Bayesian Approach for Data Fusion in Sensor Networks

J. K. Wu, Y.F. Wong
Institute for Infocomm Research, Singapore, jiankang@i2r.a-star.edu.sg

Abstract - We formulate the target tracking based on received signal strength in the sensor networks using Bayesian network representation. Data fusion among the same type of sensors in an active sensor neighborhood is referred to as cross-sensor fusion, conceptualized as “cooperative fusion”. This data fusion is embedded in the likelihood function derivation. Fusion of signals collected by multiple types of sensors are referred to as cross-modality fusion. It is “complementary”, and represented by the contribution of their likelihood functions to the state update. The tracking algorithm is implemented using particle filter. Very good experimental results are obtained using SensIt data.

Keywords: sensor networks, data fusion, Bayesian networks

Tracking multiple targets has attracted a lot of research interest, and it remains a challenge [1]. Tracking of targets is usually modeled as a dynamic system, by using Bayesian network representation. The state of the targets is then estimated using Maximum A Posterior (MAP). In most cases, the measurements and state distributions are non Gaussian, and the dynamic system is not linear. In these cases, the dynamic equation of Bayesian network representation is not tractable. Importance sampling, especially, particle filter, has been studied, and used in target tracking with radar and other signals based on time delay of arrival [2], as well as by means of video cameras [3]. By using either radar or video camera, a rough location of the target can be obtained. Bayesian filter here provides a solid foundation for robust tracking.

We are here dealing with tracking of targets in sensor networks, where each sensor node is equipped with acoustic and seismic sensors. This is different from visual tracking and tracking using radar signals in the following:

First, tracking with acoustic and seismic signals is based on the signal strength the sensors receive. The targets are considered as the sources of the signal. The problem is formulated as that of recovering source location with known signal strength at a few points. It is an ill-posed problem. With the signal propagation model by using maximum likelihood estimation, Hu and his students were able to locate the source, and then obtain the target location [4]. Because of the ill-posed nature of the problem, they have to do exhaustive search in a search region. Similar work is in the area of heat transfer research. People try to recover the heat source and derive heat flux by placing thermocouples in some points. The Markov Chain Monte Carlo (MCMC) algorithm was used to obtain estimates of the statistics of the unknown heat flux [5]. By using MCMC sampling strategy they were able to extend the Bayesian inference approach to inverse problems, which are characterized by high-dimension, non-standard and/or complex distribution functions. These two works dealt with topics which are related to, but different from target tracking.

Second, here in this article, the target tracking happens in the context of sensor networks. As targets move in, out and around, sensors which can receive signals change over time. The tracking is a collaborative effort of all those sensors which receive signals. In other words, collaborative tracking is a feature for tracking in sensor networks.

In summary, we will formulate the target tracking using received signal strength in the context of sensor networks. Particle filtering is used to implement the tracking algorithm. In the rest of this article, Section 1 defines the formulism for target tracking in sensor networks, while implementation of inverse source localization tracking using particle filter is described in Section 2. Section 3 presents experimental results.

1. Bayesian network representation

Networked sensors are deployed in a certain area, collecting signals emitted from the targets. We choose random field and graph model to model sensor network data:

- Sensor network can be represented by a graph $G=(V, E)$, with $V$ denoting vertex (or node) set and $E$ the edge set. Where $v$ denotes a sensor node at a spatial location, with its coordinates in the geographic space.
- Let $z_v={z_v^1, \ldots, z_v^M}$ be random value measurements indexed by vertex / node $v$. $z_v^j$ represents the measurement for the $j$th sensor mounted on the sensor node $v$ at its location.
- The edge between two nodes represents the relationships and dependences between the two sensor nodes, and the data vectors they collected.

1.1 Neighborhood system and collaboration Region

All sensor networks are meant to accomplish certain tasks. A group of sensors collaboratively working
together to fulfill a task is one of the major characteristics of sensor networks. A task is location-variant or dynamic when it is to track moving targets. We define a neighborhood system as follows:

- Nodes are related to one another via a neighborhood system. A neighborhood consists of one or a group of sensor nodes, performing a common task or forming a homogeneous region.
- A neighborhood system is defined as:

\[ S = \{ S_i \mid \forall i \in V \} \]

Where \( S_i \) is the set of nodes neighboring node \( i \), and node \( i \) is referred to as the site of a neighborhood \( S_i \).
- A node is not neighboring to itself, and neighboring relationship is mutual

\[ i \notin S_i \quad \text{and} \quad i \in S_i \Leftrightarrow i' \in S_i \]
- A neighborhood can be dynamic. The site of the neighborhood can change as the task location changes. In this case, nodes in the neighborhood change accordingly.

There are two types of sensor network architectures for tracking. One is hierarchical, consisting of lead nodes and sensing nodes. Each lead node has its territory being monitored by a group of sensing nodes. Sensing nodes collect signals and report to their lead nodes. The lead nodes perform data fusion, keep record of the targets, and communicate with the neighboring lead nodes. When a target enters into the territory, the group of sensing nodes and their lead node become active and perform tracking task until the target leaves the territory. The lead node then passes the tracking results to the next active lead node.

The other is flat, and shown in Figure 1. A neighborhood of sensor nodes is dynamically formed when a target presents. All sensors receiving signals with strength above a threshold constitute an active neighborhood. The one with strongest signal plays the leading role. We choose to use this dynamic neighborhood system in this article.

### 1.2 Dynamic system model

We model the target tracking in sensor networks as a dynamic system, and represent it by Bayesian network in Figure 2. In the figure, \( x_k^i \) and \( Z_k^i \) denote the state and measurements at time \( k \) of the neighborhood at site \( i \), respectively. It is important to note that the measurement here is a composite one for the whole neighborhood, produced by all types of sensors mounted on all those nodes in the neighborhood.

\[
\begin{align*}
  k-1 & \quad k & \quad k+1 & \quad \text{Time} \\
  Z_{k-1} & \quad \vdots & \quad Z_k & \quad Z_{k+1} & \quad \text{Measurements} \\
  p(x_k^i \mid x_{k-1}^i) & \quad p(Z_k^i \mid x_k^i) & \quad p(x_{k+1}^i \mid x_k^i) & \quad p(Z_{k+1}^i \mid x_{k+1}^i) & \quad \text{States} \\
  x_k^i & \quad \Rightarrow & \quad x_{k+1}^i & \quad \Rightarrow & \quad x_{k+1}^i
\end{align*}
\]

Figure 2. Bayesian network representation of a spatiotemporal dynamic system

We assume that the state sequence is a 1st order Markov process, and we have the state transition equation:

\[
x_k^i = f_k^i(x_{k-1}^i, u_{k-1}^i, n_{x_k}^i) \quad (1)
\]

We also assume that the measurement only depends on the state. The measurement equation then is:

\[
Z_k^i = h_k(x_k^i, u_k^i, n_{z_k}^i) \quad (2)
\]

where \( u \) is known input, \( n_y \) is state noise and \( n_z \) is measurement noise. In order to evaluate the state in the model to solve the dynamic system problem, we usually evaluate the maximum a posterior (MAP) distribution for a given sequence of measurement. This will take the following two steps: In the prediction step, the state transition model is used to generate the estimate of the state at the next time interval. Then in the update step, the likelihood calculated based on measurement model is used to update the state generated at prediction step, so as to reach the state estimation for time \( k \).

Prediction:

\[
p(x_k^i \mid Z_{k-1}^i) = \int p(x_k^i \mid x_{k-1}^i)p(x_{k-1}^i \mid Z_{k-1}^i)dx_{k-1}^i \quad (3)
\]

Update:

\[
p(x_k^i \mid Z_k^i) = \frac{p(Z_k^i \mid x_k^i)p(x_k^i \mid Z_{k-1}^i)}{p(Z_k^i \mid Z_{k-1}^i)} \quad (4)
\]

Principally, using above model and equation, for a given initial state and measurement sequences, we can evaluate states at any time instance \( k \). Unfortunately, for most cases, there is little information on these probability distribution functions, and then equations (3) and (4) are not tractable.

### 1.3 Cross-sensor and cross-modality data fusion

![Collaborative Neighborhood: Time K
Task: Tracking Car 2](image)

![Collaborative Neighborhood: Time K+1
Task: Tracking car 1](image)
As we have defined, the measurement at time \( k \), \( Z^k \), in above equations is a composite measurement for the whole neighborhood. There will be cross-sensor and cross-modality data fusion to derive the composite measurement and its corresponding likelihood function.

The idea of Monte Carlo simulation is to draw an identical independently distributed set of samples from a target PDF. These \( L \) samples can be used to approximate the target density with the empirical point-mass function. The simulation starts with a weighted set of samples, \( \{x_{k,i}, w_{k,i}\}_{i=1}^L \), which are approximately distributed according to \( p(x_{k,i} | z_{k,i}) \). New samples are then generated according to a suitably chosen proposal distribution, which may depend on the old state and the new measurements

\[
x_k^i \rightarrow q(x_k^i | x_{k-1}^i, z_k^i), \quad i = 1, \ldots, L
\]

The expectation of the state can be then approximated by those samples:

\[
E(x_{0k}) = \frac{1}{L} \sum_{i=1}^L w_{0k}^i x_{0k}^i
\]

Re-sampling is necessary to prevent from de-generation.

2. Dynamic Model

In the target tracking problem, for simplicity, the state of a target is taken to be its location. Therefore, the state transition model can be simply represented as

\[
x_k = x_{k-1} + \text{disp}_{k-1} + n_x
\]

where \( \text{disp}_{k-1} \) is the displacement of the target at time \( k-1 \). \( \text{disp}_{k-1} \) will be simply estimated by the actual displacement at last time instance. The noise term, \( n_x \), is taken to be the same for all time instances. It is simulated by using a Gaussian random noise.

Particle filters have evolved into many different varieties over the past few years [6]. The key issue is the choice of proposal distribution, which can best approximate the target posterior distribution. Here we choose Sampling Importance Resampling (SIR) for its simplicity and effectiveness. With SIR, the proposal distribution and the update formula are reduced to the following forms:

\[
q(x_k^i | x_{k-1}^i, z_k^i) \rightarrow p(x_k^i | x_{k-1}^i)
\]

\[
w_k^i = p(z_k^i | x_k^i), \quad \sum w_k^i = 1
\]

With this proposal distribution, the samples are drawn around the predicted location.

2.3 Measurement model and likelihood

Let us look into the acoustic sensors. When the sound propagates in the free and homogenous space, the acoustic energy decay function can be modeled by the following equation:

\[
z_n^{aco} = g_a \frac{S}{\| y - r_n \|^2} + n_z = g_a q_n^{aco} S + n_z
\]

Where \( z_n^{aco} \) is the acoustic signal received by the \( n \)th sensor, \( n_z \) is the noise term that summarizes the net
effects of background additive noise and the parameter modeling error. \( g_n \) and \( r_n \) are the gain factor and location of the \( n^{th} \) sensor, respectively. \( S \) and \( y \) are respectively, the energy emitted by the source and its location. Here we only consider one target, and we assume that sensor nodes are dense enough so that there is only one target enters into the neighborhood. \( d_{aco} \) represent the distance factor for acoustic. In this case, it is \( 1/d^2 \). The acoustic measurement in the neighborhood can be represented by a vector as follow.

\[
z_{aco}^{\text{aco}} = (z_1^{aco}, z_2^{aco}, \ldots, z_N^{aco})^T = GDS + nz \quad (13)
\]

where we omit subscript “aco” with the understanding that this measurement equation for the neighborhood with \( Ns \) number of acoustic sensors and the acoustic source at location \( y \).

Assume that the noise term in the measurement function is Gaussian white noise with standard deviation \( \sigma \), the likelihood function of having measurement \( z \) at state \( x \) is

\[
p(z^{\text{aco}} | x^i) = \frac{1}{(2\pi\sigma)^{d/2}} \exp(-\frac{(GDS - z^{\text{aco}})^T (GDS - z^{\text{aco}})}{2\sigma^2}) \quad (14)
\]

In the case of seismic, the attenuation model is \( z_{sei}^{\text{sei}} = S^{\text{sei}} e^{-\kappa d} \), where \( d \) is the distance, and \( \kappa \) is attenuation factor, being 0.0013 – 0.003 in [8]. The likelihood of seismic can be computed in the similar way as in equation (14). Let \( e = (GDS - z)^T (GDS - z) \), the weight update function for fusion of multiple modalities of measurements can be written as:

\[
w_k^j = p(z^{aco} | x_k^i) p(z^{sei} | x_k^i) = \exp(-\frac{c^{aco} e^{aco} + c^{sei} e^{sei}}{2\sigma^2}) \quad (15)
\]

where we have omitted the constant in front of exponential function because there will be a normalization process for weights. The coefficients \( c^{aco} \) and \( c^{sei} \) reflect the quality of measurements and the contribution of the measurements to the overall likelihood.

### 2.4 Particle filter implementation

We summarize the particle filter implementation of Bayesian filter for target tracking in sensor networks using acoustic and seismic sensors, for Cross-Sensor and Cross Modality (CSCM) data fusion as follows:

**CSCM Algorithm**

1. **Initialize** \( x^i, i = 1, 2, \ldots, L \) (number of samples)
   
   **Prediction**
   
   FOR each particle \( i = 1: L \),
   
   Draw \( x_k^i = x_k^{i-1} + disp_{k-1} \)
2. Determine active neighborhood by threshold of received sensor signals to generate vector \( z^{aco} and z^{sei} \).
3. **Weight update.**
   
   \[ w_k^j = \exp(-\frac{c^{aco} e^{aco} + c^{sei} e^{sei}}{2\sigma^2}) \]

   and normalization.

4. Compute expectation of the target state
   
   \[ E(x_k) = \frac{1}{L} \sum_{i=1}^{L} w_k^i x_k^i \]
5. **Resample** — drop those samples with too small weights, and split those samples with largest weights so that the number of samples remain to be \( L \).

### 3. Experimental results

![Figure 2(a)](image)

Tracking using acoustic sensor only. The mean square tracking error is 4.7.

![Figure 2(b)](image)

Tracking using seismic sensor only. The mean square tracking error is 2.44.

![Figure 2(c)](image)

Tracking by fusion of seismic and acoustic
signal. The mean square tracking error is 1.397.

After simulations by using generated data using measurement model in equation (12), we have applied our algorithm to the SensIt dataset [4]. Among all data, we selected 6 collections so that there are enough number of sensor nodes, and the quality of the data are reasonable. For all 6 data collections, three runs of experiments were conducted. Figure 3 shows the experimental results for data collection of “aav6”. In the figure, “*” denotes sensor nodes. The true track of vehicle is plotted using black dots, while the red dots indicate the estimated track. The blue symbols are samples used at the last step in the tracking.

The acoustic data collected by sensors indicate that there are large differences of gain of sensors: sensors located farther may output larger signal than those located near. In the experiments, we did not spend much effort to estimate the sensor gain. As such, we can observe from the Figure 3(a) that the tracking behaves well for those locations where there are sensors on the both sides of the track. On contrary, there are obvious tracking errors for those locations where there are sensors on only one side of the track.

The seismic signals from all sensors are comparatively uniform. The tracking results are quite satisfactory, as shown in Figure 3(b), with mean square error of 2.44.

By fusion of seismic and acoustic signal, we were able to obtain much better tracking results as shown in Figure 3(c). The mean square error is 1.397, which is much smaller than tracking using seismic only. Here in this experiments, we take $c_{seg} = 0.1$, and $c_{aco} = 1$, because of the poor quality of acoustic data.

4. Conclusion

We have proposed a cross-sensor and cross-modality data fusion framework and algorithm for target tracking in sensor networks, based on Bayesian network representation and particle filter implementation. Good results were obtained using SensIt data set.

Our further work will be on the detailed theoretical foundation of the framework and algorithm. More comprehensive experiments on multiple targets and nonlinear dynamic model will be further studied.

REFERENCES

2. Z. Khan, T. Balch, F. Dellaert, “MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets”, IEEE Trans. PAMI, 27(11), 2005, pp. 1805-1819