IEICE TRANS. INF. & SYST., VOL.E99–D, NO.12 DECEMBER 2016

PAPER Special Section on Parallel and Distributed Computing and Networking

A Mobile Agent Based Distributed Variational Bayesian Algorithm for Flow and Speed Estimation in a Traffic System

Mohiyeddin MOZAFFARI[†a)], *Student Member and* **Behrouz SAFARINEJADIAN**[†b)], *Nonmember*

SUMMARY This paper provides a mobile agent based distributed variational Bayesian (MABDVB) algorithm for density estimation in sensor networks. It has been assumed that sensor measurements can be statistically modeled by a common Gaussian mixture model. In the proposed algorithm, mobile agents move through the routes of the network and compute the local sufficient statistics using local measurements. Afterwards, the global sufficient statistics will be updated using these local sufficient statistics. This procedure will be repeated until convergence is reached. Consequently, using this global sufficient statistics the parameters of the density function will be approximated. Convergence of the proposed method will be also analytically studied, and it will be shown that the estimated parameters will eventually converge to their true values. Finally, the proposed algorithm will be applied to one-dimensional and two dimensional data sets to show its promising performance.

key words: *density estimation, Gaussian mixture model, mobile agent, sensor networks, variational Bayesian*

1. Introduction

Sensor networks are composed of several intelligent sensor nodes, which are distributed over an environment and collect the required information. This kind of networks has a wide range of application in various fields such as industrial productions, environmental monitoring, automatic transportation systems, and traffic control [1]. Parameter estimation in a sensor network using local computation and information exchange between neighbor nodes is called distributed estimation. Unlike the centralized estimation, in distributed estimation it is not necessary to send local observations collected by all the sensors to a powerful central node. As a result the complexity and resource consumption will be reduced, and distributed estimation is more flexible and robust to node and/or link failure [2], [3]. Recently, many distributed estimation algorithms have been proposed, such as distributed recursive least square (RLS) [4], distributed least-mean square (LMS) [5], distributed sparse estimation [6], [7], distributed expectation maximization (EM) algorithm [8], distributed Gaussian process regression [9], and distributed variational Bayesian (VB) algorithm [10].

The EM algorithm is a powerful method for density estimation and data clustering in sensor networks [11]. Distributed EM (DEM) algorithm for data clustering and density estimation assuming Gaussian mixture model and finite mixture of component for measurements can be found in [12]–[16]. In addition, convergence analysis of DEM algorithm has been done in [14]. Computation of maximum-likelihood estimate in presence of incomplete observation using EM algorithm has been studied in [17]. Diffusion based EM algorithm for distributed density estimation in sensor networks using noisy observations has been found in [18], [19]. The parameters of a Gaussian mixture model can be estimated using the EM algorithm, while in the Variational Bayesian algorithm, a probability density function can be obtained for each of the parameters. In EM algorithm, not only the number of components of the Gaussian mixture model cannot be estimated but also when the model order is greater than the real order of the observed data in EM algorithm, singularity in the estimated parameters may be happened, which is not desired. On the other hand, using the VB algorithm the order of the mixture model can be also estimated. Therefore, the VB algorithm provides complete information for each parameter.

Recently, variational methods have been used in parameter estimation especially in finite mixture models. As it mentioned, one of the advantages of the variational based method is the ability to estimate the order of mixture model. A distributed variational Bayesian algorithm (DVBA) for data clustering and density estimation in sensor networks was proposed in [20], [21]. Measurements of the sensors in these references are modeled by a common Gaussian mixture model. The convergence analysis of the mentioned DVBA was also investigated using variational free energy. A variational Bayesian estimate for normal mixture model using an iterative approach was studied in [22]. In addition, it was shown that the variational Bayesian estimator converged to the maximum likelihood estimator in especial conditions at rate $O(1/N)$ as the sample size N goes to infinity. The aforementioned DVBA require that the global summary quantities be transmitted through the entire network. Furthermore, since the algorithm needs to access all nodes in each updating step, its fault tolerance is poor. In traditional client/server based scheme in sensor networks, all the clients should send their observation to the server for data fusion, which results in a high network traffic. In this scheme if any fault happens in the server, the network will not work properly. To solve this problem in sensor networks, mobile agents have been proposed [23], [24].

Intelligent transport systems have been attracting increasing research interest. This is evidenced by recent de-

Manuscript received December 31, 2015.
Manuscript revised May 2, 2016.
Manuscript publicized August 24, 2016.

[†]The authors are with the Electrical Engineering Department, Shiraz University of Technology, Shiraz, Iran.
a) E-mail: m.mozaffari@sutech.ac.ir
b) E-mail: safarinejad@sutech.ac.ir

DOI: 10.1587/transinf.2016PAP0002

velopments from both automobile industry and the wireless communication that make the vehicles more and more intelligent. Road traffic density estimation provides important information in intelligent transport systems for road planning, intelligent road routing, and road traffic control. Traffic data estimation as a part of intelligent transportation systems (ITS) provides a better traffic handling. The performance of ITS technology is strongly related to the accessibility of exact and on-time estimates of traffic information [25]. Traffic sensor networks consist of distributed devices which monitor traffic system environment, track vehicles and do a broad range of applications such as collaborative processing of information, and gathering data from distributed sources. The mentioned networks have some limited sensing, processing and communication capabilities. For example, Real-time traffic data including mean speed, traffic density and traffic flow are usually gathered by such distributed sensor networks in distributed transportation systems. Assume that the data set distributed over a traffic sensor network is modeled by a Gaussian mixture model. Here, a mobile-agent-based distributed variational Bayesian (MABDVB) algorithm will be proposed for density estimation and data clustering in mentioned sensor network. In MABDVB algorithm, agents move in different routes and execute the DVB algorithm. These agents then carry the required information to the fusion center to be combined. Then, the agents carry the resulted fusion data to sensor nodes to be updated. This procedure will be repeated until a convergence criterion is met. At this moment, the information of each node can be clustered based on the resulting GMM. Decrease in Computational load and required memory of the fusion center is one of the main advantages of the proposed algorithm. Another advantage is decrease in iterations of the VB algorithm and increase in the convergence rate.

The rest of the paper is organized as follows. In Sect. 2, the main problem formulation is given. Section 3 introduces the variational Bayesian algorithm for mixture density estimation, and a distributed algorithm for its implementation is presented. Section 4 develops MABDVB algorithm. Section 5 is devoted to convergence analysis of MABDVB algorithm. The results of simulated and real data sets are presented in Sect. 6. Finally, Sect. 7 concludes the paper.

2. Problem Formulation

Consider a network of M sensors and a d-dimensional random vector $\mathbf{Y}_m$ with probability density function $f(y_m)$ which corresponds to sensor node m. Each data (measurement) $y_{m,i}$ of node m is a realization of the random vector $\mathbf{Y}_m$. The environment is assumed to be a finite mixture setting with J mixture probabilities $\alpha_{m,j}, (j = 1, \ldots, J)$.

$$f(y_m; \alpha_m, \psi) = \sum_{j=1}^{J} \alpha_{m,j} f_j\left(y_m; \psi_j\right) \tag{1}$$

where $\left\{\alpha_{m,j}\right\}_{j=1}^{J}$, ψ_j, and J are the mixture probabilities at node m, the set of parameters defining the j'th component, and the number of mixture components, respectively. The mixture probabilities $\left\{\alpha_{m,j}\right\}$ may be changed at different nodes while the parameters ψ_j are the same throughout the network. The set of data points of the m'th node is represented by $\mathcal{Y}_m = \{y_{m,i}\}_{i=1}^{N_m}$ where N_m is the number of observations at node m. it is worth noting that measurements of each node are assumed to be independent and identically distributed.

Consider set of missing variables is shown by $\mathcal{Z}_m = \{z_{m,i}\}$ corresponding to $\mathcal{Y}_m = \{y_{m,i}\}$. Each component of $z_{m,i} = \left[z_{m,i}^1, \ldots, z_{m,i}^J\right]$ represents a binary vector indicating by which component the data $y_{m,i}$ is produced. In another words, $y_{m,i}$ is produced by the j'th component of the mixture if $z_{m,i}^r = 0$ for all $r \neq j$ and $z_{m,i}^j = 1$. The pair $x_{m,i} = (y_{m,i}, z_{m,i})$ is called the complete data and we write $\mathcal{X}_m = \{\mathcal{Y}_m, \mathcal{Z}_m\}$ where $\mathcal{X}_m = \{x_{m,i}\}$.

While our approach can be applied to arbitrary models, for simplicity we consider here Gaussian component distributions $f_j\left(y_m; \psi_j\right) = \mathcal{N}\left(y_m; \mu_j, T_j\right)$ where μ_j is the mean and T_j the precision (inverse covariance) matrix. In this paper, a distributed variational Bayesian algorithm is developed to estimate the Gaussian mixture model parameters using the data set $\mathcal{Y} = \{\mathcal{Y}_m\}_{m=1}^{M}$.

3. The Variational Bayesian Algorithm

Suppose we have observed the data set $\mathcal{Y} = \{y_i\}$. Here, it is assumed that there is a parametric model with parameters ψ, and some missing or unobserved values are denoted by $\mathcal{Z} = \{z_i\}$. For a given Y, the posterior distribution of ψ conditioned to Y will be of interest. The main idea of the variational approximation is to approximate the joint conditional density of ψ and Z, by a more tractable distribution $\hat{f}(z, \psi)$, by minimizing the Kullback-Leibler (KL) divergence between the approximating density $\hat{f}(z, \psi)$ and the true joint conditional density $f(z, \psi|y)$. The variational approximation will tend to a tight lower bound on the log likelihood $\mathcal{L} = \log f(y)$. A lower bound for $\mathcal{L}$ can be written as follows:

$$\begin{aligned}
\log f(y) &= \log \int \sum_{z} f(y, z, \psi)\, d\psi \\
&= \log \int \sum_{z} \hat{f}(z, \psi) \frac{f(y, z, \psi)}{\hat{f}(z, \psi)} d\psi \\
&\geq \int \sum_{z} \hat{f}(z, \psi) \log \frac{f(y, z, \psi)}{\hat{f}(z, \psi)} d\psi
\end{aligned} \tag{2}$$

The last term was obtained by Jensen's inequality. This lower bound is denoted by $\mathcal{F}$, that represents the variational free energy.

3.1 Variational Method for Mixtures of Gaussian Distributions

In this section the variational Bayesian method will be applied to estimate the density of a Gaussian mixture model.

Consider a mixture of J Gaussian distributions with unknown means, variances and mixing weights as:

$$f(y;\alpha,\psi)=\sum_{j=i}^{J}\alpha_j\mathcal{N}\left(y;\mu_j,T_j\right) \tag{3}$$

The mixing weights are assigned a Dirichlet prior distribution

$$f_\alpha(\alpha)=Dir\left(\alpha;a_1^0,\ldots,a_J^0\right) \tag{4}$$

where

$$Dir\left(\alpha;a_1^0,\ldots,a_J^0\right)=\frac{1}{B(a^0)}\prod_{i=1}^{J}\alpha_i^{a_i^0-1} \tag{5}$$

In the right hand side of the above equation the multinomial beta function can be expressed in terms of the gamma function as follows

$$B(a)=\frac{\prod_{i=1}^{J}\Gamma(a_i)}{\Gamma\left(\sum_{i=1}^{J}a_i\right)},\quad a=(a_1,\ldots,a_J) \tag{6}$$

Distribution of the means are based on independent multivariate normal conjugate priors, conditional to the covariance matrices, as follows

$$f_{\mu|T}(\mu|T)=\prod_{j=1}^{J}\mathcal{N}\left(\mu_j;m_j^0,\left(b_j^0T_j\right)^{-1}\right) \tag{7}$$

where $\mu=(\mu_1,\ldots,\mu_J)$ and $T=(T_1,\ldots,T_J)$. And distribution of the precision matrices are given using independent Wishart prior distributions,

$$f_T(T)=\prod_{j=1}^{J}\mathcal{W}\left(T_j;\upsilon_j^0,\Sigma_j^0\right) \tag{8}$$

where

$$\begin{aligned}&\mathcal{W}\left(T_j;\upsilon_j^0,\Sigma_j^0\right)\\&=\frac{\left|T_j\right|^{\frac{1}{2}\left(\upsilon_j^0-d-1\right)}\exp tr\left\{-\frac{1}{2}\Sigma_j^0T_j\right\}}{2^{\frac{\upsilon_j^0d}{2}}\pi^{\frac{d(d-1)}{4}}\left|\Sigma_j^0\right|^{-\frac{n}{2}}\prod_{i=1}^{d}\Gamma\left(\frac{1}{2}\left(\upsilon_j^0+1-i\right)\right)}\end{aligned} \tag{9}$$

As a result, the joint distribution of all of the random variables is given by

$$f(y,z,\psi)=f(y,z|\psi)\,f(\alpha)\,f(\mu|T)\,f(T) \tag{10}$$

The quantities a_j^0, m_j^0, b_j^0, υ_j^0 and Σ_j^0 are called hyperparameters. Using the variational approximations, the posterior distributions are

$$\hat{f}_\alpha(\alpha)=Dir\left(\alpha;a_1,\ldots,a_j\right) \tag{11}$$

$$\hat{f}_{\mu|T}(\mu|T)=\prod_{j=1}^{J}\mathcal{N}\left(\mu_j;m_j,\left(b_jT_j\right)^{-1}\right) \tag{12}$$

$$\hat{f}_T(T)=\prod_{j=1}^{J}\mathcal{W}\left(T_j;\upsilon_j,\Sigma_j\right) \tag{13}$$

where the hyperparameters given by

$$a_j=a_j^{(0)}+\sum_{i=1}^{N}\Phi_{i,j} \tag{14}$$

$$b_j=b_j^{(0)}+\sum_{i=1}^{N}\Phi_{i,j} \tag{15}$$

$$m_j=\frac{b_j^{(0)}m_j^{(0)}+\sum_{i=1}^{N}\Phi_{i,j}y_i}{b_j} \tag{16}$$

$$\Sigma_j=\Sigma_j^{(0)}+\sum_{i=1}^{N}\Phi_{i,j}y_iy_i^T+b_j^{(0)}m_j^{(0)}m_j^{(0)^T}-b_jm_jm_j^T \tag{17}$$

$$\upsilon_j=\upsilon_j^{(0)}+\sum_{i=1}^{N}\Phi_{i,j} \tag{18}$$

where

$$\Phi_{i,j}=\frac{\varphi_{i,j}}{\sum_{j=1}^{J}\varphi_{i,j}} \tag{19}$$

$$\varphi_{i,j}=\tilde{\alpha}_j\tilde{T}_j^{1/2}\,e^{-\frac{1}{2}(y_i-m_j)^TE[T_j](y_i-m_j)-\frac{d}{(2b_j)}} \tag{20}$$

$$\tilde{\alpha}_j=\exp\left[\int\hat{f}(\alpha)\log\alpha_j d\alpha\right] \tag{21}$$

$$\tilde{T}_j=\exp\left[\int\hat{f}\left(T_j\right)\log\left|T_j\right|dT_j\right] \tag{22}$$

$$E\left[T_j\right]=\upsilon_j\Sigma_j^{-1} \tag{23}$$

It is worth noting that if the previous procedure is done using only the local data (measurements) of a node, the resulting method is called a standard VB algorithm. On the other hand, in a centralized VB algorithm, all of the nodes of the sensor network send their raw data (measurements) to a center node where the VB algorithm (the previous procedure) is executed.

3.2 Distributed Variational Bayesian Algorithm

In this subsection, observed data is assumed to be distributed in a network with M nodes using a finite mixture of components. Since the observed data is distributed in M different nodes, first we define a vector of sufficient statistics as:

$$S_j^t=\left\{GSS1_j^t,GSS2_j^t,GSS3_j^t\right\} \tag{24}$$

where j and t are the component and iteration number, respectively. The global sufficient statistics are defined as follows

$$GSS1_j^t\triangleq\sum_{m=1}^{M}\sum_{i=1}^{N_m}SS1_{m,i,j}^t \tag{25}$$

$$GSS2_j^t\triangleq\sum_{m=1}^{M}\sum_{i=1}^{N_m}SS1_{m,i,j}^ty_{m,i} \tag{26}$$

$$GSS3_j^t \triangleq \sum_{m=1}^{M}\sum_{i=1}^{N_m} SS1_{m,i,j}^t y_{m,i} y_{m,i}^T \tag{27}$$

where the $SS1_{m,i,j}^t$ is the local quantity calculated at node m and iteration t as follows

$$SS1_{m,i,j}^t = \frac{\varphi_{m,i,j}^t}{\sum_{j=1}^{J}\varphi_{m,i,j}^t} \tag{28}$$

$$\varphi_{m,i,j}^t = \tilde{\alpha}_j^{t-1}\left(\tilde{T}_j^{t-1}\right)^{1/2} e^{-\frac{1}{2}\left(y_{m,i}-m_j^{t-1}\right)^T E\left[T_j^{t-1}\right]\left(y_{m,i}-m_j^{t-1}\right)-\frac{d}{\left(2b_j^{t-1}\right)}} \tag{29}$$

At node m, the local summary statistics can be computed using:

$$LSS1_{m,j}^t = \sum_{i=1}^{N_m} SS1_{m,i,j}^t \tag{30}$$

$$LSS2_{m,j}^t = \sum_{i=1}^{N_m} SS1_{m,i,j}^t y_{m,i} \tag{31}$$

$$LSS3_{m,j}^t = \sum_{i=1}^{N_m} SS1_{m,i,j}^t y_{m,i} y_{m,i}^T \tag{32}$$

Then the global sufficient statistics can be updated using the following equations, and it will be sent to the next node (node $m+1$)

$$GSS1_j^t = GSS1_j^t + LSS1_{m,j}^t \tag{33}$$

$$GSS2_j^t = GSS2_j^t + LSS2_{m,j}^t \tag{34}$$

$$GSS3_j^t = GSS3_j^t + LSS3_{m,j}^t \tag{35}$$

After updating the global sufficient statistics, the hyperparameters can be updated using the following equations,

$$a_j^t = a_j^0 + GSS1_j^t \tag{36}$$

$$b_j^t = b_j^0 + GSS1_j^t \tag{37}$$

$$m_j^t = \frac{b_j^0 m_j^0 + GSS2_j^t}{b_j^t} \tag{38}$$

$$\Sigma_j^t = \Sigma_j^0 + GSS3_j^t + b_j^0 m_j^0 m_j^{0^T} - b_j^t m_j^t m_j^{t^T} \tag{39}$$

$$\upsilon_j^t = \upsilon_j^0 + GSS1_j^t \tag{40}$$

It is worth noting that using the vector of sufficient statistics; each node can compute all of the required parameters based on the variational Bayesian algorithm. Finally the mean of the parameters μ_j^t, T_j^t and α_j^t can be computed using the following equations,

$$E\left[\mu_j^t\right] = m_j^t \tag{41}$$

$$E\left[T_j^t\right] = \upsilon_j^t\left(\Sigma_j^t\right)^{-1} \tag{42}$$

$$\alpha_j^t = \frac{GSS1_j^t}{N} \tag{43}$$

whrere $N = \sum_{m=1}^{M} N_m$. This process iterates until the convergence criteria is met.

4. The Mobile Agent Based Distributed Variational Bayesian (MABDVB) Algorithm

In this section a novel MABDVB algorithm is proposed for density estimation and clustering of a finite mixture model in a distributed sensor network. It is assumed that the nodes of the network can be organized in a multiring topology. Consider a distributed sensor network of M nodes, a fusion center, and K mobile agents. Suppose that the nodes can be organized in K independent routes. In the proposed algorithm, the agents move throughout the independent routes and collect the information while executing the VB algorithm in an iterative manner. For example, at iteration t, the agents migrate through different nodes of the different routes; execute the VB algorithm at each node. Afterwards, the agents carry the resulted sufficient statistics to the next node in each route. Finally, all agents will carry the sufficient statistics of each route to the fusion center, and the sufficient statistics of all routes will be combined to compute a new global sufficient statistics. Then, at iteration $t+1$, the mobile agents will carry the updated global sufficient statistics vector to different routes, where it will be updated using the local measurements. This procedure will be done iteratively until convergence is reached. The MABDVB algorithm will be presented in details in the following.

At iteration $t+1$ of the MABDVB algorithm, the mobile agents carry the global sufficient statistics values $GSS1_j^t$, $GSS2_j^t$ and $GSS3_j^t$ from the node $m-1$ to node m in each route, and the hyperparameters will be calculated using the Eqs. (36)–(40). Then, the mean of the parameters μ_j^t and T_j^t will be computed using Eqs. (41) and (42). Using Eq. (28) the value of $SS1_{m,i,j}^t$ can be calculated and consequently from the Eqs. (30)–(32) the local sufficient statistics will be calculated. Finally, at node m in each route the mixture probability and the global sufficient statistics will be updated using the following equations,

$$\alpha_{m,j}^{t+1} = \frac{LSS1_{m,j}^t}{N_m} \tag{44}$$

$$GSS1_j^{t+1} = GSS1_j^t + LSS1_{m,j}^{t+1} - LSS1_{m,j}^t \tag{45}$$

$$GSS2_j^{t+1} = GSS2_j^t + LSS2_{m,j}^{t+1} - LSS2_{m,j}^t \tag{46}$$

$$GSS3_j^{t+1} = GSS3_j^t + LSS3_{m,j}^{t+1} - LSS3_{m,j}^t \tag{47}$$

Then the mobile agents will carry the updated global sufficient statistics values $\left\{GSS1_j^{t+1},\ GSS2_j^{t+1},\ GSS3_j^{t+1}\right\}$ to the next node and the aforementioned process will be iterated. After the last node, mobile agents will carry the global sufficient statistics vectors to the fusion center. The preceding process will be done by all of mobile agents in different routes independently. Injection of the mobile agents in different routes at the beginning of each iteration and receive them and fusion of the global sufficient statistics of all the routes after the last iteration is done by fusion center. Suppose that the final global sufficient statistics vector of the $k'th$ route denoted by GSS_k^t. After the last iteration, when

the fusion center receives the global sufficient statistics vectors, they will be averaged using the following equation.

$$GSS^{t+1} = \frac{\sum_{k=1}^{K} D_k \,.\, GSS_k^t}{\sum_{k=1}^{K} D_k} \tag{48}$$

where D_k denotes the number of data points in *k'th* route, and the total number of routes in the network is denoted by K. At the next iteration $t+1$, the resulting global sufficient statistics vector (GSS^{t+1}) will be carried to different routes by the mobile agents, and the preceding procedure will be repeated until reaching convergence.

In the proposed MABDVB algorithm, the variations of the log-likelihood function can be used to obtain the convergence criterion. If this variation is less than or equal to a certain threshold value, the algorithm will stop. In the proposed algorithm, after updating parameters using the local data of each node, the value of local log-likelihood function corresponding to each can be calculated using the following equation,

$$\mathcal{L}\left(\theta^t\right) \equiv \sum_{i=1}^{N} \log\left(\sum_{j=1}^{J} \alpha_j^t \mathcal{N}\left(\mathbf{y_i}|\mu_j^t, T_j^t\right)\right) \tag{49}$$

The MABDVB algorithm stops whenever $\mathcal{L}\left(\theta^{t+1}\right) - \mathcal{L}\left(\theta^t\right)$ become less than the convergence threshold value.

The proposed MABDVB algorithm can be represented in a stepwise procedure as follows,

1. A mobile agent is assigned to each dependent route.
2. Mobile agents move through the assigned routes, and each agent performs the following,
   - Calculates the local sufficient statistics using (28)–(32) based on local measurements.
   - Updates the values of the mixture probability and global sufficient statistics using (44)–(47).
   - Updates the values of hyperparameters using (36)–(40), and then calculates the mean of the parameters (41) and (42).
3. Each mobile agent carries the global sufficient statistics vector of its corresponding route to the fusion center to be combined using (48).
4. If the convergence criterion is reached, the algorithm will be stopped; otherwise, the resulting global sufficient statistics vector is sent to different routes using the mobile agents, and step 2 will be repeated.

5. Convergence Analysis of the Proposed Algorithm

In the variational approximation, the variational free energy $\mathcal{F}\left(\hat{f}_z, \hat{f}_\theta\right)$ provides a lower bound on the log-likelihood value. At each iteration of the variational Bayesian algorithm, the value of $\mathcal{F}$ will be optimized with respect to the selected posterior distributions $\hat{f}_z$ and $\hat{f}_\theta$. It should be noted

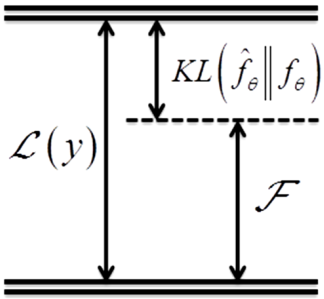

Fig. 1 Graphical representation of the negative free energy

that $\hat{f}_\theta$ and $\hat{f}_z$ are the density of the parameters vector θ and missing variable z, respectively. The negative free energy can be represented as

$$\begin{aligned}\mathcal{F}\left(\hat{f}_Z, \hat{f}_\theta\right) &= \int \sum_{Z} \hat{f}_\theta(\theta)\, \hat{f}_Z(\mathbf{z}) \log \frac{f(y, \mathbf{z}, \theta)}{\hat{f}_\theta(\theta)\, \hat{f}_Z(\mathbf{z})} d\theta \\ &= \log f(y) - KL\left(\hat{f}_\theta \| f_\theta\right) = \mathcal{L}(y) - KL\left(\hat{f}_\theta \| f_\theta\right)\end{aligned} \tag{50}$$

where $KL\left(\hat{f}_\theta \| f_\theta\right)$ represents the Kullback-Leibler (KL) divergence criteria between $\hat{f}_\theta$ and f_θ. A graphical representation of the negative free energy is also illustrated in Fig. 1. This figure shows that $\mathcal{F}$ provide a lower bound on the log-likelihood function, and the difference between these values is KL divergence criterion.

In the proposed MABDVB algorithm, it is assumed that the data sets of different nodes are independent. As a result we have $\hat{f}_z = \prod_{m=1}^{M} \hat{f}_{z_m}$ and $\hat{f}_\theta = \prod_{m=1}^{M} \hat{f}_{\theta_m}$. Therefore, the $\mathcal{F}$ function is equal to the sum of local negative free energy over all of the network nodes as follows,

$$\mathcal{F}\left(\hat{f}_\theta, \hat{f}_Z\right) = \sum_{m=1}^{M} \mathcal{F}_m\left(\hat{f}_{\theta_m}, \hat{f}_{Z_m}\right) \tag{51}$$

where $\mathcal{F}_m$ is the local negative free energy at node m, and it can be written as

$$\begin{aligned}&\mathcal{F}\left(\hat{f}_\theta, \hat{f}_Z\right) = \sum_{m=1}^{M} \mathcal{F}_m\left(\hat{f}_{\theta_m}, \hat{f}_{Z_m}\right) = \\ &\sum_{m=1}^{M}\left[\int \sum_{Z_m} \hat{f}_{\theta_m}(\theta_m)\, \hat{f}_{Z_m}(\mathbf{Z}_m) \log \frac{f(\mathbf{Y}_m, \mathbf{Z}_m, \theta_m)}{\hat{f}_\theta(\theta_m)\, \hat{f}_Z(\mathbf{Z}_m)} d\theta_m\right]\end{aligned} \tag{52}$$

To improve convergence of the proposed MABDVB algorithm, it should be shown that the value of $\mathcal{F}$ increasingly moves toward its maximum value at each node. In other words, it should be exposed that $\mathcal{F}$ is a non-decreasing function. In this case, assuming that $\bar{\mathcal{F}}$ is a local maximum value and that initial values of the parameters are selected appropriately, the proposed algorithm will eventually converge to its fixed point.

In the proposed MABDVB algorithm, at each node m, first the value of $\hat{f}_{Z_m}(z_m)$ is calculated such that the value of $\mathcal{F}_m$ is maximized with respect to $\hat{f}_{Z_m}$, then the value of

$\hat{f}_{\theta_m}(\theta_m)$ is calculated to maximize $\mathcal{F}_m$ with respect to $\hat{f}_{\theta_m}$. In other words, updating the value of local sufficient statistics maximizes the value of $\mathcal{F}_m$ with respect to $\hat{f}_{Z_m}$, and by calculating the hyperparameters the value of $\mathcal{F}_m$ will be maximized with respect to $\hat{f}_{\theta_m}$. Therefore, the final value of $\mathcal{F}_m$ in each route will be maximized. At the end of each iteration, the new global sufficient statistic values of all agents will be calculated using (48) in the fusion center. In fact, the new global sufficient statistics obtained by averaging the global sufficient statistics of all the mobile agents that results in averaging the $\mathcal{F}$ values of all of the routes. It is shown that the $\mathcal{F}$ values of all of the routes have increased (or at least not decreased), thus their average value will not decrease. Therefore, it can be concluded that MABDVB is a non-decreasing algorithm, and the proposed algorithm will be converged to the optimal parameters.

6. Simulation Results

In this section the performance of the proposed MABDVB algorithm will be illustrated in different one-dimensional and two-dimensional data cases. The MATLAB software has been used for the simulations.

6.1 One Dimensional Data

Consider a single-ring sensor network that has 50 nodes and a fusion center. Each node contains 100 sensor measurements. It is assumed that measurements of each node can be modeled with a mixture of Gaussians. Consider a 1-D data set simulated from a three-component Gaussian mixture given by

$$0.2\mathcal{N}(3, 0.25) + 0.3\mathcal{N}(0, 0.25) + 0.5\mathcal{N}(-2, 0.25)$$

The true and estimated parameters obtained from MABDVB and centralized VB algorithms are presented in Table 1. The values of the estimated means and variances of the parameters are presented in this table. As it is shown in this table, very good estimates of the true values have been obtained. The estimated values using the MABDVB algorithm are closer to the true values than those using the centralized VB algorithm. The true density of the simulated data set and the density fitted by MABDVBA and centralized VBA are also illustrated in Fig. 2. Referring to this figure, the estimated probability density function using both algorithms closely approximates the true one; however the MABDVB algorithm is more accurate than the centralized one. Figure 3 can be used for comparison between the convergence speed of the proposed algorithm and the centralized VB algorithm. In both centralized and MABDVB algorithms using equal number of nodes and data points at each node of the network, the iteration number of both algorithm can be seen in Fig. 3. As shown in this figure, the number of iterations of the MABDVB algorithm is less than that of the centralized for different number of nodes. Indeed, faster convergence rate of the proposed MABDVB algorithm is obvious.

Table 1 True and estimated parameters using the MABDVB and centralized VB algorithms at node 1

Component		1	2	3
True values	means	3	0	−2
	variances	0.25	0.25	0.25
Estimated values using centralized VB	means	3.03	−0.132	−1.982
	variances	0.267	0.256	0.152
Estimated values using MABDVB	means	3.017	−0.008	−1.998
	variances	0.238	0.251	0.238

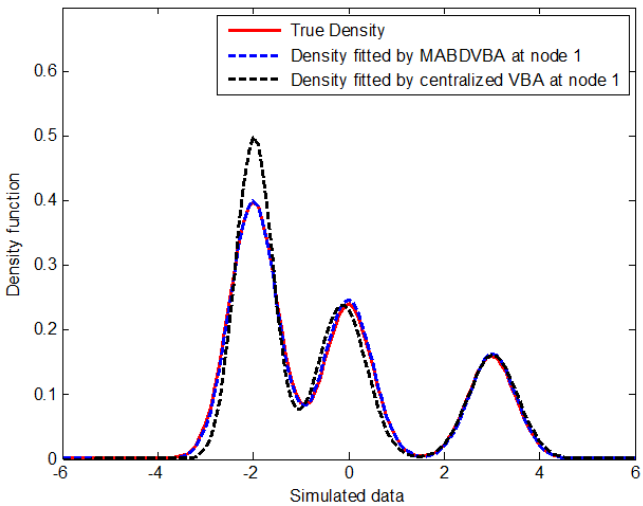

Fig. 2 True and fitted Density function using MABDVBA and centralized VBA

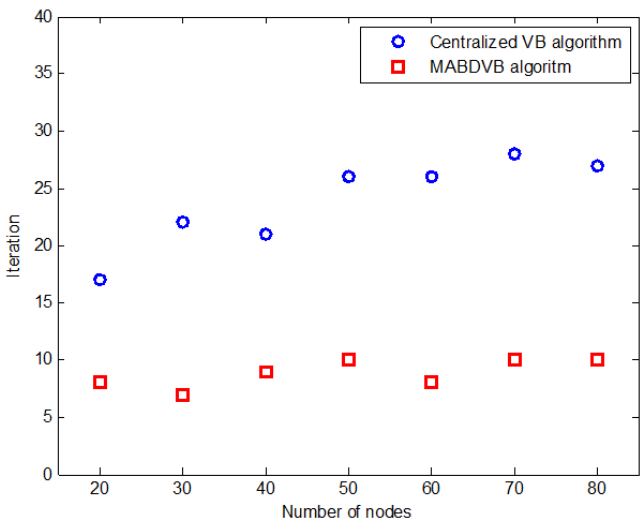

Fig. 3 Number of iterations of MABDVB and Centralized VB algorithms for different number of nodes.

6.2 Two Dimensional Data

Traffic modeling, estimation, and prediction can be considered as a duty of intelligent transportation systems. In this subsection the proposed MABDVB algorithm will be used to estimate the density function in a freeway. Traffic flow and mean traffic speed are the parameters to be estimated. A

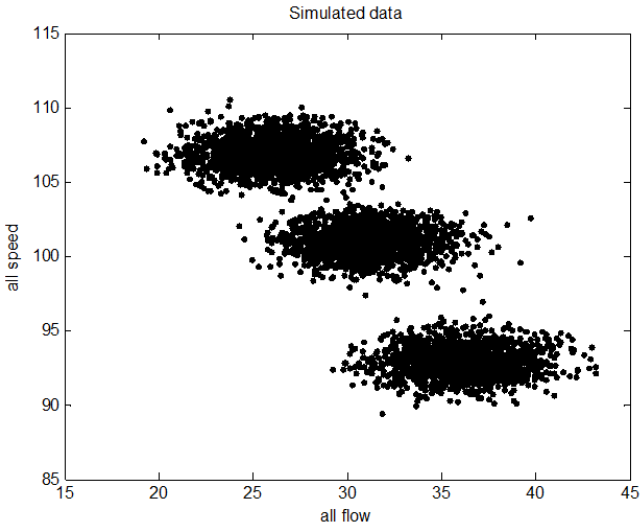

Fig. 4 Data distribution

statistical GMM has been considered, and traffic measurements, are obtained by some inductive loop detectors. The proposed MABDVB algorithm is used to estimate the parameters of the GMM in a distributed manner. Consider a traffic sensor network with 50 nodes ($M = 50$) for simulation. The sensors are placed in a square $25km \times 25km$ based on traffic engineering principal and the best geographical position. Each node has 100 data observations ($N_m = 100$), and the observations are generated from three Gaussian components ($J = 3$). The data distribution is shown in Fig. 4. Each of the Gaussian components is a 2D Gaussian density, which can represent 2D environment data clusters. In the first 20 nodes (nodes 1 to 20), 40% of observations come from the second Gaussian component, and the other 60% of observations evenly come from the other two Gaussian components. In the next 15 nodes (nodes 21 to 35), 40% of observations come from the third component, and in the last 15 nodes (36 to 50), 40% of observations come from the first component. Other observations of these 30 nodes evenly come from the other two Gaussian components.

For comparison, first the standard VB algorithm is executed in each node using only local data. After that, the proposed MABDVB algorithm will be used to estimate the parameters in all nodes. True and fitted means and variances using the standard VBA and the proposed algorithm are presented in Table 2. The estimated mean values of flow using standard VB algorithm and proposed algorithm are shown in Fig. 5, and the estimated speed mean values using the mentioned algorithms are represented in Fig. 6.

As shown, the curve of the estimated mean values of each component using MABDVB algorithm are almost constant and very close to the true values for all nodes. Therefore, the proposed algorithm provides a more accurate estimate of the parameters than the standard VB algorithm. The true density of the simulated data set and the density fitted by MABDVB and centralized VB algorithms are also illustrated in Fig. 7. Referring to this figure, the estimated prob-

Table 2 True and estimated parameters using the MABDVB and centralized VB algorithms.

Component		1, 2, 3
True values	Means	$\begin{bmatrix}26\\107\end{bmatrix}, \begin{bmatrix}31\\101\end{bmatrix}, \begin{bmatrix}36\\93\end{bmatrix}$
	Covariances	$\begin{bmatrix}5&0\\0&1\end{bmatrix}, \begin{bmatrix}5&0\\0&1\end{bmatrix}, \begin{bmatrix}5&0\\0&1\end{bmatrix}$
Estimated values	Means	$\begin{bmatrix}26.04\\106.98\end{bmatrix}, \begin{bmatrix}30.93\\100.99\end{bmatrix}, \begin{bmatrix}36.01\\93.01\end{bmatrix}$
(centralized VB)	Covariances	$\begin{bmatrix}4.9&0.0\\0.0&1.0\end{bmatrix}, \begin{bmatrix}5.3&0.1\\0.1&1.0\end{bmatrix}, \begin{bmatrix}5.0&0.1\\0.1&0.9\end{bmatrix}$
Estimated values	Means	$\begin{bmatrix}25.98\\107.01\end{bmatrix}, \begin{bmatrix}30.98\\101.00\end{bmatrix}, \begin{bmatrix}36.02\\92.96\end{bmatrix}$
(MABDVB)	Covariances	$\begin{bmatrix}4.8&0.1\\0.1&1\end{bmatrix}, \begin{bmatrix}4.9&0.0\\0.0&1.0\end{bmatrix}, \begin{bmatrix}5.1&0.0\\0.0&0.9\end{bmatrix}$

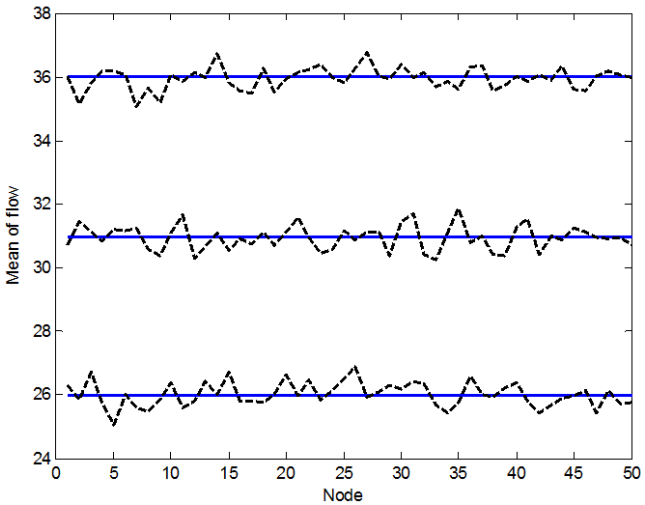

Fig. 5 Estimated mean values of flow using MABDVB (solid line) and standard VB (dashed line) algorithms in a network with 50 nodes.

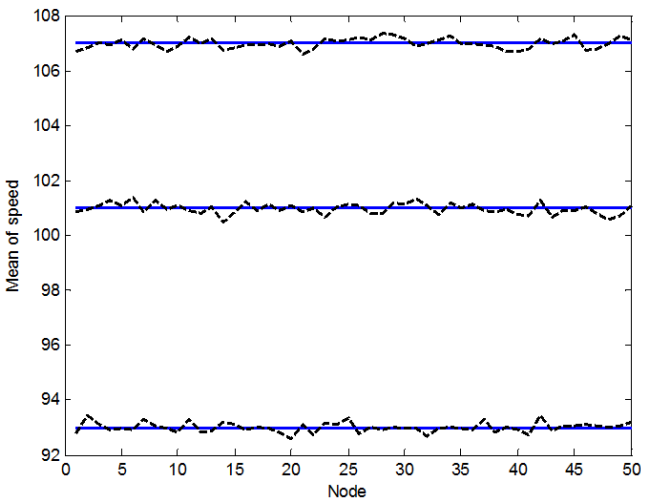

Fig. 6 Estimated mean values of speed using MABDVB (solid line) and standard VB (dashed line) algorithms in a network with 50 nodes.

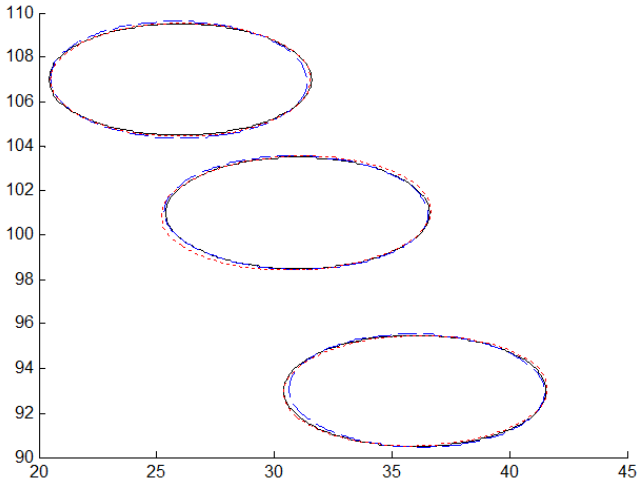

Fig. 7 True density of simulated data (solid ellipses), the fitted Density using centralized VB (dotted ellipses) and MABDVB (dashed ellipses) algorithms.

ability density function using both algorithms closely approximates the true one; however the MABDVB algorithm is more accurate than the centralized one.

7. Conclusion

In this paper, a MABDVB algorithm has been proposed for distributed density estimation in sensor networks. In the proposed method, the mobile agents execute the VB algorithm by moving between different nodes and carry the resulting sufficient statistics vector to the fusion center at the end of iterations to be updated. Convergence of the proposed method has been also studied, and it has been proven that the estimated parameters will eventually converge to the true values. Also, the proposed method has been used for flow and speed density prediction with distributed traffic sensors. The simulation results illustrate the good performance of the proposed algorithm for one-dimensional and two-dimensional data sets.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Commun. Mag., vol.40, no.8, pp.102–114, Aug. 2002.
[2] A.H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," IEEE Signal Process. Mag., vol.30, no.3, pp.155–171, 2013.
[3] J. Chen and A.H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," IEEE Trans. Signal Process., vol.60, no.8, pp.4289–4305, 2012.
[4] G. Meteos and G.B. Giannakis, "Distributed recursive least-squares: stability and performance analysis," IEEE Trans. Signal Process., vol.60, no.7, pp.3740–3754, 2012.
[5] F.S. Cattivelli and A.H. Sayed, "Diffusion LMS strategies for distributed estimation," IEEE Trans. Signal Process., vol.58, no.3, pp.1035–1048, 2010.
[6] G. Meteos, J.A. Bazerque, and G.B. Giannakis, "Distributed sparse linear regression," IEEE Trans. Signal Process., vol.58, no.10, pp.5262–5276, 2010.
[7] Z. Liu, Y. Liu, and C. Li, "Distributed sparse recursive least-squares over networks," IEEE Trans. Signal Process., vol.62, no.6, pp.1385–1395, 2014.
[8] B. Safarinejadian and K. Hasanpour, "Distributed data clustering using mobile agents and EM algorithm," IEEE Syst. J., vol.10, no.1, pp.281–289, 2016.
[9] D. Gu and H. Hu, "Spatial Gausssian process regression with mobile sensor networks," IEEE Trans. Neural Netw. Learn. Syst., vol.23, no.8, pp.1279–1290, 2012.
[10] B. Safarinejadian and M. Estakhri Estahbanati, "Consensus Filter-Based Distributed Variational Bayesian Algorithm for Flow and Speed Density Prediction with Distributed Traffic Sensors," IEEE Syst. J., vol.PP, no.99, pp.1–10, 2015.
[11] G.J. McLachlan and T. Krishnan, The EM Algorithm and Extensions, pp.120–211, Wiley, New York, NY, USA, 1997.
[12] 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.
[13] D. Gu, "Distributed EM algorithm for Gaussian mixtures in sensor networks," IEEE Trans. Neural Netw., vol.19, no.7, pp.1154–1166, July 2008.
[14] B. Safarinejadian, M.B. Menhaj, and M. Karrari, "A distributed EM algorithm to estimate the parameters of a finite mixture of components," Knowl. Inf. Syst., vol.23, no.3, pp.267–292, June 2010.
[15] 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, Aug. 2011.
[16] F. Fagnani, S.M. Fosson, and C. Ravazzi, "A DISTRIBUTED CLASSIFICATION/ESTIMATION ALGORITHM FOR SENSOR NETWORKS," SIAM J. CONTROL OPTIM. vol.52, no.1, pp.189–218, 2014.
[17] A.P. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data via the EM algorithm," J.R. Stat. Soc., ser. B, vol.39, no.1, pp.1–38, 1977.
[18] S.S. Pereira, A. Pages-Zamora, and R. Lopez-Valcarce, "A diffusionbased distributed EM algorithm for density estimation in wireless sensor networks," in Proc. IEEE ICASSP, pp.4449–4453, May 2013.
[19] S.S. Pereira, R. Lopez-Valcarce, and A. Pages-Zamora, "A diffusion-based EM algorithm for distributed density estimation in unreliable wireless sensor networks," IEEE Signal Process. Lett., vol.20, no.6, pp.595–598, June 2013.
[20] B. Safarinejadian, M.B. Menhaj, and M. Karrari, "A distributed variational Bayesian algorithm for density estimation in sensor networks," IEICE Trans. Inf. & Syst., vol.E92-D, no.5, pp.1037–1048, May 2009.
[21] B. Safarinejadian, M.B. Menhaj, and M. Karrari, "Distributed variational Bayesian algorithms for Gaussian mixtures in sensor networks," Signal Process., vol.90, no.4, pp.1197–1208, April 2010.
[22] B. Wang and D.M. Titterington, "Convergence properties of a general algorithm for calculating variational Bayesian estimates for a normal mixture model," Bayesian Anal., vol.1, no.3, pp.625–650, 2006.
[23] Y. Xu and H. Qi, "Mobile agent migration modeling and design for target tracking in wireless sensor networks," Ad Hoc Netw., vol.6, no.1, pp.1–16, Jan. 2008.
[24] P.K. Biswas, H. Qi, and Y. Xu, "Mobile-agent based collaborative sensor fusion," Inf. Fusion, vol.9, no.3, pp.399–411, July 2008.
[25] A. Ramezani, B. Moshiri, A.R. Kian, B.N. Aarabi, and B. Abdulhai, "Distributed maximum likelihood estimation for flow and speed density prediction in distributed traffic detectors with Gaussian mixture model assumption," IET Intelligent Transport Systems, vol.6, no.2, pp.215–222, 2012.

Mohiyeddin Mozaffari was born in 1989. He received B.S. degree in Electrical Engineering from Shahid Bahonar University of Kerman, Kerman, Iran, in 2011 and M.S. degree in Control Engineering from Shiraz University of Technology, Shiraz, Iran, in 2013. He is currently with the Control Engineering Department, Shiraz University of Technology as a Ph.D student. His research interests include state estimation, system identification, and sensor networks.

Behrouz Safarinejadian received B.S. and M.S. degrees from Shiraz University, Shiraz, Iran, in 2002 and 2005, respectively, and Ph.D. degree from Amirkabir University of Technology, Tehran, Iran, in 2009. Since 2009, he has been with the Faculty of Electrical and Electronic Engineering, Shiraz University of Technology, Shiraz, Iran. His research interests include computational intelligence, control systems theory, estimation theory, statistical signal processing, sensor networks and fault detection.