Sensors **2015**, *15*, 19047-19068; doi:10.3390/s150819047

OPEN ACCESS
sensors
ISSN 1424-8220
www.mdpi.com/journal/sensors

Article

Distributed Density Estimation Based on a Mixture of Factor Analyzers in a Sensor Network

Xin Wei [1,*], Chunguang Li [2], Liang Zhou [1] and Li Zhao [3]

[1] College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; E-Mail: liang.zhou@njupt.edu.cn
[2] Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China; E-Mail: cgli@zju.edu.cn
[3] School of Information Science and Engineering, Southeast University, Nanjing 210096, China; E-Mail: zhaoli@seu.edu.cn

* Author to whom correspondence should be addressed; E-Mail: xwei@njupt.edu.cn; Tel./Fax: +86-25-5706-2695.

Academic Editor: Leonhard M. Reindl

Received: 27 May 2015 / Accepted: 30 July 2015 / Published: 5 August 2015

Abstract: Distributed density estimation in sensor networks has received much attention due to its broad applicability. When encountering high-dimensional observations, a mixture of factor analyzers (MFA) is taken to replace mixture of Gaussians for describing the distributions of observations. In this paper, we study distributed density estimation based on a mixture of factor analyzers. Existing estimation algorithms of the MFA are for the centralized case, which are not suitable for distributed processing in sensor networks. We present distributed density estimation algorithms for the MFA and its extension, the mixture of Student's *t*-factor analyzers (M*t*FA). We first define an objective function as the linear combination of local log-likelihoods. Then, we give the derivation process of the distributed estimation algorithms for the MFA and M*t*FA in details, respectively. In these algorithms, the local sufficient statistics (LSS) are calculated at first and diffused. Then, each node performs a linear combination of the received LSS from nodes in its neighborhood to obtain the combined sufficient statistics (CSS). Parameters of the MFA and the M*t*FA can be obtained by using the CSS. Finally, we evaluate the performance of these algorithms by numerical simulations and application example. Experimental results validate the promising performance of the proposed algorithms.

Keywords: distributed density estimation; mixture of factor analyzers; mixture of Student's *t*-factor analyzers; sensor network

1. Introduction

Sensor networks are composed of tiny, intelligent sensor nodes that are deployed over a geographic region. This type of network has a broad range of applications, such as environmental monitoring, precision agriculture and military surveillance [1–3]. Distributed estimation over sensor networks is to estimate some parameters of interest through local computation and information exchange among neighbor nodes. Compared to centralized estimation, it does not need to send observations collected by all of the sensors to a powerful central node, so the complexity and resource consumption can be reduced. Furthermore, distributed estimation is more flexible and robust to node and/or link failure [2,4]. Recently, many distributed estimation algorithms have been proposed, such as distributed LMS [5], distributed recursive least squares (RLS) [6], distributed source location [7], distributed power allocation [8], distributed sparse estimation [9,10], distributed information theoretic learning [11] and distributed Gaussian process regression [12].

Mixture of Gaussians (GMM) is a flexible and powerful probabilistic modeling tool for density estimation. It has been used in several areas, such as pattern recognition, computer vision, signal and image analysis and machine learning. When estimating parameters in the GMM by the maximum likelihood criterion, the expectation maximization (EM) algorithm [13,14] is usually adopted. It iteratively performs the expectation step (E-step) to calculate the conditional expectations of unobserved/hidden variables and runs the maximization step (M-step) to estimate parameters of data distributions based on the result of the E-step. However, when the dimension of observations is high, the fitting performance of the GMM deteriorates or even the associated EM algorithm cannot work [15]. The main reason is that the GMM cannot realize dimensionality reduction, which is to compress highly-correlated components of observations. In this case, a mixture of factor analyzers (MFA) [16,17] can be considered. The MFA combines local factor analysis in a form of a finite mixture. As factor analysis can describe variability among high-dimensional observations in terms of potentially low-dimensional latent factors, the MFA can carry out dimensionality reduction simultaneously when finishing specific tasks. Moreover, in order to process the non-normality of data or outliers, normal distributions in the MFA can be replaced by Student's *t*-distributions, obtaining the mixture of Student's *t*-factor analyzers (M*t*FA) [18,19]. Therefore, the MFA and its extension M*t*FA are effective tools for processing high-dimensional observations [20]. They have been successfully applied in the domains of signal processing [21,22], bioinformatics [23,24] and other applied fields.

In sensor networks, GMM has been introduced for density estimation of observations [25–32]. The estimation process for the GMM needs to be realized by distributed EM algorithms. According to the way by which nodes communicate with each other, distributed EM algorithms can be classified into the incremental type [25,26], the consensus type [27] and the diffusion type [28–32]. In the incremental scheme [25,26], a long way from the first node to last node of the pre-selected path is needed. When any node along the path fails, reliability problem may happen. In the consensus-based distributed EM

algorithm for the GMM [27], a consensus filter, by which global statistics for each node is achieved, is carried out between the E-step and the M-step at each iteration. The objective is to obtain the same estimations for all nodes at each iteration. In the diffusion type of distributed estimation for the GMM, each node exchanges information only with its neighbors through a diffusion cooperative protocol. Good performance is obtained while communication overhead is kept low [2,4]. In this paper, we focus on a diffusion type of distributed estimation. Among previous studies, a distributed model order and parameter estimation algorithm for the GMM was proposed in [28]. Moreover, algorithm performance was analyzed. In [29], a diffusion-based EM algorithm was presented for distributed estimation in unreliable sensor networks. In this scenario, some nodes may be subject to data failures and report only noise. The aim of the algorithm was to achieve the optimal performance within the whole range of SNRs. In [30], information diffusion and averaging were considered and performed simultaneously. In [31], an adaptive diffusion scheme was proposed. In [32], the performance of the diffusion-based EM algorithm was analyzed. It could be considered as a stochastic approximation method [33] to find the maximum likelihood estimation of the GMM.

As the MFA can handle high-dimensional observations, which are also usually encountered in sensor networks, in this paper, we propose distributed density estimation algorithms for the MFA and its extension M*t*FA. We represent these two algorithms as D-MFA and D-M*t*FA, respectively. Specially, for each node in the sensor network, we define an objective function as the linear combination of local log-likelihoods, whose combination weights are determined by the number of observations in the corresponding neighbor nodes. After local sufficient statistics are computed, the current node calculates its combined sufficient statistics by a linear weighed combination of these local sufficient statistics from nodes in its neighborhood set. Finally, parameters of the MFA and M*t*FA are updated using the combined sufficient statistics. Apart from the distributed processing of the MFA and the M*t*FA in this paper, there are two other differences from the existing algorithms. First, in the relevant algorithms [25,27,28], mixing proportions in the GMM are different at each node, whereas means and covariances are the same throughout the network. On the contrary, in this paper, all of the parameters in the MFA or the M*t*FA are the same throughout the network. Using this design, distributed clustering and classification can be done in arbitrary nodes after the estimation process finishes. Second, for each node, the objective function is directly defined. The combination of weights in the objective function is effectively designed.

The rest of this paper is organized as follows. In Section 2, a brief overview of the MFA and the M*t*FA are provided. In Section 3, the D-MFA algorithm and D-M*t*FA algorithm are formulated. In Section 4, numerical simulations for the synthetic observations are performed to illustrate the effectiveness and advantages of the proposed algorithms. Moreover, the application of these algorithms to distributed clustering is also presented. Finally, conclusions are drawn in Section 5.

The acronyms mentioned in this paper are listed in the following.

Acronym list:

GMM	Gaussian mixture model
EM	expectation maximization
E-step	expectation step
M-step	maximization step
MFA	mixture of factor analyzers

M*t*FA	mixture of Student's *t*-factor analyzers
D-MFA	distributed density estimation algorithm for the MFA
D-M*t*FA	distributed density estimation algorithm for the M*t*FA
CSS	combined sufficient statistics
LSS	local sufficient statistics
S-MFA	standard EM algorithm for the MFA
S-M*t*FA	standard EM algorithm for the M*t*FA
NC-MFA	non-cooperation MFA
NC-M*t*FA	non-cooperation M*t*FA
D-GMM	distributed density estimation algorithm for the GMM
D-*t*MM	distributed density estimation algorithm for the Student's*t*-mixture model

2. Preliminaries: MFA and M*t*FA

2.1. Mixture of Factor Analyzers

Let the observed dataset be $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_N\}$. In the MFA, it assumes that each p-dimensional data vector $\mathbf{y}_n$ is generated as:

$$\mathbf{y}_n = \boldsymbol{\mu}_i + \mathbf{A}_i \mathbf{u}_n + \mathbf{e}_{ni} \quad \text{with prob. } \pi_i \ (i = 1, \ldots, I) \tag{1}$$

where I is the number of mixing components. The corresponding q-dimensional ($q < p$) factor $\mathbf{u}_n \sim \mathcal{N}(\mathbf{u}_n|\mathbf{0}, \mathbf{I}_q)$ is independent of the $\mathbf{e}_{ni} \sim \mathcal{N}(\mathbf{e}_{ni}|\mathbf{0}, \mathbf{D}_i)$, where $\mathbf{D}_i$ is a $p \times p$ diagonal matrix. The parameter $\boldsymbol{\mu}_i$ is the mean of the i-th analyzer, and $\mathbf{A}_i$ ($p \times q$) is the linear transformation known as the factor loading matrix. The so-called mixing proportions π_i ($i = 1, ..., I$) are nonnegative and sum to one. The standard EM algorithm for the MFA is given in [15,16].

2.2. Mixture of Student's t-Factor Analyzers

Since the MFA adopts the normal family for the distributions of the errors and the latent factors, it is sensitive to outliers. An obvious way to improve the robustness of this model for observations having longer tails than normal is using the t-family of elliptically-symmetric distributions. Therefore, the M*t*FA has been proposed in [18]. In the M*t*FA, it assumes that p-dimensional data vector $\mathbf{y}_n$ is generated in the same way as that in the MFA, as shown in Equation (1). However, the distributions of q-dimensional ($q < p$) factor $\mathbf{u}_n$ and noise $\mathbf{e}_{ni}$ are $t(\mathbf{u}_n|\mathbf{0}, \mathbf{I}_q, \nu_i)$ and $t(\mathbf{e}_{ni}|\mathbf{0}, \mathbf{D}_i, \nu_i)$, respectively. In the above Student's t distributions, ν_i is called the degree of freedom that controls the length of the tails of the distributions. With this modification, the M*t*FA is more robust to outliers and can process the non-normality of observations in a better way [23]. In essence, $t(\mathbf{u}_n|\mathbf{0}, \mathbf{I}_q, \nu_i)$ and $t(\mathbf{e}_{ni}|\mathbf{0}, \mathbf{D}_i, \nu_i)$ can be respectively regarded as average Gaussian scale distributions $\mathcal{N}(\mathbf{u}_n|\mathbf{0}, \mathbf{I}_q/w_{ni})$ and $\mathcal{N}(\mathbf{e}_{ni}|\mathbf{0}, \mathbf{D}_i/w_{ni})$ with the Gamma distributed precision scalar w_{ni}, that is:

$$t(\mathbf{u}_n|\mathbf{0}, \mathbf{I}_q, \nu_i) = \int \mathrm{d}w_{ni} \mathcal{N}(\mathbf{u}_n|\mathbf{0}, \mathbf{I}_q/w_{ni}) \mathcal{G}(w_{ni}|\nu_i/2, \nu_i/2)$$

$$t(\mathbf{e}_{ni}|\mathbf{0}, \mathbf{D}_i, \nu_i) = \int \mathrm{d}w_{ni} \mathcal{N}(\mathbf{e}_{ni}|\mathbf{0}, \mathbf{D}_i/w_{ni}) \mathcal{G}(w_{ni}|\nu_i/2, \nu_i/2)$$

where $\mathcal{G}(\cdot)$ denotes the Gamma distribution. The standard EM algorithm for the M*t*FA is given in [14,18].

3. Distributed Estimation Algorithms for the MFA and M*t*FA

3.1. Network Model and Objective Function

Consider a sensor network with M nodes. The m-th node has N_m data observations $\mathbf{Y}_m = \{\mathbf{y}_{m,n}\}_{n=1,...,N_m}$ $(m = 1, ..., M)$, and $\mathbf{y}_{m,n}$ denotes the n-th observation in node m. The distribution of each p-dimensional observation $\mathbf{y}_{m,n}$ is modeled by the MFA, defined in Equation (1). It is noted that the factor associated with $\mathbf{y}_{m,n}$ here is represented as $\mathbf{u}_{m,n}$. The parameter set of the MFA is $\mathbf{\Theta} = \{\pi_i, \boldsymbol{\mu}_i, \mathbf{A}_i, \mathbf{D}_i\}_{i=1,...,I}$, which is to be estimated.

The network topology is described by a graph. Let W denote the distance that a node can communicate via wireless radio links. Nodes m and l are connected if the Euclidean distance $d_{m,l}$ between m and l is less than or equal to W. Moreover, a graph is connected if for any pair of nodes (m, n), there exists a path from m to n. The neighborhood set of node m, denoted by R_m, is defined as the one-hop neighbors of node m (including m itself). For example, in Figure 1, the dashed circle represents the neighborhood set of node m, containing Node 1, Node 2, node l and node m itself.

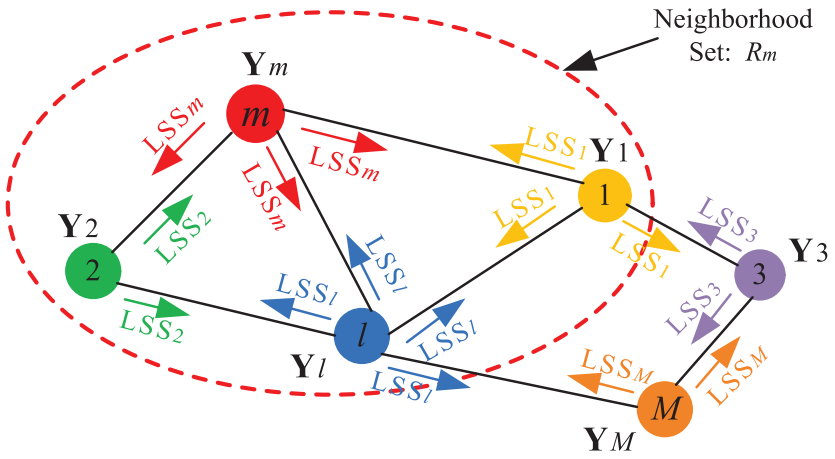

Figure 1. A sensor network consists of a collection of cooperating nodes. Node m only exchanges information (e.g., local sufficient statistics (LSS) in the proposed D-MFA and D-M*t*FA algorithms) with nodes in R_m.

In order to design the D-MFA and the D-M*t*FA algorithms, the objective functions should be carefully specified at first. Here, we take node m in the sensor network for example. We define the objective function $\mathcal{F}_m(\mathbf{\Theta})$ of the MFA at node m as a linear combination of local log-likelihoods $\log p(\mathbf{Y}_l|\mathbf{\Theta})$ associated with nodes l in its neighborhood R_m.

$$\mathcal{F}_m(\mathbf{\Theta}) = \sum_{l \in R_m} c_{lm} \cdot \log p(\mathbf{Y}_l|\mathbf{\Theta}) = \sum_{l \in R_m} c_{lm} \sum_{n=1}^{N_l} \log \left(\sum_{i=1}^{I} \pi_i N(\mathbf{y}_{l,n}|\boldsymbol{\mu}_i, \mathbf{A}_i, \mathbf{D}_i) \right) \tag{2}$$

where $\{c_{lm}\}_{l \in R_m}$ are some non-negative combination coefficients satisfying the condition $\sum_{l \in R_m} c_{lm} = 1$, $c_{lm} = 0$ if node $l \notin R_m$.

It should be emphasized that when defining $\mathcal{F}_m(\mathbf{\Theta})$, we consider two important factors. First, as node m can only communicate with its neighbors, it is reasonable to define $\mathcal{F}_m(\mathbf{\Theta})$ as a combination of local log-likelihoods $\log p(\mathbf{Y}_l|\mathbf{\Theta}),\ (l \in R_m)$. When estimating $\mathbf{\Theta}$, node m can make use of the information from nodes in R_m. Due to the effect of information diffusion, each node can obtain all information directly or indirectly from other nodes. Second, the contributions of different local log-likelihoods $\log p(\mathbf{Y}_l|\mathbf{\Theta}),\ (l \in R_m)$ for the estimation of $\mathbf{\Theta}$ at node m may also be different. The combination coefficient c_{lm} weights the importance of information flow from the node l $(l \in R_m)$. Therefore, how to choose c_{lm} is important. Here, we adopt a simple, but effective mechanism that c_{lm} is determined by:

$$c_{lm} = \frac{N_l}{\sum_{l' \in R_m} N_{l'}} \tag{3}$$

If node l has a larger number of observations N_l, the information from this node makes a larger contribution to obtain more accurate parameter estimations. Therefore, a larger combination coefficient c_{lm} in Equation (3) can further make this contribution prominent. In the future, a more effective implementation, such as an adaptive strategy [4], can be considered to determine these combination coefficients better.

3.2. Distributed Density Estimation Algorithm for the MFA

After the objective functions $\mathcal{F}_m(\mathbf{\Theta})$ $(m = 1, ..., M)$ have been determined, the next task is to estimate parameters $\mathbf{\Theta}$ in the MFA by maximizing $\mathcal{F}_m(\mathbf{\Theta})$. For node l, an I-dimensional binary latent variable $\mathbf{z}_{l,n}$, which is associated with $\mathbf{y}_{l,n}$, is introduced. As the MFA is a mixture model, $z_{l,n,i}(= 1)$ denotes that $\mathbf{y}_{l,n}$ belongs to the i-th component of the MFA. The latent variables for node l in the neighborhood R_m are $\mathbf{U}_l = \{\mathbf{u}_{l,n}\}_{n=1,...,N_l}$, $\mathbf{Z}_l = \{\mathbf{z}_{l,n}\}_{n=1,...,N_l}$, $(l \in R_m)$. Now, $\mathcal{F}_m(\mathbf{\Theta})$ $(m = 1, ..., M)$ in Equation (2) can be expressed as:

$$\mathcal{F}_m(\mathbf{\Theta}) = \sum_{l \in R_m} c_{lm} \cdot \left\{ \log \sum_{\mathbf{Z}_l} \int \mathrm{d}\mathbf{U}_l \, p(\mathbf{Y}_l, \mathbf{Z}_l, \mathbf{U}_l|\mathbf{\Theta}) \right\}$$

where:

$$p(\mathbf{Y}_l, \mathbf{Z}_l, \mathbf{U}_l|\mathbf{\Theta}) = p(\mathbf{Z}_l|\mathbf{\Theta}) p(\mathbf{U}_l|\mathbf{Z}_l, \mathbf{\Theta}) p(\mathbf{Y}_l|\mathbf{Z}_l, \mathbf{U}_l\mathbf{\Theta}) \tag{4}$$

The three conditional probabilities in Equation (4) are:

$$p(\mathbf{Z}_l|\mathbf{\Theta}) = \prod_{n=1}^{N_l} \prod_{i=1}^{I} \pi_i^{z_{l,n,i}}$$

$$p(\mathbf{U}_l|\mathbf{Z}_l, \mathbf{\Theta}) = \prod_{n=1}^{N_l} \prod_{i=1}^{I} \mathcal{N}(\mathbf{u}_{l,n}|\mathbf{0}, \mathbf{I}_q)^{z_{l,n,i}}$$

$$p(\mathbf{Y}_l|\mathbf{Z}_l, \mathbf{U}_l, \mathbf{\Theta}) = \prod_{n=1}^{N_l} \prod_{i=1}^{I} \mathcal{N}(\mathbf{y}_{l,n}|\boldsymbol{\mu}_i + \mathbf{A}_i\mathbf{u}_{l,n}, \mathbf{D}_i)^{z_{l,n,i}}$$

Here, we derive the distributed estimation algorithm with the aid of the standard EM algorithm [13]. First, we introduce two distributions $q(\mathbf{Z}_l)$ and $q(\mathbf{U}_l|\mathbf{Z}_l)$ defined over the latent variables. For any choice of $q(\mathbf{Z}_l)$ and $q(\mathbf{U}_l|\mathbf{Z}_l)$, the following decomposition holds:

$$\mathcal{F}_m(\boldsymbol{\Theta}) = \sum_{l \in R_m} c_{lm} \cdot \mathcal{L}(q_l, \boldsymbol{\Theta}) + \sum_{l \in R_m} c_{lm} \cdot \mathrm{KL}(q_l || p_l) \tag{5}$$

where:

$$\mathcal{L}(q_l, \boldsymbol{\Theta}) = \sum_{\mathbf{Z}_l} q(\mathbf{Z}_l) \int \mathrm{d}\mathbf{U}_l q(\mathbf{U}_l|\mathbf{Z}_l) \log \left\{ \frac{p(\mathbf{Y}_l, \mathbf{Z}_l, \mathbf{U}_l|\boldsymbol{\Theta})}{q(\mathbf{Z}_l) q(\mathbf{U}_l|\mathbf{Z}_l)} \right\}$$

$$\mathrm{KL}(q_l||p_l) = -\sum_{\mathbf{Z}_l} q(\mathbf{Z}_l) \int \mathrm{d}\mathbf{U}_l q(\mathbf{U}_l|\mathbf{Z}_l) \log \left\{ \frac{p(\mathbf{Z}_l|\mathbf{Y}_l, \boldsymbol{\Theta}) p(\mathbf{U}_l|\mathbf{Z}_l, \mathbf{Y}_l, \boldsymbol{\Theta})}{q(\mathbf{Z}_l) q(\mathbf{U}_l|\mathbf{Z}_l)} \right\}$$

The verification of log-likelihood decomposition can be found in [34]. As $\mathcal{F}_m(\boldsymbol{\Theta})$ as a combination of local log-likelihoods, the whole decomposition can also be expressed by a combination of local log-likelihood decompositions, as shown in Equation (5).

Moreover, $\mathrm{KL}(q_l||p_l)$ in Equation (5) is the Kullback–Leibler divergence between $q(\mathbf{Z}_l)q(\mathbf{U}_l|\mathbf{Z}_l)$ and $p(\mathbf{Z}_l|\mathbf{Y}_l, \boldsymbol{\Theta})p(\mathbf{U}_l|\mathbf{Z}_l, \mathbf{Y}_l, \boldsymbol{\Theta})$, which satisfies $\mathrm{KL}(q_l||p_l) \geq 0$. Therefore, it can be seen from Equation (5) that $\sum_{l \in R_m} c_{lm} \cdot \mathcal{L}(q_l, \boldsymbol{\Theta}) \leq \mathcal{F}_m(\boldsymbol{\Theta})$. In other words, $\sum_{l \in R_m} c_{lm} \cdot \mathcal{L}(q_l, \boldsymbol{\Theta})$ is a lower bound on $\mathcal{F}_m(\boldsymbol{\Theta})$. As direct maximization of the $\mathcal{F}_m(\boldsymbol{\Theta})$ is difficult, it can be solved by the maximization of this lower bound instead.

Suppose that the parameters estimated in the last iteration are $\boldsymbol{\Theta}^{\mathrm{old}} = \{\pi_i^{\mathrm{old}}, \boldsymbol{\mu}_i^{\mathrm{old}}, \mathbf{A}_i^{\mathrm{old}}, \mathbf{D}_i^{\mathrm{old}}\}_{i=1,\ldots,I}$. In the first stage, the lower bound $\sum_{l \in R_m} c_{lm} \cdot \mathcal{L}(q_l, \boldsymbol{\Theta}^{\mathrm{old}})$ is maximized with respect to $q(\mathbf{Z}_l)q(\mathbf{U}_l|\mathbf{Z}_l)$ while holding $\boldsymbol{\Theta}^{\mathrm{old}}$ fixed. From Equation (5), this maximum can be achieved when $\mathrm{KL}(q_l||p_l) = 0$. In other words, $q(\mathbf{Z}_l)q(\mathbf{U}_l|\mathbf{Z}_l) = p(\mathbf{Z}_l|\mathbf{Y}_l, \boldsymbol{\Theta}^{\mathrm{old}})p(\mathbf{U}_l|\mathbf{Z}_l, \mathbf{Y}_l, \boldsymbol{\Theta}^{\mathrm{old}})$. Therefore, two conditional distributions, $p(\mathbf{Z}_l|\mathbf{Y}_l, \boldsymbol{\Theta}^{\mathrm{old}})$ and $p(\mathbf{U}_l|\mathbf{Z}_l, \mathbf{Y}_l, \boldsymbol{\Theta}^{\mathrm{old}})$, should be computed.

Concretely, for node l ($l \in R_m$), $p(\mathbf{Z}_l|\mathbf{Y}_l, \boldsymbol{\Theta}^{\mathrm{old}})$ can be calculated by:

$$p(\mathbf{Z}_l|\mathbf{Y}_l, \boldsymbol{\Theta}^{\mathrm{old}}) = \prod_{n=1}^{N_l} \prod_{i=1}^{I} p(z_{l,n,i}|\mathbf{y}_{l,n})$$

where:

$$p(z_{l,n,i}|\mathbf{y}_{l,n}) = \frac{\pi_i^{\mathrm{old}} N(\mathbf{y}_{l,n}|\boldsymbol{\mu}_i^{\mathrm{old}}, \mathbf{A}_i^{\mathrm{old}}(\mathbf{A}_i^{\mathrm{old}})^{\mathrm{T}} + \mathbf{D}_i^{\mathrm{old}})}{\sum_{i'=1}^{I} \pi_{i'}^{\mathrm{old}} N(\mathbf{y}_{l,n}|\boldsymbol{\mu}_{i'}^{\mathrm{old}}, \mathbf{A}_{i'}^{\mathrm{old}}(\mathbf{A}_{i'}^{\mathrm{old}})^{\mathrm{T}} + \mathbf{D}_{i'}^{\mathrm{old}})} \tag{6}$$

Moreover, $p(\mathbf{U}_l|\mathbf{Z}_l, \mathbf{Y}_l, \boldsymbol{\Theta}^{\mathrm{old}})$ should also be obtained by:

$$p(\mathbf{U}_l|\mathbf{Z}_l, \mathbf{Y}_l, \boldsymbol{\Theta}^{\mathrm{old}}) = \prod_{n=1}^{N_l} \prod_{i=1}^{I} p(\mathbf{u}_{l,n}|\mathbf{y}_{l,n}, z_{l,n,i}) = \prod_{n=1}^{N_l} \prod_{i=1}^{I} N(\mathbf{u}_{l,n}|\overline{\mathbf{u}}_{l,n,i}, \boldsymbol{\Omega}_i) \tag{7}$$

The mean $\overline{\mathbf{u}}_{m,n,i}$ and covariance $\boldsymbol{\Omega}_i$ are:

$$\begin{aligned} \overline{\mathbf{u}}_{l,n,i} &= \mathbf{g}_i^{\mathrm{T}}(\mathbf{y}_{l,n} - \boldsymbol{\mu}_i^{\mathrm{old}}) \\ \boldsymbol{\Omega}_i &= \mathbf{I}_q - \mathbf{g}_i^{\mathrm{T}} \mathbf{A}_i^{\mathrm{old}} \end{aligned} \tag{8}$$

where:

$$\mathbf{g}_i = \left[\mathbf{A}_i^{\mathrm{old}}(\mathbf{A}_i^{\mathrm{old}})^{\mathrm{T}} + \mathbf{D}_i^{\mathrm{old}}\right]^{-1} \cdot \mathbf{A}_i^{\mathrm{old}} \tag{9}$$

is an intermediate variable introduced to simplify expressions in the following steps.

When the above two conditional distributions have been obtained, $q(\mathbf{Z}_l)q(\mathbf{U}_l|\mathbf{Z}_l)$ is determined and held fixed, and the lower bound $\sum_{l\in R_m} c_{lm}\cdot\mathcal{L}(q_l,\boldsymbol{\Theta})$ is maximized with respect to $\boldsymbol{\Theta}$ to get new estimated $\boldsymbol{\Theta}^{\text{new}}$. This will cause the lower bound to increase, which will necessarily cause the corresponding $\mathcal{F}_m(\boldsymbol{\Theta})$ to increase.

Concretely, the current lower bound is expressed as:

$$\sum_{l\in R_m} c_{lm}\mathcal{L}(q_l,\boldsymbol{\Theta}) = \sum_{l\in R_m} c_{lm}\sum_{\mathbf{Z}_l} p(\mathbf{Z}_l|\mathbf{Y}_l,\boldsymbol{\Theta}^{\text{old}})\int \mathrm{d}\mathbf{U}_l p(\mathbf{U}_l|\mathbf{Z}_l,\mathbf{Y}_l,\boldsymbol{\Theta}^{\text{old}}) \times\Big\{\log p(\mathbf{Y}_l,\mathbf{Z}_l,\mathbf{U}_l|\boldsymbol{\Theta}) - \log\left[p(\mathbf{Z}_l|\mathbf{Y}_l,\boldsymbol{\Theta}^{\text{old}})p(\mathbf{U}_l|\mathbf{Z}_l,\mathbf{Y}_l,\boldsymbol{\Theta}^{\text{old}})\right]\Big\} \tag{10}$$

Discard the second logarithmic term unrelated to $\boldsymbol{\Theta}$ in Equation (10); the objective function represented by $\mathcal{Q}_m(\boldsymbol{\Theta})$ at node m is:

$$\mathcal{Q}_m(\boldsymbol{\Theta}) = \sum_{l\in R_m} c_{lm}\sum_{n=1}^{N_l}\sum_{i=1}^{I} p(z_{l,n,i}|\mathbf{y}_{l,n})p(\mathbf{u}_{l,n}|\mathbf{y}_{l,n},z_{l,n,i}) \times\Big\{z_{l,n,i}\log\Big[\pi_i N(\mathbf{u}_{l,n}|\mathbf{0},\mathbf{I}_q)N(\mathbf{y}_{l,n}|\boldsymbol{\mu}_i+\mathbf{A}_i\mathbf{u}_{l,n},\mathbf{D}_i)\Big]\Big\}$$

Now, parameters in $\boldsymbol{\Theta}$ can be obtained by taking derivation of $\mathcal{Q}_m(\boldsymbol{\Theta})$ with respect to $\boldsymbol{\Theta}$.

First, π_i and $\boldsymbol{\mu}_i$ are updated to:

$$\pi_i = \frac{\sum_{l\in R_m} c_{lm}\sum_{n=1}^{N_l}\langle z_{l,n,i}\rangle}{\sum_{i'=1}^{I}\sum_{l\in R_m} c_{lm}\sum_{n=1}^{N_l}\langle z_{l,n,i'}\rangle} \tag{11}$$

and:

$$\boldsymbol{\mu}_i = \frac{\sum_{l\in R_m} c_{lm}\sum_{n=1}^{N_l}\langle z_{l,n,i}\rangle\mathbf{y}_{l,n}}{\sum_{l\in R_m} c_{lm}\sum_{n=1}^{N_l}\langle z_{l,n,i}\rangle} \tag{12}$$

respectively. Subsequently, by performing derivation of $\mathcal{Q}_m(\boldsymbol{\Theta})$ with respect to $\mathbf{A}_i$, we have:

$$\mathbf{A}_i = \frac{\sum_{l\in R_m} c_{lm}\sum_{n=1}^{N_l}\langle z_{l,n,i}\rangle(\mathbf{y}_{l,n}-\boldsymbol{\mu}_i)\langle\mathbf{u}_{l,n}\rangle^{\mathrm{T}}}{\sum_{l\in R_m} c_{lm}\sum_{n=1}^{N_l}\langle z_{l,n,i}\rangle\cdot\langle\mathbf{u}_{l,n}\mathbf{u}_{l,n}^{\mathrm{T}}\rangle} \tag{13}$$

Finally, the expression of $\mathbf{D}_i$ can be obtained in the same way,

$$\mathbf{D}_i = \mathrm{diag}\left\{\frac{\sum_{l\in R_m} c_{lm}\sum_{n=1}^{N_l}\langle z_{l,n,i}\rangle\left[(\mathbf{y}_{l,n}-\boldsymbol{\mu}_i)(\mathbf{y}_{l,n}-\boldsymbol{\mu}_i)^{\mathrm{T}} - \mathbf{A}_i\langle\mathbf{u}_{l,n}\mathbf{u}_{l,n}^{\mathrm{T}}\rangle\mathbf{A}_i^{\mathrm{T}}\right]}{\sum_{l\in R_m} c_{lm}\sum_{n=1}^{N_l}\langle z_{l,n,i}\rangle}\right\} \tag{14}$$

where $\mathrm{diag}\{\cdot\}$ denotes the operator setting off-diagonal terms to zero. In Equations (11)–(14), $\langle z_{l,n,i}\rangle$ is the expectation of $z_{l,n,i}$ given by Equation (6). $\langle\mathbf{u}_{l,n}\rangle$ and $\langle\mathbf{u}_{l,n}\mathbf{u}_{l,n}^{\mathrm{T}}\rangle$ can be obtained from Equation (7), which are:

$$\langle\mathbf{u}_{l,n}\rangle = \overline{\mathbf{u}}_{l,n,i} \;\text{ and }\; \langle\mathbf{u}_{l,n}\mathbf{u}_{l,n}^{\mathrm{T}}\rangle = \boldsymbol{\Omega}_i + \overline{\mathbf{u}}_{l,n,i}\overline{\mathbf{u}}_{l,n,i}^{\mathrm{T}} \tag{15}$$

respectively. Substituting Equation (15) into Equations (13) and (14), we have:

$$\mathbf{A}_i = \mathbf{V}_i\mathbf{g}_i(\mathbf{g}_i^{\mathrm{T}}\mathbf{V}_i\mathbf{g}_i+\boldsymbol{\Omega}_i)^{-1} \tag{16}$$

and:

$$\mathbf{D}_i = \mathrm{diag}\left\{\mathbf{V}_i - \mathbf{A}_i(\mathbf{g}_i^{\mathrm{T}}\mathbf{V}_i\mathbf{g}_i + \boldsymbol{\Omega}_i)\mathbf{A}_i^{\mathrm{T}}\right\} \tag{17}$$

where:

$$\mathbf{V}_i = \frac{\sum_{l\in R_m} c_{lm} \sum_{n=1}^{N_l} \langle z_{l,n,i}\rangle (\mathbf{y}_{l,n} - \boldsymbol{\mu}_i)(\mathbf{y}_{l,n} - \boldsymbol{\mu}_i)^{\mathrm{T}}}{\sum_{l\in R_m} c_{lm} \sum_{n=1}^{N_l} \langle z_{l,n,i}\rangle} \tag{18}$$

From Equations (11), (12) and (16)–(18), we can see, when estimating parameters $\boldsymbol{\Theta}$ at node m, that three combined sufficient statistics (CSS) must be obtained, represented as:

$$\begin{aligned}
\mathrm{CSS}_m^{(1)}[i] &= \sum_{l\in R_m} c_{lm} \cdot \mathrm{LSS}_l^{(1)}[i] \\
\mathrm{CSS}_m^{(2)}[i] &= \sum_{l\in R_m} c_{lm} \cdot \mathrm{LSS}_l^{(2)}[i] \\
\mathrm{CSS}_m^{(3)}[i] &= \sum_{l\in R_m} c_{lm} \cdot \mathrm{LSS}_l^{(3)}[i]
\end{aligned} \tag{19}$$

where:

$$\begin{aligned}
\mathrm{LSS}_l^{(1)}[i] &= \sum_{n=1}^{N_l} \langle z_{l,n,i}\rangle \\
\mathrm{LSS}_l^{(2)}[i] &= \sum_{n=1}^{N_l} \langle z_{l,n,i}\rangle \cdot \mathbf{y}_{l,n} \\
\mathrm{LSS}_l^{(3)}[i] &= \sum_{n=1}^{N_l} \langle z_{l,n,i}\rangle \cdot \mathbf{y}_{l,n} \cdot \mathbf{y}_{l,n}^{\mathrm{T}}
\end{aligned} \tag{20}$$

$\mathrm{LSS}_l = \{\mathrm{LSS}_l^{(1)}[i],\ \mathrm{LSS}_l^{(2)}[i],\ \mathrm{LSS}_l^{(3)}[i]\}_{i=1,\ldots,I}$ are local sufficient statistics (LSS) of node l. Therefore, CSS in node m is a linear combination of the LSS of nodes in R_m. If node l has a large number of observations, the accuracy of calculated LSS_l should be high and should make an important contribution to the CSS of node m. A relatively large c_{lm} in Equation (3) can make this contribution prominent, obtaining accurate estimation of $\boldsymbol{\Theta}$.

In the following, we summarize the realization process of the D-MFA algorithm.

Step 1 (Initialization): This initializes the parameters $\{\pi_i, \boldsymbol{\mu}_i, \mathbf{A}_i, \mathbf{D}_i\}_{i=1,\ldots,I}$. Each node l broadcasts the number of its observations N_l to its neighbors. When receiving this information, each node calculates the combination coefficient by Equation (3).

Step 2 (Computation): Each node l in the sensor network computes $\langle z_{l,n,i}\rangle$, $\boldsymbol{\Omega}_i$ and $\mathbf{g}_i$ by Equations (6), (8) and (9), respectively. Then, it computes three local sufficient statistics $\mathrm{LSS}_l = \{\mathrm{LSS}_l^{(1)}[i],\ \mathrm{LSS}_l^{(2)}[i],\ \mathrm{LSS}_l^{(3)}[i]\}_{i=1,\ldots,I}$ according to its own observations $\mathbf{Y}_l$, by Equation (20).

Step 3 (Diffusion): Each node l in sensor networks diffuses its local sufficient statistics LSS_l, as shown in Figure 1.

Step 4 (Combination): When node m ($m = 1, \ldots, M$) receives the local sufficient statistics from all of its one-hop neighbor nodes l ($l \in R_m$), it computes the combined sufficient statistics $\{\mathrm{CSS}_m^{(1)}[i]$, $\mathrm{CSS}_m^{(2)}[i]$, $\mathrm{CSS}_m^{(3)}[i]\}_{i=1,\ldots,I}$ by Equation (19).

Step 5 (Estimation): Node m ($m = 1, ..., M$) estimates π_i, $\boldsymbol{\mu}_i$, $\mathbf{A}_i$ and $\mathbf{D}_i$ according to Equations (11), (12), (16) and (17), respectively. Here, we substitute Equation (19) into Equations (11), (12) and (18), reformulating the estimation step as follows:

$$\begin{aligned}\pi_i &= \frac{\mathrm{CSS}_m^{(1)}[i]}{\sum_{i'=1}^{I}\mathrm{CSS}_m^{(1)}[i']}\\ \boldsymbol{\mu}_i &= \frac{\mathrm{CSS}_m^{(2)}[i]}{\mathrm{CSS}_m^{(1)}[i]}\\ \mathbf{A}_i &= \mathbf{V}_i\mathbf{g}_i(\mathbf{g}_i^{\mathrm{T}}\mathbf{V}_i\mathbf{g}_i + \boldsymbol{\Omega}_i)^{-1}\\ \mathbf{D}_i &= \mathrm{diag}\left\{\mathbf{V}_i - \mathbf{A}_i(\mathbf{g}_i^{\mathrm{T}}\mathbf{V}_i\mathbf{g}_i + \boldsymbol{\Omega}_i)\mathbf{A}_i^{\mathrm{T}}\right\}\end{aligned}$$

where:

$$\mathbf{V}_i = \frac{\mathrm{CSS}_m^{(3)}[i] - 2\mathrm{CSS}_m^{(2)}[i]\cdot\boldsymbol{\mu}_i + \mathrm{CSS}_m^{(1)}[i]\cdot\boldsymbol{\mu}_i\boldsymbol{\mu}_i^{\mathrm{T}}}{\mathrm{CSS}_m^{(1)}[i]}$$

Step 6 (Termination): Node m ($m = 1, ..., M$) calculates its current local log-likelihood as:

$$\log p(\mathbf{Y}_m|\boldsymbol{\Theta}^{\mathrm{new}}) = \sum_{n=1}^{N_m}\log\left(\sum_{i=1}^{I}\pi_i N(\mathbf{y}_{m,n}|\boldsymbol{\mu}_i, \mathbf{A}_i, \mathbf{D}_i)\right)$$

where superscript "new" denotes the newly estimated parameters at the current iteration. If $\log p(\mathbf{Y}_m|\boldsymbol{\Theta}^{\mathrm{new}}) - \log p(\mathbf{Y}_m|\boldsymbol{\Theta}^{\mathrm{old}}) < \epsilon$, node m enters the terminated state; else, go to Step 2 and start the next iteration. It is noted that the terminated nodes do no computation or communication in the following iterations. If one node cannot receive information from a neighbor node in the next iteration, the node will use the received and saved LSS information from that neighbor node at the last iteration when updating CSS. When there is no message communication or information exchange in the network, implying all nodes reach the terminated state, the algorithm ends.

3.3. Distributed Density Estimation Algorithm for the MtFA

Compared to the MFA, the main difference of the M*t*FA is that it has an additional degree of freedom parameter ν_i $(i = 1, ..., I)$. Therefore, the parameter set of the M*t*FA to be estimated is $\boldsymbol{\Theta} = \{\pi_i, \boldsymbol{\mu}_i, \mathbf{A}_i, \mathbf{D}_i, \nu_i\}_{i=1,...,I}$. Moreover, apart from $\mathbf{Z}_l$ and $\mathbf{U}_l$, the latent variable $\mathbf{W}_l = \{w_{l,n,i}\}_{i=1,...,I}^{n=1,...,N_l}$ should be introduced, explained in Section 2.2. Similarly, for node m in a sensor network, a linear combination of local log-likelihoods associated with nodes in R_m is defined as:

$$\mathcal{F}_m(\boldsymbol{\Theta}) = \sum_{l\in R_m} c_{lm}\cdot\log p(\mathbf{Y}_l|\boldsymbol{\Theta}) = \sum_{l\in R_m} c_{lm}\sum_{n=1}^{N_l}\log\left(\sum_{i=1}^{I}\pi_i\cdot t(\mathbf{y}_{l,n}|\boldsymbol{\mu}_i, \mathbf{A}_i\mathbf{A}_i^{\mathrm{T}} + \mathbf{D}_i, \nu_i)\right) \quad (21)$$

The derivation process of the D-M*t*FA algorithm is similar to that of the D-MFA, except that in Step 2, the posterior distribution of $p(w_{m,n,i}|\mathbf{y}_{m,n}, z_{m,n,i})$ and $p(\mathbf{u}_{m,n}|\mathbf{y}_{m,n}, z_{m,n,i}, w_{m,n,i})$ should be computed, and in Step 5, ν_i needs to be estimated. We put this derivation in the Appendix in detail and directly describe the D-M*t*FA algorithm here.

Step 1 (Initialization): This initializes the values of the parameters $\{\pi_i, \boldsymbol{\mu}_i, \mathbf{A}_i, \mathbf{D}_i, \nu_i\}_{i=1,...,I}$. Each node l broadcasts the number of its observations N_l to its neighbors. When receiving this information, each node calculates the combination coefficient by Equation (3).

Step 2 (Computation): Each node l in the sensor network computes five local sufficient statistics $\mathrm{LSS}_l = \{\mathrm{LSS}_l^{(1)}[i],\ \mathrm{LSS}_l^{(2)}[i],\ \mathrm{LSS}_l^{(3)}[i],\ \mathrm{LSS}_l^{(4)}[i],\ \mathrm{LSS}_l^{(5)}[i]\}_{i=1,\ldots,I}$ according to its observations $\mathbf{Y}_l$, given as:

$$\begin{aligned}
\mathrm{LSS}_l^{(1)}[i] &= \sum_{n=1}^{N_l} \langle z_{l,n,i} \rangle \\
\mathrm{LSS}_l^{(2)}[i] &= \sum_{n=1}^{N_l} \langle z_{l,n,i} \rangle \cdot \langle w_{l,n,i} \rangle \\
\mathrm{LSS}_l^{(3)}[i] &= \sum_{n=1}^{N_l} \langle z_{l,n,i} \rangle \cdot \langle w_{l,n,i} \rangle \cdot \mathbf{y}_{l,n} \\
\mathrm{LSS}_l^{(4)}[i] &= \sum_{n=1}^{N_l} \langle z_{l,n,i} \rangle \cdot \langle w_{l,n,i} \rangle \cdot \mathbf{y}_{l,n} \mathbf{y}_{l,n}^{\mathrm{T}} \\
\mathrm{LSS}_l^{(5)}[i] &= \sum_{n=1}^{N_l} \langle z_{l,n,i} \rangle \cdot \log \langle w_{l,n,i} \rangle
\end{aligned} \tag{22}$$

The expressions of expectations $\langle z_{l,n,i} \rangle$ and $\langle w_{l,n,i} \rangle$ in Equation (22) are given in the Appendix. Moreover, the intermediate variables $\mathbf{\Omega}_i$ and $\mathbf{g}_i$ should also be prepared for simplifying expressions in Step 5, shown in the Appendix.

Step 3 (Diffusion): Each node l in the sensor network diffuses its local sufficient statistics LSS_m, as shown in Figure 1.

Step 4 (Combination): When node m ($m = 1, ..., M$) receives the local sufficient statistics from all of its one-hop neighbor nodes l ($l \in R_m$), it calculates the combined sufficient statistics, shown as:

$$\mathrm{CSS}_m^{(H)}[i] = \sum_{l \in R_m} c_{lm} \cdot \mathrm{LSS}_l^{(H)}[i] \quad H = 1, 2, 3, 4, 5 \tag{23}$$

Step 5 (Estimation): Node m ($m = 1, ..., M$) estimates the parameters of the M*t*FA:

$$\pi_i = \frac{\mathrm{CSS}_m^{(1)}[i]}{\sum_{i'=1}^{I} \mathrm{CSS}_m^{(1)}[i']} \tag{24}$$

$$\boldsymbol{\mu}_i = \frac{\mathrm{CSS}_m^{(3)}[i]}{\mathrm{CSS}_m^{(2)}[i]} \tag{25}$$

$$\mathbf{A}_i = \mathbf{V}_i \mathbf{g}_i (\mathbf{g}_i^{\mathrm{T}} \mathbf{V}_i \mathbf{g}_i + \mathbf{\Omega}_i)^{-1} \tag{26}$$

$$\mathbf{D}_i = \mathrm{diag}\left\{ \mathbf{V}_i - \mathbf{A}_i (\mathbf{g}_i^{\mathrm{T}} \mathbf{V}_i \mathbf{g}_i + \mathbf{\Omega}_i) \mathbf{A}_i^{\mathrm{T}} \right\} \tag{27}$$

where:

$$\mathbf{V}_i = \frac{\mathrm{CSS}_m^{(4)}[i] - 2\mathrm{CSS}_m^{(3)}[i] \cdot \boldsymbol{\mu}_i^{\mathrm{T}} + \mathrm{CSS}_m^{(2)}[i] \cdot \boldsymbol{\mu}_i \boldsymbol{\mu}_i^{\mathrm{T}}}{\mathrm{CSS}_m^{(1)}[i]} \tag{28}$$

In addition, ν_i is updated by solving the following equation:

$$\log\left(\frac{\nu_i}{2}\right) - \psi\left(\frac{\nu_i}{2}\right) + 1 - \frac{\mathrm{CSS}_m^{(5)}[i] - \mathrm{CSS}_m^{(2)}[i]}{\mathrm{CSS}_m^{(1)}[i]} - \log\left(\frac{\nu_i^{\mathrm{old}} + p}{2}\right) + \psi\left(\frac{\nu_i^{\mathrm{old}} + p}{2}\right) = 0 \tag{29}$$

where $\psi(\cdot)$ is the digamma function and ν_i^{old} is the value of ν_i on the last iteration of this algorithm. Equation (29) can be solved by some numerical methods, *i.e.*, the Newton method.

Step 6 (Termination): Node m ($m = 1, ..., M$) calculates its current local log-likelihood $\log p(\mathbf{Y}_l|\boldsymbol{\Theta}^{\text{new}})$, expressed in Equation (21). The superscript "new" denotes the newly-estimated parameters at the current iteration. The termination condition of the algorithm is the same as that in the D-MFA algorithm.

4. Experimental Results

4.1. Synthetic Data

In this subsection, we test the performances of the proposed algorithms on synthetic data. Here, we consider a sensor network composed of 100 nodes to evaluate the estimation performance of the proposed algorithms. Nodes are randomly placed in a square of 5×5. The communication distance is taken as 0.8. In this setting, the connected graph reflecting network topology is shown in Figure 2.

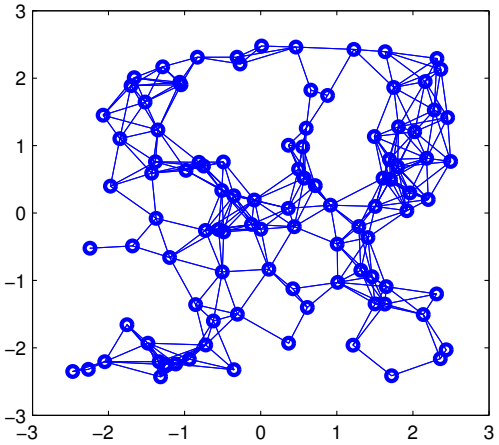

Figure 2. Network connection.

In the first 30 nodes (Node 1–Node 30), each node has 80 observations. In the next 40 nodes (Node 31–Node 70), each node contains 100 observations. In the last 30 nodes (Node 71–Node 100), each node has 120 observations. All of the 10-dimensional observations in the 100 nodes are assumed to be generated from three-component Gaussian mixtures. The parameters are as follows:

$(\pi_1, \pi_2, \pi_3) = (0.3,\ 0.5,\ 0.2)$;
$\boldsymbol{\mu}_1 = (3\ 3\ 3\ 3\ 3\ 0\ 0\ 0\ 0\ 0)$, $\boldsymbol{\mu}_2 = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$,
$\boldsymbol{\mu}_3 = (-3\ -3\ -3\ -3\ -3\ 0\ 0\ 0\ 0\ 0)$;
$\boldsymbol{\Sigma}_1 = \text{diag}(1\ 1\ 1\ 1\ 1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1)$,
$\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_1$, $\boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_1$.

We adopt several models to represent the distributions of these observations, and the task is to estimate parameters in the models. Here, we compare the performance of four schemes. In the first scheme, the standard EM algorithm for the MFA (S-MFA) is implemented in a centralized unit using all observations from 100 nodes. In the second scheme, the D-MFA algorithm proposed in Section 3.2 performs simultaneously in all nodes. In the third scheme, the EM algorithm for the MFA runs in each node using

only local observations of that node. In other words, there is no information exchange among nodes. We abbreviate it as non-cooperation MFA (NC-MFA) for description convenience. In the last scheme, the distributed EM algorithm for the GMM (D-GMM) is implemented. In the D-GMM, the objective function is similar to that of the proposed D-MFA, except that MFA is replaced by GMM. It should be emphasized that the centralized unit is assumed to be always reliable in the S-MFA under ideal conditions. However, this condition is not always fulfilled when the centralized unit fails. Therefore, the S-MFA is seldom adopted in sensor networks. The aim is to test whether the estimation performance of the D-MFA can approach that of the S-MFA.

In the initialization of these MFA schemes, the dimension of factors are set to five. $(\pi_1^0, \pi_2^0, \pi_3^0) = (1/3, 1/3, 1/3)$, $\{\boldsymbol{\mu}_1^0, \boldsymbol{\mu}_2^0, \boldsymbol{\mu}_3^0\}$ are set as randomly-selected observations in the those nodes. The initial elements in $\{\mathbf{D}_1^0, \mathbf{D}_2^0, \mathbf{D}_3^0\}$ and $\{\mathbf{A}_1^0, \mathbf{A}_2^0, \mathbf{A}_3^0\}$ are generated by standardized normal distributions. In order to make the estimation results visible, the principal component analysis is performed for observations, obtaining the two largest eigenvalues and the associated eigenvectors. Then, the observations, the estimated means $\boldsymbol{\mu}_i$ $(i = 1, 2, 3)$ and the covariances $\boldsymbol{\Sigma}_i$ $(\boldsymbol{\Sigma}_i = \mathbf{A}_i\mathbf{A}_i^{\mathrm{T}} + \mathbf{D}_i)$ after the termination of the algorithms can be projected into 2D principal subspace [34]. Figure 3 illustrates the results of the estimated parameters at the 2D principal subspace in these four schemes. In this figure, the estimated mean $\boldsymbol{\mu}_i$ of each component is denoted by "+", and the estimated covariance $\boldsymbol{\Sigma}_i$ is represented by shaded ellipse. Concretely, in Figure 3a, parameters can be correctly estimated by the S-MFA, as the centralized unit can use all of the observations directly. In Figure 3b–d, the results of a randomly-selected node are given. For the NC-MFA, the appropriate parameters are incorrectly estimated, as it can only use its own observations, which also happens in other nodes. For the D-GMM, as it is based on GMM, it cannot describe and process high-dimensional observations well. Finally, in the D-MFA, each node can receive the calculated LSS from nodes in its neighborhood set and combine them for parameter estimation. Compared to GMM, MFA can reflect the properties of these high-dimensional observations more accurately. Therefore, the estimated means and covariances are correct in the D-MFA, as shown in Figure 3b. The other nodes have the same results as this selected node, which are not given here due to space limitation. Moreover, as the same observations and models are used in three MFA schemes, the changes of the average log-likelihood over all nodes in the S-MFA, log-likelihood of the D-MFA and the NC-MFA are shown in different lines in Figure 4. We can see that as the iteration increases, the D-MFA is convergent. Its convergence performance approaches that of the S-MFA.

In order to further show the estimation accuracy of the D-MFA at all of the nodes in a sensor network, we select two kinds of parameters (π_1, π_2, π_3) and $\boldsymbol{\mu}_1$, giving the estimation results of these parameters. In Figure 5, the estimated (π_1, π_2, π_3) in the D-MFA and the NC-MFA at all 100 nodes are provided. In the NC-MFA, each node cannot correctly estimate parameters due to limited observations and no information exchange with other nodes, shown by dashed lines. On the contrary, after the D-MFA converges, the estimated values (π_1, π_2, π_3) in all 100 nodes approach their true values $(0.3,\ 0.5,\ 0.2)$. In Figure 6, we compare all of the vector components in $\boldsymbol{\mu}_1$ estimated by these three MFA schemes. It is noted that for the D-MFA and the NC-MFA, we give the mean and standard deviation of each vector component over 100 nodes to reflect the whole performance of network. We can clearly see that the S-MFA can correctly estimate $\boldsymbol{\mu}_1$, as it can use all of the observations. For the D-MFA, the mean of estimated $\boldsymbol{\mu}_1$ in each vector component approaches the corresponding true value $(3\ 3\ 3\ 3\ 3\ 0\ 0\ 0\ 0\ 0)$,

while the mean of $\boldsymbol{\mu}_1$ obtained by the NC-MFA is not consistent with true value. Moreover, the standard deviation of D-MFA is smaller than that of the NC-MFA. Since the other parameters lead to similar results, we omit them here.

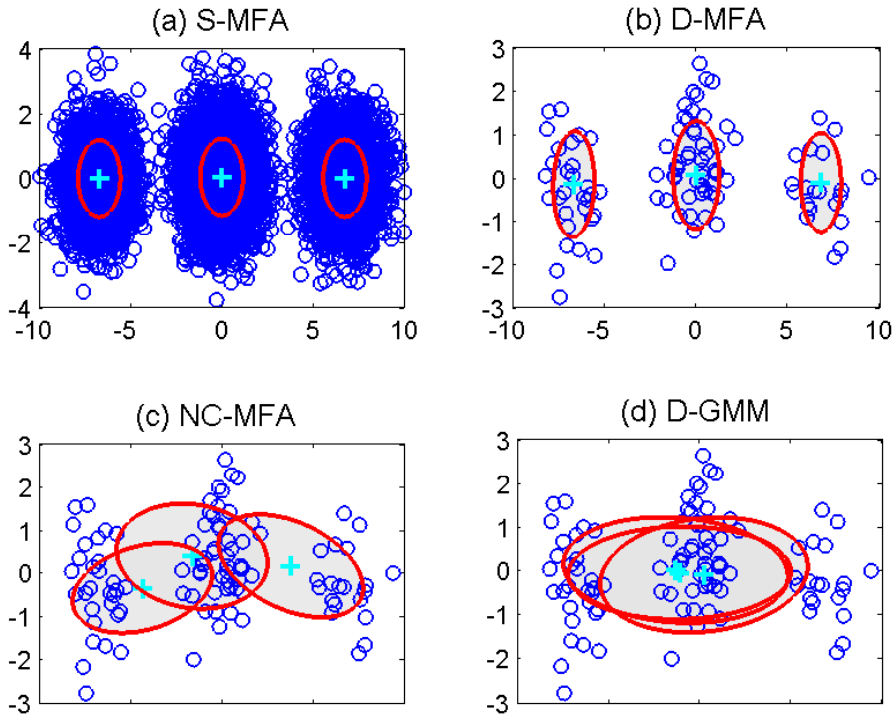

Figure 3. Scatter plot of observations with the estimated parameters at 2D principal subspace using different schemes: (**a**) S-MFA; (**b**) D-MFA; (**c**) NC-MFA; (**d**) D-GMM.

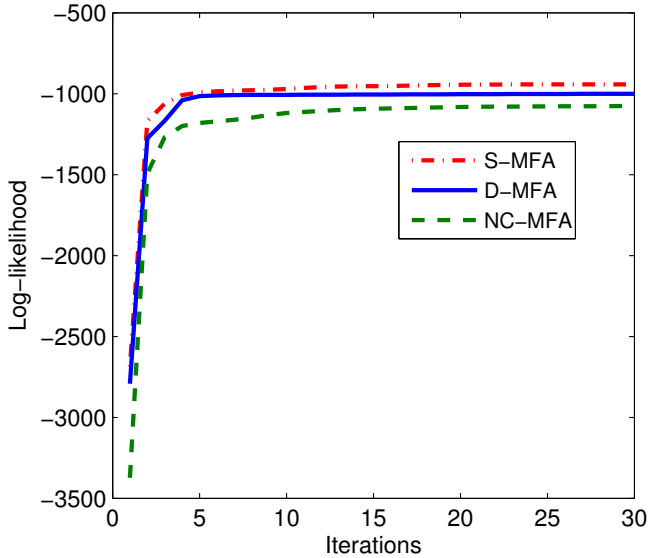

Figure 4. Log-likelihood changes of three MFA schemes during 30 iterations.

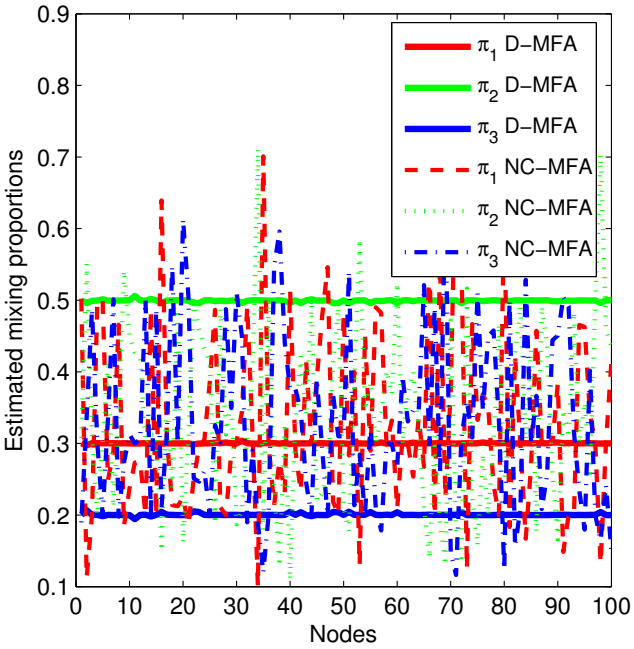

Figure 5. The estimated mixing proportions $(\pi_1,\ \pi_2,\ \pi_3)$ in the D-MFA and the NC-MFA at 100 nodes.

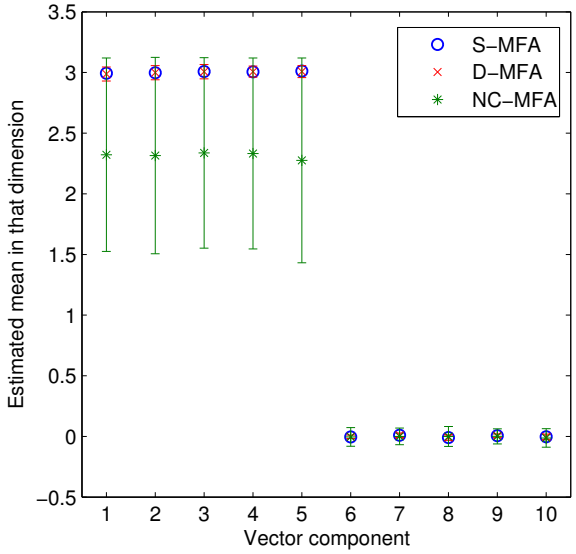

Figure 6. The mean and standard deviation of all of the vector components in estimated $\boldsymbol{\mu}_1$ over 100 nodes.

For the D-M*t*FA algorithm, we test its performance and compare it to the S-M*t*FA, the NC-M*t*FA and the D-*t*MM. It is noted that the S-M*t*FA, the NC-M*t*FA and the D-*t*MM can be realized by replacing Gaussian distributions in the S-MFA, the NC-MFA and the D-GMM with Student's *t*-distributions, respectively. Here, observations are generated by mixtures of Student's *t*-distributions. The parameters $\{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1,2,3}$ are unchanged while $\nu_1 = \nu_2 = \nu_3 = 5$. In Figure 7, the scatter plot of observations with the estimated parameters at the 2D principal subspace is shown. It is noted that for the D-M*t*FA, the NC-M*t*FA and the D-*t*MM, the results of a randomly-selected node are given. From this figure, we can see several observations located out of ordinary regions, which can be taken as outliers. The S-M*t*FA in the centralized unit, shown in Figure 7a, can grasp the distributions, as it can make use

of all observations. On the contrary, the performance of the NC-M*t*FA and the D-*t*MM are bad. The reasons are similar to those of NC-MFA and D-GMM, which have been explained. When implementing the D-M*t*FA, parameters can be accurately estimated, while the property of robustness to outliers is still maintained, as shown in Figure 7b. In summary, the proposed D-MFA and the D-M*t*FA can accurately estimate parameters in a distributed way when each node in sensor network has part of the high-dimensional observations.

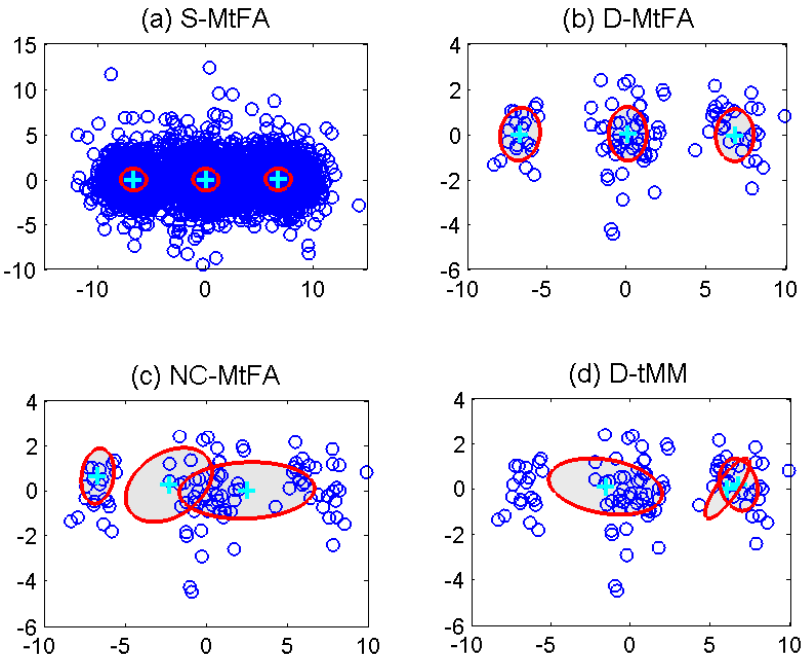

Figure 7. Scatter plot of observations with the estimated parameters at the 2D principal subspace using different schemes: (**a**) S-M*t*FA; (**b**) D-M*t*FA; (**c**) NC-M*t*FA; (**d**) D-*t*MM.

4.2. Real Data

In several countries, there are monitoring sites located in different regions, whose tasks are to detect nutritional ingredients in the wine samples. These sites form a sensor network, in which each site only communicates with its neighbors and can implement local computations. The wine samples sent to these monitoring sites may belong to different cultivars. Therefore, in each monitoring site, these samples need to be classified, which is good for analyzing in-depth the relationship of nutritional ingredients in the wine and their cultivars. It is certain that the more the references are available for each site, the better the results will be. Therefore, the network and cooperation between sites are required.

In this subsection, we consider the wine cultivar clustering problem as a simulation of the above scenario. The database of this problem is the wine dataset, which is one of the most popular datasets in the UCImachine learning repository [35]. In this wine dataset, 178 samples are collected from a chemical analysis of wines grown in three different cultivars in Italy (the No. 1∼No. 59 samples belong to the first class; the No. 60∼No.130 samples belong to the second class; the No. 131∼No. 178 samples belong to the third class). Each sample has 13 attributes, so the dimension of observations is 13. The sensor network is composed of eight nodes, represented as eight monitoring sites. The average number

of nodes in the neighborhood set is two, and the graph is guaranteed to be connected. The average number of samples in each site is 22.

Clustering belongs to the unsupervised learning paradigm in machine learning. When the D-MFA (or the D-M*t*FA) is adopted for clustering, initial values of parameters in the D-MFA are set, which are the same as those in Section 4.1. Then, the corresponding algorithm derived in Section 3.2 (or Section 3.3) is performed. After algorithm converges, an additional computation step based on the estimated parameters $\mathbf{\Theta}$ at node m is carried out to obtain $\langle z_{m,n,i} \rangle$ by Equation (6). Finally, the cluster decision for each observation $\mathbf{y}_{m,n}$ is:

$$\mathcal{C}_{m,n} = \text{argmax}_{i=1}^{I} \langle z_{m,n,i} \rangle. \ m = 1, ...M, n = 1, ..., N_m$$

The clustering results of the D-MFA at Node 1∼Node 8 are shown in Figure 8. In this figure, blue "∘" represents correctly-clustered observations, while red "×" denotes wrongly-clustered ones. From these figures, we can see that the correct ratio in eight nodes are 100%, 100%, 95.2%, 95.5%, 100%, 95.5%, 100%, 92.9%. There are five wrongly-clustered observations in all. The correct ratio in the entire network is 97.2%. In order to compare the performance of the D-MFA with that of the S-MFA, we perform the D-MFA and the S-MFA algorithms 20 times. The average correct ratio of these 20 runs by the D-MFA is 96.9%, approaching that by the S-MFA, which is 98.2%. The reason for the small performance gap between the S-MFA and the D-MFA may be that the number of observations for each node is small and the dimension is relatively high. The accuracy of the calculated LSS in Step 2 of the D-MFA are a little worse than those global sufficient statistics obtained by all of the observations in the E-step of the S-MFA. For the NC-MFA, as the number of observations in this example at each node is small, the clustering cannot be implemented. For the D-GMM, as the dimension of observations is high, it also cannot finish the task of this example effectively. Moreover, as there are no outliers in this dataset, the clustering result of the D-M*t*FA is the same as that of the D-MFA, which is not shown here. In summary, we can use the proposed schemes to realize distributed clustering.

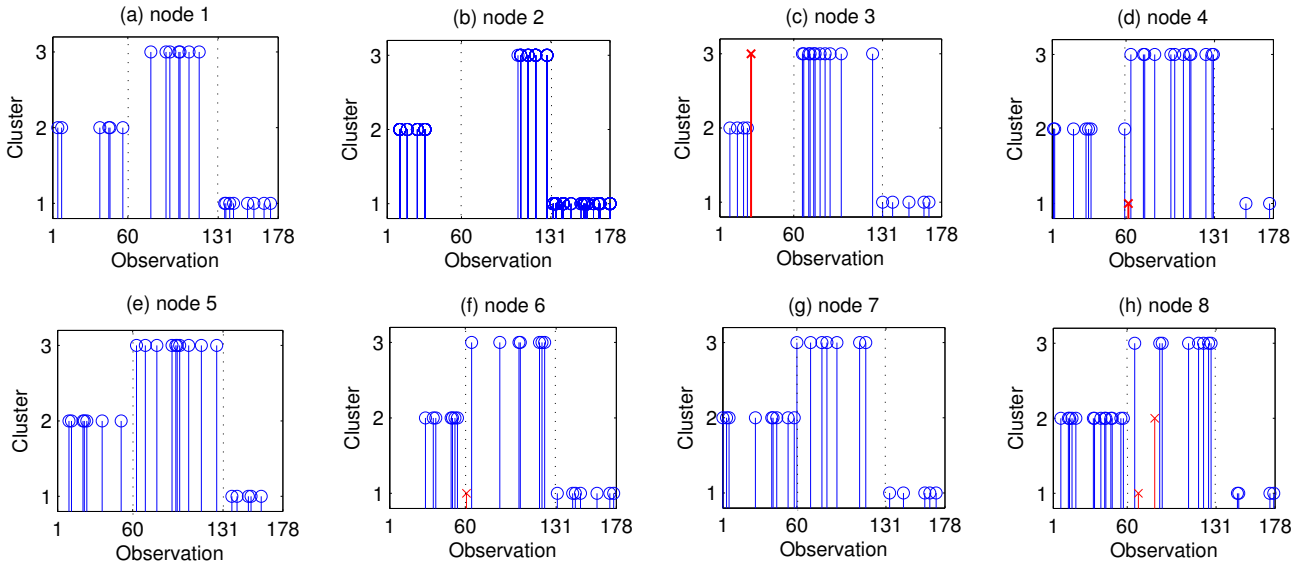

Figure 8. Clustering results of the wine dataset at (**a**–**h**) Node 1∼Node 8.

5. Conclusions

In this paper, we propose a distributed density estimation method base on a mixture of factor analyzers in sensor networks. First, a linear combination of local log-likelihoods associated with nodes in its neighborhood (including itself) is defined as the objective function. In this objective function, the combination coefficients are determined by the number of observations in corresponding nodes. Then, the D-MFA and the D-M*t*FA algorithms are derived. In these algorithms, the combined sufficient statistics of each node are used to estimate parameters, which can be obtained by performing a linear combination of local sufficient statistics from nodes in its neighborhood. Finally, we evaluate the performances of the proposed algorithms and apply them to the tasks of clustering and classification. Experimental results show that they are promising and effective statistical tools for processing the high-dimensional datasets in a distributed way in sensor networks.

In our future work, we will investigate distributed algorithms that can automatically determine the structure of MFA, e.g., the number of components. We also intend to design adaptive strategies to adjust combination coefficients more flexibly. Moreover, the coverage problem [36] in a sensor network is important when implementing distributed algorithms. We will consider this issue in the future.

Acknowledgments

This work is partly supported by the National Natural Science Foundation of China (Grant Nos. 61171153, 61201326, 61273266, 61201165, 61271240, 61322104, 61401228), the State Key Development Program of Basic Research of China (2013CB329005), the Natural Science Fund for Higher Education of Jiangsu Province (Grant Nos. 12KJB510021, 13KJB510020), the Natural Science Foundation of Jiangsu Province (BK20140891), the Priority Academic Program Development of Jiangsu Higher Education Institutions, the Scientific Research Foundation of NUPT(Grant Nos. NY211032, NY211039), the Qing Lan Project, the Zhejiang Provincial Natural Science Foundation of China (Grant No. LR12F01001) and the National Program for Special Support of Eminent Professionals.

Author Contributions

All authors conceived the algorithms and designed the simulations; XinWei wrote the initial research manuscript; The other authors contributed to the revision of the final paper.

Conflicts of Interest

The authors declare no conflict of interest.

Appendix

A. Derivation of the D-M*t*FA Algorithm

In this Appendix, we give the derivation of the D-M*t*FA after defining the objective function $\mathcal{F}_m(\mathbf{\Theta})$ in Section 3.3. After introducing three distributions $q(\mathbf{Z}_l)$, $q(\mathbf{W}_l|\mathbf{Z}_l)$ and $q(\mathbf{U}_l|\mathbf{W}_l,\mathbf{Z}_l)$, the decomposition shown by Equation (5) holds, where:

$$\mathcal{L}(q_l,\mathbf{\Theta})=\sum_{\mathbf{Z}_l}q(\mathbf{Z}_l)\int \mathrm{d}\mathbf{W}_l q(\mathbf{W}_l|\mathbf{Z}_l)\int \mathrm{d}\mathbf{U}_l q(\mathbf{U}_l|\mathbf{W}_l,\mathbf{Z}_l)\times\log\left\{\frac{p(\mathbf{Y}_l,\mathbf{U}_l,\mathbf{W}_l,\mathbf{Z}_l|\mathbf{\Theta})}{q(\mathbf{Z}_l)q(\mathbf{W}_l|\mathbf{Z}_l)q(\mathbf{U}_l|\mathbf{W}_l,\mathbf{Z}_l)}\right\}$$

$$\begin{aligned}\mathrm{KL}(q_l||p_l)=&-\sum_{\mathbf{Z}_l}q(\mathbf{Z}_l)\int \mathrm{d}\mathbf{W}_l q(\mathbf{W}_l|\mathbf{Z}_l)\int \mathrm{d}\mathbf{U}_l q(\mathbf{U}_l|\mathbf{W}_l,\mathbf{Z}_l)\\&\times\log\left\{\frac{p(\mathbf{Z}_l|\mathbf{Y}_l,\mathbf{\Theta})p(\mathbf{W}_l|\mathbf{Z}_l,\mathbf{Y}_l,\mathbf{\Theta})p(\mathbf{U}_l|\mathbf{W}_l,\mathbf{Z}_l,\mathbf{Y}_l,\mathbf{\Theta})}{q(\mathbf{Z}_l)q(\mathbf{W}_l|\mathbf{Z}_l)q(\mathbf{U}_l|\mathbf{W}_l,\mathbf{Z}_l)}\right\}\end{aligned}$$

In the first stage, three conditional distributions are needed to compute. Concretely, $p(z_{l,n,i}|\mathbf{y}_{l,n})$ at node l is:

$$p(z_{l,n,i}|\mathbf{y}_{l,n})=\frac{\pi_i^{\mathrm{old}}t(\mathbf{y}_{l,n}|\boldsymbol{\mu}_i^{\mathrm{old}},\mathbf{A}_i^{\mathrm{old}}(\mathbf{A}_i^{\mathrm{old}})^{\mathrm{T}}+\mathbf{D}_i^{\mathrm{old}},\nu_i^{\mathrm{old}})}{\sum_{i'=1}^{I}\pi_{i'}^{\mathrm{old}}t(\mathbf{y}_{l,n}|\boldsymbol{\mu}_{i'}^{\mathrm{old}},\mathbf{A}_{i'}^{\mathrm{old}}(\mathbf{A}_{i'}^{\mathrm{old}})^{\mathrm{T}}+\mathbf{D}_{i'}^{\mathrm{old}},\nu_{i'}^{\mathrm{old}})}\tag{A1}$$

The conditional distribution of $w_{l,n,i}$ given $z_{l,n,i}$ and $\mathbf{y}_{l,n}$ is obtained:

$$p(w_{l,n,i}|z_{l,n,i},\mathbf{y}_{l,n})=\mathcal{G}\left(w_{l,n,i}\middle|\frac{\nu_i^{\mathrm{old}}+p}{2},\frac{\nu_i^{\mathrm{old}}+\mathcal{W}_{l,n,i}}{2}\right)\tag{A2}$$

where:

$$\mathcal{W}_{l,n,i}=(\mathbf{y}_{l,n}-\boldsymbol{\mu}_i^{\mathrm{old}})^{\mathrm{T}}\left[\mathbf{A}_i^{\mathrm{old}}(\mathbf{A}_i^{\mathrm{old}})^{\mathrm{T}}+\mathbf{D}_i^{\mathrm{old}}\right](\mathbf{y}_{l,n}-\boldsymbol{\mu}_i^{\mathrm{old}})$$

The expectations $\langle z_{l,n,i}\rangle$ are given by Equation (A1), and $\langle w_{l,n,i}\rangle$ can be obtained from Equation (A2), which is:

$$\langle w_{l,n,i}\rangle=\frac{\nu_i^{\mathrm{old}}+p}{\nu_i^{\mathrm{old}}+\mathcal{W}_{l,n,i}}$$

The conditional distribution of $\mathbf{u}_{l,n}$ given $w_{l,n,i}$, $z_{l,n,i}$ and $\mathbf{y}_{l,n}$ can be computed:

$$p(\mathbf{u}_{l,n}|w_{l,n,i},z_{l,n,i},\mathbf{y}_{l,n})=N(\mathbf{u}_{l,n}|\overline{\mathbf{u}}_{l,n,i},\mathbf{\Omega}_i/\langle w_{l,n,i}\rangle)$$

The mean $\overline{\mathbf{u}}_{l,n,i}$ and $\mathbf{\Omega}_i$ are:

$$\begin{aligned}\overline{\mathbf{u}}_{l,n,i}&=\mathbf{g}_i^{\mathrm{T}}(\mathbf{y}_{l,n}-\boldsymbol{\mu}_i^{\mathrm{old}})\\\mathbf{\Omega}_i&=\mathbf{I}_q-\mathbf{g}_i^{\mathrm{T}}\mathbf{A}_i^{\mathrm{old}}\end{aligned}$$

where:

$$\mathbf{g}_i=\left[\mathbf{A}_i^{\mathrm{old}}(\mathbf{A}_i^{\mathrm{old}})^{\mathrm{T}}+\mathbf{D}_i^{\mathrm{old}}\right]^{-1}\cdot\mathbf{A}_i^{\mathrm{old}}$$

is an intermediate variable introduced to simplify expressions in the following stage. Moreover, the expectations $\langle\mathbf{u}_{l,n}\rangle$ and $\langle\mathbf{u}_{l,n}\mathbf{u}_{l,n}^{\mathrm{T}}\rangle$ are:

$$\begin{aligned}\langle\mathbf{u}_{l,n}\rangle&=\overline{\mathbf{u}}_{l,n,i}\ \text{ and}\\\langle\mathbf{u}_{l,n}\mathbf{u}_{l,n}^{\mathrm{T}}\rangle&=\mathbf{\Omega}_i\Big/\langle w_{l,n,i}\rangle+\overline{\mathbf{u}}_{l,n,i}\overline{\mathbf{u}}_{l,n,i}^{\mathrm{T}}\end{aligned}$$

Similar to the D-MFA, when the first stage finishes, the current lower bound is:

$$\mathcal{Q}_m(\boldsymbol{\Theta}) = \sum_{l \in R_m} c_{lm} \sum_{n=1}^{N_l} \sum_{i=1}^{I} p(z_{l,n,i}|\mathbf{y}_{l,n}) p(w_{l,n,i}|\mathbf{y}_{l,n}, z_{l,n,i}) p(\mathbf{u}_{l,n}|\mathbf{y}_{l,n}, z_{l,n,i}, w_{l,n,i}) \times \left\{ z_{l,n,i} \log \left[\pi_i N(\mathbf{u}_{l,n}|\mathbf{0}, \mathbf{I}_q) \mathcal{G}(w_{l,n,i}|\nu_i/2, \nu_i/2) N(\mathbf{y}_{l,n}|\boldsymbol{\mu}_i + \mathbf{A}_i \mathbf{u}_{l,n}, \mathbf{D}_i) \right] \right\}$$

Now, at node m, parameters can be obtained by taking derivation of $\mathcal{Q}_m(\boldsymbol{\Theta})$ with respect to $\boldsymbol{\Theta}$. First, we obtain:

$$\pi_i = \frac{\sum_{l \in R_m} c_{lm} \sum_{n=1}^{N_l} \langle z_{l,n,i} \rangle}{\sum_{i'=1}^{I} \sum_{l \in R_m} c_{lm} \sum_{n=1}^{N_l} \langle z_{l,n,i'} \rangle} \tag{A3}$$

and:

$$\boldsymbol{\mu}_i = \frac{\sum_{l \in R_m} c_{lm} \sum_{n=1}^{N_l} \langle z_{l,n,i} \rangle \cdot \langle w_{l,n,i} \rangle \mathbf{y}_{l,n}}{\sum_{l \in R_m} c_{lm} \sum_{n=1}^{N_l} \langle z_{l,n,i} \rangle \cdot \langle w_{l,n,i} \rangle} \tag{A4}$$

respectively. By substituting Equations (22) and (23) into Equations (A3) and (A4), we can obtain Equations (24) and (25).

Subsequently, by respectively performing derivation of $\mathcal{Q}_m(\boldsymbol{\Theta})$ with respect to $\mathbf{A}_i$ and $\mathbf{D}_i$, we have:

$$\mathbf{A}_i = \frac{\sum_{l \in R_m} c_{lm} \sum_{n=1}^{N_l} \langle z_{l,n,i} \rangle \cdot \langle w_{l,n,i} \rangle (\mathbf{y}_{l,n} - \boldsymbol{\mu}_i) \langle \mathbf{u}_{l,n} \rangle^{\mathrm{T}}}{\sum_{l \in R_m} c_{lm} \sum_{n=1}^{N_l} \langle z_{l,n,i} \rangle \cdot \langle w_{l,n,i} \rangle \cdot \langle \mathbf{u}_{l,n} \mathbf{u}_{l,n}^{\mathrm{T}} \rangle} \tag{A5}$$

and:

$$\mathbf{D}_i = \mathrm{diag} \left\{ \frac{\sum_{l \in R_m} c_{lm} \sum_{n=1}^{N_l} \langle z_{l,n,i} \rangle \langle w_{l,n,i} \rangle \left[-\mathbf{A}_i \langle \mathbf{u}_{l,n} \mathbf{u}_{l,n}^{\mathrm{T}} \rangle \mathbf{A}_i^{\mathrm{T}} (\mathbf{y}_{l,n} - \boldsymbol{\mu}_i)(\mathbf{y}_{l,n} - \boldsymbol{\mu}_i)^{\mathrm{T}} \right]}{\sum_{l \in R_m} c_{lm} \sum_{n=1}^{N_l} \langle z_{l,n,i} \rangle} \right\} \tag{A6}$$

By substituting $\langle z_{l,n,i} \rangle$, $\langle w_{l,n,i} \rangle$, $\langle \mathbf{u}_{l,n} \rangle$, $\langle \mathbf{u}_{l,n} \mathbf{u}_{l,n}^{\mathrm{T}} \rangle$ into Equations (A5) and (A6) and by considering the simplified expressions in Equations (22), (23) and (28), we can obtain Equations (26) and (27).

Finally, by performing derivation of $\mathcal{Q}_m(\boldsymbol{\Theta})$ with respect to ν_i, the update equation for ν_i is obtained, shown by Equation (29).

References

1. Akyildiz, I.; Su, W.; Sankarasubramniam, Y. A survey on sensor networks. *IEEE Commun. Mag.* **2002**, *40*, 102–114.
2. Sayed, A.H.; Tu, S.Y.; Chen, J.; Zhao, X.; Towfic, Z. Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior. *IEEE Signal Process. Mag.* **2013**, *30*, 155–171.
3. Poza-Lujan, J.; Posadas-Yagüe, J.; Simó-Ten, J.; Simarro, R.; Benet, G. Distributed sensor architecture for intelligent control that supports quality of control and quality of service. *Sensors* **2015**, *15*, 4700–4733.
4. Chen, J.; Sayed, A.H. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Trans. Signal Process.* **2012**, *60*, 4289–4305.

5. Cattivelli, F.S.; Sayed, A.H. Diffusion LMS strategies for distributed estimation. *IEEE Trans. Signal Process.* **2010**, *58*, 1035–1048.
6. Meteos, G.; Giannakis, G.B. Distributed recursive least-squares: Stability and performance analysis. *IEEE Trans. Signal Process.* **2012**, *60*, 3740–3754.
7. Cao, M.; Meng, Q.; Zeng, M.; Sun, B.; Li, W.; Ding, C. Distributed least-squares estimation of a remote chemical source via convex combination in wireless sensor networks. *Sensors* **2014**, *14*, 11444–11466.
8. Cao, L.; Xu, C.; Shao, W.; Zhang, G.; Zhou, H.; Sun, Q.; Guo, Y. Distributed power allocation for sink-centric clusters in multiple sink wireless sensor networks. *Sensors* **2010**, *10*, 2003–2026.
9. Lorenzo, P.D.; Sayed, A.H. Sparse distributed learning based on diffusion adaption. *IEEE Trans. Signal Process.* **2013**, *61*, 1419–1433.
10. Liu, Z.; Liu, Y.; Li, C. Distributed sparse recursive least-squares over networks. *IEEE Trans. Signal Process.* **2014**, *62*, 1385–1395.
11. Li, C.; Shen, P.; Liu, Y.; Zhang, Z. Diffusion information theoretic learning for distributed estimation over network. *IEEE Trans. Signal Process.* **2013**, *61*, 4011–4024.
12. Gu, D.; Hu, H. Spatial Gausssian process regression with mobile sensor networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1279–1290.
13. Dempster, A.P.; Laird, N.M.; Robin, D.B. Maximum likelihood estimation from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 1–38.
14. McLachlan, G.J.; Krishnan, T. *The EM Algorithm and Extensions*, 2nd ed.; Wiley: New York, NY, USA, 2008.
15. McLachlan, G.J.; Peel, D. *Finite Mixture Models*; Wiley: New York, NY, USA, 2000.
16. Ghahramani, Z.; Hinton, G.E. *The EM Algorithm for Mixtures of Factor Analyzers*; Tech. Rep. CRG-TR-96-1; Department of Computer Science, University of Toronto: Toronto, ON, USA, 1997.
17. Zhao, J.; Yu, P.L.H. Fast ML estimation for the mixture of factor analyzers via an ECM algorithm. *IEEE Trans. Neural Netw*. **2008**, *19*, 1956–1961.
18. McLachlan, G.J.; Bean, R.W.; Jones, L.B. Extension of the mixture of factor analyzers model to incorporate the multivariated *t*-distribution. *Comput. Stat. Data Anal.* **2007**, *51*, 5327–5338.
19. Andrews, J.L.; McNicholas, P.D. Mixtures of modified *t*-factor analyzers for model-based clustering, classification, and discriminant analysis. *J. Stat. Plan. Infer.* **2011**, *141*, 1479–1486.
20. Carin, L.; Baraniuk, R.G.; Cevher, V.; Dunson, D.; Jordan, M.I.; Sapiro, G.; Wakin, M.B. Learning low-dimensional signal models. *IEEE Signal Process. Mag.* **2011**, *28*, 39–51.
21. Li, R.; Tian, T.P.; Sclaroff, S. 3D human motion tracking with a coordinated mixture of factor analyzers. *Int. J. Comput. Vis.* **2010**, *87*, 1–2.
22. Wu, Z.; Kinnunen, T.; Chng, E.S. Mixture of factor analyzers using priors from non-parallel speech for voice conversion. *IEEE Signal Process. Lett.* **2012**, *19*, 914–917.
23. Baek, J.; McLachlan, G.J. Mixtures of common t-factor analyzers for clustering high-dimensional microarray data. *Bioinformatics* **2011**, *21*, 1269–1276.
24. Wei, X.; Li, C. Bayesian mixtures of common factor analyzers: Model, variational inference, and applications. *Signal Process.* **2013**, *93*, 2894–2904.

25. Nowak, R.D. Distributed EM algorithms for density estimation and clustering in sensor neworks. *IEEE Trans. Signal Process.* **2003**, *51*, 2245–2253.
26. Safarinejadian, B.; Menhaj, M.B.; Karrari, M. Distributed variational Bayesian algorithms for Gaussian mixtures in sensor networks. *Signal Process.* **2010**, *90*, 1197–1208.
27. Gu, D. Distributed EM algorithm for Gaussian mixtures in sensor networks. *IEEE Trans. Neural Netw.* **2008**, *19*, 1154–1166.
28. Safarinejadian, B.; Menhaj, M.B.; Karrari, M. Distributed unsupervised Gaussian mixture learning for density estimation in sensor networks. *IEEE Trans. Instrum. Meas.* **2010**, *59*, 2250–2260.
29. Pereira, S.S.; Valcarce, R.L.; Zamora, A.P. A diffusion-based EM algorithm for distributed estimation in unreliable sensor networks. *IEEE Signal Process. Lett.* **2013**, *20*, 595–598.
30. Pereira, S.S.; Zamora, A.P.; Valcarce, R.L. A diffusion-based distributed EM algorithm for density estimation in wireless sensor networks. In Proceedings of the International Conference on Acoustics Speech, and Signal Processing (ICASSP), Vancouver, BC, Canada, 26–31 May 2003; pp. 4449–4453.
31. Towfic, Z.J.; Chen, J.; Sayed, A.H. Collaborative learning of mixture models using diffusion adaptation. In Proceedings of the 2011 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Santander, Spain, 18–21 September 2011; pp. 1–6.
32. Weng, Y.; Xiao, W.; Xie, L. Diffusion-based EM algorithm for distributed estimation of Gaussian mixtures in wireless sensor networks. *Sensors* **2011**, *11*, 6297–6316.
33. Pascal, B.; Gersende, F.; Walid, H. Performance of a distributed stochastic approximation algorithm. *IEEE Trans. Inf. Theory* **2013**, *59*, 7405–7418.
34. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
35. UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science, 2013 [online]. Available online: http://archive.ics.uci.edu/ml (accessed on 16 April 2015)
36. Katsuma, R.; Murata, Y.; Shibata, N.; Yasumoto, K.; Ito, M. Extending k-coverage lifetime of wireless sensor networks using mobile sensor nodes. In Proceedings of the 5th Annual IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Marrakech, Morocco, 12–14 October 2009; pp. 48–54.