Distributed EM Algorithms for Density Estimation and Clustering in Sensor Networks

Robert D. Nowak, *Member, IEEE*

Abstract

This paper considers the problem of density estimation and clustering in distributed sensor networks. 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. This paper presents a distributed EM algorithm for estimating the Gaussian components, which are common to the environment and sensor network as a whole, as well as the mixing probabilities which may vary from node to node. The algorithm produces an estimate (in terms of a Gaussian mixture approximation) of the density of the sensor data without requiring the data to be transmitted to and processed at a central location. Alternatively, the algorithm can be viewed as an distributed processing strategy for clustering the sensor data into components corresponding to predominant environmental features sensed by the network. The convergence of the distributed EM algorithm is investigated, and simulations demonstrate the potential of this approach to sensor network data analysis.

I. Introduction

The slogan "the network is the sensor," coined at Oakridge National Labs, aptly captures the sensor networking spirit — massively distributed, small devices, networked for communication and equipped with sensing and processing capabilities, that give us a new eye with which to explore our universe. This vision begs the question: What is the network sensing? This paper proposes a new framework for distributed data exploration in sensor networks. Density estimation and unsupervised clustering are central first steps in exploratory data analysis. They aim to answer the question: What are the basic patterns and structures in the measured data? Both problems can also be naturally posed as maximum likelihood estimation problems, and have been widely studied under the assumption that data are stored and processed at a central location. Here that assumption is changed; we assume that the data are not centralized, but rather are distributed across a collection of networked devices. Moreover, it is assumed that the cost (in terms of power or related resources) of computation at each node is much less than the cost of communication between nodes, which makes the option of centralized data processing very expensive and unattractive. The approach pursued here is based on the following model. It is assumed that each node in the network senses an environment that can

This work was supported by the National Science Foundation, grant nos. MIP–9701692 and ANI-0099148, the Office of Naval Research, grant no. N00014-00-1-0390, and the Army Research Office, grant no. DAAD19-99-1-0290.

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. A distributed EM-type algorithm is developed for estimating the Gaussian components, which are common to the environment and sensor network as a whole, as well as the mixing probabilities which may vary from node to node. This amounts to an unsupervised clustering of the data into components corresponding to common environmental conditions.

The distributed EM algorithm developed here can be viewed as an application and adaptation of the incremental EM algorithm [6]. The incremental EM algorithm was proposed as a fast alternative to the standard EM algorithm for mixture density estimation. To the best of our knowledge, the use of incremental EM for distributed processing and estimation has not been previously considered. The distributed nature of the sensor network problem places emphasis on the trade-off between local processing and communication. In this paper, it is assumed that local processing is much less expensive than communication. Therefore, the distributed EM algorithm proposed here is a variation on the incremental EM algorithm that aims to reduce the number of iterations, and hence the number of communications required. The incremental EM algorithm was shown to converge to a fixed point in [6], and more recently it was shown that the incremental EM algorithm and the standard EM algorithm have the same fixed points [8]. The rate of convergence is investigated in this paper. The analysis herein demonstrates that in certain cases incremental EM converges at a linear rate to a maximum likelihood point. In the sensor network context, the convergence rate analysis is vital since it guarantees that the algorithm will converge (to within a prespecified tolerance) in a reasonable number of computations and communications.

The paper is organized as follows. In Section II, the basic problem statement is given, and measurement and data models are defined. Section III reviews the standard EM algorithm for mixture density estimation, and a distributed algorithm for its implementation in sensor network applications is presented. Section IV develops a fully distributed EM algorithm that aims to reduce the number of iterations and hence communications required to compute the maximum likelihood estimate. Section V presents an analysis of the convergence rate of the distributed EM algorithm. This analysis demonstrates theoretically the benefits of the distributed EM algorithm compared to the centralized EM algorithm. Section VI discusses some of the issues surrounding communication requirements and scalability of the distributed EM algorithm. It is shown that under reasonable conditions distributed EM is feasible in high density wireless networks. Section VII studies simulated sensor networking problems and the distributed EM algorithm. Concluding remarks are made in Section VIII.

II. Problem Statement

Assume that we have M nodes, $1, \ldots, M$. The m-th node senses and records N_m iid measurements $y_{m,1}, \ldots, y_{m,N_m}$. The iid assumption implies that the environmement is stationary and unchanging dur-

ing the course of the measurement process. Let $\mathcal{N}(\mu, \Sigma)$ denote the Gaussian density function with mean μ and covariance Σ. The measurements are assumed to obey a Gaussian mixture distributions of the form

$$y_{m,i} \sim \sum_{j=1}^{J} \alpha_{m,j} \mathcal{N}(\mu_j, \Sigma_j), \ i = 1, \ldots, N_m.$$

where the mixing parameters $\{\alpha_{m,j}\}$ are potentially unique at each node, but the means $\{\mu_j\}$ and covariances $\{\Sigma_j\}$ are common at all nodes. All parameters are unknown. The goal of this work is a distributed algorithm for estimation of these parameters from the data $y = \{y_{m,i}\}$. Figure 1 depicts a sensor network in an inhomogeneous environment. Figure 3 shows sensor network data in a simulated experiment.

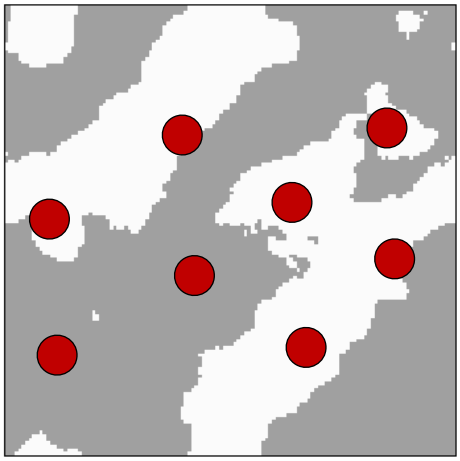

Fig. 1. Sensor network in an inhomogeneous environment. Discs represent nodes in the sensor network. Background represents spatially varying environmental conditions being sensed by the nodes.

Define $\phi \equiv \{\mu_j, \Sigma_j\}_{j=1}^{J}$, the set of means and covariances. For each node $m = 1, \ldots, M$ define $\alpha_m \equiv \{\alpha_{m,j}\}_{j=1}^{J}$, the mixing probabilities for node m. Finally, define $\theta = \phi \cup \{\alpha_m\}_{m=1}^{M}$. This paper describes a distributed algorithm for computing a maximum likelihood estimate; i.e., θ maximizing the log-likelihood function

$$l_y(\theta) \equiv \sum_{m=1}^{M} \sum_{i=1}^{N_m} \log \left(\sum_{j=1}^{J} \alpha_{m,j} \mathcal{N}(y_{m,i} | \mu_j, \Sigma_j) \right), \tag{1}$$

where $\mathcal{N}(y|\mu, \Sigma)$ denotes the evaluation of a Gaussian density with mean μ and covariance Σ at the point y.

The data at each node are assumed to be statistically independent in this paper, but this assumption can be relaxed. If the data are (spatially or temporally) correlated, then the simple "independent" likelihood model can still be employed by interpreting it as a *pseudolikelihood* [16]. Under mild conditions the maximum pseudolikelihood estimates tend to the true maximum likelihood estimates as the number of data tends to infinity.

III. The Standard EM Algorithm

Introduce a set of missing data $z = \{z_{m,i}\}$. Each $z_{m,i}$ takes on a value from the set $\{1, \ldots, J\}$, where $z_{m,i} = j$ indicates that $y_{m,i}$ was generated by the j-th mixture component. In other words,

$$y_{m,i}|z_{m,i} = j \;\sim\; \mathcal{N}(\mu_j, \Sigma_j).$$

This is the usual choice of missing data in EM approaches to mixture modeling. The quantity $x = (y, z)$ is referred to as the complete data for y [1, 2].

Define $\phi^t \equiv \{\mu_j^t, \Sigma_j^t\}_{j=1}^J$, the set of means and covariances at the t-th iteration of the EM algorithm. For each node $m = 1, \ldots, M$ define $\alpha_m^t \equiv \{\alpha_{m,j}^t\}_{j=1}^J$, the mixing probabilities for node m at the t-th iteration. Finally, define $\theta^t = \phi^t \cup \{\alpha_m^t\}_{m=1}^M$. Define the conditional expectation

$$Q(\theta, \theta^t) = E[\log p(x|\theta)\,|y, \theta^t], \tag{2}$$

where $p(y, z|\theta)$ denotes the joint density of y and z with parameters θ (see equation (8) for a more detailed definition). This is a standard Gaussian mixture missing data formulation and it is easy to verify that

$$Q(\theta, \theta^t) \;=\; \sum_{m=1}^{M} \sum_{i=1}^{N_m} \sum_{j=1}^{J} w_{m,i,j}^{t+1} \left(\log \alpha_{m,j} + \log \mathcal{N}(y_{m,i}|\mu_j, \Sigma_j)\right),$$

where

$$w_{m,i,j}^{t+1} = \frac{\alpha_{m,j}^t \mathcal{N}(y_{m,i}|\mu_j^t, \Sigma_j^t)}{\sum_{k=1}^J \alpha_{m,k}^t \mathcal{N}(y_{m,i}|\mu_k^t, \Sigma_k^t)}.$$

From this it is easy to see that the E-Step, computing the conditional expectation $Q(\theta, \theta^t)$, boils down to computing $\{w_{m,i,j}^{t+1}\}$. The M-Step is

$$\theta^{t+1} = \arg\max_{\theta} Q(\theta, \theta^t),$$

and has a simple closed form expression. Specifically, for each node $m = 1, \ldots, M$ and for $j = 1, \ldots, J$

$$\alpha_{m,j}^{t+1} \;=\; \frac{1}{N_m} \sum_{i=1}^{N_m} w_{m,i,j}^{t+1}.$$

And for each component $j = 1, \ldots, J$

$$\begin{aligned}
\mu_j^{t+1} &= \frac{\sum_{m=1}^M \sum_{i=1}^{N_m} w_{m,i,j}^{t+1} y_{m,i}}{\sum_{m=1}^M \sum_{i=1}^{N_m} w_{m,i,j}^{t+1}}, \\
\Sigma_j^{t+1} &= \frac{\sum_{m=1}^M \sum_{i=1}^{N_m} w_{m,i,j}^{t+1} (y_{m,i} - \mu_j^{t+1})(y_{m,i} - \mu_j^{t+1})'}{\sum_{m=1}^M \sum_{i=1}^{N_m} w_{m,i,j}^{t+1}}.
\end{aligned}$$

The EM algorithm usually converges to a local maximum of the log likelihood function (although convergence to a saddle-point is possible). It can be shown that if the Gaussian components are well separated, then EM converges more rapidly than gradient methods [4]. In such cases the convergence rate is superlinear,

comparable to that of Newton-type methods [5]. On the other hand, EM is a conservative algorithm in general, with better stability properties than more aggressive schemes such as Newton's method. These facts make EM a good choice for mixture estimation, and distributed (and unsupervised) applications like those arising in sensor networks, especially.

In anticipation of a distributed version of this EM algorithm, define the local "summary" quantities

$$\begin{aligned} w_j^t &= \sum_{m=1}^{M}\sum_{i=1}^{N_m} w_{m,i,j}^t, \\ a_j^t &= \sum_{m=1}^{M}\sum_{i=1}^{N_m} w_{m,i,j}^t y_{m,i} \\ b_j^t &= \sum_{m=1}^{M}\sum_{i=1}^{N_m} w_{m,i,j}^t y_{m,i}^2. \end{aligned}$$

Notice that with these summaries the M-Step calculations for the means can be written as

$$\begin{aligned} \mu_j^{t+1} &= \frac{a_j^t}{w_j^t}, \\ \Sigma_j^{t+1} &= \frac{b_j^t}{w_j^t} - \mu_j^{t+1}(\mu_j^{t+1})', \end{aligned}$$

where the superscript $'$ denotes matrix transposition (although the measurements may be scalar in which case this is unnecessary).

A distributed implementation of the standard EM algorithm is obtained as follows. Assume that all nodes have the current parameter estimate θ^t. The next EM iterate θ^{t+1} can be computed by performing two message passing cycles through the nodes. Each message passing operation involves the transmission of the (partial) sufficient statistics from one node to another. The message passing follows a prescribed sequence through the nodes, and for illustration assume that the messages are passed from node to node in the order $1, 2, \ldots, M, M-1, M-2, \ldots, 1$. To begin, initialize the sufficient statistic $s^t = \{w_j^t, a_j^t, b_j^t\}_{j=1}^J$ to zero. Each node computes its local updates for the sufficient statistic:

$$\begin{aligned} w_{m,j}^t &= \sum_{i=1}^{N_m} w_{m,i,j}^t, \\ a_{m,j}^t &= \sum_{i=1}^{N_m} w_{m,i,j}^t y_{m,i} \\ b_{m,j}^t &= \sum_{i=1}^{N_m} w_{m,i,j}^t y_{m,i}^2. \end{aligned}$$

Note that these updates are computed from $\{y_{m,i}\}$ and θ^t, available locally at each node. In the forward path,

in succession from $1, \ldots, M$, node m increments $\{w_j^t, a_j^t, b_j^t\}_{j=1}^J$ according to

$$\begin{aligned} w_j^t &= w_j^t + w_{m,j}^t, \\ a_j^t &= a_j^t + a_{m,j}^t \\ b_j^t &= b_j^t + b_{m,j}^t, \end{aligned}$$

and passes $s^t = \{w_j^t, a_j^t, b_j^t\}_{j=1}^J$ to node $m+1$. This process takes advantage of the fact that the E-Step can be separated into M separate expectations followed by accumulation. At the last node, M, the complete sufficient statistic for the M step are in hand. The complete sufficient statistic is passed back in the reverse order, from $M, \ldots, 1$. All M nodes now have the sufficient statistic and can compute the M-Step to obtain θ^{t+1}. Note that each iteration of the EM algorithm requires the transmission of $2M-2$ messages of dimension $\dim(s^t)$.

IV. A Distributed EM Algorithm for Sensor Networks

This section proposes a fully distributed EM (DEM) algorithm that eliminates the need for the forward and backward message passing process in the distributed implementation of the standard EM algorithm discussed above. The DEM algorithm cycles through the network and performs incremental E and M steps 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 EM algorithm [3] above, DEM is guaranteed to monotonically converge to a local maximum (or saddle point) as described in the next section. Moreover, because of its incremental form, DEM often converges much more rapidly than the standard EM algorithm in practice.

Specifically, the DEM algorithm operates as follows. Initialize $\{\mu_j^0, \Sigma_j^0, \alpha_{m,j}^0\}$ at some chosen values (possibly random) and set the quantities w_j^0, a_j^0 and b_j^0 to zero. Assume that the algorithm proceeds in a cyclic fashion (i.e., messages are passed between nodes in the order $1, 2, \ldots, M, 1, 2 \ldots, M, \ldots$); other non-cyclic possibilities are also possible. The following processing and communication is carried out at each node in succession. At iteration $t+1$ node m receives w_j^t, a_j^t and b_j^t from the preceding node. The node then computes the means and variances

$$\begin{aligned} \mu_j^t &= \frac{a_j^t}{w_j^t}, \\ \Sigma_j^t &= \frac{b_j^t}{w_j^t} - \mu_j^t(\mu_j^t)', \end{aligned} \tag{3}$$

and

$$w_{m,i,j}^{t+1} = \frac{\alpha_{m,j}^t \mathcal{N}(y_{m,i}|\mu_j^t, \Sigma_j^t)}{\sum_{k=1}^J \alpha_{m,k}^t \mathcal{N}(y_{m,i}|\mu_k^t, \Sigma_k^t)}. \tag{4}$$

Then node m updates its mixing probabilities according to

$$\alpha_{m,j}^{t+1} = \frac{1}{N_m} \sum_{i=1}^{N_m} w_{m,i,j}^{t+1}, \tag{5}$$

and computes $w_{m,j}^{t+1} = \sum_{i=1}^{N_m} w_{m,i,j}^{t+1}$. Finally, the summary quantities are updated according to

$$\begin{aligned} w_j^{t+1} &= w_j^t + w_{m,j}^{t+1} - w_{m,j}^t, \\ a_j^{t+1} &= a_j^t + a_{m,j}^{t+1} - a_{m,j}^t, \\ b_j^{t+1} &= b_j^t + b_{m,j}^{t+1} - b_{m,j}^t. \end{aligned} \tag{6}$$

Here, all that is done is that the old values of the local summary statistics are being replaced by updated values. Note that no processing is performed at any node other than m on this iteration. In particular, for $k \neq m$ set $w_{k,j}^{t+1} = w_{k,j}^t$, $a_{k,j}^{t+1} = a_{k,j}^t$, and $b_{k,j}^{t+1} = b_{k,j}^t$. The updated values $\{w_j^{t+1}, a_j^{t+1}, b_j^{t+1}\}$ are then transmitted to the next node and the above process is repeated there. Figure 2 depicts the communication process in the sensor network. Recall that $s^t = \{w_j^t, a_j^t, b_j^t\}$.

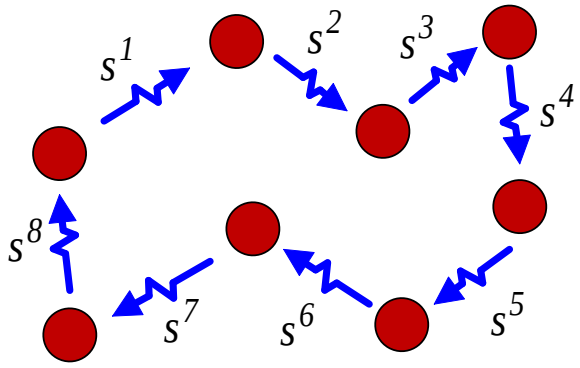

Fig. 2. Communication/iteration cycle in a sensor network.

The DEM algorithm monotonically converges to a local maximum (or saddle point) as described in the next section. Before moving on, a variant of DEM is discussed. In wireless sensor network applications, it is likely that communications are the major source of power consumption, rather than computation. Therefore, it may be desirable to employ more effective (and intensive) computations at each node in order to reduce the number of communications (cycles through the nodes). In the DEM algorithm above, in effect each node computes a single, local E-Step and M-Step. It is possible, however, that additional local E-Steps and M-Steps (more computation at each node) may lead to faster overall convergence (in terms of the number of required communications). Specifically, the computations in (3)-(6) can be repeated several times in succession until updated means and covariances $\{\mu_j^{t+1}, \Sigma_j^{t+1}\}$ reach a fixed point (or until the incremental change from one set of parameters to the next falls below a preset tolerance). This process is seeking to maximize, as opposed to simply increasing, the local log-likelihood at each node before moving on to the next. This algorithm is referred to as DEMM (DEM with Multiple steps at each node). The simulation experiments in the following sections demonstrate that this procedure can lead to significant speed-ups in the rate of convergence per communication. This variant is also guaranteed to converge to a local maximum (or saddle point). However, it should be mentioned that both DEM and DEMM operate much like a coordinate-ascent type algorithm,

and thus these algorithms may be more prone to convergence to local (suboptimal) maxima than standard EM.

V. Convergence Rate of Distributed EM

The results for the incremental EM algorithm in [6,8] can be adapted to prove that DEM converges to a fixed point. However, the rate at which DEM (or incremental EM) converges has not been investigated theoretically in previous work. Empirically, it has been observed that incremental EM often converges more rapidly than standard EM [6, 14], which has been one of the primary motivations for it. The convergence behavior of standard EM in the Gaussian mixture case is examined thoroughly in [4, 5]. Usually, the EM fixed points are points of local maxima of the log likelihood (although saddle points are also possible). The standard EM algorithm converges linearly in general, and can display superlinear convergence for well separated Gaussian mixtures. Here, it is shown that in certain cases DEM is (at least) linearly convergent to a local maximum of the log likelihood $l_y(\theta)$.

Assume that the sequence $\{\theta^t\}$ converges to a point θ^* where the log likelihood l_y assumes a local maximum. In this section it is shown that for sufficiently large t there exists a constant $0 \leq \beta < 1$ such that

$$\|\theta^t - \theta^*\| \quad \leq \quad \beta \|\bar{\theta}^{t-1} - \theta^*\|, \tag{7}$$

where $\|\cdot\|$ denotes the ℓ_2 norm and $\bar{\theta}^{t-1}$ is a certain average (defined below) of the past $\{\theta^{t-m}\}_{m=1}^{M}$. This result is crucial in applications since it can ensure that DEM converges reasonably quickly to θ^*. In the sensor network context, the convergence rate guarantees that the parameter estimates converge to θ^* (within some prespecified tolerance) in a finite number of iterations/communications.

Before analyzing the convergence of DEM, consider the convergence analysis of the standard EM algorithm [1, 7]. The EM algorithm updates can be written compactly as

$$\theta^{t+1} = \arg\max_{\theta} E[\log p(x|\theta) \,|y, \theta^t] = \arg\max_{\theta} \int \log p(x|\theta) p(x|y, \theta^t) dx, \tag{8}$$

where $p(x|y, \theta^t) = p(x|\theta^t)/p(y|\theta^t)$. Recall that $x = (y, z)$. Therefore $p(x|\theta) = p(x|y, \theta)p(y|\theta)$, and thus

$$E[\log p(x|\theta)|y, \theta^t] \quad = \quad \log p(y|\theta) + E\left[\log p(x|y, \theta)|y, \theta^t\right].$$

Define $l_y(\theta) \equiv \log p(y|\theta)$, $H_y(\theta, \theta^t) \equiv E[\log p(x|y, \theta)|y, \theta^t]$, and $Q(\theta, \theta^t) = l_y(\theta) + H_y(\theta, \theta^t)$. With this notation, the EM algorithm is equivalent to the recursion

$$\theta^{t+1} = \arg\max_{\theta} Q(\theta, \theta^t)$$

The convergence analysis of standard EM [1] is summarized as follows. It is easy to verify that for all θ, $\nabla^{11} H_y(\theta, \theta) = -\nabla^{20} H_y(\theta, \theta) > 0$ (positive definite), where ∇^{ij} denotes the i-th order partial derivatives with respect to the first argument and the j-th order partial derivatives with respect to the second argument. Let

θ^* denote a local maximum of the likelihood function. The gradient $\nabla^{10}Q(\theta_2, \theta_1)$ can be expanded in a Taylor series about the point (θ^*, θ^*) to obtain

$$\nabla^{10}Q(\theta_2, \theta_1) \;=\; \nabla^{10}Q(\theta^*, \theta^*) + \nabla^{20}Q(\theta^*, \theta^*)(\theta_2 - \theta^*) + \nabla^{11}Q(\theta^*, \theta^*)(\theta_1 - \theta^*) + R,$$

where R denotes the remainder of higher order terms. Note that $\nabla^{10}Q(\theta^*, \theta^*) = 0$ since θ^* is an EM fixed point. Evaluating this series at the point (θ^{t+1}, θ^t) produces

$$\nabla^{20}Q(\theta^*, \theta^*)(\theta^{t+1} - \theta^*) + \nabla^{11}Q(\theta^*, \theta^*)(\theta^t - \theta^*) \;=\; 0,$$

since $\nabla^{10}Q(\theta^{t+1}, \theta^t) = 0$ as a result of the M-step. Note that in the expression above the remainder R is dropped (for t sufficiently large the remainder is negligible). Re-arranging the above expression, one obtains

$$\theta^{t+1} - \theta^* = \left[-\nabla^{20}Q(\theta^*, \theta^*)\right]^{-1} \nabla^{11}Q(\theta^*, \theta^*)(\theta^t - \theta^*).$$

Taking the 2-norm on each side produces

$$\begin{aligned}
\|\theta^{t+1} - \theta^*\| &= \| \left[-\nabla^{20}Q(\theta^*, \theta^*)\right]^{-1} \nabla^{11}Q(\theta^*, \theta^*)(\theta^t - \theta^*)\|, \\
&\leq \| \left[-\nabla^{20}Q(\theta^*, \theta^*)\right]^{-1} \nabla^{11}Q(\theta^*, \theta^*)\| \, \|\theta^t - \theta^*\|, \\
&= \| \left[\nabla^2 l_y(\theta^*) + \nabla^{20}H_y(\theta^*, \theta^*)\right]^{-1} \nabla^{20}H_y(\theta^*, \theta^*)\| \, \|\theta^t - \theta^*\|,
\end{aligned}$$

where the fact $\nabla^{11}Q(\theta, \theta) = -\nabla^{20}H_y(\theta, \theta)$ is used in the last step. The (asymptotic) convergence rate is identified as the matrix 2-norm (largest eigenvalue)

$$\beta \;=\; \| \left[\nabla^2 l_y(\theta^*) + \nabla^{20}H_y(\theta^*, \theta^*)\right]^{-1} \nabla^{20}H_y(\theta^*, \theta^*)\|.$$

Under mild assumptions [1], both $-\nabla^2 l_y(\theta^*)$ and $-\nabla^{20}H_y(\theta^*, \theta^*)$ are positive definite and it follows that $\beta < 1$.

Now consider the sensor network situation. Since the data collected at each sensor is independent of the data at other sensors, the criterion above can be written as a sum. For each node, $m = 1, \ldots, M$, let $y_m = \{y_{m,i}\}_{i=1}^{N_m}$ be the set of measurements, $z_m = \{z_{m,i}\}_{i=1}^{N_m}$, the set of missing data, and $x_m = (y_m, z_m)$, the complete data, at node m. Then the EM update equation can be expressed as

$$\theta^{t+1} \;=\; \arg\max_{\theta} \left\{ \sum_{m=1}^{M} \left(l_{y_m}(\theta) + H_{y_m}(\theta, \theta^t) \right) \right\} \;,$$

where $l_{y_m}(\theta)$ is the "local" log likelihood function at node m and $H_{y_m}(\theta, \theta^t)$ is the corresponding local version of $H_y(\theta, \theta^t)$.

DEM and other incremental versions of the EM algorithm are based on an alternating-maximization prodedure applied to the following functional:

$$Q(\theta, \theta_1, \ldots, \theta_M) \;\equiv\; l_y(\theta) + \sum_{m=1}^{M} H_{y_m}(\theta, \theta_m),$$

Note that setting $\theta_m = \theta^t$, $m = 1, \ldots, M$, and maximizing with respect to θ produces the usual EM iteration. DEM is based on the following recursion. Let $n = (t)_M$, the node at step t of DEM. Set

$$\theta_n^t \;=\; \theta^{t-1}, \quad \theta_m^t \;=\; \theta_m^{t-1}, \; m \neq n, \tag{9}$$

and update according to

$$\theta^t \;=\; \arg\max_{\theta} Q(\theta, \theta_1^t, \ldots, \theta_M^t).$$

The maximization is solved by the DEM update equations (6) given in the preceding section. A slight wrinkle is that DEM does not satisfy the likelihood monotoniticy property of standard EM. Each iterate of standard EM satisfies the property that $l_y(\theta^{t+1}) \geq l_y(\theta^t)$. DEM iterates are not guaranteed to satisfy this monotonicity property. Note, however, that each step of DEM does satisfy the monotonicity condition

$$Q(\theta^{t+1}, \theta_1^t, \ldots, \theta_M^t) \geq Q(\theta^t, \theta_1^t, \ldots, \theta_M^t),$$

which shows that the objective function Q is improved at each step.

Now consider the convergence behavior of DEM. Just as in the standard EM case, for all θ and each m, $\nabla^{11} H_{y_m}(\theta, \theta) = -\nabla^{20} H_{y_m}(\theta, \theta) > 0$. Let $\nabla^1 Q(\theta, \theta_1, \ldots, \theta_M)$ denote the gradient with respect to the first variable θ, $\nabla^{1m} Q(\theta, \theta_1, \ldots, \theta_M)$ denote the first partial derivatives with respect to θ and θ_m, and $\nabla^2 Q(\theta, \theta_1, \ldots, \theta_M)$ denote the second partial derivatives with respect to θ. Expand $\nabla^1 Q(\theta, \theta_1, \ldots, \theta_M)$ in a Taylor series about the point $(\theta^*, \theta^*, \ldots, \theta^*)$:

$$\begin{aligned} \nabla^1 Q(\theta, \theta_1, \ldots, \theta_M) \;&=\; \nabla^1 Q(\theta^*, \theta^*, \ldots, \theta^*) + \nabla^2 Q(\theta^*, \theta^*, \ldots, \theta^*)(\theta - \theta^*) \\ &\quad + \sum_{m=1}^{M} \nabla^{1m} Q(\theta^*, \theta^*, \ldots, \theta^*)(\theta_m - \theta^*) \;+\; R, \end{aligned}$$

where R denotes the remainder of higher order terms. Now considering this expansion at $\nabla^1 Q(\theta^{t+1}, \theta_1^t, \ldots, \theta_M^t) = 0$, and making substitutions similar to that in the standard EM analysis above, produces

$$\left[\nabla^2 l_y(\theta^*) + \sum_{m=1}^{M} \nabla^{20} H_{y_m}(\theta^*, \theta^*)\right] (\theta^{t+1} - \theta^*) \;=\; \sum_{m=1}^{M} \nabla^{20} H_{y_m}(\theta^*, \theta^*)\, (\theta_m^t - \theta^*), \tag{10}$$

where, as in the standard EM analysis above, the remainder R is dropped. Re-arranging this expression gives

$$\theta^{t+1} - \theta^* \;=\; \left[\nabla^2 l_y(\theta^*) + \sum_{m=1}^{M} \nabla^{20} H_{y_m}(\theta^*, \theta^*)\right]^{-1} \sum_{m=1}^{M} \nabla^{20} H_{y_m}(\theta^*, \theta^*)\, (\theta_m^t - \theta^*). \tag{11}$$

Now define the matrices

$$C_m \;=\; \left[\nabla^2 l_y(\theta^*) + \sum_{m=1}^{M} \nabla^{20} H_{y_m}(\theta^*, \theta^*)\right]^{-1} \nabla^{20} H_{y_m}(\theta^*, \theta^*)$$

and

$$C \;=\; \sum_{m=1}^{M} C_m \;=\; \left[\nabla^2 l_y(\theta^*) + \sum_{m=1}^{M} \nabla^{20} H_{y_m}(\theta^*,\theta^*)\right]^{-1} \sum_{m=1}^{M} \nabla^{20} H_{y_m}(\theta^*,\theta^*).$$

C_m is positive definite for each m, and thus so is C. Let $\bar{\theta}^t$ denote the weighted average

$$\bar{\theta}^t \;=\; \sum_{m=1}^{M} C^{-1} C_m\, \theta_m^t.$$

Note that by this definition, $\sum_{m=1}^{M} C_m \bar{\theta}^t = \sum C_m \theta_m^t$. With these properties and notation, equation (11) can be written as

$$\begin{aligned} \theta^{t+1} - \theta^* &= \sum_{m=1}^{M} C_m(\theta_m^t - \theta^*), \\ &= \sum_{m=1}^{M} C_m(\bar{\theta}^t - \theta^*), \\ &= C\,(\bar{\theta}^t - \theta^*). \end{aligned} \tag{12}$$

Taking the norm of each side produces the bound $\|\theta^{t+1} - \theta^*\| \leq \|C\|\,\|\bar{\theta}^t - \theta^*\|$. Recalling the definition of C and letting $\nabla^{20} H_y(\theta^*,\theta^*) = \sum_{m=1}^{M} \nabla^{20} H_{y_m}(\theta^*,\theta^*)$ shows that

$$\|C\| \;=\; \left\| \left[\nabla^2 l_y(\theta^*) + \nabla^{20} H_y(\theta^*,\theta^*)\right]^{-1} \nabla^{20} H_y(\theta^*,\theta^*) \right\| \;<\; 1,$$

which is less than unity under the same conditions as standard EM, from which the DEM convergence bound of (7) follows.

The bound in (7) does not necessarily guarantee linear convergence of DEM, due to the fact that $\bar{\theta}^t$ is generally not a simple "straight" average. Therefore, before closing this section, a further characterization of $\bar{\theta}^t$ is given.

$$\begin{aligned} \bar{\theta}^t &= \sum_{m=1}^{M} C^{-1} C_m\, \theta_m^t, \\ &= \sum_{m=1}^{M} \left[\sum_{m=1}^{M} \nabla^{20} H_{y_m}(\theta^*,\theta^*)\right]^{-1} \nabla^{20} H_{y_m}(\theta^*,\theta^*)\, \theta_m^t. \end{aligned}$$

Furthermore, it is easily verified that $\nabla^{20} H_{y_m}(\theta^*,\theta^*) = -\nabla^{02} D_{y_m}(\theta^*,\theta^*)$, where $D_{y_m}(\theta_2,\theta_1)$ is the Kullback-Leibler divergence

$$D_{y_m}(\theta_2,\theta_1) \equiv \int \log \frac{p(x_m|y_m,\theta_2)}{p(x_m|y_m,\theta_1)} p(x_m|y_m,\theta_2) dx_m.$$

Therefore,

$$\bar{\theta}^t \;=\; \sum_{m=1}^{M} \left[\nabla^{02} D_y(\theta^*,\theta^*)\right]^{-1} \nabla^{02} D_{y_m}(\theta^*,\theta^*)\, \theta_m^t,$$

where $D_y = \sum_{m=1}^{M} D_{y_m}$. Note that if $\nabla^{02} D_{y_m}(\theta^*, \theta^*)$ is the same for all m, then

$$\bar{\theta}^t \;=\; \frac{1}{M}\sum_{m=1}^{M} \theta_m^t \;=\; \frac{1}{M}\sum_{m=1}^{M} \theta^{t-m+1},$$

a simple average of the past M iterates. This situation holds approximately if all sensors make i.i.d. observations in equal and sufficient numbers. In this case, it follows that

$$\begin{aligned} \|\theta^t - \theta^*\| &\leq \beta \,\|\frac{1}{M}\sum_{m=1}^{M} \theta^{t-m} - \theta^*\|, \\ &\leq \beta \max_{1\leq m\leq M} \|\theta^{t-m} - \theta^*\|, \end{aligned}$$

from which it follows that the maximum normed error decays at least linearly with each full cycle of DEM. A similar convergence behavior can be established if the matrices $\{C_m\}$ are diagonal (which is the case when $\nabla^{02} D_{y_m}(\theta^*, \theta^*)$, $m = 1, \ldots, M$, share a set of common eigenvectors). In that case, the errors can be examined coordinate-wise, and it can be shown that the maximum absolute error on each coordinate decays at least linearly with each full cycle of DEM.

VI. Communication Requirements and Scalability of Distributed EM

The total amount of communication in a (wireless) sensor network consisting of a large number of devices in a confined area must be limited. Specifically, under reasonable assumptions it has been demonstrated that the *transport capacity* of a wireless network consisting of M nodes positioned in a region of unit area is $O(\sqrt{M})$ bit-meters/sec [9]. A bit-meter is defined as the transport of 1 bit over a distance of 1 meter. This implies that throughput per-node is $O(1/\sqrt{M})$ bit-meters/sec. Note that the per node throughput vanishes as the density of nodes increases. This key result places a fundamental bound on the amount of communication that can be carried out in a very dense (wireless) sensor network. Naturally, this fundamental bound on the capacity raises the question as to whether or not the DEM algorithm is feasible in very dense sensor networks.

DEM, as implemented above, is quite feasible in dense networks. At any given time (iteration step of DEM) one and only one node is transmitting an updated sufficient statistic to another node (one or more hops away). In this case, because no simultaneous transmissions occur, the communication requirements are easily met (at least within the physical power/bandwidth constraints of the network). Interference is not an issue. In fact, DEM can be carried out while other communications are taking place in the network (the impact of DEM is minimal to the overall capacity). However, more aggressive variations on the DEM procedure are possible, which may be limited by the $O(1/\sqrt{M})$ capacity bound. Faster (in real-time) convergence may be accomplished by performing updates at all nodes in parallel followed by parallel (simultaneous) transmissions of updates to neighbors. For example, M cyclic iteration processes could be run in parallel, with each node updating the statistics sent from one of its neighbors and then transmitting them to the next neighbor. This "parallel-update" DEM is feasible in dense networks under the following assumptions.

A.1 The nodes are distributed over a region of total area 1 square meter and average distance between neighboring nodes is of $O(\sqrt{1/M})$ meters.

A.2 DEM cycles through the nodes via near-neighbor hops; i.e., the average distance between successive nodes in DEM is $O(\sqrt{1/M})$ meters.

Under *A.1* and *A.2* each iteration of the distributed EM algorithm requires the transmissions of the updated sufficient statistics from each node to its neighbor (M transmissions simultaneously) over an average distance of $O(1/\sqrt{M})$. There are a fixed number of statistics. Therefore, carrying out DEM with 64 bit floating point precision results in $O(1)$ bits per transmission, and the throughput per transmission is $O(1/\sqrt{M})$ bit-meters/sec. The overall transport requirement is $O(\sqrt{M})$, in agreement with the capacity bound of [9].

More sophisticated network coding might also be employed. In a dense sensor network, the data recorded at neighboring nodes becomes more and more similar in distribution and may even be correlated. Correlations between neighboring nodes have been exploited in sensor networks to devise more efficient codes [12] and, in particular, codes that meet the capacity bound in certain sensor network estimation problems have been constructed [11]. In the distributed EM context, one can exploit the fact that the distributions at nearby nodes are very similar, tending to identical as the node density increases. Thus, in the dense setting, a collaborative communication scheme, similar to that proposed in [10], may be applicable, which could reduce the overall transport requirement, thereby conserving power. More specifically, the sensor network could be organized into $O(\sqrt{M})$ local subnetworks, each consisting of $O(\sqrt{M})$ neighboring nodes. If the nodes in each subnetwork are very close to one another, then subnetworks can act as multiple transmit-receive antennas and a transport capacity of $O(M)$ bit-meters/sec is possible [10]. This suggests a hierarchical updating scheme for distributed EM wherein each subnetwork updates its parameter estimates internally (perhaps in a cyclic fashion) and then communicates the updated estimate to the next subnetwork. Such a scheme could reduce the power consumption of DEM.

There are several simple approaches that could reduce the constant number of bits transmitted per iteration, which may reduce the power requirements of DEM and the sensor network. For example, the sparse update strategy proposed in [6], based on updating only the most significant component at each iteration, could be employed. Another approach is to optimize the coding of the sufficient statistics in each transmission. The incremental nature of the EM algorithm can be exploited in this regard. Each node knows what it sent to its neighbor at the previous iteration, and this side information can be used to more efficiently encode the updated statistics at the current iteration. Specifically, at the t-th iteration a node must transmit the updated sufficient statistic $s^t = \{w_j^t, a_j^t, b_j^t\}_{j=1}^J$ to one of its neighbors. The statistic s^t can be encoded using the probability law known to the destination, namely $p(\cdot|\theta^{t-M})$ based on the statistic s^{t-M} transmitted to that node at the previous iteration $t-M$. The Shannon code length $\log p(s^t|\theta^{t-M})$ can be nearly achieved using Huffman or arithmetic coding. The statistics $\{w_j^t\}$ can be transmitted first. Then, note that $a_j^t|w_j^t$ is

roughly the sum of w_j^t i.i.d. Gaussians with mean μ_j^t and covariance Σ_j^t. Similarly $b_j^t - a_j^t(a_j^t)'|w_j^t$ is roughly Wishart distributed. Hence, a_j^t and b_j^t can be coded using appropriate Gaussian and Wishart codebooks. One issue not addressed here is the stability of the distributed EM algorithm under quantization. This is an important issue to be addressed in future work.

VII. A Simulated Sensor Networking Application

A simulated sensor network application is presented here to illustrate the performance of the DEM algorithm. The following scenario is considered. Suppose that a network of M wireless nodes is each equipped with two sensors, a temperature sensor and a sensor for certain microorganisms. Understanding the relationship between microorganisms and their environmental conditions is viewed as one of the important potential application areas for sensor networks. An example is the study of the relationship between marine microorganisms and temperature [13]. In that setting, clusters in temperature/microorganism-density feature space can be expected due to the existence of thermoclines in the ocean. The simulation here is meant to mimic this situation. Each sensor records N measurements. Each measurement is a pair of numbers corresponding to a temperature reading and a microorganism density reading. The units of the measurements are assumed to be scaled so that the feature space is the unit square $[0,1]^2$.

Figure 3(a)-(d) depicts a simulated set of data. In this simulation $M = N = 100$. The data were generated according to a three-component Gaussian mixture model. The mixing probabilities at each node were selected randomly, but in each case roughly 90% of the total mass was placed on one of the components to simulate the effect of the thermoclines. To mimic the effect of sensor saturation, the Gaussian data was thresholded to force the data into the unit square (which is apparent especially in the upper right hand corner). Standard EM (distributed implementation) and DEM (single EM step at each node), and DEMM (multiple EM steps at each node) are were used to estimate the three components. The algorithms were randomly initialized with the Gaussian mixture components depicted by the dashed cirlces in Figure 3(e). All three algorithms converged to the same solution. The solid ellipsoids in Figure 3(e) indicate the estimated components, which agree very well with the data clusters. The estimated means and covariances are very close to the values used to generate the data. The normalized squared errors (squared errors divided by squared norms of the true parameters) were on the order of 10^{-4}. The estimated mixing parameters were also close to their true values. The average absolute error between the estimated and true probabilities was 0.0179.

The rate of convergence of the three algorithms, as a function of number of transmitted bits — which corresponds to numbers of messages passed between nodes — is compared in Figure 3(f). Clearly the DEM algorithm with multiple EM steps per node converges most rapidly in this case. Convegence is declared when the norm of the difference between successive parameter estimates is less than a prespecified tolerance (in this simulation the tolerance was 10^{-5}). According this tolerance, the standard EM, using its distributed

implementation, converged in 12 iterations (corresponding to roughly $12 \times M = 1200$ iterations of DEM in terms of communication and computation costs), DEM converged in 343 iterations, and DEMM converged in 185 iterations (the number of iterations was determined according to the same tolerance used to declare global convergence). Each iteration requires the communication of 15 floating point numbers (3 components, 2 means and 3 covariance values per component). Assuming 64-bit precision, this amounts to roughly 1kbits (over a short near neighbor hop) per iteration. The total communication cost for DEM with multiple EM steps is under 200kbits (over a short distance).

Note that upon convergence, every node has (roughly) the same estimates of the global mean and covariance parameters. Therefore, any one of the nodes may be called upon to transmit the result to a remote site. Any node may also be queried for its local mixing probability estimates. Thus, global and local information can be retrieved from the sensor network with low bandwidth/power communications (relative to the communication cost of transmitting all the data to a remote site).

The results of this experiment are representative of many other experiments (not shown here) that were conducted varying the amounts and distributions of data, the feature dimensions, the numbers of components, and the random initializations. DEM with multiple EM steps at each node converged most rapidly in most cases, usually in about $2M$ iterations of the algorithm (DEM with a single EM step per node typically took more than twice as many iterations to converge).

VIII. Conclusions

This paper presented a distributed EM algorithm suitable for clustering and density estimation in sensor networks. DEM 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. Under mild conditions, DEM converges to a stationary point of the log likelihood function, usually a local maximum. In certain cases, DEM converges at a linear rate (in a certain sense), potentially converging more rapidly than standard EM. This makes DEM attractive for sensor network applications. With reasonable assumptions, it was shown that DEM's communication requirements are quite modest. Even with simultaneous/parallel processing implementations, the communication requirements in a region of fixed area scale like $O(\sqrt{M})$, where M is the number of nodes. This requirement satisfies known transport capacity bounds [9] for wireless sensor networks. Thus, DEM is feasible for very dense wireless sensor networks. A simulation study demonstrated the potential of DEM for sensor network data analysis.

DEM may also be useful in other distributed or networked data analysis tasks. For example, proprietary issues may prohibit the sharing of raw data between Internet service providers, but the sharing of cumulative sufficient statistics may be acceptable. Thus, one could envision using DEM for Internet performance data analysis. Incremental EM algorithms have been used previously for data mining in large databases [14]. In

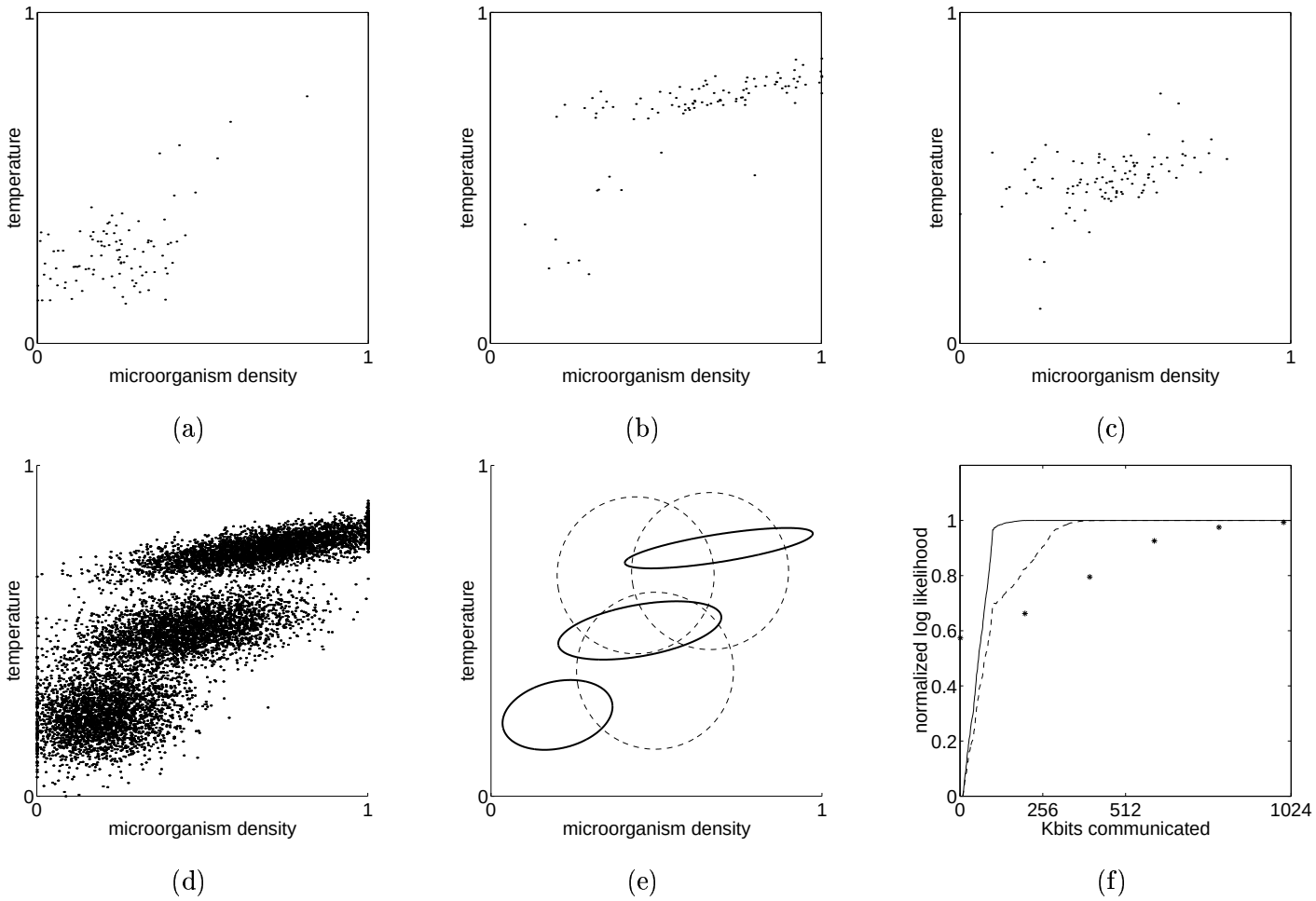

Fig. 3. Sensor network simulation. (a)-(c) Representative cases of data collected at three different nodes. (d) Scatter plot of data collected at all sensors. (e) Estimated Gaussian components (solid), initial Gaussian components (dashed). (f) Log likelihood as a function of communicated bits (assuming 64-bit floating point precision and no transmission errors) for DEM with multiple EM steps at each node (solid), DEM with single EM step at each node (dashed), distributed implementation of standard EM (*).

that work, it is shown that the number of data involved at each incremental step can dramatically effect the efficiency of the algorithm, and a method for selecting a near optimal data block size is given. A variant of incremental EM, called Lazy EM, is also developed in that paper which focuses incremental updates on more informative blocks of data. Both methods can reduce the computational costs, and may also provide reductions in communication overhead of DEM. The convergence rate analysis of DEM in this paper provides further motivation for using incremental EM for very large scale data mining problems.

The DEM algorithm can be extended and enhanced in several ways. Model order selection criteria could be incorporated into DEM to automatically identify an appropriate number of components. The approach proposed in [17] is an especially attractive candidate for automatic model order selection because it is easily and naturally integrated into the EM iteration process. It may also be possible to develop on-line dynamic versions of DEM based on the on-line EM algorithm [15]. Another key issue is that, in this paper, the

data at each node are assumed to be statistically independent, but as pointed out earlier this assumption can be relaxed. If the data are possibly correlated, then the DEM algorithm can still be applied with the independent likelihood structure employed here. In that case, the independent likelihood can be interpreted as a *pseudolikelihood* [16], and under mild conditions the maximum pseudolikelihood estimates tend to the true maximum likelihood estimates as the number of data tends to infinity. It may also be possible to employ non-cyclic, non-sequential updating strategies (potentially parallel and/or asynchronous schemes) in DEM that could further increase the convergence rate (in terms of real time). Finally, most of the results discussed in this paper can be easily extended to other mixture models consisting of component distributions from the exponential family.

Acknowledgments

The author would like to thank Professor Urbashi Mitra and Rui Castro for helpful feedback and discussions on topics related to this paper.

References

[1] A. Dempster, N. Laird, and D. Rubin. "Maximum likelihood estimation from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, pp. 1–38, 1977.

[2] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* John Wiley & Sons, New York, 1997.

[3] C. Wu. "On the convergence properties of the EM algorithm," *Annals of Statistics*, vol. 11, pp. 95-103, 1983.

[4] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithm for Gaussian Mixtures," Neural Computation, v.8, pp. 129-151, 1996.

[5] J. Ma, L. Xu, and M. I. Jordan, "Asymptotic convergence rate of the EM algorightm for Gaussian Mixtures," Neural Computation, v.12, pp. 2881-2907, 2000.

[6] R. M. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants", *Learning in Graphical Models*, M. I. Jordan (editor) pp. 355-368, Dordrecht: Kluwer Academic Publishers, 1998.

[7] A.O. Hero and J.A. Fessler, "Convergence in norm for EM-type algorithms," , *Statistica Sinica*, v. 5, n. 1, pp. 41-54, Jan. 1995.

[8] A. Gunawardana, "The Information Geometry of EM Variants for Speech and Image Processing," Ph.D. Dissertation, The Johns Hopkins University, Baltimore, MD, 2001.

[9] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," IEEE Transactions on Information Theory, vol. IT-46, no. 2, pp. 388-404, March 2000.

[10] P. Gupta and P. R. Kumar, "Towards an information theory of large networks: an achievable rate region," Proceedings of 2001 IEEE International Symposium, 2001, p. 159.

[11] S. D. Servetto, "On the Feasibility of Large-Scale Wireless Sensor Networks," Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing, Urbana, IL, October 2002.

[12] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," IEEE Signal Processing Magazine , Volume: 19 Issue: 2 , March 2002 pp. 51 -60, March 2002.

[13] CENS - Center for Embedded Networked Sensing, a NSF Science & Technology Center, http://cens.ucla.edu/.

[14] B. Thiesson, C. Meek, and D. Heckerman. "Accelerating EM for Large Databases," Machine Learning, v. 45, pp. 279-299, 2001.

[15] M. Sato and S. Ishii, "On-line EM Algorithm for the Normalized Gaussian Network", Neural Computation, v. 12, no. 2, pp. 407-432, 2000.

[16] J. Besag, "On the Statistical Analysis of Dirty Pictures," Journal of the Royal Statistical Society. Series B (Methodological), Vol. 48, No. 3., pp. 259-302, 1986.

[17] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," IEEE Trans. Pattern Analysis and Machine Intelligence, v. 24 no. 3, pp. 381-396, March 2002.