Routing for Statistical Inference in Sensor Networks

A. Anandkumar, A. Ephremides, A. Swami, and L. Tong

Abstract

In the classical approach, the problem of distributed statistical inference and the problem of minimum cost routing of the measurements to the fusion center are treated separately. Such schemes cannot exploit the "inherent" saving in routing costs arising from data reduction in a sufficient statistic for inference. Our approach is to conduct in-network processing of the likelihood function which is the minimal sufficient statistic and deliver it to the fusion center for inference. We employ the Markov random field (MRF) model for spatial correlation of sensor data. The structure of the likelihood function is well known for a MRF from the famous Hammersley-Clifford theorem. Exploiting this structure, we show that the minimum cost routing for computation and delivery of the likelihood function is a Steiner tree on a transformed graph. This Steiner-tree reduction preserves the approximation ratio, which implies that any Steiner-tree approximation can be employed for minimum cost fusion with the same approximation ratio. In this chapter, we present an overview of this approach to minimum cost fusion.

1 Introduction

Routing in communication networks, both wireline and wireless, has been a subject of extensive and in-depth study over the last few decades. It is a subject that is fairly well understood. Its "state-of-the-art" status can be summarized as follows; If a well-defined performance measure can be translated to a link metric, then there are low-complexity, efficient, robust, fast-converging, and often distributed algorithms for finding the optimal routes. Note the important distinction regarding the possibility of mapping the performance measure to a link metric. For example, in the internet, if end-to-end latency is the performance measure, then the link metric is delay over the link. Bellman-Ford-type algorithms then perform very well and quickly discover the best routes [1]. By contrast, on the traditional circuit-switched voice telephone network, where the performance measure is blocking probability, there is no known link metric that captures the performance measure and, hence, up to this day we only have heuristic routing algorithms for assigning routes to accepted calls.

At this point it is also important to note that the routing problem, being basically a discrete optimization task, has always a default solution that consists of the exhaustive search over the finite number of possible routes. The only reason this solution is unattractive is the prohibitive complexity of this search as the network size increases.

Another example of successful mapping of a performance measure to a link metric that allows the use of efficient algorithms is energy consumption in a wireless network. The energy consumption on a single link is then the right metric. That link energy consumption, depending on the assumptions on the network operation, consists of the transmission energy (proportional to the transmission power needed to reach the destination at a given rate and bit-error-rate target for chosen modulation and coding schemes, as well as to the channel attenuation), the energy expended for reception at the receiving end of the link, and, finally, the residual energy at the battery of the node at that end.

What all routing problems to date share is the traditional IP paradigm of store-and forward, which treats the source packets as "sacrosanct" monoliths that must be carried through the network intact until they are received at the destination node. Already, the idea of network coding has shown how it is possible to improve performance if this paradigm is reconsidered [2]. In this chapter we will examine a different issue that arises in specialized routing that shows equally well the inadequacy of traditional packet forwarding.

Our focus will be the case of wireless sensor networks. The unique characteristic of such networks is that the performance measure is typically associated with the "mission" of the network. For example, if the sensor network is deployed for the purpose of detecting the presence of a target, then the objective is to maximize the probability of correct detection, subject to the usual constraint o the false alarm rate. In other words, the mission of the network is statistical inference. Thus, the collected measurement data at the source sensors need not be forwarded to the fusion center (i.e. the ultimate destination node) in their entirety. Of course, such complete forwarding remains an option (just as the exhaustive search over all possible routes was an option in ordinary routing). But it is an inefficient option that is highly undesirable in networks that must also prolong their lifetime as much as possible. Wireless sensor networks often do not have the possibility of recharging or replacing the node batteries and thus it is important to "compress" the source data as much as possible. For statistical inference we do know that the collected data often map into a "sufficient statistic" that consists of substantially fewer bits than the original data [3]. In addition, intermediate nodes that are chosen to route the sufficient statistic information from a neighboring node, make their own measurements as well and therefore need to combine the received information with their own data to form a collective sufficient statistic for further forwarding. Thus, the problem of routing is intimately intertwined with the process of statistical inference in a novel way. It calls for a distributed computation of a global sufficient statistic that is based on all the collected measurements but with as little energy expenditure as possible for the needed data exchange among the nodes.

In fact, this distributed computation process under energy efficiency constraints is a prime example of cross-layer optimization in wireless networks. It couples the process of routing with the physical layer (where the energy expenditure occurs) and the application layer (where the statistical inference takes place). And it is an example of a totally new and unexplored aspect of wireless sensor operation. At the same time it raises some fundamental issues of energy consumption for distributed computation of a function and introduces the trade-off between communication and computation that has been examined in different contexts before.

In this chapter we aim at a comprehensive presentation of this new aspect of wireless sensor networking and at a unified study of routing, inference, and energy consumption. Inherent in this presentation is the notion of combinatorial optimization (which remains the underpinning element of the routing task) and of spatial information modeling (which defines the information dependencies in the data the sensor nodes gather). The treatment is self-contained but depends on several pieces of recent work that is referred to and summarized at appropriate parts of the chapter.

2 Spatial Data Correlation

In many realistic scenarios the sensor measurements are correlated, and our framework takes this into account. Examples of correlated signals include temperature and humidity sensors, and magnetometric sensors tracking a moving vehicle. Acoustic data are rich in spatial correlations due to the presence of echoes caused by multipath reflections. Moreover, in general, spatial signals are acausal in contrast to temporal signals. In the literature, the two are usually distinguished by referring to acausal signals as random fields and to causal signals as random processes. An example of exploiting correlation in a causal propagation setting can be found in [4, 5].

The model for spatially-correlated data crucially affects in-network processing of raw data. Various assumptions on correlation have been made in the literature. Joint-Gaussian distributions and distance-based correlation function have been widely assumed due to their simplicity [6–9]. Alternatively, diffusion-based [10] and joint-entropy based models [11] have also been employed. The use of remote-sensing data, proposed in [12], may not meet the resolution requirements. The model proposed in [13] is a special case of a Markov random field (MRF).

Markov random fields, as a class of parametric models for spatial data, were introduced by Besag [14, 15], and were known as conditional auto-regressions in his works. Prior to these works, Hammersley and Clifford formulated their now famous theorem on the equivalence of MRF to a Gibbs field [16]. However, the manuscript was never published, and a sketch of the original proof can be found in [17], along with further discussion on the historical aspects of research on MRF.

The use of the MRF model for spatial data in sensor networks is relatively new (e.g., [18]), although it is widely used in image processing [19] and geo-statistics [20]. This could be due to the complexity of the model for arbitrarily-placed nodes. We will see that the use of a Markov random field model leads to the formation of "clusters" that are based on the statistical dependence, rather than other considerations such as residual energy [21, 22]. The notion of clustering has been used extensively in sensor networks, where nodes send their data to one member of the cluster, which then processes and forwards to the destination. However, here, the issues are complicated by the fact that measurements processed in these statistical "clusters" have to be further aggregated rather than simply being forwarded to the destination.

We assume that all the sensors know the Markov random field model. In practice, the dependency structure and the model parameters of the Markov random field model can be estimated by incorporating a training phase. The seminal work of Chow and Liu in [23] considers the problem of approximating an unknown distribution from its samples using a procedure for learning the tree model that maximizes the likelihood of the training samples among the set of all possible tree models . Recently, learning graphical models from data samples specifically for binary hypothesis testing has been considered in [24]. Their procedure learns each hypothesis model from both sets of training samples.

In this chapter, we employ the Markov random field model for spatial correlation, taking into account only its graphical dependency structure; but no parametric correlation function is assumed. Moreover, any general random field without special properties can be represented as a MRF with a complete dependency graph (called the *saturated models* [25]).

2.1 Notations and Basic Definitions

An undirected graph G is a tuple $G = (V, E)$, where V is the vertex set and $E = \{(i, j)\}$ is the edge set. We allow graphs to have multiple or parallel edges, but no loops. The neighborhood function $\mathcal{N}(i;G)$ of a node i is the set of all other nodes having an edge with it in G. Let $\text{Deg}(i)$ denote the degree of node i. A subgraph induced by $V' \subset V$ on G is denoted by $G(V')$, and a complete subgraph or a clique has edges between any two nodes in V'. A maximal clique is one that is not contained in any other clique. Throughout this chapter, a clique refers to a maximal clique, unless otherwise mentioned. For a directed graph (digraph), we denote the edges (arcs) by $< i, j >$, where the direction is from i to j, and node j belongs to the set of immediate successors of i, and i is in the set of immediate predecessor of j. The above graph functions are extended to sets, for example, (i, A) denotes the set of edges between i and members of A. For sets A and B, let $A\backslash B = \{i : i \in A, i \notin B\}$ and let $|\cdot|$ denote cardinality. For matrix $\mathbf{A}$, $A(i, j)$ is the element in the i^{th} row and j^{th} column and $|\mathbf{A}|$ its determinant.

2.2 Definition and Properties of MRF

The MRF falls under the framework of acausal graphical models and satisfies conditional-independence properties, based on an undirected graph known as the *dependency graph* and is defined below.

Definition 1 (Markov Random Field) *Let* $\mathbf{Y}_V = [Y_i, i \in V]^T$ *denote the random vector of measurements in set* V*.* $\mathbf{Y}_V$ *is a Markov random field with an (undirected) dependency graph* $G_d = (V, E_d)$*, if* $\forall\, i \in V$*,*

$$Y_i \perp \mathbf{Y}_{V\backslash\{i,\mathcal{N}(i)\}} | \mathbf{Y}_{\mathcal{N}(i)}, \tag{1}$$

where $\perp$ *denotes conditional independence.*

In words, the above definition states that the value at any node, given the values at its neighbors, is conditionally independent of the rest of the network.

2.2.1 Example: One Dimensional MRF

A simple example is the first order auto-regressive (AR-1) process, given by

$$Y_t = A_{t-1}Y_{t-1} + \epsilon_{t-1}, \qquad Y_{t-1} \perp \epsilon_{t-1}, \quad \forall t \in V = \{1, \ldots, n\}. \tag{2}$$

Since Y_t is conditionally independent of the past, given the measurement Y_{t-1}, we write

$$Y_t \perp Y_{1,\ldots,t-2} | Y_{t-1}, \quad 2 < t \leq n.$$

Similarly, we can write

$$Y_{t+2,\ldots,n} \perp Y_t | Y_{t+1}, \quad 1 \leq t < n.$$

This implies that

$$Y_t \perp \mathbf{Y}_{V\backslash\{t-1,t,t+1\}} | \{Y_{t-1}, Y_{t+1}\}, \quad \forall t = 2, \ldots, n{-}1, \qquad Y_1 \perp \mathbf{Y}_{V\backslash\{1,2\}} | Y_2, \quad Y_n \perp \mathbf{Y}_{V\backslash\{n,n-1\}} | Y_{n-1}.$$

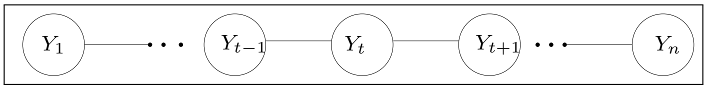


Figure 1: Linear dependency graph for a MRF representation of autoregressive process of order 1.

Hence, we have the dependency graph with neighborhood function

$$\mathcal{N}(t) = \{t-1, t+1\}, \quad \text{for } t \neq 1, n, \quad \mathcal{N}(1) = 2, \quad \mathcal{N}(n) = n-1.$$

In other words, the dependency graph is a linear chain, as shown in Fig.1. Hence, the conditional independence relations of the AR-1 process have a simple graphical representation which is not apparent in (2). However, the dependency graph does not capture all the information of the AR-1 process, in particular, that the process is causal. On the other hand, the dependency graph can be used to model more general acausal dependencies, typically found in spatial random fields.

2.2.2 Properties of a general MRF

For a Markov random field, in fact, three types of Markov properties can be defined:

1. Local Markov Property: $Y_i \perp \mathbf{Y}_{V\backslash(i\cup\mathcal{N}(i))}|\mathbf{Y}_{\mathcal{N}(i)}, \forall i \in V$.
2. Global Markov Property: $\mathbf{Y}_A \perp \mathbf{Y}_B|\mathbf{Y}_C$, where A, B, C are disjoint sets. A, B are non-empty and C separates A, B. See Fig.2.
3. Pairwise Markov Property: $Y_i \perp Y_j|\mathbf{Y}_{V\backslash\{i,j\}} \iff (i,j) \notin E$

In definition 1, we have used the local Markov property. We can immediately see that the global Markov property implies the local Markov property, since we can set

$$A = \{i\}, B = V\backslash\{i, \mathcal{N}(i)\}, C = \mathcal{N}(i).$$

Similarly, the global Markov property implies the pairwise Markov property, since we can set

$$A = \{i\}, B = \{j\}, C = V\backslash\{i,j\}, \quad \forall (i,j) \notin E_d.$$

The three properties can be shown to be equivalent under the positivity condition [25]. The positivity condition is as follows: for all $A \subset V$ with samples $\mathbf{y}_A$, $\mathbf{y}_{V\backslash A}$ such that $f(\mathbf{y}_A), f(\mathbf{y}_{V\backslash A}) > 0$, the conditional is also positive

$$f(\mathbf{y}_A|\mathbf{y}_{V\backslash A}) > 0,$$

where f is the density function. An equivalent condition for positivity is

$$(f(\mathbf{y}_V) = 0) \Rightarrow (f(y_i) = 0), \quad \forall i \in V.$$

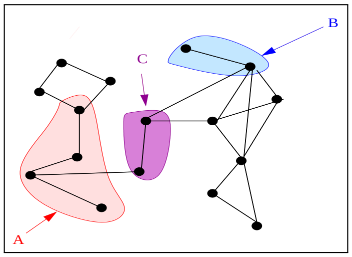


Figure 2: Global Markov Property: $\mathbf{Y}_A$ is conditionally independent of $\mathbf{Y}_B$ given $\mathbf{Y}_C$.

An example that does *not* satisfy positivity is the fully correlated case: $Y_1 = Y_2 \ldots = Y_n$. In this case, the joint likelihood is zero whenever all the samples are not equal, but the marginal likelihood is not necessarily zero.

The Hammersley-Clifford theorem [17] states that for a MRF $\mathbf{Y}_V$ with dependency graph $G_d = (V, E_d)$, the joint pdf f, under the positivity condition, can be expressed as

$$-\log f(\mathbf{Y}_V; \Upsilon) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{Y}_c), \tag{3}$$

where $\mathcal{C}$ is a collection of (maximal) cliques in G_d, the functions ψ_c, known as *clique potentials*, are real valued, non-negative and not zero everywhere on the support of $\mathbf{Y}_c$. Thus, the tuple $\Upsilon = \{G_d, \mathcal{C}, \psi\}$ specifies the MRF in (3). We assume that the normalization constant is already incorporated in the potential functions, in order to ensure that we have a valid pdf. For general potentials, finding the normalizing constant (called the *partition function*) is NP-hard, but approximate algorithms have been proposed in [26].

From (3), we see that the complexity of the likelihood function is vastly reduced for sparse dependency graphs; here, the conditional-independence relations in (1) results in the factorization of the joint likelihood into a product of components, each of which depends on a small set of variables. This form is already exploited by distributed algorithms such as belief propagation [27] for local inference of hidden measurements. In this chapter, we exploit the MRF model for a global inference problem, explained in Section 3.

In this chapter, we assume that the number of cliques $|\mathcal{C}|$ of the MRF is polynomial in the number of nodes. This is satisfied by many graph families such as bounded-degree graphs [28]. Note that in (3), the set of cliques $\mathcal{C}$ contains only those cliques with non-zero potentials. For example, for independent measurements, $\mathcal{C}$ is the vertex set, and we have the likelihood function as a weighted sum function,

$$-\log f(\mathbf{Y}_V) = -\sum_{i \in V} \log f_i(Y_i), \quad \mathbf{Y}_V \sim \prod_{i \in V} f_i,$$

where f_i is the marginal pdf of Y_i. Besag's auto-model [15] is a special MRF with only pairwise dependencies, and hence, the clique set $\mathcal{C}$ is the set of edges E_d. This leads to a simplified expression for the likelihood function,

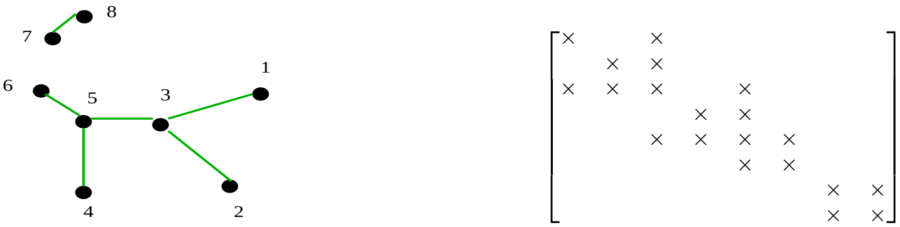
$$-\log f(\mathbf{Y}_V; \{G_d, E_d, \psi\}) = \sum_{(i,j) \in E_d} \psi_{i,j}(Y_i, Y_j). \tag{4}$$

Multi-parameter exponential family of conditional probabilities can be used to define such pairwise Markov random fields [29]. An example of Besag's model is the Ising Model, which was first introduced to study phase transition in ferromagnetic materials.

2.2.3 Gauss-Markov Random Field

The Gauss-Markov Random Field (GMRF) has some special properties. In this case, (3) is equivalent to (4), since the likelihood function of $\mathbf{Y}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ is given by

$$\log f(\mathbf{Y}_V; \mathbf{A}) = \frac{1}{2}(-n \log 2\pi + \log |\mathbf{A}| + \sum_{i \in V} A(i,i) Y_i^2 + \sum_{i,j \in V} A(i,j) Y_i Y_j), \tag{5}$$



(a) A labeled simple undirected graph.

(b) × : Non-zero elements of potential matrix.

Figure 3: One-to-one correspondence between dependency graph edges and non-zero elements of potential matrix for a GMRF.

where $\mathbf{A}:=\mathbf{\Sigma}^{-1}$ is the inverse of the covariance matrix. For a given dependency graph $G_d = (V, E_d)$, the GMRF should also satisfy (4). Hence, comparing the two equations (4) and (5), we have

$$A(i,j) = 0 \iff (i,j) \notin E_d.$$

Hence, there is a one-to-one correspondence between the non-zero elements of $\mathbf{A}$ and the dependency graph edges E_d, and is illustrated in Fig.3. Since $\mathbf{A}$ is associated with the potentials, it is called the *potential matrix*. Hence, for the Gaussian distribution, we only need the edges of the dependency graph and not the higher-order cliques. Moreover, for the Gaussian case, the edge potential $\psi_{i,j}(Y_i, Y_j)$ in (4) reduces to the sum of squares and cross-products of the measurements, weighted by the coefficients of the potential matrix $\mathbf{A}$. When the dependency graph is acyclic, we can additionally obtain a closed form for the elements of the potential matrix $\mathbf{A}$, in terms of the elements of the covariance matrix $\mathbf{\Sigma}$.

Fact 1 (GMRF with Acyclic Dependency Graph) *The coefficients of the potential matrix* $\mathbf{A}:=\mathbf{\Sigma}^{-1}$*, with zero mean and covariance matrix* $\mathbf{\Sigma}$ *and acyclic dependency graph* $G_d = (V, E_d)$*, are*

$$\begin{aligned} A(i,i) &= \frac{1}{\Sigma(i,i)}\Big(1 + \sum_{j \in \mathcal{N}(i)} \frac{\Sigma(i,j)^2}{\Sigma(i,i)\Sigma(j,j) - \Sigma(i,j)^2}\Big), \qquad (6) \\ A(i,j) &= \begin{cases} \dfrac{-\Sigma(i,j)}{\Sigma(i,i)\Sigma(j,j) - \Sigma(i,j)^2} & \textit{if } (i,j) \in E_d, \\ 0 & o.w. \end{cases} \qquad (7) \end{aligned}$$

The determinant of the potential matrix of $\mathbf{A}$ *is given by*

$$|\mathbf{A}| = \frac{1}{|\mathbf{\Sigma}|} = \frac{\prod_{i \in V} \Sigma(i,i)^{Deg(i)-1}}{\prod\limits_{\substack{(i,j)\in E_d \\ i<j}} [\Sigma(i,i)\Sigma(j,j) - \Sigma(i,j)^2]}. \qquad (8)$$

In fact, for any MRF with acyclic dependency graph G_d, the joint pdf $f_{\mathbf{Y}_V}$ can be expressed in terms of marginals at nodes f_{Y_i} and pairwise joint pdfs f_{Y_i,Y_j} as

$$f_{\mathbf{Y}_V}(\mathbf{y}_V) = \prod_{i \in V} f_{Y_i}(y_i) \prod_{(i,j) \in E_d} \frac{f_{Y_i,Y_j}(y_i, y_j)}{f_{Y_i}(y_i) f_{Y_j}(y_j)}. \tag{9}$$

See [30] for details.

3 Statistical Inference of Markov Random Fields

The problem of distributed detection considers a set of sensors, one of them designated as the fusion center or the decision node, and all the sensor observations are ultimately routed (in some form) to it. This setup is relevant when we need to make a global decision on the phenomenon (contrasting to local inference algorithms such as belief propagation). We consider the binary hypothesis-testing problem with two given hypotheses, the null hypothesis $\mathcal{H}_0$ and the alternative $\mathcal{H}_1$. We limit ourselves to only simple hypothesis testing, i.e., the probability measures under both the hypotheses are known to all the sensors.

In statistical theory, a *sufficient statistic* is a well-behaved function of the data, which is as informative as the raw data for inference. Formally, a function $T(Y)$ is said to be a sufficient statistic for model P_θ, if conditioned on $T(Y)$, $Y \sim P_\theta$ does not depend on θ. It is said to be *minimal* if it is a function of every other sufficient statistic for P_θ [3]. A minimal sufficient statistic for inference represents the maximum possible reduction in dimensionality of the raw data, without destroying information about the underlying phenomenon [3]. The log-likelihood ratio (LLR) is the minimal sufficient statistic for hypothesis testing [31]. Let $f(\mathbf{Y}_V; \mathcal{H}_j)$ be the pdf of the measurements $\mathbf{Y}_V$ under hypothesis j. The optimal decision rule at the fusion center is a threshold test based on the log-likelihood ratio (LLR),

$$\text{LLR}(\mathbf{Y}_V) := \log \frac{f(\mathbf{Y}_V; \mathcal{H}_0)}{f(\mathbf{Y}_V; \mathcal{H}_1)}. \tag{10}$$

The result is also true for the M-ary hypothesis testing problem, where the LLR vector

$$\left[\log \frac{f(\mathbf{Y}_V; \mathcal{H}_0)}{f(\mathbf{Y}_V; \mathcal{H}_1)}, \ldots, \log \frac{f(\mathbf{Y}_V; \mathcal{H}_0)}{f(\mathbf{Y}_V; \mathcal{H}_{M-1})}\right]^T$$

is minimally sufficient.

Form of Log-Likelihood Ratio for MRF

In this chapter, we assume that the measurement samples are drawn from distributions specified by distinct Markov random fields, defined on the same node set. In particular, we consider

$$\mathcal{H}_0 : \Upsilon_0 = \{G_0(V), \mathcal{C}_0, \psi_0\} \quad \text{vs.} \quad \mathcal{H}_1 : \Upsilon_1 = \{G_1(V), \mathcal{C}_1, \psi_1\}. \tag{11}$$

From (3) and (10), the LLR is given by the difference of the respective clique potentials,

$$\text{LLR}(\mathbf{Y}_V) = \sum_{a \in \mathcal{C}_1} \psi_{1,a}(\mathbf{Y}_a) - \sum_{b \in \mathcal{C}_0} \psi_{0,b}(\mathbf{Y}_b). \tag{12}$$

It is easily seen that the LLR can be expressed as the sum of potentials of an "effective" Markov random field $\Upsilon = \{G_d, \mathcal{C}, \phi\}$ specified as follows: the effective dependency graph $G_d = (V, E_d)$, has the edge set $E_d = E_0 \cup E_1$; the effective clique set is $\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1$, with only the resulting maximal cliques retained; the effective potential functions ϕ_c are given by

$$\phi_c(\mathbf{Y}_c) := \sum_{a \in \mathcal{C}_1, a \subset c} \psi_1(\mathbf{Y}_a) - \sum_{b \in \mathcal{C}_0, b \subset c} \psi_0(\mathbf{Y}_b), \quad \forall c \in \mathcal{C}. \tag{13}$$

Therefore, the LLR has a succinct form, which will be used in the rest of this chapter,

$$\mathrm{LLR}(\mathbf{Y}_V; \Upsilon) = \sum_{c \in \mathcal{C}} \phi_c(\mathbf{Y}_c). \tag{14}$$

4 Optimal Routing for Inference with Local Processing

By optimal routing for inference, we mean the fusion scheme that minimizes the total costs of routing under the constraint that the likelihood function in (14) is delivered to the fusion center. Such a scheme involves computing the likelihood function consisting of clique potential functions, each depending only on the measurements in the clique. Hence, these clique potential functions can be computed independently at various nodes. To exploit the Markovian structure of the underlying hypotheses, we consider a class of data fusion schemes that perform local processing within the cliques of the MRF. Specifically, an aggregation scheme involves the following considerations, viz., each clique potential is assigned a computation site or a processor; measurements of the clique members are then transported to its processor to enable computation of the clique potentials. These values are then summed up and delivered to the fusion center. See Fig.5.

4.1 Network and Communication Model

We assume the presence of a medium-access control (MAC) that eliminates collisions or interferences among the nodes. The network is connected, i.e., communication is feasible via a multi-hop route between any two nodes in the network. We assume that communication is bidirectional. We consider the unicast mode of routing, where a packet from a node is routed to a single destination and the intermediate nodes do not perform any processing or store the packet for future use.

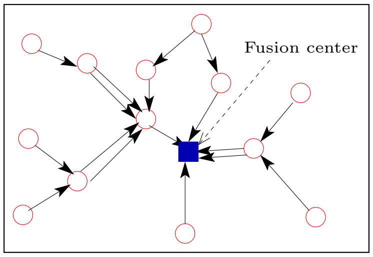
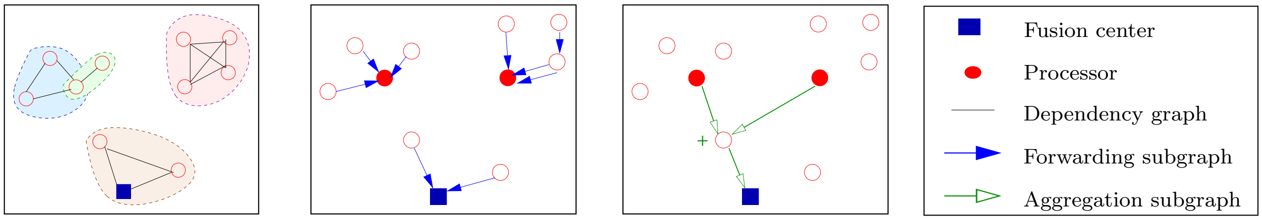


Figure 4: Routing to the designated fusion center is represented through a digraph. Each arc represents the routing path of one packet, carrying one real number.



(a) Maximal cliques of dependency graph

(b) Forwarding subgraph computes clique potentials

(c) Aggregation subgraph adds computed potentials

(d) Legend

Figure 5: Schematic of dependency graph of Markov random field and stages of data aggregation. The forwarding and aggregation subgraphs transport raw and aggregated data.

In our formulation, the processing costs are assumed constant, thus ignored in the optimization. Usually the routing costs reflect transmission energy, but it could also represent, for example, delay, bandwidth, or a combination of these considerations. We represent the routing of a real number by a packet. A symmetric routing cost function is assumed, and is denoted by $C_{i,j} > 0$ between i and j. The metric closure on graph G, is defined as the complete graph where the cost of each edge (i, j) in the metric closure is the cost of the shortest path between i and j in G. [32, p. 58]. Henceforth, we only consider the metric closure of the communication graph, denoted by G_c, and denote the metric costs by $\bar{C}_{i,j}$. There is no loss of generality, since the edges of the metric closure can be replaced with the corresponding shortest paths. For any graph $G \subset G_c$, let $\mathrm{C}(G)$ denote the total cost of its links,

$$\mathrm{C}(G) := \sum_{e \in E} C_e, \tag{15}$$

where C_e is the cost of the link e and E is the set of links in G; if a link is used m times, then E contains m parallel links to incorporate the costs in our formulation.

In our formulation all real numbers are quantized with sufficiently high precision to ignore the quantization error and all nodes function as both sensors and routers. Quantization is indeed an important issue for detection and communication. However, even in the classical distributed setup, optimal quantization is not tractable for the correlated case. The recent works on this topic consider conditionally i.i.d. measurements with a fixed network topology of bounded-height tree [33] or a tandem network [34].

4.2 Formulation of Minimum Cost Fusion

Recall the succinct form of LLR in (14),

$$\mathrm{LLR}(\mathbf{Y}_V; \Upsilon) = \sum_{c \in \mathcal{C}} \phi_c(\mathbf{Y}_c). \tag{16}$$

Hence, the LLR consists of the sum of the clique potential functions ϕ and is amenable to localized processing within the cliques of the Markov random field. Hence, we propose a hierarchical order of processing the LLR. In the first stage, raw data are forwarded to compute all the potential functions at various nodes in the network. In the second stage, the computed values are summed up and delivered to the fusion center.

For the first stage of LLR computation, each clique potential function ϕ_c is assigned a unique computation site, known as the *processor* for clique c, denoted by $Proc(c)$. Once the processor for clique c is assigned, measurement Y_i of each clique member $i \subset c$ (other than the processor) is routed to $Proc(c)$ along a path of feasible communication links. Since we are considering unicast mode of communication, the minimum cost is along the shortest path represented by the link $< i, Proc(c) > \in G_c$ with cost $\bar{C}(i, Proc(c))$, where G_c is the metric closure of the communication graph. The set of all links used by a fusion scheme in the first stage of computation to forward raw data to the processors is called the *forwarding* subgraph, denoted by FG,

$$\text{FG}{:=}\{< i, Proc(c) >: i \subset c, i \neq Proc(c), c \in \mathcal{C}\}.$$

In the second stage of LLR computation, all the computed potential functions are summed up to obtain the LLR which is then delivered to the fusion center. The set of links used by a fusion scheme in the second stage of LLR computation to sum up the computed potential values is known as the *aggregation* subgraph, denoted by AG. The tuple with the forwarding and aggregation subgraphs of a fusion scheme is referred to as the *fusion digraph*, $G_f{:=}\{\text{FG}, \text{AG}\}$. A schematic of a fusion scheme is shown in Fig.5. The total routing costs of a fusion scheme is given by

$$\text{C}(G_f) = \text{C}(\text{FG}) + \text{C}(\text{AG}).$$

Hence, any fusion scheme in our setup is specified by a processor-assignment mapping $Proc$ and a fusion digraph $G_f = \{\text{FG}, \text{AG}\}$, and we represent the scheme by the tuple $\Gamma{:=}\{Proc, \text{FG}, \text{AG}\}$. Note that we do not explicitly specify the sequence in which data is transported and processed by a fusion scheme; we impose constraints to ensure that such a feasible sequence exists.

We first need the constraint that the scheme delivers the LLR to the fusion center

$$AggVal(v_0; \Gamma) = \text{LLR}(\mathbf{Y}_V; \Upsilon), \tag{17}$$

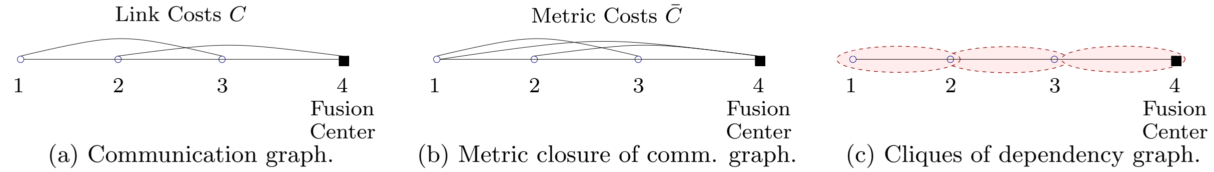
where $AggVal(i; \Gamma)$ is the value at node i at the end of fusion.

4.2.1 Local Processor Assignment

We now make the following additional assumption which simplifies the fusion scheme: each clique potential function ϕ_c is assigned a "local" processor, which is one of the clique members,

$$Proc(c) \subset c, \quad \forall\, c \in \mathcal{C}. \tag{18}$$

The local processor assignment also implies that local knowledge of potential function parameters is sufficient, i.e., each sensor i only needs to know the potential functions ϕ_c of the cliques c to which it belongs, and hence, the storage requirement at the sensors is considerably reduced. In practice, the potential function parameters are sent to the nodes by the fusion center after empirically estimating the joint-pdf of the measurements. Through this, the nodes also implicitly receive information about their clique memberships. Hence, local processor assignment can also reduce the communication overhead during the learning stage. Localized processing can be especially efficient when the dependency graph of the Markov random field is a proximity graph, where edges are



(a) Communication graph. (b) Metric closure of comm. graph. (c) Cliques of dependency graph.

Figure 6: Inputs to the problem of minimum cost fusion for inference.

based on local point configuration [35]. We now formally define the minimum-cost fusion scheme Γ^* which minimizes the total routing costs

$$\Gamma^* := \arg\min_{\Gamma} \mathrm{C}(G_f), \tag{19}$$

subject to the constraints in (17) and (18). Hence, the problem of minimum cost fusion takes the metric closure of communication graph and the maximal cliques of the dependency graph as inputs and provides a processor assignment and fusion digraph as outputs. An example of the problem of minimum cost fusion is illustrated in Fig.6, with the communication graph in Fig.6a and the chain dependency graph in Fig.6c, which are independent of one another. The resulting metric closure of communication graph in Fig.6b and cliques of dependency graph are taken as the inputs for the problem of minimum cost fusion.

4.2.2 0-1 Integer Programming Formulation

We now write a 0-1 integer program whose optimal solution provides the minimum cost fusion scheme in (19) for computing the LLR and delivering it to the fusion center v_0. We can map any valid fusion digraph $G_f = \{\mathrm{FG}, \mathrm{AG}\}$ and the processor assignment mapping $Proc$ to variables $\mathbf{y}$ and $\mathbf{z}$, defined as

$$z(j,c) := I[Proc(c) == j], \quad y(i,j) := I[< i,j > \in \mathrm{AG}],$$

where I is the indicator function. Once the processor assignment is fixed, the set of shortest paths from clique members to the processors minimizes the routing costs in the forwarding subgraph.

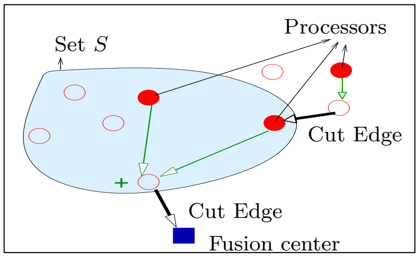


Figure 7: Cut edges of a set S separating the set of processors and fusion center.

Hence, we can set the forwarding subgraph as

$$\mathrm{FG} \leftarrow \{< i,j >: I(\sum_{c:i\subset c} z(j,c) \geq 1)\},$$

where we ensure that every node i forwards its measurement to node j, whenever j is the processor of cliques c that contain node i along the link in the metric closure (which has the same cost as the shortest path). Hence, the total routing costs of the fusion digraph can be expressed as,

$$\mathrm{C}(G_f) = \mathrm{C}(\mathrm{FG}) + \mathrm{C}(\mathrm{AG}) = \frac{1}{2}\sum_{i,j\in V}[I(\sum_{c:i\subset c} z(j,c) \geq 1) + y(i,j)]\bar{C}(i,j),$$

where the factor of $\frac{1}{2}$ ensures that each edge is counted only once. We now write a constraint equivalent to the local processor constraint in (21) and ensuring that at least one processor is selected,

$$\sum_{j\subset c} z(j,c) \geq 1, \quad \forall c \in \mathcal{C}.$$

We now need a constraint on the the aggregation subgraph to ensure that the sum of the potential functions is delivered to the fusion center, and hence, the constraint in (17) is satisfied. To this end, we define that A *separates* B if $A \cap B \neq \emptyset$ and $A \cap B \neq B$. We consider all sets $S \subset V$ separating the union of the set of processors and the fusion center. A cut edge of set S is one that has exactly one endpoint in S. As illustrated in Fig.7, since all the values at the processors contained within S can be summed up to a single packet, for the information to flow out of S (or into S), at least one cut edge of S is needed. Hence, we write the constraint that

$$\sum_{i\in S, j\notin S} y(i,j) \geq 1, \forall S \subset V \text{ separating } \{\bigcup_{c\in\mathcal{C}} Proc(c) \bigcup v_0\}.$$

We now have the integer program,

$$\begin{aligned}
\frac{1}{2}\min_{\mathbf{y},\mathbf{z}} \quad & \sum_{i,j\in V}[I(\sum_{c:i\subset c} z(j,c) \geq 1) + y(i,j)]\bar{C}(i,j) \qquad \text{(IP-1)}, && (20)\\
s.t. \quad & \sum_{j\subset c} z(j,c) \geq 1, \quad \forall c \in \mathcal{C}, \quad \text{let } Proc(c){:=}\{j : z(j,c) = 1\}, && (21)\\
& \sum_{i\in S, j\notin S} y(i,j) \geq 1, \forall S \subset V \text{ separating } \{\bigcup_{c\in\mathcal{C}} Proc(c) \bigcup v_0\}, && (22)\\
& y(i,j), z(j,c) \in \{0,1\}. && (23)
\end{aligned}$$

Hence, the optimal solutions to (19) and (20) are the same.

4.3 Special Case: IID Measurements

In the special case when the measurements are i.i.d. conditioned on either hypothesis, the log-likelihood ratio (LLR) in (10) is the sum of the log-likelihoods of individual sensor measurementsi.e.,

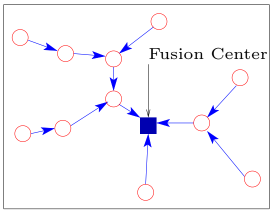


Figure 8: Minimum spanning tree is energy-optimal for detection of i.i.d. data.

$$\mathrm{LLR}(\mathbf{Y}_V) = \sum_{v \in V} \mathrm{LLR}(Y_v), \quad \mathbf{Y}_v \overset{i.i.d.}{\sim} \mathcal{H}_0 \text{ or } \mathcal{H}_1. \tag{24}$$

The minimum-energy routing in this case is given by the directed minimum spanning tree (DMST), with the directions toward the fusion center. See Fig.8. The sum function can be calculated hierarchically along DMST, starting at the leaves and ending at the fusion center. The minimum spanning tree MST(V) over a node set V is defined as the tree of minimum total length that spans all the nodes in V. The i.i.d. case in fact turns out to be a lower bound for the cost of optimal fusion in a general Markov random field.

Lemma 1 (Lower bound on C(G^*_f)) *The total routing cost for optimal fusion in (19) is no less than that of the minimum spanning tree (MST), based on the routing cost function, i.e.,*

$$C(MST(V)) \leq C(G^*_f(V)). \tag{25}$$

4.4 Minimum Spanning Tree-based Heuristic

We first propose a simple heuristic (AggMST), based on the minimum spanning tree. Here, we separate the design of processor selection and aggregation tree. We arbitrarily assign a clique member as the clique processor and then exploit the fact that it is feasible to compute the sum of the potentials along the MST. Of course, here only the processors have useful information in the form of potential functions and the other nodes just forward the aggregated information. This heuristic is simple to implement since there are efficient distributed algorithms for finding the MST [36, 37].

We specify the AggMST scheme in Fig.9. For a clique c, the processor is assigned arbitrarily to the clique member with the lowest index (line 3). Other suitable factors such as residual energy can instead be used for the assignment. The shortest-path routes from other members of c to the processor are added to the forwarding subgraph FG (line 5), and the raw data is routed along these links to enable the computation of the clique potentials. Note that the construction of the FG can be implemented in a localized manner whenever the dependency graph is local (e.g., k nearest-neighbor graph, disk graph). The aggregation subgraph AG is DMST(V), the minimum spanning tree, directed towards the fusion center (line 9) and potentials are added hierarchically along AG.

We now quantify the performance of the AggMST scheme for a special scenario that allows us to utilize the lower-bound in Lemma 1.

```
Require: $V = \{v_0, \ldots, v_{|V|-1}\}$, $v_0$: Fusion center, $\mathcal{C} = \{c_0, \ldots, c_{|\mathcal{C}|-1}\}$: maximal cliques of MRF,
    DMST($V$): Minimum spanning tree, direct toward $v_0$
SP($i, j$)= (Directed) shortest path from $i$ to $j$
for $j \leftarrow 0, |\mathcal{C}| - 1$ do
    $Proc(c_j) \leftarrow \min_{v_i \in c_j} v_i$                ▷ Arbitrary processor assignment
    if $|c_j| > 1$ then
        Add $\bar{C}(c_j \backslash Proc(c_j), Proc(c_j))$ to FG
    end if
end for
AG $\leftarrow$ DMST($V$), $\Gamma \leftarrow \{Proc, \mathrm{FG}, \mathrm{AG}\}$
return $\Gamma$
```

Figure 9: Heuristic for aggregation in a Markov random field (AggMST).

Theorem 1 (Approximation) *For the case when the routing costs are Euclidean and the dependency graph is a subgraph of the Euclidean MST, the AggMST scheme has an approximation ratio of* 2*.*

Proof: The MST in the lower bound (Lemma 1) is Euclidean, since the transmission costs are Euclidean. Since the dependency graph is a subgraph of the Euclidean MST, all the links in AggMST are contained in the Euclidean MST. Hence, we have the approximation ratio of 2. To show that the bound is tight, we note that the case of extended equilateral triangles on the Euclidean plane achieves this bound. □

4.5 Overview of Steiner Tree

In this section, we briefly define the Steiner tree and study its properties. These will be employed to describe our results in the subsequent sections. The material in this section is mainly from [38]. We first define the *Steiner minimal tree* [32, p. 148] below.

Definition 2 (Steiner tree) *Let G be an undirected graph with non-negative edge weights. Given a set $L \subset V$ of terminals, a Steiner tree (ST) is the tree $T \subset G$ of minimum total edge weight such that T includes all vertices in L.*

Finding the Steiner tree is NP-hard and there has been extensive work on finding approximation algorithms. A $0-1$ integer program to find the Steiner tree can be written as

$$\min_{\mathbf{y}} \quad \frac{1}{2} \sum_{i,j \in V} y(i,j) \bar{C}(i,j), \tag{26}$$

$$s.t. \quad \sum_{i \in S, j \notin S} y(i,j) \geq 1, \forall S \subset V \text{ separating } L, y(i,j) \in \{0,1\}, \tag{27}$$

where we say that A *separates* B if $A \cap B \neq \emptyset$ and $A \cap B \neq B$. This condition ensures that all the terminals are connected, as illustrated in Fig.7.

```
Require: $V = \{v_0, \dots, v_{|V|-1}\}$, $v_0$: Fusion center, $\mathcal{C} = \{c_0, \dots, c_{|\mathcal{C}|-1}\}$: maximal cliques of MRF,
$G_c$= Metric closure of comm. graph, $\bar{C}$ = Link costs in $G_c$,
$\mathrm{ST}(G, \mathcal{L})$ = $\delta$-approx. Steiner tree on $G$, terminal set $\mathcal{L}$
$G', V_c \leftarrow Map(G_c; \bar{C}, \mathcal{C})$
$\mathrm{DST} = \mathrm{ST}(G', V_c \cup v_0)$ and directed towards $v_0$
$\Gamma \leftarrow RevMap(\mathrm{DST}; V_c, V, \mathcal{C})$
return $\Gamma$
```

Figure 10: δ-approx. min. cost aggregation scheme Γ with processor assignment and fusion digraph via Steiner-tree reduction (AggApprox).

A simple *MST heuristic* approximates the Steiner tree over G and terminal set L with the minimum spanning tree spanning the set L, over the metric closure of G. The MST heuristic has an approximation bound of 2 [39]. The best known approximation bound for Steiner tree on graphs is 1.55, derived in [40]. The Steiner tree can be generalized to group Steiner tree, introduced by Reich and Widmayer [41].

Definition 3 (Group Steiner tree) *Let G be an undirected graph with non-negative edge weights. Given groups of vertices $g_i \subset V$ of terminals, a group Steiner tree is the tree $T \subset G$ of minimum total edge weight such that T includes at least one vertex from each group g_i.*

Since the group Steiner tree is a generalization of the Steiner tree, it is also NP-hard. For a group Steiner tree, polylogarithmic (in the number of groups) approximation algorithms have been proposed [42]. A series of polynomial-time heuristics are described in [43] with worst-case ratio of $O(|g|^\epsilon)$ for $\epsilon > 0$.

4.6 Steiner Tree Reduction

In this section, we show that optimal fusion has a Steiner-tree reduction. We specify the graph transformations required for such a reduction and finally obtain a valid fusion scheme with processor assignment and fusion digraph. We also show that the Steiner-tree reduction is approximation factor preserving. This implies that any approximation algorithm for Steiner tree provides the same ratio for minimum cost fusion.

4.7 Simplified Integer Program

We first note that if the processor assignment is already predetermined and not part of the routing cost optimization, then we can easily characterize the optimal solution. In practice, a predetermined processor assignment might be enforced by considering other factors such as processing capabilities or residual energies of different nodes. In this case, the forwarding subgraph is also predetermined by the shortest paths to the processors. The optimal aggregation subgraph is the Steiner tree with the set of processors and the fusion center, as the terminals. This is because the sum of the potential function values at the processors is computed optimally along the Steiner tree.

We next consider a modified cost optimization problem, where we ignore the routing costs of the forwarding subgraph, incurred in transporting the raw measurements to a processor. In [44, Lemma 3], we show that the minimum cost aggregation subgraph is the group Steiner tree [41], with nodes in each clique of the Markov random field forming a group.

The presence of processor assignment in cost optimization in (20) makes the problem harder than the above versions. It influences the costs of both the forwarding and aggregation subgraphs in a fusion scheme. It is not immediately clear that there is a Steiner tree reduction for (20). In fact, if we directly relax the integers to $\mathbf{y}, \mathbf{z} \geq 0$ in (20), the program is non-linear. We now use the local processor assignment constraint in (21) to write an equivalent integer program with a linear relaxation. Let $\mathbf{z}^*$ be the optimal solution to (20). We have

$$\begin{aligned}\sum_{i,j \in V} I(\sum_{c:i \subset c} z^*(j,c) \geq 1)\bar{C}(i,j) &= \sum_{i,j \in V} I(\sum_{c:i,j \subset c} z^*(j,c) \geq 1)\bar{C}(i,j), \\ &= \sum_{i,j \in V} \sum_{c:i,j \subset c} z^*(j,c)\bar{C}(i,j), \\ &= \sum_{\substack{c \in \mathcal{C} \\ |c|>1}} \sum_{i,j \subset c} z^*(j,c)\bar{C}(i,j), \end{aligned} \tag{28}$$

where the first equality is from local processor assignment constraint, the second equality is due to the fact that we need to assign only one processor and that there is a unique maximal clique c, if it exists, containing both i and j. Note that if the local assignment constraint is removed, then j might be assigned as the processor to many cliques c and hence, the equality does not hold. Interchanging the sums in the last equality is possible since the terms are non-zero when there is a clique c containing both i and j, and this implies that $|c| > 1$. Hence, we can now write an equivalent IP for minimum cost fusion under local processor assignment

$$\min_{\mathbf{y},\mathbf{z}} [\sum_{\substack{c \in \mathcal{C} \\ |c|>1}} \sum_{i,j \subset c} z(j,c)\bar{C}(i,j) + \sum_{i,j \in V} y(i,j)\bar{C}(i,j)] \qquad \text{(IP-2)}, \tag{29}$$

subject to the same constraints (21)-(23). Upon relaxation of the integer constraints, IP-2 is a linear program.

We now show that a Steiner tree on the transformed communication graph is the optimal solution to IP-2 in (29). To this end, we define an operation $Map(G_c)$ in Fig.11 which involves adding new virtual clique-representative nodes v_c for each non-trivial clique ($|c| > 1$) and adding edges between v_c and all the members of clique c with costs,

$$\bar{C}(v_c, j) := \sum_{i \subset c} \bar{C}(i,j), \quad \forall j \subset c.$$

The above cost represents the cost incurred in the forwarding subgraph upon assigning a node j as the processor for clique c. Let the set of all added clique representative vertices be V'. Hence, IP-2 in (29) is now equivalent to

$$\frac{1}{2} \min_{\mathbf{y},\mathbf{z}} \sum_{v_c \in V', j \in V} z(j,c)\bar{C}(v_c, j) + \sum_{i,j \in V} y(i,j)\bar{C}(i,j),$$

subject to the same constraints (21)-(23). For the final step, we define the set of nodes V'' to account for trivial cliques

$$V'' := \{i : i \in V, i \subset c, \text{ for some } c \in \mathcal{C}, |c| = 1\}.$$

```
function $Map(G_c(V); \bar{C}, \mathcal{C})$
    $\mathcal{N}(v; G)$ = Neighborhood of $v$ in $G$
    Initialize $G' \leftarrow G_c$, $V_c \leftarrow \emptyset$, $n \leftarrow |V|$
    for $j \leftarrow 0, |\mathcal{C}| - 1$ do                          ▷ Let $V$ and $\mathcal{C}$ be ordered
        if $|c_j| > 1$ then
            $V_c \leftarrow v_{n-1+j}$, Add new node $v_{n-1+j}$ to $G'$,
            for all $v_i \subset c_j$ do
                Add node $v_i$ to $\mathcal{N}(v_{n-1+j}; G')$
                $\bar{C}(v_{n-1+j}, v_i; G') \leftarrow \sum_{v_k \subset c_j, k \neq i} \bar{C}(v_i, v_k; G_c)$
            end for
        else
            $V_c \leftarrow v_i$, for $v_i \subset c_j$                  ▷ For trivial cliques
        end if
    end for
return $G'$, $V_c$
end function
```

Figure 11: $Map(G_c; \bar{C}, \mathcal{C})$ adds nodes corresponding to each non-trivial clique and returns the expanded graph G' and node set representing cliques V_c.

The set of clique representative nodes is $V_c{:=}V' \cup V''$, the set of newly added virtual nodes and the trivial cliques. We now write the equivalent IP which is the Steiner tree with the set of clique representatives V_c and the fusion center v_0 as the terminals,

$$\frac{1}{2} \min_{\mathbf{x}} \quad \sum_{i,j \in V} x(i,j) \bar{C}(i,j), \tag{30}$$

$$s.t. \quad \sum_{i \in S, j \notin S} x(i,j) \geq 1, \forall S \subset V \cup V' \text{ separating } \{V_c \cup v_0\}, \quad x(i,j) \in \{0,1\}. \tag{31}$$

The equivalence holds since in the above Steiner tree, each clique representative node $v_c \in V'$ has to be connected to at least one clique member and hence, the local processor assignment constraint in (21) is satisfied, and the constraint (31) which ensures that all the terminals $V' \cup v_0$ are connected implies that all the processors and the fusion center are connected and hence, the constraint in (22) is satisfied. Hence, the optimal solution to minimum cost routing for inference is a Steiner tree on the transformed graph $Map(G_c)$.

In order to obtain the fusion scheme, we need another transformation after finding the Steiner tree in (30) on the transformed graph $Map(G_c)$. We first direct the Steiner tree towards the fusion center, denoted by DST. The reverse mapping *RevMap*(DST) in Fig.12 assigns the unique immediate successor of every clique-representative node v_c in DST as the processor of the clique c. The edges from the representative nodes in DST are replaced by links in the metric closure from other clique members to the processor and added to the forwarding subgraph of the fusion scheme. All other edges, not belonging to representative nodes in DST, are assigned as the aggregation subgraph.

```
function $RevMap(G'; V_c, V, \mathcal{C})$
    $\mathcal{N}_s(v; G)$, $\mathcal{N}_p(v; G)$ = Imm. successor, predecessor of $v$
    Initialize $G \leftarrow G'$, $n \leftarrow |V|$
    for all $v_j \in V_c$ do
        if $j > n - 1$ then
            $k \leftarrow j - n + 1$,
            $Proc(c_k) \leftarrow \mathcal{N}_s(v_j; G')$, for $c_k \in \mathcal{C}$,
            $V_j \leftarrow c_k \backslash Proc(c_k)$, Replace $< v_j, Proc(c_k) >$ in $G$ with edges $< V_j, Proc(c_k) >$, mark them
            if $\mathcal{N}_p(v_j; G) \neq \emptyset$ then Replace $< \mathcal{N}_p(v_j), v_j >$ in $G$ with edges $< \mathcal{N}_p(v_j), Proc(c_k) >$
            end if
        else
            $Proc(c_l) \leftarrow v_j$, for $v_j \subset c_l$                    ▷ For trivial cliques
        end if
    end for
    FG $\leftarrow$ Marked edges of $G$, AG $\leftarrow G\backslash$FG
    $\Gamma \leftarrow \{Proc, \mathrm{FG}, \mathrm{AG}\}$
return $\Gamma$
end function
```

Figure 12: $RevMap(G; V_c, V, \mathcal{C})$ maps tree G' to fusion scheme Γ with processor assignment $Proc$, forwarding and aggregation subgraphs FG, AG.

In the above discussion, we have shown that the optimal solution is a Steiner tree involving transformations Map and $RevMap$, summarized in Fig.10. We now prove in addition that the above Steiner-tree reduction is approximation-factor preserving. To this end, we state the conditions under which the reduction preserves the approximation ratio [38, A.3.1].

Definition 4 (Approximation-factor preserving reduction) *Let Π_1 and Π_2 be two minimization problems, with OPT_{Π_i} denoting the values of their optimal solutions. An approximation factor preserving reduction from Π_1 to Π_2 consists of two polynomial time algorithms, f and g, such that,*

- *for any instance I_1 of Π_1, $I_2 = f(I_1)$ is an instance of Π_2 such that*

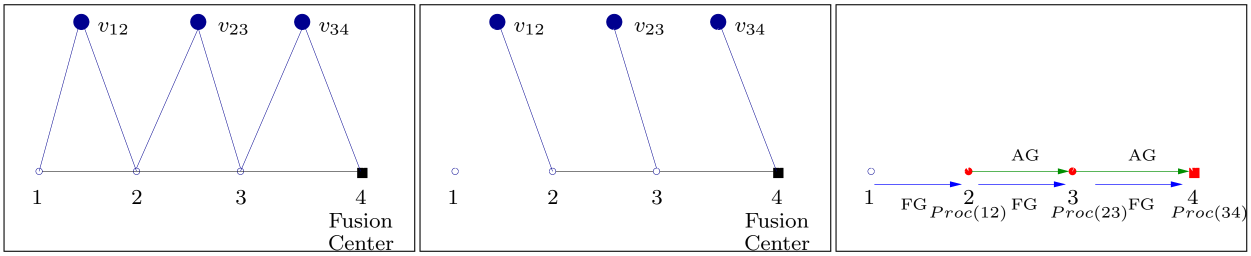
$$OPT_{\Pi_2}(I_2) \leq OPT_{\Pi_1}(I_1). \tag{32}$$

- *for any solution t of I_2, $s = g(I_1, t)$ is a solution of I_1 such that*

$$obj_{\Pi_1}(I_1, s) \leq obj_{\Pi_2}(I_2, t). \tag{33}$$

We now note that AggApprox results in a feasible fusion and runs in polynomial time since there are polynomial number of cliques. For any feasible solution to Steiner tree, replacement of links in line 9 of $RevMap$ in Fig.12 reduces the sum cost, and hence, (33) holds.

The approximation-ratio preserving Steiner tree reduction implies that any approximation algorithm for Steiner tree provides the same approximation ratio for minimum cost fusion, when



(a) Expanded communication graph $Map(G_c)$ with costs under different scenarios of processor selection.

(b) Steiner tree on $Map(G_c)$ with nodes v_{12}, v_{23}, v_{34} and fusion center 4 as terminals.

(c) Fusion scheme with processor assignment, forwarding and aggregation subgraphs after $RevMap$.

Figure 13: Minimum cost fusion through Steiner reduction for chain dependency graph.

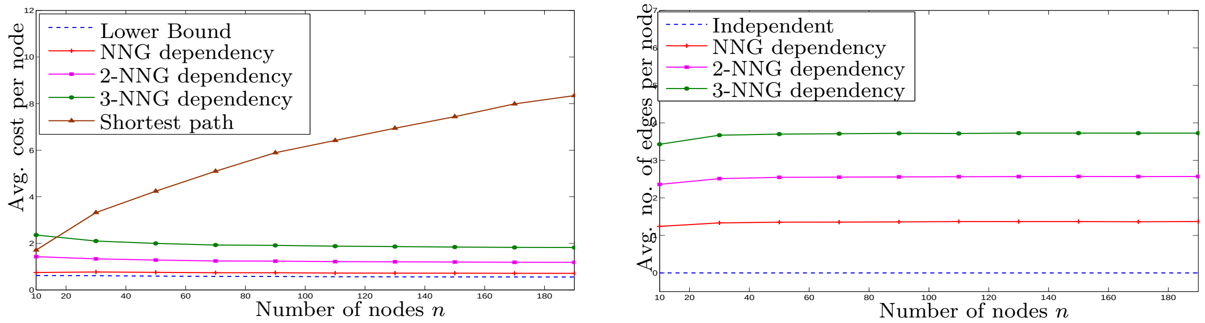
applied with the above transformations. Since currently the best known ratio for Steiner tree is 1.55, it is also the best possible approximation for minimum cost fusion for inference.

4.8 Chain Dependency Graph

We now illustrate the optimal fusion scheme through Steiner-tree reduction for the simple example of a chain dependency graph in Fig.13, where the link communication costs and the metric closure are implicit and not shown. For this simple example, we can intuitively see that the optimal scheme first forwards raw data in the direction of fusion center. Upon computing the potential functions at the processors, the values are added along the chain, starting with the farthest processor. In Fig.13c, this optimal fusion scheme with forwarding and aggregation subgraphs is shown along with the values transported along the links. We now illustrate that the Steiner tree with transformations provides the same optimal solution. In Fig.13a, the expanded communication graph $Map(G_c)$ is shown with added clique-representative nodes and edges. The added edges represent the costs in the forwarding subgraph on choosing a node as a processor. In Fig.13b, the optimal Steiner tree on the expanded graph is shown with the clique representative nodes and the fusion center as terminals. Using $RevMap$, the Steiner tree is mapped to a fusion scheme by first directing the tree towards the fusion center, and then, assigning the immediate successor of clique representative nodes as processors. Hence, the member closer to the fusion center is chosen as the processor in this example. The edges from clique representative nodes are replaced with forwarding subgraph edges, and we can see that the costs are conserved. The remaining edges in the Steiner tree form the aggregation subgraph. Hence, the $RevMap$ operation provides the optimal fusion scheme shown in Fig.13c.

4.9 Discussion

We now plot some simulation results in Fig.14. We see that savings due to aggregation are considerable compared to shortest-path routing for k-nearest neighbor graphs (k-NNG), at low values of k. These graphs are probably the best candidates, after the independent data case, for in-network processing of the likelihood function. We also observe that there is direct correspondence between the number of cliques and the routing cost for fusion. Hence, it appears that the number of cliques is a good measure for judging the effectiveness of in-network processing. The gap between the



(a) Avg. routing cost per node.

(b) Avg. no. of Edge potentials calculated.

Figure 14: Simulation results for k nearest-neighbor dependency graphs. Uniform random placement of nodes. 500 runs. Constant density node placement. Routing Cost on link $(i, j) \propto \text{dist}(i, j)^2$.

heuristics and the lower bound, represents the overhead arising due to correlation. A dense dependency graph has high routing costs due to the complexity of its likelihood function. This is unlike the case of compression with the aim of routing all the raw data to a destination, where a dense dependency graph (more correlation) implies redundancy and hence, reduction in routing costs.

The use of localized processing constraint and unicast mode of communication are crucial to obtaining the above Steiner-tree reduction. They lead to the separation of costs of routing raw measurements (in the forwarding subgraph) to compute different potential functions. On the other hand, in the absence of these constraints, the edge costs in the forwarding graph are no longer independent, and finding the optimal scheme requires the use of hyper-edges. However, once a scheme is designed under the unicast setup, the broadcast nature of the wireless medium could be exploited to further reduce costs by broadcasting raw data from each node to all its processors.

We have so far considered minimum cost routing for optimal inference. A relaxation of this problem is where we only select a subset of measurements for routing and fusion, and we aim to achieve optimal tradeoff between routing costs and end detection performance. This problem requires first the characterization of the detection performance, and one possibility is to use the detection error exponent, which is the asymptotic rate of exponential decay of error probability. It will be interesting to explore if this problem has reduction to well known optimization problems, as it turned out in the case of optimal inference with local processing.

5 Conclusion and Future Work

In this chapter, we have presented an instance of cross-layer design where information from the application layer is used to reduce the routing costs for a statistical inference application. Our approach combines the rich fields of statistical inference, graphical models and approximation algorithms for network design. The joint study of these rich fields has so far been only sparsely explored, and this chapter is an effort in this direction. We exploit the data reduction in the sufficient statistic from the statistical inference literature to reduce the routing costs and formulate the minimum cost fusion scheme that computes and delivers the likelihood function to the fusion center. We employ the Markov random field model for spatial correlation and obtain the likelihood function as the sum of potential functions over the cliques of the MRF from the famous Hammersley-Clifford theorem. This structure of the likelihood function enables a two-stage in-network processing and delivery scheme. In the first stage, a processor is selected for each clique which locally computes the potential function from the raw data. In the second stage, the values at the processors are summed up and delivered to the fusion center. We employ the machinery of approximation algorithms to prove a Steiner tree reduction, enabling us to use any Steiner tree approximation algorithm for minimum cost fusion. Our simulations show a significant saving in cost due to in-network processing compared to routing all the data to the fusion center for proximity-based sparse dependency graph models. Our results demonstrate that inference in sensors networks can no longer be treated as well separated problems in signal processing and networking.

Bibliographic Notes

Markov random fields, also known as conditional auto-regressions (CAR), were introduced by Besag [14, 15]. Detailed exposition on the MRF can be found in [45–47]. For use of the MRF model in sensor networks, see [18], which deals with belief propagation (BP), also known as the sum-product algorithm. It has been applied to sensor-network applications such as multi-target multi-sensor data association and to self localization, in [48] and [49]. However, the goal of belief propagation is to find the marginal pdf or the maximum-posterior-marginal (MPM) estimator locally, in contrast to our goal of obtaining an optimal global decision at the fusion center. In [50], a dynamic-programming approach to resource management for object tracking is proposed. However, the possibility of data fusion, enroute, is not considered. In [51, 52], a decision-theoretic approach to local inference with single bit communication is considered and the network topology is predefined by a directed acyclic graph. Another local inference problem is consensus propagation [53], which is an asynchronous distributed protocol for averaging numbers across a network and has been applied to sensor networks in [54, 55]. The requirement of every node knowing the final value of the function is imposed.

It is beyond the scope of this chapter to provide an extensive review of the works on routing. An overview of routing for mobile-wireless networks can be found in a number of surveys [56]. Correlated data gathering has been considered in [57–61]. But these schemes focus on compression, with the aim of routing all the measurements to a designated sink. Efficient aggregation schemes have been studied in [21, 62–65], but without taking into account the spatial correlation among the measurements. For example, in [65], it is assumed that multiple incoming packets at a node can be processed to a single outgoing packet; this holds only for some special functions such as sum, maximum etc. A survey of in-network processing of various functions may be found in [66,67]. In [68], a link metric for detection is proposed based on the model of one-dimensional Gauss-Markov random process.

References

[1] D. P. Bertsekas and R. Gallager, *Data Networks.* Prentice Hall, 1992.

[2] R. Koetter and M. Medard, “An algebraic approach to network coding,” *IEEE/ACM Tran. on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[3] H. V. Poor, *An Introduction to Signal Detection and Estimation.* New York: Springer-Verlag, 1994.

[4] Y. Sung, S. Misra, L. Tong, and A. Emphremides, “Cooperative Routing for Signal Detection in Large Sensor Networks,” *IEEE J. Select. Area Comm.*, vol. 25, no. 2, pp. 471–483, 2007.

[5] Y. Sung, L. Tong, and H. Poor, "Neyman-Pearson Detection of Gauss-Markov Signals in Noise: Closed-Form Error Exponent and Properties," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1354–1365, April 2006.

[6] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *VLDB*, 2004.

[7] R. Cristescu and M. Vetterli, "On the optimal density for real-time data gathering of spatio-temporal processes in sensor networks," in *IPSN*, 2005, pp. 159–164.

[8] D. Marco, E. Duarte-Melo, M. Liu, and D. Neuhoff, "On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data," in *Proc. IPSN*, 2003, pp. 1–16.

[9] S. Yoon and C. Shahabi, "The Clustered Aggregation technique leveraging spatial and temporal correlations in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 3, no. 1, 2007.

[10] J. Faruque and A. Helmy, "RUGGED: RoUting on finGerprint Gradients in sEnsor Networks," in *IEEE/ACS Intl. Conf. on Pervasive Services (ICPS)*, 2004, pp. 179–188.

[11] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *Proceedings of the third international symposium on Information processing in sensor networks*, 2004, pp. 28–35.

[12] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann, "An evaluation of multi-resolution search and storage in resource-constrained sensor networks," in *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2003.

[13] A. Jindal and K. Psounis, "Modeling spatially correlated data in sensor networks," *ACM Tran. on Sensor Net.*, vol. 2, no. 4, pp. 466–499, 2006.

[14] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of Royal Stat. Soc.*, vol. 36, no. B, pp. 192–225, 1974.

[15] ——, "Statistical analysis of non-lattice data," *The Statistician*, vol. 24, no. 3, pp. 179–195, 1975.

[16] J. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," *Unpublished manuscript*, 1971.

[17] P. Clifford, "Markov random fields in statistics," *Disorder in Physical Systems*, pp. 19–32, 1990.

[18] M. Cetin, L. Chen, J. Fisher, A. Ihler, O. Kreidl, R. Moses, M. Wainwright, J. Williams, and A. Willsky, "Graphical Models and Fusion in Sensor Networks," in *Wireless sensor networks: signal processing & comm. perspectives.* J. Wiley & Sons, 2007, ch. 9, pp. 215–250.

[19] S. Li, *Markov random field modeling in computer vision.* Springer-Verlag London, 1995.

[20] N. Cressie, *Statistics for spatial data.* J. Wiley, 1993.

[21] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Tran. on W. Comm.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[22] T. Kwon and M. Gerla, "Clustering with power control," in *Military Communications Conference Proceedings, IEEE*, vol. 2, 1999.

[23] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Tran. on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[24] S. Sanghavi, V. Tan, and A. Willsky, "Learning Graphical Models for Hypothesis Testing," in *IEEE Workshop on Stat. Signal Proc.*, 2007.

[25] S. Lauritzen, *Graphical models: Clarendon Press.* Clarendon Press, 1996.

[26] K. Jung and D. Shah, “Approximate message-passing inference algorithm,” in *IEEE ITW*, 2007, pp. 224–229.

[27] J. Pearl, *Probabilistic Reasoning in Intelligent Systems—Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[28] D. Eppstein, “All maximal independent sets and dynamic dominance for sparse graphs,” in *Proc. of ACM-SIAM symp. on discrete algorithms*, 2005, pp. 451–459.

[29] N. Cressie and S. Lele, “New Models for Markov Random Fields,” *Journal of Applied Probability*, vol. 29, no. 4, pp. 877–884, 1992.

[30] R. Cowell, *Probabilistic Networks and Expert Systems.* Springer, 1999.

[31] E. Dynkin, “Necessary and sufficient statistics for a family of probability distributions,” *Tran. Math, Stat. and Prob.*, vol. 1, pp. 23–41, 1961.

[32] B. Wu and K. Chao, *Spanning Trees and Optimization Problems.* Chapman & Hall, 2004.

[33] W. P. Tay, J. N. Tsitsiklis, and M. Z. Win, “Data fusion trees for detection: Does architecture matter?” *IEEE Tran. on Information Theory*, 2007, submitted for publication.

[34] ——, “On the sub-exponential decay of detection error probabilities in long tandems,” *IEEE Tran. on Information Theory*, 2007, submitted for publication.

[35] L. Devroye, “The expected size of some graphs in computational geometry.” *Comp. & math. with app.*, vol. 15, no. 1, pp. 53–64, 1988.

[36] R. Gallager, P. Humblet, and P. Spira, “A Distributed Algorithm for Minimum-Weight Spanning Trees,” *ACM Tran on Prog. Languages and Sys. (TOPLAS)*, vol. 5, no. 1, pp. 66–77, 1983.

[37] P. Humblet, “A Distributed Algorithm for Minimum Weight Directed Spanning Trees,” *IEEE Trans. on Communications*, vol. 31, no. 6, pp. 756–762, 1983.

[38] V. Vazirani, *Approximation Algorithms.* Springer, 2001.

[39] L. Kou, G. Markowsky, and L. Berman, “A fast approximation algorithm for Steiner trees,” *Acta Informatica*, vol. 15, pp. 141–145, 1981.

[40] G. Robins and A. Zelikovsky, “Improved Steiner tree approximation in graphs,” in *Proc. of ACM-SIAM symposium on Discrete algorithms*, 2000, pp. 770–779.

[41] G. Reich and P. Widmayer, “Beyond steiner’s problem: a vlsi oriented generalization,” in *Proc. of Intl. workshop on Graph-theoretic concepts in computer science*, 1990, pp. 196–210.

[42] N. Garg, G. Konjevod, and R. Ravi, “A polylogarithmic approximation algorithm for the group steiner tree problem,” *J. Algorithms*, vol. 37, no. 1, pp. 66–84, 2000.

[43] C. S. Helvig, G. Robins, and A. Zelikovsky, “Improved approximation bounds for the group steiner problem,” in *DATE ’98: Proceedings of the conference on Design, automation and test in Europe*, 1998, pp. 406–413.

[44] A. Anandkumar, L. Tong, A. Swami, and A. Ephremides, “Minimum Cost Data Aggregation with Localized Processing for Statistical Inference,” in *Proc. of IEEE INFOCOM*, Phoenix, USA, April 2008, pp. 780–788.

[45] P. Brémaud, *Markov Chains: Gibbs fields, Monte Carlo simulation, and queues.* Springer, 1999.

[46] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications.* London: Chapman and Hall, 2005.

[47] X. Guyon, *Random Fields on a Network: Modeling, Statistics, and Applications.* Springer, 1995.

[48] L. Chen, M. Wainwright, M. Cetin, and A. Willsky, "Multitarget-multisensor data association using the tree-reweighted max-product algorithm," in *Proceedings of SPIE*, vol. 5096, 2003, pp. 127–138.

[49] A. Ihler, J. Fisher III, R. Moses, and A. Willsky, "Nonparametric Belief Propagation for Self-Localization of Sensor Networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 4, pp. 809–819, 2005.

[50] J. Williams, J. Fisher III, and A. Willsky, "An Approximate Dynamic Programming Approach to a Communication Constrained Sensor Management Problem," in *Intl. Conf. on Information Fusion*, vol. 1, 2005.

[51] O. Kreidl and A. Willsky, "Inference with Minimal Communication: a Decision-Theoretic Variational Approach," in *Advances in Neural Information Processing Systems*, 2006.

[52] ——, "Efficient message-passing algorithms for optimizing decentralized detection networks," in *IEEE Conf. on Decision and Control*, 2006.

[53] C. Moallemi and B. Van Roy, "Consensus Propagation," vol. 52, no. 11, pp. 4753–4766, 2006.

[54] A. Dimakis, A. Sarwate, and M. Wainwright, "Geographic gossip: efficient aggregation for sensor networks," in *Proceedings of the fifth international conference on Information processing in sensor networks*, 2006, pp. 69–76.

[55] R. Olfati-Saber, E. Franco, E. Frazzoli, and J. Shamma, "Belief consensus and distributed hypothesis testing in sensor networks," in *Workshop on Network Embedded Sensing and Control, Notre Dame University, Oct*, 2005.

[56] K. Akkaya and M. Younis, "A Survey of Routing Protocols in Wireless Sensor Networks," *Elsevier Adhoc Networks*, vol. 3, pp. 325–349, 2005.

[57] R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer, "Network correlated data gathering with explicit communication: NP-completeness and algorithms," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 1, pp. 41–54, 2006.

[58] P. von Rickenbach and R. Wattenhofer, "Gathering correlated data in sensor networks," in *Joint workshop on Foundations of mobile computing*, 2004, pp. 60–66.

[59] A. Goel and D. Estrin, "Simultaneous Optimization for Concave Costs: Single Sink Aggregation or Single Source Buy-at-Bulk," *Algorithmica*, vol. 43, no. 1, pp. 5–15, 2005.

[60] H. Gupta, V. Navda, S. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," in *Proc. of ACM Intl. symposium on Mobile ad hoc networking and computing*, 2005, pp. 402–413.

[61] A. Scaglione and S. Servetto, "On the Interdependence of Routing and Data Compression in Multi-Hop Sensor Networks," in *Proc. MobiCom 2002*, Atlanta, Georgia, September 2002.

[62] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TinyDB: an acquisitional query processing system for sensor networks," *ACM Transactions on Database Systems*, vol. 30, no. 1, pp. 122–173, 2005.

[63] J. Gehrke and S. Madden, "Query processing in sensor networks," *IEEE Pervasive Computing*, vol. 03, no. 1, pp. 46–55, 2004.

[64] C. Intanagonwiwat, R. Govindan, and D. Esterin, "Directed Diffusion : A Scalable and Robust Paradigm for Sensor Networks," in *Proc. 6th ACM/Mobicom Conference*, Boston,MA, 2000, pp. pp 56–67.

[65] B. Krishnamachari, D. Estrin, and S. Wicker, "Modeling data centric routing in wireless sensor networks," in *INFOCOM*, New York, 2002.

[66] A. Giridhar and P. Kumar, "Toward a theory of in-network computation in wireless sensor networks," *Comm. Magazine, IEEE*, vol. 44, no. 4, pp. 98–107, 2006.

[67] R. Rajagopalan and P. Varshney, “Data aggregation techniques in sensor networks: A survey,” *Comm. Surveys & Tutorials, IEEE*, vol. 8, no. 4, pp. 48–63, 2006.

[68] S. Misra, L. Tong, and A. Ephremides, “Application dependent shortest path routing in ad-hoc sensor networks,” in *Wireless sensor networks: signal processing & comm. perspectives.* J. Wiley & Sons, 2007, ch. 11, pp. 277–310.

Animashree Anandkumar received the B.Tech degree in Electrical Engineering from Indian Institute of Technology Madras, Chennai, India in 2004. She is currently pursuing her Ph.D. degree in Electrical Engineering at Cornell University, Ithaca, NY. She is a member of the Adaptive Communications and Signal Processing Group (ACSP) at Cornell University.

Anima received the Fran Allen IBM Ph.D fellowship for the year 2008-09, presented annually to one female Ph.D. student in conjunction with the IBM Ph.D. Fellowship Award, in recognition of her technical accomplishments, as well as her commitment to mentoring and community building. She was named a finalist for the Google Anita-Borg Scholarship 2007-08. She received the Student Paper Award at the 2006 International Conference on Acoustic, Speech and Signal Processing (ICASSP). Her research interests are in the broad area of statistical-signal processing, information theory and networking. Specifically, she has been working on detection and estimation, asymptotic analysis and routing, in the context of wireless sensor networks.

Anthony Ephremides holds the Cynthia Kim Professorship of Information Technology at the Electrical and Computer Engineering Department of the University of Maryland in College Park where he holds a joint appointment at the Institute for Systems Research, of which he was among the founding members in 1986. He obtained his PhD in Electrical Engineering from Princeton University in 1971 and has been with the University of Maryland ever since.

He has held various visiting positions at other Institutions (including MIT, UC Berkeley, ETH Zurich, INRIA, etc) and co-founded and co-directed a NASA-funded Center on Satellite and Hybrid Communication Networks in 1991. He has been the President of Pontos, Inc, since 1980 and has served as President of the IEEE Information Theory Society in 1987 and as a member of the IEEE Board of Directors in 1989 and 1990. He has been the General Chair and/or the Technical Program Chair of several technical conferences (including the IEEE Information Theory Symposium in1991 and 2000, the IEEE Conference on Decision and Control in 1986, the ACM Mobihoc in 2003, and the IEEE Infocom in 1999). He has served on the Editorial Board of numerous journals and was the Founding Director of the Fairchild Scholars and Doctoral Fellows Program, a University-Industry Partnership from 1981 to 1985.

He has received the IEEE Donald E. Fink Prize Paper Award in 1991 and the first ACM Achievement Award for Contributions to Wireless Networking in 1996, as well as the 2000 Fred W. Ellersick MILCOM Best Paper Award, the IEEE Third Millennium Medal, the 2000 Outstanding Systems Engineering Faculty Award from the Institute for Systems Research, and the Kirwan Faculty Research and Scholarship Prize from the University of Maryland in 2001, and a few other official recognitions of his work. He also received the 2006 Aaron Wyner Award for Exceptional Service and Leadership to the IEEE Information Theory Society.

He is the author of several hundred papers, conference presentations, and patents, and his research interests lie in the areas of Communication Systems and Networks and all related disciplines, such as Information Theory, Control and Optimization, Satellite Systems, Queueing Models, Signal Processing, etc. He is especially interested in Wireless Networks and Energy Efficient Systems.

Lang Tong is the Irwin and Joan Jacobs Professor in Engineering at Cornell University, Ithaca, NY. Prior to joining Cornell University, he was on faculty at the West Virginia University and the University of Connecticut. He was also the 2001 Cor Wit Visiting Professor at the Delft University of Technology. Lang Tong received the B.E. degree from Tsinghua University, Beijing, China, in 1985, and M.S. and Ph.D. degrees in electrical engineering in 1987 and 1991, respectively, from the University of Notre Dame, Notre Dame, Indiana. He was a Postdoctoral Research Affiliate at the Information Systems Laboratory, Stanford University in 1991.

Lang Tong is a Fellow of IEEE. He received the 1993 Outstanding Young Author Award from the IEEE Circuits and Systems Society, the 2004 best paper award (with Min Dong) from IEEE Signal Processing Society, and the 2004 Leonard G. Abraham Prize Paper Award from the IEEE Communications Society (with Parvathinathan Venkitasubramaniam and Srihari Adireddy). He is also a coauthor of five student paper awards. He received Young Investigator Award from the Office of Naval Research.

Lang Tong's research is in the general area of statistical signal processing, wireless communications and networking, and information theory. He has served as an Associate Editor for the IEEE Transactions on Signal Processing, the IEEE Transactions on Information Theory, and IEEE Signal Processing Letters.

Ananthram Swami received the B.Tech. degree from IIT-Bombay; the M.S. degree from Rice University, and the Ph.D. degree from the University of Southern California (USC), all in Electrical Engineering. He has held positions with Unocal Corporation, USC, CS-3 and Malgudi Systems. He was a Statistical Consultant to the California Lottery, developed a Matlab-based toolbox for non-Gaussian signal processing, and has held visiting faculty positions at INP, Toulouse. He is currently with the US Army Research Laboratory where his work is in the broad area of signal processing, wireless communications, sensor and mobile ad hoc networks.

He has served as associate editor of the IEEE Transactions on Wireless Communications, IEEE Signal Processing Letters, IEEE Transactions on Circuits & Systems-II, IEEE Signal Processing Magazine, and IEEE Transactions on Signal Processing. He was co-guest editor of a 2004 Special Issue (SI) of the IEEE Signal Processing Magazine (SPM) on 'Signal Processing for Networking', a 2006 SPM SI on 'Distributed signal processing in sensor networks', a 2006 EURASIP JASP SI on Reliable Communications over Rapidly Time-Varying Channels', a 2007 EURASIP JWCN SI on 'Wireless mobile ad hoc networks', and is the Lead Editor for a 2008 IEEE JSTSP SI on "Signal Processing and Networking for Dynamic Spectrum Access". He is a co-editor, with Qing Zhao, Yao-Win Hong and Lang Tong, of the 2007 Wiley book "Wireless Sensor Networks: Signal Processing & Communications Perspectives."