DISTRIBUTED EM ALGORITHMS FOR DENSITY ESTIMATION IN SENSOR NETWORKS

Robert Nowak

ECE Department, Rice University, Houston, TX 77005

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. A distributed EM 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. 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 discussed, and simulations demonstrate the potential of this approach to sensor network data analysis.

1. INTRODUCTION

The slogan "the sensor is the network," 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. Viewing the network as a single sensing entity motivates the basic 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 cenralized 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 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 to estimate 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.

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.

2. PROBLEM STATEMENT

Assume that we have M nodes, $1, \ldots, M$. The m-th node senses and records N_m independent and identically distributed (i.i.d.) measurements $y_{m,1}, \ldots, y_{m,N_m}$. The i.i.d. assumption implies that the environmement is stationary and unchanging during 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 Gaussian mixture distributions of the form

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

where the mixing parameters $\{\alpha_{j,m}\}$ 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 2 shows sensor network data in a simulated experiment.

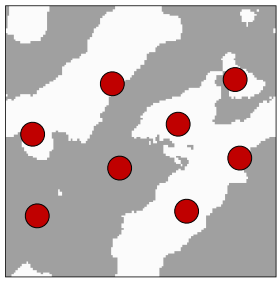

Fig. 1. Sensor network in an inhomogeneous environment. Discs represent nodes in the sensor network. Background represents spatially varying environmental conditions being sensed by the nodes.

Define $\phi \equiv \{\mu_j^t, \Sigma_j^t\}_{j=1}^J$, 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

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

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

3. 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].

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

$$Q(\theta,\theta^t) = E_{\theta^t}\left[\log p(y,z|\theta)\right], \quad (2)$$

where $p(y,z|\theta)$ denotes the joint distribution of y and z with parameters θ and E_{θ^t} denotes expectation with respect to the joint probability law with parameters θ^t. This is the usual missing data formulation, and it is easy to verify [1] 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

$$w_{m,i,j}^{t+1} = \frac{\alpha_{m,j}^t\mathcal{N}(y_{m,i}|\mu_j^t,\Sigma_j^t)}{\sum_{k=1}^J\alpha_{m,k}^t\mathcal{N}(y_{m,i}|\mu_k^t,\Sigma_k^t)}. \quad (3)$$

From this it is easy to see that the E-Step, computing the conditional expectation $Q(\theta,\theta^t)$, boils down to computing $\{w_{m,i,j}^{t+1}\}$ according to (3). 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 $j = 1,\ldots,J$

$$\mu_j^{t+1} = \frac{\sum_{m=1}^M\sum_{i=1}^{N_m} w_{m,i,j}^{t+1} y_{m,i}}{\sum_{m=1}^M\sum_{i=1}^{N_m} w_{m,i,j}^{t+1}},$$

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

Under certain conditions, the EM algorithm converges to a local maximum of the log likelihood function. It can be shown that in certain cases EM converges more rapidly than gradient methods [2], and in certain cases the convergence rate is superlinear, comparable to that of Newton-type methods [3]. On the other hand, overall EM is a conservative algorithm with better stability properties than more aggressive schemes such as Newton's method. These facts make EM a good choice for mixture estimation in general, and distributed (and unsupervised) applications like those arising in sensor networks especially.

In anticipation of a distributed version of this EM algorithm, define the "sufficient" statistics

$$w_j^t = \sum_{m=1}^M\sum_{i=1}^{N_m} w_{m,i,j}^t,$$

$$a_j^t = \sum_{m=1}^M\sum_{i=1}^{N_m} w_{m,i,j}^t y_{m,i}$$

$$b_j^t = \sum_{m=1}^M\sum_{i=1}^{N_m} w_{m,i,j}^t y_{m,i}^2. \quad (4)$$

Notice that with these summaries the M-Step calculations for the means and covariances can be computed according to

$$\mu_j^t = \frac{a_j^t}{w_j^t},$$

$$\Sigma_j^t = \frac{b_j^t}{w_j^t} - \mu_j^t(\mu_j^t)', \quad (5)$$

where the superscript $'$ denotes matrix transposition (unnecessary in the scalar case).

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. In the first cycle, each message passing operation involves the transmission of the partial sum of the sufficient statistics in (4) from one node to another. Upon the completion of one full cycle, the last node has the complete sums in (4), which are then passed backwards in the cycle. Each node now has the summary statistics in (4) and can compute the update θ^{t+1}. Note that each iteration of the EM algorithm requires the transmission of $2M-2$ messages of dimension $\dim(s^t)$.

4. 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.

The DEM algorithm operates as follows. Initialize $\{\mu_j^0,\Sigma_j^0,p_{j,m}^0\}$ 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 according to (5), and

$$w_{m,i,j}^{t+1} = \frac{\alpha_{m,j}^t \mathcal{N}(y_{m,i}|\mu_j^t, \Sigma_j^t)}{\sum_{k=1}^J \alpha_{m,k}^t \mathcal{N}(y_{m,i}|\mu_k^t, \Sigma_k^t)}. \quad (6)$$

Then node m updates its mixing probabilities according to

$$\alpha_{m,j}^{t+1} = \frac{1}{N_m} \sum_{i=1}^{N_m} w_{m,i,j}^{t+1}, \quad (7)$$

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

$$\begin{aligned} w_j^{t+1} &= w_j^t + w_{m,j}^{t+1} - w_{m,j}^t, \\ a_j^{t+1} &= a_j^t + a_{m,j}^{t+1} - a_{m,j}^t, \\ b_j^{t+1} &= a_j^t + b_{m,j}^{t+1} - b_{m,j}^t. \end{aligned} \quad (8)$$

Here, all that is done is that the old values of the sufficient statistics are being replaced by updated values. 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. 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 DEM algorithm can be viewed as a type of incremental EM algorithm. Incremental versions of EM were first considered in [4]. It follows from the general theory of incremental EM algorithms, DEM monotonically converges to a local maximum (or saddle point). 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 (5)-(8) 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 "DEM with multiple EM steps at each node" as opposed to "DEM with a single EM step 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).

5. CONVERGENCE BEHAVIOR OF DISTRIBUTED EM

Although the DEM is intuitively reasonable, a formal convergence proof of incremental versions of EM was not given in [4]. Recently, an analysis was presented in [5] which shows that the accumulation points of incremental EM algorithms are are fixed points of the corresponding standard (global) EM algorithms. This analysis can be specialized to apply to DEM. The convergence behavior of standard EM in the Gaussian mixture case is examined thoroughly in [2, 3]. Usually, the EM fixed points are points of local maxima of the log likelihood (although saddle points are also possible). It can be shown that in the Gaussian mixture case DEM is linearly convergent (in a certain sense) to a local maximum of the log likelihood $l_y(\theta)$ [6].

Assume that the sequence $\{\theta^t\}$ converges to a point θ^* where the log likelihood l_y assumes a local maximum. It can be shown [6] that for sufficiently large t there exists a constant $0 \leq \beta < 1$ such that

$$\|\theta^t - \theta^*\| \leq \beta \|\bar{\theta}^{t-1} - \theta^*\|, \quad (9)$$

where $\bar{\theta}^{t-1}$ is a weighted average of the past $\{\theta^{t-m}\}_{m=1}^M$. This result is crucial in applications since it ensures 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. The precise form of the weighted average $\bar{\theta}^{t-1}$ depends on the nature of the distribution (see [6] for details). In cases in which all sensors make i.i.d. observations in equal and sufficient numbers, then

$$\bar{\theta}^t \approx \frac{1}{M} \sum_{m=1}^M \theta^{t-m+1}, \quad (10)$$

a simple average of the past M iterates. From here it follows that $\|\theta^{t+1} - \theta^*\| \leq \beta \max_{m=1,\ldots,M} \|\theta^{t-m+1} - \theta^*\|$, demonstrating that the error converges at least linearly over each cycle.

6. A SIMULATED SENSOR NETWORKING APPLICATION

A simulated sensor network application is presented here to illustrate the proposed ideas. Consider a network of M wireless nodes, each equipped with two sensors, a temperature sensor and a sensor that measures the presence of 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 [8]. 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 2(a)-(d) depict 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 with single EM step at each node, and DEM with 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 circles in Figure 2(e).

All three algorithms converged to the same solution. The solid ellipsoids in Figure 2(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 there 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 2(f). Clearly the DEM algorithm with multiple EM steps per node converges most rapidly in this case. 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). Other experiments (not discussed here) demonstrated similar behavior and performance.

7. 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. 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. A simulation study demonstrated the potential of DEM for sensor network data analysis.

There are several potential avenues for future work. The crucial question of selecting the appropriate number of components in the Gaussian mixture might be dealt with by incorporating the MML criterion proposed in [9]. The density estimates or data clusters derived by the DEM algorithm could play a role in subsequent processing and analysis, perhaps in the optimization of distributed coding techniques [7]. 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.

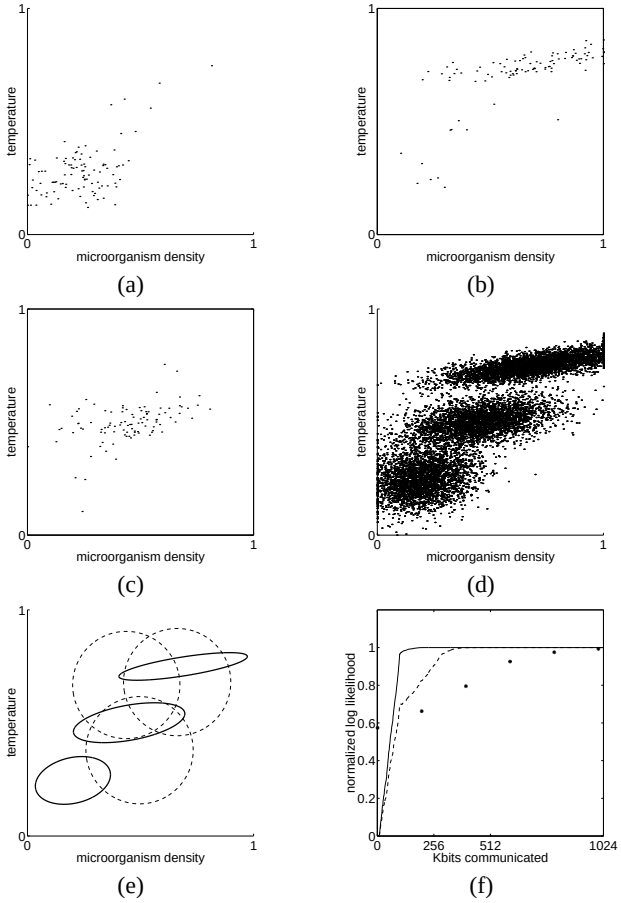

Fig. 2. Sensor network simulation. (a)-(c) Representative cases of data collected at three different nodes. (d) Scatter plot of data collected at all sensors. (e) Estimated Gaussian components (solid), initial Gaussian components (dashed). (f) Log likelihood as a function of communicated bits (assuming 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 (*).

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

[3] 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.

[4] 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.

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

[6] R. Nowak, "Distributed EM algorithms for density estimation and data clustering in sensor networks," Technical Report TREE0203, Department of Electrical and Computer Engineering, Rice University, October 2002.

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

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

[9] 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.