NONPARAMETRIC BELIEF PROPAGATION FOR SENSOR SELF-CALIBRATION

Alexander T. Ihler, *John W. Fisher III*, *Randolph L. Moses*†, *and Alan S. Willsky*

* Massachusetts Institute of Technology
Cambridge MA 02139

† Ohio State University
Columbus OH 43210

ABSTRACT

Automatic self-calibration of ad-hoc sensor networks is a critical need for their use in military or civilian applications. In general, self-calibration involves the combination of absolute location information (e.g. GPS) with relative calibration information (e.g. estimated distance between sensors) over regions of the network. We formulate the self-calibration problem as a graphical model, enabling application of *nonparametric belief propagation* (NBP), a recent generalization of particle filtering, for both estimating sensor locations and representing location uncertainties. NBP has the advantage that it is easily implemented in a distributed fashion, can represent multi-modal uncertainty, and admits a wide variety of statistical models. This last point is particularly appealing in that it can be used to provide robustness against occasional high-variance (outlier) noise. We illustrate the performance of NBP using Monte Carlo analysis on an example network.

1. INTRODUCTION

Improvements in sensing technology and wireless communication are rapidly increasing the importance of sensor networks for a wide variety of application domains [1, 2]. Collaborative networks are created by deploying a large number of low-cost, self-powered sensor nodes. Sensor localization, i.e. 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 large sensor networks. Manual calibration of each sensor may be impractical or even impossible, while equipping every sensor with a GPS receiver (or equivalent technology) may be cost prohibitive. Consequently, methods of *self-calibration* which can exploit relative information (e.g. estimated distance between sensors) to make the most of a limited amount of global reference information are desirable. In the wireless sensor network context self-calibration is further complicated by the need to minimize inter-sensor communication in order to preserve energy resources.

We describe a self-calibration method making use of noisy distance measurements between neighboring sensors. In the special case that the noise is well modeled by a Gaussian distribution, self-calibration may be formulated as a nonlinear least-squares optimization problem. In [3] it was shown that this optimization could be solved using an iterative, centralized approach.

In contrast, we reformulate the process of self-calibration as an inference problem on a graphical model. This allows us to apply nonparametric belief propagation (NBP, [4]) to obtain an approximate solution. NBP provides several advantages, including a naturally distributed estimation procedure and an inherent estimate of the location uncertainties. However, here we focus primarily on the ability of NBP to incorporate non-Gaussian noise models, which can be used to add robustness to outlier measurements.

2. SELF-CALIBRATION OF SENSOR NETWORKS

We restrict our attention to scenarios in which individual sensors obtain noisy distance measurements of a (usually nearby) subset of the other sensors in the network. This includes, for example, scenarios in which each sensor contains a transceiver and distance is estimated by received signal strength or time delay of arrival between sensor locations. Although this formulation is slightly less general than that presented in [3], it is straightforward to extend our methodology to allow for the inclusion of direction-of-arrival information and/or scenarios in which sources are not co-located with a cooperating sensor.

Specifically, let us take N sensors scattered in a planar region. Denote the two-dimensional location of sensor t by x_t, and let two sensors t and u obtain a noisy measurement d_{tu} of the distance between them with some probability $P_o(x_t, x_u)$:

$$d_{tu} = \|x_t - x_u\| + \nu_{tu} \quad \nu_{tu} \sim p_\nu \tag{2.1}$$

We use the binary random variable o_{tu} to indicate whether this observation is available, i.e. $o_{tu} = 1$ if d_{tu} is observed, and $o_{tu} = 0$ otherwise. Finally, each sensor t has a (potentially uninformative) prior distribution, denoted $p_t(x_t)$.

In general, finding the maximum likelihood (ML) sensor locations x_t given a set of observations $\{d_{tu}\}$ is a complex nonlinear optimization problem. If the uncertainties above are Gaussian (i.e. the distributions $p_\nu = \mathrm{N}(0, \sigma_\nu^2)$, $p_t(x_t) = \mathrm{N}(\hat{x}_t, \sigma_x^2)$) and P_o is assumed constant, ML estimation of the x_t's reduces to a nonlinear least-squares optimization [3].

We further note the distinction between solving for a *relative* sensor geometry versus estimating the sensor locations with respect to some absolute frame of reference. It was shown in [3] that in some cases these two problems can be equivalent, essentially when the influence of prior information $\prod_t p_t(x_t)$ is weak or nonexistent. Given only the relative measurements $\{d_{tu}\}$, the sensor locations x_t may only be solved up to an unknown rotation, translation, and negation (mirror image) of the entire network. We avoid ambiguities in the relative calibration case by assuming known conditions on three sensors' locations (denoted x_1, x_2, x_3):

1. *Translation*: $x_1 = [0; 0]$
2. *Rotation*: $x_2 = [0; a]$ for some $a > 0$
3. *Negation*: $x_3 = [b; c]$ for some b, c with $b > 0$

For absolute calibration, these assumptions are unnecessary if the priors $p_t(x_t)$ are sufficiently informative to resolve the ambiguity.

A number of methods have already been proposed to estimate sensor locations when only a subset of the pairwise distances are measured [5, 6]. For example, one may approximate each unobserved distance by the length of the shortest path along observed distances (other choices are described in [5]) between them,

0-7803-8484-9/04/$20.00 ©2004 IEEE

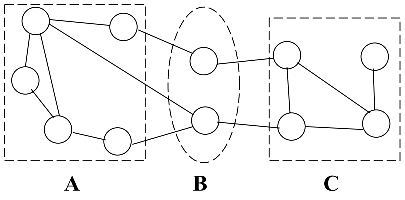

Fig. 1. Graph separation and conditional independence: all paths between the sets A and C pass through B, implying $p(x_A, x_C|x_B) = p(x_A|x_B)p(x_C|x_B)$.

then apply classical multidimensional scaling [7] or other techniques [5]. Alternatively, iterative least-squares methods have also been observed to yield good performance [3, 6]. Yet another possibility is to minimize rank using heuristics while preserving the fidelity of the observed distances [8].

In this paper we reformulate the self-calibration problem as a graphical model, and propose an approximate solution making use of a recent sample-based message-passing estimation technique called *nonparametric belief propagation* (NBP). We conclude with several example simulations demonstrating the ability of NBP to solve difficult distributed localization problems, in particular focusing on its ability to robustly account for the possibility of occasional, large errors (outliers) in the estimated distances.

3. GRAPHICAL MODELS

Graphical models [9] are popular formalisms for encoding known conditional independence relationships between random variables. An undirected graphical model consists of a set of vertices $V = \{v_t\}$ and edges $e_{tu} \in E$. Two vertices v_t, v_u are *connected* if there exists an edge $e_{tu} \in E$ between them, and a subset $A \subset V$ is *fully connected* if all pairs of vertices $v_t, v_u \in A$ are connected. Each vertex v_t is associated with a random variable x_t, and the edges of the graph are used to indicate conditional independence relationships through *graph separation*. Specifically, if every path from a set $A \subset V$ to another $C \subset V$ passes through a set $B \subset V$ (see Figure 1), then the sets of random variables $x_A = \{x_a : v_a \in A\}$ and $x_C = \{x_c : v_c \in C\}$ are independent given $x_B = \{x_b : v_b \in B\}$.

The Hammersley-Clifford theorem [9] quantifies the relationship between a graph and the joint distribution of its random variables x_t, in terms of *potential functions* ψ which are defined solely on the cliques (the fully connected subsets of V), denoted Q:

$$p(x_1, \ldots, x_N) \propto \prod_{\text{cliques } Q} \psi_Q(\{x_i : i \in Q\}) \qquad (3.1)$$

Again taking x_t to be the location of sensor t, we relate Equation (3.1) to the self-calibration problem by examining the form of the joint distribution between locations $\{x_t\}$ and observations $\{o_{tu}\}, \{d_{tu}\}$. This joint distribution is

$$p(\{x_t\}, \{o_{tu}\}, \{d_{tu}\}) = \prod_t p_t(x_t)$$
$$\prod_{(t,u)} p(o_{tu}|x_t, x_u) \prod_{(t,u):o_{tu}=1} p(d_{tu}|x_t, x_u) \qquad (3.2)$$

since by definition, both the binary variable o_{tu} and (if $o_{tu} = 1$) the observed distance d_{tu} depend only on the sensor locations x_t, x_u, and the prior information $p_t(x_t)$ is assumed independent for each x_t.

From (3.2) we can immediately define potential functions to satisfy (3.1). Notably, this only requires functions defined over single nodes and pairs of nodes. We take

$$\psi_t(x_t) = p_t(x_t) \qquad (3.3)$$

to be the single-node potential at node v_t, and

$$\psi_{tu}(x_t, x_u) = \begin{cases} P_o(x_t, x_u)\, p_\nu(d_{tu} - \|x_t - x_u\|) & \text{if } o_{tu} = 1 \\ 1 - P_o(x_t, x_u) & \text{otherwise} \end{cases} \qquad (3.4)$$

to be the pairwise potential between nodes v_t, v_u. It follows that the joint posterior likelihood of the x_t is given by

$$p(\{x_t\}|\{o_{tu}\}, \{d_{tu}\}) \propto \prod_t \psi_t(x_t) \prod_{t,u} \psi_{tu}(x_t, x_u) \qquad (3.5)$$

This implies that *every* pair of sensors be joined by an edge. We further simplify the problem by constructing a *local* approximation; specifically, we retain only the edges corresponding to observed distances. Although including additional edges enables NBP to benefit from more of the information provided by P_o, this approximation is an implicit part of the nonlinear least-squares estimate of [3] and thus leads to a more equitable comparison. Furthermore, it should be noted that this approximation is exact if P_o is a constant.

3.1. Belief Propagation

We now turn to the task of estimating the sensor locations. Inference between variables defined on a graphical model is a problem which has received considerable attention. Although exact inference in general graphs can be NP-hard, approximate inference algorithms such as loopy belief propagation (BP) [10, 11] produce excellent empirical results in many cases. BP can be formulated as an iterative, local message passing algorithm, in which each node v_t computes its "belief" about its associated variable x_t, communicates this belief to and receives messages from its neighbors, then updates its belief and repeats. In the wireless localization context, such algorithms are particularly apropos.

The object of the BP algorithm is to estimate the posterior marginal distributions $p(x_t|\{o_{ij}\}, \{d_{ij}\})$ of each variable x_t. Ideally, we might prefer the joint MAP (maximum *a posteriori*) configuration of sensor locations; however, approximating the joint MAP by the max likelihood location of each posterior marginal is a common practice in problems defined on graphs [12]. Thus, we apply BP to estimate each sensor's posterior marginal, and use the ML locations and associated uncertainty to characterize sensor placements.

The computations performed at each iteration of BP are relatively simple. Each node v_t computes its belief about x_t (an estimate of the posterior likelihood of x_t) at iteration n by taking a product of its local potential ψ_t (if any) with the messages from its neighbors Γ_t:

$$\hat{p}^n(x_t) = \alpha \psi_t(x_t) \prod_{u \in \Gamma_t} m^n_{ut}(x_t) \qquad (3.6)$$

```
Compute messages: Given $M$ weighted samples $\{w_t^{(i)}, x_t^{(i)}\}$ from $\hat{p}^n(x_t)$, estimate $m_{tu}^n(x_u)$ for each neighbor $u \in \Gamma_t$:
  • Draw random values for $\theta^{(i)} \sim U[0, 2\pi)$ and $\nu^{(i)} \sim p_\nu$
  • Means: $m_{tu}^{(i)} = x_t^{(i)} + (d_{tu} + \nu^{(i)})[\sin(\theta^{(i)}); \cos(\theta^{(i)})]$
  • Weights: $w_{tu}^{(i)} = \frac{P_o(m_{tu}^{(i)}, x_t^{(i)})\ w_t^{(i)}}{m_{ut}^{n-1}(x_t^{(i)})}$
  • Variance: e.g. $\Sigma_{tu} = M^{-\frac{1}{6}} \cdot \text{Covar}[m_{tu}^{(i)}]$
Compute marginals: Given Gaussian mixture messages $m_{ut}^n = \{m_{ut}^{(i)}, w_{ut}^{(i)}, \Sigma_{ut}\}, u \in \Gamma_t$, draw samples from $\hat{p}^{n+1}(x_t)$:
  • For each neighbor $u \in \Gamma_t$,
      – Draw $\frac{kM}{|\Gamma_t|}$ samples $\{x_t^{(i)}\}$ from each message $m_{ut}^n$
      – Weight by $w_{ut}^{(i)} = \prod_{v \in \Gamma_t \setminus u} m_{vt}^n(x_t^{(i)})$
  • From these $kM$ locations, re-sample by weight $M$ times
```

Algorithm 1: Using NBP to compute messages and marginals for sensor localization.

Here α denotes an arbitrary constant of proportionality, usually chosen to normalize $\hat{p}^n$, i.e. $\int \hat{p}^n(x_t)dx_t = 1$. The messages m_{tu} from the node v_t to v_u are computed by

$$m_{tu}^n(x_u) = \alpha \int \psi_{tu}(x_t, x_u) \frac{\hat{p}^{n-1}(x_t)}{m_{ut}^{n-1}(x_t)}\, dx_t \qquad (3.7)$$

(with α again chosen to normalize m_{tu}^n). Both equations are easily computed for discrete or Gaussian likelihood functions; however for more general likelihoods (such as those occurring in sensor localization) exact computation is intractable. We thus approximate the computations using a recent Monte Carlo method called *nonparametric belief propagation* (NBP), discussed next.

3.2. Nonparametric Belief Propagation

Neither discrete nor Gaussian BP is well-suited for the sensor self-calibration problem, as even the two-dimensional space in which the x_t reside is too large to accommodate an efficient discretized estimate, and the presence of non-Gaussian uncertainties and nonlinear relationships makes Gaussian BP undesirable as well. Fortunately, the recent development of a version of BP making use of particle-based representations, called nonparametric belief propagation (NBP, [4]) enables the application of BP to inference in sensor networks.

In NBP, each message is represented using a sample-based density estimate (as a mixture of Gaussians). The belief and message update equations (3.6-3.7) are performed using stochastic approximations, in two stages: first drawing samples from the estimated marginal $\hat{p}(x_t)$, then using these samples to approximate each outgoing message m_{tu}. We discuss each of these steps in turn, and summarize the procedure in Algorithm 1.

Given M samples $\{x_t^{(i)}\}$ from the marginal estimate $\hat{p}_t^n(x_t)$ obtained at iteration n, computing a Gaussian mixture estimate of the outgoing message from t to u is relatively simple. Each sample $x_t^{(i)}$ is moved in a random direction by d_{tu} plus noise:

$$m_{tu}^{(i)} = x_t^{(i)} + (d_{tu} + \nu^{(i)})[\sin(\theta^{(i)}); \cos(\theta^{(i)})] \qquad (3.8)$$

where $\theta \sim U[0, 2\pi)$ and $\nu \sim p_\nu$. The samples are then weighted by $\frac{P_o(m_{tu}^{(i)}, x_t^{(i)})}{m_{ut}(x_t^{(i)})}$ (see Equation (3.7)), and a single covariance Σ_{tu} is assigned to all samples. There are a number of possible techniques for choosing the covariance Σ_{tu}; one simple method is the *rule of thumb* estimate [13], given by computing the (weighted) covariance of the samples (denoted Covar$[m_{tu}^{(i)}]$) divided by $M^{\frac{1}{6}}$.

Estimation of the marginal $\hat{p}^n = \psi_t \prod m_{ut}$ is potentially more difficult. Since it is the product of several Gaussian mixtures, computing $\hat{p}^n$ exactly is exponential in the number of incoming messages. However, efficient methods of drawing samples from the product of several Gaussian mixture densities has been previously investigated in [14]; in this work we primarily use a technique called *mixture importance sampling*. In order to draw M samples from $p^n(x_t)$, we create a collection of $k \cdot M$ weighted samples (where $k \geq 1$ is a parameter of the sampling algorithm) by drawing $\frac{kM}{|\Gamma_t|}$ samples from each message $m_{ut}, u \in \Gamma_t$ and assigning each sample a weight equal to the product of the other messages $\prod_{v \in \Gamma_t \setminus u} m_{vt}$. We then draw M values from this collection with probability proportional to their weight, yielding samples drawn from the product of all incoming messages.

Furthermore, it is trivial to change the form of the noise distribution p_ν so long as sampling remains tractable. This fact can be used to approximate a broad outlier process, but due to the form of NBP's messages as Gaussian mixtures there is a more elegant solution available. We augment the Gaussian mixtures in each message by a single, high-variance Gaussian to approximate an outlier process in the uncertainty about d_{tu}. To be precise, we add an extra particle to each outgoing message, centered at the mean of the other particles and with weight and variance chosen to model the expected outliers, e.g. weight equal to the probability of an outlier, and standard deviation sufficiently large to cover the expected support of P_o. This method requires fewer samples to adequately represent the message, and thus is also more computationally efficient when sampling from the product of many messages.

4. SELF-CALIBRATION SIMULATIONS

We demonstrate NBP's utility for solving self-calibration problems on an example sensor network. Figure 2(a) shows a small network ($N = 10$), with distance measurements indicated as lines, on which we have introduced a single outlier measurement (the dashed line). We perform calibration relative to the three sensors shown as circles. If the erroneous measurement could be detected and discarded, the optimal joint MAP sensor locations can be found using an iterative nonlinear least-squares optimization [3]; the residual errors after this procedure (for a single noise realization) are shown in Figure 2(b). However, with the outlier measurement present, the same nonlinear least-squares procedure results in a large distortion in the estimates of some sensor locations (Figure 2(c)). NBP, by virtue of the measurement outlier process discussed in Section 3.2, remains robust to this error and produces the near-optimal estimate shown in Figure 2(d).

In order to provide a measure of the robustness of NBP in the presence of non-Gaussian (outlier) distance measurements, we perform Monte Carlo trials, keeping the same sensor locations and connectivity used in Figure 2(a) but introducing different sets of outlier measurements. At every trial, each distance measurement is replaced with probability .05 by a value drawn uniformly between zero and the maximum distance ($d_{\max}$) between any two sensors. As there are 37 measurements in the network, on average approximately two outlier measurements are observed in each trial. We then measure the number of times each sensor's estimated location is within distance R of its true location, as a function of R (nor-

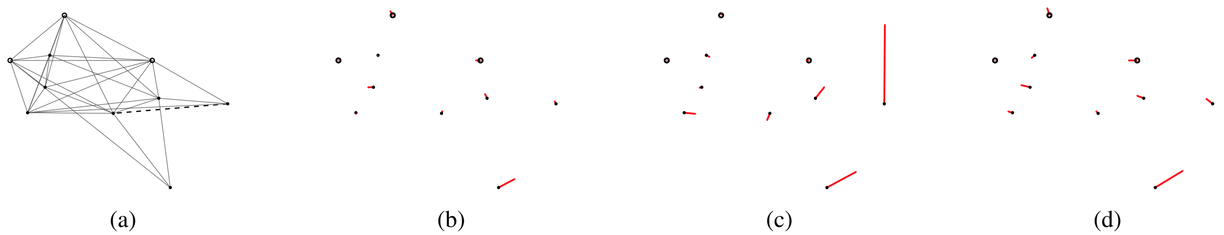

Fig. 2. (a) A small (10-sensor) graph and the observable pairwise distances; calibration is performed relative to the location of the sensors shown in green. One distance (shown as dashed) is highly erroneous, due to a measurement outlier. (b) The MAP estimate of location, discarding the erroneous measurement. (c) A nonlinear least-squares estimate of location is highly distorted by the outlier; (d) NBP is robust to the error by inclusion of a measurement outlier process in the model.

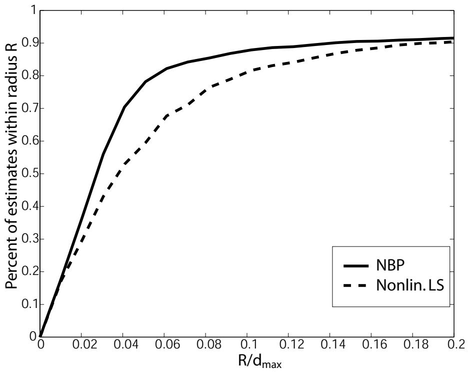

Fig. 3. Monte Carlo self-calibration trials on the sensor network in Figure 2(a). We measure the probability of a sensor's estimated location being within a radius R of its true location (normalized by the max inter-sensor distance $d_{\max}$) for both NBP and nonlinear least-squares, indicating NBP's superior performance in the presence of outlier measurements.

malized by $d_{\max}$). This curve is shown in Figure 3 for both NBP and nonlinear least-squares estimation. As can be seen, NBP more often provides an estimate which is "nearby" to the true sensor location, indicating its increased robustness to the outlier noise.

However, Figure 3 understates the advantages of NBP for this scenario. One of the features of NBP is that it also provides an estimate of the *uncertainty* in sensor position; the trials which resulted in large errors also showed highly uncertain (often bimodal) estimates for the sensor locations in question. Thus, in addition to providing a more robust estimate of sensor location, NBP also provides a measure of the reliability of each estimate.

5. CONCLUSIONS

We have described a method for sensor self-calibration based on NBP, a nonparametric message-passing algorithm for inference in graphical models. NBP provides a number of advantages, including naturally distributed computation, an inherent estimate of uncertainty, and the ability to incorporate non-Gaussian measurement models. Focusing primarily on the latter, we demonstrated that by incorporating an outlier process in the model, NBP's estimates of sensor location can be made robust to the occurrence of a few highly erroneous measurements.

6. REFERENCES

[1] S. Kumar, F. Zhao, and D. Shepherd (Eds.). Special issue on collaborative information processing. *IEEE Signal Processing Mag.*, 19(2), March 2002.

[2] H. Gharavi and S. Kumar (Eds.). Special issue on sensor networks and applications. *Proc. IEEE*, 91(8), August 2003.

[3] R. Moses and R. Patterson. Self-calibration of sensor networks. In *SPIE vol. 4743: Unattended Ground Sensor Technologies and Applications IV*, 2002.

[4] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. In *CVPR*, 2003.

[5] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43(4):499–518, November 2003.

[6] Neal Patwari and Alfred Hero. Relative location estimation in wireless sensor networks. *IEEE Trans. Signal Processing*, 51(8):2137–2148, August 2003.

[7] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), December 2000.

[8] M. Fazel, H. Hindi, and S. P. Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proceedings, American Control Conference*, 2003.

[9] S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, 1996.

[10] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufman, San Mateo, 1988.

[11] K. Murphy, Y. Weiss, and M. Jordan. Loopy-belief propagation for approximate inference: An empirical study. In *UAI 15*, pages 467–475, July 1999.

[12] B. Frey, R Koetter, G. Forney, F. Kschischang, R. McEliece, and D. Spielman (Eds.). Special issue on codes and graphs and iterative algorithms. *IEEE Trans. Inform. Theory*, 47(2), February 2001.

[13] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, New York, 1986.

[14] A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky. Efficient multiscale sampling from products of Gaussian mixtures. In *NIPS 17*, 2003.