An Information Theoretic Framework for Field Monitoring Using Autonomously Mobile Sensors

Hany Morcos[1], George Atia[2], Azer Bestavros[1], and Ibrahim Matta[1]

[1] Computer Science Department, Boston University, Boston, MA,
[2] Electrical and Computer Engineering Department, Boston University, Boston, MA

Abstract. We consider a mobile sensor network monitoring a spatio-temporal field. Given limited caches at the sensor nodes, the goal is to develop a distributed cache management algorithm to efficiently answer queries with a known probability distribution over the spatial dimension. First, we propose a novel distributed information theoretic approach assuming knowledge of the distribution of the monitored phenomenon. Under this scheme, nodes minimize an entropic utility function that captures the average amount of uncertainty in queries given the probability distribution of query locations. Second, we propose a correlation-based technique, which only requires knowledge of the second-order statistics, relaxing the stringent constraint of a priori knowledge of the query distribution, while significantly reducing the computational overhead. We show that the proposed approaches considerably improve the average field estimation error. Further, we show that the correlation-based technique is robust to model mismatch in case of imperfect knowledge of the underlying generative correlation structure.

1 Introduction

Early sensor network research assumed that sensors are static with very low computation and storage capabilities, and once deployed, these nodes are not likely to be recharged or moved. Hence, once separated from the network (*e.g.,* due to failure of nodes on the path to the rest of the network), nodes will remain disconnected until their batteries die. Sensor network technologies have matured to the degree that they are expected to be embedded in many platforms. Some of these platforms are mobile, *e.g.,* automobiles, handheld devices and wearable computers, giving rise to a rather new paradigm for sensor networks, which allows for the consideration of mobility, including the possibility of leveraging it for new classes of sensor network applications.

A paradigm, in which sensor networks are mobile, not only changes many traditional sensor network assumptions (*e.g.*, node isolation may be only temporary due to mobility), but also it gives rise to new applications, or to old applications under new settings. One such application is field monitoring. An extensive body of research studied this problem in the context of static sensor networks [8, 21, 17, 13, 23, 7]. Dense

[§] This work has been partially supported by a number of National Science Foundation grants, including CISE/CSR Award #0720604, ENG/EFRI Award #0735974, CISE/CNS Award #0524477, CNS/NeTS Award #0520166, CNS/ITR Award #0205294, and CISE/EIA RI Award #0202067.

node deployment is usually assumed. A unique party in the network (*i.e.,* the sink) is assumed to be responsible for posing queries to the rest of the network. Flooding (whether network-wide or limited) is leveraged to discover the best forwarding paths to and from the sink. Lack of change in the network topology allows these paths to be useful for handling multiple queries, validating the cost of flooding.

Besides mobility, the field monitoring setting we consider in this paper is different from the above scenario. Specifically, sensor nodes are not viewed as reactive elements whose sole role is to sample a single location and respond to queries about this specific location. Rather, we view sensors as being embedded or attached to larger entities (*e.g.*, cars and handheld devices), which constitute points of interaction between the system and its users. As such, users may pose queries to the system, and get replies from the system through these points of interaction (nodes). As an example for this setup, consider a firefighter's backpack that contains a number of sensors (*e.g.,* temperature sensor, smoke sensor, carbon-monoxide sensor, *etc.*), along with a head-mounted display and a keyboard to allow interaction between each firefighter and the system [1]. In such a system, sensors could sample the environment in which firefighters work. Collected samples should be managed and stored in order to satisfy queries issued by firefighters to the system. A query can target any location in the scene, not only locations sampled by the inquirer. For example, if one firefighter needs to go to some location in the scene, then measurements of temperature, smoke levels, and concentration of carbon monoxide would prove valuable to this firefighter. Thus, the goal of the system is to provide an accurate estimation of the phenomenon of interest at the given query location. A defining characteristic of this system is the mobility pattern of the mobile hosts (firefighters in this example). This pattern is not governed by the need to optimize the system performance, rather it is governed by an overarching mission (*e.g.,* the need to save someone trapped in a room, or constraints due to how the fire progresses). This same setting applies equally well to a group of soldiers in a battlefield, or a group of researchers performing a study in some urban field.

Another important factor in the paradigm we consider in this paper is that, users may have specific preferences when posing queries to the system. Specifically, the spatial distribution of interest over the field might be skewed as opposed to uniform (*i.e.,* there might exist some zones in the field that users are likely to inquire about more frequently – *e.g.*, near exits). Also, different phenomena of interest (*e.g.,* temperature, and carbon monoxide) might have different interest distributions. Knowledge of such distributions can be leveraged to optimize the system performance.

We assume that in such systems the storage space of mobile nodes allotted to *each* phenomenon of interest is limited. This is a realistic assumption for two reasons: 1) considering the fact that data from different phenomena share the same storage space (or cache). Adding more sensor types increases the number of phenomena that the system is able to handle, but also increases contention over the limited memory available for storage. 2) As we alluded above the type of applications we target are *parasitic* applications; in the sense that these applications exploit mobility of the host and its resources (*e.g.,* storage of a firefighter wearable system) to provide some service. Hence, it is conceivable that, although the host might have plenty of storage, our target applications will be allowed access to a limited fraction of this storage. These two reasons motivate the need for a cache management algorithm. We assume that samples from different phenomena are independent, hence, solving the problem for one phenomenon is enough.

To this end, in this paper we propose two cache management algorithms for tackling this problem. Our techniques aim to minimize some utility function that captures the average amount of uncertainty in queries given the distributional characteristics of query locations. Our contributions are as follows:

- Assuming knowledge of the entire spatio-temporal distribution of the target phenomenon, we develop an information-theoretic framework to optimize the cache content, and provide accurate answers to queries (Section 3).
- We propose a different approach based on optimizing a correlation-based function relaxing the stringent constraint of full distribution knowledge. We develop a strategy that only requires knowledge of the second order statistics of the phenomenon of interest. Furthermore, this technique lowers the required computational complexity (Section 4).
- We provide extensive performance evaluation of our techniques, showing (and quantifying the impact of) the various factors and parameters that affect performance (Section 6). We, also, study the robustness of the technique developed in Section 4 to model mismatch in case of imperfect knowledge of the correlation structure.

The rest of the paper is organized as follows. In Section 2 the setup and problem definition are provided. Details of the proposed techniques are presented in Sections 3 and 4 together with an analysis of their corresponding computational complexity. Based on these two cache management algorithms, we show how to design a cooperative scheme in Section 5, where nodes benefit from samples cached at their neighbors to obtain more accurate query estimates. We then present in Section 6 an evaluation of the cache management strategies for two phenomena generated using different processes. We provide a summary of related work in Section 7, discuss future work and conclude the paper in Section 8.

2 Problem Definition

We start with the problem definition along with a description of the system goal. The setup, system parameters, and notation we use are as follows:

- The system consists of n autonomously mobile nodes (*i.e.*, node mobility is not controlled by the system).
- Each node has a cache of size c.
- The nodes move in a field $\mathcal{F}$ with area $A = L \times L$.
- While roaming the field, sensor nodes sample a target phenomenon and this process continues for T time units.
- Location information is accessible to the sensor nodes, such that they can associate each sample with the location where it was collected.

We leave the relaxation of this assumption to future work on this problem.

- We use capital letters to represent random variables and small letters to represent realizations of these random variables.
- $V_{\ell,t}$ is a random variable that represents the value of the field phenomenon at location ℓ and time t. $v_{\ell,t}$ denotes a realization of this random variable.
- We use the boldfaced letter $\mathbf{s}_t^i = [s_1, s_2, .., s_c] \in \mathbb{R}^c$ to denote the c-dimensional cache content vector of node i at time t. To simplify notation and since we would be generally referring to any arbitrary node i, we will drop the superscript i, unless it is not clear from the context. Note that any cached sample s_j corresponds to a field value v_{ℓ_j,t_j}, where ℓ_j is the location from which this sample was collected and t_j its corresponding time stamp.
- It is assumed that a query posed at any time instant τ inquiring about location ℓ targets the value of the field phenomenon $v_{\ell,\tau}$.
- The field phenomenon is fully characterized by a space-time multivariate probability distribution $p(\{v_{\ell,t}\}; \ell \in \mathcal{F}, 0 \leq t \leq T)$ with a $L^2 \times T \times L^2 \times T$ correlation matrix R, such that $R(v_{\ell_1,t_1}, v_{\ell_2,t_2})$ represents the correlation between two values of the phenomenon with space-time coordinates (ℓ_1, t_1) and (ℓ_2, t_2), respectively.
- Define the random variable $L(q)$ as the location which query q targets (called the *query target*). We assume that $L(q)$ follows some spatial distribution Q, where $Q(\ell(q))$ is the probability of querying field location $\ell(q)$. Q is assumed to be stationary. Similarly, we use $t(q)$ to denote the time at which query q was posed. Obviously, the best answer to q would be $v_{\ell(q),t(q)}$.

System Goal: After some warm-up time, each node in the system is expected to answer queries about the target phenomenon in the field. The query specifies some field location, the node is expected to provide an estimate of the phenomenon at the query target and the goal is to minimize the mean square estimation error (MSE) of the system's response. Hence, the nodes are required to maintain an efficient cache content to be able to answer queries reliably. In the next sections, we develop different strategies for cache management at the sensor nodes.

3 Information Theoretic Cache Management

In this section we develop an information theoretic strategy via which nodes locally update their caches based on knowledge of the space-time distribution of the phenomenon of interest.

3.1 DEBT Cache Maintenance Strategy

At each time instant, local decisions are made at the mobile nodes concerning which samples to keep, and whether or not a new sample should be acquired at the current location. These decisions are made so as to minimize an entropic utility function that captures the average amount of uncertainty in queries given the probabilistic query target distribution — hence the name of the strategy: Distributed Entropy Based Technique (DEBT). Specifically, at each time instant t, a node i greedily decides in favor of the cache content that minimizes the conditional differential entropy averaged over the query distribution Q, *i.e.*,

$$\begin{aligned} \mathbf{s_t} &= \arg\min h(V_{L(q),t}/\mathbf{s_t}, L(q)) \\ &= \arg\min_{\mathbf{s_t}\in\mathcal{S}_t} \int_{\ell(q)\sim Q} Q(\ell(q)) h(V_{\ell(q),t}/\mathbf{s_t}, \ell(q)) \end{aligned} \quad (1)$$

where, $\mathbf{s_t} \in \mathbb{R}^c$ is the cache content selected by node i at time t, and $h(V_{L(q),t}/\mathbf{s_t}, L(q))$ is the differential entropy of the values of the phenomenon, conditioned on a given cache content, at the possible query locations $\ell(q)$ which follow a spatial distribution Q. $\mathcal{S}_t$ is the set of all possible decisions leading to all possible cache contents at node i at time t which is given by:

$$\mathcal{S}_t = \{\mathbf{s_t} : \mathbf{s_t} \in \mathcal{C}_{c,c+1}(\mathbf{s_{t-1}} \bigcup \{v_{\ell_t,t}\})\} \quad (2)$$

where $\mathcal{C}_{c,c+1}(\mathcal{A})$ denotes all the $(c+1$ choose $c)$ possible combinations of the elements of a set $\mathcal{A}$ and $v_{\ell_t,t}$ denotes the value of the phenomenon at the current location of the i-th node, ℓ_t.

The expression above simply enumerates all the possible cache contents at time t; the options being to drop any of the samples from time $t-1$ and acquiring the new sample at the current location of node i, or just keep the old set of samples.

The intuition behind DEBT is that a node always keeps a cache content that minimizes the uncertainty in the values of the phenomenon (captured by the conditional entropy) given the knowledge of the spatial distribution of the query targets over the field of interest. It might well be true that an old sample taken at a specific location is more valuable, and hence is worth caching than a newer sample taken at a different location given the aggregate effect of the spatial query distribution and the spatio-temporal distribution of the phenomenon.

It is worth mentioning that the computation of $h(V_{\ell(q),t}/\mathbf{s_t})$ (Eq.3 [5]) requires knowledge of the posterior density $p(v_{\ell(q),t}/\mathbf{s_t})$, which can be generally obtained by proper marginalization of the full space-time distribution. For the Gaussian case, this simplifies to a computation of the conditional mean and variance $\mu_{v_{\ell(q),t}/\mathbf{s_t}}$ and $\lambda_{v_{\ell(q),t}/\mathbf{s_t}}$.

$$h(V_{\ell(q),t}/\mathbf{s}) = -\int_{v_{\ell(q),t}} p(v_{\ell(q),t}/\mathbf{s}) \ln p(v_{\ell(q),t}/\mathbf{s}) dv_{\ell(q),t} \quad (3)$$

3.2 Least Square Error (LSE) Query Response Strategy

To answer a posed query q, a node computes an estimate of the phenomenon at the query target given its cache content. Given the knowledge of the space-time distribution, it would be natural to resort to a Bayesian Least Square Estimate (BLSE), which is

Note that the differential entropy $h(V_{L(q),t}/\mathbf{s})$ that we use in the minimization of Equation(1) is conditioned on a given realization of the cache content. That is to say, no averaging is taken over the conditioning random vector since we are dealing with real-time selection of the samples. This is clearly different from the standard quantity $h(V_{L(q),t}/\mathbf{S})$ with $\mathbf{S}$ being a random variable.

given by the conditional expectation of the posterior density, to minimize the mean square estimation error. Hence each node's task is to compute the expected value of the phenomenon at q given its cache content $\mathbf{s}$, that is:

$$\hat{V}_{\ell(q),t(q)} = E[V_{\ell(q),t(q)}/\mathbf{s}] \tag{4}$$

where $\hat{V}_{\ell(q),t(q)}$ is the node estimate. Again we point out that this generally requires the computation of the posterior density $p(v_{\ell(q),t(q)}/\mathbf{s_t})$. Under Gaussian assumptions, the BLSE estimate in Eq.(4) is always linear in the cache content, that is the BLSE is equal to the Linear Least Square Estimate (LLSE). For general distributions, the computational complexity could be reduced if we only restrict ourselves to linear functions of the cache content, *i.e.* LLSE, which would only require knowledge of the second-order statistics of the phenomenon. Note that the LLSE, $\hat{X}_{LLSE}$, of a random variable X with mean μ_X, given a random vector $Y = y$, with mean vector μ_Y is given by [22]:

$$\hat{X}_{LLSE} = \mu_X + \Lambda_{XY}\Lambda_Y^{-1}(y - \mu_Y) \tag{5}$$

where Λ_{XY} denotes the cross-covariance between X and Y, and, Λ_Y is the covariance matrix of the observation vector Y. While the DEBT/LSE techniques outlined in this section are expected to yield accurate performance, they are not practical. Specifically, we note the following two types of limitations on DEBT practicality:

- *Informational Limitations:* DEBT assumes knowledge of the entire distribution of the target phenomenon. Such information may not be always available, or if available (*e.g.*, through historical monitoring of the phenomenon of interest), it may not be accurate.
- *Computational Limitations:* In order to provide optimized decisions about whether or not to sample visited field locations, and how to manage the cache, DEBT calculates the conditional differential entropy of the query distribution Q given any cache setting. This requires performing multiple numerical integration operations, which might not be always suitable due to the limited computational capabilities at the sensor nodes.

This motivates taking a different approach that is less-demanding in terms of knowledge about the spatio-temporal field. In the next section, we propose a more practical (yet quite competitive) strategy that only requires knowledge of the correlation structure, *i.e.*, second-order statistics.

4 Correlation-Based Cache Management

In this section, we propose a Correlation-Based Technique (CBT) as a practical alternative to the DEBT approach presented before.

CBT averts the limitations of DEBT by only assuming knowledge of the space-time correlation structure of the field phenomenon R. Namely, instead of calculating the conditional entropy to make caching decisions, CBT decides which samples to cache using only the correlation structure of the target phenomenon R. Notice that defining R implies only knowledge of the second-order statistics of the target phenomenon, as opposed to knowledge of the entire distribution in case of DEBT. Like DEBT, the crux of the CBT technique is to be able to assign a measure of utility capturing knowledge about the field to any given set of samples $\mathrm{s} = \{s_1, s_2, .., s_c\}$ with respect to the query distribution Q. Then, it retains the set of samples that maximizes the utility. First, we need to assign a measure of utility $u(q, \mathrm{s})$ to a set of samples s with respect to a specific query q with location $\ell(q)$, and time $t(q)$. Then by averaging $u(q, \mathrm{s})$ over the spatial distribution Q, we get a weighted information metric over the entire field, $M(Q, \mathrm{s})$. More specifically, for a query q, we gauge the utility of s with respect to q as follows:

$$u(q, \mathrm{s}) = \frac{Q(\ell(q))}{\varLambda_{q|\mathrm{s}}} \tag{6}$$

Averaging $u(q, \mathrm{s})$ over Q, we get

$$M(Q, \mathrm{s}) = \int_Q u(q, \mathrm{s}) = \int_{\ell \sim Q} \frac{Q(\ell)}{\varLambda_{q|\mathrm{s}}}\, d\ell \tag{7}$$

where $Q(\ell(q))$ is the probability of querying field location $\ell(q)$, and $\varLambda_{q|\mathrm{s}}$ is the conditional covariance of $q|\mathrm{s}$, given by

$$\varLambda_{q|\mathrm{s}} = \varLambda_q - \varLambda_{q,\mathrm{s}} \varLambda_{\mathrm{s}}^{-1} \varLambda_{q,\mathrm{s}}^T \tag{8}$$

where $\varLambda_q$ is the variance of the stationary process, $\varLambda_{q,\mathrm{s}}$ is the cross-covariance between q and s, and $\varLambda_{\mathrm{s}}$ is the covariance matrix of the cache content s. Notice that calculation of $\varLambda_{q|\mathrm{s}}$ only requires knowledge of the correlation matrix R. Then, CBT makes its caching decisions by maximizing the total utility over the choice of possible cache content s (*i.e.,* $\max_{\mathrm{s}} M(Q, \mathrm{s})$).

5 Nodes Cooperation

So far we have described operation of a single node. However, in a mobile network of numerous nodes, cooperation between nodes could be engineered to yield a better performance. In this paper, we limit our attention to cooperation concerning query response. This is done as follows. Whenever a node i gets a query q, i broadcasts q to its direct neighbors. Upon receiving the query, each neighbor j of i estimates its answer based on its local cache content, then, submits the estimate back to i along with a measure of confidence in this answer. Node i performs the same task, and receives query replies from its neighbors. The answer with the highest confidence is used as the query response. In our setting we use the conditional covariance $\varLambda_{q|\mathrm{s}}$ (Equation 8) as the measure of confidence in the estimated answer. The intuition is that a lower conditional covariance corresponds to less uncertainty about the query. Notice that, the radius of flooding the query could be increased to values larger than one (*i.e.,* consult nodes beyond direct neighbors), however, we choose not to do this in order to avoid query flooding and its associated communication overhead.

Also, notice that, while we chose to limit nodes cooperation to the query handling (*i.e.,* estimation) plan, cooperation between nodes could be done on different plans, for example, the sample caching (*i.e.,* decision-making) process. In this case, nodes would take decisions as to which samples to cache and which ones to evict based not only on the contents of local cache, but on the contents of neighboring caches as well. This would require broadcasting the cache content (or a summary of it thereof) to neighbors, which is a costly process in terms of power. Also, performing cooperation on the decision making plan requires more coordination in presence of mobility, since the set of neighbors changes with time. In this paper we evaluate the first option, and leave investigation of the second to future work.

6 Performance Evaluation

In this section we evaluate the performance of the different proposed cache management techniques. We start in Subsection 6.1 with a description of the data generation models we used to generate the input data. In Subsection 6.2, we provide the details of our evaluation methodology. Next, in Subsection 6.3, we introduce the performance metrics we use in our evaluation. Finally, we present the results of our experiments in Subsections 6.4, and 6.5.

6.1 Data Generation model

In this subsection, we describe the two data generation models we used in this study.

Model 1: A Gaussian Phenomenon: In the first model, the underlying space-time distribution of the phenomenon is a multivariate Gaussian. Thus, the field distribution is fully captured by the mean vector and the joint spatio-temporal correlation (STC) matrix R, $L^2 \times T \times L^2 \times T$. To generate the field, we first generate the data to satisfy the spatial correlation using the standard Cholesky decomposition transformation by pre-multiplying a matrix of independent Gaussian random variables by the square root of the desired spatial covariance [18]. Each individual temporal signal associated with a given location is then filtered using a temporal filter to provide the correct spectral shape. This approach results in an STC covariance structure where the off-diagonal blocks are scalings of the diagonal blocks with a scaling factor that depends on the corresponding time lag. Here we note that other methods based on techniques described in [6] could also be used for generation of fields with arbitrary joint space-time correlation.

Model 2: A Random Phenomenon: In the second model, the generated data does not follow a Gaussian distribution. The purpose of this experiment is to study the performance of the CBT technique proposed in Section 4, which only requires knowledge of the second-order statistics, when the underlying field follows an arbitrary distribution. We generated data that satisfies a desired STC by first applying a spatial transformation to a vector V of uniformly distributed random variables, and then by filtering the resulting vector through an autoregressive (AR) digital filter to introduce the desired temporal correlation. The coefficients of the autoregressive filter were obtained using the standard Levinson-Durbin algorithm which takes as input the targeted correlation for the different time lags, and outputs the filter coefficients for the specified order [9]. Since the driving noise (V) we used in the first place is non-Gaussian, the resulting process is also non-Gaussian, and only matches the second-order statistics requirements.

6.2 Simulation Model and Methodology

We assume that n nodes, each with a cache of size c, perform a random walk in a 2-D field of dimensions $L \times L$. At every time unit, each node decides whether or not to sample its current location. This decision is made based on the utility that this new sample provides compared to utility of the original cache content. If the new sample does not increase the utility of the cache, it is not kept in the cache. Otherwise, one of the old samples that provides the least utility is evicted in favor of the newly acquired one. After allowing a warmup period of w time units, each node is required to answer a query every time unit. The query specifies a location in the field, referred to as *query target*. A query answer is an estimate of the value of the phenomenon at the query target given each node's locally cached field samples. Notice that each node is asked an independent query whose target is drawn from the spatial query distribution Q. This distribution is assumed to be a bivariate normal distribution whose mean is the center of the field, and variance is $\sigma_Q^2 \times I$, where I is the identity matrix of size 2×2. The answer to any query is calculated using Eq. (5), where Y in Eq. (5) is the vector of samples cached by the queried node.

In the experiment with the Gaussian phenomenon, evaluation of the posterior densities by the mobile nodes only required evaluation of a mean vector and a covariance matrix which capture the entire distribution. However, in the non-Gaussian scenario, the computational complexity of DEBT becomes prohibitively expensive, especially for large cache sizes. The reason is that the evaluation of the posteriors requires marginalization of the space-time distribution over the range of the variables of interest for the entire duration of the evaluation (*i.e.*, length of the simulation in time units). Hence, in the experiment with the Random phenomenon, we only evaluate CBT.

In order to assess the robustness of CBT to model mismatch, we also conducted another experiment in which noise is added to the second-order statistics knowledge used by the nodes for managing their caches (to reflect uncertainty in correlation knowledge). We then evaluate the performance for different signal-to-noise ratios (SNR), where SNR is defined as:

$$SNR = 10\, log_{10} \frac{\sigma^2}{\sigma^2_{noise}} \tag{9}$$

where σ^2 is the variance of the phenomenon, and the added noise is Gaussian with mean $\mu = 0$, and variance σ^2_{noise}. We experimented with SNR's = 2db, and 15db.

To quantify the gains achieved by the proposed techniques, we compare them to random caching, which provides us with a lower bound on performance. With random caching, at every time unit, each node randomly decides whether or not to sample its current location. If a node decides to sample its current location, and its cache is full, it randomly chooses one of its local samples to be evicted to accommodate the newly acquired sample.

In the following evaluation, we set the default value of the parameters of our simulation and data models as follows. $L = 8$, $c = 10$, $n = 5$, simulation time = 100 time units, warmup time $w = 50$ time units, variance of the Gaussian phenomenon σ_G^2 = 50, variance of the random phenomenon σ_R^2 = 50, and variance of the spatial query distribution σ_Q^2 = 4. The default mobility model is a random walk on a 2D discrete field, under which, each node is initially placed at random location in the field. Then at every time unit, each node moves to one of its four neighboring locations with the same probability (*i.e.,* 0.25 for each location).

6.3 Performance Metrics

The main performance metric we used in our evaluation is the Mean Squared Error (MSE): Given a specific query, a node returns an estimate of the value of the phenomenon at the query location. We then measure the mean squared error associated with this estimate. Thus, given a query q at time t whose target is $\ell(q)$, the MSE in the estimation of q is:

$$MSE = E[(V_{\ell(q),t} - \hat{V}_{\ell(q),t/\mathbf{s_t}})^2] \tag{10}$$

We calculate the MSE for each query received by each node after the warmup period, then we report the average of 20 independent simulation runs.

We start by showing results of a single node as a function of the cache size c, and the variance of the query distribution σ_Q^2. Then we show results of cooperation between a number of nodes. More results can be found in the extended version of this paper [15].

6.4 Single-Node Results

Effect of Cache Size: Figure 1 (left) shows the effect of cache size on the MSE of the different considered strategies for a Gaussian and non-Gaussian phenomena. Intuitively, as the cache size increases, the better the MSE performance of CBT and DEBT since a larger cache size implies a better reconstruction of the phenomenon by the queried nodes. DEBT has a lower MSE compared to CBT, however, CBT's performance is very competitive at a much lower computational cost.

Similar effects could also be observed for the non-Gaussian phenomenon (Figure 1 right), regarding the efficiency of CBT. CBT outperforms random caching by a factor of two orders of magnitude. As expected, adding noise to the correlation structure of the phenomenon (*i.e.,* decreasing SNR), degrades the CBT performance. However, even with SNR of as low as 2db, CBT still outperforms random caching with a significant gain.

Query Spatial Distribution Variance: Figure 2 quantifies the effect of a larger variance, σ_Q^2, for the query distribution on the MSE for both Gaussian and non-Gaussian phenomena. Intuitively, a larger variance implies more uncertainty in the target query locations for a fixed cache size and a fixed number of nodes, which explains the decrease in estimation quality for the various schemes.

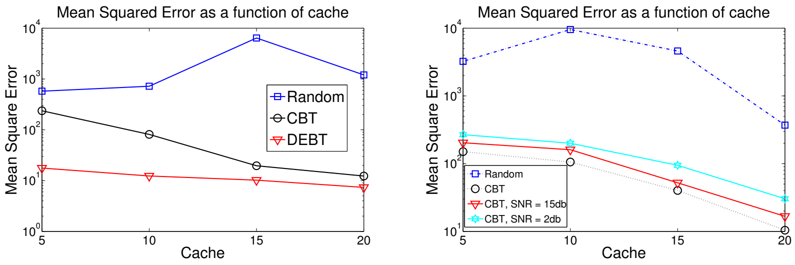

Fig. 1. Performance as a function of the cache size for a Gaussian phenomenon (left), and a non-Gaussian phenomenon (right).

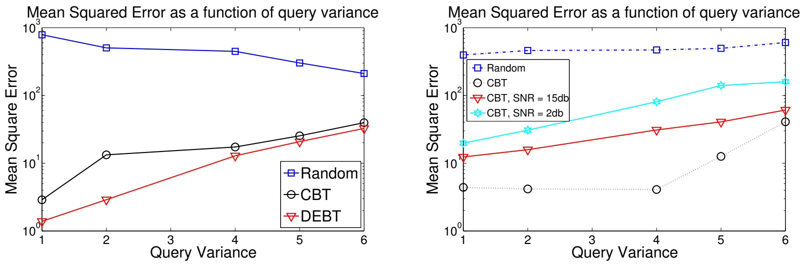

Fig. 2. Performance as a function of the variance of the query distribution for a Gaussian phenomenon (left), and a non-Gaussian phenomenon (right).

In case of a Gaussian phenomenon (Figure 2 left) both DEBT and CBT have MSE that is an order of magnitude lower than that of random caching. While in case of a non-Gaussian phenomenon (Figure 2 right), CBT achieves a huge improvement over random caching, with respect to the MSE. Adding noise to the correlation information decreases the performance of CBT, but is still much better than random caching.

6.5 Multi-Node Results

In the following experiments, we gauge the performance improvement due to cooperation between multiple nodes, as we explained it in Section 5 for a non-Gaussian phenomenon. Intuitively, we expect cooperation between nodes to improve the performance of all techniques, where the degree of improvement depends on the density of the nodes. We study this effect by varying the cache size and the number of nodes in the field. We also plot the cooperation gain, which is defined as the ratio between MSE from experiments with one node to MSE of the same node when there are n cooperating nodes in the network. In [15], we show results of varying the variance of the distribution of query targets. In the following experiments, $n = 5$, and communication range = 8.

Effect of Cache Size: Figure 3 shows the effect of cache size on the MSE of the different considered strategies for a non-Gaussian phenomenon. The improvement of MSE due to cooperation is evident. It is clear that, after increasing the cache size to a certain point, cooperation causes the gap between random and CBT to shrink. The reason is

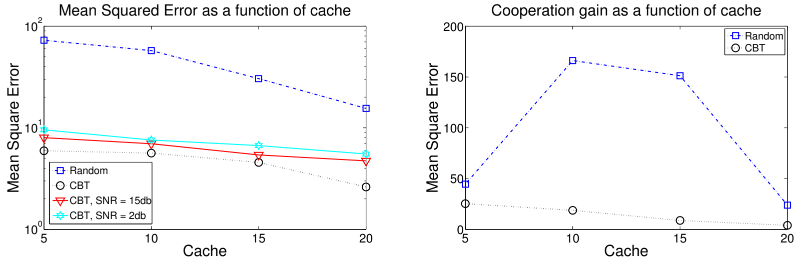

Fig. 3. Performance of multiple nodes as a function of the cache size for a random phenomenon (right), cooperation gain (ratio of MSE with a single node and with n nodes).

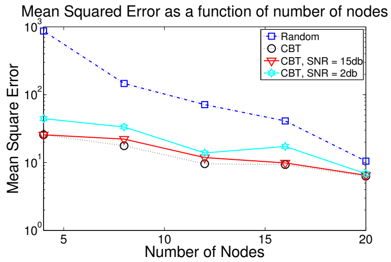

Fig. 4. Performance of multiple cooperative nodes as a function of the number of nodes n.

that, at this point, there is enough storage capacity in the system, such that the performance of a smart algorithm and that of a naive algorithm seem to be close. However, the improvement of performance comes at a cost of added communication overhead. This is an important factor in system design. It implies that, in dense systems where nodes are not power-limited, a smart caching algorithm is not the only option to consider. However, in sparse systems, or in systems where nodes are power-constrained, applying a smart caching algorithm makes a noticeable difference in performance.

Effect of Number of Nodes: Figure 4 shows the effect of varying the number of nodes, n, on the MSE of CBT and random caching for a non-Gaussian phenomenon. Increasing the number of nodes increases the amount of cooperation between nodes, and the storage capacity of the entire system. This improves the estimation by all nodes. Random caching has noticeable improvement as we increase the number of nodes. This trend matches the expectation that when storage is abundant, the caching algorithms make a minor difference. However, for all the parameter ranges we experimented with, CBT, even with noisy versions, performs better than random caching.

7 Related Work

The main goal of data placement in sensor networks is to minimize the access cost [16, 20], where cost is quantified in terms of communication energy.

In order to save energy in the context of caching, Kotidis [10] tries to optimize energy consumption by trying to put some sensor nodes to sleep mode, without affecting the query ability of the network. This is done by building a correlation model for the samples of sleeping nodes in neighboring active nodes. However, the built model is only local and can not be used to answer general queries about the entire network. It also involves packet exchange and fitting neighbors' data to a linear model. In this paper, given knowledge of the spatio-temporal correlation model, we use it to locally (with no packet exchange) answer queries about the entire network.

In all of the above efforts, the entire network is assumed to be static, while our work considers mobility, which is a harder problem.

Spatio-temporal queries have been studied in static networks with both static [4] and mobile [12] sinks. Our model is different in that, queries are handled only locally. Moreover, the temporal dimension to the problem is manifested in the correlation structure of the phenomenon.

Caching and replication have been considered in ad hoc networks [24, 11, 19]. Nodes are assumed to be interested in a fixed set of objects such that each object has a well-defined source. In our case, queries may target field locations that may not have been sampled by any node.

Leveraging mobile sensor networks to perform field monitoring has been studied [2, 3, 25]. While these efforts assume control over the mobility pattern and optimize it in order to maximize the utility of the system, our work maximizes the utility of the cache given the uncontrolled mobility model of the hosts.

Finally, we utilized information theory to assign a measure of merit to any set of samples. Information theory has been used in similar problems [14].

8 Conclusion

In this paper we focused on the problem of field monitoring using autonomously mobile sensor nodes. Nodes make local decisions about whether or not to sample their current location and how to manage their limited storage. We proposed a distributed entropic based technique (DEBT) to solve this problem. DEBT assumes knowledge of the entire distribution of the target phenomenon, and leverages this knowledge to make decisions about the cache management. DEBT has two major limitations: 1) high computational complexity, and 2) knowledge of the entire distribution of the target phenomenon is not always feasible. We then proposed CBT, a more practical approach, which assumes knowledge of only second-order statistics of the target phenomenon. CBT has a much lower computational complexity, and very competitive performance. We evaluated both techniques and showed that the resulting gains in MSE are substantial for both Gaussian and random phenomena. Furthermore, CBT still delivers very good estimation of the field, even when its knowledge about the correlation structure is not perfect.

We intend to extend the model we presented here to incorporate node cooperation on the caching (*e.g.,* decision making) plan, such that nodes can benefit from the knowledge attained by their neighbors in sample management. We also intend to study the effect of different mobility models on the performance of different cache management techniques.

References

1. Wireless sensor system guides urban firefighters http://mrtmag.com/mag/radio_wireless_sensor_system/.
2. Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing*, 5(6):640–652, 2006. Guiling Wang and Guohong Cao and Thomas F. La Porta.
3. M. A. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G. S. Sukhatme, W. J. Kaiser, M. Hansen, G. J. Pottie, M. Srivastava, and D. Estrin. Call and response: experiments in sampling the environment. In *SenSys*, 2004.
4. A. Coman, M. A. Nascimento, and J. Sander. A framework for spatio-temporal query processing over wireless sensor networks. In *DMSN*, 2004.
5. T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.
6. G. F. Hatke and A. F. Yegulalp. A novel technique for simulating space-time array data. volume 1, pages 542–546, Pacific Grove, CA, USA, 2000.
7. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS*, 2000.
8. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *MobiCom '00*, pages 56–67, 2000.
9. S. Kay. *Modern Spectral Estimation: Theory and Application*. Prentice Hall, 1988.
10. Y. Kotidis. Snapshot queries: Towards data-centric sensor networks. In *The 21st International Conference on Data Engineering (ICDE)*, Tokyo, Japan, April 2005.
11. W. H. O. Lau, M. Kumar, and S. Venkatesh. A cooperative cache architecture in support of caching multimedia objects in manets. In *WOWMOM*, pages 56–63, 2002.
12. C. Lu, G. Xing, O. Chipara, C. Fok, and S. Bhattacharya. A spatiotemporal query service for mobile users in sensor networks. In *ICDCS '05*, pages 381–390, 2005.
13. S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In *SIGMOD '03*, pages 491–502, 2003.
14. D. Marco, E. J. 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. 2003.
15. H. Morcos, A. Bestavros, and I. Matta. An information theoretic framework for field monitoring using autonomously mobile sensors. Technical Report BUCS-TR-2008-003, Computer Science Department, Boston University, 111 Cummington Street, Boston, MA 02135, January 2008.
16. K. S. Prabh and T. F. Abdelzaher. Energy-conserving data cache placement in sensor networks. *ACM Trans. Sen. Netw.*, 1(2):178–203, 2005.
17. S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. Ght: a geographic hash table for data-centric storage. In *WSNA*, pages 78–87, 2002.
18. R. Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley & Sons, 1981.
19. F. Sailhan and V. Issarny. Cooperative caching in ad hoc networks. pages 13–28, 2003.
20. B. Sheng, Q. Li, and W. Mao. Data storage placement in sensor networks. In *MobiHoc '06*, pages 344–355, 2006.
21. S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin. Data-centric storage in sensornets. *SIGCOMM Comput. Commun. Rev.*, 33(1):137–142, 2003.
22. H. L. V. Trees. *Detection, Estimation, and Modulation Theory, Part I*. John Wiley and Sons, 2001.
23. F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *MobiCom '02*, pages 148–159, 2002.
24. L. Yin and G. Cao. Supporting cooperative caching in ad hoc networks. In *Infocom*, Hong Kong, March 2004. IEEE Infocom.
25. Y. Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. San Francisco, CA, USA, 2003.