Nonparametric techniques for graphical model-based target tracking in collaborative sensor groups

Viji Paul P.  
Central Research laboratory,  
Bharat Electronics Limited, Bangalore, India.  
vijipaulp@bel.co.in

V. Rajbabu  
Dept. of Electrical Engineering,  
Indian Institute of Technology Bombay, India.  
rajbabu@ee.iitb.ac.in

Abstract—Target tracking using collaborative sensor groups is an effective mechanism for reducing the scalability issues in distributed sensor networks. Using graphical models for such a sensor group together with appropriate class of nonparametric message passing algorithms, we explore efficient approaches to handle the related data fusion problems characterized by spatially distributed observations. Messages consisting of multiple Gaussian components have been efficiently handled with the help of nonparametric belief propagation techniques. The advantage of such an approach in a myopic radar network has been verified here using Monte Carlo simulations by comparing the tracking performance obtained with centralized and distributed fusion schemes.

Keywords- target tracking; graphical model

I. INTRODUCTION

A fundamental problem in sensor network design is the tradeoff between performance and scalability. Traditional centralized schemes require all measurements to be available for fusion at central processor. While this is provably optimal in estimation performance, it exhibits poor scalability. For attaining performance and scalability, sensors can be organized to collaborative groups, each responsible for tracking a single target. The process of forming such groups in target tracking application has been described in [1, 2, and 3]. The aim of this paper is to obtain the best information from the measurements provided by such group of homogenous sensors, with graphical models and nonparametric techniques. Graphical models can be used to represent the statistical structure of the system and measurement process through the common language of graphs, where each node represents a set of random variables and the edges between nodes represent the Markov properties of the random variables belonging to the nodes. The information structure of the multi-sensor problem is determined by the relations among the state variables and measurements constituting the problem. Graphical models provide an efficient way for representing the information structure revealing the interactions of locally collected measurements in each sensor within the overall picture. By using graphical models for the multi-sensor tracking problem along with a class of message passing algorithms (MPAs), we achieve a means to handle the required data fusion problems characterized by spatially distributed observations obtained in sensor network.

The message passing structure, known as belief propagation (BP), based on sum-product algorithms suitable for tree structured graphical models and its nonparametric extension have been discussed in this paper in a multi-sensor tracking context. For Gaussian target states, messages can be reduced to the transmission of means and covariances. However, for non-Gaussian continuous random variables, as is common in target tracking applications, one must transmit a representation of the entire message function. The process of message transmission can be achieved with less computational complexity using nonparametric belief propagation (NBP) technique, which is a significant generalization of particle filtering methods [11]. The estimation process using NBP has the advantage of incorporating all the available measurements from the network. The discrete posterior pdf required for NBP based techniques will be available at each sensor node with the help of particle filter (PF) based estimation techniques. By sending the posterior density as messages across the nodes graphical model-based target tracking technique avoids the requirement of sending the raw measurements to the fusion centre and the consequent prohibitive communication requirements. The tracking performance in a single target scenario has been compared in this paper using various sampling schemes employed in NBP to obtain efficient representation of message products. The advantage of graphical modeling and NBP based message passing technique in a networked radar scenario has been verified using Monte Carlo simulations.

II. GRAPHICAL MODEL BASED DISTRIBUTED TRACKING

When sensors have a limited power budget and myopic (only observe objects that are closer to their own location), it is typical to perform target state estimation at some sensor which is closer to the target. The sensor in charge of the data fusion will be referred in this paper as leader node. At each time $t$, the leader node and potentially few other sensors collect their observations and the leader will use these measurements to update the current estimate of target state [12]. This scenario is depicted in Fig. 1.

Since the object is moving, the most appropriate leader is also a function of time. Therefore, at each time the leader uses the state estimate obtained from the target-state posterior distribution to select a new leader node for time $t+1$. The old leader may select itself as the new leader; or else communicate the current model of the posterior distribution to the new leader. There are number of possible protocols for selecting when and to which sensor the leader should transfer control [1, 2, and 3]. Issues related to transfer of control in heterogeneous sensor networks known as routing, are described in [4]. The transition of observation across nodes is shown in Fig.1 (a), where the
shaded circles represent the lead node at each instant of time. Appropriate message passing algorithms across nodes using PF framework can be applied to Fig. 1 (b) to obtain a combined inference across the network. The simulation carried out in this paper assumes that at each time, the sensor nearest to the target of interest acts as the leader.

Figure 1 (a). Transfer of leadership between nodes in an arbitrary sensor network. (b). Lead node and the information collection centers forms a tree structure. Measurements from sensor $s$ are represented as $z_i$.

A. Graphical models

A sensor network can be assumed to contain a collection of random variables $x_s$ associated with a sensor $s$, called node and the common variables among the neighboring sensors denoted as an edge $E$ connecting the nodes. The resulting structure can be represented as a Markov random field (MRF), based on an undirected graphical model consisting of a set of nodes or vertices $V$ and set of edges denoted as $E$. Using this graphical model structure it is possible to create efficient methods for performing inference over the collection of random variables. If $S$ and $T$ are two nodes in a graph $G$, $(s,t) \in E$ means $s$ and $t$ are adjacent. Those nodes adjacent to $t$ are called its neighbors, denoted as $\Gamma_t \subseteq V$.

For each vertex $s \in V$, let $x_s$ be the associated random variable and the concatenated variables across the nodes yield the random vector $x = \{x_s \mid s \in V\}$. The graphical model formalism requires that the random vector $x$ satisfy certain Markov properties associated with the graph $G$. For any subset $S \subset V$, define $x_S = \{x_s \mid s \in S\}$. Removing node $s_2$ from the graph shown in Fig. 1(b) we obtain $\{x_1\}$ and $\{s_3, s_4, s_5\}$ as conditionally independent sets. In this case the Hammersley-Clifford theorem [5, 7] guarantees that the distribution $p(x)$ can be factorized as the product of functions defined on the cliques (i.e., fully connected subsets of the vertex set $V$) of the graph. The node potential function $\psi_i(x_i)$ corresponding to each node in Fig. 1(b) is an implicit function of the observation $z_i$. With this notation, a MRF consists of a collection of random variables $x$ and the distribution $p(x)$ factorizes as

$$p(x) = \frac{1}{Z} \prod_{C \in C} \psi_C(x_C),$$

where $C$ contains the set of all cliques of $G$, and $k$ denotes normalization constant.

III. INFERENCE FROM GRAPHICAL MODEL

In a tree structured graph as shown in Fig. 1(b) an arbitrary node $s$ can be denoted as root and its neighbors as children and the nodes with no children are called as leaf nodes. Taking advantage of this partial ordering of variables defined by the tree structure, target states can be inferred by a set of iterative message-passing operations. In particular each node $t$ computes the posterior density as a message $m_t$ to send to its neighbor $s$. Message $m_t$ is a function of the state variable $x_t$, defined on the domain of random variable $x_s$ along with the messages received by the other neighbors of $t$, $\Gamma_t \setminus s$. Where $\Gamma_t \setminus s$ denotes neighborhood of $t$ not including $s$.

The posterior marginal distribution of the variables at each node can be computed efficiently on a tree using sum-product message passing algorithm [15]. Beginning with the leaf nodes, compute the messages to parents as shown in Fig. 2(a) (an upward sweep through the tree), with each node computing the outgoing message to its parent only after it has received messages from all its children. Then, beginning with the root node, compute the messages to children in a similar downward sweep through the tree as depicted in Fig. 2(b). Parent or root node will obtain information from many such branches having children and a summation process as shown in Fig. 2(c), to consolidate the information [8, 15].

Consider the process of computing the posterior marginal distribution for state variables $x_i$ corresponding to each node $s$ shown in Fig 1(b). A similar graphical model with upward message passing structure is shown in Fig. 2(a). Let $s_1$ be the root node in this graph with node potential $p(x_1) = p(x_1, z_1)$, where $x_1$ is the state variable and $z_1$ is the measurement. The pair-wise potentials are conditionals like $p(x_1, x_2) = p(x_2 \mid x_1)$, and the remaining node potentials are observation likelihoods, e.g., $p(z_2 \mid x_2)$. Applying Bayes’ rule, the posterior marginal of $x_i$ is $p(x_i \mid \{z_i\}_{i=1,\ldots,5}) = p(x_i, \{z_i\}) / p(\{z_i\})$. The denominator $p(\{z_i\})$ is a constant. Using the law of total probability and the factorization implied by the graph structure,

$$p(x_1, \{z_i\}) = \int \int \int \int p(x_1, z_1) p(x_2 \mid x_1) p(z_2 \mid x_2).$$
\[ p(x_3 \mid x_2)p(x_4 \mid x_2)p(x_5 \mid x_2) \]
\[ \times p(x_5 \mid z_5)p(x_4 \mid z_5)p(z_3 \mid x_3)dx_2dx_3dx_4dx_5. \]  
(2)

By distributive law (2) can be rewritten as,
\[ p(x_1 \mid E) \propto p(x_1 | z_1) p(x_2 | x_1) p(z_2 | x_2) \]
\[ \times \left( p(x_3 \mid x_2)p(z_3 \mid x_3)dx_2 \right) p(x_4 \mid x_2)p(z_4 \mid x_4)dx_4 \]
\[ \times \left( p(x_5 \mid x_2)p(z_5 \mid x_5)dx_5 \right) G_2. \]

Using the sum-product messaging,
\[ p(x_1 \mid E) \propto p(x_1, z_1) x \]
\[ \times \left( p(x_2 \mid x_1)p(z_2 \mid x_2) \right) m_{21} \left( x_2 \right) \]
This can be written in a compact form as,
\[ p(x_1 \mid E) \propto p(x_1, z_1) \]
\[ \times \left( p(x_2 \mid x_1)p(z_2 \mid x_2) \right) m_{21} \left( x_2 \right) \]
(3)

The upward messages shown in the graphical model depicted in Fig. 2(a) are likelihood functions and the downward messages shown in Fig. 2(b) are conditionals. Thus the messages \( m_n(x_n) \) provide sufficient statistics for each marginal computation in the graph. In general the outgoing message from \( v \) to \( s \) is computed using the incoming message from \( v \)'s other neighbors as
\[ m_{vs}@x_s = \int \psi_{vs}(x_v, x_s, \psi_s(x_v)) \prod_{i \in \Gamma(v \setminus s)} m_{vi}(x_v) dx_v. \]
(4)

Here \( \psi_{vs} \) is the edge potential between node \( s \) and \( v \). The posterior marginal or belief at node \( s \) can be computed by combining the observations from all the nodes in the graph as
\[ p(x_s \mid z) = \sum_{x_s} m_{ts}(x_s). \]
(5)

A. Nonparametric representation of belief propagation

For graphical models with continuous variables analytical evaluation of the BP update integral (4) is intractable. So, the first step to realize (4) and (5) in a mathematically tractable form is to develop a nonparametric extension of belief propagation algorithm (NBP). Using Monte Carlo methods it is possible to provide a general procedure to recursively update particle based approximations.

NBP allows particle based inference in arbitrary graphs, and approximates complex, continuous sufficient statistics with kernel based density estimates [14]. Let \( \mathcal{N}(x; \mu, \Lambda) \) denote a normalized Gaussian density with mean \( \mu \) and covariance \( \Lambda \). An \( M \) component mixture approximation of \( m_n(x_n) \) using kernel based density estimation techniques [12] takes the form
\[ m_{ts}(x_s) = \sum_{i=1}^{M} w_{s}^{(i)} \mathcal{N}(x_s; \mu_{s}^{(i)}, \Lambda_{s}). \]
(6)

where \( w_{s}^{(i)} \) is the weight associated with the \( i \)th kernel mean \( \mu_{s}^{(i)} \), and \( \Lambda_{s} \) is a bandwidth or smoothing parameter. The weights are normalized so that \( \sum_{i=1}^{M} w_{s}^{(i)} = 1 \). Efficient multi-scale sampling methods then allow these nonparametric BP algorithms to flexibly adapt to many different applications [12, 14].

IV. DISTRIBUTED TRACKING WITH PARTICLE FILTER

In a collaborative sensor group each sensor will have its own measurement set. The goal here is to find the optimal estimate of target states, without actually sharing the raw observations. Using graphical modeling of the sensor group and with nonparametric approaches a global inference on the target state can be made. Each sensor acting as a node in the graph has to compute the target state estimate using its own measurement set prior to obtaining information from other nodes. The estimation process using particle filter (PF) has the ability to handle nonlinearities involved in the system [6]. Apart from that, PF framework provides a stochastic method of approximating the BP update equation in the forward pass at vertex \( v \) as \( v_{t-1} \rightarrow v_{t} \rightarrow v_{t+1} \), where index \( t \) represents time instants. The goal of particle filtering is thus to estimate posterior marginal distributions \( p(x_i \mid z_s) \). In particular these distributions may not have any closed, parametric form. Hence, uncertainty at each node \( v_i \) is represented nonparametrically by a collection of weighted particles. The basic idea behind particle filtering is to approximate the posterior distributions sequentially.

As the first step to obtain a discrete posterior probability density function (pdf), create a set of weighted samples or particles that represent the distribution \( p(x_i \mid z_s) \). Using the system dynamics this particle set can be time propagated to obtain the prior. Once the measurement set \( z_i \) are received at time \( t \), the conditional likelihood of \( i \)th particles \( x_{t,i} \), are evaluated by evaluating the pdf \( p(z_i \mid x_{t,i}) \), using the pdf of the measurement noise and the measurement prediction as [10],
\[ q_i = \frac{1}{(2\pi)^{m/2} R^{m/2}} \sum_{j=1}^{m} \exp \left( -\frac{1}{2R} \left[ z^{\top} - h(x_{t,j}) \right] \left[ z^{\top} - h(x_{t,j}) \right] \right). \]
(7)

Where \( q_i \) is the likelihood (importance weights) for the \( i \)th particle with \( m \) measurements, \( R \) is the measurement variance and \( h(x_{t,j}) \) is the measurement prediction. The likelihood obtained in (7) has been computed with the assumption that measurements are from a normally distributed cloud around target states. Moreover from the \( m \) measurements there is only one measurement corresponding to the target, and the remaining measurements either correspond to other targets or false alarms. Using the prior and the likelihood value obtained from (7) construct a new set of weighted samples which represents the distribution \( p(x_1 \mid z_2, z_i) \). Iterate this procedure to estimate each of the desired posterior marginal distributions in turn.

V. CONVENTIONAL FUSION SCHEMES

The estimates in PF based approaches can be obtained as a weighted sum of particles and the covariance can be obtained from the particle distribution. In track-to-track fusion schemes [11] state estimates and covariances are transmitted to the central tracking processor under the
general Gaussian assumption. Let $\hat{\Sigma}_f$ be the state covariance for track $T_f$, then the fused covariance $\hat{\Sigma}_f$ and the fused state estimate $\hat{x}_f$ for $n$ tracks can be obtained with the assumption of independent estimation errors as [11, 12],

$$\hat{\Sigma}_f = \left( \sum_{i=1}^{n} \hat{\Sigma}_i \right)^{-1}, \hat{x}_f = \hat{\Sigma}_f \left( \sum_{i=1}^{n} \hat{\Sigma}_i \right)^{-1} \hat{x}_i.$$  \hspace{1cm} (8)

In centralized measurement fusion scheme measurements from all the sensors has to be transmitted to the central tracking processor. All the $m$ measurements can be utilized in the PF frame work as given in (7) to find the measurement likelihood. Though this approach provides better estimation results, the scalability and robustness weaken as more and more sensors are included in the network.

VI. BELIEF PROPAGATION USING PARTICLES

The scalability issues in the conventional fusion scheme can be circumvented with NBP techniques implemented on a graphical model framework. For distributed tracking example, consider representing the true posterior distribution using N samples. At each step, particle filter sends all these N samples exactly, under no communication constraints, to the next leader node. At each time $t$, message representing $p(x_t | z_{1:t}, \ldots, z_{1:t})$ is compressed by the leader node at time $t-1$ and communicated at time $t$. Because it has been compressed, this message is in general not a collection of $N$ particles, but rather some mixture of Gaussian components, and the number of components be proportional to number of targets. The number of components in the compressed mixture will be less than $N$. The leader at time $t$ recreates collection of $N$ particles by drawing independent identically distributed samples from the message, and weighs these samples according to the likelihood information given by the observation $z_t$. The node then samples from this collection $N$ times with replacements to obtain $N$ equally weighted particles, and propagates these particles through the forward dynamics $p(x_{t+1} | x_t)$. This procedure results in a particle representation of $p(x_{t+1} | z_t, \ldots, z_1)$, which is then compressed and communicated to the leader at time $t+2$ in the same manner.

A. Product of Gaussian mixtures

In general, computationally the most difficult part of the NBP algorithm is the procedure for drawing samples from the product of several Gaussian mixtures. Let $L = \{p_1(x), \ldots, p_d(x)\}$ denote a set of $d$ mixtures each of which contains $M$ components. The product of $d$ Gaussian densities is itself a Gaussian, with mean and covariance given by (12) [11, 14],

$$\prod_{i=1}^{d} \mathcal{N}(x; \mu_i, \Sigma_i) \approx \mathcal{N}(x; \mu, \Sigma),$$  \hspace{1cm} (9)

$$\Sigma = \sum_{i=1}^{d} \Sigma_i^{-1}, \ \ \ \mu = \sum_{i=1}^{d} \Sigma_i^{-1} \mu_i.$$  \hspace{1cm} (10)

When each input mixture consist of $M$ components, the Gaussian product mixture will consist of $M^d$ components. To obtain the product mixture $L$, draw one of the $M$ component from $\mathcal{N}(x; \mu_i, \Sigma_i)$ and the weight $w_{i,j}$ associated with this $m^{\text{th}}$ component is given by,

$$w_{i,j} = \prod_{l=1}^{M} w_{i,l} \mathcal{N}(x; \mu_l, \Sigma_l), \ m = 1, \ldots, M,$$  \hspace{1cm} (11)

where $\{w_{i,j}\}$ are the weights associated with the input Gaussians. The message updates performed exactly by repeated use of (9) and (11) requires $M^d$ operations. The goal of NBP algorithm is to draw $N$ samples from the $M^d$ component mixture density $p(x) \approx \prod_{i=1}^{d} p_i(x)$ efficiently [12, 13 and 14]. The following section describes importance sampling scheme for selecting the $N$ components.

B. Importance Sampling

Importance sampling is a Monte Carlo method for approximately sampling from an intractable distribution $p(x)$ using a proposal distribution $q(x)$. To draw $N$ samples from $p(x)$, an importance sampler draws $kN \geq N$ samples $x_j \sim q(x)$, and assigns a weight $w_j$ to the $j^{\text{th}}$ particle, such that, $w_j \propto \frac{p(x_j)}{q(x_j)}$. The weights are normalized by their sum, $Z = \sum_j w_j$, and $N$ samples are drawn with replacement from the discrete distribution $p(x) \approx \frac{w_j}{Z}$, meaning that the value $x_j$ will be drawn multiple times. There are two possible proposal distributions discussed here. The first one referred is mixture importance sampling. In this approach samples are drawn from proposal distribution $q(x) = \frac{1}{d} \sum_j p_i(x)$ and the importance weight for each sample is given by, $w = \prod_{i} \frac{p_i(x)}{\sum_j p_i(x)}$. \hspace{1cm} (12)

In the second method each input mixture $p_i(x)$ is approximated by a Gaussian density $q_i(x)$, and the samples are drawn from $q(x) \approx \prod q_i(x)$.

VII. SIMULATION RESULTS

To verify the advantage of graphical model-based target tracking, the scenario shown in Fig.3 has been considered in our simulations. The scenario consists of 20 myopic radars with a single moving target. The target is assumed to be moving with constant velocity in the search area of the radar network. Each radar considered as a node in the network is having a circular field of vision of 10 km. Each node obtains measurements in range and bearing with a Gaussian noise of 10m variance in the position. The field of vision of sensor-10 has been marked as a circle in Fig. 3.
The target is assumed to have velocity in X and Y direction as $V_x = -70$ m/s and $V_y = -50$ m/s. At each time instant target passes through overlapping area of many sensors.

Tracking at each sensor has been carried out with particle filter consisting of 100 particles. To compare the error performance, let $(x_i^t, y_i^t, \hat{x}_i^t, \hat{y}_i^t)$ denotes the true and estimated target states at time $t$ for the $i^{th}$ Monte Carlo (MC) run. Let M denote the total number of independent MC runs. The RMS position error at $t$ can be computed as

$$RMS_t = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (x_i^t - \hat{x}_i^t)^2 + (y_i^t - \hat{y}_i^t)^2}.$$  

The performance obtained using mixture important sampling and Gaussian important sampling is compared to track-to-track fusion scheme using NBP techniques. The advantage of using graphical model-based tracking architecture with NBP based message passing structure has been demonstrated with a simulated multi sensor scenario.

**ACKNOWLEDGMENT**

Viji Paul would like to express sincere thanks to Ajit Kalghatgi, Chief Scientist, CRL, for the guidance and support to carry out this work.

**REFERENCES**


