CUCS-25-82

TOWARDS DISTRIBUTED SENSORS NETWORKS

AN EXTENDED ABSTRACT

Y. YEMINI & A. LAZAR
COLUMBIA UNIVERSITY

1982 Conference on Information Sciences and Systems; Princeton University March, 1982.

This research was supported in part by the Defense Advanced Research Projects Agency. Project N00039-82-C-0427.

TOWARDS DISTRIBUTED SENSORS NETWORKS
AN EXTENDED ABSTRACT

Y. YEMINI & A. LAZAR
COLUMBIA UNIVERSITY

1. INTRODUCTION

Consider a network consisting of a multitude of sensors distributed over some region. When a target crosses the region, observations will be taken by many sensors. These observations at the network nodes are noisy, partial, and have a varying degree of credibility. Therefore although each individual sensor may not have enough information the total system may possess enough information to track the targets. However the total information is distributed in the whole system and cannot be collected or processed by any single node in real-time due to limited communication and processing resources. Nor should it be centrally processed due to reliability and vulnerability considerations.

The Distributed Sensor Networks (DSN) problem consists of building a sensor-communication-processing architecture to fuse noisy and partial observations from a multitude of geographically distributed sensors of varying credibilities. The program involves challenging problems of pushing the boundaries of signal processing technology, artificial intelligence and computer communication technology and integrating these disciplines. The objective of this paper is to present some of the interesting open problem areas arising in DSN.

1.1. FROM SIGNAL PROCESSING TO ARTIFICIAL INTELLIGENCE

Can signal processing technology solve the DSN problem? At the signal level, the information available to the system is simply too massive to be communicated. Therefore, one can expect that nodes will extract some parameters from the signal and communicate those parameters to each other. If this information communicated to nodes may be considered to be observations at a different level of abstraction, the process of information extraction and communication can proceed recursively. That is, the system function can be viewed as a distributed pipeline which narrows the massive signal information entering at one end of the pipeline to a small message at the other hand.

This process of incremental feature extraction proceeds from analog information at one end to symbolic information at the other end of the pipe. At one end we have estimation-filtering technology to extract information from analog observations and at the other end we have artificial intelligence technology to extract hypotheses from symbolic information. Providing a bridge among the two technologies is a major challenge posed by the DSN problem.

1.2. THE CHALLENGES OF DISTRIBUTED PROCESSING

While our world consists of multitudes of concurrent asynchronous loosely coupled processing activities, we do not possess adequate formal methodologies to build systems that have these attributes. The mathematical tools that we have, usually assume a centralized implementation. Different signal processing algorithms, for instance, subsume that they will be processed on a single processor. Little is known about organizing a community of processors to perform a global task.

This research was supported in part by the Defense Advanced Research Projects Agency Project N00039-82-C-0427

Even problems that are trivial in a centralized setting become very difficult in a distributed environment. To illustrate the difficulties consider a simple problem such as finding a spanning tree in a graph. A centralized algorithm is an easy high school programming problem. Now suppose the nodes of the graph represent network nodes and edges represent communication links. Each node can only observe edges incident on it and communicate with its neighbors. How should we design an algorithm (protocol) for the nodes to mark the edges of a spanning tree. The generic aspects of the problem include the problem of fusing partial observations to compute a global object (e.g., the spanning tree), merging computing and communications to obtain an adequate (optimal?) computation, and coordinating the decisions of distributed computing agents. These problems are far from having methodological solutions and appear recurrently in most algorithmic problems of a DSN.

1.3. INTEGRATION

Finally, we have to consider the problem of integration, that is, building a DSN architecture which best integrates sensor, signal processing, estimation, artificial intelligence, distributed processing and computer communication technologies.

The very process of integration leads to interesting problems where the requirements posed by one technology impact the development of other technologies.

2. AN INCREMENTAL POSITION LOCATION SYSTEM

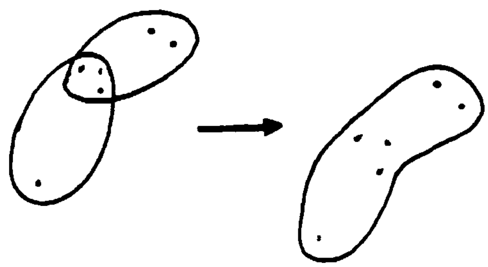
As an example for a typical DSN problem consider the problem of position location (PL). The problem arises in packet radio networks where two nodes within hearing distance can measure the range from each other, using a simple protocol. In a narrow sense, the position location problem is that of computing the location of network nodes from range measurements between some node pairs.

The PL problem offers a microcosm of typical DSN problems: The observations of ranges are distributed, the objects to be computed (relative position coordinates) are global, broader observations other than range measurements may be incorporated (e.g., knowledge that a given node is a vehicle that must be located on a road).

The incremental position location system (IPLS) to be sketched below presents a framework for handling incremental fusion of positional knowledge from distributed observations. The major ideas underlying IPLS can be summarized as follows.

The system uses hypergraphs to represent its positional knowledge. A hypergraph consists of a set of nodes and a set of hyperedges, where a hyperedge is a subset of nodes (a graph is a special case of a hypergraph where all hyperedges consist of exactly 2 nodes). Hyperedges are used to represent subsets of nodes whose position coordinates, relative to a common coordinate system, are known.

Positional knowledge is incremented using grammatical rules for the fusion of positional knowledge. That is, when a set of hyperedges are rigidily attached to each other (so that all participating nodes may be positioned in a common coordinate system), they are amalgamated into a single hyperedge. For instance, one of the rules is that two hyperedges having 3 joint nodes may be amalgamated together. That is, the positions of all nodes in the two hyperedges may be computed relative to a single coordinate system. Below we demonstrate the rule schematically. We call the grammar consisting of these rules an amalgamation grammar.



The system separates the PL problem into structural topological analysis and geometrical synthesis. At the analysis phase the system identifies which rules of the amalgamation grammar should be applied and in what order. During the geometrical synthesis phase each one of the grammatical rules is interpreted in terms of a procedure to compute positions. For example, in the rule presented above the system separates the actual computations of positions relative to any coordinate system from the fact that two sets of nodes positioned in two respective coordinate systems and having 3 joint points may be located in a single coordinate system. This separation of geometry from the topology has a few advantages:

1. It is possible to change the geometrical algorithms to synthesize positions without having to change the structural combinatorial analysis of the problem.

2. For a mobile system it is possible to perform the structural analysis once and then use the results of this analysis to compute positions while the range measurements change (as long as the topology does not change the structural analysis remains the same). This simplifies the computations substantially since the topology might be expected to change at a slower rate than the values of the ranges measured.

3. The system can be stopped at any moment and provide its current knowledge, it can keep track of the manner in which it reaches results and perhaps (a matter for a future research) optimize the computations. For instance, it is known that PL computations may exhibit numerical instabilities caused by successive truncation errors. Such errors may be avoided by optimizing the application of the grammatical rules. The system can be augmented to include additional amalgamation rules including rules provided interactively by the user to support the computation. Finally, the system can recognize redundancy of information and use it in eliminating the effects of measurement errors. It is also possible (a matter of current research) to identify gross failures (e.g., if a unit provides a grossly erroneous measurement one would like to detect this as failure rather than average this measurement with others).

It is possible to show that the order in which one applies the amalgamation rules is immaterial. That is, the computations would result in the same value. This implies that the application of amalgamation rules may proceed in an asynchronous distributed manner. In fact, the amalgamation grammar provides crude mechanisms to evaluate the extent to which one can solve the PL problem using localized computations and communications.

Current work on IPLS is focused on distributing its computations, studying 3-dimensional amalgamation grammars, applying the incremental approach to other PL problems (e.g., the problem of computing the structure of giant molecules from knowledge of local positional constraints such as the length of bonds). Finally, the grammatical approach to fusion of distributed information seems to extend to other distributed problem solving in a DSN. Work on extending the approach in these direction is in progress.

3. NONLINEAR ESTIMATION AND FILTERING

Another class of interesting DSN problems concern extension of estimation-filtering technology to handle the type of observations and parameters to be extracted in a DSN environment. Of particular significance are two generic issues: First, typical processes observed in a DSN environment are nonlinear counting processes. Second, the problem of distributed estimation of distributed phenomena needs to be resolved.

One possible begining is to observe, model and explain similar distributed sensors networks designed by nature. One interesting DSN example is the inner ear. The inner ear integrates distributed observations of counting processes represented by the spikes carried over the different nerve cells. Several questions need to be addressed here:

1. What is the amount of information carried by a single nerve fiber?
2. How can one combine the simultaneous information from several fibers to gain better knowledge about the stimulus?

The processing of information extracted from several nerve fibers appears to be the crucial question for the understanding of the neural coding taking place in the cochlea. The possibility of estimating the stimulus given the histogram data recorded from adjacent nerve fibers needs to be investigated. By interpreting the observations derived from adjacent nerve fibers as a multidimensional point process one arrives at a distributed nonlinear estimation model.

In view of the above, it is important to investigate the possibility of reformulating the modeling and nonlinear estimation problem with counting point process observations. In addressing this task we propose the investigation of the possibility of using concepts of distributed processing and parallel computation.

The great advantage of this approach is that it yields schemes for distributing the computational load to various sensor locations. At the location of each sensor a simple preprocessing of data will take place (for instance, counting of a time events in a given time interval). Therefore, it appears to be of considerable interest to study how to optimaly take observations corresponding to some criteria with or without redundancy of information of adjacent sensors. Having obtained the preprocessed information of each of the sensors, the remaining problem is to combine the data in such a way as to regain the needed information. To achieve this goal we propose to restate the nonlinear estimation problem by first questioning the essential ingredient in the standard accepted procedure: the centralized recording of observations (CRO). The CRO is replaced in our model by a method based upon the concepts of parallel and distributed processing. Two basic open questions are:

What information should be extracted?

How should it be combined?

The above problem appears to be a formidable task due to the complete lack of previous work on distributed processing with counting point process observations. Reformulation of the nonlinear estimation problem makes it therefore imperative to consider both the theoretical and practical aspects of the proposed distributed recording of observations (DRO) and the implementation of the central processor (nonlinear filter).

4. DECENTRALIZED OPTIMIZATION PROBLEMS

Many DSN problems involve design of an optimum decentralized algorithm. How should one derive optimal behavior algorithms for computer communication networks? The classic approach to the problem views the network as a single entity to which a global performance objective is assigned. This leads to a centralized optimization problem. The major shortcoming of this approach is that when one has overcome the complexity of deriving an optimal solution the network problem is still not solved, since the centralized objective usually leads to a centralized behavior policy. This centralized optimal behavior needs to be decentralized to serve as an adequate solution. The process of decentralization is usually more difficult than that of solving the original optimization problem. Therefore decentralization is usually an ad-hoc approximation process with little formal methodological support.

An alternative approach is to view the network as a loose collection of interfering agents (i.e., nodes, processes), each of which is assigned a selfish utility function which it seeks to optimize. The problem then becomes that of finding an adequate compromise among the selfish needs of the agents. One usually adopts Pareto optimality as the norm for rational behavior. That is, the agents should select a policy which is not dominated by any other policy (in the sense that no subset of agents can improve their performance without a performance degradation of some other agents). The major advantage of this selfish approach to optimal network behavior, compared with the global approach, is that it generates policies that are immediately decentralized.

The selfish-optimization approach has been successfully applied to two classes of problems: the problem of channel access scheme and the problem of flow control in virtual-circuit networks. Both problems do not lend themselves to a global solution (the flow control problem, provably so). The selfish approach yields characterization of optimum solutions, as well as decentralized greedy algorithms to obtain them. Moreover, the solution separates the problem into establishing some global cost criteria by a centralized authority (these correspond to some lagrangian multipliers) and decentralized local adaptation rules which are based on this global pricing rule. This renders the solution.