

Nonparametric Boxed Belief Propagation for Localization in Wireless Sensor Networks

Vladimir Savic, Santiago Zazo
The Signal Processing Applications Group
Polytechnic University of Madrid, Spain
{vladimir, santiago}@gaps.ssr.upm.es

Abstract

Obtaining estimates of each sensor's position as well as accurately representing the uncertainty of that estimate is a critical step for effective application of wireless sensor networks (WSN). Nonparametric Belief Propagation (NBP) is a popular localization method which uses particle based approximation of belief propagation. In this paper, we present a new variant of NBP method which we call Nonparametric Boxed Belief Propagation (NBBP). The main idea is to constraint the area from which the samples are drawn by building a box that covers the region where anchors' radio ranges overlap. These boxes, which are created almost without any additional communication between nodes, are also used to filter erroneous samples of the beliefs. In order to decrease the computational and the communication cost, we also added incremental approach using Kullback-Leibler (KL) divergence as a convergence parameter. Simulation results show that accuracy, computational and communication cost are significantly improved.

1. Introduction

The localization consists in obtaining the relative or absolute position of a sensor node together with the uncertainty of its estimate. Equipping every sensor with a GPS receiver or equivalent technology may be expensive, energy prohibitive and limited to outdoor applications. Therefore, we consider the problem in which some small number of sensors, called *anchor nodes*, obtain their coordinates via GPS or by installing them at points with known coordinates, and the rest, *unknown nodes*, must determine their own coordinates. If unknown nodes were capable of high-power transmission, they would be able to make measurements with all anchor nodes (*single-hop* technique). However, we would like to use energy-

conserving devices without power amplifier, with lack the energy necessary for long-range communication. In this case, each sensor has available noisy measurements to several neighboring sensors (*multi-hop* technique). Typical measurements techniques [1] are time-of-arrival (TOA), receive-signal-strength (RSS), angle-of-arrival (AOA) or connectivity.

There are a large number of previously proposed localization methods. *Range-free* or *connectivity-based* localization algorithms [2, 3, 4] rely on connectivity between the nodes. The principle of this algorithm is to determine whether or not a sensor is in the transmission range of another sensor. The most attractive feature of the range-free algorithms is their simplicity. However, they can only provide a coarse grained estimate of each node's location, which means that they are only suitable for applications requiring an approximate location estimate. *Range-based* or *distance-based* localization algorithms [5, 6] use the inter-sensor distance measurements in a sensor network to locate the entire network. This type of algorithms is usually more accurate, but also sensitive to measurement errors. Based on the approach of processing the individual inter-sensor data, localization algorithms can be also considered in two main classes: *centralized* and *distributed* algorithms. Centralized algorithms [3, 5] utilize a single central processor to collect all the individual inter-sensor data and produce a map of the entire sensor network, while distributed algorithms [2, 4, 6] rely on self-localization of each node in the sensor network using the local information it collects from its neighbors. From the perspective of location estimation accuracy, centralized algorithms are likely to provide more accurate location estimates than distributed algorithms. However, centralized algorithms suffer from the scalability problem and generally are not feasible to be implemented for large scale sensor networks.

A recent direction of research in distributed sensor network localization is the use of particle filters [7, 8]. In [9], Ihler *et al.* formulated the sensor network localization problem as an inference problem on a graphical model and applied one variant of belief propagation (BP) methods [10], the so-called nonparametric belief propagation (NBP) algorithm, to obtain an approximate solution to the sensor locations. Comparing with deterministic approaches [2-6], the main advantages of this statistical approach are its easy implementation in a distributed fashion and sufficiency of a small number of iterations to converge. Furthermore it is capable of providing information about location estimation uncertainties and accommodating non-Gaussian distance measurement errors. In this paper, we present a new variant of this method.

The remainder of this paper is organized as follows. In Section 2, we review Ihler's graphical model for the localization problem. In Section 3, we propose nonparametric boxed belief propagation method (NBBP). Simulation results are presented in Section 4. Finally, Section 5 provides some conclusions and future work perspective.

2. Graphical model

We begin by describing a graphical model [9] for the sensor network localization problem. We suppose that all sensors with unknown positions obtain noisy distance measurements of nearby subset of the other sensors in the network. Typically, this measurement procedure can be accomplished using a broadcast transmission (acoustic or wireless) from each sensor as all other sensors listen.

Let us assume that we have N_s sensors (N_a anchors and N_u unknowns) scattered randomly in a planar region, and denote the two-dimensional location of sensor t by x_t . The unknown node t obtains a noisy measurement d_{tu} of its distance from node u with some probability $P_d(x_t, x_u)$:

$$d_{tu} = \|x_t - x_u\| + v_{tu}, \quad v_{tu} \sim p_v(x_t, x_u) \quad (1)$$

The binary variable o_{tu} will indicate whether this observation is available ($o_{tu} = 1$) or not ($o_{tu} = 0$). Each sensor t has some prior distribution denoted $p_t(x_t)$. This prior could be an uninformative one for the unknowns and the delta impulse for the anchors. Then, the joint distribution is given by:

$$\begin{aligned} p(x_1, \dots, x_{N_u} | \{o_{tu}\}, \{d_{tu}\}) &= \\ &= \prod_{(t,u)} p(o_{tu} | x_t, x_u) \prod_{(t,u)} p(d_{tu} | x_t, x_u) \prod_t p_t(x_t) \end{aligned} \quad (2)$$

The final goal of this localization problem is to estimate the maximum a posteriori (MAP) sensor location x_t given a set of observations $\{d_{tu}\}$. There are two different ways to do this, to estimate MAP of each x_t , or to estimate MAP of all x_t jointly. We select the first one because this is the only way for our incremental approach where unknowns are located one by one at different time points. Ihler also choose the same way in order to minimize the bit-error rate [9]. The measured distances d_{tu} and d_{ut} may be different, and it's even possible to have $o_{tu} \neq o_{ut}$, indicating that only one of the sensors u and t can observe the other. However, we can assume that both sensors obtain the same single observation, so $d_{tu} = d_{ut}$ and $o_{tu} = o_{ut}$, otherwise there is no observed distance.

For large-scale sensor networks, it is reasonable to assume that only a subset of pairwise distances will be available, primarily between sensors which are located within the some radius R . We use improved model [9] which assumes that the probability of detecting nearby sensors falls off exponentially with squared distance:

$$P_d(x_t, x_u) = \exp\left(-\frac{1}{2} \|x_t - x_u\|^2 / R^2\right) \quad (3)$$

We also need to exchange information between the nodes which are not directly connected. Let's define a pair of nodes s and t to be a 1-step neighbors of one another if they observe their pairwise distance d_{st} . Then, we define 2-step neighbors of node s to be all nodes t such that we do not observe the d_{st} , but do observe d_{su} and d_{ut} for some node u . We can follow the same pattern for the 3-step neighbors, and so forth. These n -step neighbors ($n > 1$) contain some information about the distance between them. Therefore, if two nodes do not observe the distance between them, they should be far away from each other. In our case, we will include all 1-step and 2-step neighbors, others could be neglected without losing accuracy of the results.

The relationship between the graph of the nodes and joint distribution may be quantified in terms of potential functions ψ [9, 10] which are defined over each of the graph's cliques:

$$p(x_1, \dots, x_{N_u}) \propto \prod_{\text{cliques } C} \psi_C(\{x_i : i \in C\}) \quad (4)$$

This only requires potential functions defined over variables associated with single nodes and pairs of nodes. Single-node potential at each node t , and the pairwise potential between nodes t and u , are respectively given by:

$$\psi_t(x_t) = p_t(x_t), \quad (5)$$

$$\psi_{tu}(x_t, x_u) = \begin{cases} P_d(x_t, x_u) p_v(d_{tu} - \|x_t - x_u\|), & \text{if } o_{tu} = 1, \\ 1 - P_d(x_t, x_u), & \text{otherwise.} \end{cases} \quad (6)$$

Finally, the joint posterior distribution for each node t is given by:

$$p(x_1, \dots, x_{N_u} | \{o_{tu}, d_{tu}\}) \propto \prod_t \psi_t(x_t) \prod_{t,u} \psi_{tu}(x_t, x_u) \quad (7)$$

Having defined a graphical model for sensor localization, we can now estimate the sensor locations by applying the belief propagation (BP) algorithm. The form of BP as an iterative, local message passing algorithm makes this procedure trivial to distribute among the wireless sensor nodes. We apply BP to estimate each sensor's posterior marginal, and use the mean value of this marginal and its associated uncertainty to characterize sensor positions.

Each node t computes its belief $M_t^i(x_t)$ (the posterior marginal distribution of two-dimensional position $x_t(x_t^a, x_t^b)$ at iteration i) by taking a product of its local potential ψ_t with the messages from its set of neighbors G_t :

$$M_t^i(x_t) \propto \psi_t(x_t) \prod_{u \in G_t} m_{ut}^i(x_t) \quad (8)$$

The messages m_{ut} , from node u to node t , are computed by:

$$m_{ut}^i(x_t) \propto \int \psi_{ut}(x_t, x_u) \frac{M_u^{i-1}(x_u)}{m_{tu}^{i-1}(x_u)} dx_u \quad (9)$$

In the first iteration of this algorithm it's necessary to initialize $m_{ut}^1 = 1$ and $M_t^1 = p_t$ for all u, t , then repeat computation using (8) and (9) until sufficiently converge. In fact, the number of iteration should be at least the length of the longest path in the graph.

3. Nonparametric Boxed Belief Propagation (NBBP)

The presence of nonlinear relationships and potentially highly non-Gaussian uncertainties in sensor localization makes described BP algorithm

unacceptable. Besides, to obtain acceptable spatial resolution for the sensors, the discrete state space must be made too large for BP to be computationally feasible. However, using particle-based representations via nonparametric belief propagation (NBP) enables the application of BP to inference in sensor networks. The belief and message update equations, (8) and (9), are performed using stochastic approximations, in two stages: first, drawing samples from the belief $M_t^i(x_t)$, then using these samples to approximate each outgoing message m_{tu}^i .

In this section we propose nonparametric boxed belief propagation (NBBP) which includes three modifications:

- 1) Constrain the area from which the samples are drawn by building a box that covers the region where anchors' radio ranges overlap (Figure 1).
- 2) In each iteration, filter erroneous samples of the beliefs (all the samples which are out of the appropriate box).
- 3) Locate nodes in the incremental way: As soon as the belief sufficiently converges, characterize sensor positions with mean value and uncertainty, and from that point consider this node as an anchor.

3.1. Computing messages

Given N weighted samples $\{W_t^{j,i}, X_t^{j,i}\}$ from the belief $M_t^i(x_t)$ obtained at iteration i , we can compute a Gaussian mixture estimate of the outgoing BP message m_{tu}^i . We first consider the case of observed edges (1-step) between unknown nodes. The distance measurement d_{tu} provides information about how far sensor u is from sensor t , but no information about its relative direction. To draw a sample of the message $(X_{tu}^{j,i+1})$, given the sample X_t^j which represents the position of sensor t , we simply select a direction $\theta^{j,i}$ at random ($i=1$), uniformly in the interval $[0, 2\pi)$. However, starting from the second iteration, we include information from the previous iteration using already computed beliefs. Therefore, for $i > 1$, a direction is calculated by $\theta^{j,i} = \arctan(X_u^{j,i} - X_t^{j,i})$. We then shift X_t^j in the direction of $\theta^{j,i}$ by an amount which represents the estimated distance between nodes u and t ($d_{tu} + v^j$):

$$x_{tu}^{j,i+1} = X_t^{j,i} + (d_{tu} + v^j)[\sin(\theta^{j,i}) \cos(\theta^{j,i})], \quad (10)$$

The samples are then weighted by the reminder of (9):

$$w_{tu}^{j,i+1} = P_d(X_t^{j,i}, x_{tu}) \frac{W_t^{j,i}}{m_{tu}^i(X_t^{j,i})} \quad (11)$$

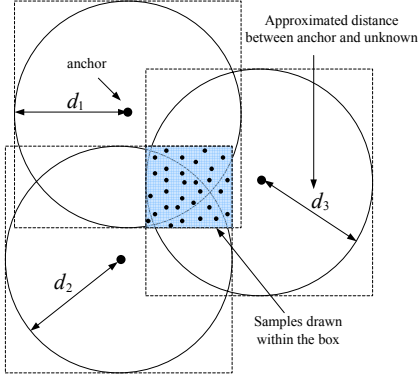


Figure 1. Building the box and sampling

The optimal value for bandwidth h_{tu}^{i+1} could be obtained in a number of possible techniques. The simplest way is to apply the “rule of thumb” estimate [11]:

$$h_{tu}^{i+1} = N^{-\frac{1}{3}} \text{Var}(\{x_{tu}^{i+1}\}) \quad (12)$$

We have also to define messages from anchor nodes, using (9) and the belief of the anchor node x_t^* ($M_t^i(x_t) = \delta(x_t - x_t^*)$):

$$m_{tu}^{i+1}(x_u) \propto \psi_{tu}(x_t^*, x_u) \quad (13)$$

Messages along unobserved edges (2-step,...) must be represented as analytic functions since their potentials have the form $1 - P_d(x_t, x_u)$ which is typically not normalizable. Using the probability of detection P_d and samples from the belief M_t^i , an estimate of outgoing message to node u is given by:

$$m_{tu}^{i+1}(x_u) = 1 - \sum_j W_t^{j,i} P_d(X_t^{j,i}, x_u) \quad (14)$$

Finally, the messages along unobserved edges from anchor nodes ($W_t^{j,i} = 1/N$) are given by:

$$m_{tu}^{i+1}(x_u) = 1 - P_d(x_t^*, x_u) \quad (15)$$

3.2. Computing beliefs

In the initialization phase for each unknown, we construct the box (Figure 1) using only 1-step and 2-step edges from the anchor nodes. 1-step edges are already measured with some noise v_{tu} , and 2-step edges are approximated with the worst case scenario: sum of the two measured distances. Using this distances (d_{tu}) and the positions (x_t^*) of 1-step and 2-

step anchors, the bounds of the box for node $x_u(x_u^a, x_u^b)$ are given by:

$$\begin{aligned} x_{u,\min}^a &= \max_{t=1}^{N_a^1} (x_t^{a*} - d_{tu}), & x_{u,\max}^a &= \min_{t=1}^{N_a^1} (x_t^{a*} + d_{tu}), \\ x_{u,\min}^b &= \max_{t=1}^{N_a^1} (x_t^{b*} - d_{tu}), & x_{u,\max}^b &= \min_{t=1}^{N_a^1} (x_t^{b*} + d_{tu}) \end{aligned} \quad (16)$$

where N_a^1 is the number of 1-step and 2-step edges between this unknown and anchors. Of course, these bounds are limited to the bounds of the deployment area. To initialize belief $M_u^1(x_u)$, we draw N samples with the uniform distribution within this box.

To estimate the belief $M_u^{i+1}(x_u)$ using (8), we draw samples from the product of several Gaussian mixture and analytic messages. In our case it is very difficult to draw samples from this product, so we use *proposal distribution*, sum of the Gaussian mixtures, and then re-weight all samples. This procedure is well-known as *mixture importance sampling* [12].

Denote the set of neighbors of u , having observed edges to u and not including anchors, by G_u^0 , and the set of all neighbors by G_u . In order to draw N samples, we create a collection of kN weighted samples (where $k \geq 1$ is a parameter of the sampling algorithm) by drawing $kN/|G_u^0|$ samples from each message m_{tu} with $t \in G_u^0$ and assigning each sample a weight equal to the ratio:

$$W_u^{j,i+1} = \prod_{v \in G_u} m_{vu}^{i+1} / \sum_{v \in G_u^0} m_{vu}^{i+1} \quad (17)$$

If the sample of the belief $M_u^{i+1}(x_u)$ is out of its box, we filter it by assigning $W_u^{j,i+1} = 0$.

Some of these calculated weights are much larger than the rest, especially after more iterations. This means that any sample-based estimate will be unduly dominated by the influence of a few of the particles, and the estimate could be erroneous. To avoid this, we then draw N values independently from collection $\{W_u^{j,i+1}, X_u^{j,i+1}\}$ with probability proportional to their weight, using *resampling with replacement* [8]. This means that we create N equal-weight samples drawn from the product of all incoming messages.

3.3. Convergence

A node is located when a convergence criteria is met. We use Kullback-Leibler (KL) divergence [13], a common measure of error between two distributions. For the particle based beliefs in our algorithm, KL-divergence between beliefs in two consecutive iterations, is given by:

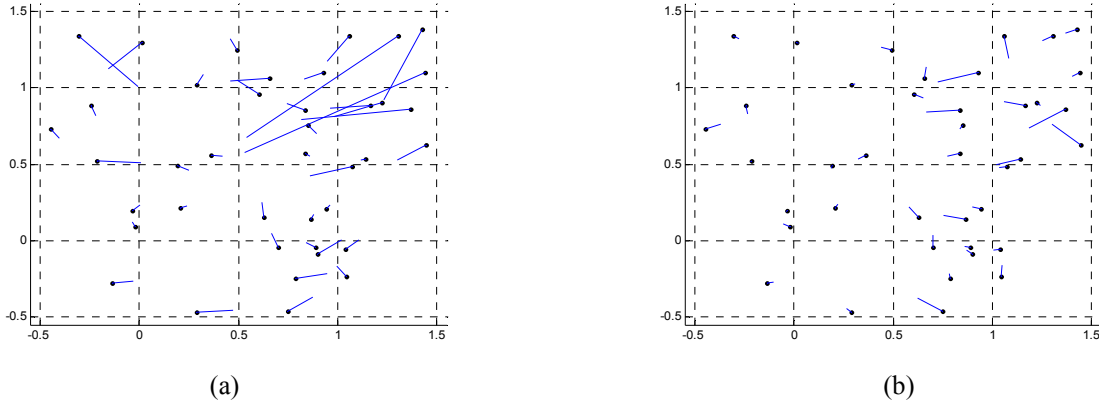


Figure 2. Comparison of the results for a 50-node network (a) NBP, (b) NBBP

$$KL_u^{i+1} = \sum_j W_u^{j,i+1} \log[M_u^{i+1}(X_u^{j,i+1})/M_u^i(X_u^{j,i})] \quad (18)$$

When KL_u^{i+1} drops below the predefined threshold, the node u is located and starts to behave as an anchor. In this way, we can locate all nodes incrementally. The execution is over when KL drops below the threshold for all nodes, or when the maximum number of iteration is reached. In any case, estimated positions of all unknowns and their uncertainties will be available.

4. Simulation results

In the simulation, we placed 50 nodes randomly in $2m \times 2m$ area, 40 of them are unknowns. The values of parameters are set as follow: standard deviation of the Gaussian noise ($\sigma = 0.1m$), transmission radius ($R = 30\%$ of diagonal length of the deployment area - $d_{\max} = 2\sqrt{2}m$), number of particles ($N = 100$), and the KL threshold ($KL_{\min} = 0.02$). We run the simulation for both algorithm (NBP and NBBP), and obtained results shown in Figure 2. Obviously, the location estimates for the NBBP are more accurate since all estimates are placed within its box. In Figure 3 and 4, we show comparison of the average error and coverage (percentage of located nodes) with respect to transmission radius. Finally, in Figure 5 and 6, we show comparison of the computational and communication cost with respect to the number of particles. The main conclusion is that NBBP algorithm performs better than NBP. This result is expected because constructed boxes increase accuracy, and incremental approach decreases computational and communication cost.

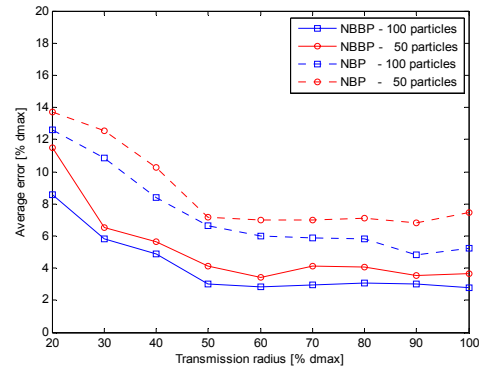


Figure 3. Comparison of accuracy

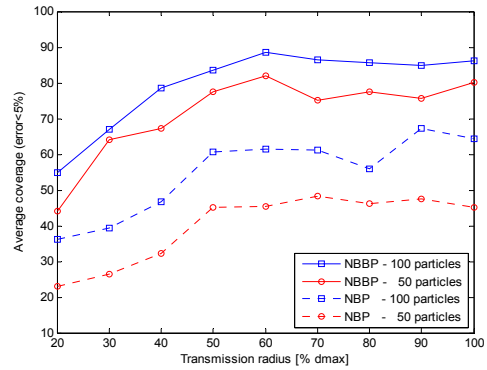


Figure 4. Comparison of coverage (percentage of located nodes with error less than 5%)

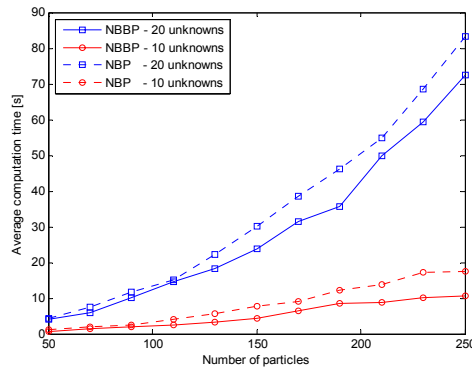


Figure 5. Comparison of computational cost

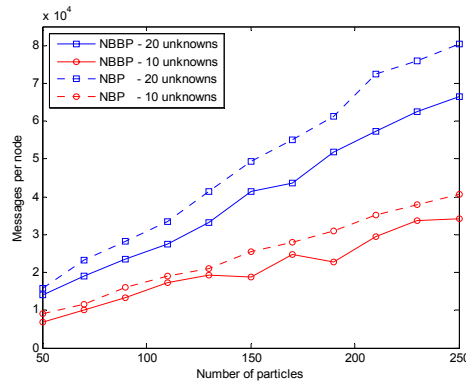


Figure 6. Comparison of communication cost

5. Conclusion and future work

In this paper, we presented nonparametric boxed belief propagation (NBBP), a new variant of NBP. Our main goal was to increase the performance of the algorithm and we achieved it by adding boxes which constraint the area from which the samples are drawn. These boxes, which are created almost without any additional communication between nodes, are also used to filter erroneous samples of the beliefs. We also added incremental approach in order to decrease the computational and the communication cost. There remain few open directions for improving this algorithm. Accuracy could be improved by clustering the nodes and passing messages between the groups using some version of general belief propagation (GBP) [10]. Moreover, communication cost could be decreased using some specific message passing protocol, e.g. grouping similar samples and sending them like one. This will be a part of our future work.

6. Acknowledgment

This work has been performed in the framework of the ICT project ICT-217033 WHERE, which is partly

funded by the European Union and partly by the Spanish Education and Science Ministry under Grant TEC2007-67520-C02-01/02/TCM. Furthermore, we thank partial support by program CONSOLIDER-INGENIO 2010 CSD2008-00010 COMONSENS.

7. References

- [1] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero III, R.L. Moses and N.S. Correal, "Locating the nodes", *IEEE Signal Processing Magazine*, vol. 22, issue 4, pp. 54-69, July 2005.
- [2] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)", in *IEEE GLOBECOM*, vol. 5, pp. 2926-2931, November 2001.
- [3] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from Connectivity in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961-974, November 2004.
- [4] V. Vivekanandan and V.W.S. Wong, "Concentric Anchor Beacon Localization Algorithm for Wireless Sensor Networks", *IEEE Transactions on Vehicular Technology*, vol. 56, issue 5, pp. 2733 - 2744, September 2007.
- [5] A. Savvides, H. Park, and M. B. Srivastava, "The Bits and Flops of the N-hop Multilateration Primitive for Node Localization Problems", in *International Workshop on Sensor Networks Application*, pp. 112-121, September 2002.
- [6] N.B. Priyantha, H. Balakrishnan, E. Demaine and S. Teller, "Anchor-Free Distributed Localization in Sensor Networks", MIT Laboratory for Computer Science, Tech Report, April 2003.
- [7] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", *IEEE Transactions on Signal Processing*, vol. 50, issue 2, pp. 174-188, February 2002.
- [8] P.M. Djuric, J.H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M.F. Bugallo, J. Miguez, "Particle Filtering", *IEEE Signal Processing Magazine*, vol. 20, issue 5, pp. 19-38, September 2003.
- [9] A. T. Ihler, J. W. Fisher III, R. L. Moses, and A. S. Willsky, "Nonparametric Belief Propagation for Self-Localization of Sensor Networks", *IEEE Journal On Selected Areas In Communications*, vol. 23, issue 4, pp. 809-819, April 2005.
- [10] J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations", *Exploring artificial intelligence in the new millennium*, ACM, pp. 239-269, 2003.
- [11] B.W. Silverman, "Density Estimation for Statistics and Data Analysis", Chapman and Hall, New York, 1986.
- [12] D. MacKay, "Introduction to monte carlo methods", In M. I. Jordan, editor, *Learning in Graphical Models*. MIT Press, 1999.
- [13] A. T. Ihler, J. W. Fisher, III, and A. S. Willsky, "Communication-Constrained Inference", Technical Report TR-2601, Laboratory for Information and Decision Systems, MIT, 2004.