A Distributed Variational Bayesian Algorithm for Density Estimation in Sensor Networks

Behrooz SAFARINEJADIAN†(a), Student Member, Mohammad B. MENHAJ†(b), and Mehdi KARRARI†(c), Nonmembers

SUMMARY In this paper, the problem of density estimation and clustering in sensor networks is considered. It is assumed that measurements of the sensors can be statistically modeled by a common Gaussian mixture model. This paper develops a distributed variational Bayesian algorithm (DVBA) to estimate the parameters of this model. This algorithm produces an estimate of the density of the sensor data without requiring the data to be transmitted to and processed at a central location. Alternatively, DVBA can be viewed as a distributed processing approach for clustering the sensor data into components corresponding to predominant environmental features sensed by the network. The convergence of the proposed DVBA is then investigated. Finally, to verify the performance of DVBA, we perform several simulations of sensor networks. Simulation results are very promising.

key words: sensor networks, clustering, density estimation, mixture of Gaussians, variational approximations.

1. Introduction

Advances in network technology, like peer-to-peer networks on the internet or sensor networks, have highlighted the need for efficient ways to deal with large amounts of data that are distributed over a set of nodes. Examples are financial data reported on the internet, weather data observed by a set of sensors, etc. In particular, in many data mining applications we are interested in learning a global model from data reported on the internet or sensor networks, have highlighted the need for e
ductions are then obtained through a gossip-based randomized method. A distributed incremental EM algorithm has also been developed in [9] for density estimation in peer-to-peer networks. An important problem of the EM algorithm is that singularity may happen in the estimated parameters. Especially, if the model order is not properly selected and the assumed order is greater than the real order of the observed data, singularity will be inevitable.

Recently, variational methods have gained popularity in the machine learning literature and have also been used to estimate the parameters of finite mixture models. The variational Bayesian method aims to construct a tight lower bound on the data marginal likelihood and then seeks to optimize this bound using an iterative scheme [10]–[15]. Furthermore, using variational methods, the order of mixture model may also be estimated. In other words, the variational approach allows the simultaneous estimation of the component parameters and the model complexity. We will consider distributed estimation of the model complexity in later publications. An important advantage of the variational approach is that, despite use of the EM algorithm, we do not have the singularity problem.

In this paper, it is assumed that measurements of the sensors are statistically modeled by a common mixture of Gaussians. A distributed variational Bayesian algorithm (DVBA) is then developed for estimating the Gaussian components which are common to the sensor network. DVBA can also be used as a general distributed data mining algorithm for density estimation and clustering of the data distributed over the nodes of a network.

The proposed DVBA can be considered as a distributed implementation of an incremental variational approximation to distributed inferencing in sensor networks has been studied in [5], [6]. The underlying probability distribution is represented as a graph, and inferencing proceeds as a sequence of message-passing operations between nodes of the graph. In [6], such belief propagation based inferencing has been applied to multivariate Gaussian distributions.

A distributed EM (Expectation Maximization) algorithm has been developed in [7] for density estimation in sensor networks assuming that the measurements are statistically modeled by a mixture of Gaussians. In [8] a gossip-based distributed EM algorithm has been proposed for Gaussian mixture learning named Newscast EM, in which the E and M steps of the EM algorithm are first performed locally, and the global estimate of means and covariances are then obtained through a gossip-based randomized method. A distributed incremental EM algorithm has also been developed in [9] for density estimation in peer-to-peer networks. An important problem of the EM algorithm is that singularity may happen in the estimated parameters. Especially, if the model order is not properly selected and the assumed order is greater than the real order of the observed data, singularity will be inevitable.

The applicability of graphical models and belief propa-
which is a fast alternative to the standard variational approximation for mixture density estimation. In other words, the proposed DVBA implements the variational approximation in a distributed structure, it requires a modest communication cost and its convergence rate is superior to the standard variational Bayesian algorithm. Convergence properties of the DVBA are also studied here and it is shown that the DVBA converges to a maximum likelihood point.

The rest of the paper is organized as follows. In Sect. 2, the basic problem statement is given, and observations and data models are defined. Section 3 reviews the variational approximations for mixture density estimation, and a distributed algorithm for its implementation is presented. Section 4 develops a fully distributed variational Bayesian algorithm that aims to reduce the number of iterations and hence communications required to compute the maximum likelihood estimate. Section 5 is devoted to convergence analysis of DVBA. The results of simulations are presented in Sect. 6. Finally, Sect. 7 concludes the paper.

2. Problem Statement

Consider a network of $M$ nodes and a $d$-dimensional random vector $Y_m$ with probability density function $f(Y_m)$ which corresponds to node $M$. Each data (measurement) $y_{m,i}$ is a realization of the random vector $Y_m$. Assume that distribution of the observations is represented by a finite mixture of components:

$$f(y_m; \pi_m, \theta) = \sum_{j=1}^{J} \pi_{m,j} f_j(y_m, \theta_j)$$  \hspace{1cm} (1)

where $\{\pi_{m,j}\}_{j=1}^{J}$ are the mixture probabilities at node $m$, $\theta_j$ is the set of parameters defining the $j$th component and $J$ is the number of mixture components. The mixture probabilities $\{\pi_{m,j}\}$ may be different at various nodes while the parameters $\theta_j$ are the same throughout the network. The set of data points of the $m$th node is represented by $Y_m = \{y_{m,i}\}_{i=1}^{N_m}$ in which $N_m$ is number of observations at node $m$. It is assumed that measurements of each node are independent and identically distributed.

Consider a set of missing variables $Z_m = \{z_{m,i}\}$ corresponding to $Y_m = \{y_{m,i}\}$. Each $z_{m,i} = \{z_{m,i}^1, z_{m,i}^2, \ldots, z_{m,i}^J\}$ is a binary vector indicating by which component the data $y_{m,i}$ is produced.

We would say $y_{m,i}$ is produced by the $j$th component of the mixture if for all $r \neq j$, $z_{m,i}^r = 0$ and $z_{m,i}^j = 1$. The pair $x_{m,i} = \{y_{m,i}, z_{m,i}\}$ is regarded as the complete data and we write $X_m = \{Y_m, Z_m\}$ in which $X_m = \{x_{m,i}\}$. While our approach can be applied to arbitrary models, for simplicity we consider here Gaussian component distributions, $f_j(y_m; \theta_j) = N(\mu_j, T_j)$, where $\mu_j$ is the mean and $T_j$ the precision (inverse covariance) matrix. In this paper, we develop a distributed variational Bayesian algorithm to estimate the parameters of this Gaussian mixture model using the data set $Y = \{Y_m\}_{m=1}^{M}$.

3. The Variational Bayesian Inference Approach

Variational methods for inference about mixture models have been appearing in the machine learning literature over recent years. A Bayesian framework based on the variational approximation was proposed in [16] for estimating the parameters and hyperparameters of a mixture model as an alternative to the maximum likelihood approach which tends to over-fit the model. In [10] the variational Bayes technique was extended to perform model selection as well as estimating parameters by introducing a prior over the model structure. For mixture models this leads to a posterior distribution over the number of components in the model. In [16] and [10] the connection between the EM (Expectation Maximization) algorithm and the variational Bayes algorithm was emphasized. Variational Bayes is an EM-like algorithm, the expectation step of EM corresponding to finding the expected value of the posterior of the component indicator variables in variational Bayes. The maximization step of EM relates to estimating the model parameters in variational Bayes by maximizing the lower bound on the marginal log-likelihood.

In [13] the variational learning technique was also applied to the analysis of a finite mixture of Gaussians considering an approach which estimates the number of components as well as estimating component parameters. They do this by optimizing the mixing co-efficients using type-2 maximum likelihood and marginalizing out the model parameters using variational methods. This leads to automatic recovery of the number of components given a fixed maximum potential number of components. In [17] variational methods were applied to the Bayesian analysis of mixture of Gaussian distributions. It was also shown how the Deviance Information Criterion (DIC), can be extended to these types of model by exploiting the use of variational approximations. They use a hierarchical model for implementation of the variational method. Here, the variational approach is first introduced and then it is shown that how it can be used for estimating the parameters of a Gaussian mixture model. Afterwards, assuming that measurements are obtained through a sensor network, a two pass distributed variational Bayesian algorithm is developed for estimating the mixture parameters.

3.1 The Variational Bayesian Algorithm

Suppose we have observed the data set $Y = \{y_i\}$. Here, a parametric model with parameters $\theta$ is assumed and missing or unobserved values are denoted by $Z = \{z_i\}$. Of interest is the posterior distribution of $\theta$ given $Y$. The idea of the variational approximation is to approximate the joint conditional density of $\theta$ and $Z$, by a more tractable distribution $f(\theta, z)$, by minimizing the Kullback-Leibler (KL) divergence between the approximating density $\tilde{f}(\theta, z)$ and the true joint conditional density, $f(z, \theta \| y)$. The motivation for this is that we wish to obtain a tight lower bound on the log
likelihood \( L = \log f(y) \). We can establish a lower bound for \( L \) as follows:
\[
\log f(y) = \log \int \sum_z f(y, z, \theta) d\theta \\
= \log \int \sum_z \hat{f}(z, \theta) \frac{f(y, z, \theta)}{f(z, \theta)} d\theta \\
= \int \sum_z \hat{f}(z, \theta) \log \frac{f(y, z, \theta)}{f(z, \theta)} d\theta
\]
The last term was obtained by Jensen’s inequality. This lower bound which is denoted by \( F \), represents the variational free energy. If \( \hat{f}(\theta, z) \) is restricted to have a factorized form \( \hat{f}(\theta, z) = \hat{f}_\theta(\theta) \hat{f}_z(z) \), the function \( F \) may also be defined as:
\[
F(\hat{f}(\theta), \hat{f}(z)) = \\
E_Z,\theta \left[ \log \frac{f(y, z|\theta)}{\hat{f}(z)} \right] - KL \left[ \hat{f}_\theta(\theta)||f_\theta(\theta) \right]
\]
where the average in the first term on the r.h.s. is taken w.r.t. \( \hat{f}(z, \theta) \). The first term corresponds to an averaged likelihood and the second term is the KL distance between the prior and posterior over the parameters, given by:
\[
KL \left[ \hat{f}_\theta(\theta)||f_\theta(\theta) \right] = \int \hat{f}_\theta(\theta) \log \frac{\hat{f}_\theta(\theta)}{f_\theta(\theta)} d\theta
\]
The variational Bayes (VB) algorithm maximizes the negative free energy \( F \) via an EM-like strategy (see [10]). In the E-step, we compute the posterior over the hidden variables by solving \( \partial F / \partial \hat{f}_z = 0 \) to get
\[
\hat{f}_z(z) \propto \exp[I(z)]
\] (2)
where
\[
I(z) = E_\theta[\log f(y, z|\theta)]
\]
and the expectation is taken w.r.t. \( \hat{f}_\theta(\theta) \).

In the M-step, rather than the optimal parameters, we compute the posterior distribution over the parameters by solving \( \partial F / \partial \hat{f}_\theta = 0 \) to get
\[
\hat{f}_\theta(\theta) \propto \exp[I(\theta)]f(\theta)
\] (3)
where
\[
I(\theta) = E_z[\log f(y, z|\theta)]
\]
and the expectation is taken w.r.t. \( \hat{f}_z(z) \).

This is where the concept of conjugate priors becomes useful. We chose the prior \( f(\theta) \) from a family of distributions such that \( \hat{f}_\theta(\theta) \propto \exp[I(\theta)]f(\theta) \) belongs to the same family. \( f(\theta) \) is then said to be conjugate to \( \exp[I(\theta)] \). In particular, learning in the VB framework simply amounts to updating the hyperparameters, i.e., transforming the prior parameters to the posterior parameters.

3.2 Application of the Variational Method for Mixtures of Gaussian Distributions

This section is devoted to apply the variational Bayesian method for a Gaussian mixture model. Consider a mixture of \( J \) multivariate Gaussian distributions with unknown means, variances and mixing weights as:
\[
f(y; \pi, \theta) = \sum_{j=1}^J \pi_j f(z; \mu_j, \Sigma_j)
\]
Here, conjugate priors are assigned to the parameters \( \pi \) and \( \theta \). The mixing weights are assigned a Dirichlet prior distribution
\[
f_\pi(\pi) = \text{Dir}(\pi; \alpha_1^0, \ldots, \alpha_J^0)
\]
in which
\[
\text{Dir}(\pi; \alpha_1^0, \ldots, \alpha_J^0) = \frac{1}{B(\alpha^0)} \prod_{i=1}^J \alpha_i^{\alpha_i^0-1}
\]
The normalizing constant is the multinomial beta function, which can be expressed in terms of the gamma function
\[
B(\alpha) = \frac{\prod_{i=1}^J \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^J \alpha_i)}
\]
and the means are assigned independent multivariate normal conjugate priors, conditional on the covariance matrices, as
\[
f_{\mu T}(\mu|T) = \prod_{j=1}^J N(\mu_j; m_j^0, (\beta_j^0 T_j)^{-1})
\]
where \( \mu = (\mu_1, \ldots, \mu_J) \) and \( T = (T_1, \ldots, T_J) \). The precision matrices are given independent Wishart prior distributions,
\[
f_{T}(T) = \prod_{j=1}^J W(T_j; v_j, \Sigma_j^0)
\]
where
\[
W(T_j; v_j, \Sigma_j^0) = \\
\frac{|T_j|^{\frac{v_j-d-1}{2}} \exp tr \left[ -\frac{1}{2} \Sigma_j T_j \right]}{2^{\frac{d(d-1)}{4}} \pi^{\frac{d(d-1)}{4}} \Gamma \left( \frac{1}{2} v_j + 1 \right)}
\]
Thus, the joint distribution of all of the random variables is given by
\[
f(y, z, \pi, \theta) = f(y, z|\pi, \theta)f(\pi)f(\mu|T)f(T)
\]
The quantities \( \alpha_i^0, m_i^0, \beta_i^0, v_j^0 \) and \( \sigma_j^0 \) are called hyperparameters.

For the variational approximation of \( f(\theta, z|y) \), we take \( \hat{f}(\theta, z) \) to have the factorized form \( \hat{f}(\theta, z) = \hat{f}_\theta(\theta) \hat{f}_z(z) \). Hyperparameters of the posterior distributions obtained using the Eqs. (2) and (3) are then
\[ \hat{f}_\pi(\pi) = \text{Dir}(\pi; \alpha_1, \ldots, \alpha_J) \]
\[ \hat{f}_\mu(T) = \prod_{j=1}^J N(\mu_j; m_j, (\beta_j T_j)^{-1}) \]
\[ \hat{f}_T(T) = \prod_{j=1}^J W(T_j; v_j, \Sigma_j) \]

with hyperparameters given by
\[ \alpha_j = \alpha_j^0 + \sum_{i=1}^N q_{i,j} \]
\[ \beta_j = \beta_j^0 + \sum_{i=1}^N q_{i,j} \]
\[ m_j = \frac{\beta_j^0 m_j^0 + \sum_{i=1}^N q_{i,j} y_i}{\beta_j} \]
\[ \Sigma_j = \Sigma_j^0 + \sum_{i=1}^N q_{i,j} y_i y_i^T + \beta_j^0 m_j^0 m_j^0 - \beta_j m_j m_j^T \]
\[ v_j = v_j^0 + \sum_{i=1}^N q_{i,j} \]

In the above
\[ q_{i,j} = \frac{\varphi_{i,j}}{s_i} \]

with \( s_i = \sum_{j=1}^J \varphi_{i,j} \) and
\[ \varphi_{i,j} = \hat{\pi}_j T_j^{1/2} e^{-\frac{1}{2}(y_i - m_j)^T E(T_j)(y_i - m_j) - \frac{1}{2} \lambda_j} \]

\[ \hat{\pi}_j = \exp \left[ \int \hat{f}(\pi) \log \pi_j d\pi \right] \]
\[ \hat{T}_j = \exp \left[ \int \hat{f}(T_j) \log |T_j| dT_j \right] \]
\[ E[T_j] = v_j \Sigma_j^{-1} \]

3.3 A Distributed Implementation of the Variational Bayesian Algorithm

Here, we assume that the observed data is distributed over nodes of a network. Suppose that distribution of the observations is represented by the finite mixture of components shown in Eq. (1). The parameters of this model can be estimated using a distributed EM (Expectation Maximization) algorithm [7], [8]. An important problem of the EM algorithm is that singularity may happen in the estimated parameters. Especially when the model order is not properly selected and the specified order is greater than the real order of the observed data, singularity will be inevitable. In this section, we propose a two-pass distributed algorithm to implement the variational approach for estimating parameters of the Gaussian mixture model. In the next section, a full distributed variational algorithm is developed that performs in a cyclic communication structure and aims to reduce the number of iterations.

Since the observed data is distributed in \( M \) different nodes, we define a vector of sufficient statistics as:
\[ s^t = \{q_j^t, d_j^t, b_j^t\} \]

where
\[ q_j^t = \sum_{m=1}^M \sum_{j=1}^{N_m} q_{m,i,j} \]
\[ d_j^t = \sum_{m=1}^M \sum_{i=1}^{N_m} q_{m,i,j} y_{m,i} \]
\[ b_j^t = \sum_{m=1}^M \sum_{i=1}^{N_m} q_{m,i,j} y_{m,i}^T \]

Assume that \( q_{m,i,j}^t \) is the local quantity computed at node \( m \) and iteration \( t \) using
\[ q_{m,i,j}^t = \frac{\varphi_{m,i,j}^t}{\sum_{j=1}^J \varphi_{m,i,j}^t} \]

\[ \hat{\pi}_j^{-1}(\hat{T}_j^{-1})^{1/2} e^{-\frac{1}{2}(y_i - m_j)^T E(T_j)(y_i - m_j) - \frac{1}{2} \lambda_j} \]

During the first pass, the value of \( q_{m,i,j}^t \) is obtained at each node using:
\[ q_{m,i,j}^t = \sum_{i=1}^{N_m} q_{m,i,j}^t \]
and
\[ d_{m,i,j}^t = \sum_{i=1}^{N_m} d_{m,i,j}^t y_{m,i} \]
\[ b_{m,i,j}^t = \sum_{i=1}^{N_m} b_{m,i,j}^t y_{m,i}^T \]

Then \( q_{m,i,j}^t \) is added up with the value received from the previous node and the resulting quantity is sent to the next node
\[ q_j^t = q_j^t + q_{m,i,j}^t \]

The value of \( d_j^t \) and \( b_j^t \) are also updated as follows
\[ d_j^t = d_j^{t-1} + d_{m,i,j}^t \]
\[ b_j^t = b_j^{t-1} + b_{m,i,j}^t \]
This procedure is continued until the forward pass is finished. Then, during the backward pass, the value of parameters is estimated at each node using:

\[
\alpha_j^t = \alpha_j^0 + q_j^t \\
\beta_j^t = \beta_j^0 + q_j^t \\
m_j^t = \frac{\beta_j^0 m_j^0 + a_j^t}{\beta_j^t} \\
\Sigma_j^t = \Sigma_j^0 + b_j^t + \beta_j^0 m_j^0 m_j^0 T - \beta_j^t m_j^t m_j^t T \\
\nu_j^t = \nu_j^0 + q_j^t
\]

This procedure requires the computation of \( q_j^t \) in \( M \) steps due to the fact that:

\[
d_j^t = \sum_{m=1}^{M} q_{m,j} \\
da_j^t = \sum_{m=1}^{M} a_{m,j} \\
b_j^t = \sum_{m=1}^{M} b_{m,j}
\]

This two pass procedure is also repeated in next iterations. Note that using the vector of sufficient statistics, each node can compute all of the required parameters based on the variational approximation method. Besides, since only this sufficient statistics vector is to be transmitted, the communication requirement of the proposed algorithm is quite modest.

4. A Distributed Variational Bayesian Algorithm

This section proposes a fully distributed variational Bayesian algorithm (DVBA) that eliminates the need for forward and backward message passing processes in the implementation of the standard variational Bayesian approximation discussed above. The DVBA cycles through the nodes of the network and implements the variational Bayesian estimation incrementally at each node using only the local data at each node and summary statistics passed from the previous node in the cycle. Similar to the standard VB algorithm [18], DVBA is guaranteed to converge to a local maximum as described in the next section. Moreover, in practice because of its incremental form, DVBA often converges much more rapidly than that of the standard VB algorithm.

At each node \( m \), we consider \( \hat{f}_m (\theta_m, z_m) \) as a variational approximation for \( f_m (\theta_m, z_m | y_m) \). Assuming that the data set of all nodes are independent and identically distributed (i.i.d.), we have

\[
\hat{f}(\theta, z) = \prod_{m=1}^{M} \hat{f}_m (\theta_m, z_m) \\
f(\theta, z | y) = \prod_{m=1}^{M} f_m (\theta_m, z_m | y_m)
\]

Now, we assume that the posterior distribution \( \hat{f}_m (\theta_m, z_m) \) can be factorized as

\[
\hat{f}_m (\theta_m, z_m) = \hat{f}_{m|z} (\theta_m) \hat{f}_{z|m} (z_m)
\]

At each node \( m \), the proposed DVBA tries to estimate the posterior distributions \( \hat{f}_{m|z} (\theta_m) \) and \( \hat{f}_{z|m} (z_m) \) such that the lower bound of the log likelihood \( F \) is maximized.

The proposed DVBA performs as follows. Initialize \( \{ \alpha_{m,0}^0, m_{m,0}^0 \}, \{ \beta_{m,0}^0 \}, \{ \nu_{m,0}^0 \} \) and \( \{ \Sigma_{m,0}^0 \} \) at some chosen value. Assume that the algorithm proceeds in a cyclic manner. The following processing and communication are carried out at each node in sequence. At iteration \( t + 1 \) node \( m \) receives \( q_j^t, a_j^t \) and \( b_j^t \), from the immediate previous node. The node then computes the hyperparameters using:

\[
\alpha_j^t = \alpha_j^0 + q_j^t \\
\beta_j^t = \beta_j^0 + q_j^t \\
m_j^t = \frac{\beta_j^0 m_j^0 + a_j^t}{\beta_j^t} \\
\Sigma_j^t = \Sigma_j^0 + b_j^t + \beta_j^0 m_j^0 m_j^0 T - \beta_j^t m_j^t m_j^t T \\
\nu_j^t = \nu_j^0 + q_j^t
\]

The mean value of the means \( \mu_j \) and the precisions \( T_j \) at this node are obtained through

\[
E[\mu_j^t] = m_j^t \\
E[T_j^t] = \nu_j^t (\Sigma_j^t)^{-1}
\]

Then \( q_{m,i,j}^{t+1} \) is computed using Eq. (4) and the local values of the sufficient statistics vector is then calculated as:

\[
d_{m,i,j}^{t+1} = \sum_{i=1}^{N} d_{m,i,j}^{t+1} \\
d_{m,i,j}^{t+1} = \sum_{i=1}^{N} d_{m,i,j}^{t+1} y_{m,i} \\
b_{m,i,j}^{t+1} = \sum_{i=1}^{N} d_{m,i,j}^{t+1} y_{m,i} y_{m,i}^T
\]

Finally, node \( m \) updates its mixing probabilities and sufficient statistic vectors \( q_j^t, a_j^t \) and \( b_j^t \) according to

\[
x_{m,j}^{t+1} = \frac{q_j^t}{N_m}
\]
The DVBA algorithm stops whenever the difference in log-likelihood function corresponding to this node is calculated using the data of node \( m \) and iteration \( t \). Here, after updating the parameters using the data of node \( m \), the value of local log-likelihood function corresponding to this node is calculated using:

\[
L^m(\theta) = \sum_{m=1}^{M} \sum_{i=1}^{N} \log \left( \sum_{j=1}^{I} \pi_{m,j} N(y_{m,i} | \mu_{m,j}, \Sigma_{m,j}) \right)
\]

The DVBA algorithm stops whenever the difference \( L^m(\theta^{t+1}) - L^m(\theta^t) \) becomes less than a convergence threshold \( \epsilon \). Instead of likelihood variations, parameter variations can be also used as another stopping criterion. Note that this recursive equation requires that log-likelihood value of node \( m \) at the previous iteration should be available at the current iteration.

This algorithm requires each node to know in advance number of Gaussian mixtures. Further work can be done to estimate this number as well as the component parameters using a distributed VB algorithm.

The energy consumed for communicating a bit in a node can be many orders of magnitude greater than that of required for a single local computation; hence the required energy consumed in communication is important in sensor networks. Now assume that the nodes are distributed over a squared area with nodes on a uniform grid. The scenario where the nodes are distributed randomly over a squared area is equivalent to that with a uniform grid. Denoting \( N_b \) as the number of bytes communicated between two nodes per time step, it can be found that the amount of communications in bytes for the centralized method in which all nodes send their data to the center of the network becomes \( \sqrt{M} (1 + 2 + \ldots + \sqrt{M}/2) N_b = O(M^{3/2}) \). The worst case in this method is that the centralized unit is not in the center of the network.; in fact it is at the end of the area. The communication in bytes for such a case is \((M - 1 + M - 2 + \ldots + 1) N_b = O(M^2)\). Once the centralized unit receives all data, it can run the standard VB algorithm.

The proposed DVBA executes the communication and computations iteratively. The communication cost is related to the number of loops, i.e., the accuracy of the estimated results. Denoting \( T \) as the number of loops, the communication in bytes for DVBA is \( MN_b T = O(M) \). Therefore, unlike the centralized method the proposed DVBA is scalable.

4.1 Initialization

We first apply the distributed k-means clustering algorithm [3] which returns estimates of \( \bar{\pi}_j, \beta_j \) and \( \Sigma_j \). The posterior for the means is a Gaussian with mean \( \bar{\mu}_j \) which we initialize to \( m_j = \bar{\mu}_j \). The posterior for the precision is a Wishart with mean of \( E[T_j] = \nu_j (\Sigma_j)^\top \). Hence, \( \Sigma_j \) is initialized as \( \Sigma_j = \nu_j \Sigma_j \). The posterior mixing distribution is parameterized by the coefficients \( \alpha_j \). Since \( \frac{\alpha_j}{\bar{\alpha}_j} \) represents the weight of component \( j \) and \( \alpha_j = \alpha_j^0 + q_j \), neglecting \( \alpha_j^0 \), the initialization of \( \alpha_j = N \bar{\pi}_j, \beta_j \) and \( \nu_j \) are also initialized as \( \beta_j = N \bar{\pi}_j \) and \( \nu_j = N \bar{\pi}_j \), respectively. In the above relations, the RHS quantities are the k-means estimates.

5. Convergence Analysis

As mentioned before, in the variational approximation, the variational free energy \( F(f_\theta, f_\bar{\theta}) \) constructs a lower bound on the log likelihood value \( L \). At each iteration of variational Bayesian algorithm, the posterior distributions of \( f_\theta \) and \( f_\bar{\theta} \) are chosen such that the value of \( F \) is optimized. Note that \( f_\theta \) and \( f_\bar{\theta} \) denote the density of the missing variables \( z \) and the parameters vector \( \theta \), respectively. The negative free energy may be written as

\[
F(f_\theta, f_\bar{\theta}) = \int_z f_\theta(\theta)f_\bar{\theta}(z) \log \frac{f(y, z, \theta)}{f_\theta(\theta)f_\bar{\theta}(z)} d\theta = \log f(y) - KL(f_\theta || f_\bar{\theta}) = L(y) - KL(f_\theta || f_\bar{\theta})
\]

In which \( KL(f_\theta || f_\bar{\theta}) \) denotes the Kullback-Leibler divergence between \( f_\theta \) and \( f_\bar{\theta} \). Figure 2 shows the above relation graphically.
Assuming that the data set of different nodes are independent, we have \( \hat{f}_0 = \prod_{m=1}^{M} \hat{f}_{\theta_m} \) and \( \hat{f}_c = \prod_{m=1}^{M} \hat{f}_{\theta_m} \). Therefore, the \( F \) function can be written as a sum of local variational free energy values over all of nodes. Assume that \( F_m \) denotes the local variational free energy at node \( m \), can be written as:

\[
F \left( \hat{f}_0, \hat{f}_c \right) = \sum_{m=1}^{M} F_m \left( \hat{f}_{\theta_m}, \hat{f}_{\theta_m} \right)
\]

where

\[
F_m \left( \hat{f}_{\theta_m}, \hat{f}_{\theta_m} \right) = \int \sum_{zM} \hat{f}_{\theta_m} \left( \theta_m \right) \hat{f}_{\theta_m} \left( \theta_m \right) f \left(y_m, \hat{z}_m, \theta_m \right) d\theta_m
\]

To improve convergence of the DVBA, it should be shown that the value of \( F \) before reaching to its maximum is increased at each node. In other words, it should be exposed that \( F \) is a non-decreasing function in the DVBA. In this case, assuming that \( F^* \) represents a local maximum value and that initial values of the parameters are sufficiently near the optimal values, the DVBA will eventually converge to its fixed point.

In the DVBA, at each node \( m \), the value of \( \hat{f}_{\theta_m} \) is first calculated to maximize \( F_m \) with respect to \( \hat{f}_{\theta_m} \), and then the value of \( \hat{f}_{\theta_m} \) is calculated to maximize \( F_m \) with respect to \( \hat{f}_{\theta_m} \). In fact, by updating the value of \( q_{m,i,j} \) using equations (4), the value of \( F_m \) is maximized with respect to \( \hat{f}_{\theta_m} \) and using the equations (5)-(9), (10), it is maximized with respect to \( \hat{f}_{\theta_m} \).

Therefore the DVBA, cycles through the nodes and increases the value of \( F_m \) at each node \( m \), while the value of \( F_n \), \( n \neq m \) will not change. Hence, the total value of \( F \) will also increase. As a result, DVBA is a non decreasing algorithm and assuming that \( F^* \) is a local upper bound for \( F \), this algorithm will finally converge to the optimal parameter values.

6. Simulation Results

This section is devoted to show the performance of the proposed DVBA algorithm. Two different cases have been considered as given below.

6.1 Case 1: Environmental Modeling

Here, we use a network of 50 nodes, each containing 100 data points, to evaluate performance of DVBA in environmental modeling. It is assumed that each node in the network senses an environment that can be described as a mixture of some elementary conditions. The measurements are thus statistically modeled with a mixture of Gaussians; each Gaussian component corresponding to one of the elementary conditions. We first consider a one dimensional data set simulated from a 4-component Gaussian mixture given by

\[
0.288N(0, 0.04) + 0.26N(-1.5, 0.25) + 0.171N(2.2, 11.56) + 0.281N(3.3, 0.25)
\]

In our approach, the means \( \mu_j \) were all set to zero. The parameter \( \beta_j \) has chosen to be 0.05, to give a broad prior over the mean. The initial values for the degree of freedom \( \nu_j \) and the scale matrix \( \Sigma_j \) were taken to be 2 and 0, respectively. The mixing weights were given a Dirichlet prior with the initial \( \alpha \)’s set to 0.

The estimated parameters obtained through DVBA are presented in Table 1. The average values of estimated means, variances and the standard deviation of these parameters obtained for each node are presented in this table. As it is seen, very good estimates of the true values have been obtained. The small standard deviation values show that the estimated means and variances at all 50 nodes are almost the same. The true density of the simulated data set and density fitted by the DVBA at node 1 are also illustrated in Fig. 3. This figure shows that the estimated probability density function closely approximates the true one.

Table 1: Fitted means and variances using the DVBA.

<table>
<thead>
<tr>
<th>Component</th>
<th>Average of means</th>
<th>Std of means (×10−3)</th>
<th>Average of variances</th>
<th>Std of variances (×10−3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>−0.0019</td>
<td>0.0163</td>
<td>0.0402</td>
<td>0.0006</td>
</tr>
<tr>
<td>2</td>
<td>−1.5072</td>
<td>0.0385</td>
<td>0.2556</td>
<td>0.0104</td>
</tr>
<tr>
<td>3</td>
<td>2.2480</td>
<td>0.8429</td>
<td>11.2847</td>
<td>0.6535</td>
</tr>
<tr>
<td>4</td>
<td>3.3040</td>
<td>0.0557</td>
<td>0.2565</td>
<td>0.0236</td>
</tr>
</tbody>
</table>

Figure 4 illustrates the log-likelihood value of DVBA as a function the number of transmitted bits. This number corresponds to the number of messages passed between nodes. Number of messages passed between nodes is related to the number of transmitted bits as:

\[
K\text{bits communicated } = \frac{MsgNo}{1024} \left( 64J \left( 1 + d + d^2 \right) \right)
\]

where \( MsgNo \) stands for number of messages. Here, a 64-bit precision is assumed and the dimension of sufficient statistics vector \( s_j = \{ a_j, b_j \} \) transmitted at each iteration is equal to \( 1 + d + d^2 \), in which \( d \) is the dimension of data.

Convergence criterion is assumed to be the local log-likelihood variations at each node. In other words, when likelihood variations becomes less than a convergence

Fig. 2 The quantity \( F \) provides a lower bound on the log likelihood \( L(y) \), with the difference being given by the Kullback-Leibler divergence \( KL(f_0||f_0) \).
threshold $\epsilon$, the algorithm will be stopped. Here, the convergence threshold is assumed to be $\epsilon = 10^{-8}$. At this simulation, the DVBA has converged after 39 iterations. Other simulations have shown similar results.

A 2D data set is used for the next simulation. A sensor network with 50 nodes ($M = 50$) is considered again in which each node has 100 data observations ($N_m = 100$). The observations are generated from three Gaussian components ($J = 3$) distributed in Fig. 5. Each of these components represents a 2D environmental data cluster. In the first 20 nodes, 80% of observations come from the first Gaussian component and other 20% of observations evenly come from the other two Gaussian components. In the next 15 nodes, 80% of observations come from the second component and in the last 15 nodes 80% of observations come from the third component. Other observations of these nodes come evenly from the other two Gaussian components.

For the purpose of comparison, we performed three different tests. First, the standard variational Bayesian algorithm is executed in a center unit using all data from 50 nodes. In the second test, this algorithm is also conducted in each of nodes employing VB algorithm individually using only local data resulting obviously in different estimates. The estimated mean values of each component in all 50 nodes are shown in Fig. 6. For the sake of simplicity, this figure only shows the second element of the mean vector. As it is seen, the curve of the estimated mean values of each component has a flat section over an interval number of nodes. These flat sections correspond to the nodes that could yield reliable estimates of the mean value. Other nodes cannot properly estimate the parameters due to limited observations used.

In the third test, the proposed DVBA is executed over the nodes of this network. Second elements of the estimated mean vectors in 50 nodes are shown in Fig. 7. It can be seen that the estimated mean values in all nodes are very close to their true values (the true values are 2, 0 and $-2$). The true and estimated parameters of the components using DVBA are shown in Tables 2 and 3, respectively. As seen, good estimates of the true values have been obtained. The values offered in these tables are the mean value of the estimated parameters at all nodes of the network. Standard deviations of the estimated parameters at various nodes are in order of $10^{-5}$. The small values of the standard deviations imply that the estimated values obtained at each node are almost the
Fig. 7 Three estimated mean values using DVBA in the network with 50 nodes.

Table 2 True mean and covariance matrices for the 2D data set.

<table>
<thead>
<tr>
<th>Component</th>
<th>Mean vector</th>
<th>Covariance matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[0, -2]</td>
<td>2 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 0.2</td>
</tr>
<tr>
<td>2</td>
<td>[0, 0]</td>
<td>2 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 0.2</td>
</tr>
<tr>
<td>3</td>
<td>[0, 2]</td>
<td>2 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 0.2</td>
</tr>
</tbody>
</table>

Table 3 Fitted mean and covariance matrices using the DVBA.

<table>
<thead>
<tr>
<th>Component</th>
<th>Mean vector</th>
<th>Covariance matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[0.0004, -1.9891]</td>
<td>2.0912 0.0112</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0112 0.2077</td>
</tr>
<tr>
<td>2</td>
<td>[0.0297, -0.0169]</td>
<td>2.0582 -0.0292</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.0292 0.1836</td>
</tr>
<tr>
<td>3</td>
<td>[0.0553, 1.9944]</td>
<td>1.9480 -0.0201</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.0201 0.1970</td>
</tr>
</tbody>
</table>

Fig. 8 Log-likelihood values of DVBA and standard VB algorithms.

The DVBA algorithm has converged after 24 iterations while the standard VB algorithm has converged in 109 iterations. Easily observed that convergence rate of the DVBA is superior to that of the standard VB algorithm. Computation time of DVBA is also considerably less than that of the standard VB algorithm. Using a dual-core 1.86 GHz CPU, the standard VB algorithm has converged in 184 seconds while the DVBA has converged in 46 seconds. Table 4 shows the estimated parameter values using the standard VB algorithm. As it is seen, the estimated values are the same as the values obtained through the DVBA. Therefore, the DVBA has performed as good as the centralized VB but in less number of iterations and based on a distributed structure. Mean value of the estimated Gaussian densities by the DVBA and Gaussian densities estimated by the centralized VB algorithm are shown in Fig. 9. As it is seen, the estimated densities by these two algorithms are almost the same. True density of the simulated data is also shown in this figure.

6.2 Case 2: Distributed Target Classification

With the increasing use of camera surveillance in public areas, the need for automated surveillance solutions is rising. A particular problem is camera surveillance in wide areas, such as airports, shopping centers, etc. Such areas typically cannot be fully observed by a single camera, and surveillance of such places relies on a network of sparsely distributed cameras.
In this particular setting the problem of tracking persons across all cameras is difficult. Someone is first observed by one camera, then he/she is outside of sight of any camera, and later on he reappears at another camera. We would like to know whether the two observed persons are in fact the same individual.

In this part, we employ DVBA in a distributed system for target recognition. In this system, each camera is a stand alone tracking unit, which stores its own observations and exchanges only limited data with other cameras. The local observations in combination with the exchanged data allow each camera to learn its own local model.

Similar to other approaches [19] we use appearance cues such as average color, or length to find the correspondence between observations and persons. Since the same person will appear differently each time that she/he is observed, we model the observations as a stochastic process. We assume that the observations are samples drawn from a Gaussian distribution with person specific parameters, which are constant over time. In a system where J persons are monitored, observations of all persons are generated by a Gaussian Mixture Model (GMM) with J components. Here, we consider the learning of the parameters of the GMM with the proposed DVBA. Given the learned GMM we assign the most likely person to each observation.

We have performed a series of tests with the presented DVBA. The performance of the algorithm will be compared with that of a standard VB implementation. We evaluate the algorithms on artificially generated data.

A set of 1000 observations from 5 persons, which are distributed over several cameras is used as a standard set. Every observation consists of a 3-dimensional appearance vector. The observations are randomly distributed over the cameras according to a uniform pdf. The difficulty of the generated data is measured by the c-separation and eccentricity values [20]. An increasing difficult recognition problem is indicated by increasing eccentricity or decreasing c-separation values. The dataset has a c-separation of 1 and an eccentricity of 3. Several datasets are generated to investigate the performance of the algorithm by variations of the number of cameras, and the distribution of the observations over the cameras.

Evaluation The evaluation criteria should reflect two aspects of proper clustering. It is described that (i) all observations within a single reconstructed cluster belong to a single person, and (ii) all observations of a single person are grouped together in a single reconstructed cluster. These criteria are analogous to the precision and recall criteria often used in Information Retrieval settings. In order to evaluate both systems on one parameter we use the F1-measure defined in (13).

Because the considered clustering problem is unsupervised, the true and proposed clusters are arbitrary ordered. Therefore, we define the precision (11) and recall (12) for the proposed cluster over the best match with a real cluster. Importantly precision and recall have to be considered jointly, because it is trivial to gain recall of 1. This is done by clustering all observations into one cluster. However, this will result in a very low precision. The F1 measure (13) is the harmonic mean of precision and recall and will penalize for such cases.

\[
Pr = \frac{1}{J} \sum_{i=1}^{J} \frac{\max(\hat{C}_s \cap C_i)}{|\hat{C}_s|}
\]

\[
Rc = \frac{1}{J} \sum_{i=1}^{J} \frac{\max(\hat{C}_s \cap C_i)}{|C_i|}
\]

\[
F1 = \frac{2 \cdot Pr \cdot Rc}{Pr + Rc}
\]
of cameras. As it is shown in this figure, convergence rate of the DVBA is much faster than that of the standard VB which is an important advantage.

7. Conclusion

In this paper, a distributed variational Bayesian algorithm (DVBA) was developed for density estimation and clustering in a sensor network. DVBA is a distributed algorithm that performs local computations on the sensor data at each node and passes a small set of sufficient statistics from node-to-node in the iteration process. Therefore, the DVBA’s communication requirements are quite modest. Meanwhile, convergence rate of the DVBA is superior to that of the standard VB algorithm due to its incremental nature. We have shown that DVBA converges to a stationary point of the log likelihood function, usually a local maximum. The proposed DVBA is a general distributed data mining algorithm for density estimation and clustering of the data distributed over the nodes of a network.

As discussed in Sect. 4, the variational approach allows the simultaneous estimation of the component parameters and the model complexity. Therefore, distributed estimation of the model complexity as well as the component parameters can be studied in the future. Another field of research is exploring whether other distributed data mining problems can be formulated using the variational optimization framework, and whether the variational method leads to communication-efficient techniques for solving these problems. These issues are currently under investigate and will be reported in the near future.

References

Mohammad B. Menhaj received his Ph.D. degree from school of electrical and computer engineering at OSU in 1992. After completing one year with OSU as postdoctoral fellow, in 1993, he joined the Amirkabir University of Technology, Tehran, Iran. December 2000 to Aug. 2003, he was with school of Electrical and Computer Engineering and Department of Computer Science at OSU as visiting faculty member and research scholar. He is author and co-author of more than 250 technical papers, and four books: Fundamentals of Neural Networks, 1998, and Application of Computational Intelligence in Control, 1998, Neural Networks, 2000, and Fuzzy Computations, 2007, all in Persian. His main research interests are: theory of computational intelligence, learning automata, adaptive filtering and their applications in control, power systems, image processing, pattern recognition, and communications, and other areas of interests are: theory of rough set and knowledge discovery. Email: menhaj@aut.ac.ir.

Mehdi Karrari received his Ph.D. degree in Control Engineering from Sheffield University in 1991. Since then, he has been with the Amirkabir University of Technology, Tehran, Iran. He is author and co-author of more than 100 technical papers, and one book: Power System Dynamics and Control, 2004, in Persian. His main research interests are: power system modeling, modeling and identification of dynamic systems, large-scale and distributed systems. Email: karrari@aut.ac.ir.