LAZY INFERENCE ON OBJECT IDENTITIES IN WIRELESS SENSOR NETWORKS

Jaewon Shin, Nelson Lee, Sebastian Thrun and Leonidas Guibas

Stanford University
Stanford, CA 94305

ABSTRACT

Tracking the identities of moving objects is an important aspect of most multi-object tracking applications. Uncertainty in sensor data, coupled with the intrinsic difficulty of the data association problem, suggests probabilistic formulations over the set of possible identities. While an explicit representation of a distribution over all associations may require exponential storage and computation, in practice the information provided by this distribution is accessed only in certain stylized ways, as when asking for the identity of a given track, or the track with a given identity. Exploiting this observation, we proposed in [1] a practical solution to this problem based on maintaining marginal probabilities and demonstrated its effectiveness in the context of tracking within a wireless sensor network. That method, unfortunately, requires extensive communication in the network whenever new identity observations are made, in order for normalization operations to keep the marginals consistent [2]. In this paper, we propose a very different solution based on accumulated *log-likelihoods*, which can postpone all normalization computations until actual identity queries are made. In this manner the continuous communication and computational expense of repeated normalizations is avoided and that effort is expended only when actual queries are made of the network. We compare the two methods in terms of their computational complexities, inference accuracies, and distributed implementations. Simulation and experimental results from a RFID system are also presented.

1. INTRODUCTION

A wireless sensor network (WSN) is a large scale distributed system consisting of small, untethered, low-power nodes capable of sensing, processing and communicating. WSNs are unique in their ability to monitor phenomena widely distributed in space and time, such as microclimate variations in a forest, earthquake vibration monitoring in buildings, machine control and diagnosis in factories, traffic monitoring in highways, etc.

In many of these scenarios it is infeasible to have the WSN simply collect all potentially relevant data. Instead, it is far more efficient to be able to query the network as the need for particular kinds of information arises. Such queries can often be formulated as distributed inference problems, where the goal is to estimate a global quantity or state of interest X, given local pieces of evidence provided by the sensor nodes. Furthermore, this inference should be probabilistic in nature, due to the inherent noise in sensor readings and the uncertainty associated with physical phenomena. In this setting, it is very important to capture the *information structure* of the problem and the dependencies between both the problem's local and global variables. For example, say we are tracking a large chemical plume in a region R using a WSN, and are assuming that we know the total amount T of the chemical involved. If a sensor locally determines the quantity of the chemical, say, t in a subregion r, then we also know that there is $T - t$ of the chemical in the rest of the region $R - r$, thus updating the global information.

In this paper, we study the problem of tracking identities of multiple moving objects within a WSN – what we call the *identity management* problem. The significance of identity management lies in that it can overcome incorrect *identity swapping* due to sub-optimal data association algorithms using only local object identity information, as shown in Figure 1.[1] This problem has an interesting information structure – local evidence about the identity, say, i of an object implies all the other objects cannot have identity i. This *exclusion* among identities defines a mathematical relation among identities of moving objects and allows us to exploit local knowledge to update global information.

In our earlier work [1], we proposed a practical solution to the above problem which, for N moving objects, maintains N^2 marginal probabilities of each track having each identity and normalizes/updates these probabilities whenever new local evidence becomes available. This normalization requires communication among nodes in the WSN in the vicinity of the objects and can potentially affect performance of the WSN by draining node energy quickly. However, we would prefer a method that computes probabilities only at a user's request, since renormalizing probabilities all the time is wasteful of energy. In the current work we seek to maintain different quantities, which can be converted to probabilities at a user's request, but do not require normal-

[1] The data association problem is known to be NP-hard.

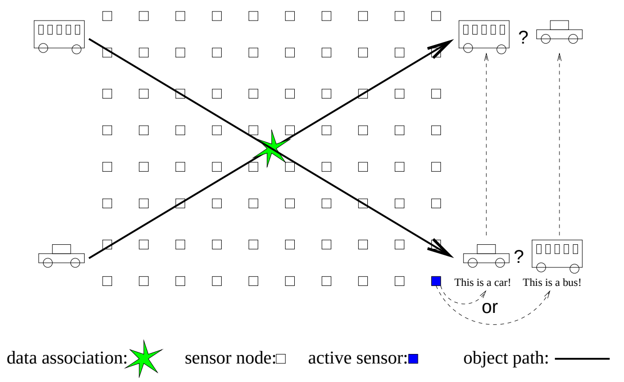

Fig. 1. Identity swapping due to sub-optimal data association and its correction using local identity information at a sensor

ization otherwise. In other words, we seek to accumulate information in a *lazy* fashion, from which the desired probabilities can be derived on demand. This is the crux of the identity management approach in this paper.

The main contributions of the paper are as follows. First, we introduce a general mathematical framework for identity management, including a formulation in which no information is lost. Our previous work [1] turns out to be an approximation to this optimal strategy. Second, we propose another feasible approximation to the optimal identity management problem that allows us to evaluate probabilities in a lazy fashion and retains practical storage and computational complexities. Finally, we demonstrate the effectiveness of our method through simulations and real experiments with a real-time people tracking system augmented with RFID readers.

2. OPTIMAL IDENTITY MANAGEMENT

We are interested in maintaining identities of N moving objects, such as people or cars, using only local evidence from sensor nodes in the WSN. We first define the notion of the *identity state* X of the N objects in this setting.

Definition 1. *The joint identity state of the N objects is $X = (x_1, \cdots, x_N)$, where x_j is the marginal identity state for the j_{th} object. The quantity x_j can have a value $x_j = i \in \{1, \cdots, N\}$, indicating that the j_{th} physical object has an identity i. No two different objects can have the same identity.*

According to the above definition, X can take on $N!$ different permutations – $X \in S_N$, where S_N is the symmetric group on N elements.[2] S_N can be represented by the set of all $N \times N$ *permutation matrices* – 0-1 $N \times N$ matrices with exactly one 1 in each row and column, each of which represents an identity assignment between a physical object and an identity. For example, two joint states $X = (x_1 = 1, x_2 = 2, x_3 = 3)$, $X' = (x_1 = 2, x_2 = 1, x_3 = 3)$ in the $N = 3$ case can be represented as the following permutation matrices.

$$X = (x_1 = 1, x_2 = 2, x_3 = 3) \quad \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I$$

$$X' = (x_1 = 2, x_2 = 1, x_3 = 3) \quad \rightarrow \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I_{(1,2)}$$

where I is an identity matrix and $I_{(1,2)}$ is an matrix obtained by swapping i_{th} and j_{th} columns of I. Throughout this paper, we will use $(x_1, \cdots, x_N)$ or permutation matrices to denote instances of an identity state X.

We define a joint probability distribution over all $N!$ identity assignments, $p(X)$, $X \in S_N$. When objects are moving in a sensor network, there are two kinds of *events* that modify this distribution – *mixing* events and *local evidence* events, as shown in Figure 2. Intuitively, a *mixing* event happens when two object locations are so close that their identities are no longer distinguishable. This will increase the uncertainty of identity assignments in $p(X)$. A *local evidence* event happens when a sensor node makes measurements on the identity of a specific object[3] and updates $p(X)$ using Bayes rule. A local evidence event reduces the entropy of $p(X)$ in general. Our goal is to maintain $p(X)$ on-line while these two types of events are occurring. The two events will be precisely defined in the following sections.

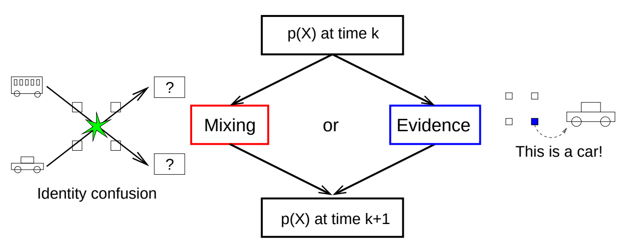

Fig. 2. How $p(X)$ evolves while objects are moving (mixing events) and sensors are sensing (local evidence events)

For the sake of brevity, in the sequel we may drop the word "event" and talk simply of a "mixing", or of a "Local evidence".

[2]The symmetric group S_N is the set of all permutations on N objects under permutation composition.

[3]In a WSN setting, objects may pass near sensor nodes which can then determine their identity, either through signal classification techniques or directly, as in the case of RFID tag readers.

2.1. Mixing

Mathematically, a mixing corresponds to a convolution operation between a joint probability distribution $p(X)$ and a mixing probability distribution $m(X)$.

Definition 2. *Suppose the identities of i_{th} and j_{th} objects are mixed - due to their geographical proximity, nearby sensors can't distinguish them. The joint identity distribution $p(X)$ is updated using the convolution,*

$$p \star m\ (X = x) = \sum_{s \in S_n} p(s) m(xs^{-1})$$

where

$$m(X) = \begin{cases} 1-\alpha & X = I \\ \alpha & X = I_{(i,j)} \\ 0 & \text{otherwise;} \end{cases}$$

here α denotes the mixing probability, I is the $N \times N$ identity matrix and $I_{(i,j)}$ is its (i,j)-transposition.

Therefore, the above convolution can be simply written as

$$p \star m\ (X = x) = (1-\alpha) p(x) + \alpha p(x I_{(i,j)})\,. \quad (1)$$

According to the above definition, the following is true.

Lemma 1. *The statistical entropy of the joint identity distribution can only increase after a mixing.*

$$H(q(X)) \geq H(p(X))$$

Proof. The above convolution operation can be represented as the following matrix multiplication:

$$\overrightarrow{q} = M \overrightarrow{p}\,,$$

where $\overrightarrow{p}$ and $\overrightarrow{q}$ are $N! \times 1$ vectors representing the two distributions and M is a $N! \times N!$ mixing matrix, in which each row (and column) has only two non-zeros values, α and $1-\alpha$. Since M is a doubly stochastic matrix, we can re-write the right side of the above equation as a convex combination of the $(N!)!$ permutation matrices. Using the concavity of the statistical entropy function, it follows that:

$$\begin{aligned} H(\overrightarrow{q}) &= H(\sum_i \alpha_i \Pi_i \overrightarrow{p}) \\ &\geq \sum_i \alpha_i H(\Pi_i \overrightarrow{p}) \\ &= \sum_i \alpha_i H(\overrightarrow{p}) \\ &= H(\overrightarrow{p}) \end{aligned}$$

This concludes the proof. □

The claim of the above lemma, that the uncertainty never decreases with mixing, certainly agrees with our intuition that mixing only adds uncertainty. After repeated mixings, the identity state distribution converges to the uniform distribution.

2.2. Local Evidence

A local evidence is a piece of information on the identity of an object and can be though of as a likelihood function. We will use the following sensor measurement model for identity sensing.

Definition 3. *The likelihood of an identity measurement Z on the i_{th} object at a sensor node is defined as follows.*

$$p(Z = (i,j)|X) = \begin{cases} 0.9 & x_j = i \\ 0.1/(N-1) & x_j \neq i \end{cases}$$

From the above definition, given a set of local evidence events $Z_1, \cdots, Z_k$, Bayes rule derives the posterior distribution $p(X|Z_1, \cdots, Z_k)$ as follows:[4]

$$p(X|\mathbf{Z} = Z_1, \cdots, Z_k) \propto p(X) p(Z_1|X) \cdots p(Z_k|X)\,.$$

2.3. Identity Management as Discrete Bayesian Filtering

The two operations on $p(X)$ defines a *discrete Bayesian filter* on $p(X)$, completely analogous to usual continuous Bayesian filtering – the mixing (convolution) corresponds to the *prediction* step in the continuous filtering and the local evidence corresponds to the *observation or likelihood incorporation* step. The only difference is that the prediction step in discrete Bayesian filtering happens only at discrete-time mixing events, while the prediction step in the continuous case happens at every time step.

Now, we consider a case where we have both local evidence and mixing together. For example, say we are given events $m_1(X)$, Z_1, Z_2, $m_2(X)$ in that order (here m denotes mixing and Z local evidence events). The posterior can be computed as[5]

$$p(X) \propto [\{p_0(X) \star m_1(X)\} p(Z_1|X) p(Z_2|X)] \star m_2(X)\,.$$

The above computation , unfortunately, is practically infeasible due to the exponential complexity of the two operations – convolution from a mixing is $O((N!)^2)$ and Bayesian normalization from a local evidence is $O(N!)$. Therefore, we need **to approximate the joint distribution $p(X)$ together with the two operations so that we can implement them in a WSN**. In [1], we proposed a practical approximation based on the marginal probabilities. In the upcoming sections, we will introduce another practical approximation based on log-likelihoods, the so-called *information matrix based approach*.

[4] We assume measurements are conditionally independent given X, i.e., $p(Z_1, Z_2|X) = p(Z_1|X)p(Z_2|X)$.

[5] The real computation is done in an iterative fashion – whenever there is a mixing or a local evidence event, we update $p(X)$ accordingly.

3. INFORMATION MATRIX APPROACH

3.1. Information filtering

The second approximation is based on the idea of *information filtering* introduced in [3]. As in the previous approach, we maintain a $N \times N$ matrix, now called the *information matrix*, whose elements are sum of log-likelihoods as follows.

$$L = \begin{bmatrix} l_{11} & \cdots & l_{1i} & \cdots & l_{1N} \\ l_{21} & \cdots & l_{2i} & \cdots & l_{2N} \\ l_{31} & \cdots & l_{3i} & \cdots & l_{3N} \\ & & \vdots & & \\ l_{N1} & \cdots & l_{Ni} & \cdots & l_{NN} \end{bmatrix},$$

where l_{ij} is the sum of log-likelihoods that object i has identity j. Specifically, if Z^t is a sensor measurement at time t, then

$$l_{ij} = \sum_t \log(p(Z^t = (k,j)|x_j = i)),\ k \in \{1, \cdots, N\}.$$

In other words, the information matrix is obtained by adding $\log(.9)$ to $(i,j)_{th}$ element and $\log(\frac{.1}{N-1})$ to all the other elements in j_{th} column whenever there is a measurement $Z_{(i,j)}$ – assuming that L is initialized as a zero matrix.

A striking property of information matrix is that the $N!$ joint likelihoods $l(X = \Pi_k)$ can be recovered from an information matrix L, which is just a collection of N^2 log-likelihoods. For example, suppose L and $X = \Pi_k$ are given as follows,

$$L = \begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \Pi_k = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then the following is true.

$$\begin{aligned} \log(l(X = \Pi_k)) &= \log(p(Z = (2,1)|\Pi_k) \\ &+ \log(p(Z = (1,2)|\Pi_k) \\ &+ \log(p(Z = (3,3)|\Pi_k) \\ &= l_{22} + l_{12} + l_{33}. \end{aligned}$$

Using matrix algebra, we can simplify the above equation for joint likelihoods as follows:

$$l(X = \Pi_k) \propto \exp(\text{Tr}(\Pi_k^T L)),$$

where Π_k is k_{th} permutation matrix and $\text{Tr}(\cdot)$ is a matrix trace operation – the sum of the diagonal elements. If the prior distribution is uniform, then the joint distribution $p(X)$ is simply the normalized joint likelihoods.

$$p(X = \Pi_k) = \frac{l_k}{\sum_i l_i} = \frac{\exp(\text{Tr}(\Pi_k^T L))}{\sum_l \exp(\text{Tr}(\Pi_l^T L))}.$$

Another interesting property of the information matrix is that there are infinitely many information matrices that encode the same joint distribution.

Property 1. *Adding a constant to each element of any of the rows or columns of an information matrix does not affect its underlying joint likelihoods.*

Proof. Suppose C is a matrix, whose i_{th} column (or row) is c and all the other elements are zero. Then,

$$\begin{aligned} l_k &\propto \exp(\text{Tr}(\Pi_k^T (C + L))) \\ &= \exp(\text{Tr}(\Pi_k^T C)) \exp(\text{Tr}(\Pi_k^T L)) \\ &= \exp(c) \exp(\text{Tr}(\Pi_k^T L)). \end{aligned}$$

The likelihoods l_k do not change when all l_k are scaled by the same factor $\exp(c)$ – they will be normalized anyway. This concludes the proof. □

Due to the above property, we can simplify l_{ij} as follows.

$$l_{ij} = n_{ij}(\log(.9) - \log(\frac{.1}{N-1})),$$

where n_{ij} is the number of $Z_{(i,j)}$ measurements observed thus far – counts of $Z_{(i,j)}$. This gives another interpretation of the information matrix – as a collection of *counts of evidence*.

3.2. Local Evidence and Information Matrices

As we have seen before, incorporating local evidence $Z_{(i,j)}$ into an information matrix is trivial – we just add $\log(.9) - \log(\frac{.1}{N-1})$ to the $(i,j)_{th}$ element of L and there is no need to re-normalize.

3.3. Mixing and Information Matrices

Let L_p and L_q be information matrices corresponding to distributions $p(X = \Pi_k)$ and $q(X = \Pi_k) = p \star m$ $(X = \Pi_k)$, respectively. We will use the mixing ratio $\alpha = \frac{1}{2}$ in the sequel for simplicity, so after mixing $p(\Pi_k) = p(\Pi_k I_{(i,j)})$. From the definition of mixing in joint space, we can write down the following equation, whose solution is the new information matrix L_q after mixing:

$$\frac{\exp(\text{Tr}(\Pi_k^T L_q))}{\sum_l \exp(\text{Tr}(\Pi_l^T L_q))} = \frac{1}{2}\left[p(\Pi_k) + p(\Pi_k\ I_{(i,j)})\right]. \quad (2)$$

To compute L_q, we need to solve a set of $\binom{N!}{2}$ equations given below to take into account the normalization constraint.

$$\frac{q(\Pi_m)}{q(\Pi_n)} = \frac{\exp(\text{Tr}(\Pi_m^T L_q))}{\exp(\text{Tr}(\Pi_n^T L_q))} = \frac{p(\Pi_m) + p(\Pi_m\ I_{(i,j)})}{p(\Pi_n) + p(\Pi_n\ I_{(i,j)})},$$

where $m \neq n$ and $m, n \in \{1, \cdots, N!\}$. If we further simplify the above equation by taking logs on both sides, we get

$$\mathrm{Tr}((\Pi_m - \Pi_n)L_q) = \log\left(\frac{p(\Pi_m) + p(\Pi_m\ I_{(i,j)})}{p(\Pi_n) + p(\Pi_n\ I_{(i,j)})}\right).$$

The left side of the above equation is just a linear combination of elements of L_q and the right side is a constant. Therefore, we can write a matrix equation with proper vectorization

$$\mathbf{\Phi}\,\overrightarrow{\mathbf{l}} = \overrightarrow{\eta}\,,$$

where Φ is a $\binom{N!}{2} \times N^2$ matrix, $\overrightarrow{\mathbf{l}}$ is a $N^2 \times 1$ vector, and $\overrightarrow{\eta}$ is a $\binom{N!}{2} \times 1$ vector. However, there is no exact solution in general for this overdetermined set of equations. The least square solution can be used as an approximate solution, although it is not practical due to the prohibitive amount of computation – the pseudo inverse of a $\binom{N!}{2} \times N^2$ matrix must be computed.

The above discussion suggests that we need more constraints in order to facilitate the derivation of the information matrix after mixing. Specifically, we assume that after a mixing event between the i_{th} and j_{th} objects, in the information matrix:

- **Only the i_{th} and j_{th} columns are modified.**
- **The i_{th} and j_{th} columns will remain the same.**

The first assumption reduces the number of unknowns and also seems reasonable – why change the other columns when mixing involves only two objects? We introduce the second assumption that the probabilities of a permutation Π_k and its transposition $\Pi_k I_{(i,j)}$ will be the same after mixing, since the information matrix with the two same columns has the following property.

Property 2. *For an information matrix L, whose i_{th} and j_{th} columns are the same, its joint likelihood $l(\Pi_k)$ and $l(\Pi_k\ I_{(i,j)})$ are the same.*

Proof. The log-likelihood of Π_k is the sum of l_{mn} entries at positions where there are 1's in Π_k. Since Π_k and $\Pi_k I_{(i,j)}$ are the same except that their i_{th} and j_{th} columns have been swapped, the difference of the two log-likelihoods is given as follows,

$$\begin{aligned}\log(l(\Pi_k)) - \log(l(\Pi_k I_{(i,j)})) &= l_{mi} + l_{nj} - l_{mj} - l_{ni} \\ &= 0\,,\end{aligned}$$

since the i_{th} and j_{th} columns of L are the same. □

Under these two assumptions, the number of the unknowns in the updated information matrix is only N. Let us consider the simple case of $N = 3$ to see how these assumptions can simplify the computation of L after a mixing. L_p^* and L_q^* are exponential versions of L_p and L_q respectively. ($l_{ij}^* = \exp(l_{ij})$) and their elements are given as follows. Again, note that only the d_i's are unknowns:

$$L_p^* = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}, L_q^* = \begin{bmatrix} d_1 & d_1 & c_1 \\ d_2 & d_2 & c_2 \\ d_3 & d_3 & c_3 \end{bmatrix}.$$

From equation (2), the following holds, assuming that the mixing is between the i_{th} and j_{th} columns.[6]

$$\begin{aligned} d_1\ d_2\ c_3 &= (a_1\ b_2\ c_3 + a_2\ b_1\ c_3)/2\,, \\ d_1\ d_3\ c_2 &= (a_1\ b_3\ c_2 + a_3\ b_1\ c_2)/2\,, \\ d_2\ d_3\ c_1 &= (a_2\ b_3\ c_1 + a_3\ b_2\ c_1)/2\,. \end{aligned}$$

Taking logs on the both sides, we get

$$\begin{aligned} \log(d_1) + \log(d_2) &= \log((a_1\ b_2 + a_2\ b_1)/2)\,, \\ \log(d_1) + \log(d_3) &= \log((a_1\ b_3 + a_3\ b_1)/2)\,, \\ \log(d_2) + \log(d_3) &= \log((a_2\ b_3 + a_3\ b_2)/2)\,. \end{aligned}$$

The above set of equations always has a unique solution $[d_1\ d_2\ d_3]^T$, so we now have a perfect local mixing rule for $N \leq 3$ that involves only the i_{th} and j_{th} columns of the information matrix.

To extend the above solution to the general case $N > 3$, suppose that the i_{th}, j_{th} columns of L_p^* are $[a_1 \cdots a_N]^T$, $[b_1 \cdots b_N]^T$ respectively, and that $\mathbf{d} = [d_1 \cdots d_N]^T$ is the new merged column of L_q^* for both i and j. Now consider the following equation for merging $\mathbf{a}$ and $\mathbf{b}$ for all $\binom{N}{2}$ pairs of (m, n) combinations.

$$\log(d_m) + \log(d_n) = \log((a_m\ b_n + b_m\ a_n)/2)\,.$$

The above equation can be rewritten as a matrix equation as follows.

$$\mathbf{P} \cdot \beta = \gamma\,, \qquad (3)$$

where

- $\beta = [\log(d_1) \cdots \log(d_N)]^T$, is a $N \times 1$ vector
- $\gamma = [\cdots \log((a_m \cdot b_n + b_m \cdot a_n)/2) \cdots]^T$, is a $\binom{N}{2} \times 1$ vector
- P is a $\binom{N}{2} \times N$ matrix where each row has two ones at the m_{th} and n_{th} positions respectively, and zeros elsewhere.

In general, the system (3) is an overdetermined set of equations and does not have a solution. Therefore, we propose to use a least-square approach to obtain an approximate solution, which will be our mixing rule for the information matrix:

$$\beta = \mathbf{P}^{\dagger}\gamma\,,$$

[6] These set of equations satisfy the normalization constraint on equation (2).

where $P^\dagger = (P^T P)^{-1} P^T$ is the pseudo inverse of P. Thus, the computational complexity of the mixing for information matrix is $O(N^4)$.

Theorem. *The information matrix approach is optimal for $N \leq 3$. For $N \geq 4$, the information matrix approach is sub-optimal due to the approximate mixing (4).*

3.4. Inference Using the Information Matrix

For tracking applications, we are mostly interested in the marginal probabilities p_{ij} of the j_{th} object having the i_{th} identity. To compute marginal probabilities, $N!$ joint probabilities need to be computed first. Computing joint probabilities, however, requires $O(N^3 N!)$ operations, which is not feasible. Therefore, we used the Metropolis sampling algorithm [4] as a heuristic to estimate these marginal probabilities. Simulation results in section 5.1 confirm that the Metropolis algorithm with reasonable number of samples approximates the marginal probabilities well.

4. COMPARISON OF THE BELIEF MATRIX AND INFORMATION MATRIX APPROACHES

In this section, we compare the information matrix with the previous approach based on marginal probabilities in [1] in terms of both computational complexity and distributed implementation.

4.1. Storage and Computation

In terms of representation, both approaches require $O(N^2)$ storage since they use $N \times N$ matrices as their data structures. For mixing operations, the belief matrix approach requires $O(N)$ computation, while the information matrix approach requires $O(N^4)$. For incorporating local evidence, the information matrix approach requires $O(1)$ computation, while the belief matrix approach requires $O(N^2)$ computations in practice. One can see there is a tradeoff between the two approaches in terms of computational complexity – the belief matrix has a simpler mixing operation, while the information matrix has a simpler evidence incorporation operation.

4.2. Distributed Implementation

The aforementioned computational complexities, however, do not represent the realistic costs of these operation in actual implementation due to the additional communication cost incurred by the belief matrix approach. Let us first briefly describe how these two approaches can be implemented in a purely distributed fashion in a WSN and what the assumptions are.

A sensor network we consider has the following characteristics. Sensor nodes are stationary and know their own geographic locations. Each node can exchange messages only with its neighbors, which is a set of nodes one wireless hop away from itself. Objects appear only at the boundary of the network. When nodes at the boundary of the sensor network detect an object, a node in the vicinity is selected and spawns a software agent whose job is to track and accumulate information about that object. As the object moves within the WSN, the agent hops from node to node so as to stay close to the object.

Under this setting, each column of these matrices can be maintained by a single agent as shown. Since mixing under both approaches updates only two columns of the belief and information matrices, only the two agents involved in the mixing need to talk to each other to update their columns. This is just local communication, since mixing happens only when two objects are very close. To update information given local evidence, however, the agent with local evidence needs to send the normalization message to other relevant agents using a group management protocol [2] for the belief matrix approach. For the information matrix approach, the agent only needs to add the log-likelihood $(\log(.9) - \log(\frac{.1}{N-1}))$ to the proper element of the column.

5. EXPERIMENTAL RESULTS

In this section, we present experimental results from simulation and a real tracking system with an RFID system.

5.1. Simulation

Figures 3 and 4 summarize simulation results where we compare three different approaches – using the marginal belief matrix, the information matrix with exact inference and the information matrix with approximate inference. Each representation is used for the tracking and identity management of objects as mixing and local evidence incorporation events take place. In each of our simulations, we process fifty events where the ratio of the number of mixing to evidence incorporation events is fixed. We then record the difference between the inferred marginal distribution of identities to objects from our approximation method to the true marginal probability distribution summed out over the true joint $(N!)$-size distribution. Figure 3 shows simulations for a system that managed the identities of three objects and identities. The x-axis represents the ratio of mixing to local incorporation events, and each data point corresponds to the average difference of the true marginal probabilities to the inferred marginal probabilities over one hundred random simulations. Figure 4 shows the results when our system managed six identities and objects.

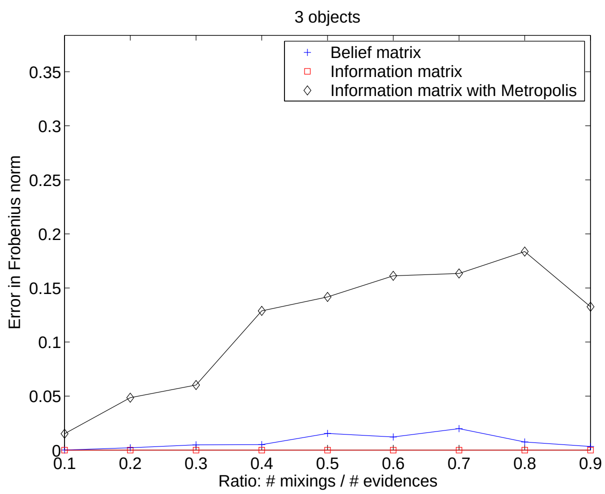

Fig. 3. Comparison of the three approaches: Marginal belief matrix, Information matrix with exact inference and Information matrix with approximate inference

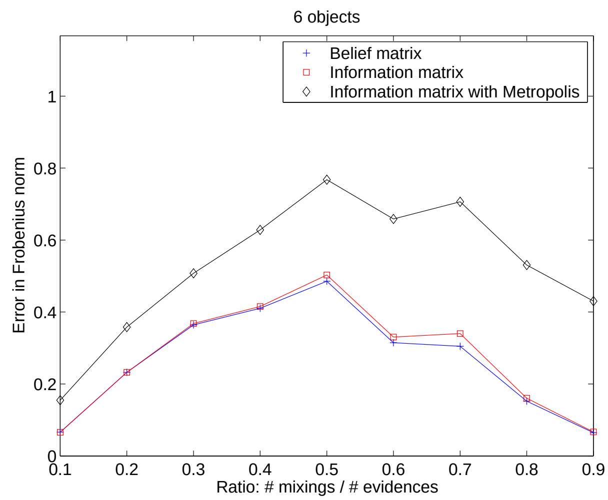

Fig. 4. Comparison of the three approaches: 6 objects

Observing the results of our simulations, the information matrix with exact inference and the belief matrix perform comparably, meaning their relative errors are approximately equal when compared to the true marginal probabilities. The information matrix using the Metropolis sampling algorithm performs worse, but only marginally. Both of these observations are consistent when we increase the number of identities and objects tracked. However, it is important to note that the information matrix approach with approximate inference performs much better when the ratio of the number of mixing to local incorporation events is small. This is true and consistent with the well known result in the Markov Chain Monte Carlo community [4] that a sharply peaked distribution can be accurately represented using $N \log N$ samples for a probability distribution over S_N. Thus, when the number of evidence incorporation events to mixing events is high, our joint distribution over S_N is likely to be sharply peaked, which explains why our sampling method performs better with a low ratio of mixing to evidence incorporation events.

5.2. Experimental Setup: People Tracking System

Two SICK laser range finders are mounted in the Stanford AI Lab. The laser range finders return range measurements over a 180 degree field of view, which provides measurements for estimating positions of moving objects. Furthermore, we have augmented our people tracking system with a radio frequency identification (RFID) system. The RFID system has eight readers that detect the presence of unique tags for identity information. The readers, when activated, send out radio messages to detect the presence of tags within its radio range. Four readers are mounted in the hallways to detect all identities entering and exiting the lab. The remaining four readers are mounted in the lab each activated by motion sensors. Figure 5 is a map of the lab area annotated with where the laser range finders and RFID readers are.

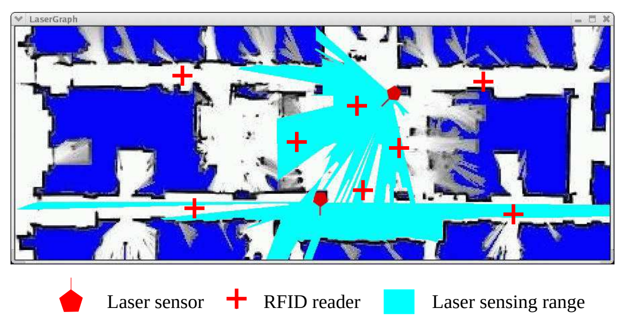

Fig. 5. Experimental setup in the Stanford AI lab

5.3. Experiments Performed

In our experiments we had three individuals starting from different hallways connecting to the lab, walking into the lab, interacting with one another, and leaving the lab. Each person was carrying a RFID tag, which was used to trigger a nearby RFID reader. Figure 6 shows three ground-truth tracks.

Figure 7 shows results after applying the identity management to the data from the scenario in Figure 6. The two graphs on the left show how the uncertainties of track identities measured as a statistical entropy evolve through the mixing and local evidence incorporation events. The x axis,

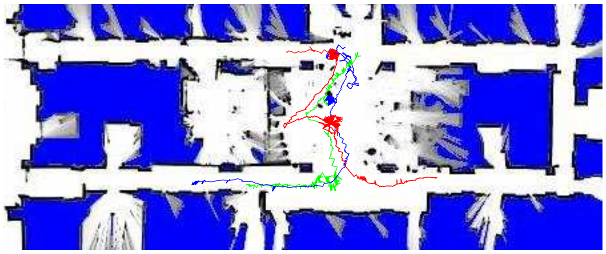

Fig. 6. Experiment scenario: Three people walking in the lab over 87.79 second period. Their individual tracks are shown in different colors.

y axis and z axis represent tracks, events and uncertainty in track identities, respectively. The *events* axis has a sequence of events [Initial M E E E M M E], where M stands for mixing and E stands for evidence. The two matrices on the right are the marginal probabilities of the two approaches after all the events, where the i_{th}, j_{th} entry of each matrix represents the probability that object j has identity i. As the results show, we observe that the information matrix estimates more accurately the true probabilities than the belief matrix. This is expected since as we proved earlier, the information matrix exactly represents the marginals for $N \leq 3$.

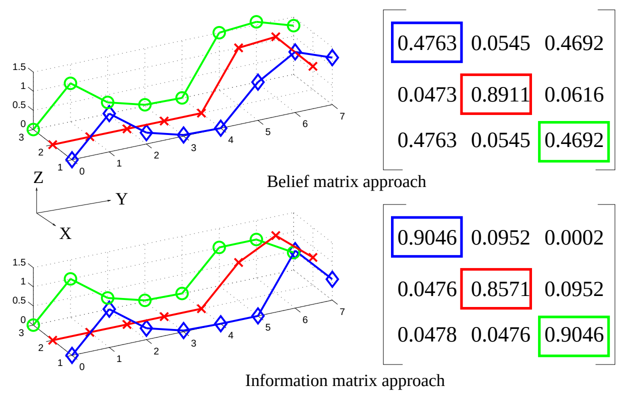

Fig. 7. Uncertainties of object identities after many mixing and evidence incorporation events from the data in Figure 6

6. DISCUSSION AND CONCLUSION

From the analysis and experiments, we come to the conclusion that there is a trade-off when choosing between the two proposed methods for approximating an exponentially-sized distribution through mixing and local evidence incorporation events. In the WSN setting, the weakness of the belief matrix representation is apparent in its need to continuously re-normalize its marginal distribution after each local evidence incorporation event. Furthermore, since this operation requires extensive communication throughout the WSN, this representation is energy-consuming. In this paper, we propose an alternate method for approximating the joint identity distribution using the information matrix. The main advantage of the information matrix representation over the belief matrix is how it handles local evidence incorporation – whenever sensor nodes record evidence they process the evidence locally without communicating throughout the network. Thus, nodes are lazy and perform communication only when users of the system request information rather than whenever sensor nodes process local evidence events, as is the case with the belief matrix representation.

Compared to the belief matrix, the information matrix performs equally well in terms of approximating the true marginal distributions summed over the joint distribution over possible permutation of objects and identities, but at an increased cost of performing the inference necessary to retrieve the approximate marginal probabilities. However, we overcome this added computational cost by performing approximate inference using the Metropolis sampling method. Thus, inference costs using the information matrix representation are significantly reduced. As experiments show, the difference in accuracy between approximate and exact inference for the information matrix is small. Thus, the information matrix representation is the preferable candidate for object tracking and identity management in the WSN setting.

7. REFERENCES

[1] J. Shin, L. Guibas, and F. Zhao, "A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks," in *Proceedings of 2nd International Workshop on Information Processing in Sensor Networks (IPSN 2003)*, April, 2003, pp. 223–238.

[2] Jaewon Shin, Anthony Mancho So, and Leonidas Guibas, "Supporting group communication for interacting agents in wireless ad-hoc sensor network applications," in *IEEE Wireless Communications and Networking Conferences (WCNC 2005)*, 2005.

[3] D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun, "Towards lazy data association in SLAM," in *Proceedings of the 11th International Symposium of Robotics Research (ISRR'03)*, Sienna, Italy, 2003, Springer.

[4] Persi Diaconis and Laurent Saloff-Coste, "What do we know about the Metropolis algorithm?," in *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, Las Vegas, Nevada, United States, 1995, pp. 112–129, ACM Press.