

RESEARCH

Open Access



Robust and adaptive diffusion-based classification in distributed networks

Patricia Binder* , Michael Muma and Abdelhak M. Zoubir

Abstract

Distributed adaptive signal processing and communication networking are rapidly advancing research areas which enable new and powerful signal processing tasks, e.g., distributed speech enhancement in adverse environments. An emerging new paradigm is that of multiple devices cooperating in multiple tasks (MDMT). This is different from the classical wireless sensor network (WSN) setup, in which multiple devices perform one single joint task. A crucial first step in order to achieve a benefit, e.g., a better node-specific audio signal enhancement, is the common unique labeling of all relevant sources that are observed by the network. This challenging research question can be addressed by designing adaptive data clustering and classification rules based on a set of noisy unlabeled sensor observations. In this paper, two robust and adaptive distributed hybrid classification algorithms are introduced. They consist of a local clustering phase that uses a small part of the data with a subsequent, fully distributed on-line classification phase. The classification is performed by means of distance-based similarity measures. In order to deal with the presence of outliers, the distances are estimated robustly. An extensive simulation-based performance analysis is provided for the proposed algorithms. The distributed hybrid classification approaches are compared to a benchmark algorithm where the error rates are evaluated in dependence of different WSN parameters. Communication cost and computation time are compared for all algorithms under test. Since both proposed approaches use robust estimators, they are, to a certain degree, insensitive to outliers. Furthermore, they are designed in a way that they are applicable to on-line classification problems.

Keywords: Adaptive distributed classification, Clustering, Labeling, Robust, Outlier, Multi device multi task (MDMT)

1 Introduction

Recent advances in distributed adaptive signal processing and communication networking are currently enabling novel paradigms for signal and parameter estimation. Based on the principles of adaptive filtering theory [1], a network of devices with node-specific interests adaptively optimizes its behavior, e.g., to jointly solve a decentralized least mean squares problem [2–6]. Under this new paradigm, multiple devices cooperate in multiple tasks (MDMT). This is different from the classical wireless sensor network setup, in which multiple devices perform one single joint task [2].

The MDMT paradigm can be beneficial, e.g., for speech enhancement in adverse environments [7]. Consider, for example, distributed audio signal enhancement in a public

area, such as an airport, a train-station, etc. By cooperating with each other, various devices (e.g., smart-phones, hearing aids, tablets) benefit in enhancing their node-specific audio source of interest, given a received mixture of interfering sound sources [2, 8], e.g., by suppressing noise and interfering sound sources that are not of interest to the user.

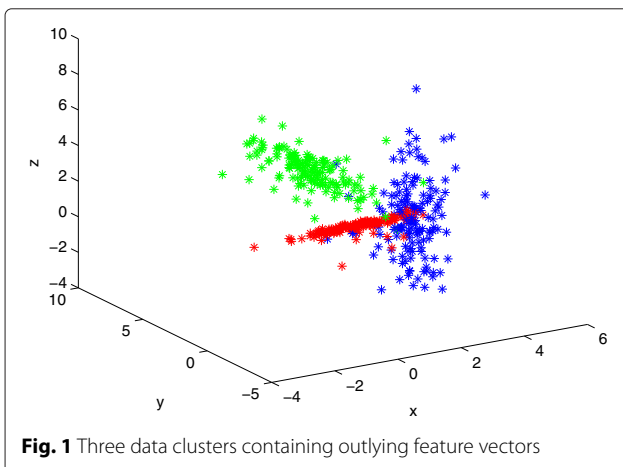
Note that in such scenarios, the devices must operate under stringent power and communication constraints and the transmission of observations to a fusion center (FC) is, in many cases, infeasible or undesired. A crucial first step in order to achieve a benefit, e.g., a better node-specific audio signal enhancement, is the common *unique labeling* of all relevant speech sources that are observed by the network [8]. Also in other MDMT signal-enhancement tasks, such as image enhancement, it is of practical importance to answer the question: who observes what? [9].

*Correspondence: binder@spg.tu-darmstadt.de
Signal Processing Group, Technische Universität Darmstadt, 64283 Darmstadt, Germany

This challenging research question can be tackled by designing adaptive data clustering and classification rules where each sensor collects a set of unlabeled observations that are drawn from a known number of classes. In particular, object or speaker labeling can be solved by in-network adaptive classification algorithms where a minimum amount of information is exchanged among single-hop neighbors. Various methods have been proposed that deal with distributed data clustering and classification, e.g., [8–24]. In the last few years, several distributed adaptive strategies, such as incremental, consensus, and diffusion least mean squares algorithms have been developed [25]. In [17], a distributed K-Means (DKM) algorithm that uses the consensus strategy was proposed.

In this paper, we provide an adaptive and robust hybrid diffusion-based approach which extends our previously published algorithm [21] by a robust distance measure that improves the classification/labeling performance, especially if the covariances of the clusters differ significantly. Robust methods become necessary whenever the distribution of the extracted features is heavy tailed or contains outliers [26, 27] due to errors in the feature estimation step. A scenario containing a high amount of outliers, as depicted in Fig. 1, complicates the classification considerably. In such a scenario, we propose to base the classification/labeling on robust adaptive centroid estimation and data clustering.

Contributions: Two robust in-network distributed classification algorithms, i.e., the RDiff K-Med and the CE RDiff K-Med, are proposed. It is shown that the performance of the first algorithm can be approached by the second algorithm with a considerably lower between-sensor communication cost. Unlike the DKM, which serves as a benchmark, the proposed algorithms are adaptive, instead



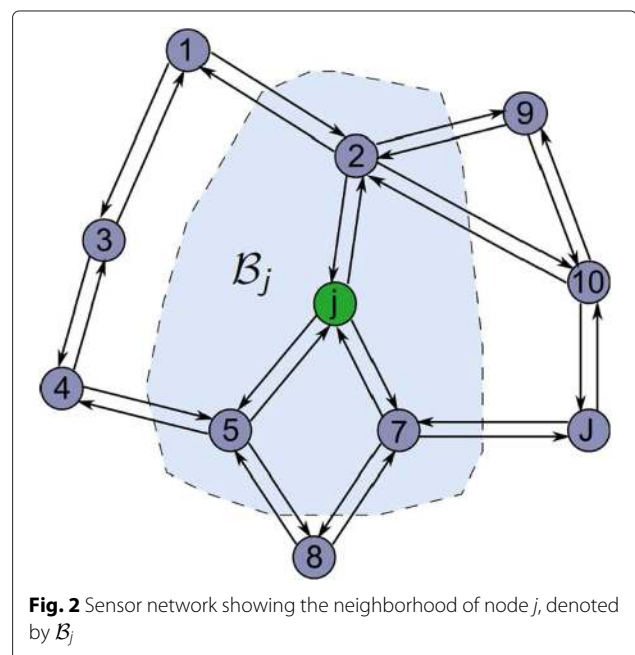
of working with a batch of data. They are thus applicable to real-time classification problems. Furthermore, they are robust against outliers in the feature vectors and can handle non-stationary features. An extensive simulation-based performance analysis is provided that investigates the error rates in dependence of different WSN parameters, and also considers communication cost and computation time.

Organization: Section 2 provides the problem formulation, Section 3 provides a brief introduction to the topic of robust estimation of class centroid and covariance. Section 4 is dedicated to the proposal and description of two robust diffusion-based classification algorithms, while Section 5 provides an extensive Monte-Carlo simulation study. Section 6 concludes the paper and provides future research directions.

2 Problem formulation and data model

Consider a network with J nodes distributed over some geographic region (see Fig. 2). Two nodes are connected if they are able to communicate directly with each other. The set of nodes connected to node $j \in 1, \dots, J$ is called the neighborhood of node j and is denoted by $\mathcal{B}_j \subseteq \mathcal{J}$. The communication links between the nodes are symmetric and a node is always connected to itself. The number of nodes connected to node j is called the degree of node j and is denoted by $|\mathcal{B}_j|$.

This paper is concerned with adaptive data clustering and classification/labeling when each sensor collects



a set of unlabeled observations that are drawn from a known number of classes. This task should be accomplished in a decentralized manner by communicating only within directly connected neighborhoods \mathcal{B}_j , instead of transmitting all observations to a master node or FC. Each observation is assumed to belong to a certain class \mathcal{C}_k with $k \in 1, \dots, K$ with k denoting the label of the given class. The total number of classes K is assumed to be known, or estimated a priori. Each class is described by a number of application dependent descriptive statistics (features). The feature estimation process is an application-specific research area of its own (see, e.g., [8, 9]) and is not considered in this article, where we seek for generic adaptive robust clustering and classification methods. In the following, it is assumed that the feature extraction has already been performed, so that the uncertainty of the feature estimation within each class can be modeled by a probability distribution, e.g., the Gaussian. Further, we account for estimation errors in the feature extraction process that we consider as outliers, thus arriving at the following observation model for feature vectors at time instant n , $n = 1, \dots, N$:

$$\mathbf{d}_{jkn} = \mathbf{w}_k + \mathbf{e}_{jkn} + \mathbf{o}_{jn}. \quad (1)$$

Here, \mathbf{w}_k denotes the class centroid, \mathbf{e}_{jkn} represents the class-specific uncertainty term with covariance matrix Σ_{jk} , \mathbf{o}_{jn} denotes the outlier term which models disturbances of an unspecified density and $\mathbf{d}_{jkn}, \mathbf{w}_k, \mathbf{e}_{jkn} \in \mathbb{R}^q$. \mathbf{e}_{jkn} is assumed to be temporally and spatially independent, i.e.,

$$\mathbb{E}\{\mathbf{e}_{jn}^* \mathbf{e}_{lm}\} = \sigma_{e_j}^2 \cdot \delta_{jl} \cdot \delta_{nm} \quad (2)$$

with $j, l = 1, \dots, J$, $n, m = 1, \dots, N$ and δ denoting the Kronecker delta function. \mathbf{e}_{jkn} is assumed to be zero mean. For reasons of clarity, we drop the index k in the observation vectors and refer to them as \mathbf{d}_{jn} .

The aim of this paper is thus to enable every node j to assign each observation to a cluster k based on an estimated feature \mathbf{d}_{jn} . The classification/labeling should be real-time capable so that a new observation can be assigned on-line without the necessity of all recorded observations being available. Furthermore, outliers in Eq. (1) should not have a huge effect on the labeling performance. This will be achieved by using robust techniques to estimate the class centroids and covariances, as well as robust distance measures, as described in the next section.

3 Robust estimation of class centroid and covariance

The presence of even a small amount of outliers in a data set can have a high impact on classical estimators like the sample mean vector and sample covariance matrix. Though these estimators are optimal under the Gaussian noise assumption, they are extremely sensitive to uncharacteristic observations in the data [26]. For this purpose, robust estimators have been developed which are, to a certain degree, resistant towards outliers in the data.

In the following, a short overview of the concept of M-estimation for the multivariate case is presented, as required by our methods. For a more detailed treatment of the fundamental concepts, see, e.g., [26, 28].

The hybrid classification approach developed in this paper involves estimating the mean and covariance for vector-valued data $\mathbf{d}_{jn} = (d_{1jn}, d_{2jn}, \dots, d_{qjn})^T$ with $\mathbf{d}_{jn} \in \mathbb{R}^q$, where q is the dimension of the feature space.

In the univariate case, it is possible to define the robust estimates of location and dispersion separately. In the multivariate case, in order to obtain equivariant estimates, it is of advantage to estimate location and dispersion simultaneously [28].

The multivariate Gaussian density is

$$f_{\mathbf{D}}(\mathbf{d}; \mathbf{w}, \Sigma) = \frac{1}{\sqrt{|\Sigma|}} h_{\mathbf{D}}(g_{\mathbf{D}}(\mathbf{d}; \mathbf{w}, \Sigma)) \quad (3)$$

where $|\Sigma|$ denotes the determinant of Σ , $h_X(x) = c \exp(-x/2)$ with $c = (2\pi)^{-q/2}$ and $g_{\mathbf{D}}(\mathbf{d}; \mathbf{w}, \Sigma) = (\mathbf{d} - \mathbf{w})^T \Sigma^{-1} (\mathbf{d} - \mathbf{w})$.

Let $\mathbf{d}_{j1}, \dots, \mathbf{d}_{jN}$ be an i.i.d. sample from a density of the form (3). M-estimates of the cluster centroids and covariance matrices are defined as solutions of the general system equations

$$\sum_{n=1}^N \phi_1(g_{\mathbf{D}}(n)) (\mathbf{d}_{jn} - \hat{\mathbf{w}}_k) = \mathbf{0}_q \quad (4)$$

$$\frac{1}{N-1} \sum_{n=1}^N \phi_2(g_{\mathbf{D}}(n)) (\mathbf{d}_{jn} - \hat{\mathbf{w}}_k) (\mathbf{d}_{jn} - \hat{\mathbf{w}}_k)^T = \hat{\Sigma}_k, \quad (5)$$

where the functions ϕ_1 and ϕ_2 may be chosen differently. Uniqueness of solutions of (4) and (5) requires that $g_{\mathbf{D}} \phi_2(g_{\mathbf{D}})$ is a nondecreasing function of $g_{\mathbf{D}}$ [28].

A common choice are *Huber's functions* [29] with

$$\rho(\mathbf{d}) = \begin{cases} \mathbf{d}_{jn}^2 & , \text{ if } |\mathbf{d}_{jn}| \leq \mathbf{c}_{\text{hub}} \\ 2\mathbf{c}_{\text{hub}} |\mathbf{d}_{jn}| - \mathbf{c}_{\text{hub}}^2 & , \text{ if } |\mathbf{d}_{jn}| > \mathbf{c}_{\text{hub}} \end{cases} \quad (6)$$

and

$$\phi_{1,2}(\mathbf{d}) = \frac{\partial \rho(\mathbf{d})}{\partial \mathbf{d}} \quad (7)$$

with c_{hub} denoting the Huber's tuning constant. The function $\rho(\mathbf{d})$ from Eq. (6) shows quadratic behavior in the central region while increasing linearly to infinity. Outliers are therefore assigned less weight than data close to the model. Note that all maximum likelihood estimators are also M-estimators.

4 Proposed methods

In this section, two new robust in-network distributed classification algorithms are presented that extend our previously published algorithm [21] by a robust distance measure that improves the classification/labeling performance, especially if the covariances of the clusters differ significantly.

Since we have no training data available for the classification process, the general idea of the methods is to split the classification/labeling procedure into two main steps: in a *local clustering phase* each node calculates a preliminary estimate of the cluster characteristics (i.e., centroids and covariances) of each cluster using a small number of feature vectors. These preliminary estimates serve as an initialization for the subsequent *global classification phase*. Here, based on these estimates, a new feature is classified using a robust distance measure. The aim is to improve the local classification result by a combination of local processing and communication between the agents.

An advantage of this procedure is that this hybrid approach turns into a mere classification algorithm when the cluster characteristics are known beforehand. In this case, the local clustering phase is not needed.

The methods are based on the diffusion LMS strategy that was introduced in [30]. In this way, the classification is adaptive and can handle streaming data coming from a distributed sensor network. Since the communication cost between the nodes should be kept as low as possible, the second approach is designed with reduced in-network communication. A robust design makes sure that the proposed algorithms are, to a certain degree, resistant towards outliers in the feature vectors. In the following, the two proposed approaches are described in detail.

4.1 Robust distance-based K-medians clustering/classification over adaptive diffusion networks (RDiff K-Med)

The first proposed hybrid classification methodology is the "Robust Distance-Based Clustering/Classification

Algorithm over Adaptive Diffusion Networks" (RDiff K-Med). It begins with a local initialization phase where each node j collects a number of N_t observations and performs K-medians clustering on these observations. In this way, each node locally partitions its first N_t observations $\mathbf{D}_{jn} = \{\mathbf{d}_{jn}, n = 1, \dots, N_t\}$ into k sets C_k so that the ℓ_1 -distance within each cluster is minimized:

$$\arg \min_{\mathbf{w}_k} \sum_{k=1}^K \sum_{n=1}^{N_t} \|\mathbf{d}_{jn} - \mathbf{w}_k\|_1 \quad (8)$$

Each center is the component-wise median of the points of each cluster. The features assigned to each class C_k are stored in an initial feature matrix \mathbf{S}_{jk}^0 . Based on all elements in \mathbf{S}_{jk}^0 , local intermediate estimates of the cluster centroid $\hat{\boldsymbol{\psi}}_{jk}^0$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_{jk}^0$ are determined. In the following, the calculation steps are presented in detail.

First, as robust local initial estimate of the cluster center, compute the column-wise median of \mathbf{S}_{jk}^0

$$\hat{\boldsymbol{\psi}}_{jk}^0 = \text{median}(\mathbf{S}_{jk}^0). \quad (9)$$

$\hat{\boldsymbol{\psi}}_{jk}^0$ is thus obtained by computing the median separately for each spatial direction of all elements in \mathbf{S}_{jk}^0 .

Next, proceed by computing a robust local initial estimate of the cluster covariances. In this paper, we compare three estimators, i.e. the sample covariance, Huber's M-estimator and a computationally simple robust covariance estimator based on the median absolute deviation (MAD).

The sample covariance matrix estimate is given by

$$\hat{\boldsymbol{\Sigma}}_{jk}^0 = \frac{1}{N_t - 1} \sum_{n=1}^{N_t} (\mathbf{d}_{jn} - \hat{\boldsymbol{\psi}}_{jk}^0) (\mathbf{d}_{jn} - \hat{\boldsymbol{\psi}}_{jk}^0)^H. \quad (10)$$

Huber's M-estimator, as defined in Eq. (6), is computed via an iteratively reweighted least-squares algorithm, as detailed in [28] with the previously computed $\hat{\boldsymbol{\psi}}_{jk}^0$ as location estimate.

In case of the MAD based covariance estimate, for each feature \mathbf{d}_{jn} in \mathbf{S}_{jk}^0 the difference vector

$$\mathbf{d}_{\text{diff},jk} = |\mathbf{d}_{jn} - \text{median}(\mathbf{S}_{jk}^0)| \quad (11)$$

is calculated and stored in the matrix $\mathbf{S}_{\text{diff},jk}^0$. Based on the elements in $\mathbf{S}_{\text{diff},jk}^0$, the MAD is given by

$$\hat{\sigma}_{jk}^0 = 1.483 \cdot \text{median}(\mathbf{S}_{\text{diff},jk}^0) \quad (12)$$

and the corresponding covariance matrix is

$$\hat{\Sigma}_{jk}^0(r, s) = \begin{cases} (\hat{\sigma}_{jks}^0)^2, & \forall r = s = 1, \dots, q \\ 0, & \forall r \neq s \end{cases} \quad (13)$$

with $\hat{\sigma}_{jks}^0$ denoting the standard deviation estimate in each spatial direction of the feature space. Note that the covariance matrix calculated in Eq. (13) is a diagonal matrix. This computationally simple robust estimator is only applicable when the entries of the feature vectors are assumed to be independent of each other. The estimates of the sample covariance matrix and the M-estimator do not require this assumption and are, in general, not diagonal matrices.

Since the order in which the cluster centroids are stored by K-Medians is random, it may differ between two nodes. Thus, it has to be assured that the data which is exchanged by the nodes refers to the same classes. This is achieved by a unique initial ordering of the class centroids and covariance matrices among all nodes in the network: starting with the class centroids and covariance matrices stored for the first class of a preset reference node, all other nodes calculate the Euclidean distance of the respective entries corresponding to all stored classes and those of the first class of the reference node. The data with the smallest Euclidean distance to the reference entries are re-stored at the position corresponding to the first class. This procedure is repeated for all classes stored by the nodes in the network.

Having obtained a consistent data structure, each node j exchanges its own feature vectors S_{jk}^0 for each class C_k with its neighbors $i \in \mathcal{B}_j$. All nodes store their own as well as the features received from their neighbors in an initial matrix V_{jk}^0 . In the following clustering/classification

procedure, S_{jk}^0 and V_{jk}^0 are extended to S_{jkn} and V_{jkn} in every time step n by adding columns containing the new feature vectors received at time step n .

This completes the initialization phase, which is followed by the exchange phase, where each new observation \mathbf{d}_{jn} , $n = N_t + 1, \dots, N$, is classified according to the following diffusion-procedure:

1. **Exchange Step:** If there are new, unshared feature vectors, each node j adds them to V_{jkn} and broadcasts them to its neighbors $i \in \mathcal{B}_j$.
2. **Adaptation Step:** Each node j determines preliminary local estimates $\hat{\psi}_{jkn}$ and $\hat{\Sigma}_{jkn}^*$ at time n based on the feature vectors stored in V_{jkn} analogously to (13) with V_{jkn} replacing S_{jkn} . In order to be capable of dealing with non-stationary time-varying signals, a window length l_w is introduced which limits the size of V_{jkn} by only retaining the latest l_w elements which were added to V_{jkn} .
3. **Exchange Step:** Each node exchanges its intermediate estimates $\hat{\psi}_{jkn}$ and $\hat{\Sigma}_{jkn}^*$ with its neighbors.
4. **Combination Step:** Each node j adapts its estimates according to

$$\hat{w}_{jkn} = \alpha \cdot \hat{\psi}_{jkn} + (1 - \alpha) \cdot \sum_{b \in \mathcal{B}_j / \{j\}} a_{bkn} \cdot \hat{\psi}_{bkn} \quad (14)$$

and

$$\hat{\Sigma}_{jkn} = \alpha \cdot \hat{\Sigma}_{jkn}^* + (1 - \alpha) \cdot \sum_{b \in \mathcal{B}_j / \{j\}} a_{bkn} \cdot \hat{\Sigma}_{bkn}^* \quad (15)$$

with α denoting an adaptation factor which determines the weight which is given to the own estimate and the

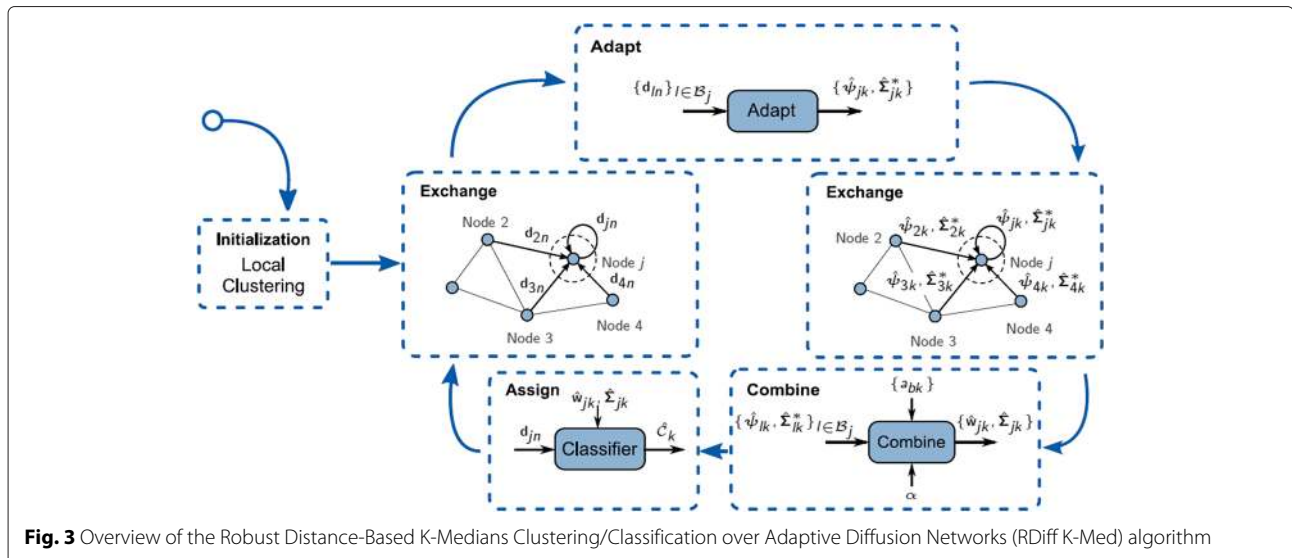


Fig. 3 Overview of the Robust Distance-Based K-Medians Clustering/Classification over Adaptive Diffusion Networks (RDiff K-Med) algorithm

neighborhood estimates, respectively, and a_{bkn} being a weighting factor chosen as

$$a_{bkn} = 1 / \left[\|\hat{\psi}_{bk} - \text{median}(\mathbf{V}_{jkn})\|^2 \right] \quad (16)$$

with subsequent normalization such that $\sum_{b \in \mathcal{B}_j \setminus \{j\}} a_{bkn} = 1$.

5. Classification Step: In the next step, feature vector \mathbf{d}_{jn} is classified by evaluating its distance to each of the estimated class centroids $\hat{\mathbf{w}}_{jk}$. The considered distance measures are the Euclidean distance and the Mahalanobis distance given by

$$d_{\text{Eucl}}(\mathbf{d}_{jn}, \hat{\mathbf{w}}_{jk}) = \sqrt{(\mathbf{d}_{jn} - \hat{\mathbf{w}}_{jk})^T (\mathbf{d}_{jn} - \hat{\mathbf{w}}_{jk})} \quad (17)$$

and

$$d_{\text{Mahal}}(\mathbf{d}_{jn}, \hat{\mathbf{w}}_{jk}) = \sqrt{(\mathbf{d}_{jn} - \hat{\mathbf{w}}_{jk})^T \hat{\Sigma}_{jk}^{-1} (\mathbf{d}_{jn} - \hat{\mathbf{w}}_{jk})}. \quad (18)$$

\mathbf{d}_{jn} is assigned to the class \mathcal{C}_k for which the respective distance is minimized.

With *Step 1*, the processing chain then starts at the beginning where the previously classified feature vectors are broadcasted to the neighborhood.

An overview of the *RDiff K-Med* algorithm is depicted in Fig. 3, a summary is provided in Table 1.

4.2 Communicationally Efficient Robust Distance-Based K-Medians Clustering/Classification over Adaptive Diffusion Networks (CE RDiff K-Med)

Since the *RDiff K-Med* may be demanding in terms of communication between sensors, which is a major contributor to the energy consumption of the devices [31], an algorithm is proposed which yields similar performance with reduced in-network communication: the ‘‘Communicationally Efficient Robust Distance-Based K-Medians Clustering/Classification over Adaptive Diffusion Networks’’ (CE RDiff K-Med).

The general procedure is similar to the RDiff K-Med except that there is no exchange of feature vectors between the nodes. The steps of the *CE RDiff K-Med* are the following:

1. Adaptation Step: Based on the feature vectors \mathbf{d}_{jn} stored in \mathcal{S}_{jkn} , each node calculates its intermediate estimates $\hat{\psi}_{jkn}$ and $\hat{\Sigma}_{jkn}^*$ according to (9)–(13).

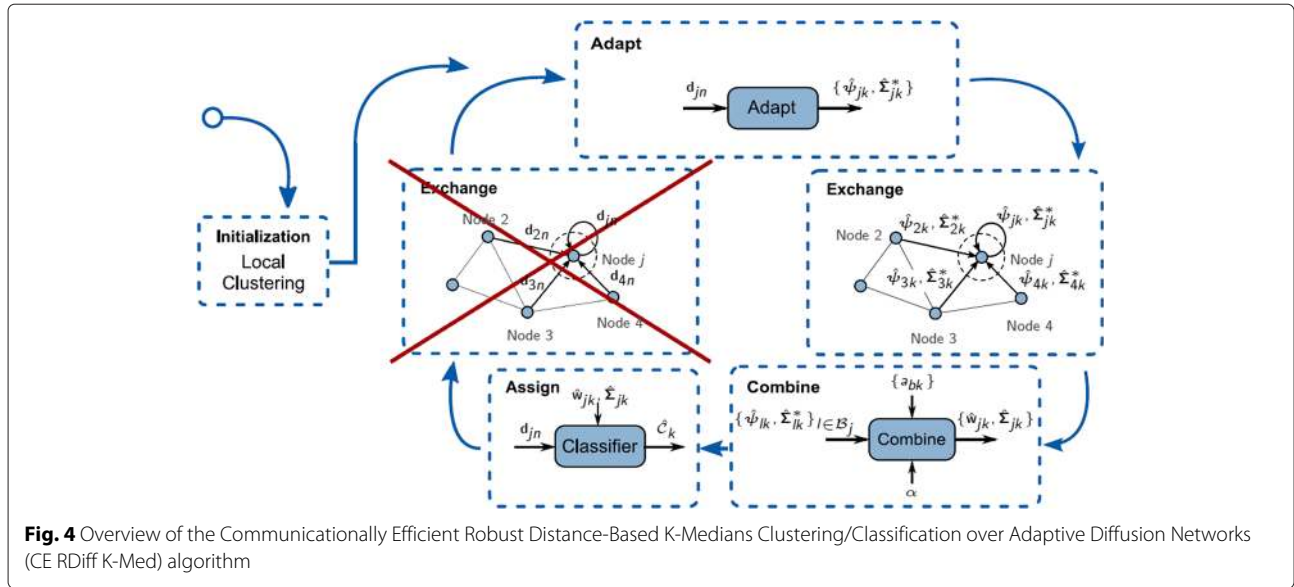
2. Exchange Step: Instead of broadcasting the entire feature vectors, the nodes share only their estimates of the cluster centers $\hat{\psi}_{jkn}$ and the respective covariance matrices $\hat{\Sigma}_{jkn}^*$ with their neighbors.

Table 1 Summary of the RDiff K-Med algorithm

Algorithm: RDiff K-Med	
<i>Local Clustering Phase</i>	
1.	for the first N_t feature vectors do
2.	for all $j = 1, \dots, J$ do
3.	perform K-medians according to (8)
4.	calculate $\hat{\mathbf{w}}_{jk}^0$ and $\hat{\Sigma}_{jk}^0$ via (9)–(13)
5.	store classified data in \mathcal{S}_{jk}^0
6.	end for
7.	for all $j = 1, \dots, J$ do
8.	perform synchronization of cluster estimates
9.	end for
10.	for all $j = 1, \dots, J$ do
11.	exchange \mathcal{S}_{jk}^0 with all neighbors $i \in \mathcal{B}_j$
12.	store received data in \mathcal{V}_{jk}^0
13.	end for
14.	end for
<i>Distributed Classification Phase</i>	
15.	for $n = N_t, N_t + 1, \dots, N$ do
16.	for all $j = 1, \dots, J$ do
17.	broadcast an update for \mathbf{V}_{jkn} to all neighbors $i \in \mathcal{B}_j$
18.	end for
19.	for all $j = 1, \dots, J$ do
20.	determine $\hat{\psi}_{jkn}$ and $\hat{\Sigma}_{jkn}^*$ via (9)–(13)
21.	end for
22.	for all $j = 1, \dots, J$ do
23.	broadcast $\hat{\psi}_{jkn}$ and $\hat{\Sigma}_{jkn}^*$ to all neighbors $i \in \mathcal{B}_j$
24.	end for
25.	for all $j = 1, \dots, J$ do
26.	determine $\hat{\mathbf{w}}_{jk}(n)$ and $\hat{\Sigma}_{jk}(n)$ via (14) & (15)
27.	calculate distances from feature vector \mathbf{d}_{jn} to all $\hat{\mathbf{w}}_{jk}(n)$ by evaluating (17) or (18)
28.	assign \mathbf{d}_{jn} to class \mathcal{C}_k which minimizes (17) or (18)
29.	end for
30.	end for

3. Combine Step: Each sensor j combines its neighbor’s estimates analogously to (14) and (15) in order to obtain improved estimates $\hat{\mathbf{w}}_{jkn}$ and $\hat{\Sigma}_{jkn}$.

4. Classification Step: Based on the estimates determined in the previous step, the distance measure of the feature vector to the estimates of the class centroids is evaluated and \mathbf{d}_{jn} is classified analogously to the RDiff K-Med. Subsequently, \mathbf{d}_{jn} is added to \mathcal{S}_{jkn} .



An overview of the *CE RDiff K-Med* algorithm is provided in Fig. 4, a summary is given in Table 2.

5 Numerical experiments

This section evaluates the performance of the proposed algorithm numerically in terms of the error rate in a broad range of conditions, i.e., different distributions of the outliers, different percentages of outliers in the feature vectors, different dimensions of the input data, different numbers of clusters and in terms of the adaptation speed in case of non-stationary data. Furthermore, the communication cost for different neighborhood sizes and the computation time as a function of the data dimension is considered. When reasonable, we compare our proposed method to the DKM [17].

5.1 Benchmark: distributed K-means (DKM)

As a benchmark, this paper considers the *Distributed K-Means* (DKM) algorithm by Forero et al., for details, see [17]. The basic idea of the DKM is to cluster the observations into a given number of groups, such that the sum of squared-errors is minimized, that is

$$\arg \min_{\mathbf{w}_k, \mu_{jnk}^p} \frac{1}{2} \sum_{j=1}^J \sum_{k=1}^K \sum_{n=1}^{N_j} \mu_{jnk}^p \|\mathbf{d}_{jn} - \mathbf{w}_k\|^2, \quad (19)$$

where \mathbf{w}_k is the cluster center for class k , $\mu_{jnk} \in [0, 1]$ is the membership coefficient of \mathbf{d}_{jn} to class k , and $p \in [1, +\infty]$ is a tuning parameter. The DKM iteratively solves the surrogate augmented Lagrangian of a distributed clustering problem based on (19) while exchanging the resulting parameters among neighboring nodes.

Although the DKM achieves very good performance in many scenarios, a major drawback is that the clustering is

Table 2 Summary of the CE RDiff K-Med

Algorithm: CE RDiff K-Med	
<i>Local Clustering Phase</i>	
1.	for the first N_t feature vectors do
2.	for all $j = 1, \dots, J$ do
3.	perform K-medians according to (8)
4.	calculate w_{jk}^0 and $\hat{\Sigma}_{jk}^0$ via (9)-(13)
5.	store classified data in S_{jk}^0
6.	end for
7.	for all $j = 1, \dots, J$ do
8.	perform synchronization of cluster estimates
9.	end for
10.	end for
<i>Distributed Classification Phase</i>	
11.	for $n = N_t, N_t + 1, \dots, N$ do
12.	for all $j = 1, \dots, J$ do
13.	calculate intermediate estimates $\hat{\psi}_{jkn}$ and $\hat{\Sigma}_{jkn}^*$ according to (9)-(13)
14.	broadcast $\hat{\psi}_{jkn}$ and $\hat{\Sigma}_{jkn}^*$ to all neighbors $i \in \mathcal{B}_j$
15.	end for
16.	for all $j = 1, \dots, J$ do
17.	determine $\hat{\mathbf{w}}_{jkn}$ and $\hat{\Sigma}_{jkn}$ via (14)
18.	calculate distances from feature vector \mathbf{d}_{jn} to all $\hat{\mathbf{w}}_{jkn}$ by evaluating (17) or (18)
19.	assign \mathbf{d}_{jn} to class \mathcal{C}_k which minimizes (17) or (18)
20.	end for
21.	end for
22.	end for

performed based on all available data and that it may need a high number of iterations until it converges to its final solution. This property makes the DKM difficult to use in real-time applications where an observation needs to be classified based on streaming data, such as for example in speaker labeling for MDMT speech enhancement [2] or object labeling in MDMT video enhancement for camera networks [9]. In addition to that, the performance of the DKM is limited in scenarios where feature vectors contain outliers.

5.2 Simulation setup

The simulations are based on a scenario with $J = 10$ nodes which are randomly distributed in space. Each node is connected to the four neighboring nodes which have the smallest Euclidean distance. Unless mentioned otherwise, classification is performed on $K = 3$ classes with centers $\mathbf{w}_1 = (1, 1, 1)^T$, $\mathbf{w}_2 = (1, 4, 3)^T$, $\mathbf{w}_3 = (3, 1, 1)^T$. Each sample \mathbf{d}_{jn} is drawn at random from class k from the density $\mathcal{N}(\mathbf{d}_{jn}; \mathbf{w}_k, \Sigma_k)$ with covariance matrices $\Sigma_1 = (1, 0.01, 0.01)^T \mathbf{I}_3$, $\Sigma_2 = (0.16, 4, 0.16)^T \mathbf{I}_3$ and $\Sigma_3 = (0.25, 0.01, 4)^T \mathbf{I}_3$. Each node has $N_j = 80$ samples available, 20 for the initialization and 60 for real-time classification. K-Medians is run three times, and the result which minimizes (8) is used for the classification. The parameters for the benchmark algorithm DKM are set $p = v = 2$, where $p = 2$ enables soft clustering and $v = 2$ is the tuning parameter which yields the best results in the performance tests in [17]. The result is obtained having all $N_j = 80$ samples per node available. Since the performance of the DKM depends on the number of iterations, we provide simulation results for multiple choices of the amount of iterations.

The generation of outliers considers a certain percentage of samples to be replaced by a new sample which

is drawn from a contaminating distribution (Gaussian or chi-square). The error rate is calculated based on the classified samples excluding any outliers. The displayed results represent the averages that are based on 100 Monte-Carlo runs.

5.3 Simulation results

In Fig. 5, the impact of the dimension of the feature vectors on the performance is depicted. The data is generated by concatenating the mean values and covariance matrices until they have the according dimension. For example, $\mathbf{w}_3 = (3, 1, 1)^T$ is changed to $\mathbf{w}_3 = (3, 1, 1, 3, 1, 1)^T$ and $\Sigma_3 = (0.25, 0.01, 4)^T \mathbf{I}_3$ becomes $\Sigma_3 = (0.25, 0.01, 4, 0.25, 0.01, 4)^T \mathbf{I}_6$ in order to obtain data of dimension $q = 6$ and so on. For increasing data dimension, the error rates for all considered algorithms decreases continuously.

Figure 6 depicts the error rate of the algorithms under test as a function of the percentage of outliers in the data, where 0% corresponds to the outlier free case. Here, the outliers are drawn at random from a Gaussian distribution with the density $\mathcal{N}((10, 10, 10)^T, \mathbf{I}_3)$. The simulation is run with the different estimators of covariance introduced in Section 3, the location is estimated using the median. The robust distance measures result in smaller error rates than the Euclidean distance.

Since in real-world scenarios, the outliers usually do not follow any specific distribution, the question arises how the algorithms deal with other types of outliers, e.g., from a skewed heavy tailed distribution. For the evaluation of this scenario, the outliers are now generated by a chi-square distribution with different degrees of freedom ν for each class: to a certain percentage of the feature vectors a vector is added which is drawn at random from a chi-square distribution where for each class, different values

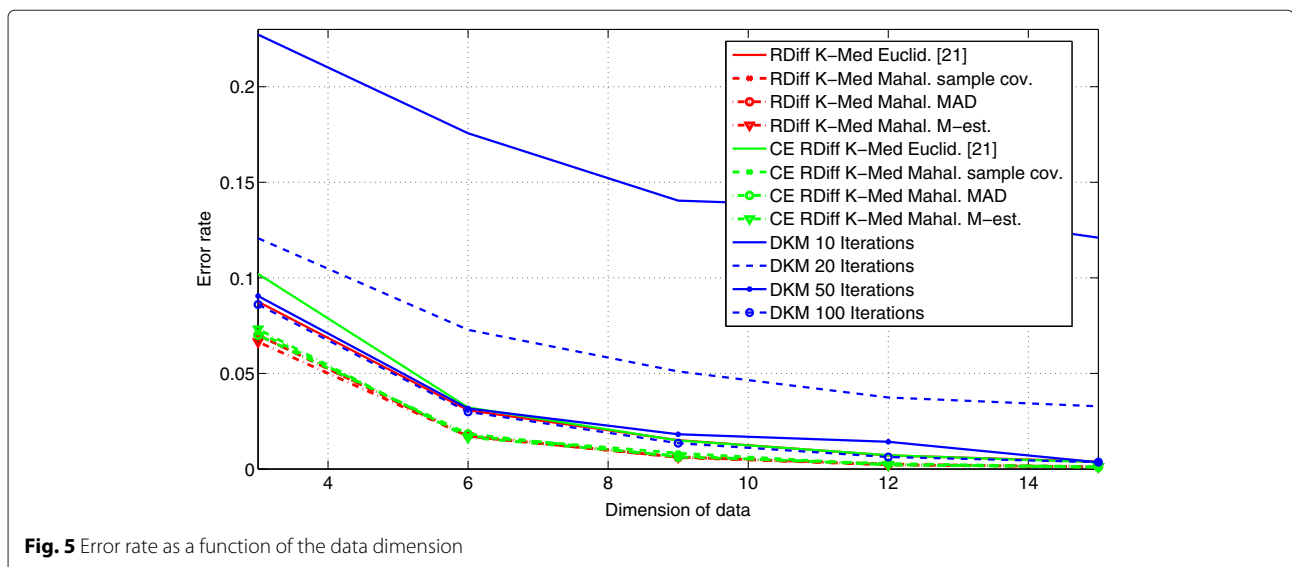
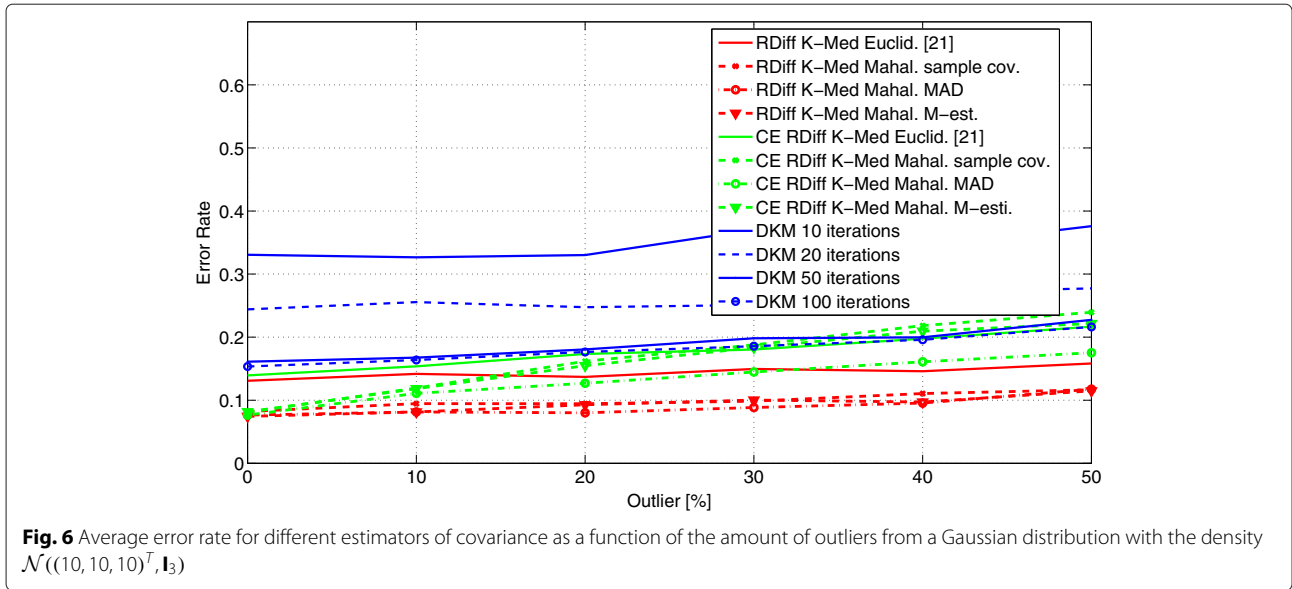


Fig. 5 Error rate as a function of the data dimension

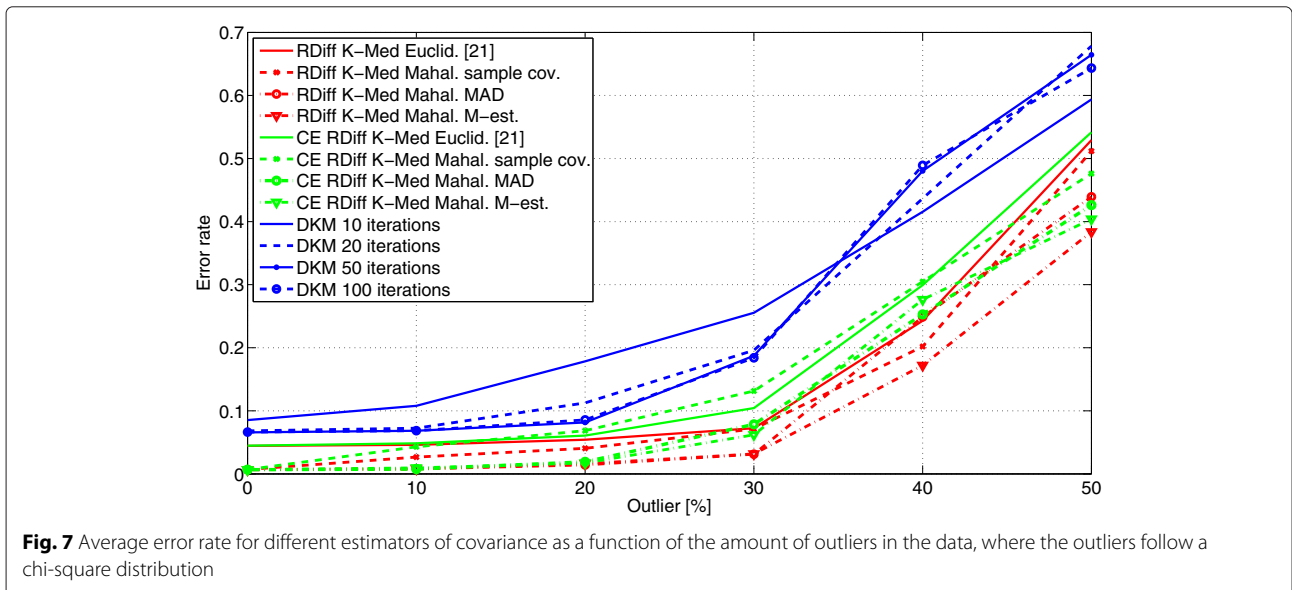


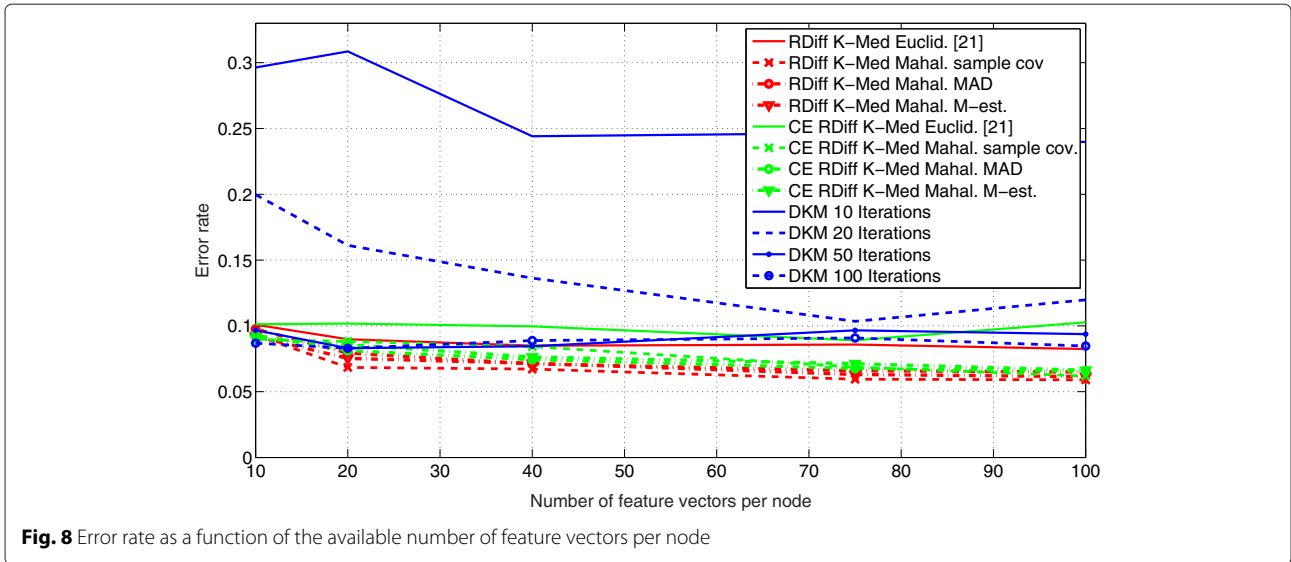
for ν are chosen. This is done in order to create a non-symmetric outlier distribution instead of a constant shift of the mean of the outlier distribution for all classes. In this manner, for the first class \mathcal{C}_1 , a randomly drawn vector of dimension q with $\nu_1 = 3$ is added to a certain number of data vectors, a vector with $\nu_2 = 5$ is subtracted from corresponding feature vectors of class \mathcal{C}_2 and for \mathcal{C}_3 a different random number is drawn for each direction in space: generated with $\nu_{3,1} = 4, \nu_{3,2} = 1$ and $\nu_{3,3} = 7$ for x, y and z direction, respectively, whereby $\nu_{3,2} = 1$ is subtracted from the y -component. For this simulation, a scenario is chosen with more distinct clusters with centroids $\mathbf{w}_1 =$

$(1, 1, 1)^T, \mathbf{w}_2 = (0, 5, 3)^T, \mathbf{w}_3 = (3, 3, 7)^T$. The result is given in Fig. 7.

Figure 8 shows the error performance as a function of the number of feature vectors which are available per node. Both the DKM for $i = 10$ and $i = 20$ and the RDiff K-Med and CE RDiff K-Med with robust estimation methods show a slightly decreasing error rate with a growing number of feature vectors.

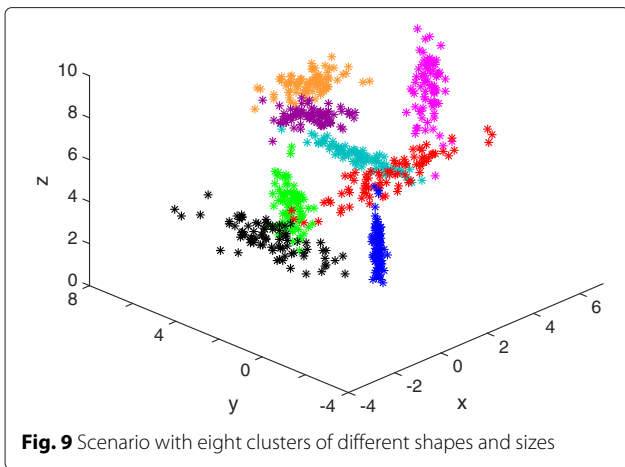
For the next experiment, we evaluate a more complex scenario consisting of eight clusters of different shapes and sizes distributed in space (see Fig. 9). The centroids are chosen as $\mathbf{w}_1 = (1, 0, 3)^T, \mathbf{w}_2 = (1, 4, 3)^T,$





$\mathbf{w}_3 = (1, 0, 6)^T$, $\mathbf{w}_4 = (-1, 3, 3)^T$, $\mathbf{w}_5 = (4, 4, 4)^T$, $\mathbf{w}_6 = (6, 3, 7)^T$, $\mathbf{w}_7 = (4.5, 7, 6)^T$ and $\mathbf{w}_8 = (2, 4, 7)^T$ with corresponding covariance matrices $\Sigma_1 = (0.1, 0.1, 1)^T \mathbf{I}_3$, $\Sigma_2 = (0.1, 0.4, 1)^T \mathbf{I}_3$, $\Sigma_3 = (2, 0.1, 0.5)^T \mathbf{I}_3$, $\Sigma_4 = (0.4, 1.6, 0.4)^T \mathbf{I}_3$, $\Sigma_5 = (0.2, 1.2, 0.1)^T \mathbf{I}_3$, $\Sigma_6 = (0.25, 0.3, 1.5)^T \mathbf{I}_3$, $\Sigma_7 = (0.8, 0.5, 0.2)^T \mathbf{I}_3$ and $\Sigma_8 = (0.5, 0.5, 0.3)^T \mathbf{I}_3$. The outliers are drawn randomly from a Gaussian distribution with the density $\mathcal{N}((10, 10, 10)^T, \mathbf{I}_3)$. The results are provided in Fig. 10.

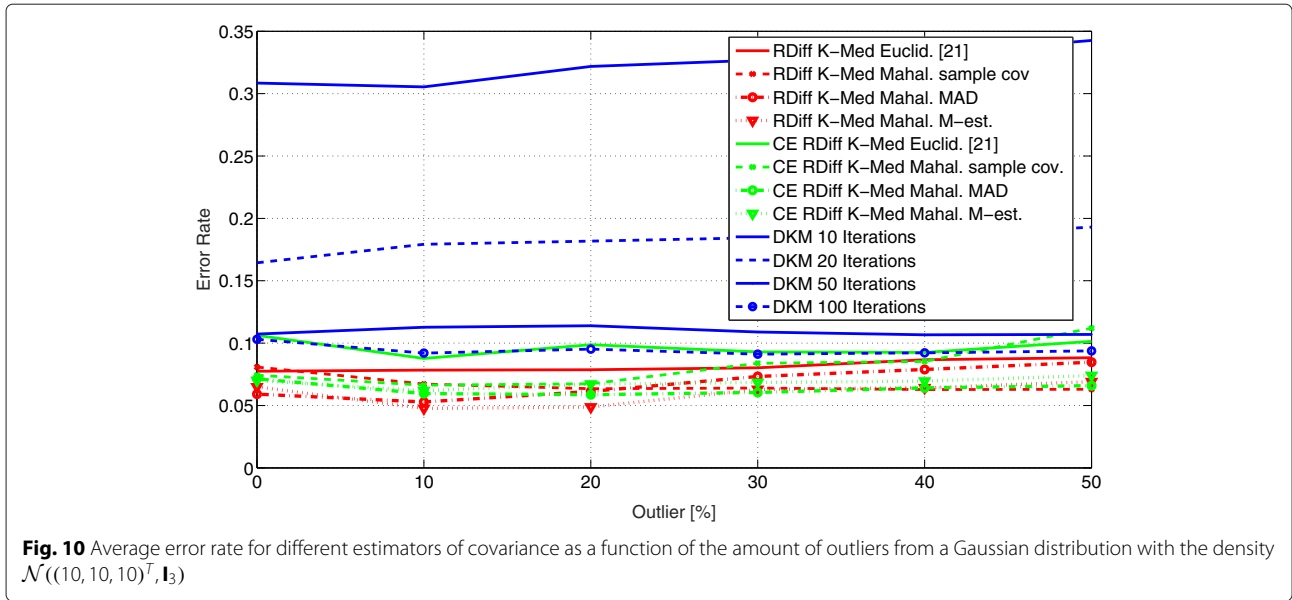
The former performance studies were based on the assumption that the data is stationary. Next, it is examined how the proposed algorithms perform for non-stationary feature vectors. For this purpose the value of a single considered cluster centroid is instantly changed during the classification process. The adaptation speed of the RDiff K-Med and the CE RDiff K-Med is examined for different window sizes l_w and different values for α (see Eqs. (14) and (15)) by calculating the error which is given by the



norm of the difference between the true value and the estimate of the cluster centroid. Unlike the CE RDiff K-Med the RDiff K-Med stores not only its own feature vectors, but also the feature vectors from its neighborhood, it has $(|\mathcal{B}_j| + 1)$ data vectors per time step available instead of only one. In order to make the window sizes for both algorithms comparable, l_w is chosen such that it contains the feature vectors of l_w time steps. As a consequence, the compared window length of the RDiff K-Med corresponds to $(|\mathcal{B}_j| + 1)$ times the window length of the CE RDiff K-Med. The result is shown in Fig. 11. As depicted in the upper plot, a large window size results in a slower adaptation speed. The RDiff K-Med adapts faster to the true cluster centroid than the CE RDiff K-Med since its estimation is based on more available samples. However, the CE RDiff K-Med yields a smaller error compared to the RDiff K-Med when both have adapted to the true value. The choice of the factor α (see lower plot) has no significant impact on the RDiff K-Med. For the CE RDiff K-Med a smaller value for α (and therefore a higher weighting of the estimates of the neighboring nodes) leads to a higher adaptation speed. Since it has only a small amount of feature vectors available, this method is dependent on the data exchange with its neighbors.

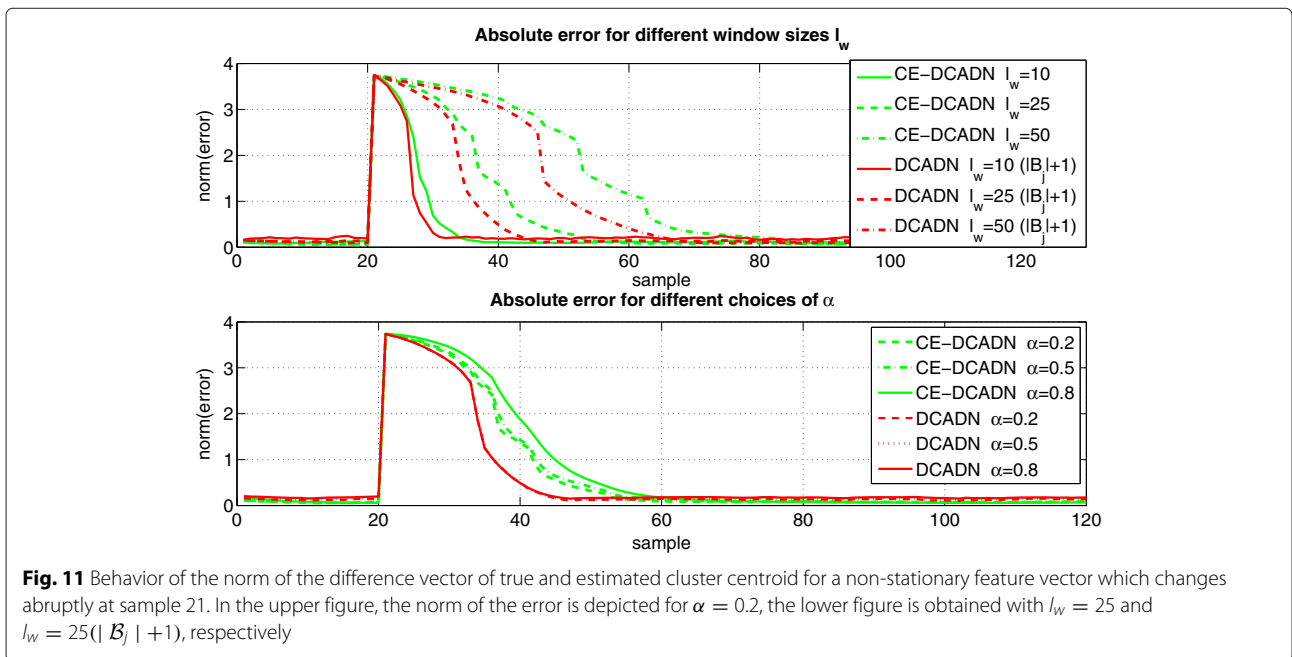
5.4 Communication cost and computation time

Apart from the error rate, further performance measures of great importance are the communication cost as well as computation time. Since the communication costs contribute stronger to the energy consumption in the wireless devices than the computational costs [31], the former should be kept as low as possible. Figure 12 depicts the communication costs for the standard scenario in dependence of various neighborhood cardinalities of



each node. The communication cost displayed in Fig. 12 is specified in data units, where one matrix entry forms one unit. It becomes clear that the choice of the neighborhood size has a high impact on the communication costs. For the DKM the number of iterations is crucial. While for a small amount of clusters few iterations may be sufficient, the number of iterations that is necessary for a good performance increases for higher cluster numbers (see [17] for more detailed information) which results in strongly increasing communication costs.

The computation time as a function of the dimension of the data is provided by Fig. 13 and given in seconds (using an Intel Core i7 5820K). Whereas the DKM has a constant computation time independent of the data dimension, it increases with the data dimension for the proposed algorithms. The resulting computation time for using the M-estimator is notably higher than for the other approaches which makes it hardly real-time capable. The other estimation methods take equally long for each algorithm while the CE RDiff K-Med has a much shorter



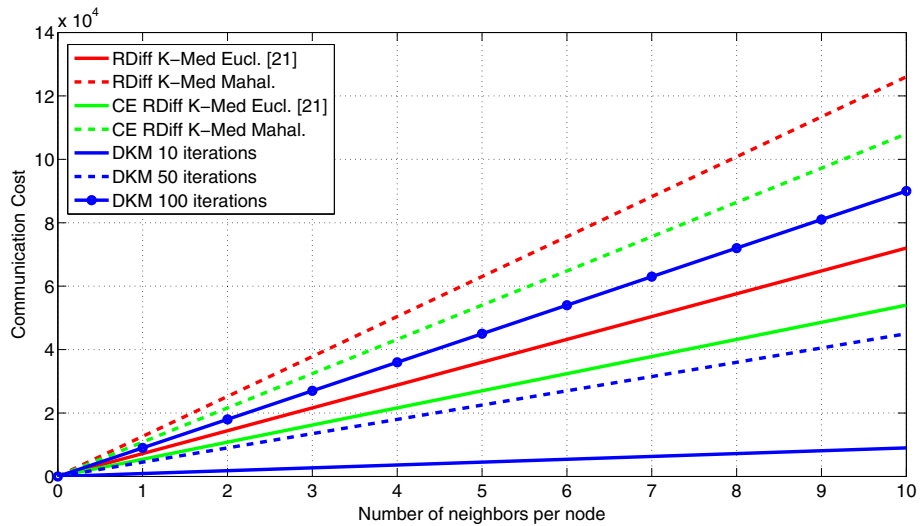


Fig. 12 Communication cost for different neighborhood sizes

computation time due to the smaller data sets it has to work with.

6 Conclusions

Two generic robust diffusion-based distributed hybrid classification algorithms were proposed, which can be adapted to various object/source labeling applications in a decentralized MDMT network. A performance comparison to the DKM was provided and the proposed methods showed promising results. Even in direct comparison with the DKM which permanently has access to all available samples, since it is operating in batch mode, our proposed online methods provide comparable error rates to the DKM using 50 iterations and more. Unlike the DKM,

both the RDiff K-Med and CE RDiff K-Med are potentially real-time capable.

The choice of the distance metric has a considerable impact on the performance of the proposed classification algorithms. Using the Mahalanobis distance yields significantly smaller error rates compared to the Euclidean distance while resulting in higher communication costs and computation time.

Future work will include the application of this algorithm to real-world speech source labeling, object labeling in camera networks as well as labeling of semantic information based on occupancy grid maps for autonomous mapping and navigation with multiple rescue robots [32].

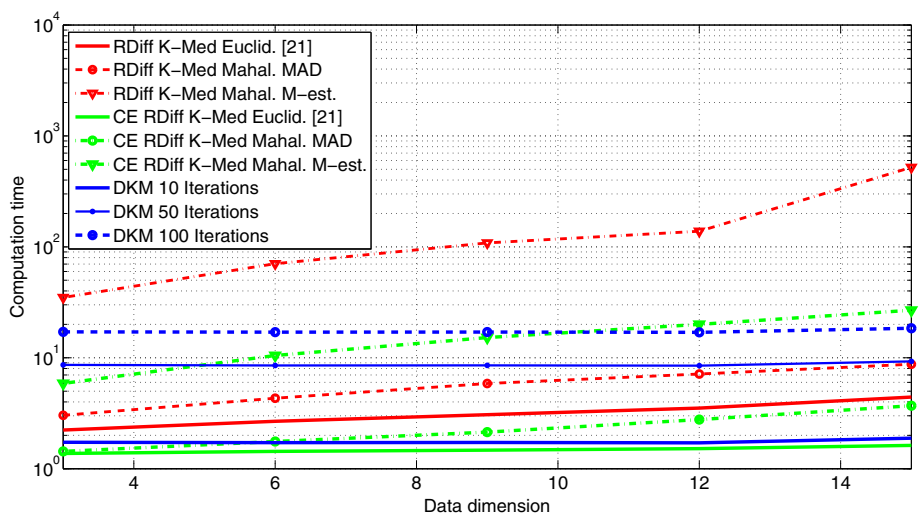


Fig. 13 Computation time in dependency on the dimension of the feature vectors

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work of P. Binder was supported by the LOEWE initiative (Hessen, Germany) within the NICER project and by the German Research Foundation (DFG). The work of M. Muma was supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission (HANDICAMS), under FET-Open grant number: 323944.

Received: 6 August 2015 Accepted: 26 February 2016

Published online: 11 March 2016

References

1. E Hänsler, *Statistische Signale: Grundlagen und Anwendungen*. (Springer, Berlin, 2013)
2. A Bertrand, M Moonen, Distributed signal estimation in sensor networks where nodes have different interests. *Signal Process.* **92**(7), 1679–1690 (2012)
3. N Bogdanovic, J Plata-Chaves, K Berberidis, Distributed incremental-based LMS for node-specific adaptive parameter estimation. *IEEE Trans. Signal Process.* **62**(20), 5382–5397 (2014)
4. J Plata-Chaves, A Bertrand, M Moonen, in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE Int. Conf. on*. Distributed signal estimation in a wireless sensor network with partially overlapping node-specific interests or source observability, (2015), pp. 5808–5812
5. J Chen, C Richard, AH Sayed, Diffusion LMS over multitask networks. *IEEE Trans. Signal Process.* **63**(11), 2733–2748 (2015)
6. J Chen, C Richard, AH Sayed, Multitask diffusion adaptation over networks. *IEEE Trans. Signal Process.* **62**(16), 4129–4144 (2014)
7. E Hänsler, G Schmidt, *Speech and Audio Processing in Adverse Environments*. (Springer, Berlin, 2008)
8. S Chouvardas, M Muma, K Hamaidi, S Theodoridis, AM Zoubir, in *Proc. 40th IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*. Distributed robust labeling of audio sources in heterogeneous wireless sensor networks, (2015), pp. 5783–5787
9. FK Teklehaymanot, M Muma, B Béjar-Haro, P Binder, AM Zoubir, M Vetterli, in *Proc. 12th IEEE AFRICON (accepted)*. Robust diffusion-based unsupervised object labelling in distributed camera networks, (2015)
10. A D'Costa, A Sayeed, in *IEEE Military Communications Conference (MILCOM)*. Data versus decision fusion for distributed classification in sensor networks, vol. 1, (2003), pp. 585–5901
11. D Li, KD Wong, YH Hu, AM Sayeed, Detection, classification and tracking of targets in distributed sensor networks. Technical report, Department of Electrical and Computer Engineering, University of Wisconsin-Madison, USA
12. F Fagnani, S Fosson, C Ravazzi, A distributed classification/estimation algorithm for sensor networks. *SIAM J Control Optim.* **52**(1), 189–218 (2014)
13. M Hai, S Zhang, L Zhu, Y Wang, in *Ind. Control and Electron. Eng. (ICICEE), 2012 Int. Conf. On*. A survey of distributed clustering algorithms, (2012), pp. 1142–1145
14. E Kokiopoulou, P Frossard, Distributed classification of multiple observation sets by consensus. *IEEE Trans. Signal Process.* **59**(1), 104–114 (2011)
15. B Malhotra, I Nikolaidis, J Harms, Distributed classification of acoustic targets in wireless audio-sensor networks. *Comput. Netw.* **52**(13), 2582–2593 (2008)
16. RD Nowak, Distributed em algorithms for density estimation and clustering in sensor networks. *IEEE Trans. Signal Process.* **51**(8), 2245–2253 (2003)
17. P Forero, A Cano, GB Giannakis, et al., Distributed clustering using wireless sensor networks. *IEEE J. Sel. Topics Signal Process.* **5**(4), 707–724 (2011)
18. S-Y Tu, AH Sayed, Distributed decision-making over adaptive networks. *IEEE Trans. Signal Process.* **62**(5), 1054–1069 (2014)
19. D Wang, J Li, Y Zhou, in *IEEE/SP 15th Workshop on Stat. Signal Process. (SSP)*. Support vector machine for distributed classification: a dynamic consensus approach, (2009), pp. 753–756
20. X Zhao, AH Sayed, Distributed clustering and learning over networks. *IEEE Trans. Signal Process.* **63**(13), 3285–3300 (2015)
21. P Binder, M Muma, AM Zoubir, in *Proc. 40th IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*. Robust and computationally efficient diffusion-based classification in distributed networks, (Brisbane, Australia, 2015), pp. 3432–3436
22. X Zhao, AH Sayed, in *Proc. International Workshop on Cognitive Information Processing (CIP)*. Clustering via diffusion adaptation over networks, (2012), pp. 1–6
23. AH Sayed, Adaptation, learning, and optimization over networks. *Found. Trends Mach. Learn.* **7**(4-5), 311–801 (2014)
24. S Khawatmi, AM Zoubir, AH Sayed, in *Proc. 23rd European Signal Processing Conf. (EUSIPCO)*. Nice, France. Decentralized clustering over adaptive networks, (Nice, France, 2015), pp. 2745–2749
25. AH Sayed, Adaptive networks. *Proc. IEEE.* **102**(4), 460–497 (2014)
26. AM Zoubir, V Koivunen, Y Chakhchoukh, M Muma, Robust estimation in signal processing: a tutorial-style treatment of fundamental concepts. *Signal Process. Mag. IEEE.* **29**(4), 61–80 (2012)
27. PA Forero, V Kekatos, GB Giannakis, Robust clustering using outlier-sparsity regularization. *IEEE Trans. Signal Process.* **60**(8), 4163–4177 (2012)
28. R Maronna, D Martin, V Yohai, *Robust Statistics*. (John Wiley & Sons, Chichester, 2006)
29. PJ Huber, et al., Robust estimation of a location parameter. *Ann. Math. Stat.* **35**(1), 73–101 (1964)
30. FS Cattivelli, AH Sayed, Diffusion LMS strategies for distributed estimation. *IEEE Trans. Signal Process.* **58**(3), 1035–1048 (2010)
31. D Estrin, L Girod, G Pottie, M Srivastava, in *IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*. Instrumenting the world with wireless sensor networks, vol. 4, (2001), pp. 2033–2036
32. S Kohlbrecher, J Meyer, T Graber, K Petersen, U Klingauf, O von Stryk, in *RoboCup 2013: Robot World Cup XVII*. Hector open source modules for autonomous mapping and navigation with rescue robots (Springer, Berlin, 2014), pp. 624–631

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com