SENSOR LOCALIZATION USING GENERALIZED BELIEF PROPAGATION IN NETWORK WITH LOOPS

Vladimir Savic, Santiago Zazo

Dpto. de Señales, Sistemas y Radiocomunicaciones-ETSIT, Universidad Politécnica de Madrid
Ciudad Universitaria S/N, 28040 Madrid, España
phone: + (34) 915495700 (Ext. 4006), fax: + (34) 913367350, email: vladimir@gaps.ssr.upm.es

ABSTRACT

Belief propagation (BP), also called "sum-product algorithm", is one of the best-known graphical model for inference in statistical physics, artificial intelligence, computer vision, etc. Furthermore, a recent research in distributed sensor network localization showed us that BP is an efficient way to obtain sensor location as well as appropriate uncertainty. However, BP convergence is not guaranteed in a network with loops. In this paper, we propose localization using generalized belief propagation based on junction tree (GBP-JT) method. We illustrate it in a network with loop where BP shows poor performance. In fact, we compared estimated locations with Nonparametric Belief Propagation (NBP) algorithm. According to our simulation results, GBP-JT resolved the problems with loops, but the price for this is unacceptable large computational cost. The main conclusion is that this algorithm could be used with some approximation which keeps improved accuracy and significantly decreases the computational cost.

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 prefer 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 (e.g. distance or angle) only to several neighboring sensors (*multi-hop* technique).

A recent direction of research in distributed sensor network localization is the use of particle filters [1, 2]. In [3], *Ihler et al.* formulated the sensor network localization problem as an inference problem on a graphical model and applied particle based variant of belief propagation (BP) methods [4], the so-called nonparametric belief propagation (NBP) algorithm, to obtain an approximate solution to the sensor locations. Comparing with deterministic algorithms [5, 6, 7], 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. However, it is not guaranteed to converge in network with loops [4, 8]. In this paper, we present a new variant of this method which solves problem with loops.

We propose localization using generalized belief propagation based on junction tree (GBP-JT). Junction tree model is a generalization of belief propagation (BP) that is correct for arbitrary graphs. *Jordan* proved it using *elimination procedure* [9]. Compared with *Ihler's* Nonparametric Belief Propagation (NBP) algorithm, GBP-JT converge well in network with loops, but the price for this is unacceptable large computational cost. Therefore, some suitable approximation will be part of our future work.

The remainder of this paper is organized as follows. In Section 2, we review standard BP and condition for its convergence. In Section 3, we propose GBP-JT algorithm. Simulation results are presented in Section 4. Finally, Section 5 provides some conclusions and future work perspective.

2. CONVERGENCE OF BELIEF PROPAGATION

In the standard BP algorithm, the belief at a node *i* is proportional to the product of the local evidence at that node ($\psi_i(x_i)$), and all the messages coming into node *i*:

$$M_i(x_i) = k\psi_i(x_i) \prod_{j \in N(i)} m_{ji}(x_i) \tag{1}$$

where x_i is a state of node *i*, *k* is a normalization constant and *N(i)* denotes the neighbors of node *i*. The messages are determined by the message update rules:

$$m_{ji}(x_i) = \sum_{x_j} \psi_j(x_j)\psi_{ij}(x_i, x_j) \prod_{k \in N(j)\backslash i} m_{kj}(x_j) \tag{2}$$

where $\psi_{ij}(x_i, x_j)$ is pairwise potential between nodes *i* and *j*. On the right-hand side, there is a product over all messages going into node *j* except for the one coming from node *i*. In practical computation, one starts with nodes at the edge of the graph, and only computes a message when one has available all the messages necessary. It is easy to see [4] that each message needs to be computed only once for single connected graphs. That means that whole computation takes a time proportional to the number of links in the graph, which is dramatically less that the exponentially large time that would be required to compute marginal probabilities naively. In other words, BP is a way of organizing the "global" computation of marginal beliefs in terms of smaller local computations.

The BP algorithm, defined by equations (1) and (2), does not make a reference to the topology of the graph that it is running on. Thus, there is nothing to stop us from implementing it on a graph that has loops. One starts with some initial set of messages, and simply iterates the message-update rules (2) until they eventually converge, and then can read off the approximate beliefs from the belief equations (1). But if we ignore the existence of loops and permit the nodes to continue communicating with each other, messages may circulate indefinitely around these loops, and the process may not converge to a stable equilibrium. One can indeed find examples of graphical models with loops, where, for certain parameter values, the BP algorithm fails to converge or predicts beliefs that are inaccurate. On the other hand, the BP algorithm could be successful in graphs with loops, e.g. error-correcting codes defined on Tanner graphs that have loops [10]. This can be proved using *Bethe* approximation to the "free energy" [4, 8]. The fixed points of the BP algorithm correspond to the stationary points of the Bethe "free energy". To make this more clear, let's define for one graphical model, a joint probability function $p(\{x\})$. If we have some other approximate joint probability function $b(\{x\})$, we can define a "distance" between $p(\{x\})$ and $b(\{x\})$, called Kullback-Leibler (KL) distance, by:

$$D(b(\{x\}) \,||\, p(\{x\})) = \sum_{\{x\}} b(\{x\}) \ln \frac{b(\{x\})}{p(\{x\})} \tag{3}$$

The KL distance is useful because it is always non-negative and is zero if and only if the two probability functions $p(\{x\})$ and $b(\{x\})$ are equal.

Statistical physicists generally assume that Boltzmann's law is true:

$$p(\{x\}) = \frac{1}{Z} e^{-E(\{x\})/T} \tag{4}$$

where Z is normalization constant, and the "temperature" T is just a parameter that defines a scale of units for the "energy" E. For simplicity, we can choose $T = 1$. Using (3) and (4), we find the KL distance:

$$D(b(\{x\}) \,||\, p(\{x\})) = \sum_{\{x\}} b(\{x\}) E(\{x\}) + \sum_{\{x\}} b(\{x\}) \ln b(\{x\}) + \ln Z$$

So we see that this KL distance will be zero when approximate probability function $b(\{x\})$ will equal to the exact probability function $p(\{x\})$.

Finally, the Bethe approximation is the case when joint belief $b(\{x\})$ is function of single-node beliefs $b(x_i)$ and two-node beliefs $b(x_i, x_j)$. Yedidia *et al.* proved [4] that for a single-connected graph, values of these beliefs that minimize the Bethe free energy, will correspond to the exact marginal probabilities. For graph with loops, these beliefs will only be approximations, although a lot of them are quite good.

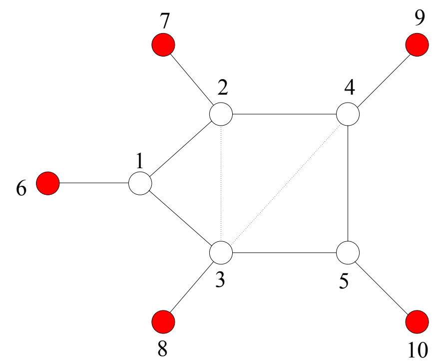


Figure 1 – Example of 10-node network with loop

3. LOCALIZATION USING GENERALIZED BELIEF PROPAGATION

Our goal in this section is to develop new localization algorithm using generalized belief propagation based on *junction tree* method (GBP-JT). Junction tree algorithm is a standard method for exact inference in graphical model [9]. The graph is first *triangulated* (added "virtual" edges so that every loop of length more than 3 has a chord). Given a triangulated graph, with cliques C_i and potentials $\psi_{C_i}(x_{C_i})$, and given corresponding junction tree which defines links between the cliques, we send the following message from clique C_i to clique C_j by the message update rule:

$$m_{ij}(x_{S_{ij}}) = \sum_{C_i \setminus S_{ij}} \psi_{C_i}(x_{C_i}) \prod_{k \in N(i) \setminus j} m_{ki}(x_{S_{ki}}) \tag{5}$$

where $S_{ij} = C_i \cap C_j$, and where *N(i)* are the neighbors of clique C_i in the junction tree. The belief at clique C_i is proportional to the product of the local evidence at that clique and all the messages coming into clique *i*:

$$M_i(x_{C_i}) = k \psi_{C_i}(x_{C_i}) \prod_{j \in N(i)} m_{ji}(x_{S_{ji}}) \tag{6}$$

Beliefs for single nodes can be obtained via further marginalization:

$$M_i(x_i) = \sum_{C_i \setminus i} M_i(x_{C_i}) \quad \text{for } i \in C_i \tag{7}$$

The equations (5), (6), and (7) represent generalized belief propagation algorithm which is valid for arbitrary graphs. The BP algorithm defined with (1) and (2) is a special case of GBP-JT, obtaining by noting that the original tree is already triangulated, and has only pairs of nodes as cliques. In this case, sets S_{ij} are single nodes, and marginalization using (7) is unnecessary.

Let's show how it works in our example in Figure 1. The network has 10 nodes, 5 anchors (nodes 6-10) and 5 unknowns (nodes 1-5). There is a loop 1-2-4-5-3, so we have to triangulate it by adding two more edges (2-3 and 3-4). Then we can define 8 cliques in the graph: $C_1 = \{x_1, x_2, x_3\}$, $C_2 = \{x_2, x_3, x_4\}$, $C_3 = \{x_3, x_4, x_5\}$, $C_4 = \{x_4, x_9\}$, $C_5 = \{x_5, x_{10}\}$, $C_6 = \{x_1, x_6\}$, $C_7 = \{x_2, x_7\}$, $C_8 = \{x_3, x_8\}$.

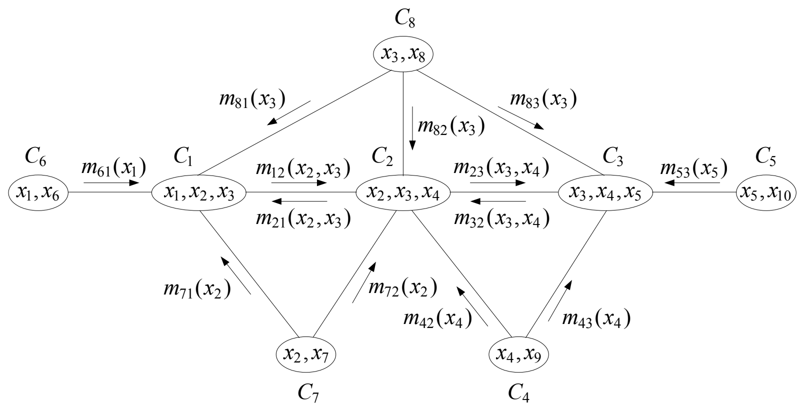


Figure 2 – The junction tree corresponding to the network in Figure 1

The appropriate potentials of 3-node cliques are given by:

$$\psi_{C_1}(x_1, x_2, x_3) = \psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)$$
$$\psi_{C_2}(x_2, x_3, x_4) = \psi_{24}(x_2, x_4)$$
$$\psi_{C_3}(x_3, x_4, x_5) = \psi_{35}(x_3, x_5)\psi_{45}(x_4, x_5)$$

Note that "virtual" edges do not appear in these equations since they are used only to define cliques. Other cliques, defined over pairs of nodes, are nothing else than potential functions between two nodes already known from standard BP:

$$\psi_{C_4}(x_4, x_9) = \psi_{49}(x_4, x_9), \quad \psi_{C_5}(x_5, x_{10}) = \psi_{510}(x_5, x_{10}),$$
$$\psi_{C_6}(x_1, x_6) = \psi_{16}(x_1, x_6), \quad \psi_{C_7}(x_2, x_7) = \psi_{27}(x_2, x_7),$$
$$\psi_{C_8}(x_3, x_8) = \psi_{38}(x_3, x_8)$$

The junction tree corresponding to the network in Figure 1 is shown in Figure 2. As we can see, "anchor cliques" ($C_4 - C_8$) do not receive messages, so this graph does not contain loops. Actually, these "anchor cliques" also include one unknown node so we can send them messages, but this node also could be located marginalizing the belief of some other clique. Besides, it's interesting to notice that there is no direct connection between C_1 and C_3 since C_2 will send all necessary information about node 3 (which belongs to C_1 and C_3).

In the next step, we can compute all messages using equation (5). The complete set of messages is given by:

$$m_{61}(x_1) = \psi_{16}(x_1, x_6^*), \; m_{53}(x_5) = \psi_{510}(x_5, x_{10}^*),$$
$$m_{71}(x_2) = m_{72}(x_2) = \psi_{27}(x_2, x_7^*),$$
$$m_{42}(x_4) = m_{43}(x_4) = \psi_{49}(x_4, x_9^*),$$
$$m_{81}(x_3) = m_{82}(x_3) = m_{83}(x_3) = \psi_{38}(x_3, x_8^*),$$
$$m_{12}(x_2, x_3) =$$
$$\psi_{27}(x_2, x_7^*)\psi_{38}(x_3, x_8^*)\sum_{x_1}\psi_{16}(x_1, x_6^*)\psi_{C_1}(x_1, x_2, x_3)$$
$$m_{32}(x_3, x_4) =$$
$$\psi_{49}(x_4, x_9^*)\psi_{38}(x_3, x_8^*)\sum_{x_5}\psi_{510}(x_5, x_{10}^*)\psi_{C_3}(x_3, x_4, x_5)$$
$$m_{21}(x_2, x_3) =$$
$$\psi_{27}(x_2, x_7^*)\psi_{38}(x_3, x_8^*)\sum_{x_4}\psi_{49}(x_4, x_9^*)\psi_{C_2}(x_2, x_3, x_4)m_{32}(x_3, x_4)$$
$$m_{23}(x_3, x_4) =$$
$$\psi_{49}(x_4, x_9^*)\psi_{38}(x_3, x_8^*)\sum_{x_2}\psi_{27}(x_2, x_7^*)\psi_{C_2}(x_2, x_3, x_4)m_{12}(x_2, x_3)$$

where asterisk denotes the known location of the anchor node and the messages from "anchor cliques" are directly replaced by appropriate potential function. The beliefs of cliques are computed using equation (6), e.g. for C_1 and C_3:

$$M_1(x_1, x_2, x_3) =$$
$$\psi_{C_1}(x_1, x_2, x_3)\psi_{16}(x_1, x_6^*)\psi_{27}(x_2, x_7^*)\psi_{38}(x_3, x_8^*)m_{21}(x_2, x_3)$$
$$M_3(x_3, x_4, x_5) =$$
$$\psi_{C_3}(x_3, x_4, x_5)\psi_{38}(x_3, x_8^*)\psi_{49}(x_4, x_9^*)\psi_{510}(x_5, x_{10}^*)m_{23}(x_3, x_4)$$

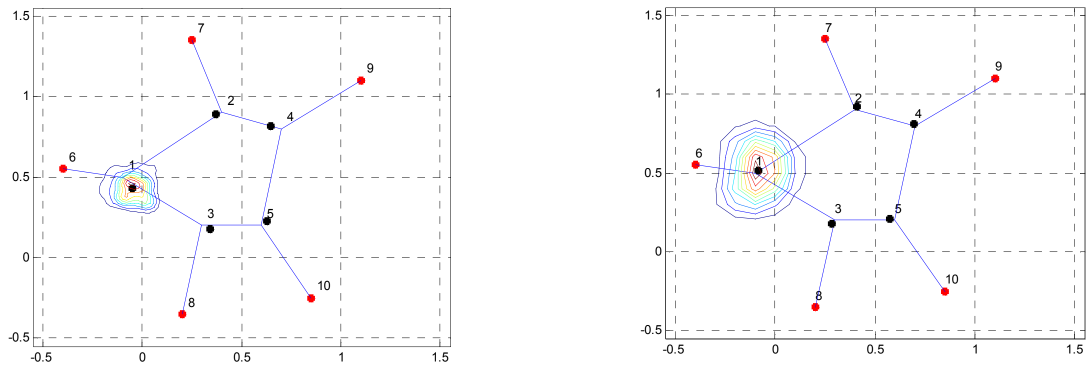
Now it's easy to compute beliefs of single nodes by marginalizing beliefs of cliques using equation (7). Obviously, it's sufficient to know beliefs of C_1 and C_3 since these cliques include all unknown nodes. Marginalization of C_2 provides degree of freedom and could be used to check the estimated positions of some nodes (in our case, for nodes 2, 3 and 4).

Finally, we have to define potential functions for this localization problem. In our case, we assumed that we didn't obtain a priori information about node position, so single-node potentials are equal to 1 (in opposite case, beliefs computed using equation (7) have to be multiplied by their own potentials). The pairwise potential between nodes *t* and *u* is given by [3]:

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

where P_d is the probability of detecting nearby sensors; in our case we used improved model which assumes that the probability of detecting nearby sensors falls off exponentially with squared distance:

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



(a) (b)

Figure 3 – Comparison of the results for a 10-node network (a) NBP, (b) GBP-JT

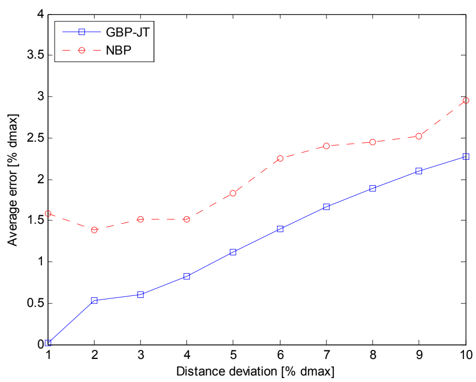


Figure 4 – Comparison of accuracy

where R is the transmission radius. The binary variable o_{tu} indicates whether this observation is available ($o_{tu} = 1$) or not ($o_{tu} = 0$). And the last remaining parameter is measured distance. The unknown node t obtains a noisy measurement d_{tu} of its distance from detected node u:

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

In our case, we used Gaussian distribution for p_v, but, as we can see, it's very easy to change it to any desired distribution (e.g. obtained by running training experiment in the deployment area).

The proposed GBP-JT algorithm is not unique. There are a lot of variations of this method; the best known is *cluster variation method* [8]. However, it can be shown that they are quite similar. For example, in [8] is described the relationship between different region-based approximations. The main goal is achieved in all of them: estimated beliefs are correct in network with loops. We illustrate it in the next section for the network from Figure 1.

4. SIMULATION RESULTS

We simulated the network from Figure 1 using NBP [3] and GBP-JT method. We placed 10 nodes in $2\text{m} \times 2\text{m}$ area, 5 anchors and 5 unknowns. The values of parameters are set as follows: standard deviation of measured distance ($sigma = 0.1$ m), transmission radius ($R = 25\%$ of diagonal length of the deployment area - $d_{\max} = 2\sqrt{2}$ m) and number of points/particles ($N = 400$)[1]. Number of iteration for NBP is set to the length of the longest path in the graph, and for GBP-JT in our example there is obviously just one iteration.

We run the simulation for both algorithm (NBP and GBP-JT), and obtained results shown in Figure 3. Obviously, the location estimates (mean values of beliefs) for the GBP-JT are more accurate since this algorithm is correct for network with loops. NBP algorithm does not converge well for a few nodes, but for some other values of parameters, or with different positions of some nodes, it provides estimates with almost same accuracy as GBP-JT. In any case, the accuracy of GBP-JT is always higher than accuracy of NBP, with respect to the deviation of measured distance (Figure 4).

Nevertheless, the GBP-JT algorithm has two important drawbacks. First, comparing uncertainties for NBP and GBP-JT (contours in Figures 3a and 3b), we can see that NBP provides us better guarantees of its estimate. Second, and the most important, computational time of GBP-JT is significantly larger comparing with NBP (few minutes comparing with few seconds). This is caused by multiplications of high-dimensional messages without any approximation. This is not the case in NBP method since NBP represents particle based approximation of BP. Therefore, the main conclusion is that this algorithm could be used with some approximation which keeps improved accuracy and significantly decreases the computational cost.

[1] This parameter represents number of particles ($N = 400$) in NBP, and number of grid points ($N = 20 \times 20$) in GBP-JT. In order to make these two algorithms comparable, they have to be same.

5. CONCLUSION AND FUTURE WORK

Junction tree model is a generalization of belief propagation (BP) that is correct for arbitrary graphs. It's already have been used for other applications (statistical physics, artificial intelligence, computer vision, etc). In this paper, we presented localization method using generalized belief propagation based on junction tree (GBP-JT). Our main goal was to solve the problem with loops and we achieved it using mentioned approach. Using one example, we demonstrated that accuracy is always better comparing with the well-known version of BP, called nonparametric belief propagation (NBP). However, the computational cost makes this algorithm useful only in high-power sensor network with small number of unknown nodes. Therefore, there remain many open directions for improving this algorithm. Computational cost could be improved using some particle based approximations of GBP-JT. Moreover, uncertainty probably could be decreased using some a priori information. Finally, the most important task is implementation of this algorithm for an ad-hoc network where we expect to outperform standard NBP. This will be part of our future work.

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.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)", in *IEEE GLOBECOM*, vol. 5, pp. 2926–2931, November 2001.

[6] 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.

[7] 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.

[8] J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms", *Information theory, IEEE Transaction On Information Theory*, vol. 51, issue 7, pp. 2282-2312, July 2005.

[9] M.I. Jordan and Y. Weiss, "Graphical model: Probabilistic inference", *The Handbook of Brain Theory and Neural Networks, 2nd edition*. Cambridge, MA: MIT Press, 2002.

[10] B.J. Frey, "A revolution: Belief propagation in graphs with cycles", *Adv. in Neural Information Processing Systems,* vol. 10, MIT Press, 1998.