IN-NETWORK ADAPTIVE CLUSTER ENUMERATION FOR DISTRIBUTED CLASSIFICATION AND LABELING

Freweyni K. Teklehaymanot[1,2]*, Michael Muma*[1]*, Jun Liu*[1]*, Abdelhak M. Zoubir*[1,2]

[1] Signal Processing Group
Technische Universität Darmstadt
Merckstr. 25, 64283 Darmstadt, Germany
{muma, jliu, zoubir}@spg.tu-darmstadt.de

[2] Graduate School CE
Technische Universität Darmstadt
Dolivostr. 15, D-64293 Darmstadt, Germany
teklehaymanot@gsc.tu-darmstadt.de

ABSTRACT

A crucial first step for signal processing decentralized sensor networks with node-specific interests is to agree upon a common unique labeling of all observed sources in the network. The knowledge “who observes what” is required, e.g. in node-specific audio or video signal enhancement to form node clusters of common interest. Recently proposed in-network distributed adaptive classification and labeling algorithms assume knowledge on the number of objects (clusters), which is not necessarily available in real-world applications. Thus, we consider the problem of estimating the number of data-clusters in the distributed adaptive network set-up. We propose two distributed adaptive cluster enumeration methods. They combine the diffusion principle, where the nodes share information within their local neighborhood only (without fusion center), with the X-means and the PG-means cluster enumeration. Performance is evaluated via simulations and the applicability of the methods is illustrated using a distributed camera network where moving objects appear and disappear from the Line-of-Sight (LOS) and the number of clusters becomes time-varying.

***Index Terms*—** Distributed Cluster Enumeration; Distributed Classification; Object Labeling; Camera Network; X-means; PG-means; MDMT; Diffusion;

1. INTRODUCTION

Distributed adaptive signal processing and communication networking are advancing rapidly. This has led to new paradigms for signal and parameter estimation. One such paradigm is where Multiple Devices cooperate in Multiple Tasks (MDMT). Herein, a network of devices with node-specific interests adaptively optimizes its behavior, e.g., to jointly solve a decentralized signal or parameter estimation problem [1, 2]. This is different from the classical wireless sensor network setup, in which multiple devices perform one single joint task [1]. A crucial first step that has been recently addressed in the MDMT paradigm is the common unique labeling of all observed sources in the network. For instance, a node-specific audio signal enhancement requires a common unique labeling of all relevant speech sources that are observed by the network [3]. Also, in an image enhancement task, it is of practical importance to answer the question: *Who observes what?* [4]. This question can be addressed via in-network adaptive classification and labeling 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., [3–5]. The above methods assume knowledge of the number of objects (clusters), which is not necessarily available in real-world applications. Thus, this paper considers the problem of estimating the number of data-clusters in a distributed adaptive network.

For the single node case, determining the number of clusters has attracted considerable interest in the last decade, e.g., [6–9]. In this paper, we propose two distributed adaptive cluster enumeration methods based on the diffusion principle in [10]. The first one, the diffusion based non-splitting X-means (DX-means), estimates the number of clusters via an improved non-splitting X-means [6–8]. While the second proposed method, the diffusion based PG-means (DPG-means), is based on the PG-means [9]. The proposed algorithms adaptively estimate the number of clusters sequentially using streaming data. This is of high practical value, e.g., in a distributed camera set-up where moving objects appear and disappear from the LOS and the number of clusters becomes time-varying.

The paper is organized as follows. Section 2 formulates the distributed cluster enumeration problem and Section 3 presents the proposed methods in detail. A numerical evaluation using simulated data and a multi-camera video example is provided in Section 4. Section 5 concludes the paper.

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. The work of F. Teklehaymanot is supported by the ’Excellence Initiative’ of the German Federal and State Governments and the Graduate School of Computational Engineering at Technische Universität Darmstadt.

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

2. PROBLEM FORMULATION

Consider a wireless camera network as the one depicted in Fig. 1, where spatially distributed cameras (nodes) monitor continuously a common scene from different viewpoints. Let J be the number of nodes in the network and let $\boldsymbol{x}_j \in \mathbb{R}^d$ denote the d-dimensional feature vector extracted at the j^{th} node with class label $\mathcal{C}_j \in \{1, \ldots, K\}$. The set of nodes that communicate with node $j \in 1, ..., J$ are denoted by neighborhood $\mathcal{B}_j$. In the camera network of Fig. 1, nodes $j \in \{5, 6\}$ are moving and the remaining nodes are stationary. Due to the different viewpoints, even at the same time instant, the number of objects observed by different cameras differs. Our research goal is to adaptively estimate the network-wide number of clusters (objects), given that this number K is in some range $K_{\text{min}} \leq K \leq K_{\text{max}}$. We propose distributed adaptive cluster enumeration methods that are based on the diffusion principle. In this way, we obtain an estimate of K, whereby each node j utilizes the information it received from its neighborhood $\mathcal{B}_j$. This allows for a global agreement regarding the number of objects in the scene, using only local interactions.

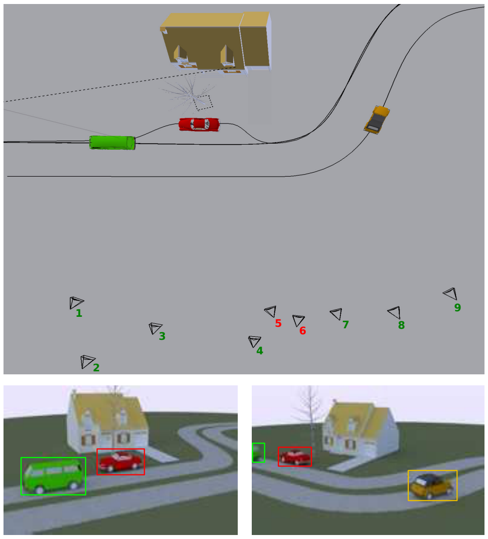


Fig. 1. A wireless camera network continuously observing a scene. The top image shows a camera network with $J = 9$ nodes distributed in space and observing the scene from different viewpoints. The bottom left and right images show frames captured at the same time by camera 1 and camera 9, respectively.

3. PROPOSED DIFFUSION BASED CLUSTER ENUMERATION METHODOLOGY

Two methods based on the diffusion principle [10] are proposed, see Fig. 2. Having observed N_t feature vectors $\boldsymbol{x}_j(n)$, each node j forms a matrix $\boldsymbol{S}_j(n)$. Optionally, $\boldsymbol{x}_j(n)$ is exchanged within $\mathcal{B}_j$ before adapting, i.e., determining the cluster number based on $\boldsymbol{S}_j(n)$ which contains all available $\boldsymbol{x}_i(n), i \in \mathcal{B}_j$. If the exchange of $\boldsymbol{x}_i(n), i \in \mathcal{B}_j$ is left out, $\boldsymbol{S}_j(n) = \boldsymbol{x}_j(n)$ and the methods proceed analogously. The proposed cluster enumeration algorithms are based on information criteria (DX-means, Sec. 3.1) or hypothesis testing (DPG-means, Sec. 3.2). The intermediate cluster number estimate $\hat{k}_j^0$ is improved upon by including neighboring estimates $\hat{k}_i^0, i \in \mathcal{B}_j$ which are combined to form the final decision. As data streams sequentially, the steps shown in Fig. 2 are repeated to provide an online in-network estimate. In this paper, the combine step at node j is chosen as $\text{median}\{\hat{k}_i^0\}, i \in \mathcal{B}_j$. Table 1 summarizes the algorithm in pseudo-code. The next sections provide details on the decision making at the j^{th} node for the two proposed methods.

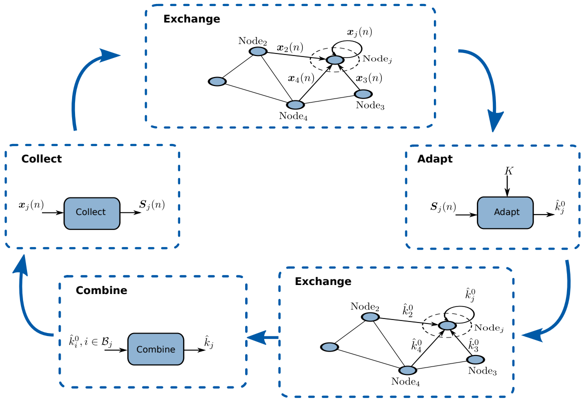


Fig. 2. An overview of the distributed diffusion based cluster enumeration methods.

3.1. The DX-means Algorithm

In the DX-means algorithm, each node j calculates the Bayesian Information Criterion (BIC) [6, 11] score of the alternative models $\mathcal{M}_{jK}$ as follows:

$$\text{BIC}_j(\mathcal{M}_{jK}) = \hat{l}_{jK}(\boldsymbol{S}_j(n)) - \frac{\gamma_{jK}}{2} \log N_j, \tag{1}$$

where $\hat{l}_{jK}$ is the log-likelihood of the feature vectors based on the model $\mathcal{M}_{jK}$, and γ_{jK} is the number of parameters in the model. The alternative models M_{jK} correspond to solutions with different values of K. Under the spherical Gaussian assumption, the log-likelihood of $\boldsymbol{S}_j(n)$ is given as:

$$\begin{aligned}\hat{l}_{jK}(\boldsymbol{S}_j(n)) &= \log \prod_{n=1}^{N_j} P(\boldsymbol{S}_j(n)) = \sum_{n=1}^{N_j} \log P(\boldsymbol{S}_j(n)) \\ &= \sum_{k=1}^{K} \left(n_{jk} \log \frac{n_{jk}}{N_j} - \frac{d n_{jk}}{2} \log(2\pi\hat{\sigma}_{jk}^2) \right) \\ &- \sum_{k=1}^{K} \left(\frac{1}{2\hat{\sigma}_{jk}^2} \sum_{\boldsymbol{S}_j(n) \in \mathcal{C}_{jk}} \|\boldsymbol{S}_j(n) - \hat{\boldsymbol{\psi}}_{jk}\|^2 \right),\end{aligned} \tag{2}$$

Distributed diffusion based cluster enumeration methods

```
for $m = 1, 2, \ldots$ do
    for all $j = 1, ..., J$ do
        collect $N_t$ feature vectors and store in
        $\boldsymbol{S}_j(n), n = 1, \ldots, N_j$, where $N_j = mN_t$
    end for
    for all $j = 1, ..., J$ do
        exchange $\boldsymbol{x}_j(n)$ within $\mathcal{B}_j$
    end for
    for all $j = 1, ..., J$ do
        adapt model order using DX-means as in
        Sec. 3.1 or DPG-means as in Sec. 3.2
    end for
    for all $j = 1, ..., J$ do
        exchange $\hat{k}_j^0$ within $\mathcal{B}_j$
    end for
    for all $j = 1, ..., J$ do
        combine $\hat{k}_i^0, i \in \mathcal{B}_j$ by taking the median
    end for
end for
```

Table 1. Summary of the distributed diffusion based cluster enumeration methods.

where $\hat{\boldsymbol{\psi}}_{jk}$ is the cluster centroid, $\hat{\sigma}_{jk}^2$ is the cluster variance maximum-likelihood (ML) estimate, and n_{jk} is the number of feature vectors that belong to cluster $\mathcal{C}_{jk}$.

After calculating the BIC score of the alternative models $\mathcal{M}_{jK}$, each node j estimates the intermediate cluster number $\hat{k}_j^0$ using the knee point detection method (KP) as in [8]. Further possibilities to find $\hat{k}_j^0$ are the Successive Difference (SD) and the global maximum of the BIC curve. The successive difference of three consecutive points is calculated as $\text{SD}_j(\mathcal{M}_{jK}) = \text{BIC}_j(\mathcal{M}_{j(K-1)}) - 2\text{BIC}_j(\mathcal{M}_{jK}) + \text{BIC}_j(\mathcal{M}_{j(K+1)})$.

3.2. The DPG-means Algorithm

DPG-means obtains the parameters of model $\mathcal{M}_{jK}$ using the EM algorithm. For a given model $\mathcal{M}_{jK}$, we compute the parameter ML estimates and assume that $\boldsymbol{S}_j(n) \sim \mathcal{N}(\hat{\boldsymbol{\psi}}_{jk}, \hat{\boldsymbol{\Sigma}}_{jk})$. DPG-means projects $\boldsymbol{S}_j(n)$ and $\mathcal{M}_{jK}$ to $\mathbb{R}^{1\times 1}$ using a unit length random projection vector $\mathcal{P}$. Now $\boldsymbol{s}_j^{\mathcal{P}} = \mathcal{P}^T \boldsymbol{S}_j \sim \mathcal{N}(\hat{\psi}_{jk}^{\mathcal{P}}, (\hat{\sigma}_{jk}^{\mathcal{P}})^2)$, where $\hat{\psi}_{jk}^{\mathcal{P}} = \mathcal{P}^T \hat{\boldsymbol{\psi}}_{jk}$ and $(\hat{\sigma}_{jk}^{\mathcal{P}})^2 = \mathcal{P}^T \hat{\boldsymbol{\Sigma}}_{jk} \mathcal{P}$.

After projection, a Kolmogorov-Smirnov (KS) test is used to check if the projected model fits the projected data. The critical value z_{jk} is calculated via

$$z_{jk} = \max_{\boldsymbol{s}_j^{\mathcal{P}}(n)} |\mathcal{F}(\boldsymbol{s}_j^{\mathcal{P}}(n)) - \mathcal{G}(\boldsymbol{s}_j^{\mathcal{P}}(n))|, \tag{3}$$

where $\mathcal{F}(\boldsymbol{s}_j^{\mathcal{P}}(n))$ is the Gaussian cumulative distribution function (cdf) formed with $\hat{\psi}_{jk}^{\mathcal{P}}$ and $(\hat{\sigma}_{jk}^{\mathcal{P}})^2$, whereas $\mathcal{G}(\boldsymbol{s}_j^{\mathcal{P}}(n))$ is the empirical cdf of $\boldsymbol{s}_j^{\mathcal{P}}(n)$. Different methods have been proposed to compute the threshold to which z_{jk} is compared to in order to accept Gaussianity, and therewith the model under test [12, 13]. Best performance was obtained by using Monte-Carlo techniques where data has been generated from $\mathcal{N}(\hat{\psi}_{jk}^{\mathcal{P}}, (\hat{\sigma}_{jk}^{\mathcal{P}})^2)$ to determine the threshold as in [12, 13].

4. RESULTS

4.1. Performance Measures

In this section, performance comparison of the DX-means and DPG-means algorithms is provided for two simulated clustering examples and one multi-camera video data set. As a performance measure, we calculate the average estimated cluster number ($\hat{k}_{\text{ave}}$) and the average estimation rate (AER) as follows

$$\hat{k}_{\text{ave}} = \frac{1}{J \times \text{mc}} \sum_{j=1}^{J} \sum_{i=1}^{\text{mc}} \hat{k}_j(i) \tag{4}$$

$$\text{AER} = \frac{1}{J \times \text{mc}} \sum_{j=1}^{J} \sum_{i=1}^{\text{mc}} (\hat{k}_j(i) == \text{true clusters}). \tag{5}$$

Here, mc indicates the number of Monte-Carlo experiments. The convergence rate ($\mathcal{R}_c$) of the proposed algorithms in terms of the number of feature vectors per node is computed as $\mathcal{R}_c = \text{AER}(m+1) - \text{AER}(m) < \epsilon$, where $\epsilon = 0.05$ for the simulated data sets and $\epsilon = 0.1$ for the multi-camera video example.

4.2. Simulation Setup

All simulation results are an average of 1000 Monte-Carlo experiments and for each experiment a different random network topology is considered. We compare our results with a distributed non-cooperative and centralized implementation. In the centralized network, all nodes send their intermediate cluster number estimate $\hat{k}_j^0$ to the fusion center and the fusion center computes $\hat{k}_j$ by taking the median of $\hat{k}_j^0$.

In the first simulated data set (Data-1), we have assumed that $\boldsymbol{x}_j(n) \sim \mathcal{N}(\boldsymbol{\psi}_k, \boldsymbol{\Sigma}_k)$ with $\boldsymbol{\psi}_1 = [-1, 0]^T$, $\boldsymbol{\psi}_2 = [4, 0]^T$, $\boldsymbol{\psi}_3 = [0, 5]^T$, $\boldsymbol{\psi}_4 = [9, 4]^T$, $\boldsymbol{\psi}_5 = [3, 9]^T$, $\boldsymbol{\Sigma}_1 = [0.2, 0.4]^T \boldsymbol{I}_2$, $\boldsymbol{\Sigma}_2 = [0.6, 0.6]^T \boldsymbol{I}_2$, $\boldsymbol{\Sigma}_3 = [0.4, 0.2]^T \boldsymbol{I}_2$, $\boldsymbol{\Sigma}_4 = [0.2, 0.2]^T \boldsymbol{I}_2$, and $\boldsymbol{\Sigma}_5 = [0.3, 0.5]^T \boldsymbol{I}_2$. $\boldsymbol{I}_d$ denotes the d-dimensional identity matrix. For the second simulated data set (Data-2), $\boldsymbol{x}_j(n) \sim \mathcal{N}(\boldsymbol{\psi}_k, \boldsymbol{\Sigma}_{jk})$, where $\boldsymbol{\Sigma}_{jk}$ vary slightly across the network: $\boldsymbol{\psi}_1 = [-1, 0, 7]^T$, $\boldsymbol{\psi}_2 = [3, 0, 8]^T$, $\boldsymbol{\psi}_3 = [0, 5, 1]^T$, $\boldsymbol{\psi}_4 = [9, 4, 4]^T$, $\boldsymbol{\psi}_5 = [3, 9, 5]^T$, $\boldsymbol{\psi}_6 = [5, 5, 1.5]^T$, $\boldsymbol{\Sigma}_{j1} = \alpha[0.2, 0.4, 0.2]^T \boldsymbol{I}_3$, $\boldsymbol{\Sigma}_{j2} = \alpha[0.6, 0.3, 0.5]^T \boldsymbol{I}_3$, $\boldsymbol{\Sigma}_{j3} = \alpha[0.4, 0.2, 0.1]^T \boldsymbol{I}_3$, $\boldsymbol{\Sigma}_{j4} = \alpha[0.3, 0.3, 0.3]^T \boldsymbol{I}_3$, $\boldsymbol{\Sigma}_{j5} = \alpha[0.3, 0.5, 0.3]^T \boldsymbol{I}_3$, and $\boldsymbol{\Sigma}_{j6} = \alpha[0.4, 0.4, 0.4]^T \boldsymbol{I}_3$, where $\alpha = 1$ for 30% of the nodes, $\alpha = 2$ for 40% of the

Table 2. The time required to reach convergence in a distributed cooperative network set-up.

	DX-means			DPG-means
	KP	SD	max(BIC)	
Data-1	10	90	40	10
Data-2	40	80	60	50
Multi-camera video	20	30	10	30

nodes, and $\alpha = 3$ for the remaining nodes. For the simulated data sets, we consider a scenario with $J = 10$ nodes, a neighborhood size of $\mathcal{B}_j = 4$ and 50 feature vectors per cluster in each case, where each node j collects $N_t = 10$ feature vectors at a time.

For the multi-camera video example, we used $J = 7$ stationary cameras and $\mathcal{B}_j = 6$. A video of 95 frames was captured by the cameras to test the performance of the proposed methods. The multi-camera video example is challenging in the sense that the video has very low resolution, the cameras monitor the moving objects (cars) from different angles, and there are few feature vectors. We used a Gaussian Mixture model (GMM) foreground detector to separate moving objects from the background and the feature vectors used are a concatenation of SURF [14] and color features. For the color histogram, the detected foreground is subdivided into three concentric rings and a 10-bin histogram per color channel is computed for every region in a cumulative manner (i.e., adding the previous region). The concatenation of these three histograms gives us the descriptor of each color channel, and concatenation of the three color channels result in a 90-dimensional color feature. Thus, the feature vectors used are 211-dimensional.

4.3. Simulation Results

The time taken for convergence and the performance of the methods at convergence is provided in Tables 2 and 3, respectively. For Data-1, both DX-means and DPG-means are able to converge very fast and attain similar average estimation rate at convergence. For Data-2, the DX-means outperforms the DPG-means in both the convergence speed and average estimation rate. For the DX-means algorithm, knee point detection performs better than successive difference and the global maximum of the BIC curve. Thus, we have used the knee point detection method of the DX-means algorithm to generate the plots.

Fig. 3 displays the average estimated clusters as a function of the number of feature vectors per node for Data-1. In this experiment, for every $n = 5m \times N_t$, feature vectors from a new cluster appear and the number of clusters increase by one. The error bars show the estimation errors and are defined as twice the standard deviation. In general, the distributed cooperative network performs much better than the distributed non-cooperative network and the result of the cooperative implementation approaches the centralized one as the number of feature vectors per node increases.

Table 3. AER (in %) at convergence in a distributed cooperative network set-up.

	DX-means			DPG-means
	KP	SD	max(BIC)	
Data-1	100	87.7	96	99.3
Data-2	94.3	84.6	96.3	87.1
Multi-camera video	100	86.5	0	58

The average estimated clusters as a function of the number of feature vectors per node for Data-2 is shown in Fig. 4. Here, feature vectors from all clusters are available from the beginning. This data set contains cluster overlap and for a small number of feature vectors per node the number of feature vectors per cluster becomes very small. For this data set, DX-means converges faster than DPG-means. In the multi-view camera network, only DX-means with knee point detection is able to provide the correct cluster number. DX-means using the global maximum of the BIC curve completely breaks down and goes for K_{max}. DPG-means is able to estimate the true number of clusters in the beginning but after that it consistently overestimates by one. Presumably this behavior is due to remaining background in the foreground of the segmented image which is treated as a separate class.

5. CONCLUSION

We proposed two in-network distributed adaptive cluster enumeration algorithms. A numerical evaluation using simulated data and a multi-camera video example have shown that the proposed diffusion based algorithms approach the performance of the centralized implementation without requiring a fusion center.

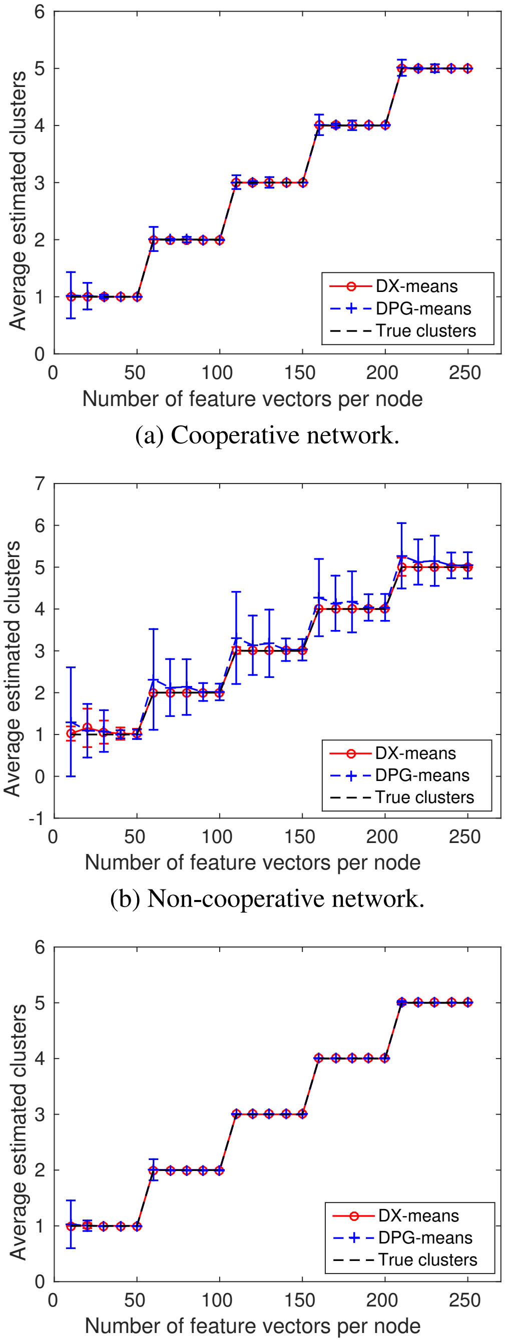
REFERENCES

[1] Bertrand, A., and Moonen, M., "Distributed signal estimation in sensor networks where nodes have different interests," *Signal Process.*, 92(7), pp. 1679–1690, 2012.

[2] Chen, J., Richard, C., and Sayed, A.H., "Diffusion LMS over multitask networks," *IEEE Trans. Signal Process.*, 63(11), pp. 2733–2748, 2015.

[3] Chouvardas, S., Muma, M., Hamaidi, K., Theodoridis, S., and Zoubir, A. M., "Distributed robust labeling of audio sources in heterogeneous wireless sensor networks," *In Proc. 40th IEEE Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*, pp. 5783–5787, 2015.

[4] Teklehaymanot, F. K., Muma, M., Béjar, B., Binder, P., Zoubir, A. M., and Vetterli, M., "Robust diffusion-based un-



(a) Cooperative network.

(b) Non-cooperative network.

(c) Centralized network.

Fig. 3. Average estimated clusters as a function of the number of feature vectors per node for Data-1. a) displays the results achieved in a distributed cooperative implementation, b) shows the results of a distributed non-cooperative implementation, and c) shows the results of a centralized implementation with a fusion center.

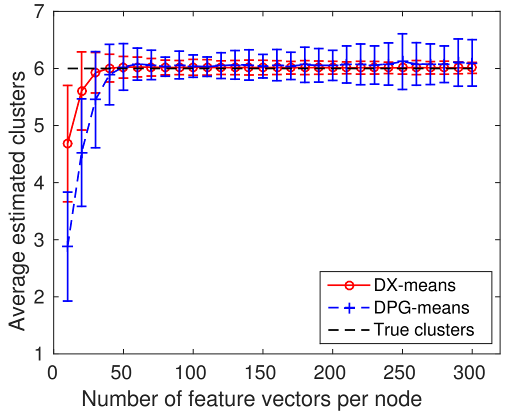


Fig. 4. Average estimated clusters as a function of the number of feature vectors per node using a distributed cooperative implementation for Data-2.

supervised object labelling in distributed camera networks," *In proc. 12th IEEE AFRICON*, 2015.

[5] Binder, P., Muma, M., and Zoubir, A. M., "Robust and adaptive diffusion-based classification in distributed networks," *EURASIP Journal on Advances in Signal Processing*, (accepted), 2016.

[6] Pelleg, D., and Moore, A., "X-means: Extending K-means with efficient estimation of the number of clusters," *In Proc. 17th Inter. Conf. on Machine Learning*, pp. 727–734, 2000.

[7] Ishioka, T., "An expansion of X-means for automatically determining the optimal number of clusters - progressive iterations of K-means and merging of the clusters," *In proc. 4th IASTED Inter. Conf. on Computational Intelligence*, pp. 91–96, July 2005,

[8] Zhao, Q., Xu, M., and Frätnti, P., "Knee point detection on Bayesian Information Criterion," *20th IEEE Inter. Conf. on Tools with Artificial Intelligence*, pp. 431–438, 2008.

[9] Feng, Y., and Hamerly, G., "PG-means: learning the number of clusters in data," *In Proc. 20th Annual Conf. on Neural Information Process. Systems*, pp. 393–400, 2006.

[10] Sayed, A. H., "Adaptive networks," *In Proc. IEEE*, 102(4), 460–497, 2014.

[11] Kass, R. E., and Wasserman, L., "A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion," *Journal of the American Statistical Association*, 90(431), 1995.

[12] Lilliefors, H.W., "On the Kolmogorov-Smirnov test of normality with mean and variance unknown," *Journal of the American Statistical Association*, 62(318), pp. 399-402, 1967.

[13] Massey, F.J., "The Kolmogorov-Smirnov test for goodness of fit," *Journal of the American Statistical Association*, 46(253), pp. 68-78, 1951.

[14] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., "Speeded Up Robust Features (SURF)," *Computer Vision and Image Understanding*, 110(3), pp. 346–359, 2008.