ROBUST AND COMPUTATIONALLY EFFICIENT DIFFUSION-BASED CLASSIFICATION IN DISTRIBUTED NETWORKS

Patricia Binder, Michael Muma and Abdelhak M. Zoubir

Signal Processing Group,
Technische Universität Darmstadt,
DE-64283 Darmstadt, Germany,
{binder,muma,zoubir}@spg.tu-darmstadt.de

ABSTRACT

Today's wireless sensor networks provide the possibility to monitor physical environments via small low-cost wireless devices. Given the large amount of sensed data, efficient and robust classification becomes a critical task in many applications. Typically, 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 challenging research question in such cases is the design of data clustering and classification rules when each sensor collects a set of unlabelled observations that are drawn from a known number of classes. We propose two robust distributed hybrid classification algorithms, i.e., the Diffusion K-Medians and the Communicationally Efficient Distributed K-Medians. An extensive performance analysis in comparison to a benchmark algorithm is provided that investigates the error rates in dependence of different parameters of a distributed sensor network, and also considers communication cost. Our proposed algorithms, which are insensitive to outliers and various parameters, are applicable to on-line classification problems and scale well w.r.t. the number of classes.

Index Terms— K-medians, diffusion, distributed classification, robust, outlier

1. INTRODUCTION

Recent advances in hardware technology, distributed signal processing and communication networking are currently leading to a world of ambient sensing, processing and communication. *Ad-hoc* wireless sensor networks (WSN), consisting of small low-cost nodes with a sensing unit, processing capability, a battery or secondary power supply, and a radio for wireless communication are able to carry out complex signal processing tasks. In a 2014 survey, application scenarios, such as ambient intelligence, pervasiveness, monitoring, have been declared to be of high research interest and market relevance [1]. A challenging research question in such applications is the design of data clustering and classification rules when each sensor collects a set of unlabelled observations that are drawn from a known number of classes.

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: Performing in-network classification without a FC is motivated by considerations that deal with privacy and secrecy when agents may not be comfortable sharing their data with remote fusion centers. In other cases, such as, e.g., in cloud computing, the data itself may already be available in dispersed locations. Furthermore, distributed methods are able to perform classification, without collapsing entirely in case of a FC failure.

This motivates looking for in-network classification algorithms where a minimum amount of information is exchanged among single-hop neighbors. Although several methods have been proposed in the last years that deal with distributed data clustering and classification [2, 3, 4, 5, 6, 7, 8, 9, 10, 11], most of them still assume the presence of a fusion center [5, 7], are hardly real-time capable [3] or need a set of prelabelled training data for training beforehand [6, 11]. Several distributed adaptive strategies, such as incremental, consensus, and diffusion algorithms have been developed in the last few years. For a recent overview, and a comparison of these techniques, see [12]. In [9], a distributed K-Means (DKM) algorithm that uses the consensus strategy was proposed. We provide an alternative hybrid diffusion-based approach which presents a classification method based on preliminary clustering of the data. Since many areas of engineering today concern problems where the distribution of the measurements is far from Gaussian as it contains outliers, which cause the distribution to be heavy tailed [13], our algorithm uses K-Medians clustering, which is less sensitive to outliers than the K-Means.

Contributions: Two robust in-network distributed classification algorithms, i.e., the Diffusion K-Medians and the Communicationally Efficient Diffusion K-Medians, are proposed. It is shown that the performance of the first algorithm can be approached with the second algorithm with a considerably lower between-sensor communication cost. Unlike the DKM, which serves as a benchmark, the proposed algorithms are applicable to real-time classification problems. Furthermore, they are robust against outliers in the feature vectors. An extensive simulation-based performance analysis is provided that investigates the error rates in dependence of different WSN parameters, and also considers communication and computational costs.

Organization: Section 2 introduces the notation, provides the problem formulation, and briefly revisits the DKM [9], which serves as a benchmark. Section 3 is dedicated to the proposal and description of two robust diffusion-based classification algorithms, while Section 4 provides an extensive Monte-Carlo simulation study. Section 5 concludes the paper and provides future research directions.

This work was supported by the project HANDiCAMS which acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 323944.

2. PROBLEM FORMULATION AND EXISTING APPROACH

Consider a network with J nodes distributed over some geographic region. Two nodes are connected if they are able to communicate directly with each other. The set of nodes connected to node $j \in \{1, ..., J\} =: \mathcal{J}$ 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 $|\mathcal{B}_j|$.

The task of the distributed network is to gather data and to perform distributed unsupervised classification. Every node j collects a set of N_j observations $\boldsymbol{D}_j = \{\boldsymbol{d}_j(n), n = 1, ..., N_j\}$ of dimension q, where $\boldsymbol{d}_j(n)$ denotes the nth observation at sensor j. The entries of the observation vectors are defined as being features and are assumed to be independent of each other.

Each observation $\boldsymbol{d}_j(n)$ is assumed to belong to a certain class $\mathcal{C}_k$ with $k \in 1, ..., K$ with k denoting the indice of the given class. The total number of classes K is assumed to be known a priori. The aim is for every node j to assign each observation $\boldsymbol{d}_j(n)$ to a class k by communicating with its neighbors instead of transmitting all observations to a master node. The classification should be real-time capable so that a new observation can be assigned on-line without the necessity of all recorded observations being available.

2.1. Distributed K-Means (DKM)

As a benchmark, this paper considers the *Distributed K-Means* (DKM) algorithm by Forero et. al., for details, see [9]. 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

$$\arg \min_{\boldsymbol{w}_k, \boldsymbol{\mu}_{jk}^p(n)} \frac{1}{2} \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N_j} \boldsymbol{\mu}_{jk}^p(n) \|\boldsymbol{d}_j(n) - \boldsymbol{w}_k\|^2, \quad (1)$$

where $\boldsymbol{w}_k$ is the cluster center for class k, $\boldsymbol{\mu}_{jk}(n) \in [0, 1]$ is the membership coefficient of $\boldsymbol{d}_j(n)$ 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 (1) while exchanging the resulting parameters among neighboring nodes.

Although the DKM achieves very good performance in many scenarios, the main drawback is that the clustering is 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 right away. In addition to that, the performance of the DKM is limited in scenarios where feature vectors contain outliers.

3. PROPOSED METHODS

In this section, two new robust in-network distributed classification algorithms are presented.

3.1. Diffusion K-Medians

The proposed classification methodology begins with a local initialization phase: each sensor j collects a number of N_t observations and performs K-medians clustering on these observations, which partitions the N_t features $\{\boldsymbol{d}_j(n), n = 1, ..., N_t\}$ into K sets $\mathcal{C}_k$ so that the ℓ_1-distance within each cluster is minimized:

$$\arg \min_{\boldsymbol{w}_{jk}} \sum_{k=1}^{K} \sum_{n=1}^{N_t} \|\boldsymbol{d}_j(n) - \boldsymbol{w}_{jk}\|_1 \quad (2)$$

Each center is the component-wise median of the points of each cluster.

The features are assigned to each class $\mathcal{C}_k$ based on the minimal ℓ_1-distances and are stored in an initial matrix $\boldsymbol{S}_{jk}^0$. Based on all elements in $\mathcal{C}_k$, the initial cluster centroid $\boldsymbol{\psi}_{jk}^0$ for each node j is estimated using the median

$$\hat{\boldsymbol{\psi}}_{jk}^0 = \text{median}\left(\boldsymbol{S}_{jk}^0\right), \quad (3)$$

This completes the initialization phase, which is followed by the exchange phase[1], where each new observation $\boldsymbol{d}_j(n)$, $n = N_t + 1, ..., N_j$, is classified as follows:

1. Exchange Step: Each node j exchanges its stored observations $\boldsymbol{S}_{jk}(n)$ with its neighbors $i \in \mathcal{B}_j$. The own and received observations for each class are stored in the matrix $\boldsymbol{V}_{jk}(n)$.

2. Adaptation Step: Each node j determines a preliminary estimate $\hat{\boldsymbol{\psi}}_{jk}(n)$ based on the observations stored in $\boldsymbol{V}_{jk}(n)$ analogously to (3) with $\boldsymbol{V}_{jk}(n)$ replacing $\boldsymbol{S}_{jk}^0$.

3. Exchange Step: Each node exchanges $\hat{\boldsymbol{\psi}}_{jk}(n)$ with its neighbors.

4. Combination Step: Each node j adapts its estimates according to

$$\hat{\boldsymbol{w}}_{jk}(n) = \alpha \cdot \hat{\boldsymbol{\psi}}_{jk}(n) + (1 - \alpha) \cdot \sum_{b \in \mathcal{B}_j / \{j\}} a_{bk}(n) \cdot \hat{\boldsymbol{\psi}}_{bk}(n) \quad (4)$$

with α denoting a factor which trades off the weight given to the own and the neighborhood estimates, respectively. Further,

$$a_{bk}(n) = 1/\sqrt{\|\hat{\boldsymbol{\psi}}_{bk}(n) - \hat{\boldsymbol{\psi}}_{jk}(n)\|} \quad (5)$$

with subsequent normalization such that $\sum_{b \in \mathcal{B}_j / \{j\}} a_{bk}(n) = 1$.

In the next step, observation $\boldsymbol{d}_j(n)$ is classified by evaluating its Euclidean distance to each of the class centroid estimates $\hat{\boldsymbol{w}}_{jk}(n)$

$$d_{\text{Eucl}}(\boldsymbol{d}_j(n), \hat{\boldsymbol{w}}_{jk}(n)) = \sqrt{(\boldsymbol{d}_j(n) - \hat{\boldsymbol{w}}_{jk}(n))^T (\boldsymbol{d}_j(n) - \hat{\boldsymbol{w}}_{jk}(n))} \quad (6)$$

and assigns the observation to $\mathcal{C}_k$ for which (6) is minimized.

3.2. Communicationally Efficient (CE) Diffusion K-Medians

Since the *Diffusion K-Medians* is demanding in terms of communication between sensors, which is a major contributor the devices energy consumption [14], an algorithm is proposed which yields similar performance with reduced communication between the nodes.

The local clustering phase is identical with the *Diffusion K-Medians*, except that $\boldsymbol{S}_{jk}(n)$ is not exchanged between nodes (see Table 1: Steps 1-9), while the classification procedure is modified as follows:

1. Adaptation Step: Based on the feature vectors $\boldsymbol{d}_j(m), m = 1, ..., n$ stored in $\boldsymbol{S}_{jk}(n)$, each node calculates its intermediate estimates $\hat{\boldsymbol{\psi}}_{jk}(n)$ according to (3).

2. Exchange Step: Instead of broadcasting the entire feature vectors, the nodes share only their estimates of the cluster centers $\hat{\boldsymbol{\psi}}_{jk}(n)$ with their neighbors.

3. Combine Step: Each sensor j combines its neighbor's estimates analogously to (4) in order to obtain improved estimates $\hat{\boldsymbol{w}}_{jk}(n)$.

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 $\boldsymbol{d}_j(n)$ is classified.

[1] Since the order in which the cluster centroids are stored by K-Medians might differ between nodes, a joint ordering of the storage and exchange of the data, is necessary. The re-sorting takes place via the Euclidean distance relative to an arbitrary reference node $\in \mathcal{B}_j$.

Algorithm: Diff. K-Medians

Local Clustering Phase

for the first N_t **feature vectors do**
  for all $j = 1, ..., J$ **do**
    perform K-medians according to (2)
    calculate $\hat{\boldsymbol{\psi}}_{jk}^{0}$ via (3)
    store classified data in $\boldsymbol{S}_{jk}^{0}$
  end for
  for all $j = 1, ..., J$ **do**
    perform synchronisation of cluster estimates
  end for
  for all $j = 1, ..., J$ **do**
    exchange $\boldsymbol{S}_{jk}(n)$ within $\mathcal{B}_j$
    store received data in $\boldsymbol{V}_{jk}(n)$
  end for
end for

Distributed Classification Phase

for $n = N_t + 1, .., N_j$ **do**
  for all $j = 1, ..., J$ **do**
    broadcast an update for $\boldsymbol{V}_{jk}(n)$ within $\mathcal{B}_j$
  end for
  for all $j = 1, ..., J$ **do**
    determine $\hat{\boldsymbol{\psi}}_{jk}(n)$ (3)
  end for
  for all $j = 1, ..., J$ **do**
    broadcast $\hat{\boldsymbol{\psi}}_{jk}(n)$ within $\mathcal{B}_j$
  end for
  for all $j = 1, ..., J$ **do**
    determine $\hat{\boldsymbol{w}}_{jk}(n)$ via (4)
    calculate distances from $\boldsymbol{d}_j(n)$ to all $\hat{\boldsymbol{w}}_{jk}(n)$ by evaluating (6)
    assign $\boldsymbol{d}_j(n)$ to class $\mathcal{C}_k$ which minimizes (6)
    add $\boldsymbol{d}_j(n)$ to $\boldsymbol{V}_{jk}(n)$
  end for
end for

Table 1. Summary of the Diffusion K-Medians algorithm.

Algorithm: "CE Diff. K-Medians"

Distributed Classification Phase

for $n = N_t + 1, .., N_j$ **do**
  for all $j = 1, ..., J$ **do**
    determine $\hat{\boldsymbol{\psi}}_{jk}(n)$ according to (3)
    broadcast $\hat{\boldsymbol{\psi}}_{jk}(n)$ within $\mathcal{B}_j$
  end for
  for all $j = 1, ..., J$ **do**
    determine $\hat{\boldsymbol{w}}_{jk}(n)$ via (4)
    calculate distances from $\boldsymbol{d}_j(n)$ to all $\hat{\boldsymbol{w}}_{jk}(n)$ by evaluating (6)
    assign $\boldsymbol{d}_j(n)$ to class $\mathcal{C}_k$ which minimizes (6)
    add $\boldsymbol{d}_j(n)$ to $\boldsymbol{S}_{jk}(n)$
  end for
end for

Table 2. Summary of the Communicationally Efficient Diffusion K-Medians algorithm.

4. NUMERICAL EXPERIMENTS

This section evaluates the performance of the proposed algorithms numerically in terms of the error rate in a broad range of conditions. Furthermore, the computational and communication costs are considered. Our proposed methods are compared to the DKM [9].

4.1. 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 4 neighboring nodes which have the smallest Euclidean distance. The classification is performed for $K = 3$ classes. Each sample $\boldsymbol{d}_j(n)$ is drawn at random from class k for which the densitiy is given by $\mathcal{N}(\boldsymbol{d}_j(n); \boldsymbol{w}_k, \boldsymbol{\Sigma}_k)$ with class centers $\mathbf{w}_1 = (1,1,1)^T$, $\mathbf{w}_2 = (1,4,3)^T$, $\mathbf{w}_3 = (3,1,1)^T$ and covariance matrices $\boldsymbol{\Sigma}_1 = (1, 0.01, 0.01)^T\mathbf{I}_3$, $\boldsymbol{\Sigma}_2 = (0.16, 4, 0.16)^T\mathbf{I}_3$ and $\boldsymbol{\Sigma}_3 = (0.25, 0.01, 4)^T\mathbf{I}_3$. Note that the classes overlap partially and the standard deviations for each direction in feature space differ considerably. Each node has $N_j = 80$ samples available, $N_t = 20$ for the initialization and $N_j - N_t$ for real-time classification. K-Medians is run 3 times, and the result which minimizes (refeq:kmedians) is used for the classification. The tuning parameters for the benchmark algorithm DKM are $p = \nu = 2$, and $t = 10, 20, 50, 100$ iterations having all $N_j = 80$ samples per node available. All displayed results are averages over 100 Monte-Carlo runs and the error rate devides the number of correctly classified feature vectors by the total number of feature vectors excluding the outlying data.

4.2. Simulation Results

Figure 1 shows the error rate as a function of the available number of feature vectors.

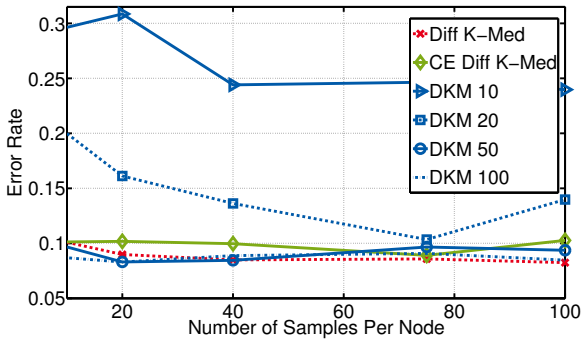

Fig. 1. Error rate as a function of the available number of feature vectors per node.

The performance for varying distances between the clusters is depicted in Fig. 2, where each integer on the x-axis describes one of the following scenarios: In *Scenario 1* the clusters overlap partially with cluster centers $\mathbf{w}_1 = (1,1,1)^T$, $\mathbf{w}_2 = (1,2,3)^T$, $\mathbf{w}_3 = (2,1,1)^T$ and covariance matrices as defined above. For *Scenario 2* $\mathbf{w}_1 = (1,1,1)^T$, $\mathbf{w}_2 = (1,4,3)^T$, $\mathbf{w}_3 = (3,1,1)^T$. For *Scenario 3* we define $\mathbf{w}_1 = (1,1,1)^T$, $\mathbf{w}_2 = (0,5,3)^T$, $\mathbf{w}_3 = (3,3,7)^T$ and *Scenario 4* is executed with $\mathbf{w}_1 = (1,1,1)^T$, $\mathbf{w}_2 = (0,7,5)^T$, $\mathbf{w}_3 = (4,4,9)^T$.

Fig. 3 depicts the error rate as a function of the percentage of zero-mean additive Gaussian outliers with $\mathcal{N}(\boldsymbol{d}_j(n); \boldsymbol{w}_k, 100\mathbf{I}_3)$.

We next investigate the robustness against skewed outlier distributions. Outliers are generated from a chi-square distribution with different degrees of freedom v for each class: For $\mathcal{C}_1$, a random vector of dimension q with $v_1 = 3$ is added, while for $\mathcal{C}_2$ a vector with

$v_2 = 5$ is subtracted and for $\mathcal{C}_3$ a different random number is drawn for each direction in space: $v_{3,1} = 4$, $v_{3,2} = 1$ and $v_{3,3} = 7$ for x, y and z direction, respectively, whereby $v_{3,2} = 1$ is subtracted from the y-component. In this setup $\mathbf{w}_1 = (1,1,1)^T$, $\mathbf{w}_2 = (0,5,3)^T$, $\mathbf{w}_3 = (3,3,7)^T$, all other parameters are unchanged.

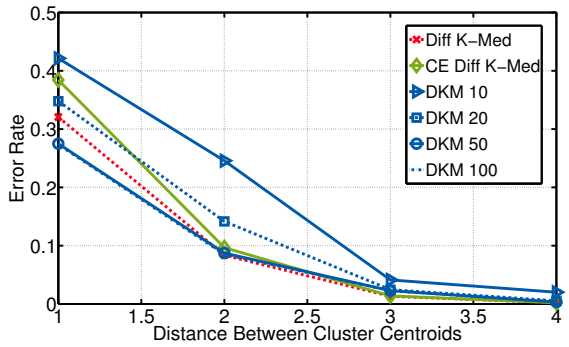

Fig. 2. Error rate in dependence of the distance between the cluster centroids.

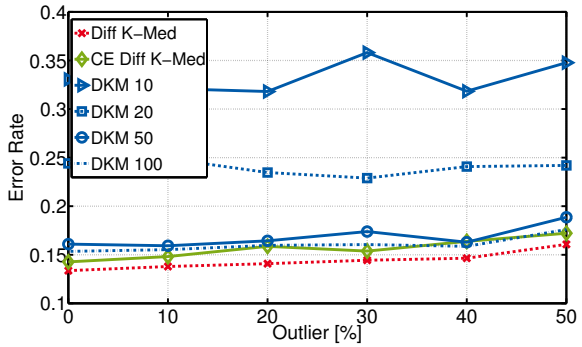

Fig. 3. Error rate for different percentages of Gaussian outliers.

4.3. Computational Time and Communication Cost

A performance measure of great importance is the communication cost, which contributes stronger to the energy consumption in the wireless devices than the computational cost [14]. Fig. 5 depicts the communication costs for the standard scenario in dependence the neighborhood cardinalities. The communication cost is specified in data units, where one matrix entry forms one unit. until the final amount of centers is reached. A significantly

Especially in practical applications with low-cost devices, computational capabilities may be limited. The computation time as a function of the dimension of the data is thus considered in Fig. 6.

5. CONCLUSIONS

Two robust diffusion-based distributed hybrid classification algorithms were proposed. A performance comparison to the DKM was provided and the proposed methods showed promising results. The proposed methods are readily extendible so as to encompass other robust estimators of the cluster centers, e.g., of the M-type. Further, other distance measures, such as the (robust) Mahalanobis distance can be integrated by exchanging the locally computed covariance matrices.

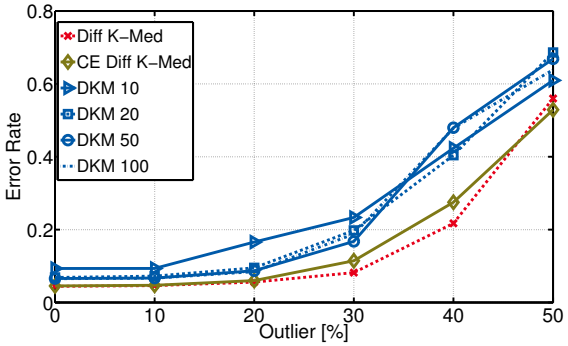

Fig. 4. Error rate for different percentages of chi-sqaured outliers with different degrees of freedom v for each class.

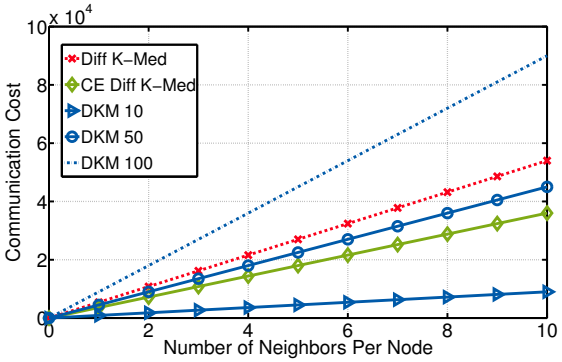

Fig. 5. Communication cost for different neighborhood sizes.

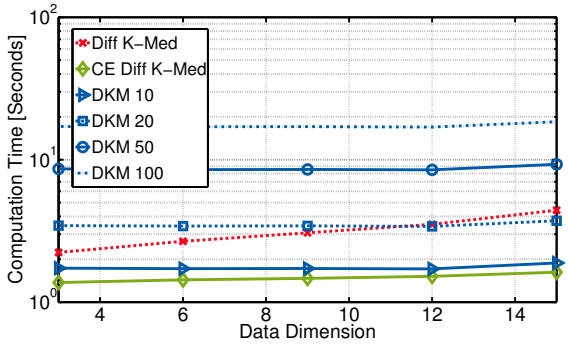

Fig. 6. Computation time in dependence of the data dimension.

6. REFERENCES

[1] M. Conti and S. Giordano, "Mobile ad hoc networking: milestones, challenges, and new research directions," *IEEE Communic. Mag.*, vol. 52, no. 1, pp. 85–96, January 2014.

[2] A. D'Costa and A. M. Sayeed, "Data versus decision fusion for distributed classification in sensor networks," in *Proc. IEEE Military Comm. Conf. (MILCOM '03)*, Oct 2003, vol. 1, pp. 585–590.

[3] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed target classification and tracking in sensor networks," *Proc. IEEE*, vol. 91, no. 8, pp. 1163–1171, Aug 2003.

[4] F. Fagnani, S. Fosson, and C. Ravazzi, "A distributed classification/estimation algorithm for sensor networks," *SIAM J. Control Optim.*, vol. 52, no. 1, pp. 189–218, 2014.

[5] M. Hai, S. Zhang, L. Zhu, and Y. Wang, "A survey of distributed clustering algorithms," in *Proc. Int. Conf. Industr. Control and Electron. Engin. (ICICEE'12)*, Aug 2012, pp. 1142–1145.

[6] E. Kokiopoulou and P. Frossard, "Distributed classification of multiple observation sets by consensus," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 104–114, Jan 2011.

[7] B. Malhotra, I. Nikolaidis, and J. Harms, "Distributed classification of acoustic targets in wireless audio-sensor networks," *Computer Networks*, vol. 52, no. 13, pp. 2582–2593, 2008.

[8] R. D. Nowak, "Distributed EM algorithms for density estimation and clustering in sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2245–2253, Aug. 2003.

[9] P. A. Forero, A. Cano, and G. B. Giannakis, "Distributed clustering using wireless sensor networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 707–724, 2011.

[10] S.-Y. Tu and A. H. Sayed, "Distributed decision-making over adaptive networks," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1054–1069, March 2014.

[11] D. Wang, J. Li, and Y. Zhou, "Support vector machine for distributed classification: A dynamic consensus approach," in *Proc. IEEE 15th Workshop Statist Signal Process. (SSP '09)*, Aug 2009, pp. 753–756.

[12] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.

[13] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma, "Robust estimation in signal processing: a tutorial-style treatment of fundamental concepts," *IEEE Signal Process. Mag.*, vol. 29, no. 4, pp. 61–80, July 2012.

[14] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP '01)*, 2001, vol. 4, pp. 2033–2036.