Message Passing and Distributed Statistical Inference

Venkat Chandrasekaran, Jason Johnson, O. Patrick Kreidl, Dmitry Malioutov, Sujay Sanghavi, and Alan S. Willsky

Abstract—In this paper, we summarize several research directions involving message-passing algorithms and distributed fusion. The starting point is inference for graphical models, but the different constraints and objectives found in sensor networks suggest new questions. In addition to providing pointers to a variety of other parts of our research in this area, we spend most of our time on two research investigations. The first involves team-theoretic approaches to distributed decision-making and the development of message-passing algorithms for sensor network organization in order to optimize distributed team objectives. The second deals with the exploitation of so-called walk-sum analysis for distributed linear estimation. In particular under an easily checked condition for walk-summability, we can show that essentially any messaging strategy converges to the optimal distributed solution; moreover with modest use of local memory, this remains true in the presence of random message losses. We also describe how this framework leads to a tractable max-spanning tree problem for optimal adaptive message routing — i.e., for determining a spanning tree for the next wave of messages to reduce an upper bound on estimation error as quickly as possible.

In this brief paper, we summarize several directions of research related to distributed inference and message passing. These efforts are presented in much more detail in the references provided at the end of the summary. The work of our group on distributed fusion has built on the message-passing structure for inference in graphical models, where the context of sensor networks — in which the messages truly are messages adds issues not generally considered in the context of graphical models. This suggests some places of commonality and differences that have driven our research. In particular, standard message-passing algorithms, when interpreted as distributed fusion algorithms for sensor networks, have clear deficiencies. For example, such algorithms can be viewed as assuming that communication is cheap but local computation and memory are expensive — exactly the opposite of what is typically the case for power-limited sensor networks. However, the context of graphical models naturally views the fusion problem as truly distributed — in particular, there is no concept of a fusion center, as all nodes in the graph are on equal footing as far as performing local inference and in providing information to aid other nodes in performing their own local inference computations. Such algorithms can also be viewed as purely “information push” — i.e., nodes transmit bits (namely likelihood messages) without regard for what the information needs are of the nodes that will either receive these messages directly or indirectly (through subsequent messages). Very importantly, however, these algorithms do have an important aspect that differentiates sensor networks from pure wireless communication networks. In particular, messages are not simply relayed from node to node; rather messages are transmitted, fused at the receiver through local computation, and these fused products are then transmitted to other nodes. Finally, in the context of graphical models, the network over which messages are sent is precisely the same as the graph that describes the conditional dependency structure of the underlying phenomenon being sensed. However, in reality, there is another network that interacts with this, namely the fusion network — which captures both which nodes are responsible for what part of the overall inference problem and what the communication structure is for message passing connecting these nodes.

The research agenda of our group has had as its focus examining many of these issues. For example, in [8] we look at the problem of distributed data association for multi-object tracking in sensor networks. Such a problem can be mapped to one of inference on a graphical model in a variety of ways, and the one that is most natural for sensor networks — essentially a sensor-centric graph — is very different than the target-centric architecture that is most efficient when processing is performed at a centralized processor to which all data have been relayed. Moreover, the resulting graph exposes clearly the issue of assigning fusion responsibilities to sensor nodes. In particular, while some of the nodes in this graph are naturally identified with individual sensors, others are not, and thus there is both the requirement and some flexibility in deciding which sensor node takes on the inference required at such a node in the statistical graph. Finally, in this work we take a first look at message-censoring, i.e., allowing nodes to decide that they have nothing significant to say and to refrain from sending a message (the method used involves setting a threshold on the KL-distance between the last message sent from one node to another and the one that message-passing computations indicate should be sent next). Not only does this introduce very interesting adaptive dynamics in messaging, it also (at least empirically) damps out and in many cases eliminates instabilities in message-passing on loopy graphs.

This and other work on particle-based messaging (when the messages to be sent are likelihood functions for continuous-valued variables) motivated the investigation in [9] on how errors in messages — due, for example to censoring or quantization — translate into errors in resulting
fusion results. When combined with the results in [10] on analyzing the quantization of messages (which provides a relationship between bits and messages errors) we have an audit trail from bits to message errors to resulting fusion errors. The work in [10] emphasizes that the transmission problem of interest here is one in which our criterion is that of accurately communicating a probability distribution (or likelihood function). For example, if we have represented that distribution as a set of particles, we aren’t really interested in accurately transmitting these particles but rather in the accuracy of the particle-based density that results at the receiver from whatever it is that we do transmit. This work, thus, provides one modest examination of something that is central to sensor networks, namely that the problems of managing the use of resources (in this case bits) and that of inference (in this case accuracy of marginal distributions) are intimately intertwined. Another issue emphasizing the fact that resource management and fusion are interrelated is that of deciding which node is responsible for which part of the fusion problem. A special but important example of this is explored in [11] in which the problem is one of tracking a target using a sensor network in which there are costs for (a) a sensor taking a measurement; (b) for it communicating its observation to a fusion node; and (c) for changing fusion nodes as the target moves. The first and second of these, which have been studied by others, capture the fact that sensors close to the target have better SNR and sensors close to the fusion node have lower power requirements for communication. The third of these, however, is new and captures the problem of sensor handoff, in which we must communicate the current pdf for the target from one fusion node to its successor. Using the results in [9], [10], we can specify a cost for the transmission of such a pdf. Including all of these leads to communication and sensor tasking strategies with greatly reduced communication cost (and no loss of tracking accuracy) as compared to simple proximity-based strategies for both sensing and handoff.

Our presentation here focuses most of its attention on two more recent components of our research program. The first of these, described in more detail in [1], [2] investigates a team-theoretic framework for distributed fusion. In this formulation, a set of sensors, with observations \( y = (y_1, \ldots, y_N) \) are connected in a network with a particular messaging and decision structure. For example, the structure studied in depth in [1], [2] is one in which there is a directed acyclic graph (DAG) used for communication. Specifically, each node may receive information from noisy channels (where we focus for the most part on erasure channels) connecting that node to each of its parents (in the DAG) and uses these together with its local observation to (a) decide what, if anything, to communicate to each of its children; and (b) what local inference decision it must make if it is a node with such an inference responsibility (the set of nodes with such responsibilities is assumed to be fixed and known and could be every node only a small subset of the nodes). The variable being sensed by these sensors, is denoted by \( x \), with most of the analysis focusing on the case in which \( x = (x_1, \ldots, x_N) \), and where we make the usual conditional independence assumption on the measurements (e.g., conditioned on \( x_i \) is independent of the other components of \( x \) and \( y \). The cost function to be optimized is one that involves two types of terms, one of which is the expected cost of the collective inference decisions (e.g., the cost of discrepancies between estimates at nodes with decision-making responsibility and the corresponding values of \( x_i \)) and the other is the expected cost of communication (where we allow each node the possibility of deciding not to communicate to one or more of its children). We focus on the case in which these costs are additive — i.e., the sum of costs on errors between local variables and the decisions made at local decision nodes and the sum of communications costs across the network.

The methodology we develop is based on person-by-person optimality conditions, i.e., on adjusting each local decision rule (on transmitting bits and on any local decision) to optimize performance given the decision rules of other nodes are specified and fixed. This leads to the conclusion that each node must implement threshold tests for communication and for final decision but where the parameters of these rules involve the solution of a set of fixed point equations that have a natural implementation in terms of message-passing. In particular each node needs to receive messages (i) from its parents that in essence provide a “measurement model” for how that node should interpret the message (or the absence of such a message) sent to it by each parent, thus representing the “push” part of fusion; and (ii) messages from its children providing information on the influence that messages sent to its children will have on costs incurred by all downstream nodes (the “pull” part of fusion). This message-passing structure represents the communication overhead required to establish a fusion protocol — i.e., for the network to organize itself to optimize team performance.

Exploring the implications of this formulation and the resulting optimized strategy exposes some interesting and important features. The first is that a coordination strategy that includes the possibility of nodes remaining silent can use this possibility to advantage — in essence each node has an extra “half-bit” of information that it can signal by not sending a message. This is useful even if there is no cost for communication and/or if the erasure probability is nonzero. The second is that while the implementation of this procedure requires local marginals and joints associated with the communication network structure, there is no requirement that the actual graphical model of \( x \) be commensurate with the communication network. This opens up interesting questions of choosing messaging pathways given that particular nodes are designated as decision-making nodes — e.g., it is important that we choose structures that emphasize information flow to these decision-making nodes. Some heuristics in terms of choosing spanning trees that have this property are used in examples in our work, but more fundamental approaches to sensor network formation are required. Preliminary work along these lines is reported in [7].

The second direction on which we focus our presentation
is our recent work [3] on distributed fusion for Gaussian estimation using so-called walk-sum analysis [4]. Gaussian graphical models are usually given in information form, i.e.,

\[ p(\mathbf{x}) \propto \exp\left\{ -\frac{1}{2} \mathbf{x}^T J \mathbf{x} + \mathbf{b}^T \mathbf{x} \right\} \tag{1} \]

where \( J \) is the information matrix, i.e., the inverse of the covariance matrix, \( P \), and the mean \( \mu \) is obtained by solving the equations

\[ J \mu = \mathbf{b} \tag{2} \]

The sparsity structure of \( J \) is coincident with that of the graph for the graphical model, i.e., \( J_{ij} \) is nonzero if and only if there is an edge between nodes \( i \) and \( j \). Indeed, for simplicity we’ll assume that we have performed a simple scaling of variables so that the diagonal entries of \( J \) are all equal to 1. In this case we can write \( J = (I - R) \) where \( R \) has zero diagonal and its \( ij \)-element is the conditional partial correlation coefficient between \( x_i \) and \( x_j \) conditioned on all other components of \( \mathbf{x} \). As discussed in [3] and [4] walk-sum analysis begins with the examination of the series expansion

\[ P = (I - R)^{-1} = I + R + R^2 + \cdots \tag{3} \]

together with the observation that the \( ij \)-element of \( R^k \) is the sum of all length-\( k \) weighted walks from node \( i \) to node \( j \), where each such walk represents a length-\( k \) path in the graph starting at node \( i \) and ending at node \( j \), and where the weight of this walk is the product of the elements of \( R \) corresponding to each edge traversed on the walk. While this power series expansion might or might not converge for a positive definite covariance matrix, a very important class of models are those that are walk-summable, namely for which this series is absolutely convergent, so that walks can be added up in any order. The computation of elements of the covariance matrix or the solution of (2) require calculating such walks (or in the case of (2) the calculation of walks weighted by components of \( \mathbf{b} \)). Note also that walk-summability is easily checked as it amounts to checking if the spectral radius of \( R \) is less than one, where \( R \) is the matrix in which each element of \( R \) is replaced by its absolute value.

One use of walk-summability is the analysis of belief propagation algorithms, and we refer the reader to [4] for that development. We focus here instead on its implications for a broad class of iterative algorithms for the solution of (2) (and hence also for the efficient computation of unbiased and accurate estimates of the diagonal elements of \( P \) using the technique described in [5]). Specifically, for general, loopy graphs solution of (2) can be quite difficult, but one can readily define a rich class of algorithms in which we imaging cutting some of the edges of the graph so as to expose a tractable computation. This corresponds to expressing

\[ J = J_t - K_t \tag{4} \]

where \( J_t \) represents a tractable subgraph portion of \( J \) and where the cutting matrix, \( K_t \), contains the elements of \( J \) — and hence the edges of the original graph — that have been cut. Obviously we can then write (2) as

\[ J_t \mu = \mathbf{b} + K_t \mu \tag{5} \]

which suggests an iterative method for solving (2). Indeed this can be further generalized by using different cutting matrices in successive iterations. Such algorithms — known as Embedded Subgraph Algorithms are developed in [12], in which empirical evidence suggests that such algorithms can converge exceedingly quickly if the sequence of embedded subgraphs is chosen well. However, only relatively restrictive sufficient conditions for convergence are provided.

The results in [3] provide a much more complete story not only for embedded subgraph algorithms but for a much larger set of algorithms that include Gauss–Seidel iterations (in which only a subset of the variables are updated at each iteration) and, more general hybrids of the two in which a subset is updated using one or more embedded subgraph steps within the subgraph containing the variables being updated. The key to this analysis is keeping track of the sets of walks computed as iterations proceed in any such algorithm. For example, consider the iterative application of (5) in which at each iteration, the “\( \mu \)” on the right-hand side is replaced by the solution at the previous iteration and in which a different embedded subgraph is possibly chosen at each iteration. In this case, solving (5) corresponds to summing all of the walks (of arbitrary length) in this embedded subgraph, while multiplying by a cutting matrix corresponds to adding walks in which there is an additional a single “hop” across a cut edge. Analogous (although more complicated) descriptions of successive walk-sum sets can be constructed for Gauss–Seidel and hybrid algorithms. What one can then easily show is that, as any such algorithm proceeds, the set of weighted walks that have been accumulated grows monotonically. Moreover, under easily checked and very simple graphical conditions (that always hold for embedded subgraph algorithms), we can easily show that any weighted walk is eventually included in the set of walks that has been accumulated. As a result, using purely graphical means we have that for walk-summable models, any such procedure yields a convergent algorithm that produces the desired optimal answers.

These basic results lead to two extensions in [3] of considerable interest for sensor networks. The first is a generalization of the results in [13] (which are for the case in which the same embedded subgraph is used at every iteration). In particular, in the context of sensor networks, walks in a graph correspond to sequences of messages in the network. Suppose, however, that some of these messages are erased. In this case, by employing a very simple procedure in which nodes remember the last messages received as well as a messaging protocol that ensures that we visit every node and edge continually, we can show that these algorithms still converge to the optimal estimates for walk-summable algorithms as long as message erasures occur independently from iteration to iteration. Once again, the proof is purely graphical in nature.
The final major result in [3] involves the adaptive choice of embedded subgraph (or Gauss-Seidel subset) to drive the error to zero as quickly as possible and hence to minimize total messaging. For example, consider the case in which we wish to choose an embedded spanning tree of the graph for the next embedded subgraph iteration. Since message-based Gaussian inference corresponds to accumulating weighted walks, maximizing the total weighted sum of walks over the spanning tree to be used at the next iteration is a natural criterion for choosing that spanning tree. However, this is an NP-hard optimization problem. Instead, we look at each individual edge and the equation errors at the pair of nodes connected by that edge, and we perform the very simple computation of determining the total weighted sum of walks on this simple one-edge graph. Summing these weights over any spanning tree provides a looser bound on the reduction in equation (and hence estimate) error if that subtree is used. Moreover choosing the tree that minimizes this bound is a simple max-weight spanning tree problem, the solution of which yields a fully adaptive algorithm for optimal distributed estimation with rapid (and hence power efficient) convergence.

We refer the reader to [3] for adaptive procedures for other iterative message-passing algorithms — and hence for other fusion protocols — such as Gauss–Seidel and hybrid algorithms. We also remark that the weights computed for each edge provide a direct measure of the informational value of communicating across that edge. Such weights can then also be used in the max-weight B-matching message-passing algorithms of [7] for adaptive sensor network formation.

Finally, we also refer the reader to a very different set of near-optimal inference algorithms developed in [6]. These algorithms involve processing data radically outward from a set of “seed” nodes, eliminating variables interior to “cavities” around these seeds in successive radially expanding stages. Of course such node elimination leads to “fill,” i.e., to additional edges between variables not connected in the original graph but which were each connected to nodes that have been eliminated. Before such “thickening” of the graph around the boundary of such a cavity becomes too severe, we apply “thinning” — essentially information projections onto thinned exponential families — and then continue with radially outward variable elimination. At some point cavities originating from different seeds will meet, are merged and thinned, with continuing outward expansion, merging and thinning until the entire region is included in a single cavity, at which point an inward propagation step passes information (through so-called Markov blankets) back toward the seeds. Except for thinning, this yields fully optimal estimates at every node in the graph, and the errors induced by thinning can be controlled in a manner that yields globally near-optimal performance for distributed phenomena with sufficient “mixing”, i.e., in which the impact of a variable at some point in the graph has exponentially decaying impact on estimates far removed from that node. This structure, which was developed for large-scale Markov random fields, also suggests yet-to-be explored fusion algorithms for sensor networks in which seed nodes initiate messaging and fusion by propagating information out to their radial neighbors (possibly waking them from slumber), and in which messaging occurs around the perimeters of cavities as they expand and as they subsequently propagate information back inward.

References


