}_i(x_i(t)) = \left\{ j \in \{1, \ldots, n\} : \left| x_i(t) - x_j(t) \right| \leq \varepsilon \right\}. \quad (6)$$

Note that, at each step t, $\mathcal{N}_i(x_i(t))$ contains the ith agent itself. This models the fact that each agent takes into account also its current opinion to form a new one.

The HK dynamic model is in the form

$$x(t+1) = A(x(t), \varepsilon)\, x(t), \quad x(0) = x_0, \quad (7)$$

where $A(x(t), \varepsilon)$ is the time-varying (actually state-dependent) $n \times n$ adjacency matrix whose entries are in the form

$$\{A(x(t), \varepsilon)\}_{ij} = \begin{cases} \dfrac{1}{\operatorname{card}\left(\mathcal{N}_i(x_i(t))\right)}, & \text{if } j \in N_i(x_i(t)), \\ 0, & \text{otherwise}, \end{cases} \quad (8)$$

where card($\mathcal{N}_i(x_i(t))$) is the cardinality of $\mathcal{N}_i(x_i(t))$. Note that A is a stochastic matrix, and all its elements belong to the interval $[0, 1/n]$.

Several works can be found in the literature which attempt to characterize the properties of the HK model. Given the complexity of the above model, most of the studies consider simple *initial opinion profiles* (i.e., the initial condition $x(0)$). Two different classes are considered in the literature [13, 21]:

(i) the *equidistant profile*, where $x_i(0) = (i-1)/(n-1)$ hence $x_i(0) \in [0, 1]$;

(ii) the *random profile*, where the opinions are uniformly distributed within $[0, 1]$.

In [13] it is conjectured that for every confidence level ε there must be a number of agents n subject to the equidistant profile for which a consensus is obtained (i.e., a single shared opinion for all the agents), while in [21] it is conjectured that, for any initial opinion profile, there exists a finite time after which the topology underlying the matrix $A(x(t), \varepsilon)$ (i.e., the structure of the mutual influence among agents) remains fixed. Recently this claim proved true, as in [22] it is proved that the convergence time is $O(n^2)$. In [23] it is proven that, during the evolution of the system, the order of the opinions is preserved, that is,

$$x_i(0) \leq x_j(0) \Longrightarrow x_i(t) \leq x_j(t), \quad \forall t. \tag{9}$$

Moreover it is proved that, if the initial opinion profile is sorted, the evolutions of the smallest opinion $x_1(t)$ and of the largest opinion $x_n(t)$ are nondecreasing and nonincreasing, respectively. Clearly, at any step t, if $|x_i(t) - x_{i+1}(t)| > \varepsilon$, this remains true for any subsequent step, and the system splits into two independent subsystems. In [23–25] the stability of the dynamical model is investigated. In particular, the fact that the system converges to a steady opinion profile in finite time is proved in [23]. However, the fact that the system might converge to a common opinion or split into clusters is still under investigation. Besides, evaluating a lower bound on the intercluster distance, some interesting conditions for the study of the equilibrium stability are established. Figure 2 shows an example of result of the HK model with $n = 100$ agents, equidistantly distributed within $[0, 1]$, for different values of the parameter ε. It is noteworthy that the number of clusters decreases when ε grows.

Let us discuss the following property that will be used to solve the clustering problem with distance constraints.

Property 1. ε is a lower bound for the intercluster distance; in fact, when a steady state is reached, any two clusters have a distance which is necessarily greater than or equal to ε, otherwise they would have merged during the evolution of the model.

The above property justifies the adoption of the HK model to solve the data clustering problem with distance constraints.

Regarding the computational complexity, at each step of the algorithm the distance among the opinion of all agents is calculated. Thus the computational complexity is $O(n^2 M)$, where M is the maximum number of iterations.

5. Consensus Algorithms

In order to provide a distributed implementation of the HK model, let us discuss distributed consensus algorithms with reference to connected, fixed, and undirected graphs.

Suppose each node in a graph G represents an agent with an initial condition $x_i(0) \in \mathbb{R}$; at each iteration t each node i updates its state as

$$x_i(t+1) = \mathcal{U}_i\left(\left\{x_j(t) : v_j \in \mathcal{N}_i^{in} \cup \{i\}\right\}\right), \tag{10}$$

where $\mathcal{U}_i$ is a function of the current state of the node v_i and his in-neighborhood.

Let $\chi(x_1(0), \ldots, x_n(0)) \in \mathbb{R}$ be any function of the initial conditions of all the nodes the *χ-consensus problem* consists of to find a function $\mathcal{U}_i(\cdot)$, such that

$$\lim_{t \to \infty} x_i(t) = \chi\left(x_1(0), \ldots, x_n(0)\right) \quad \forall i = 1, \ldots, n. \tag{11}$$

Let us now discuss the max-consensus problem, where the nodes have to converge to the maximum of the initial conditions; that is, $\chi(\cdot)$ is the maximum of its arguments.

5.1. Max-Consensus. Assuming the graph is connected and undirected, the problem is known to have a solution in finite time [15] (and specifically in no more than n steps) choosing

$$\mathcal{U}_i(\cdot) = \max_{h \in \mathcal{N}_i^{in} \cup \{i\}} x_h(t). \tag{12}$$

In the following we will denote by

$$\overline{x}_i = \text{max-consensus}_i\left(x_i(0), x_j(0) \mid j \neq i, G, t_{\max}\right) \tag{13}$$

the execution of $t_{\max}$ iterations of the max-consensus procedure by the ith agent in a network G, starting from its own initial condition $x_i(0)$ and the "unknown" initial conditions of the other agents, while $\overline{x}_i$ is the state of the ith agent at iteration $t_{\max}$. Such a formalism just represents the execution of the max-consensus or min-consensus by the ith agent, and we assume that all other agents are executing the same algorithm in a synchronous manner, each with its own initial condition.

5.2. Average-Consensus. In the *average-consensus* problem the nodes are required to converge to the average of their initial conditions, that is,

$$\chi(\cdot) = c^T \left[x_1(0) \ \cdots \ x_n(0)\right]^T, \tag{14}$$

where $c^T = (1/n)1_n^T$ and 1_n is a vector with n components all equal to 1.

Let each node choose

$$\mathcal{U}_i(\cdot) = w_{ii} x_i(t) + \sum_{j=1}^{n} w_{ij} x_j(t), \tag{15}$$

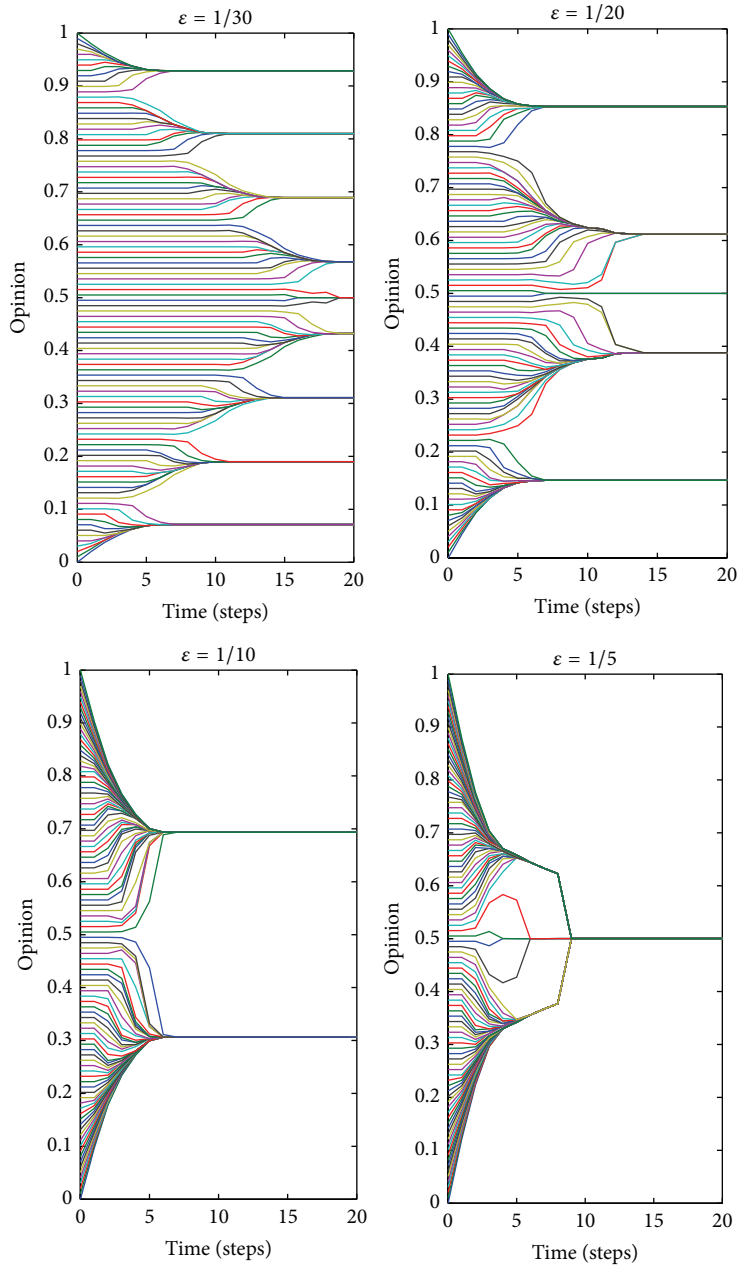

FIGURE 2: Simulation of opinion dynamics for $n = 100$ agents and for different values of ε.

where $w_{ij} = 0$ if $(v_i, v_j) \notin E$. The update strategy for the entire system can be represented as

$$x(t+1) = Wx(t), \tag{16}$$

where the $n \times n$ matrix W contains the terms w_{ij}.

According to [26], this choice of $\mathcal{U}_i(\cdot)$ yields an asymptotical solution if and only if (I) W has a simple eigenvalue at 1 and all other eigenvalues have magnitude strictly less than 1; (II) the left and right eigenvectors of W corresponding to eigenvalue 1 are 1_n and c^T, respectively.

The above condition implies that, if the underlying graph is undirected, it needs to be connected.

A possible choice for W, assuming that each node knows n (or an upper bound for n), is that each node i chooses independently the terms w_{ij} as

$$w_{ij} = \begin{cases} \dfrac{1}{n}, & \text{if } v_j \in \mathcal{N}_i^{in}, \\ 0, & \text{if } v_j \notin \mathcal{N}_i^{in}, \\ 1 - \displaystyle\sum_{l \in \mathcal{N}_l^{in}} w_{il}, & \text{if } i = j, \end{cases} \tag{17}$$

resulting in a matrix W that satisfies the conditions in [26]. Several other choices that yield to asymptotic consensus are possible (e.g., see [27]).

As for the complexity, note that at each time step t each agent i calculates the contribution of each neighbor to the next state; hence we have $O(d\ n\ t_{\max})$. Notice that, however, the algorithm in its basic setting has asymptotic convergence, hence typically $t_{\max} \gg 1$ in order to obtain a good approximation of the asymptotic result. An alternative is to resort to finite-time average-consensus algorithms like the one in [28], but we choose to omit the discussion for the sake of clarity.

In the following we will denote by

$$\overline{x}_i = \text{average-consensus}_i \left(x_i(0), x_j(0) \mid j \neq i, G, t_{\max} \right) \quad (18)$$

the execution of $t_{\max}$ iterations of the average-consensus procedure by the ith agent in a network G, starting from its own initial condition $x_i(0)$ and the "unknown" initial conditions of the other agents, while $\overline{x}_i$ is the state of the ith agent at iteration $t_{\max}$. Again, such a formalism just represents the execution of the max-consensus or min-consensus by the ith agent, and we assume that all other agents are executing the same algorithm in a synchronous manner, each with its own initial condition.

5.3. Network Size Calculation. Combining the max-consensus and the average-consensus algorithms, it is possible to calculate the number of agents n in the network in a distributed way [29].

Specifically, suppose a *leader* is elected via max-consensus over G (e.g., the nodes, each, have a unique identifier and the node with maximum identifier is elected as leader via max-consensus). Now, let the nodes execute an average-consensus algorithm with $\overline{x}_i(0) = 1$ if node v_i is the leader and $\overline{x}_i(0) = 0$ otherwise: the average-consensus yields

$$\lim_{t \to \infty} \overline{x}_i(t) = \frac{1}{n}; \quad (19)$$

hence n can be calculated in a distributed way.

6. Data Clustering with Distance Constraints via Opinion Dynamics and k-Means

In this section we propose an algorithm for data clustering with measurement distance constraints, based on a generalization of the HK model.

Specifically, measurement data is processed by an HK-like opinion dynamics "filter," which eventually segments the data into κ clusters. Moreover, based on the weight of each cluster (namely, represented by the amount of measurement data that has converged to that cluster), data outliers can be filtered out from the original set. More precisely, we assume that each agent i is provided with an initial measurement represented by a vector $x_i \in \mathbb{R}^d$. Every agent has an initial state $x_i(0) \in \mathbb{R}^d$, which is updated according to the following set of iterative rules:

$$\begin{pmatrix} x_{1,i}(t+1) \\ \vdots \\ x_{n,i}(t+1) \end{pmatrix} = A(x(t), \varepsilon) \begin{pmatrix} x_{1,i}(t) \\ \vdots \\ x_{n,i}(t) \end{pmatrix}, \quad (20)$$

for $i = 1, \ldots, d$, where the adjacency matrix $A(x(t), \varepsilon)$ is computed based on the following definition of neighborhood:

$$N_i^*(t) = \left\{ j \in \{1, \ldots, n\} \text{ s.t. } \left\| x_i(t) - x_j(t) \right\| \leq \varepsilon \right\}, \quad (21)$$

where $\| \cdot \|$ is the Euclidean norm. Note that for $d = 1$ the standard HK model is obtained.

In [22] the HK model with vectorial opinions (i.e., $d \geq 2$) is shown to converge in polynomial time.

In order to give an idea of the actual time required for convergence,, we provide the simulation results of Figure 3, where the average instant in which a steady state is reached and the number of clusters obtained are reported in the case $d = 2$ for several choices of $n = 50, 100, 200$ agents and $\varepsilon \in [0.1, 0.5]$; for each choice of n, ε, the average of 100 runs with random initial opinions in $[0, 1]$ is reported. Notice that all the executions reached an exact agreement in finite time.

The proposed approach, as discussed before, does not require the user to specify a value for the parameter k, but it finds a suitable number of clusters based on the parameter ε.

A high value of ε means very large and sparse clusters (eventually also very few of them) while a small value of ε means very compact and small clusters (eventually, many of them).

Note that, as said before, one of the biggest problems of the k-means-like algorithms is that the outliers have to be preprocessed and excluded; otherwise they would influence considerably the quality of the clustering. In the proposed approach, depending on the choice of ε, very far observation is not influenced by the others and is assigned to a singleton (or more in general to a cluster composed of very few elements).

Notice that it is always possible to execute a k-means algorithm with k equal to the number of clusters obtained via HK, in order to attempt to refine the solution found, but this can be done only if the solution of the k-means algorithm does not violate the constraints on the distance among cluster centroids.

The proposed algorithm, appears as a good candidate to allow the clustering of a set of sensors or mobile robots, based on perceived information such as position or other sensorial information (temperature, humidity, etc.).

As for the computational complexity of the extension of the HK model to opinions in $\mathbb{R}^d$, it can be noted that such complexity is the same as executing d scalar HK models; hence it is $O(d\ n^2\ M)$; since the complexity of the centralized k-means algorithm is $O(d\ k\ n\ M)$ [17], the proposed approach determines an increase in computational complexity with respect to the k-means algorithm of a factor $n/k \geq 1$.

7. Distributed HK Opinion Dynamics

As shown in the previous section, the HK opinion dynamics model can be used to provide an admissible, although sub-optimal, solution to the distance-constrained data clustering problem. However, since the topology underlying the HK model is indeed a state-dependent topology, theoretically

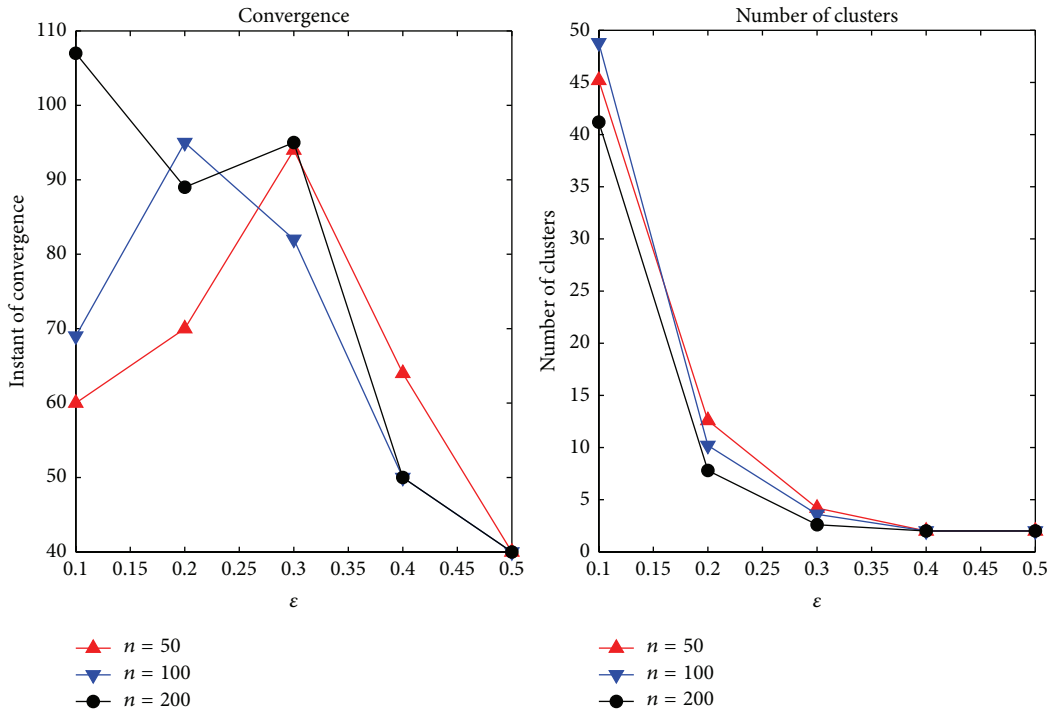

FIGURE 3: Average instant in which a steady state is reached and the number of clusters obtained is reported for several choices of $n = 50, 100, 200$ agents and $\varepsilon \in [0.1, 0.5]$; for each choice of n, ε, the average of 100 runs with random initial opinions in $[0, 1]$ is reported.

each agent may exchange information with another agent, depending just on the difference in their opinions.

In order to adopt the HK model in a distributed perspective, we need to provide a different implementation, as provided in Algorithm 1.

Since the agents have a unique identifier $h = 1, \ldots, n$, for each time step and for each agent h a distributed procedure is executed by all agents in order to calculate $x_h(t+1)$. Specifically, for a specific agent h, each agent i selects $\delta_k^i = x_h(t)$ if $i = j$ and $\delta_k^i = 0$ otherwise. Then the agents execute a max-consensus procedure using δ_k^i as initial condition; as a result of such an operation, each agent i knows $x_h(t)$ and is able to determine wether $\|x_h(t) - x_i(t)\| \leq \varepsilon$ or not. Such knowledge is stored in a variable e_h^i for each agent, while a variable z_h^i is equal to one if $\|x_h(t) - x_i(t)\| \leq \varepsilon$ and zero otherwise.

Using average-consensus, multiplied by n, in order to obtain

$$z_h = \sum_{h=1}^{n} z_h^i, \qquad e_h = \sum_{h=1}^{n} e_h^i \tag{22}$$

the value $x_h^i(t+1)$ is obtained as

$$x_h^i(t+1) = \frac{z_h}{e_h}. \tag{23}$$

for $t = 1, \ldots, M$ **do**
  for $h = 1, \ldots, n$ **do**
    /*Transmit the state of agent h to each other*/
    $$\delta_h^i = \begin{cases} x_h(t) & \text{if } i = h \\ 0 & \text{else} \end{cases};$$
    $\delta_h = \text{max-consensus}_i\left(\delta_h^i, \delta_h^j \mid j \neq i, G_c, n\right)$;
    /*Calculate $x_i(t+1)$*/
    $$e_h^i = \begin{cases} 1 & \text{if } \|x_i(t) - \delta_h\| \leq \varepsilon \\ 0 & \text{else} \end{cases};$$
    $$z_h^i = \begin{cases} x_i(t) & \text{if } \|x_i(t) - \delta_h\| \leq \varepsilon \\ 0 & \text{else} \end{cases}$$
    $e_h = n \cdot \text{average-consensus}\left(e_h^i, e_h^j \mid j \neq i, G, t_{\max}\right)$
    $z_h = n \cdot \text{average-consensus}\left(z_h^i, z_h^j \mid j \neq i, G, t_{\max}\right)$
    $x_h^i(t+1) = \frac{z_h}{e_h}$
    if $i == h$ **then**
      $x_i(t+1) = x_h^i(t+1)$;
    end if
  end for
end for

ALGORITHM 1: Distributed HK opinion dynamics algorithm.

As for the computational complexity of the distributed version, note that, for each step $t = 1, \ldots, M$ and for each agent $h = 1, \ldots, n$, the agents execute a max-consensus

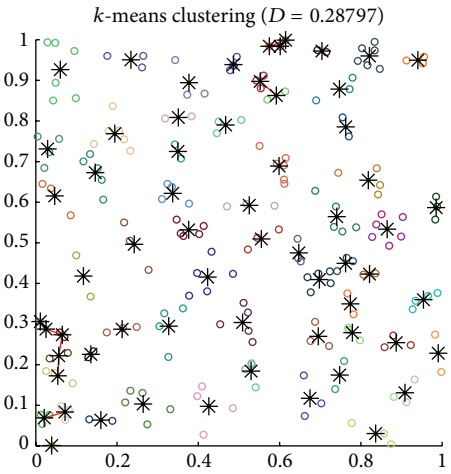

FIGURE 4: Clustering for $n = 200$ observations that are uniformly distributed in the interval $[0, 1]$ and $\varepsilon = 0.06$: the HK opinion dynamics model finds $k = 63$ clusters and $D \approx 0.35$. The solution of the k-means algorithm for $k = 63$ is better in terms of the objective function ($D \approx 0.29$), but it does not respect the distance constraints (red thick lines represent violations).

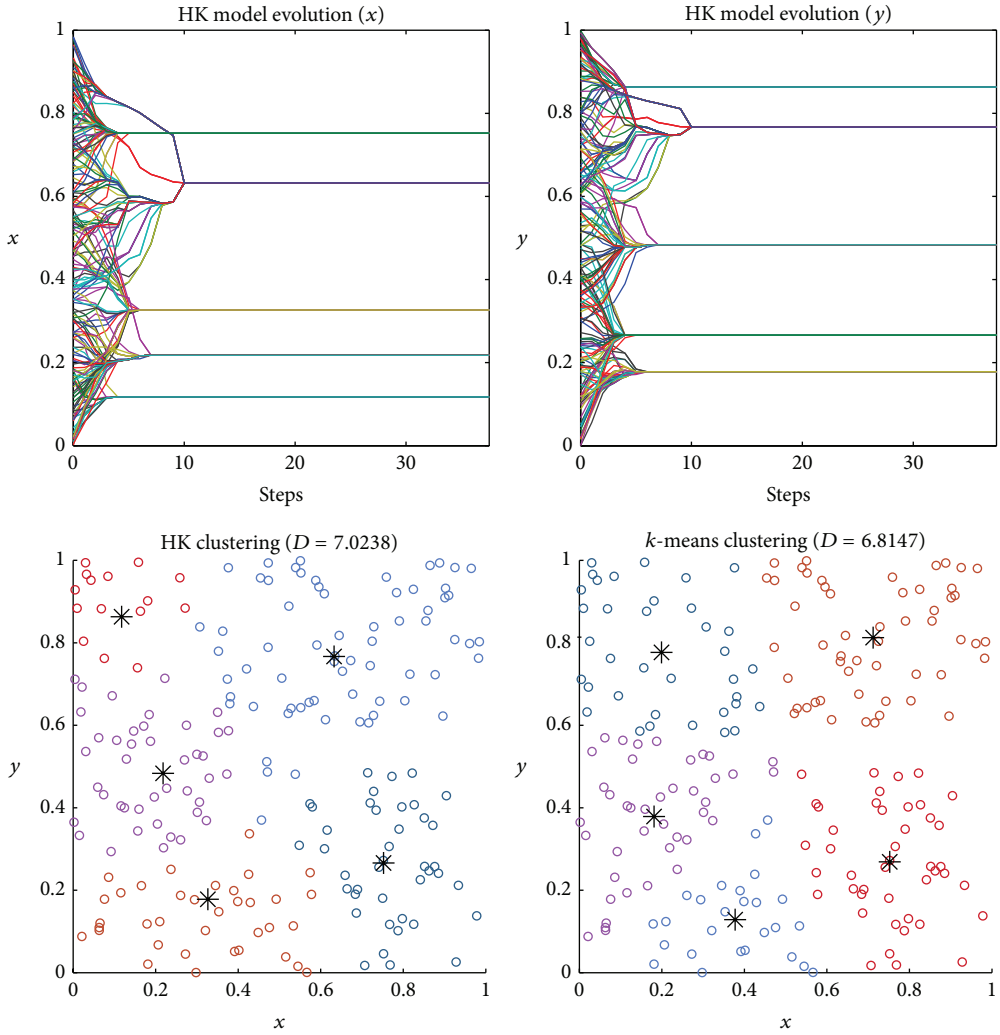

FIGURE 5: Clustering for $n = 200$ observations that are uniformly distributed in the interval $[0, 1]$ and $\varepsilon = 0.18$: the HK opinion dynamics model finds $k = 5$ clusters. The solution of the k-means algorithm for $k = 5$ is better in terms of the objective function D without violating the distance constraints.

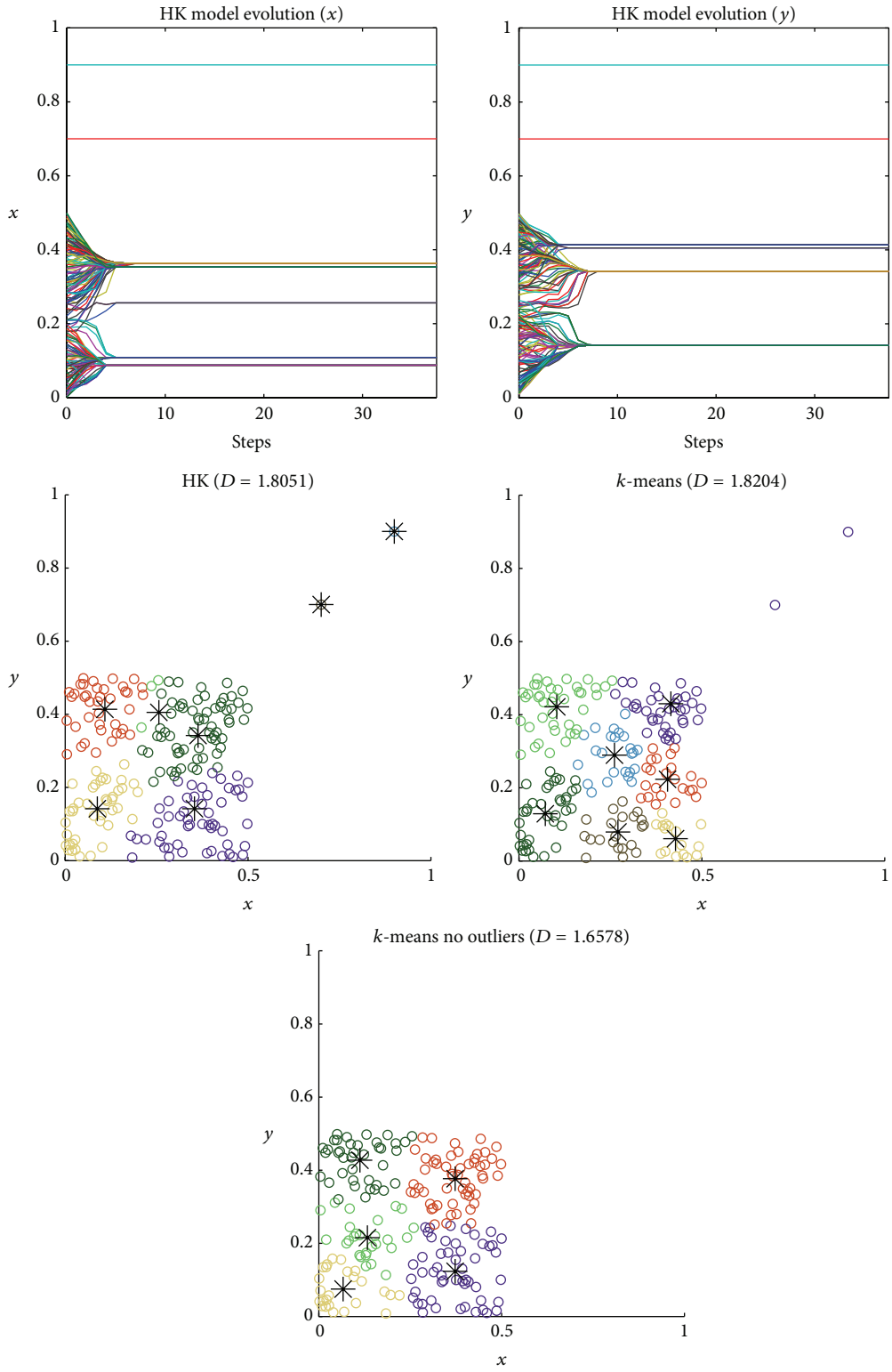

FIGURE 6: Clustering for $n = 200$ observations with $\varepsilon = 0.1$. The HK opinion dynamics model finds $k = 7$ clusters, of which 2 are singleton clusters containing one outlier each. The solution of the k-means algorithm for $k = 7$ is worse than the one found by HK model. If, however, the 2 outliers are removed (thus $k = 5$), the k-means algorithm has better results in terms of the objective function.

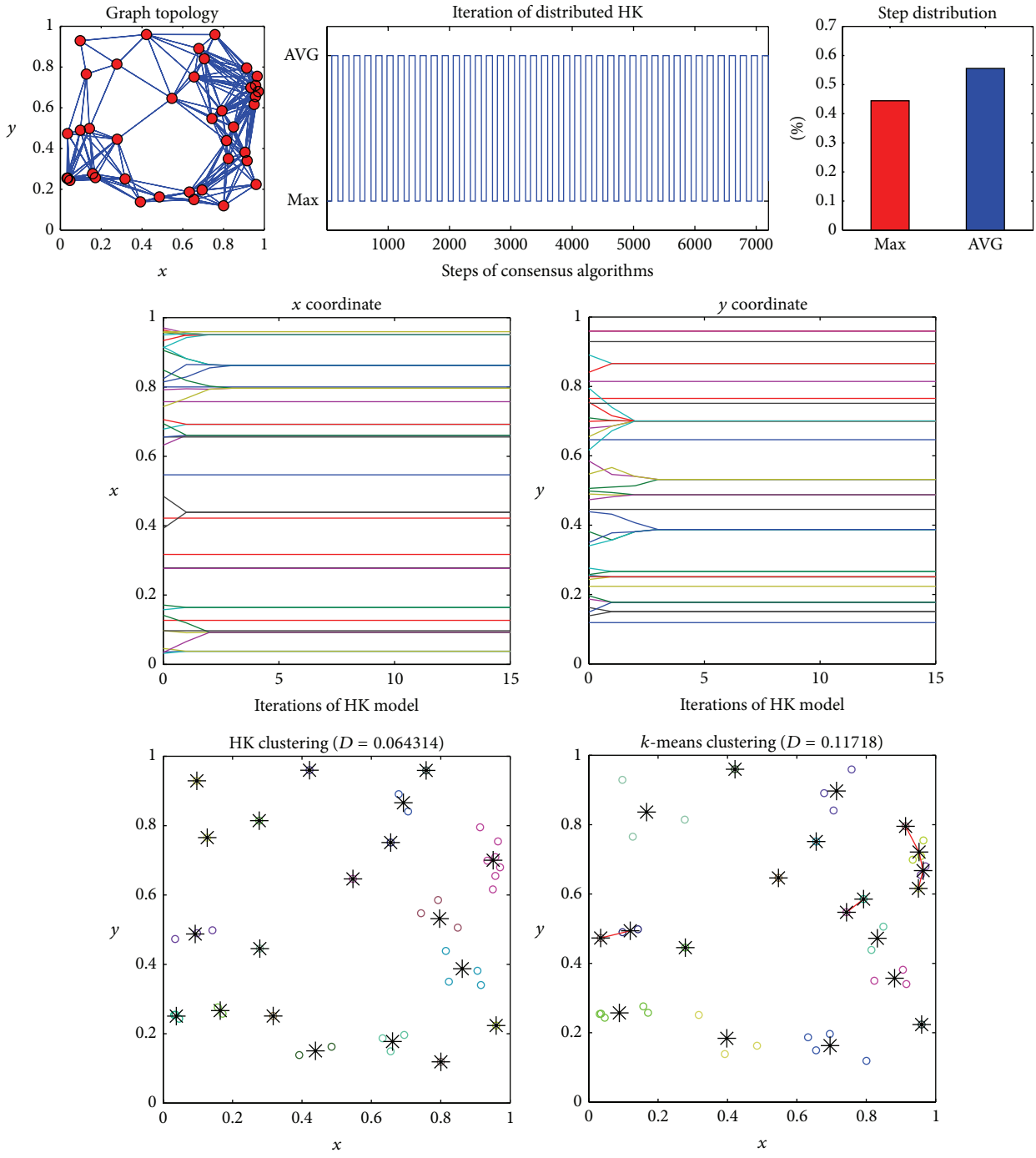

FIGURE 7: Distributed clustering on the positions of $n = 40$ agents with $\varepsilon = 0.1$: the distributed HK opinion dynamics model finds $k = 21$ clusters. The distributed implementation of the algorithm requires alternation between max-consensus algorithms and average-consensus algorithms, as shown by the upper rightmost figure. The solution of the k-means algorithm for $k = 21$ is worse in terms of the objective function D and does not respect the distance constraints (red thick lines in the lower rightmost figure represent violations).

($O(d\ n)$ steps) and two average-consensus algorithms ($O(d\ t_{\max})$ steps), both with initial conditions in $\mathbb{R}^d$, and the complexity is $O(d\ n\ M\ \max\{t_{\max}, n\})$ where $t_{\max}$ is the number of iterations of the average- consensus algorithm. Since, typically, $t_{\max} > n$, the distributed setting has $t_{\max}/n$ times the complexity of the centralized algorithm. Moreover, since the computational complexity of the distributed k-means algorithm is $O(d\ k\ n\ M)$ the proposed distributed

algorithm has $t_{\max}/(k\ n)$ times the complexity of the distributed k-means algorithm.

8. Numeric Examples

As discussed above, the k-means algorithm is generally unable to solve the data clustering problem with distance constraints. In this section, some examples are reported in order to show the effectiveness of the proposed approach. A comparative simulation between the HK model and k-means algorithm is first addressed. Afterwards the potentiality of the proposed mixed approach is showed; then the distributed implementation is discussed.

Figure 4 shows an example in $\mathbb{R}^2$ with $\varepsilon = 0.6$ and $n = 200$ observations. The application of the HK opinion dynamics model yields $k = 63$ clusters and $D \approx 0.35$. Unfortunately, the k-means algorithm finds a solution for $k = 63$ which, although having $D \approx 0.29$, is not feasible for Problem 1 (violations of the constraints are shown with red lines).

Figure 5 shows a case where $n = 200$, $\varepsilon = 0.18$ and the HK opinion dynamics model gives $k = 5$ clusters. Using the k-means algorithm for $k = 5$ a better solution is obtained, and the constraints are not violated; hence in this case postprocessing the result of the HK model via k-means algorithm yields a better result.

Figure 6 shows the ability of the proposed methodology to isolate the outliers. For $n = 200$ and $\varepsilon = 0.1$ the HK opinion dynamics model finds $k = 7$ clusters, of which 2 are singletons each containing an outlier. Executing a k-means algorithm for $k = 7$ gives worse results in terms of the objective function with respect to the HK approach. If, conversely, the two outliers are removed (k becomes equal to 5), then the k-means algorithm has better results in terms of the objective function.

Figure 7, eventually, shows an example of application of the distributed HK model provided in Algorithm 1, in a case where $n = 40$ agents, each with a random position in $[0, 1]^2$, having to be clustered depending on their positions in a way that the centroids are not closer than $\varepsilon = 0.1$. The simulation was executed for $M = 15$ iterations, and the average-consensus algorithms were executed each $t_{\max} = 100$ steps. The topology of the network of agents is given in the upper left plot, while the middle plots show the results of the distributed HK model for the x and y coordinates. The consensus steps performed by each agent during one iteration of the distribute HK algorithm are reported in the upper central plot, where "AVG" stands for average-consensus and "MAX" stands for max-consensus, while the distribution of max-consensus and average-consensus steps over the entire execution of the algorithm is reported in the upper rightmost plot. The lower plots, eventually, show the results of the distributed HK model in terms of the clustering in $[0, 1]^2$: the agents are divided in $k = 20$ groups, and the objective function has a value $D \approx 0.06$. Notice that, in this case, the k-means algorithm with $k = 20$ yields a worst solution both in terms of the objective function $D \approx 0.12$ and in terms of violation of the constraints (5 constraints are violated and are highlighted with red thick segments).

9. Conclusions and Future Work

In this paper a distributed algorithm for a sensors network to solve a data clustering problem is provided, with the constraint that the centroids of the clusters must be within ε.

The proposed approach is based on the HK opinion dynamics model, which finds admissible, although suboptimal, solutions without requiring a user-specified number of desired clusters and applies the k-means algorithm to find a better solution.

In its original application, the HK opinion dynamics model is a centralized algorithm. In this paper, a distributed implementation is provided instead, based on a combination of consensus algorithms.

Future work will focus on providing a bound for convergence time in high-dimensional spaces and testing the algorithm in a real world scenario, including noise and packet loss.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] L. Di Paola, M. de Ruvo, P. Paci, D. Santoni, and A. Giuliani, "Protein contact networks: an emerging paradigm in chemistry," *Chemical Reviews*, vol. 113, no. 3, pp. 1598–1613, 2013.

[2] A. P. Gasch and M. B. Eisen, "Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering," *Genome Biology*, vol. 3, no. 11, pp. 1–22, 2002.

[3] H. P. Ng, S. H. Ong, K. W. C. Foong, P. S. Goh, and W. L. Nowinski, "Medical image segmentation using k-means clustering and improved watershed algorithm," in *Proceedings of the 7th IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 61–65, March 2006.

[4] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B. Methodological*, vol. 39, no. 1, pp. 1–38, 1977.

[6] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ, USA, 1988.

[7] K. Bennett, P. Bradley, and A. Demiriz, "Constrained k-means clustering," Tech. Rep., Microsoft, 2000.

[8] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrodl, "Constrained kmeans clustering with background knowledge," in *Proceedings of the International Conference on Machine Learning*, pp. 577–584, 2001.

[9] I. Davidson and S. S. Ravi, "Clustering with constraints: feasibility issues and the k-means algorithm," in *Proceedings of the 5th SIAM International Conference on Data Mining (SDM '05)*, vol. 5, pp. 201–211, 2005.

[10] J. Wu and Y. Jiao, "Clustering dynamics of complex discrete-time networks and its application in community detection," *Chaos*, vol. 24, no. 3, Article ID 033104, 2014.

[11] G. Oliva, D. La Manna, A. Fagiolini, and R. Setola, "Distance-constrained data clustering by combined k-means algorithms and opinion dynamics filters," in *Proceedings of the 22nd Mediterranean Conference of Control and Automation (MED '14)*, pp. 612–619, Palermo, Italy, June 2014.

[12] A. Gasparri and G. Oliva, "Fuzzy opinion dynamics," in *Proceedings of the American Control Conference (ACC '12)*, pp. 5640–5645, Montreal, Canada, June 2012.

[13] R. Hegselmann and U. Krause, "Opinion dynamics and bounded confidence: models, analysis and simulation," *Journal of Artifical Societies and Social Simulation*, vol. 5, no. 3, 2002.

[14] A. Fagiolini and A. Bicchi, "On the robust synthesis of logical consensus algorithms for distributed intrusion detection," *Automatica*, vol. 49, no. 8, pp. 2339–2350, 2013.

[15] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[16] J. Wu, L. Jiao, and R. Ding, "Average time synchronization in wireless sensor networks by pairwise messages," *Computer Communications*, vol. 35, no. 2, pp. 221–233, 2012.

[17] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematics, Statistics and Probability*, pp. 281–296, 1967.

[18] G. Oliva, R. Setola, and C. Hadjicostis, "Distributed *k*-means algorithm," http://arxiv.org/abs/1312.4176.

[19] E. E. Miciolino, G. Oliva, and R. Setola, "Distributed opinion dynamics with heterogeneous reputation," *International Journal of System of Systems Engineering*, vol. 4, no. 3-4, pp. 277–290, 2013.

[20] P. Brucker, "On the complexity of clustering problems," in *Optimization and Operations Research*, pp. 45–54, Springer, Berlin, Germany, 1978.

[21] A. Mirtabatabaei and F. Bullo, "Opinion dynamics in heterogeneous networks: convergence conjectures and theorems," *SIAM Journal on Control and Optimization*, vol. 50, no. 5, pp. 2763–2785, 2012.

[22] A. Bhattacharyya, M. Braverman, B. Chazelle, and H. L. Nguyen, "On the convergence of the Hegselmann-Krause system," in *Proceedings of the 4th ACM Conference on Innovations in Theoretical Computer Science (ITCS '13)*, pp. 61–66, ACM, January 2013.

[23] V. D. Blondel, J. M. Hendrickx, and J. N. Tsitsiklis, "On Krause's multi-agent consensus model with state-dependent connectivity," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2586–2597, 2009.

[24] J. C. Dittmer, "Consensus formation under bounded confidence," *Nonlinear Analysis: Theory Methods & Applications*, vol. 47, no. 7, pp. 4615–4622, 2001.

[25] J. Lorenz, "A stabilization theorem for dynamics of continuous opinions," *Physica A: Statistical Mechanics and its Applications*, vol. 355, no. 1, pp. 217–223, 2005.

[26] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[27] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proceedings of the American Control Conference (ACC '05)*, pp. 1859–1864, June 2005.

[28] S. Sundaram and C. N. Hadjicostis, "Finite-time distributed consensus in graphs with time-invariant topologies," in *Proceedings of the American Control Conference (ACC '07)*, pp. 711–716, New York, NY, USA, July 2007.

[29] I. Shames, T. Charalambous, C. N. Hadjicostis, and M. Johansson, "Distributed network size estimation and average degree estimation and control in networks isomorphic to directed graphs," in *Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton '12)*, pp. 1885–1892, IEEE, October 2012.