Density-Based Clustering in Cloud-Oriented Collaborative Multi-Agent Systems

Jelena Fiosina and Maksims Fiosins *

Clausthal University of Technology,
Institute of Informatics,
Julius-Albert Str. 4, D-38678, Clausthal-Zellerfeld, Germany
{Jelena.Fiosina,Maksims.Fiosins}@gmail.com

Abstract. The development of new reliable data processing and mining methods based on the synergy between cloud computing and the multi-agent paradigm is of great importance for contemporary and future software systems. Cloud computing provides huge volumes of data and computational resources, whereas the agents make the system components more autonomous, cooperative, and intelligent. This creates the need and gives a very good basis for the development of data analysis, processing, and mining methods to enhance the new agent-based cloud computing (ABCC) architecture. Ad-hoc networks of virtual agents are created in the ABCC architecture to support the dynamic functionality of provided services, and data processing methods are very important at the input data processing and network parameter estimation stage. In this study, we present a decentralized kernel-density-based clustering algorithm that fits with the general architecture of ABCC systems. We conduct several experiments to demonstrate the capabilities of the new approach and analyse its efficiency.

Keywords: Cloud computing architecture, distributed data processing and mining, multiagent systems, decentralized clustering, kernel density estimation

1 Introduction

Cloud computing (CC) is developing rapidly due to new communication and mobile technologies, and it has been introduced recently as a new model for delivering computational resources over a network. Motivated by future Internet technologies such as the Internet of Things, it provides end users with simple on-demand access to services, such as applications or databases, through lightweight mobile applications. Simultaneously, the complexity of the infrastructure is hidden in the cloud, which allows users to focus on their goals instead of the infrastructure complexity [11].

*The research leading to these results has received funding from the EU 7th Framework Programme (FP7/2007-2013) under grant agreement No. PIEF-GA-2010-274881.

It should be noted that cloud-based systems are complex systems, distributed by regions, services, and providers. A popular paradigm for modelling complex distributed systems is the multi-agent system (MAS). Agents in an MAS are autonomous and intelligent, and capable of cooperating with each other and interacting with the environment.

The synergy between MAS and CC models (Fig. 1) reveals new perspectives for developing future intelligent information and management systems. CC provides elastic services, high performance, and scalable data storage to a large and increasing number of users [9]. MAS provides intelligent system behaviour and adaptive mechanisms for data processing, decision-making, and learning to better satisfy user needs as well as intelligent interaction and cooperation mechanisms for dealing with system distribution. In other words, MAS makes CC more intelligent, and CC makes MAS more powerful and accessible.

Agent-based cloud computing (ABCC) [4],[11] is a new research direction that enhances existing complex systems modelled using MASs with new modern technologies from communication and data analysis fields to make corresponding applications more intelligent. Talia [11] considered the implementation of CC with software agents to create intelligent cloud services. CC can offer a very powerful, reliable, predictable, and scalable computing infrastructure for the execution of an MAS implementing complex agent-based applications for modelling and simulation. On the other hand, software agents can be used as the basic components for implementing intelligence in clouds, making them more adaptive, flexible, and autonomic in resource management, service provisioning, and running large-scale applications.

The high availability of mobile devices with sensors and permanent Internet connections means that huge amounts of data are available on CC systems. The appropriate use of such data can create a complete picture of the environment for agents in an MAS, enabling optimal decisions. Hence, the novel mechanisms and algorithms for data processing and mining in ABCC are of high importance [8], [5]. Large amounts of data must be found, collected, aggregated, processed, and analysed for optimal decision-making and behaviour strategy determination. Although information is virtually centralized by cloud technologies, it should be managed in a decentralized fashion, creating challenges for research in this area.

In our previous work, we considered decentralized data processing models, such as regression forecasting and change-point analysis, and applied them for optimal decision making in an MAS, such as optimal route selection [3] or lane/speed adaption [5] in traffic. We demonstrated that appropriate data coordination mechanisms can provide almost the same forecasting accuracy as a model with central authority [2].

In this study, we focus on decentralized data clustering, which is an important data pre-processing step in cloud data repositories. By grouping similar data together, it is possible to construct more precise forecasting models as well as use only typical data representatives in the decision-making process. Complex forms of clusters require non-parametric, computationally intensive approaches such as kernel-density (KD) [6] clustering (Fig. 1). Fast KD clustering was described

by Hinnenburg and Gabriel [7]. The distributed (with central authority) version of KD clustering (KDEC scheme) was considered in [8]. Another graph-oriented decentralized clustering method not based on KD was presented in [10].

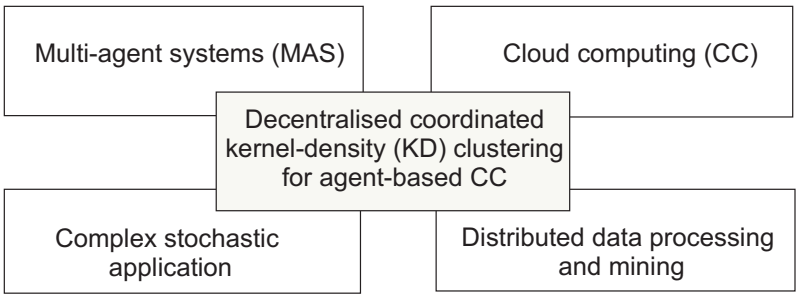

Fig. 1. Synergy of cloud computing and multiagent systems to meet decentralized density-based clustering for some application

The decentralised KD clustering algorithm was motivated by and developed for use in ABCC. The developed algorithm is an extension of the approach [7] for the multivariate case and developing a data coordination scheme based on the transmission of the number of nearest data points from the same cluster.

The remainder of this paper is organized as follows. Section 2 introduces KD clustering. In Section 3, we develop the decentralised cooperative KD clustering algorithm. In Section 4, we conduct several experiments and analyse the efficiency of the suggested approach. The last section presents the conclusion and discusses the opportunities for future work.

2 Kernel Density (KD) Clustering

Now let us formulate the clustering problem and describe the KD clustering algorithm. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, $\mathbf{x}_i \in R^d$ be a dataset to be clustered into k non-overlapping subsets $S_1, S_2, \ldots, S_k$.

Non-parametric clustering methods are well suited for exploring clusters without building a generative model of the data. KD clustering consists of a two-step procedure: estimation and optimisation. During the estimation step, the probability density of the data space is directly estimated from data instances. During the optimisation step, a search is performed for densely populated regions in the estimated probability density function.

Let us formalize the estimation step. The density function is estimated by defining the density at any data object as being proportional to a weighted sum of all objects in the data-set, where the weights are defined by an appropriately chosen kernel function [8].

A KD estimator is

$$\hat{\Psi}^{[\mathbf{X}]}(\mathbf{x}) = \frac{1}{N} \sum_{\mathbf{x}_i \in \mathbf{X}} |\mathbf{H}|^{-1} K\left(\mathbf{H}^{-1}\left\|\mathbf{x}-\mathbf{x}_i\right\|\right) = \frac{1}{N} \sum_{\mathbf{x}_i \in \mathbf{X}} K_{\mathbf{H}}\left(\left\|\mathbf{x}-\mathbf{x}_i\right\|\right), \quad (1)$$

where $\|\mathbf{x}-\mathbf{x}_i\|$ is a distance between $\mathbf{x}_i$ and $\mathbf{x}$, $\mathbf{H}$ is a *bandwidth* matrix, $K(\mathbf{x})$ is a kernel function, $K_{\mathbf{H}}(\bullet) = |\mathbf{H}|^{-1} K\left(\mathbf{H}^{-1}\bullet\right)$ [6].

$K(\mathbf{x})$ is a real-valued, non-negative function on R^d and has finite integral over R^d. We use the multivariate Gaussian function in our study: $K(\mathbf{x}) = (2\pi)^{-d/2} exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{x}\right)$. The bandwidth matrix $\mathbf{H}$ is a $d \times d$ positive-definite matrix that controls the influence of data objects and smoothness of the estimate. If no information is available with regard to correlation between factors, a diagonal matrix $\mathbf{H} = diag(h_1, \ldots, h_d)$ can be used.

Let us now formalize the optimisation step. This step detects maxima of KD and groups all of the data objects in their neighbourhood into corresponding clusters. We use a hill climbing method for KD maxima estimation with Gaussian kernels (DENCLUE2) [7] and modify the technique for the multivariate case. This method converges towards a local maximum and adjusts the step size automatically at no additional costs. Other optimization methods (DENCLUE) [7] require more steps and additional computations for step size detection.

Each KD maximum can be considered as the centre of a point cluster. With centre-defined clusters, every local maximum of $\hat{\Psi}(\cdot)$ corresponds to a cluster that includes all data objects that can be connected to the maximum by a continuous, uphill path in the function of $\hat{\Psi}(\cdot)$. Such centre-defined clusters allows for arbitrary-shaped clusters to be detected, including non-linear clusters. An arbitrary-shape cluster is the union of centre-defined clusters that have maxima that can be connected by a continuous, uphill path.

The goal of the hill climbing procedure is to maximize the KD $\hat{\Psi}^{[\mathbf{X}]}(\mathbf{x})$. By setting the gradient $\nabla\hat{\Psi}^{[\mathbf{X}]}(\mathbf{x})$ of KD to zero and solving the equation $\nabla\hat{\Psi}^{[\mathbf{X}]}(\mathbf{x}) = 0$ for $\mathbf{x}$, we get:

$$\mathbf{x}^{(l+1)} = \frac{\sum_{\mathbf{x}_i \in \mathbf{X}} K_{\mathbf{H}}\left(\left\|\mathbf{x}^{(l)}-\mathbf{x}_i\right\|\right)\mathbf{x}_i}{\sum_{\mathbf{x}_i \in \mathbf{X}} K_{\mathbf{H}}\left(\left\|\mathbf{x}^{(l)}-\mathbf{x}_i\right\|\right)}. \quad (2)$$

The formula (2) can be interpreted as a normalized and weighted average of the data points. The weights for each data point depend on the influence of the corresponding kernels on $\mathbf{x}^{(l)}$. Hill climbing is initiated at each data point $\mathbf{x}_i \in \mathbf{X}$ and is iterated until the density does not change, i.e. $[\hat{\Psi}^{[X]}(\mathbf{x}_i^{(l)}) - \hat{\Psi}^{[X]}(\mathbf{x}_i^{(l-1)})]/\hat{\Psi}^{[X]}(\mathbf{x}_i^{(l)}) \leq \epsilon$, where ϵ is a small constant. The end point of the hill climbing algorithm is denoted by $\mathbf{x}_i^* = \mathbf{x}_i^{(l)}$, corresponding to a local maximum of KD.

Now we should determine a cluster for $\mathbf{x}_i$. Let $\mathbf{X}^c = \{\mathbf{x}_1^c, \mathbf{x}_2^c, \ldots\}$ be an ordered set of already identified cluster centres (initially, we suppose $\mathbf{X}^c = \emptyset$). First we find an index of the nearest cluster centre from $\mathbf{x}_i^*$ in the set $\mathbf{X}^c$:

$$nc(\mathbf{x}_i^*) = \arg\min_{j:\mathbf{x}_j^c \in \mathbf{X}^c} \left\|\mathbf{x}_j^c - \mathbf{x}_i^*\right\|.$$

If the nearest cluster centre is close to $\mathbf{x}_i^*$, then the point $\mathbf{x}_i$ is included in this cluster; otherwise, the point is used as a cluster centre to form a new cluster

$$\Lambda(\mathbf{x}_i) \leftarrow \begin{cases} nc(\mathbf{x}_i^*) & \text{if } \frac{\left\|\mathbf{x}^c_{nc(\mathbf{x}_i^*)} - \mathbf{x}_i^*\right\|}{\mathbf{x}_i^*} \leq \delta, \\ |\mathbf{X}^c| + 1 \text{ otherwise.} \end{cases}$$

where δ is a small constant and $\Lambda(x)$ is a class labeling function. In the second case, we also create a new cluster centre: $\mathbf{X}^c \leftarrow \mathbf{X}^c \cup \{\mathbf{x}_i^*\}$.

3 Decentralized KD Clustering

In this section, we describe the cooperation for sharing the clustering experience among the agents in a network. While working with streaming data, one should take into account two main facts. The nodes should coordinate their clustering experience over some previous sampling period and adapt quickly to the changes in the streaming data, without waiting for the next coordination action.

Let us first discuss the cooperation technique (Fig. 2). We introduce the following definitions. Let $\mathbf{A} = \{A^j \mid 1 \leq j \leq p\}$ be a group of p agents. Each A^j has a local dataset $\mathbf{D}^j = \{\mathbf{x}_t^j \mid t = 1. \ldots, N^j\}$, where $\mathbf{x}_t^j \in R^d$. In order to underline the dependence of the KD function (1) on the local dataset of A^j, we denote the KD function by $\hat{\Psi}^{[\mathbf{D}^j]}(\mathbf{x})$.

Consider a case when some agent A^i is unable to classify (after optimisation has formed a new or small cluster) some future data point $\mathbf{x}_t^i$ because it does not have sufficient data in the neighbourhood of this point. It sends the data point $\mathbf{x}_t^i$ to the other neighbouring agents. Each A^j that has received the request classifies $\mathbf{x}_t^i$ using its own KD function $\hat{\Psi}^{[\mathbf{D}^j]}(\mathbf{x}_t^i)$ and performs the optimisation step to identify the cluster for this point. Let $n_{j,i}$ be a number of points in the cluster of $\mathbf{x}_t^i$, not including $\mathbf{x}_t^i$ itself. In the case of successful clustering ($n_{j,i} > 0$), A^j forms an answer $\mathbf{D}^{j,i}$ with c nearest points to the requested data point from the same cluster as $\mathbf{x}_t^i$ (or all points from the cluster, if $n_{j,i} \leq c$). Let $c_{j,i}$ be a number of points in the answer $\mathbf{D}^{j,i}$. The agent A^j sends $\mathbf{D}^{j,i}$ together with $c_{j,i}$ and $n_{j,i}$ to A^i.

After receiving all the answers, A^i forms a new dataset $\hat{\mathbf{D}}^{j,i}$. The next problem is the updating of the KD function of A^i with respect to the new knowledge $\mathbf{D}^{j,i}$. Density estimates (1) of each agent are additive, i.e. the aggregated density estimate $\hat{\Psi}^{[\mathbf{D}^i]}(\mathbf{x})$ can be decomposed into the sum of the local density estimates, one estimate for every dataset $\mathbf{D}^{j,i}$:

$$\hat{\Psi}^{[\hat{\mathbf{D}}_i]}(\mathbf{x}) = w_i \cdot \hat{\Psi}^{[\mathbf{D}^i]}(\mathbf{x}) + \frac{(1-w_i)}{\sum\limits_{A^j \in \mathbf{G}^i} n_{j,i}} \sum_{A^j \in \mathbf{G}^i} n_{j,i} \hat{\Psi}^{[\mathbf{D}^{j,i}]}(\mathbf{x}), \tag{3}$$

where w_i is a weight used for the agent's own local observations.

After updating its KD function, A^i can perform a hill-climbing optimisation procedure to identify clusters in its local data space.

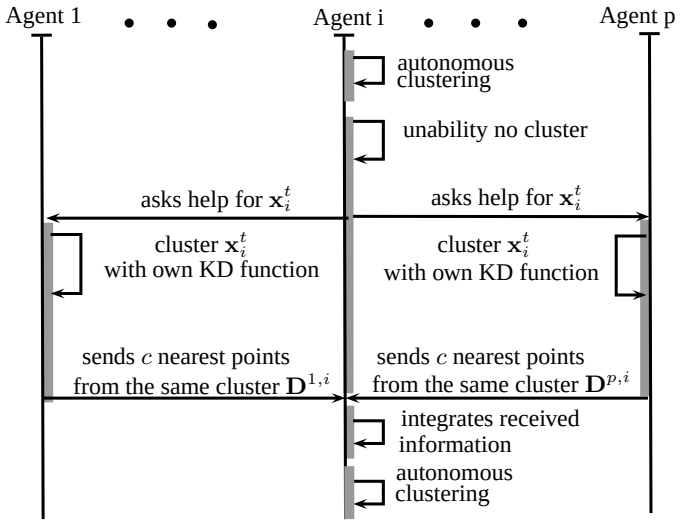

Fig. 2. Interaction between agents

To measure the **clustering similarity** [1] among the agents $A^i \in \mathbf{A}$ we use the following representation of a class labeling by a matrix C with components:

$$C_{i,j} = \begin{cases} 1 \text{ if } x_i \text{ and } x_j \text{ belong to the same cluster and } i \neq j, \\ 0 \text{ otherwise.} \end{cases}$$

Let two labelings have matrix representations $C^{(1)}$ and $C^{(2)}$, respectively. We define a dot product that computes the number of pairs clustered together $\left\langle C^{(1)}, C^{(2)} \right\rangle = \sum_i \sum_j C_{i,j}^{(1)} C_{i,j}^{(2)}$. The Jaccard's similarity measure can be expressed as

$$J(C^{(1)}, C^{(2)}) = \frac{\left\langle C^{(1)}, C^{(2)} \right\rangle}{\left\langle C^{(1)}, C^{(1)} \right\rangle + \left\langle C^{(2)}, C^{(2)} \right\rangle - \left\langle C^{(1)}, C^{(2)} \right\rangle}. \tag{4}$$

4 Experimental results

We consider a clustering model with decentralised coordinated architecture. The agents made a local clustering of their observations and used cooperative mechanisms to adjust the cluster information according to those of other agents. The amount of information transmitted was lower than that in the centralised model, because it requires no transmission of all global data.

We simulate 10 agents with the initial experience, which varies in the range from 10 to 100 observations. Most simulation experiments ran for 200 time units. For our experiments we assume that all observations are homogeneous and the

agents try to estimate the same clusters. The initial global two-dimensional sample data are presented at Fig. 3, where one can see five clusters. The points are located at the normally distributed distances from the cluster centres. Agents take random subsets from this global dataset and try to estimate the clusters by only part of observations. One data synchronization step is demonstrated at

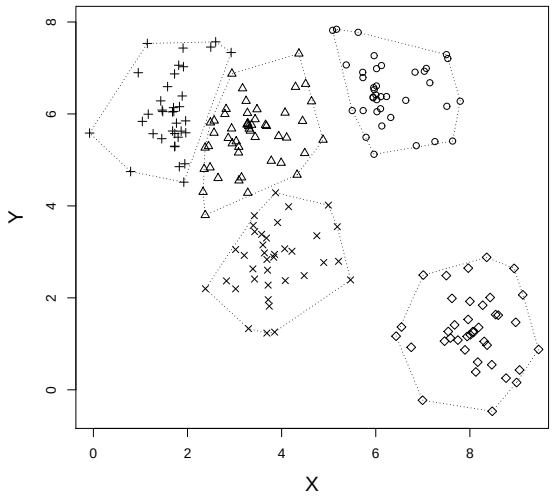

Fig. 3. All observations form clusters

Fig. 4. The agent that has difficulties with a point sends a help request. The receiving agent clusters the point using its own data and detects corresponding cluster. It sends an answer from three nearest points in the cluster back to the requesting agent. The requesting agent adds received data to its own and makes new clustering. This allows to improve clustering similarity of these two agents from 0.011 to 0.037 as well as clustering similarity of the requesting agent with the 'ideal' clustering from 0.004 to 0.006.

We demonstrate now a system dynamics for a different number of transmitted points (Fig. 5). Clustering similarity (right) increases faster for a bigger number of the transmitted points, but the number of communication events (left) decreases faster. However, we note that one communication event is more expensive for a bigger number of transmitted points, but supplies more information.

Quality of the agent models was also checked by a cross-validation technique (Fig. 6) at the beginning (left) and at the end (right) of the simulation. These histograms show a probability distribution of a similarity at the beginning and at the end of the simulation process. One can see that the similarity peak moves to bigger value during the coordination procedure.

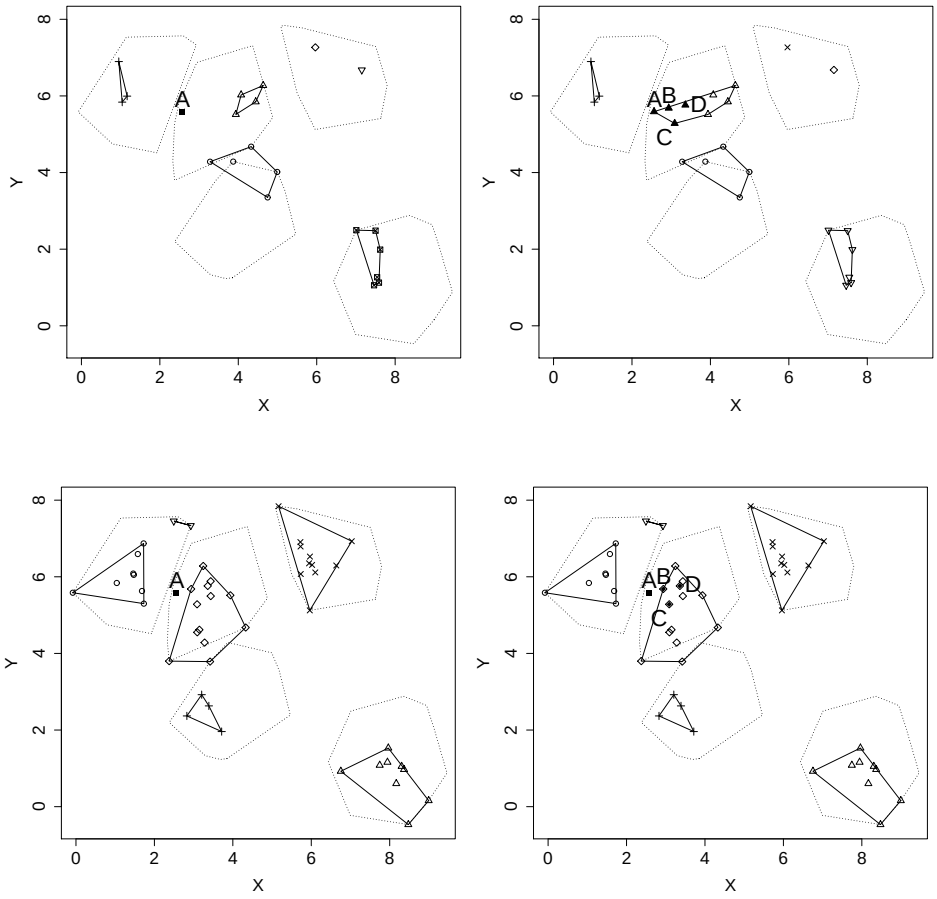

Fig. 4. A communication step between the requesting (top) and helping (bottom) agents. The requesting agent asks for help for point A (top left), the helping agent finds a corresponding cluster (bottom left), and sends the nearest three points B, C, D to the helping agent (bottom right). The helping agent adds the points to its data and makes new clustering (top right).

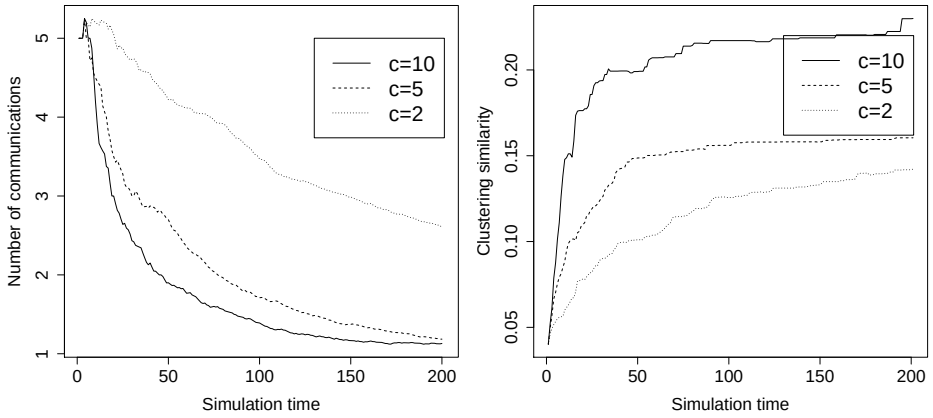

Fig. 5. A number of communication events (left) and similarity of agents' clusters (right) over time

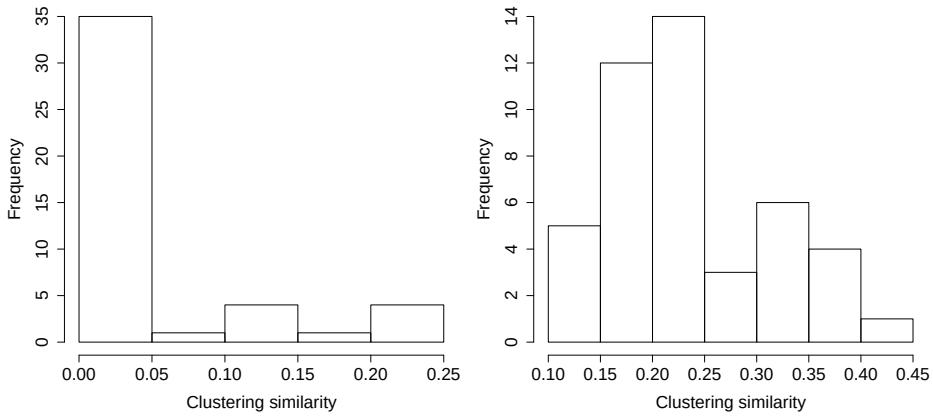

Fig. 6. Frequencies at the beginning (left) and at the end (right) of simulation

5 Future Work and Conclusions

We developed the coordinated decentralized kernel-density clustering approach for agent-based cloud computing architecture. The data coordination scheme is based on the transmission of a several nearest points from the same cluster. An experimental validation of the developed algorithm was also performed. Demonstrated algorithms of collaborative clustering can be applied in cloud-based systems from various domains (e.g. traffic, logistics, energy). Our future work is devoted to the development of new coordination schemes in proposed decentralised clustering approach as well as the application of this algorithm to real-world data.

References

1. Ben-Hur, A., Elisseeff, A., Guyon, I.: A stability based method for discovering structure in clustered data. In: Pacific Sym. on Biocomputing 7, pp. 6–17 (2002)
2. Fiosina, J., Fiosins, M.: Chapter 1: Cooperative regression-based forecasting in distributed traffic networks. In: Q.A. Memon (ed.) Distributed Network Intelligence, Security and Applications, pp. 3–37. CRC Press, Taylor and Francis Group (2013)
3. Fiosina, J., Fiosins, M.: Selecting the shortest itinerary in a cloud-based distributed mobility network. In: S.O. et al. (ed.) Proc. of 10th Int. Conf. on Distributed Computing and AI (DCAI 2013), *Adv. in Int. Syst. and Comp.*, vol. 217, pp. 103–110. Springer-Verlag, Berlin/Heidelberg (2013)
4. Fiosina, J., Fiosins, M., Müller, J.P.: Mining the traffic cloud: Data analysis and optimization strategies for cloud-based cooperative mobility management. In: Proc. of Int. Sym. on Management Int. Systems, *Adv. in Int. Syst. and Comp.*, vol. 220, pp. 25–32. Springer-Verlag, Berlin/Heidelberg (2013)
5. Fiosins, M., Fiosina, J., Müller, J., Görmer, J.: Agent-based integrated decision making for autonomous vehicles in urban traffic. Adv. in Int. and Soft Comp. **88**, 173–178 (2011)
6. Härdle, W., Müller, M., Sperlich, S., Werwatz, A.: Nonparametric and Semiparametric Models. Springer, Berlin/Heidelberg (2004)
7. Hinneburg, A., Gabriel, H.H.: Denclue 2.0: Fast clustering based on kernel density estimation. Adv. in Intelligent Data Analysis VII, LNCS **4723**, 70–80 (2007)
8. Klusch, M., Lodi, S., Moro, G.: Agent-based distributed data mining: The KDEC scheme. In: AgentLink, pp. 104–122 (2003)
9. M. Armbrust, e.a.: A view of cloud computing. Communications of the ACM **53**(4), 50–58 (2010)
10. Ogston, E., Overeinder, B., van Steen, M., Brazier, F.: A method for decentralized clustering in large multi-agent systems. In: Proc. of 2nd Int. Conf. on Autonomous Agents and Multiagent Systems, pp. 789–796 (2003)
11. Talia, D.: Cloud computing and software agents: Towards cloud intelligent services. Proc. of the 12th Workshop on Objects and Agents **741**, 2–6 (2011)