[Mujdat Çetin, Lei Chen, John W. Fisher III, Alexander T. Ihler,
Randolph L. Moses, Martin J. Wainwright, and Alan S. Willsky]

Distributed Fusion in Sensor Networks

[A graphical models perspective]

Distributed inference methods developed for graphical models comprise a principled approach for data fusion in sensor networks. The application of these methods, however, requires some care due to a number of issues that are particular to sensor networks. Chief of among these are the distributed nature of computation and deployment coupled with communications bandwidth and energy constraints typical of many sensor networks. Additionally, information sharing in a sensor network necessarily involves approximation. Traditional measures of distortion are not sufficient to characterize the quality of approximation as they do not address in an explicit manner the resulting impact on inference which is at the core of many data fusion problems. While both graphical models and a distributed sensor network have network structures associated with them, the mapping is not one to one. All of these issues complicate the mapping of a particular inference problem to a given sensor network structure. Indeed, there may be a variety of mappings with very different characteristics with regard to computational complexity and utilization of resources. Nevertheless, it is the case that many of the powerful distributed inference methods have a role in information fusion for sensor networks. In this article we present an overview of research conducted by the authors that has

1053-5888/06/$20.00©2006IEEE

sought to clarify many of the important issues at the intersection of these domains. We discuss both theoretical issues and prototypical applications in addition to suggesting new lines of reasoning.

INTRODUCTION

Fusion of information in interconnected sensor networks and the design of inference algorithms for graphical models are far from synonymous lines of inquiry. That said, the evocative "message-passing" structure of many graphical model inference algorithms has motivated a number of research groups to examine how methods for such models might be applied or adapted to challenges that arise in fusion for sensor networks. A variety of questions arise that fall outside the standard domain of graphical models. The result is a rich, new area of research that has already shown its promise and offers more for the future.

Several reasons underlie the choice of graphical models for addressing distributed fusion problems in sensor networks. First, a graphical model is well suited to capture the structure of a sensor network, which consists of nodes (for sensing, communication, and computation) as well as connections between the nodes (for modeling statistical dependencies and/or communication links). Second, there has recently been significant progress in the development and analysis of scalable inference algorithms on graphs. This includes recent work on analyzing and understanding the behavior of well-known classes of local message-passing algorithms on graphs that contain loops [14], [32]. There has also been considerable progress in developing new message-passing algorithms with superior convergence and accuracy properties; in some cases these algorithms have provable guarantees of convergence to optimal answers [18], [31], [33]. Third, the algorithms commonly used for performing estimation in graphical models involve parallel message-passing operations that are not only efficient and scalable but also well suited to parallel realization via physically distributed processors, a crucial requirement in sensor network applications. Finally, graphical models provide a suitable framework for the development and analysis of communication-constrained versions of message-passing algorithms, which are of considerable interest in the context of sensor networks given their severe power and energy limitations [15]; see also [7]. Distributed inference in sensor networks under communication constraints is a topic of much current interest. Aldosari and Moura [1] address the use of these algorithms for distributed decision making as well as the effect of graph topology on convergence rates. The message-passing algorithms studied in this article, while closely related, are designed for statistical estimation and fusion. Xiao et al. [37] consider random and nonrandom parameter estimation in networks in which sensors transmit quantized measurements.

We describe our ongoing line of inquiry on this subject. We use two applications—self-localization in sensor networks and distributed data association in multiobject tracking—first to show how sensor network fusion problems can be cast as problems of inference in graphical models and then to look more deeply at how conservation of power through judicious use of communications resources provides new insights not previously found in the graphical model literature.

The need to conserve communications resources requires that the so-called "messages" found in inference algorithms for graphical models be compressed and/or "censored" (i.e., not transmitted). We describe results of such censoring methods for the data association problem and of message approximation methods for the self-localization problem. The method we describe for message censoring—one that corresponds to a very simple and local fusion protocol—provides a sensor network with an adaptive fusion/communication mechanism that yields excellent fusion performance with significantly reduced expenditure of communications resources. This framework leads to a principled means for trading off communication load with the accuracy of the resulting fused inference results. In addition, this research provided the motivation and basis for examining the effects of message "errors" (due to approximation, censoring, or transmission loss) on overall inference accuracy, an inquiry that had the surprising and important side benefit of yielding the best known results on convergence of the most widely used inference algorithm for graphical models. These topics are developed in far greater detail in a number of publications that are referenced throughout this article.

GRAPHICAL MODELS

We begin with a brief discussion of graphical models, focusing on aspects related to inference and linking these to distributed inference in sensor networks in later sections. A *graphical model* on an (undirected) graph, $G = (V, E)$ (consisting of a vertex or node set V and edge set $E \subset V \times V$) consists of a collection of random variables or vectors, $X = \{X_v, v \in V\}$, that collectively satisfy a Markov property with respect to G; specifically, for any subset U of V, let $X_U = \{X_v, v \in U\}$. The random vector X is Markov with respect to G if for any partition of the V into disjoint sets A, B, C, in which B *separates* A and C (i.e., all paths in G from A to C include vertices in B), the random vectors X_A and X_C are conditionally independent given X_B. For the "graph" associated with time series, i.e., consecutive points in time with each point connected to its immediate predecessor and successor, this corresponds to the usual notion of temporal Markovianity (i.e., that the past and future are conditionally independent given the present). For general graphs, however, the Markov property requires a far richer set of conditional independencies and associated challenges in both specifying such distributions and in performing inference using them. By way of example, consider Figure 7(a) (used for subsequent analysis) in which the variables x and y are conditionally independent given the variables w and z. Due to the edge between w and z, they are *not* conditionally independent given x and y.

The celebrated Hammersley-Clifford Theorem [2] provides a sufficient condition (necessary for strictly positive probability distributions) for the form that the joint distribution must take to be Markov with respect to G. Specifically, let $\mathcal{C}$ denote the set of all *cliques* in G, where a subset of nodes C is a clique if it is fully connected (i.e., an edge exists between each pair of

nodes in C). The random vector X is Markov with respect to G if (and only if for strictly positive probability distributions) its distribution admits a factorization as a product of functions of variables restricted to cliques

$$p(x) = \frac{\prod_{C \in \mathcal{C}} \psi_C(x_C)}{Z} \; ; \; Z \triangleq \sum_x \prod_{C \in \mathcal{C}} \psi_C(x_C), \tag{1}$$

where Z is the partition function, and the $\psi_C(x_C)$ are so-called compatibility functions. The logarithms of these compatibility functions are commonly referred to as potentials or potential functions.

For simplicity we will assume for the remainder of this article that each of the nonzero potentials (or equivalently each compatibility function in (1) that is not constant) is a function either of the variable at a single node of the graph (*node potentials*) or of the variables at a pair of nodes corresponding to an edge in E (*edge potentials*). In this case, (1) takes the form

$$p(x) = \frac{\left(\prod_{s \in V} \psi_s(x_s)\right) \left(\prod_{(s,t) \in E} \psi_{s,t}(x_s, x_t)\right)}{Z}. \tag{2}$$

Note that any graphical model can be put into this form by appropriate node aggregation [2]. While all of the ideas that are presented here can be extended to the more general case, pairwise potentials are sufficient for the specific applications considered in this article. Moreover, the communication interpretation of the so-called message-passing algorithms used herein are more easily explained in this context.

As long as G is a relatively sparse graph, the factorizations (1) or (2) represent parsimonious means to describe the joint distribution of a large number of random variables, in the same way that specifying an initial (or final) distribution and a set of one-step transition distributions is a compact way in which to specify a Markov chain. Moreover, for many inference and estimation problems (including those described in this article), such a specification is readily available. The challenge, however, is that unless the graph has very special properties, such as in the case of Markov chains, the compatibility functions do not readily describe the quantities of most interest, such as the marginal distributions of the variables at individual (or small sets of) nodes or the overall peak of the distribution jointly optimized over all nodes. Indeed, for discrete-valued random variables the computation of such quantities for general graphs is NP-hard.

For graphs without loops (Markov chains and, more generally, graphical models on trees), computation of the marginal distributions is relatively straightforward. In this case, the node and pair-wise potentials of the joint distribution in (2) for any cycle-free graph can be expressed in terms of the marginal probabilities at individual nodes and joint probabilities of pairs of nodes connected by edges [6], [32]

$$p(x) = \prod_{s \in V} p_s(x_s) \prod_{(s,t) \in E} \frac{p_{st}(x_s, x_t)}{p_s(x_s) p_t(x_t)}. \tag{3}$$

That is, $\psi_s(x_s) = p_s(x_s)$ [or $\psi_s(x_s) = p_s(x_s)p(y_s|x_s)$ when there is a measurement y_s associated with x_s] and $\psi(x_s, x_t) = (p(x_s, x_t)/p(x_s)p(x_t))$. Marginal probabilities can be efficiently calculated in a *distributed* fashion by so-called sum-product algorithms. Specifically, as shown in [26], the marginal probabilities at any node s in the graph can be expressed in terms of the local potential ψ_s at node s, along with a set of so-called messages from each of its neighbors in the set $\mathcal{N}(s) = \{t \in V | \, (s, t) \in E\}$. The message from node t to node s is a function $M_{ts}(x_s)$ that (up to normalization) represents the likelihood function of x_s based on the subtree rooted at t and extending away from s. In particular, the marginal distribution p_s takes the form

$$p_s(x_s) \propto \psi_s(x_s) \prod_{t \in \mathcal{N}(s)} M_{ts}(x_s). \tag{4}$$

Furthermore, in the absence of loops, these messages are related to each other via a sum-product formula:

$$M_{ts}(x_s) \propto \sum_{x_t} \psi_{st}(x_s, x_t) \psi_t(x_t) \prod_{u \in \mathcal{N}(t) \setminus s} M_{ut}(x_t). \tag{5}$$

The product operation embedded in the message computation from node t to node s combines the information in the subtree rooted at node t, by combining the likelihood information from all neighbors of node t other than s with the local potential at node t. This yields a likelihood function for the random variable X_t at node t. This is then converted to a likelihood for the random variable X_s at node s by multiplying by the compatibility function between these two nodes and then "summing" or integrating out the variable at node t in a fashion analogous to the Chapman-Kolmogorov equation in a Markov chain.

Together (4) and (5) relating messages throughout the loop-free graph represent a set of fixed-point equations that can be solved in a variety of ways corresponding to different message-passing algorithms. For example, one can solve these equations explicitly, much as in Gaussian elimination, by starting at leaf nodes, working inward toward a "root" node, and then propagating back toward the leaves; this is a generalization of two-pass smoothing algorithms for Markov chains. An alternative is to solve these equations iteratively; we begin with guesses (often taken simply to be constant) of all of the messages and iteratively update messages by substitution into the fixed-point equations. Each step of this procedure involves passing the current guess of messages among neighboring nodes. While there is great flexibility in how one schedules these messages, the happy fact remains that after a sufficient number of iterations (enough so that information propagates from every node to every other), the correct messages are obtained from which the desired probabilities can then be computed.

Interestingly, for loop-free graphs, a variant of this approach also yields the solution to the problem of computing the overall maximum a posteriori (MAP) configuration for the entire graphical model. For such graphical models, there is an

alternative factorization of $p(x)$ in terms of so-called *max-marginals*. As their name would suggest, these quantities are defined by eliminating variables through maximization (as opposed to summation); in particular, we define

$$q_s(x_s) := \max_{x_{u,u\in V\setminus s}} p(x_1, \ldots, x_n) \tag{6a}$$

$$q_{st}(x_s, x_t) := \max_{x_u, u\in V\setminus\{s,t\}} p(x_1, \ldots, x_n). \tag{6b}$$

It is a remarkable fact that for a tree-structured graph, the distribution (1) can also be factorized in terms of these max-marginals, viz.,

$$p(x) \propto \prod_{s\in V} q_s(x_s) \prod_{(s,t)\in E} \frac{q_{st}(x_s, x_t)}{q_s(x_s)q_t(x_t)}. \tag{7}$$

Furthermore, there are equations analogous to those for the sum-product algorithm that show how these quantities can be computed in terms of node potentials and messages, where the fixed-point equations involve maximization rather than summation (yielding what are known as max-product algorithms). The solution of these fixed-point equations can be computed via leaf-root-leaf message passing (corresponding to dynamic programming/Viterbi algorithms) or by iterative message passing with more general message scheduling.

While the representation of marginal distributions or max-marginals at individual nodes in terms of messages holds only if the graph is loop free, (4) and (5) are well defined for any graph, and one can consider applying the sum-product algorithm to arbitrary graphs which contain loops (often referred to as loopy belief propagation); this corresponds to fusing information based on assumptions that are not precisely valid. For example, the product operation in (4) as well as (5) corresponds to the fusion of information from the different neighbors of a node, t, assuming that the information contained in the messages from these different neighbors are conditionally independent given the value of x_t, something that is valid for trees but is decidedly *not* true if there are loops.

Despite this evident suboptimality, this loopy form of the sum-product algorithm has been extremely successful in certain applications, most notably in decoding of low-density parity check codes [20], [28], which can be described by graphs with long cycles. In contrast, sensor network applications (as well as others) involve graphs with relatively short cycles. This has led to a considerable and still growing body of literature on the analysis of these algorithms on arbitrary graphs as well as the development of new ones that yield superior performance. For arbitrary loopy graphs, the reparameterization perspective on these algorithms [32], in conjunction with a new class of efficiently computable bounds [34] on the partition function Z, provide computable bounds on the error incurred via application of sum-product to loopy graphs. Similar analysis is also applicable to the max-product updates [9], [31]. The fact that the max-product algorithm may yield incorrect (i.e., non-MAP) configurations motivates the development of a new class of tree-reweighted max-product algorithms (TRMP) [33] for which, in sharp contrast with the ordinary max-product updates, there is a set of testable conditions for determining if the solution is indeed the MAP configuration. Tight performance guarantees can be provided for TRMP for specific classes of graphs [8], [18]. More broadly, we refer the reader to various research and survey papers, e.g., [23], [35], [38], as well as citations at the end of this article, which provide only a sampling of this rapidly growing literature.

NONPARAMETRIC BELIEF PROPAGATION

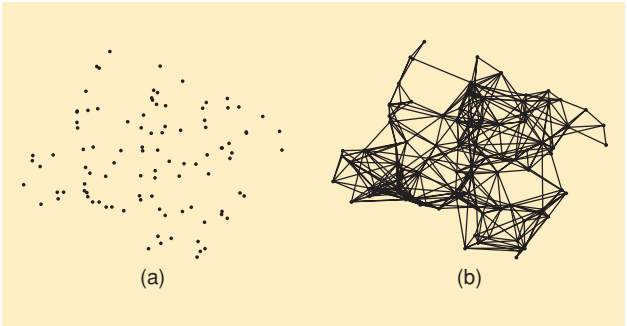
We next provide a brief discussion of one specific contribution that plays a role in what follows; specifically, whether applied to loop-free or loopy graphs, message-passing algorithms corresponding to the iterative solution of (5) require the transmission of full likelihood functions, each of which is a function of the variable at the receiving node. In the case of discrete random variables, this corresponds to the transmission of a vector of numbers, while for Gaussian models, these messages can be reduced to the transmission of means and covariances. However, for non-Gaussian continuous random variables, as commonly occur in many sensor network applications, one must transmit a representation of an entire function $M_{ts}(x_s)$. A common approach is simply to discretize the underlying continuous variables often leading to unwarranted computational complexity (and communication overhead for sensor networks). An alternative, developed recently, is nonparametric belief propagation (NBP), representing a significant generalization of particle filtering methods for Markov chains [29].

Particle filtering for Markov chains also involves an equation similar to (5), with the important distinction that there is no "product" as there is only one neighbor of node t. The set of particles, representing samples from the single message corresponding to the product term in (5), are weighted by the local node compatibility function (and perhaps resampled from the weighted particle representation). This step is followed by a "sum" operation by simulating the transition dynamics from node t to s, i.e., for each sample at node t we sample from the transition distribution to generate a sample at node s. All of these steps, with one significant exception, apply equally well to message passing in general graphical models. The additional complexity is that we have the product of particle-based messages from *several* neighbors of node s, which requires that we make sense of this product and then find an efficient method to generate particles corresponding to this product.

As developed in [29], each particle-based message can be interpreted as a nonparametric estimate of the likelihood function or probability density function corresponding to the exact message; e.g., if we use Gaussian kernels for these nonparametric densities, the set of particles corresponds to a Gaussian mixture. Consequently, the problem of generating samples from the product of messages, with each represented

by a set of particles, reduces to drawing samples from a density that is the product of a set of Gaussian mixtures. Unless we are careful, however, we may have a geometrically growing number of terms in these resulting Gaussian mixtures, making sampling intractable. The key to overcoming this problem is that of finding ways of sampling from this product without explicitly constructing the product. Generating such a sample can be accomplished in two steps: choosing one of the product Gaussian kernels from which to sample (i.e., the set of labels corresponding to specific terms in each of the Gaussian mixtures in the product) and then drawing a sample from the Gaussian corresponding to that set of labels. Since sampling from a Gaussian is straightforward, the real challenge is in sampling from the sets of labels of components, something that can be solved via importance and Gibbs sampling methods.

Moreover, dramatic speedups can be achieved through a multiresolution representation utilizing k-dimensional (KD) trees, a data structure that plays a very important role in efficient coding and communication of messages. Specifically, suppose that the variables at each node are KD real-valued vectors. Then, given a set of KD particles representing one of the messages in (5), we aggregate these into a multiresolution tree, starting with all of the particles clustered together at the root of this tree. Proceeding down the tree, at each node, we take whichever particles are clustered at that node and divide them into two subclusters (which correspond to the children of this node) along one of the components of the vector (so that we cycle through these components as we proceed down the tree). At the finest level we have individual particles and hence individual "labels" for a term in the Gaussian mixture representation of this message. Sampling of labels can then be accomplished very efficiently using this coarse-to-fine representation. We refer the reader to [11] for details. Note that here, *nonparametric* refers to the sample-based representation of the messages, while the graphical model itself remains defined in terms of the potential functions ψ (which are parametric in our applications). In contrast, the paper by Predd et al. [27] discusses nonparametric, data-based *models* in sensor networks.



[FIG1] Sensor localization: (a) the physical location of a collection of sensors may be represented as (b) a graphical model in which nodes correspond to the sensor node location variables and edges correspond to observed information, such as intersensor distance measurements from a subset of node pairs.

MAPPING TWO PROTOTYPICAL APPLICATIONS TO GRAPHICAL MODELS

Mapping fusion problems in sensor networks to problems on graphical models might at first seem to be an obvious function, as there is a natural graph that exists, defined by the sensor nodes and the intersensor communication structure. However, it is the *informational structure* of the inference problem, involving the relationships between sensed information and the variables about which we wish to perform estimation, that is just as critical as the communication structure of the problem. We describe two sensor networks applications here, focusing on the informational structure.

THERE HAS ALSO BEEN CONSIDERABLE PROGRESS IN DEVELOPING NEW MESSAGE-PASSING ALGORITHMS WITH SUPERIOR CONVERGENCE AND ACCURACY PROPERTIES.

SELF-LOCALIZATION IN SENSOR NETWORKS

A well-recognized problem for many sensor network applications is that of sensor localization. Figure 1 illustrates such a problem. Here each node corresponds to a sensor, and the random vector at that node describes sensor calibration information, such as its location, and also perhaps its orientation (e.g., if the node provides directional-sensing capability), and time offset (e.g., if intersensor time-of-flight measurements are used). While the framework we describe extends immediately to a general setting, we assume for simplicity that only the location variables are of interest. We consider a case in which the available information for estimating sensor locations consist of: i) uncertain prior information about the location of a subset of the sensors (e.g., if any of the sensors are provided with GPS); ii) the ability of sensors to "hear" one another and attempt to measure their intersensor distance (typically only possible for nearby sensors); and iii) any distance measurements so obtained.

Specifically, let us denote by $\rho_s(x_s)$ the prior location probability distribution for sensor s, if any, let $Pr(x_s, x_t)$ be the probability of obtaining a distance measurement between two sensors s, t located at x_s and x_t, and let $\rho_L(l_{st}|x_s, x_t)$ the probability distribution of measuring a distance l_{st} given that the true sensor positions are x_s and x_t. Notice that all three sources of information involve only the variables at single sensors or pairs of sensors; thus we may use a pairwise graphical model to describe the joint distribution of sensor locations, with each node in the graph associated with one of the sensors and its variables of interest. One may immediately write that the joint distribution has the form of (2), with

$$\psi_s(x_s) = \rho_s(x_s) \tag{8a}$$

$$\psi_{st}(x_s, x_t) = \begin{cases} Pr(x_s,\ x_t)\rho_L(l_{st}); & l_{st} \text{ is observed} \\ 1 - Pr(x_s,\ x_t); & \text{otherwise.} \end{cases} \tag{8b}$$

The sensor localization problem is then precisely one of computing the best estimates of all sensor locations given all

of this information. As an optimization problem this has been well studied by others (e.g., [24], [25], [30], often under the assumption that the distributions (ρ_s, ρ_L) involved are Gaussian so that its solution entails the nonlinear optimization of a quadratic cost function. However, by formulating this problem as one of inference for a graphical model (in which localization corresponds to computing the marginal distributions of variables at each node), we directly obtain message-passing algorithms (such as sum-product) that distribute the computations across the network. Moreover, the graphical model formulation allows us to easily include features which bring additional realism, accuracy, and well-posedness [12], [13]. In particular, anomalous range measurements can occur with nontrivial probability; this phenomenon is easily included through a simple modification of the edge potential likelihood functions to capture such a noise model. Also, location distributions for sensors can be multimodal (e.g., receiving perfect range measurements from two neighboring sensors whose locations are known perfectly yields two possible locations). The NBP formulation finds an estimate of the node location probability distributions (as opposed to a point estimate such as the distribution maximum); from which multimodal, non-Gaussian, or other characteristics are readily seen. Representing such multimodality at the individual node level is one of the strengths of the NBP algorithm, and is far easier than dealing with this at a centralized level.

WE USE TWO APPLICATIONS TO SHOW HOW SENSOR NETWORK FUSION PROBLEMS CAN BE CAST AS PROBLEMS OF INFERENCE IN GRAPHICAL MODELS.

The graphical model formulation of the localization problem also provides a natural mechanism for including information about *which* pairs of sensors are able to measure distance. This information can improve the well-posedness and reduce estimation error in the location estimates. For example, the topmost sensor node in Figure 1 forms a location estimate using distance measurements from two very closely located sensors. Thus, the distribution for the location estimate will have large values in an entire circle centered at these two other sensors. However, if we also account for the fact that the node on the other side of these two sensors *cannot* hear (or can hear only with small probability) the topmost sensor, this circular ambiguity is considerably reduced. Dealing with the absence of a measurement as an indicator of being more distant from each other can be difficult to accommodate in the traditional optimization framework (for example, it is certainly not a simple quadratic cost). In our graphical model formulation, however, this information can be included by adding edges to the graph. Specifically, we include 2-step (or more generally, n-step) edges, where a 2-step edge is one between sensors that are both heard by a common sensor but cannot hear each other; see [13] for details. Including these additional edges can increase the complexity of inference slightly, although experimentally it appears that a few edges are often sufficient to resolve most of the ambiguities that may arise.

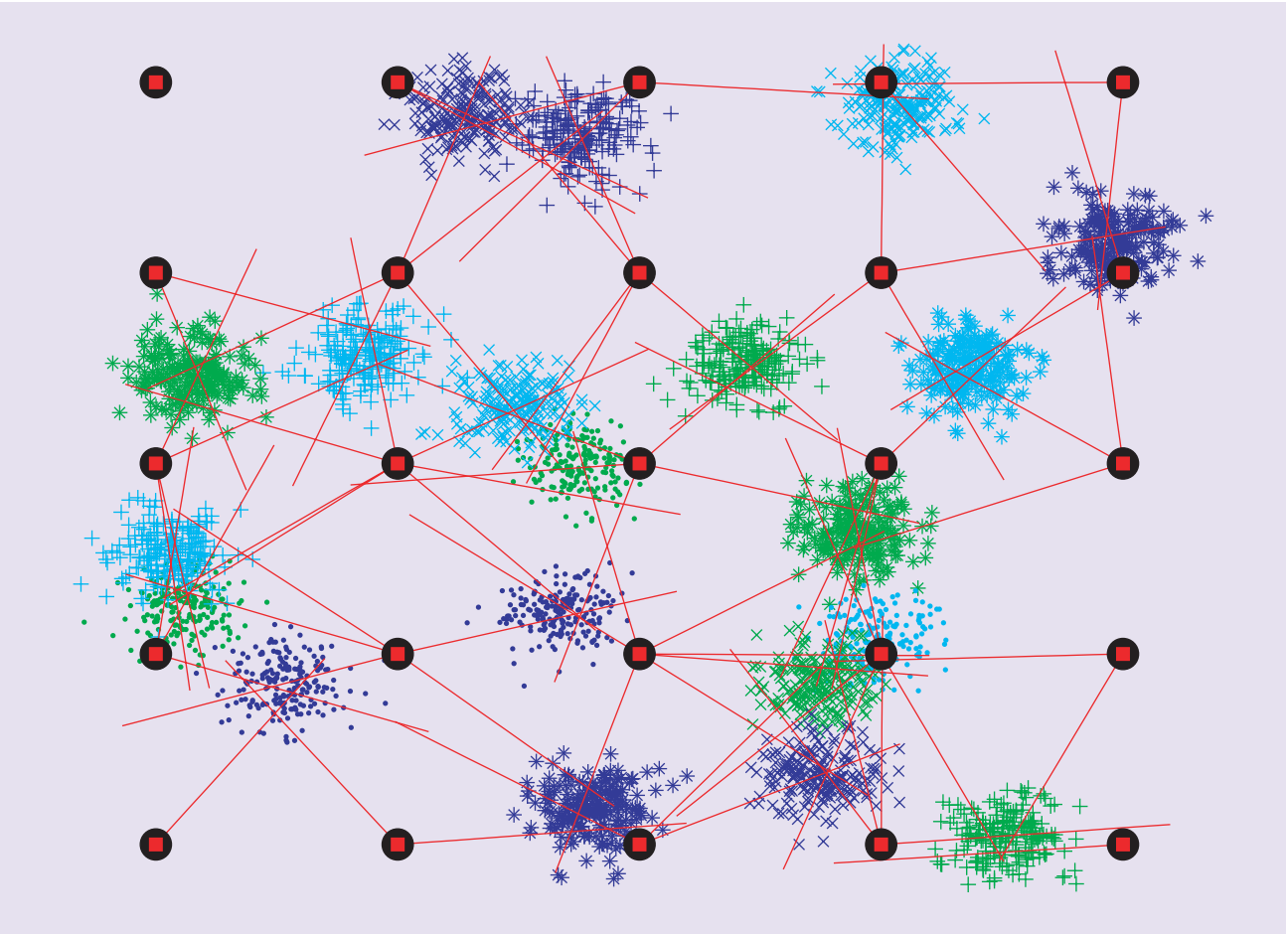
The fact that the variables in our model are continuous—leading to our use of the particle-based NBP algorithm—raises questions unique to sensor networks, namely how best to make use of scarce communication resources. For example, how many particles do we need to send, and how can we encode them efficiently? Moreover, how do we adjust this as estimates evolve dynamically (e.g., as ambiguities due to multimodality in distributions resolve themselves)? These issues are considered later in this article.

MULTIOBJECT DATA ASSOCIATION IN SENSOR NETWORKS

A second application for sensor networks is that of multisensor, multiobject tracking. This is a challenging problem even for centralized algorithms, in large part because of the embedded problem of data association, i.e., of determining which measurements from different sensors correspond to the same object. For sensor networks there are additional challenges, due to the need for distributed implementation, but typical networks also have structure; e.g., sensors have limited sensing range overlapping the range of a limited number of other sensors. This suggests new approaches for solving data association problems that are computationally feasible and fully distributed.

Figure 2 depicts a notional example of the problem. Here, a number of sensors cover a region of interest with overlapping areas of regard. A number of targets are located within the region, each sensed by one or more sensors. In this case the mapping of the inference problem to a graphical model is not unique, and different approaches present tradeoffs that lead to a very different solutions than in the well-studied case of centralized multitarget tracking. In particular, as is well-documented [22], the preferred centralized way in which to organize the various data association hypotheses is that based on so-called track hypotheses, which leads to data structures—and corresponding graphical models—in which, roughly speaking, the nodes correspond to targets.

In a sensor network, however, it is advantageous to organize the representation around sensors rather than targets. For centralized processing, such measurement-oriented approaches have been discarded for the same basic reason that purely sensor-based representations do not work here. In particular, consider a simple situation in which we know how many targets are present and we know which sensors see which targets. If we wish to use a model in which the nodes are in one-to-one correspondence with the sensors, the variable to be estimated at each node is simply the association vector that describes which measurement from that sensor goes with which target, which measurements are false alarms, and which targets in its area of regard it fails to



[FIG2] A snapshot of a typical data association scenario in a sensor network. Twenty five sensors (circle nodes) and the bearing-only measurements (line segments) are shown. Each cluster of samples represents the prior position distribution of one target.

detect. The problem with such a graphical model is that if multiple targets are seen by the same set of sensors, the likelihoods of these sets of associations (across sensors and targets) are coupled, implying that in the representation in (1), we must include cliques of size larger than two. In principle, these cliques can be quite large, and it is precisely for this reason that the association problem is NP-hard.

By taking advantage of the sparse structure of sensor networks; i.e., the fact that each sensor has only a limited field of view and thus has only a modest number of other sensors with which it interacts and small number of targets within its measurement range—one can readily construct a hybrid representation comprised of two types of nodes. Sensor nodes capture the assignment of groups of measurements to multitarget nodes in addition to assignments that do not have any multisensor/target contention. Multitarget nodes correspond to sets of targets seen by the same set of three or more sensors. In such a model, the variable at each sensor node captures the assignment of groups of measurements to each of these multitarget nodes as well as any assignments that do not have such a level of multisensor/target contention. Moreover, the resulting graphical model yields a representation as in (2) with only pairwise potentials [5]. Furthermore, while we have described the idea for the case in which we already know which targets are seen by which sets of sensors, it is also possible to formulate graphical models that deal with the problem of also determining which targets are seen by which subsets of sensors. We do so by introducing virtual nodes representing regions of space corresponding to overlaps in areas of regard of multiple sensors, as shown in Figure 3. In addition, although we have discussed the data association problem at a single time point here for simplicity, the tracking problem is dynamic, and our framework can be generalized to incorporate data from multiple time slices by using a multiple hypothesis tracking-like approach [4].

For the data association problem, both the computation of marginal probabilities and of the overall MAP estimate are of interest. The MAP estimate is of importance because it captures consistency of association across multiple sensors and targets (for example, capturing the fact that one measurement cannot correspond to multiple targets). As a result, algorithms such as sum-product and max-product are both of interest as is the issue of communications-sensitive message passing, a topic to which we turn in the next section.

SOLVING INFERENCE PROBLEMS IN COMMUNICATION-CONSTRAINED NETWORKS

The sensor network applications in the previous section are two of many that can be naturally cast as problems of inference in graphical models—at least in part, as there are other issues, including power conservation and careful use of scarce communication resources, that must be considered. Using the two previous applications as examples, we describe approaches to dealing with such power and communication issues.

MESSAGE CENSORING

As described in the preceding section, multiobject data association in sensor networks can be formulated as a problem either of computing the marginal probabilities or the overall MAP estimate for a graphical model whose variables are discrete and represent various assignments of measurements to objects or spatial regions. In our work we have applied a variety of different algorithms to solve this problem, including the sum-product algorithm [21] for the computation of approximate marginals, the max-product algorithm [31] for the computation of approximate MAP estimates, and the TRMP algorithm [31] for the computation of the true MAP estimate.

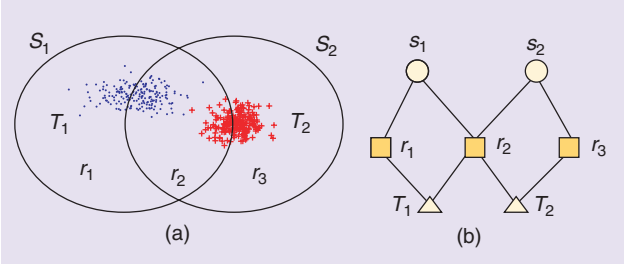
One issue with all of these algorithms is that messages are, in principle, continually transmitted among all of the nodes in the graphical model. In typical graphical model applications some type of global stopping rule is applied to decide when the inference iterations should be terminated. Also, it is often the case that convergence behavior can depend strongly on the *message schedule*—i.e., the order in which messages are created, transmitted, and processed. For sensor network applications, convergence criteria or message schedules that require centralized coordination are out of the question. Rather, what is needed are local rules by which individual nodes can decide, at each iteration, whether it has sufficient new information to warrant the transmission of a new message, with the understanding that the receiving node will simply use the preceding message if it does not get a new one and will use a default value corresponding to a noninformative message if it has not received any prior message. This formulation also allows for transmission erasures.

A simple local rule for message censoring is the following. first, we interpret each message as a probability distribution on the state of the node to which the message is to be sent; easily accomplished by normalizing the message. We then compute the Kullback-Leibler divergence (KLD) between each message and its successor

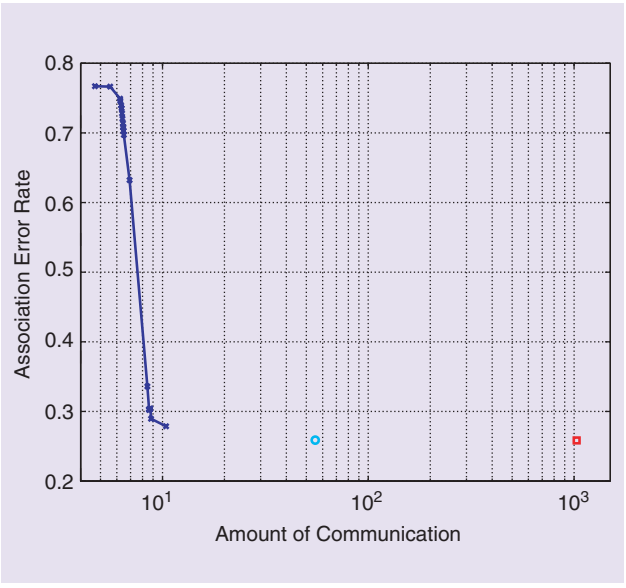
$$D\left(M_{ts}^{k} \| M_{ts}^{k-1}\right) = \sum_{x_s} M_{ts}^{k}(x_s) \log \frac{M_{ts}^{k}(x_s)}{M_{ts}^{k-1}(x_s)} \qquad (9)$$

as a measure of novel information and send M_{ts}^{k} only if $D(M_{ts}^{k} \,||\, M_{ts}^{k-1})$ exceeds a threshold ϵ. This procedure is completely local to each node and provides for network adaptivity, as these rules lead to data-dependent message scheduling; indeed, it is quite common for a node to become silent for one or more iterations and then to restart sending messages as sufficiently new information reaches it from elsewhere in the network.

When applying the above method to algorithms such as sum-product, we observe that major savings in communication (hence power) can be achieved without significant performance loss as compared to standard message passing algorithms. An example is shown in Figure 4, where the data are obtained by simulating tracking of 50 targets in a 25-sensor network and censored versions of max-product are readily compared to max-product and TRMP. In the figure, the censoring threshold is varied. It can be seen that with certain thresholds the performance deviations from the max-product



[FIG3] (a) A piece of a partially organized sensor network, where the sensor-target coverage relationship is ambiguous. Two sensors with their surveillance regions (s_1 and s_2) and nonparametric representations of two target distributions (T_2 and T_2) are shown. The surveillance area is divided into three nonoverlapping subregions (r_1, r_2, r_3), each of which is covered by a distinct subset of sensors. (b) The graphical model for the scenario in (a); circles, squares, and triangles correspond to sensors, subregions, and targets, respectively.



[FIG4] Performance-communication tradeoff with varying thresholds for message censoring. Inference performance is evaluated by the association error rate, which is defined as the ratio of the number of measurements that are assigned to wrong targets to the total number of measurements. The amount of communication is defined as the number of messages sent by each node on average. Max–product (cyan) and TRMP (red) are plotted for comparison.

and the TRMP algorithms are very small, even though the amount of communication is dramatically reduced. This shows that censored message passing can provide significant communication savings together with near-optimal performance. In addition, we have found examples in which this algorithm yields *better* performance than one without message censoring. We conjecture this is related to the so-called "rumor propagation'' behavior of algorithms such as sum-product in which the repeated propagation of messages around loops leads to incorrect corroboration of hypotheses; by censoring messages, this corroboration is attenuated. The communication-fusion performance tradeoff is examined more thoroughly in the next section.

CAN WE DEVELOP ALGORITHMS THAT USE MORE MEMORY AND PERFORM MORE LOCAL COMPUTATION AND THAT AS A RESULT REDUCE THE NUMBER OF MESSAGES THAT NEED TO BE SENT?

TRADING OFF ACCURACY FOR BITS IN PARTICLE-BASED MESSAGING

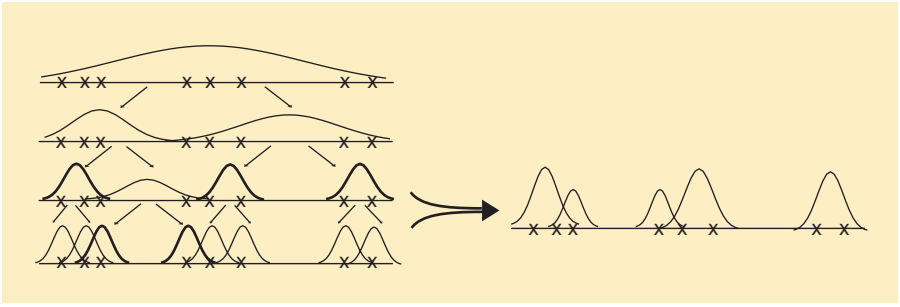
The NBP approach to inference in graphical models involves particle-based representations of messages sent from node to node. In the context of sensor networks this raises two questions: i) how does one efficiently transmit such messages, and ii) how many particles does one need; i.e., how accurate a representation of the true, continuous message is required and what transmitted particles provide that level of accuracy?

These questions can be posed as follows. We are trying to transmit a probability distribution $q(x)$ from one node to another, where the representation is specified by a set of particles $\{x_i\}$ that can be viewed as a set of independent, identically distributed samples from that distribution. While this seems like a standard problem in transmission of information (see, for example, [10]), an important distinction here is neither the order in which samples are transmitted nor the accuracy of the transmission of those *individual* particles is important, as we are ultimately interested in the accuracy with which the receiver can reconstruct $q(x)$. The flexibility in ordering samples immediately suggests protocols for transmission that can reduce the required numbers of bits. For example, for scalar particles, one can order the particles from smallest to largest and encode only the differences $x_{i+1} - x_i$. A significant savings in bits can thus be achieved [15].

Alternatively, either for scalar or vector variables, we can employ the KD tree structure used in NBP to provide a multiresolution protocol, conceptually illustrated in Figure 5. The idea of the KD tree is to group particles in a binary tree structure, starting from a root node in which all of the particles are grouped together and refining the clustering of points at each level of the tree by splitting each set into two subsets. By creating simple density approximations at each node of the KD tree, such as a Gaussian with mean and variances obtained from the samples associated with that node, we create a multiresolution representation of the distribution. This representation provides a direct means of trading off message accuracy with total bits transmitted. In particular, any cut through this tree corresponds to an approximation of the full, finest scale distribution, $q(x)$. The tree structure allows us to easily to estimate the K-L divergence between $q(x)$ and any such approximation. Furthermore, the transmit protocol can take advantage of the structure between the means of the children and their parent in the tree and the fact that the covariances of the children are typically smaller than that of the parent, to efficiently encode these means and covariances. Using a simple predictive encoder which captures this information, we can readily compute the bit cost of transmitting any particular approximation.

This structure allows one to adapt message transmission in a variety of ways. For example, specifying a desired message accuracy leads directly to a specific cut through the tree and corresponding message approximation, which then in turn leads to a specific protocol for efficient transmission of that approximation. Alternatively one can specify an upper bound on bits to be communicated and then determine the most accurate approximation whose transmission satisfies that bound.

Figure 6 illustrates the type of result that such an approach yields for the sensor localization problem, in this case for a network consisting of 25 sensors. Figure 6(a) depicts the tradeoff between message approximation error (as measured by the KLD) and localization error (where, of course, there is a nonzero irreducible error even if messages are sent exactly). A curve such as this allows the system designer to determine the threshold on message approximation error to be used to achieve an acceptable level of localization performance. This threshold, then, in turn is used as the basis for



[FIG5] Approximating a message (or density estimate) $q(x)$ using a hierarchical, KD-tree structure. The same hierarchical structure can also be used to encode the approximation, providing a quantified tradeoff between message bits and approximation error.

the adaptive message transmission procedure just described. Figure 6(b) illustrates how the entropy in successive messages behaves as localization iterations proceed. Initially, distributions and messages may be multimodal; however, as sum-product iterations proceed and individual sensor locations come into focus, this multimodality disappears. As a result, an adaptive algorithm as we have described may start with the need to transmit multiple particles but over time will use coarser, single-mode distributions, thereby saving bits. Combining this with message censoring, as described previously, provides a communication resource-sensitive solution to the sensor localization problem that trades off total resources used for message errors incurred.

THE EFFECTS OF MESSAGE APPROXIMATION

The methods described in the preceding section deal explicitly with the issue of conserving communication resources by incurring message approximations and errors due to censoring or to particle-based approximation. Of course, what one really would like is to relate the expenditure of communication resources to the accuracy of the fused estimates produced as a result of having used approximate messages. The missing piece required to yield this tradeoff is to relate message error size to the ultimate errors in the resulting fused outputs. Ihler et al. [14] examine this relationship in detail. Additionally, Xiao et al. [37] analyze the effect of quantized sensor measurements (messages in their context) on estimation performance.

The analysis in [14] involves the use of two alternate metrics to quantify the difference between an exact and an approximate message. One of these is the KLD; the other is a measure of the dynamic range in this difference. In particular, the dynamic range is

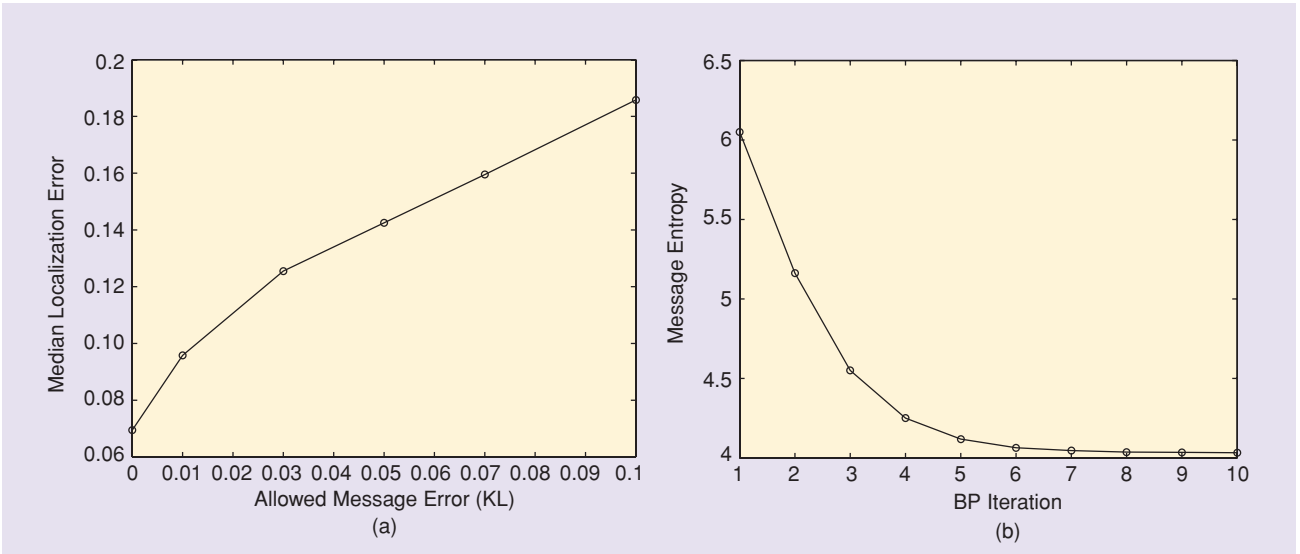
$$d\left(M_{ts}, \hat{M}_{ts}\right) = \sup_{x,x'} \left(\frac{M_{ts}(x)}{\hat{M}_{ts}(x)} \frac{\hat{M}_{ts}(x')}{M_{ts}(x')} \right)^{1/2}, \tag{10}$$

which can be shown to be equivalent [14], in the log-domain, to

$$\log d\left(M_{ts}, \hat{M}_{ts}\right) = \inf_{\alpha} \sup_{x} |\log \alpha + \log M_{ts}(x) - \log \hat{M}_{ts}(x)|. \tag{11}$$

The measure $d(\cdot)$ has several very important properties. First, as with KLD, it is insensitive to the (irrelevant) scaling of entire messages. Second, the sup-norm on measurement error in the log-messages is bounded below by $\log d(\cdot)$ and above by $2\log d(\cdot)$, enabling one to compute simple bounds on the effects of message approximation. Most importantly, $d(\cdot)$ satisfies conditions that bound how this error propagates through the two steps in message generation contained in (5). Specifically, $\log d(\cdot)$ satisfies a subadditivity condition with respect to the "product" operation in the sum-product algorithm. Furthermore, one can compute a bound on the mixing that results from the "sum" part of message generation. If there is sufficient mixing associated with this transition, the error is attenuated—the sum operation acts as a contraction. It is straightforward to specify a bound on how the log of dynamic range contracts. A particular measure of the strength of a potential ψ_{ts} that turns out to be analytically convenient (i.e., one can derive bounds) is

$$S(\psi_{ts}) = \sup_{a,b,c,d} \sup_{\psi_s, \psi_t} \frac{\psi_{ts}(a,b)}{\psi_t(a)\psi_s(b)} \frac{\psi_t(c)\psi_s(d)}{\psi_{ts}(c,d)} \tag{12}$$



[FIG6] Illustrating the basic ideas behind the adaptive message approximation and transmission procedure for network sensor localization. (a) depicts localization error as a function of message approximation error (in terms of the KLD). Such a curve allows the designer to set an approximation threshold as a function of the target localization accuracy desired; (b) illustrates the behavior of message entropy as the sum-product localization iterations proceeds. As sensors become better localized, the message entropy decreases. Using whatever approximation threshold that has been specified, our method automatically adapts to this by sending multiple Gaussian components early on (to capture multimodality and higher entropy) but then sending simple and eventually single Gaussian components as location ambiguities are resolved over time.

and if one denotes the product of incoming messages and local potential as M, one may show that

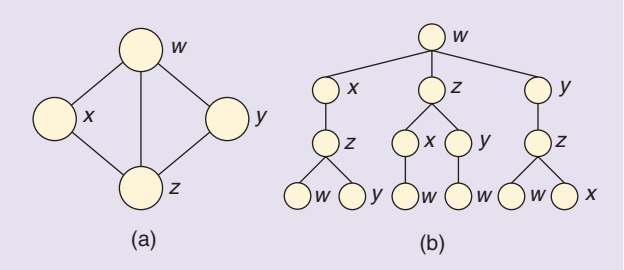
$$d\left(\sum \psi_{ts} M,\ \sum \psi_{ts} \hat{M}\right) \leq \frac{S(\psi_{ts}) d(M,\ \hat{M}) + 1}{S(\psi_{ts}) + d(M,\ \hat{M})}. \qquad (13)$$

These two conditions provide the basis for bounding how errors propagate through multiple iterations of the sum-product algorithm. This propagation is most easily visualized through a computation tree, found by "unwrapping" a loopy graph into a tree as shown in Figure 7. Combining the subadditivity and contraction bounds, it is then possible to compute a bound on the log of the dynamic range after any number of iterations of sum-product. Interestingly this procedure also provides important new results for sum-product without message approximations, namely the best conditions known to date for algorithm convergence and bounds on the distance between fixed points of the algorithm [14].

A WELL-RECOGNIZED PROBLEM FOR MANY SENSOR NETWORK APPLICATIONS IS THAT OF SENSOR LOCALIZATION.

Since the analysis yields bounds, there are cases in which the results it provides can be conservative. As an alternative, we have also developed an approximation (not a bound) built on the same principle used in roundoff noise analysis of digital filters, namely that at each point at which messages are approximated, the log dynamic range error can be modeled as a white noise perturbation. This then leads to easily computed approximations to the variance in this measure.

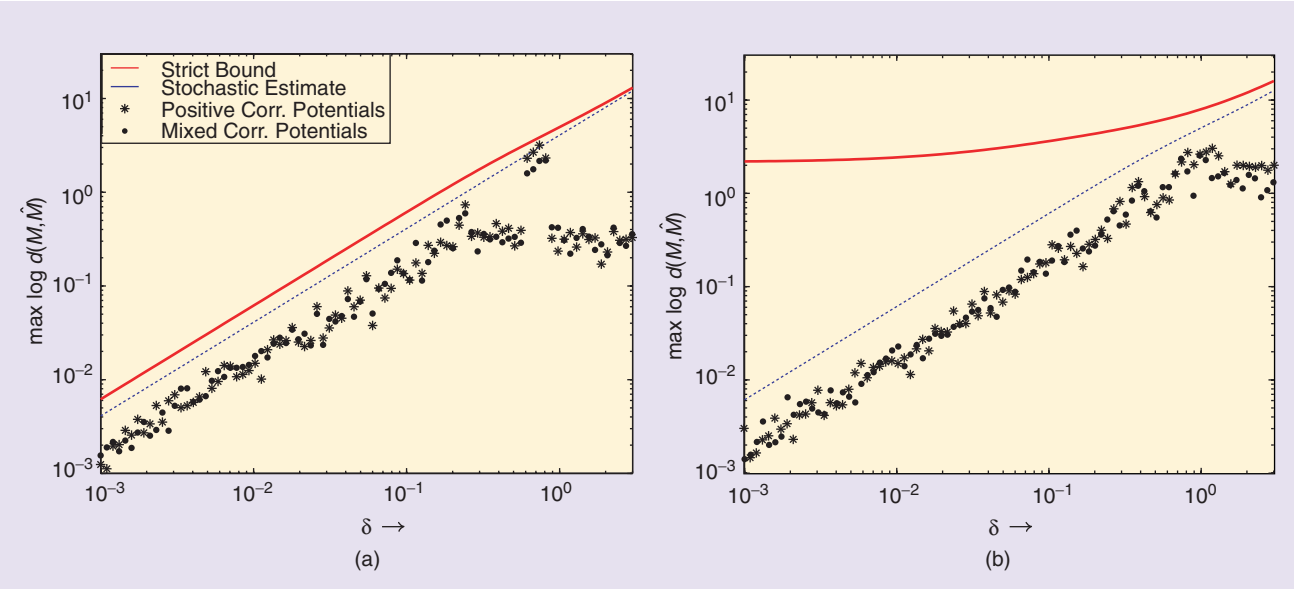
Figure 8 shows two illustrations of these analyses for a case in which errors are introduced by message quantization, one for relatively weak edge potentials (ones for which convergence of sum-product is guaranteed) and one for stronger edge potentials (typical of cases in which one would expect our strict bounds to be conservative). These curves depict resulting error measures as functions of the quantization error introduced at each stage. Note that even in the difficult case of strong potentials the approximation provides an accurate (if somewhat conservative) estimate of resulting error. Furthermore, taken together with the relationship between communication costs and message approximation error, we now have the elements needed for a complete "audit trail" from bits used to message errors incurred and finally to resulting errors in the desired fused outputs.



[FIG7] A (a) loopy graphical model and (b) its associated 3-level computation tree with root node *w*. The messages received at node *w* after three iterations of loopy sum-product in the original graph is equivalent to the messages received at the root node of the tree after three (or more) iterations.

OPEN QUESTIONS AND FUTURE DIRECTIONS

We have described a line of research that provides a bridge from the rich field of graphical models to the emerging field of *data*



[FIG8] Maximum errors incurred as a function of the quantization error δ in a quantized implementation of sum–product. The scatterplot indicates the maximum error measured in the graph for each of 200 Monte Carlo runs, as compared to our upper bound (solid) and approximation (dashed). (a) $\log S(\psi) = .25$ and (b) $\log S(\psi) = 1$.

fusion for sensor networks. We have i) demonstrated how two prototypical sensor network problems can be mapped to problems of inference on graphical models, ii) discussed issues that arise when one attempts to solve these inference problems, iii) developed a particle-based method for message passing as well as communications-sensitive messaging strategies; and iv) presented an approach to analyzing the impact of messaging errors (due to approximation, communication limitations, or simple message dropouts) on the resulting fusion results. While these results and lines of inquiry provide substantive results of importance for sensor network applications, they are far from the complete story in moving from the confines of graphical model inference to the domain of sensor networks. Indeed there are a number of questions, each of which is the subject of current research and raises new issues not seen in the graphical models literature but that have strong conceptual ties to that literature and that offer considerable promise for the future.

IN A SENSOR NETWORK, IT IS ADVANTAGEOUS TO ORGANIZE THE REPRESENTATION AROUND SENSORS RATHER THAN TARGETS.

One such issue deals with how one maps inference responsibility to sensor nodes. As we have already seen in the data-association example, there is considerable flexibility in how one maps a sensor network fusion problem to a graphical model. The ones that we have chosen here have some nodes that are explicitly identified with individual sensor nodes and other hidden or "virtual" nodes correspond to groups of objects or regions in space. Where do the inference computations associated with these parts of the model reside? Moreover, in a dynamic problem, as objects move, one would expect that these hidden variables might "migrate," which raises a new resource allocation problem, namely that sensor handoff, i.e., of deciding when and to whom the responsibility for such nodes should be handed off. Such a handoff involves communication—e.g., of particle-based representations of the object location and velocity—and the analysis in this article provides tools to quantify the communication cost/accuracy tradeoff of different handoff strategies. Having such costs allows one to develop dynamic strategies not only for querying sensors (since taking measurements and transmitting messages use power) but also for handing off fusion responsibility [36].

In addition, there is considerable motivation to consider alternative fusion algorithms to the ones based on sum-product or max-product. For example, one can imagine operational structures in which "seed" nodes initiate messaging, propagating information radially outward, fusing information around these radially expanding regions as they meet, and then propagating information back inward toward the seed nodes. Such an algorithmic structure allows great flexibility (e.g., one can imagine allowing any sensor to act as a seed if it measures something of interest) and also leads to new algorithms with great promise for inference on graphical models more generally. We refer to reader to [17] for a first treatment of this approach. Also, the computation tree interpretation of the sum-product algorithm allows one to clearly see the computations that sum-product fails to make that a truly optimal algorithm would—computations that in essence take into account the dependencies between messages that sum-product neglects [16]. This suggests another line of research that focuses on one of the significant differences between standard graphical model inference problems and sensor networks. In particular, when viewed as a sensor network fusion algorithm sum-product has the property that it makes very little use of local node memory and computational power (all that is remembered from step to step are the most recent messages, and all that are computed are essentially the sum-product computations). Can we develop algorithms that use more memory and perform more local computation and that as a result reduce the number of messages that need to be sent? Several ideas along these lines are currently under investigation. Also, a standard feature in wireless networks is the inclusion of header bits that provide information on the path a message has taken from initiator to receiver. Can we take advantage of such header bits to capture dependencies between messages so that they can then be used to fuse messages in a manner that is not as naïve as assuming conditional independence? Of course using such header bits for what we term informational pedigree means that there are fewer bits available for the actual message, so that the message quantization error will be larger. How does the error incurred by such an increased quantization error compare to the additional fusion accuracy provided by providing these pedigree bits? Current research building on what we have presented here, is addressing this question.

Finally, there is the introduction of ideas from team theory and decentralized decision making. In particular, while we have described message approximation and censoring methods, they are all based solely on the perspective of the sending node without explicit regard for the objectives (e.g., in terms of desired fusion results) of the receiver. Taking such objectives into account makes the problem of network fusion system design one of team decision making, a notoriously difficult problem if one seeks truly optimal solutions. In [3], the authors describe a set of results that deal with a widely studied problem of distributed detection in which the sensors in a network must provide limited numbers of bits, through a possibly uncertain channel to a "fusion center" which is then responsible for making an overall decision. As discussed in [3], the fact that these sensors act as a "team" has a significant effect on the signal processing strategy employed by each individual sensor.

In [19] we consider a problem in which both the sensing and the decision making are distributed throughout the network—i.e., each node acts as a provider of information to other nodes, a receiver of information from others, and potentially also a decision maker for part of the overall inference

problem. We examine all of this in the context of a loop-free directed network and in the process expose several key questions that we believe will fuel considerable research. The first is that in the presence of communication limitations, and even if the fusion responsibilities of each node have been specified (e.g., we have specified to whom responsibility for inference on hidden variables will fall), the nodes must organize to specify what we term a *fusion protocol*, namely not only the rules by which a sending node decides when and what bits to transmit, based on the impact of those bits on the fusion decisions made by other nodes subsequently affected by those transmitted bits, but also the specification of enough information at the receiving node to know how to interpret the received information. Roughly speaking, the specification of such a protocol corresponds to the design of the free parts of a composite graphical model, involving i) the variables the network senses and about which we wish to make inferences; ii) the variables that are transmitted from node to node; and iii) the decisions made by each node and their relationship to the external phenomenon, i.e., the overall objective function represented in distributed fashion. It is only the second of these that is generally at our control (assuming that the fusion responsibilities in iii) have been determined). Problems of this type can capture very subtle effects, ones that can be important but that also make clear the intractability of full optimization and the need for new methods of approximate solution, much as algorithms such as sum-product represent principled suboptimal algorithms that are in fact optimal for particularly nice (namely loop-free) graphs. For example, suppose that each node in a network has the possibility either not to communicate with its neighbor or to send 1 b, at a cost of some power. Once a network organizes—i.e., once the neighbor knows something about the rule that the transmitting node will be using, then *no news is indeed news*, i.e., not receiving a bit from its neighbor tells a node something! This idea has already shown benefit in the sensor localization problem.

THERE IS CONSIDERABLE FLEXIBILITY IN HOW ONE MAPS A SENSOR NETWORK FUSION PROBLEM TO A GRAPHICAL MODEL.

These questions and issues form just the tip of the iceberg and make it clear that sensor network problems are truly ones of information science in the large: they are not problems that are nicely separated into signal processing, communications, and optimization problems; they are all of those at the same time. We are all in for a long, enjoyable, and educational ride.

AUTHORS

Mujdat Çetin (mcetin@sabanciuniv.edu) received the Ph.D. degree from Boston University in 2001 in electrical engineering. Since 2001, he has been a member of the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, where he is a research scientist. Since September 2005 he has also been an assistant professor at Sabancı University, İstanbul, Turkey. His research interests include statistical signal and image processing, inverse problems, computer vision, data fusion, wireless sensor networks, biomedical information processing, radar imaging, and sensor array signal processing. He is a Member of the IEEE.

Lei Chen (lchen@mit.edu) received the B.S. degree from Tsinghua University, Beijing, in 1997 and the M.S. degree from Chinese Academy of Sciences, Beijing, in 2000. Since then he has been a Ph.D. student at the Massachusetts Institute of Technology. He is currently an RA of the Stochastic Systems Group, LIDS. His research interest includes statistical inference and multiple target tracking and sensor networks.

John W. Fisher, III (fisher@csail.mit.edu) received a Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, in 1997. He is currently a principal research scientist in the Computer Science and Artificial Intelligence Laboratory and affiliated with the Laboratory for Information and Decision Systems, both at the Massachusetts Institute of Technology. His research interests include information theoretic approaches to signal processing, multimodal data fusion, machine learning, and sensor networks. He is a Member of the IEEE.

Alexander T. Ihler (ihler@ics.uci.edu) received his B.S. degree from the California Institute of Technology in 1998 and S.M. and Ph.D. degrees from the Massachusetts Institute of Technology in 2000 and 2005 as a member of the Stochastic Systems Group. He is currently a Postdoctoral Scholar at the University of California, Irvine. His research interests include statistical signal processing, machine learning, nonparametric statistics, distributed systems, and sensor networks. He is a Member of the IEEE.

Randolph L. Moses (moses.2@osu.edu) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Virginia Polytechnic Institute and State University in 1979, 1980, and 1984, respectively. Since 1985 he has been with the Department of Electrical Engineering, Ohio State University, where he is currently a professor. He was a summer faculty research fellow at Rome Air Development Center, New York and a NATO postdoctoral fellow at Eindhoven University of Technology, The Netherlands. He spent sabbaticals at Uppsala University, Sweden, the Air Force Research Laboratory in Ohio, and the Massachusetts Institute of Technology. He was an associate editor for *IEEE Transactions on Signal Processing* and served on the IEEE Signal Processing Society Technical Committee on Statistical Signal and Array Processing. He is coauthor with P. Stoica of *Spectral Analysis of Signals* (Prentice Hall, 2005). He is a member of Eta Kappa Nu, Tau Beta Pi, Phi Kappa Phi, and Sigma Xi. His research interests are in stochastic signal processing, and include parametric estimation, array signal processing, sensor networks, and radar imaging. He is a Senior Member of the IEEE.

Martin J. Wainwright (wainwrig@EECS.berkeley.edu) is an assistant professor at University of California at Berkeley, with a joint appointment between the Department of Statistics and the Department of Electrical Engineering and Computer Sciences. He received his Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology. His research interests include statistical signal and image processing, coding and information theory, and machine learning. He has been awarded an Alfred P. Sloan Foundation Fellowship, the George M. Sprowls Prize for his dissertation research, a Natural Sciences and Engineering Research Council of Canada 1967 Fellowship, and several outstanding conference paper awards.

Alan S. Willsky (willsky@mit.edu) joined M.I.T. in 1973 and is the Edwin Sibley Webster Professor of Electrical Engineering. He was a founder of Alphatech, Inc. and chief scientific consultant, a role in which he continues at BAE Systems Advanced Information Technologies. From 1998–2002 he served on the U.S. Air Force Scientific Advisory Board. He has received several awards including the 1975 American Automatic Control Council Donald P. Eckman Award, the 1979 ASCE Alfred Noble Prize, the 1980 IEEE Browder J. Thompson Memorial Award, the IEEE Control Systems Society Distinguished Member Award in 1988, the 2004 IEEE Donald G. Fink Prize Paper Award, and Doctorat Honoris Causa from Universitè de Rennes in 2005. He has delivered numerous keynote addresses and is coauthor of the book *Signals and Systems*. His research interests are in the development and application of advanced methods of estimation and statistical signal and image processing. He is a Fellow of the IEEE.

REFERENCES

[1] S.A. Aldosari and J.M.F. Moura, "Decision making in sensor networks: Topology issues," submitted for publication.

[2] P. Brémaud, *Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues*. New York: Springer-Verlag, 1991.

[3] B. Chen, L. Tong, and P. Varshney, "Channel aware distributed detection in wireless sensor networks," *IEEE Signal Processing Mag.*, vol. 23, no. 4, pp. 16–25, 2006.

[4] L. Chen, M. Çetin, and A.S. Willsky, "Distributed data association for multi–target tracking in sensor networks," in *Proc. Information Fusion 2005*, Philadelphia, PA, 2005 [CD-ROM].

[5] L. Chen, M.J. Wainwright, M. Cetin, and A.S. Willsky, "Data association based on optimization in graphical models with applications to sensor networks," *Math. Comput. Model. (special issue on optimization and control for military applications)*, vol. 43, no. 9-10, pp. 1114-1135, May 2006.

[6] R.G. Cowell, A.P. Dawid, S.L. Lauritzen, and D.J. Spiegelhalter, "Probabilistic Networks and Expert Systems," in *Statistics for Engineering and Information Science*. New York: Springer-Verlag, 1999.

[7] E.B. Ermis, M. Alanyali, and V. Saligrama, "Search and discovery in an uncertain networked world," *IEEE Signal Processing Mag.*, vol. 23, no. 4, pp. 107–118, 2006.

[8] J. Feldman, M.J. Wainwright, and D.R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inform. Theory*, vol. 51, pp. 954–972, Mar. 2005.

[9] W.T. Freeman and Y. Weiss, "On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 736–744, 2001.

[10] M. Gastpar, M. Vetterli, and P.L. Dragotti, "Sensing reality and communicating bits: A dangerous liaison," *IEEE Signal Processing Mag.*, vol. 23, no. 4, pp. 70–83, 2006.

[11] A.T. Ihler, E.B. Sudderth, W.T. Freeman, and A.S. Willsky, "Efficient multiscale sampling from products of Gaussian mixtures," in *Proc. Neural Inform. Processing Syst.*, 2003, vol. 17 [CD Rom].

[12] A.T. Ihler. "Inference in sensor networks: Graphical models and particle methods," Ph.D. dissertation, MIT, June 2005.

[13] A.T. Ihler, J.W. Fisher III, R.L. Moses, and A.S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Select Areas Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

[14] A.T. Ihler, J.W. Fisher III, and A.S. Willsky, "Loopy belief propagation: Convergence and effects of message errors," *J. Mach. Learning Res.*, vol. 6, pp. 905–936, May 2005.

[15] A.T. Ihler, J.W. Fisher III, and A.S. Willsky, "Using sample-based representations under communications constraints," submitted for publication.

[16] J.K. Johnson, D.M. Malioutov, and A.S. Willsky, "Walk-sum interpretation and analysis of Gaussian belief propagation," *Adv. Neural Inform. Processing Syst.*, vol. 18, 2006.

[17] J.K. Johnson and A.S. Willsky, "A recursive model-reduction method for approximate inference in Gaussian Markov random fields," submitted for publication.

[18] V. Kolmogorov and M.J. Wainwright, "On optimality properties of tree-reweighted message-passing," in *Proc. Uncertainty Artificial Intell.*, July 2005 [CD-Rom].

[19] O.P. Kreidl and A.S. Willsky, "Inference with minimal communication: A decision-theoretic variational approach," in *Proc. Advances Neural Inform. Processing Syst.*, 2006, vol. 18 [CD-Rom].

[20] F. Kschischang, "Codes defined on graphs," *IEEE Signal Processing Mag.*, vol. 41, no. 4, pp. 118–125, Aug. 2003.

[21] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum–product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[22] T. Kurien, "Issues in the design of practical multitarget tracking algorithms. in *Multitarget-Multisensor Tracking: Advanced Applications,* vol. 1, Y. Bar-Shalom, Ed. Norwood, MA: Artech House, 1990, pp. 43–83.

[23] H.A. Loeliger. "An introduction to factor graphs," *IEEE Signal Processing Mag.*, vol. 21, no. 1, pp. 28–41, 2004.

[24] R. Moses, D. Krishnamurthy, and R. Patterson, "Self-localization for wireless networks," *Eurasip J. Appl. Signal Processing*, vol. 2003, no. 4, pp. 348–358, 2003.

[25] N. Patwari, A.O. Hero III, M. Perkins, N.S. Correal, and R.J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Processing*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

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

[27] J.B. Predd, S.R. Kulkarni, and H.V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Mag.,* vol. 23, no. 4, pp. 56–69, 2006.

[28] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[29] E.B. Sudderth, A.T. Ihler, W.T. Freeman, and A.S. Willsky, "Nonparametric belief propagation," *Comput. Vis. Pattern Recognit.*, vol. 1, pp. 609–612, 2003.

[30] S. Thrun, "Affine structure from sound," in *Advances in Neural Information Processing Systems, 18*, Y. Weiss, B. Schölkopf, and J. Platt, Ed. Cambridge, MA: MIT Press, 2006, pp. 1355–1362.

[31] M.J. Wainwright, T. Jaakkola, and A.S. Willsky, "Tree consistency and bounds on the performance of the max–product algorithm and its generalizations," *Statist. Computing*, vol. 14, pp. 143–166, Apr. 2004.

[32] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky, "Tree–based reparameterization analysis of sum–product and its generalizations," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, pp. 1120–1146, May 2003.

[33] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches." *IEEE Trans. Inform. Theory*, vol. 51, no. 11, pp. 3697–3717, Nov. 2005.

[34] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky, "A new class of upper bounds on the log partition function," *IEEE Trans. Inform. Theory*, vol. 51, no. 7, pp. 2313–2335, July 2005.

[35] M.J. Wainwright and M.I. Jordan, "A variational principle for graphical models," in *New Directions in Statistical Signal Processing,* S. Haykin, J. Principe, T. Sejnowski, and J. McWhirter, Eds. Cambridge, MA: MIT Press, 2005.

[36] J.L. Williams, J.W. Fisher III, and A.S. Willsky. "An approximate dynamic programming approach to a communication constrained sensor management problem," in *Proc. 8th Int. Conf. Information Fusion,* July 2005 [CD-Rom].

[37] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G.B. Giannakis. "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Processing Mag.*, vol. 23, no. 4, pp. 26–41, 2006.

[38] J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Constructing free energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inform. Theory*, vol. 51, no. 7, pp. 2282–2312, July 2005.

SP