A CONSENSUS-BASED DECENTRALIZED EM FOR A MIXTURE OF FACTOR ANALYZERS

Gene T. Whipps    Emre Ertin    Randolph L. Moses

The Ohio State University, ECE Department, Columbus, OH 43210

ABSTRACT
We consider the problem of decentralized learning of a target appearance manifold using a network of sensors. Sensor nodes observe an object from different aspects and then, in an unsupervised and distributed manner, learn a joint statistical model for the data manifold. We employ a mixture of factor analyzers (MFA) model, approximating a potentially nonlinear manifold. We derive a consensus-based decentralized expectation maximization (EM) algorithm for learning the parameters of the mixture densities and mixing probabilities. A simulation example demonstrates the efficacy of the algorithm.

Index Terms— decentralized learning, Gaussian mixture, mixture of factor analyzers, consensus

1. INTRODUCTION
A spatially distributed sensor network can be used to construct a rich appearance model for targets in their common field-of-view. These models can then be used to identify previously seen objects if they reappear in the network at a later time. As an example, consider a network of cameras capturing images of an object from different but possibly overlapping aspects as the object traverses through the network’s field of view. The ensemble of images captured by the network forms a low-dimensional nonlinear manifold in the high-dimensional ambient space of images. One approach to appearance modeling would be to construct independent models of a local data manifold at each sensor and share it across the network. However, such an ensemble of models suffers from discretization of the aspect space and poor parameter estimates as the number of unknown parameters necessarily scale linearly with the number of sensor nodes. Alternatively, the sensor nodes can collaborate to construct a joint model for the image ensemble. The parameter estimates of the joint model will improve with the number of sensor nodes, since the number of unknown parameters in the model is intrinsic to the object and fixed, whereas the measurements scale linearly with the number of sensor nodes. The straightforward method of pooling images to a central location for joint model construction will require large and likely impractical network bandwidth. In this paper, we develop a decentralized learning method with a potentially reduced data bandwidth need, and which results in a global appearance manifold model shared by all sensor nodes. Previously [1] developed methods for in-network learning of aspect-independent target signatures. In this work we focus on learning appearance models with aspect dependence.

We model the overall statistics as a mixture of factor analyzers (MFA) and derive a decentralized EM algorithm for learning model parameters. The MFA model is a probabilistic and generative one, and can be used for dimensionality reduction, manifold learning, and signal recovery from compressed sensing [2]. In the case of learning a data manifold, the MFA model is a linearization of a (potentially) nonlinear structure. The EM algorithm for a mixture of factor analyzers was derived in [3] in a centralized setting. There, the goal is to provide an algorithm for clustering and dimensionality reduction of high-dimensional data with a low-rank covariance model, also consistent with the structure of low-dimensional data manifolds. In this paper we extend the EM algorithm for the MFA model to the case of a spatially-distributed sensor network with goals of distributing computations across the network and being robust to individual node failures (e.g., losing connectivity to a central node in centralized or distributed systems).

Previous work [4, 5, 6, 7] employed MFA models for high-dimensional data with low-dimensional structure. The authors in [4] modify the standard EM for general mixture models by splitting and merging mixture components to avoid local maxima in mixture models when the data are dominated by some components in parts of the data space. The split and merge technique is essentially applied as post-processing after standard EM converges to a stationary point. In [5], the authors propose an algorithm that jointly learns model parameters and model order (i.e., the number of mixtures) for more general mixture models. The EM-like algorithms proposed in [6] and [7] for the MFA model address the issue of global alignment of the linear mappings to the underlying manifold using a modified EM cost function. The algorithm in [6] requires a fixed-point algorithm in the M-step, addressed in [7] in closed-form. It was shown in [8], that maximizing the cost function in [6] and [7] is equivalent to that in EM.

This paper differs from the results in [4, 5, 6, 7] in two important aspects. First, we provide a decentralized algo-
algorithm for fitting MFA models to manifold-structured high-dimensional data; the algorithm is ideally suited for distributed sensing where sensor data ensemble is not present at any central node. Second, we consider a more general MFA model suitable for modeling data observed by heterogeneous sensor nodes differing in their aspect angle with respect to the object. Specifically, we assume observations are drawn from the mixture density with mixture probabilities which can vary across the different sensor nodes. In other words, one sensor node may observe a mixture component with more (or less) probability than other nodes. In this context, the work in [4, 5, 6, 7] considers the homogenous sensing case, where all the sensor nodes observe different realizations of a single mixture model with identical mixture probabilities.

Our decentralized algorithm for MFA modeling is a consensus-based decentralized implementation of EM formulated for the general MFA model we consider. One of the earliest decentralized EM algorithms for a high-dimensional Gaussian mixture model was developed in [9] assuming the existence of a reliable network with a ring topology. The authors in [10, 11] later developed gossip-based decentralized EM algorithms for a Gaussian mixture model suitable for mesh networks. The algorithm proposed in [11] applies consensus to average local sufficient statistics across the network. The algorithm in [12] applied alternating direction method of multipliers (ADMM) with a consensus constraint to the maximization step of EM for more general mixture densities with log-concave conditionals. The goal in each of these references is to learn, in a distributed way, high-dimensional parametric models for data that are typically well-separated clusters; in contrast, our work incorporates a low-dimensional structure which is key to accurately modeling high-dimensional data observed by a network of sensors and whose relevant characteristics lie on a common low-dimensional manifold structure.

The remainder of this paper is outlined as follows. In Section 2, we outline the (approximated) observation model (i.e., a mixture of factor analyzers). In Section 3, we derive the update equations of the consensus-based decentralized EM algorithm for a mixture of factor analyzers. In Section 5, a numerical example demonstrating the efficacy of the algorithms is provided. Finally, conclusions are given in Section 6.

2. MIXTURE OF FACTOR ANALYZERS

We consider a sensor network of M sensors. Sensor m ∈ {1, 2, . . . , M} collects Nm observations. The ith measurement at the mth node is denoted by xmi ∈ Rp. The measurements are assumed to be i.i.d. observations from a nonlinear r-dimensional manifold, with r ≤ p. We approximate the system by linear mappings embedded in the higher dimensional space according to

\[ x_j = \Lambda_j y_j + \mu_j + w_j, \]

where \( y_j \) ∈ \( \mathbb{R}^r \) and \( w_j \) ∈ \( \mathbb{R}^p \) for \( j \in \{1, 2, \ldots , J\} \), and

\[ y_j \sim \mathcal{N}(0, I_r) \quad \text{and} \quad w_j \sim \mathcal{N}(0, \Sigma_j), \]

where \( I_r \) is the \( (r \times r) \) identity matrix. The number of mixture components \( J \) is assumed known. The vector \( x_j \) is a random vector conditioned on knowing \( j \), while \( x_{m,i} \) is the \( i \)th observation at the \( m \)th sensor node. Given the mixture component to which the sensor observation belongs, then \( x_{m,i} \) is a realization of the random vector \( x_j \). It is straightforward to show \( x_j \sim \mathcal{N}(\mu_j, \Sigma_j) \) with \( \Sigma_j = \Psi_j + \Lambda_j \Lambda_j^T \).

The covariance \( \Psi_j = \text{diag}(\psi_{1,j}, \psi_{2,j}, \ldots , \psi_{p,j}) \) is diagonal and positive definite. The local coordinate \( y_j \) is not directly observed and can be seen as spherical perturbations along piecewise linear segments of the manifold that are mapped to the observation space. The term \( w_j \) accounts for observation noise. The columns of \( \Lambda_j \) span the \( j \)th lower dimensional space.

Sensor nodes observe the environment composed of the same component Gaussian densities, but with potentially different proportions of each component. In the observation space, the measurements follow a mixture model according to

\[ x_{m,i} \sim \sum_{j=1}^{J} \alpha_{m,j} \mathcal{N}(\mu_j, \Sigma_j), \]

for \( m \in \{1, 2, \ldots , M\} \), \( i \in \{1, 2, \ldots , N_m\} \). The unknown parameters are \( \mu_j, \Psi_j, \Lambda_j \), and component probabilities \( \alpha_{m,j} \) for each mixture component \( j \) and sensor node \( m \). This model extends the mixture of factor analyzers model [3] to a network of sensors where sensor nodes observe the factors (mixture components) with potentially varying proportions across the network (i.e., \( \alpha_{m,j} \neq \alpha_{k,j} \) for \( m \neq k \)).

3. EXPECTATION-MAXIMIZATION ALGORITHM FOR A MIXTURE OF FACTOR ANALYZERS

The complete data is \( \{x_{m,i}, y_{m,i}, z_{m,i}\}_{m,i} \), where \( y_{m,i} \) and \( z_{m,i} \) are missing data that indicate the local space and the mixture component, respectively, from which the observation \( x_{m,i} \) was drawn. Define \( \psi_j = [\psi_{1,j}, \psi_{2,j}, \ldots , \psi_{p,j}]^T \), \( \Lambda_j = \text{vec}(\Lambda_j) \), and \( \alpha_m = [\alpha_{m,1}, \alpha_{m,2}, \ldots , \alpha_{m,J}]^T \). The notation \text{vec}(A) represents a column vector of stacked columns from matrix \( A \). Similar to the relationship between \( x_{m,i} \) and \( y_j \), the missing data from the local space \( y_{m,i} \) is a realization from the random vector \( y_j \) given that the data belongs to the \( j \)th mixture component.

We define the unknown parameter vector as

\[ \theta = [\mu_1^T, \ldots , \mu_J^T, \psi_1^T, \ldots , \psi_J^T, \Lambda_1^T, \ldots , \Lambda_J^T, \alpha_1^T, \ldots , \alpha_M^T]^T. \]

Let \( \theta^0 \) denote an iterate of the parameter vector \( \theta \). Let \( x = \{x_{m,i}\}_{m,i} \), \( y = \{y_{m,i}\}_{m,i} \), and \( z = \{z_{m,i}\}_{m,i} \) represent the collection of all samples from all nodes for the measured and missing data. The EM cost function is then given by

\[ Q(\theta, \theta^0) = E \left[ \log p(x, y, z | \theta) | \theta, \theta^0 \right] \]

\[ = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \sum_{j=1}^{J} \text{Pr}(z_{m,i} = j | x_{m,i}, \theta^0) \times \]

\[ (\log \alpha_{m,j} \mathcal{N}(x_{m,i} | \mu_j, \Sigma_j) + E \left[ \log p_j(y_{m,i} | x_{m,i}, \theta) | x_{m,i}, \theta^0 \right]) \].
In the E-step, updates of the posterior distributions are found to be

\[
Pr(z_{m,i} = j | x_{m,i}, \theta^t) = \frac{\alpha_{m,j}^t N(x_{m,i} | \mu_{j,t}^t, \Sigma_j^t)}{\sum_{k=1}^J \alpha_{m,k}^t N(x_{m,i} | \mu_k^t, \Sigma_k^t)} \equiv w_{m,j}^{t+1},
\]

and

\[
p_j(y_{m,i} | x_{m,i}, \theta^t) = p(y_{m,i} | z_{m,i} = j, x_{m,i}, \theta^t) = N(y_{m,i} | \mu_{j,t}^{t+1}, \Sigma_j^{t+1}).
\]

We define the matrix

\[
U_j = C_j \Lambda_j^t \Psi_j^{-1}.
\]  

(1)

The mean and covariance of the conditional posterior of \( y_{m,i} \)
for \( z_{m,i} = j \) are given by

\[
\kappa_{m,i,j} = U_j (x_{m,i} - \mu_j),
\]

(2)

\[
C_j = (I_p + \Lambda_j^t \Psi_j^{-1} \Lambda_j)^{-1}.
\]

(3)

The iterates \( \alpha_{m,j}^{t+1} \), \( U_j^{t+1} \), and \( \kappa_{m,i,j}^{t+1} \) are found according to equations (1)-(3) given \( \theta^t \).

The probabilities \( \alpha_{m,j}^{t+1} \) require the inverse and determinant of a \( (p \times p) \) covariance estimate \( \Sigma_j^t \). Instead of \( O(p^3) \) computations, the matrix inversion lemma permits the inverse with \( O(p^3) \) computations due to the structure imposed by the MFA model. As noted in [3], the inverse of \( \Sigma_j^{t} \) can be found by

\[
\Sigma_j^{-1} = \Psi_j^{-1} - \Psi_j^{-1} \Lambda_j C_j \Lambda_j^t \Psi_j^{-1}.
\]

A similar reduction in computations can be achieved in the computing of the determinant. The determinant of \( \Sigma_j^{t} \) reduces to [7]

\[
|\Sigma_j^{t}| = |C_j^{t}|^{-1} \prod_{k=1}^p \psi_{k,j}^{t}.
\]

Given the E-step updates of the posteriors, the unknown global parameters are then found in the M-step according to

\[
\theta^{t+1} = \arg \max_{\theta} Q(\phi, \theta^t).
\]

The maximizer of \( Q(\theta, \theta^t) \) for the MFA model has a closed-form solution. We denote the centered observations by \( \tilde{x}_{m,i,j}^t = x_{m,i} - \mu_j \). Similar to [9, 11], but for a MFA, we define the following local statistics for sensor node \( m \) for mixture component \( j \):

\[
a_{m,j}^{t+1} = \sum_{i=1}^N w_{m,i,j}^{t+1},
\]

(4)

\[
b_{m,j}^{t+1} = \sum_{i=1}^N w_{m,i,j}^{t+1} \tilde{x}_{m,i},
\]

(5)

\[
V_{m,j}^{t+1} = \sum_{i=1}^N w_{m,i,j}^{t+1} \tilde{x}_{m,i,j} (U_j^{t+1} \tilde{x}_{m,i,j})^T,
\]

(6)

and

\[
s_{m,j}^{t+1} = \text{diag} \left( \sum_{i=1}^N w_{m,i,j}^{t+1} \tilde{x}_{m,i,j} (\tilde{x}_{m,i,j})^T \right),
\]

(7)

where \( \text{diag}(A) \) is a column vector of the diagonal elements of square matrix \( A \). Note that in the standard EM implementation, the centered observations are relative to \( \mu_j^{t+1} \). Instead, we reference the observations to \( \mu_j^t \) to simplify the decentralization of EM. Otherwise, the global parameter \( \mu_j^{t+1} \) would need to be known at all nodes before computing local statistics. It was shown in [13] that the modified EM (i.e., referencing to \( \mu_j^t \) instead of \( \mu_j^{t+1} \)) is equivalent to standard EM in its convergence properties.

The mixture probabilities are updated locally according to

\[
a_{m,j}^{t} = \frac{1}{N_m} a_{m,j}^{t+1}.
\]

(8)

The global parameters are updated according to

\[
\mu_j^t = \frac{\sum_{m=1}^M b_{m,j}^t}{\sum_{m=1}^M a_{m,j}^t},
\]

(9)

\[
\Lambda_j^t = \frac{\sum_{m=1}^M V_{m,j}^t}{\sum_{m=1}^M a_{m,j}^t} \left( \Lambda_j^t + \sum_{m=1}^M V_{m,j}^t \right)^{-1},
\]

(10)

\[
\psi_j^t = \frac{\sum_{m=1}^M s_{m,j}^t}{\sum_{m=1}^M a_{m,j}^t} - \text{diag} \left( \Lambda_j^t \sum_{m=1}^M (V_{m,j}^t)^{-1} \right).
\]

(11)

The update equations (9)-(11) represent the centralized or standard M-step updates for the MFA model. We outline a decentralized approach of the global updates in the sequel.

4. CONSENSUS-BASED DECENTRALIZED EM FOR A MIXTURE OF FACTOR ANALYZERS

In a decentralized algorithm, each sensor node computes and communicates local statistics to nearby sensor nodes, instead of a central node. Then, each node estimates global parameters from updates of local statistics from its neighbors. The local statistics can be updated by various gossip-type strategies. See [14, 15, 16, 17], among many others, for works on general distributed averaging. Our focus is on a decentralized algorithm for the MFA model.

In the standard EM approach for a MFA model, the local statistics from all sensor nodes are summed and combined to estimate the global parameters. The summation across nodes \( (m = 1, 2, \ldots, M) \) in equations (9)-(11) can be replaced by sample averages since the constant \( 1/M \) is then a common factor to each numerator and denominator in the update equations (9)-(11). Thus, similar to [11], we employ consensus-based averaging [14] for the decentralized algorithm to estimate the sample average of local statistics across the sensor network.

In consensus-based averaging, nodes compute a weighted average of their information with information from neighboring nodes. Here, prior to the M-step, each sensor node first calculates its local statistics using equations (4)-(7), and then the sensor network establishes consensus on the average of the local statistics. The total number of real values shared under consensus with neighbor nodes for the MFA model is \( n_{\text{MFA}} = Jp(r+2) + J \), whereas the total for a standard GMM is \( n_{\text{GMM}} = Jp(p/2 + 3/2) + J \). Thus, a reduction in the com-
munications load (and dimensionality reduction) are achieved when $r < (p - 1)/2$. Finally, each node updates its (local) iterates of the global parameters using the consensus-based averages of the local statistics.

Each node computes their own local estimates of the global parameters. We relabel the global parameters to denote their relation to a given node $k$: $\mu_{k,j}^t$, $\Lambda_{k,j}^t$, and $\Psi_{k,j}^t$. In the E-step at each node, the parameters of the posterior distributions are computed from the locally-estimated global parameters of the previous M-step and are similarly relabeled: $w_{k,i,j}^t$, $C_{k,j}^t$, and $U_{k,j}^t$. The mixture probabilities $\alpha_{k,j}^t$ are still updated according to (8) since they are relevant to node $k$ only. Abstractly, sample averages of the form $\frac{1}{M} \sum_{m=1}^M y_m$ in the updates of the global parameters (9)-(11) are replaced by consensus averages denoted by $\bar{y}_k$ at node $k$. For example, the iterate $\mu_k^t$ at node $k$ becomes $\mu_{k,j}^t = \frac{\bar{b}_{k,j}^t}{\bar{a}_{k,j}}$ in the decentralized approach.

We assume a connected, undirected communications network. We assume a consensus matrix, based on the Metropolis-Hastings algorithm [16], given by

$$[W]_{i,j} = \begin{cases} \frac{\max(d_i,d_j)}{1 + \sum_{k \in N_i} \frac{1}{\max(d_i,d_k)}}, & \text{if } j \in N_i \\ \frac{1}{\max(d_i,d_k)}, & \text{if } j = i \\ 0, & \text{else} \end{cases}$$

where $N_m$ is the set of nodes in the (network) neighborhood of sensor node $m$, excluding itself, and $d_m = |N_m|$ is the degree of node $m$. The Metropolis-Hastings algorithm is a good fit for decentralized computations. The consensus matrix based on the Metropolis-Hastings algorithm requires local neighborhood information only (i.e., the degrees of neighbor nodes). Additionally, the Metropolis-Hastings algorithm and its convergence properties are well-studied for sampling from distributions [18], and its performance is comparable to other heuristic-based algorithms [16].

We can write an equivalent, compact expression of the consensus method across the network. Define the vector of local statistics at node $k$ from equations (4)-(7) as $\beta_k^t = [b_{k,1}^t, \ldots, b_{k,j}^t, v_{k,1}^t, \ldots, v_{k,j}^t, s_{k,1}^t, \ldots, s_{k,j}^t, a_{k,1}, \ldots, a_{k,j}]^T$, where $v_{k,j} = \text{vec}(V_{k,j}^t)$. The consensus updates of the sample average of the local statistics across all nodes can be written as

$$[\bar{\beta}_1^t, \bar{\beta}_2^t, \ldots, \bar{\beta}_M^t] = \left[\beta_1^{t-1}, \beta_2^{t-1}, \ldots, \beta_M^{t-1}\right] W,$$

where $t$ indicates the consensus iteration and is initialized as $\bar{\beta}_k^0 = \beta_k^0$ from iterate $t$ of EM. The iterates of consensus using the Metropolis-Hastings method are guaranteed to converge to the sample average provided the network is connected and not bipartite [17]. Thus, $\bar{\beta}_k^t \rightarrow \frac{1}{M} \sum_{m=1}^M \beta_m^t$ as $t \rightarrow \infty$, for each $k = 1, 2, \ldots, M$.

Each node requires a stopping criterion to exit the consensus iterations. The stopping criterion is usually based on the global optimization measure, which in this case is the joint data log-likelihood. The log-likelihood can be written in terms of a sum of local contributions, $\ell(\theta) = \sum_{m=1}^M \ell_m(\theta)$, where

$$\ell_m(\theta) = \sum_{i=1}^{N_m} \log \sum_{j=1}^{J} \alpha_{m,j} N(x_{m,i} | \mu_{j}, \Sigma_{j})$$

As a stopping criterion for the consensus iterations, we define for each node $k = 1, 2, \ldots, M$ the following consensus update error

$$e_k(t') = |\bar{p}_k^t - \bar{p}_k^{t'-1}|,$$

where $\bar{p}_k^t$ is an estimate after $t'$ consensus iterations of the average data log-likelihood, $\frac{1}{M} \sum_{m=1}^M \ell_m(\theta)$, with $\ell_k(\theta)$ evaluated at $\theta = \theta_k^t$ at node $k$. For sufficiently large $t'$, the per-step error behaves according to

$$\log e_k(t') \leq t' \log \rho(W') + \log c_k,$$

for some finite constant $c_k$ and where $\rho(W')$ is the spectral radius of $W' = W - \frac{1}{N} 11^T$ [17]. Since $\rho(W') < 1$ for a connected and non-bipartite network, the local consensus error measure $e_k(t)$ converges to 0 as $t \rightarrow \infty$ for all nodes $k = 1, 2, \ldots, M$.

5. NUMERICAL EXAMPLES

In this section, we provide an example using synthetic data to demonstrate the effectiveness of the decentralized algorithm to approximate a nonlinear, low-dimensional data manifold. As numerical examples, both [4] and [5] apply their techniques to a MFA model approximating a shrinking spiral. We use the shrinking spiral example for comparison with these centralized approaches with published results.

For this numerical example, the number of sensor nodes is $M = 9$ and the number of mixture components is $J = 12$. The number of data points per node is $N_m = N = 100$ for $m = 1, 2, \ldots, M$ and is generated according to

$$x = [(13 - 0.5t) \cos t, (0.5 - 13 \sin t, t)]^T + w,$$

where $t \in [0, 4\pi]$ and $w \sim N(0, I_j)$. The dimension of the observations is $p = 3$ while the lower intrinsic dimension is $r = 1$. Each sensor node observes equal-length segments of the data manifold with $50\%$ overlap of adjacent segments of other nodes, not necessarily those of network neighbors. Sensor nodes should observe at least one mixture component seen by at least one other sensor node.

Figure 1 shows the communications network structure. Each sensor node is a vertex on the graph, and each edge between vertices represents an undirected communications link. For the Metropolis-Hastings-based consensus matrix $W$ and the network graph in Fig. 1, the resulting convergence rate is $\rho(W - \frac{1}{N} 11^T) = 0.8345$.

The decentralized EM algorithm is initialized with estimates from a consensus-based k-means clustering (see for example [12]). While the objective functions between k-means and EM are different, the resulting update equations for the mixture means (i.e., equations (4), (5), (9), and the corresponding consensus updates of (4) and (5)) are identical, with the exception that the posterior probabilities $w_{m,i,j}^{t+1}$ are hard assignments in k-means instead of soft ones. In our experi-
ence, the consensus-based k-means converges much faster in both consensus and clustering due to the hard assignments. Since k-means initialization can be seen as a simplification and a special case of the EM algorithm [19, Chapter 9.3], we will not discuss it further here.

Figures 2(a) and 2(b) plot the noiseless decaying spiral (solid blue curve), sample observation points from all M nodes (black scatter points), and 2-σ line segments (red segments) from estimates of the global parameters. Note that a 2-σ segment from each node is plotted for all J components (i.e., 108 total line segments). In Fig. 2(a) and Fig. 2(b), the number of consensus steps are 5 and 20, respectively, per EM step in estimating the global parameters. The sample root-mean-squared-error is defined by

\[ s\text{-rmse}(t) = \sqrt{\sum_{m=1}^{M} \sum_{i=1}^{N_m} \sum_{j=1}^{J} \frac{1}{p_{m,j}} \left\| \left( 1 - P_j(\theta_m^t) \right) (x_{m,i} - \mu_{m,j}) \right\|_2^2} \]

where \( P_j(\theta) = \Lambda_j (\Lambda_j^T \Lambda_j)^{-1} \Lambda_j^T \) and, given \( \theta = \theta_m^t \), projects the point \( (x_{m,i} - \mu_{m,j}) \) onto the \( j \)th subspace estimated by node \( m \). After 40 EM iterations, the s-rmse is 0.8328 and 0.8326 for 5 and 20 consensus steps per EM step, respectively. Surprisingly, the average data-log-likelihood with just 5 consensus steps (-3181.7) is slightly greater than that with 20 consensus steps (-3182.6). However, Fig. 2(a) and Fig. 2(b) show that the local estimates of the global parameters are in better agreement across the network with more consensus steps. The agreement is made clearer in Fig. 3.

Figure 3 plots the local update error of the data-log-likelihood, given in equation (14), versus consensus iterations at the \( t = 5 \) iterate of EM steps. The solid lines are the update errors measured at each of the \( M = 9 \) nodes, and the dashed line represents the predicted convergence rate of \( \rho(W') = 0.8345 \) for the simulated network. This demonstrates that after a small number of steps, the consensus updates begin to behave as expected – linear (in the exponent) convergence. Additionally, the sensor nodes have about an order of magnitude better agreement after 20 consensus steps compared with 5 steps. More consensus iterations improve agreement, but at the expense of additional communications.

Figures 4 and 5 show the local update error of the global parameters, given by \( \left| \theta_b^{t+1} - \theta_b^t \right| \) for node \( k = 1, 2, \ldots, M \), and the average data-log-likelihood, \( \bar{f}_b^{t+1} \), versus EM iteration number. The error and average log-likelihood curves are plotted for each \( M = 9 \) sensor nodes, and each are generated after 20 consensus steps per EM iteration. Both figures show tight consensus between the nodes. As seen in Fig. 5, despite the intermediate consensus steps, the data-log-likelihood appears non-decreasing as expected in standard EM algorithms. For comparison, error and log-likelihood curves are plotted in Figs. 4 and 5 for a centralized EM (i.e., assuming the E-step and M-step updates of Section 3 are available at a central node). After 40 iterations, the centralized EM has an s-rmse and average log-likelihood of 0.8374 and -3182.2, respectively.
demonstrate the efficacy of the algorithm. We have presented a decentralized EM-based algorithm for estimating the parameters of a mixture of factor analyzers. The likelihood for the centralized EM algorithm.

Fig. 4. Local update error of the global parameters, $||\theta_k^{t+1} - \theta_k^t||$ for $k = 1, 2, \ldots, M$ versus EM iterations. All $M = 9$ error curves from the decentralized EM algorithm are plotted. The dashed curve corresponds to the update error for the centralized EM algorithm.

Fig. 5. Average data-log-likelihood versus EM iterations. All $M = 9$ log-likelihood curves of the decentralized EM algorithm are plotted. The dashed curve corresponds to the log-likelihood for the centralized EM algorithm.

6. CONCLUSION
We have presented a decentralized EM-based algorithm for estimating the parameters of a mixture of factor analyzers. This model approximates the joint distribution of noisy observations of a nonlinear manifold from a network of sensor nodes. We have provided results based on simulated data to demonstrate the efficacy of the algorithm.

7. REFERENCES