Cleaning and Querying Noisy Sensors

Eiman Elnahrawy
Department of Computer Science
Rutgers University
Piscataway, NJ 08854, USA
eiman@paul.rutgers.edu

Badri Nath
Department of Computer Science
Rutgers University
Piscataway, NJ 08854, USA
badri@cs.rutgers.edu

ABSTRACT

Sensor networks have become an important source of data with numerous applications in monitoring various real-life phenomena as well as industrial applications and traffic control. Unfortunately, sensor data is subject to several sources of errors such as noise from external sources, hardware noise, inaccuracies and imprecision, and various environmental effects. Such errors may seriously impact the answer to any query posed to the sensors. In particular, they may yield imprecise or even incorrect and misleading answers which can be very significant if they result in immediate critical decisions or activation of actuators. In this paper, we present a framework for cleaning and querying noisy sensors. Specifically, we present a Bayesian approach for reducing the uncertainty associated with the data, that arise due to random noise, in an online fashion. Our approach combines prior knowledge of the true sensor reading, the noise characteristics of this sensor, and the observed noisy reading in order to obtain a more accurate estimate of the reading. This cleaning step can be performed either at the sensor level or at the base-station. Based on our proposed uncertainty models and using a statistical approach, we introduce several algorithms for answering traditional database queries over uncertain sensor readings. Finally, we present a preliminary evaluation of our proposed approach using synthetic data and highlight some exciting research directions in this area.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]

General Terms

Algorithms, Design, Experimentation

Keywords

Wireless Sensor Networks, Noisy Sensors, Uncertainty, Bayesian Theory, Query Evaluation, Statistics

1. INTRODUCTION

The emerging field of wireless sensor networks enables large-scale sensing of the physical world. A typical sensor network consists of a large number of sensors, embedded in physical spaces, continuously collecting and communicating their readings to the base-station or the sensor database in order to answer various user-defined queries. Existing networks are used for monitoring of several physical phenomena such as contamination, climate, building structure, and so on, potentially in remote harsh environments [30, 19]. They also found several interesting applications in industrial engineering such as monitoring the quality of food, especially perishable items, as well as real life applications such as transportation and traffic control [3, 28]. In all these cases, the primary source of sensor data is actual measurements of physical or well-modelled phenomena, and thus, sensor data is subject to several different sources of errors. In general, these sources of errors can be classified broadly as either systematic errors (bias) or random errors (noise). Systematic errors arise due to changes in the operating conditions, e.g., temperature, humidity, etc., or other factors such as ageing of the sensor. They can be corrected by calibration as has been recently addressed in [6]. Calibration is not the focus of this paper. We are particularly interested in reducing the effect of random errors on sensor readings since they may seriously affect queries over sensor data. The sources of random errors include, but are not limited to, (1) noise from external sources, (2) random hardware noise, (3) inaccuracies in the measurement technique (i.e., readings are not close enough to the actual value of the measured phenomenon), (4) various environmental effects and noise, and (5) imprecision in computing a derived value from the underlying measurements (i.e., sensors are not consistent in measuring the same phenomenon under the same conditions).

Several examples from the current technology reveal that sensors vary significantly in their precision and accuracy, tolerance to hardware and external noise, etc., based on their type, cost and application. For example, experiments showed that the distribution of noise varies widely in different photovoltic sensors [6]. GPS inaccuracy in determining the position can be up to few meters (dfia.com). Precision and accuracy of humidity sensors may also vary significantly (www.veriteq.com). The environment in which the sensors operate is also usually unpredictable or harsh. Numerous other external and uncontrollable factors may in turn affect the quality (accuracy) of the reported sensor reading, and in many cases result in inaccurate measurements. An example is recording the distances to a fixed point by using signal strength (SS). The recorded distance varies widely as the SS values at the sensor are subject to external conditions. Also, weights of trucks can be measured by means of strain gauges, attached to bridges, which can be affected by other vibrations. The aim of the industry, however, is to manufacture tiny cheap sensors that can be scattered everywhere and disposed when they run out of their batteries [1]. Therefore, intolerance to internal and external noise, imprecision, and inaccura-

WSNA'03, September 19, 2003, San Diego, California, USA.

cies are inevitable and highly expected among those cheap sensors. They will basically vary with the cost of the sensors.

Such random errors may seriously impact the answer of any query posed to the sensors. In particular, they may yield imprecise or even incorrect and misleading answers. The cost of the errors can be very significant, especially when they result in immediate critical decisions or activation of actuators. We argue that errors in sensor data cannot be ignored. For example, consider the scenario of Figure 1(a), simplified for the sake of illustration. Bacteria growth in perishable items is monitored over the time by attaching cheap wireless temperature and humidity sensors over them which can be quite noisy. If the temperature and the humidity conditions of any item fall under or go over given thresholds, the item should be thrown away. Assume that the range of acceptable humidity and temperature are $[h_1, h_2]$, and $[r_1, r_2]$, respectively. t_i refers to the true temperature and humidity readings at item i, while o_i refers to the reported (observed) readings at item i. As shown in the figure and based on the reported noisy data, items 1, 4 should be thrown away while items 2,3 should remain. However, based on the true readings item 1 should remain while item 3 should be thrown away!

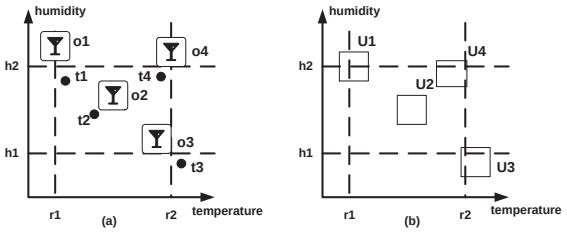

Figure 1: (a) Based on the observed readings items 1,4 will be thrown away, (b) Based on the uncertainty regions, only item 3 will be thrown away.

In traditional databases, the source of data is either an explicit data-entry operation or a transaction activity. The origin of data is typically business, financial or personnel. The data model assumes clean data. Noisy data, if any, is assumed to be cleaned off-line by a separate database functionality. Sensor data, on the other hand, has different characteristics; it is updated continuously, i.e., it forms a data stream. In addition, it is usually used for decision making or triggering of actuators in real-time. Therefore, cleaning of noisy sensor readings cannot be a separate off-line operation as in traditional databases. Recent work on query processing in sensor databases has focused on data gathering using network primitives, in-network aggregation, and query languages [18, 19, 12, 26]. The emphasis of these approaches is to take into consideration the resource constraints of sensors such as bandwidth and energy. We argue that errors is also a serious limitation of sensors as important as energy and bandwidth constraints. They result uncertainty in determining the true reading (measurement) of the sensor: since the sensor is prone to errors it is uncertain about its true reading. We therefore introduce a general framework for cleaning and querying of noisy sensors. Our cleaning functionality aims at reducing the uncertainty associated with the reading of each sensor that arises due to random noise, thus obtaining a more accurate estimate of the true "unknown" reading. Specifically, we present a Bayesian approach for reducing the uncertainty in an online fashion. This cleaning functionality can be performed either at the sensor level or at the database level (base-station). We assume that the reading of each individual sensor is important, and therefore, our cleaning functionality works on every single sensor. Even if the readings of a set of sensors are combined (aggregated) into a single more robust reading to reduce the effect of noise [29], our approach can still work on this single reading, thus yielding more accurate results. Based on our proposed uncertainty models and using a statistical approach, we introduce several algorithms for answering a wide range of traditional database queries over uncertain sensor readings. We shall show that the above scenario of perishable items can be avoided using our proposed framework for cleaning and querying. Finally, we present a preliminary evaluation of our proposed approach using synthetic data and highlight some exciting research directions in this area.

The rest of this paper is organized as follows. We describe our data domain and present our proposed framework in Section 2. In Section 3, we discuss a Bayesian approach for reducing the uncertainty associated with noisy sensors. We introduce algorithms for evaluation of queries over uncertainty models in Section 4. We discuss an experimental evaluation of our framework in Section 5. Section 6 discusses related work. Finally, Section 7 concludes this paper and highlights our major future work directions as well as challenging research problems in this area.

2. OVERALL FRAMEWORK

In this section, we give an overview of our domain. We also describe our proposed framework for dealing with noisy sensors.

2.1 Domain Description

We assume that there is a set of n sensors, $S = \{s_i\}, i = 1 \cdots n$, scattered in the space and forming a wireless sensor network. The sensors are capable of providing their measurements at each time instance t and reporting them to a specific collecting point (base-station). Low-level networking techniques for routing, topology maintenance, communication, etc., are implicitly assumed to be available. We think of each sensor s_i at a specific time instance t as a tuple in the sensor database with attributes corresponding to the readings of the sensor. Each sensor has one or more reading corresponding to each measurement. The attributes of a specific sensor, s_i, at a specific time instance t are denoted by $s_i \cdot A(t) = \{s_i.a_j(t)\}, j = 1 \cdots m$, where m is the total number of attributes. We assume that the same sensor may be used for sensing different phenomena or that many specialized sensors, installed at the same location, are combined to form one "virtual" multi-attribute sensor. Furthermore, we assume that all the sensors have the same number of attributes. If this is not the case then each phenomenon is treated separately and the sensors will have a single attribute for each phenomenon. We assume that all the attributes are real-valued. However, the proposed framework can be extended to the case of discrete-valued attributes in a fairly similar way. We are concerned with the readings of each sensor (attributes' values) at a specific time instance, hence, we may drop the time index t when referring to the sensors and their readings. Since our focus is uncertainty due to random errors, we assume that all the tuples exist (no missing tuples) and are complete (no incomplete tuples), but that the attributes are noisy.

2.2 General Model

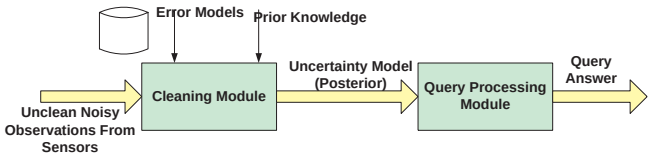

Figure 2: Overall Framework.

Our overall framework is shown in Figure 2. It consists of two

major modules; a cleaning module and a query processing module. The cleaning module is responsible for cleaning the noisy sensor data, in an online fashion, by computing accurate uncertainty models of the true "unknown" data. In particular, there are three inputs to this cleaning module: (1) the noisy observations reported from the sensors, (2) metadata about the noise characteristics of every sensor (error model), and (3) information about the distribution of the true reading at each sensor (prior knowledge). We shall discuss the latter two inputs shortly. The output of the cleaning module is probabilistic uncertainty models of the reading of each sensor (posterior), i.e., a probability density function (pdf) of the true "unknown" sensor reading taking on different values. Computing these models is the topic of the next section. The query processing module is responsible for producing answers to any posed query to the system using the uncertainty models of the current readings. Since the uncertainty models are probabilistic, traditional query evaluation algorithms, that assume a single value for each reading, cannot be used. Hence, our query processing module uses algorithms that are based on statistical approaches for computing functions over random variables. In Section 4, we shall introduce these algorithms in details.

The error model of each sensor is basically the distribution of the noise that affects it. It is assumed to follow a Gaussian distribution with zero mean. In order to fully define the model we need to compute its variance. The variance is computed based on the specification of each sensor (accuracy, precision, etc.), and on testing calibrated sensors under normal deployment conditions. This testing can be performed either by the manufacturers or by the users after installation and before usage. Environmental factors or characteristics of the field should also be taken into consideration. The error models may change over the time and new modified models may replace the old ones. The models should be stored as a metadata at the cleaning module. Sensors are not homogeneous with respect to their noise characteristics, and therefore, each sensor type, or even each individual sensor should have its own error model.

Prior knowledge, on the other hand, represents a distribution of the true sensor reading taking on different values. There are several sources to obtain prior knowledge. It can be computed using facts about the sensed phenomenon, learning over time (history), using less noisy (more precise) readings as priors for the more noisy ones, or even by expert knowledge or subjective conjectures. Nevertheless, they can be computed dynamically at each time instance if the sensed phenomena is known to follow a specific parametric model. For example, if the temperature of perishable items is known to drop by a factor of $x\%$ from time $t-1$ to time t then the (cleaned) reading of the sensor at time $t-1$ is used to obtain the prior distribution at time t. The resultant prior along with the error model and the observed noisy reading at time t are then input to the cleaning module in order to obtain the uncertainty model of the sensor at time t. Our approach in this case of dynamic priors indeed resembles Kalman filters [17].

It is worth mentioning that a straightforward approach for modeling uncertainty in sensor readings due to noise is to assume that the true unknown reading of each sensor follows a Gaussian pdf, centered around the observed noisy reading, with variance equals to the variance of the noise at this sensor. However, it is a fundamental fact among estimation theory community that the use of prior knowledge leads to more accurate estimators [15]. This motivated our use of prior knowledge in cleaning, in order to reduce the uncertainty associated with noisy sensors. We shall justify this fact in Section 3 by proving that the estimation error in our proposed cleaning approach is less than that of the straightforward case. Some priors, however, are more useful than the other; in the sense that they have less variance. This in turn results in more reduction in the uncertainty and enhance the overall accuracy of our framework. In general, if the prior knowledge is not strong enough (i.e., has a very wide distribution compared to the noise distribution), then our approach will still be superior, though not "very" advantageous in terms of estimation error. Fortunately, in many situations this is not the case as we discussed before. For example, situations where we have cheap and very noisy sensors scattered everywhere to collect measurements of a well-modelled phenomenon such as temperature, etc. A strong prior can be easily computed in this situation while the noise is expected to have a very wide variance.

The final point that we address in this section is some deployment issues. Specifically, there are two places where cleaning and query processing can be performed, at the sensor level or at the database level (or base-station). Each option has its communication and processing costs, which can be interpreted to energy consumption, and storage cost. Due to the limited space, we will not introduce explicit cost models for each case since there are many factors involved. Sensor capabilities, application, etc., will force the decision of which approach to use. Experimentation to guide this decision is part of our future work. In what follows, we aim at illustrating rough estimates of the advantages and limitations of each option.

Sensor Level When the cleaning is performed at the sensor there is a storage cost to store the prior and the error models at the sensor. The storage cost depends on the complexity of the two models; the more complex (more parameters) the more storage space. Furthermore, there may be a significant communication cost to send the prior to the sensor from the base-station. Specifically, if the priors are dynamic and depend on factors other than the readings of the sensor, e.g., readings of other sensors, then this option is not advantageous. On the other hand, if the prior is (almost) static, or dynamic but can be computed at the sensor, then the communication cost will be negligible. Cleaning also introduces a processing cost to compute the posteriors. This cost can be significant as we shall present in Section 3. However, the major advantage of performing the cleaning at the sensor level is that a point estimation of the resultant posteriors can be obtained. Consequently, traditional approaches to in-network query processing and aggregation can be used with error bounds [19, 30]. Performing the cleaning at the sensor and the query processing at the database level has no advantages. This is clearly due to the fact that communicating the noisy reading (a single value) to the base-station and performing the cleaning there always has less communication cost than communicating the parameter(s) of the resultant uncertainty model. In addition, it introduces a storage cost (prior, error models).

Database Level We assume that any processing or storage at the database level has no cost which is the major advantage of performing the cleaning and the query processing there. Furthermore, communication cost of sending dynamic priors to the sensors is saved. Answers to posed queries will also be computed exactly using techniques of Section 4. The major limitation, however, is that distributed query processing cannot be used.

3. REDUCING THE UNCERTAINTY

In this section, we present our approach for reducing the uncertainty associated with noisy sensor reading, i.e., for computing more accurate uncertainty models of each sensor. Our proposed approach is an online cleaning; we combine the prior knowledge of the true sensor reading, the error model of the sensor, and its observed noisy reading together, in "one step" and online. This step is performed using Bayes' theorem shown in Equation 1, where the

likelihood is the probability that the data x would have arisen for a given value of the parameter θ and is denoted by $p(x|\theta)$ [4, 11, 9]. This leads to the posterior pdf of θ, $p(\theta|x)$. The rest of this section includes a background of Bayes' theory. Readers who are familiar with it may skip to the next section.

$$p(\theta|x) = \frac{likelihood \times prior}{evidence} = \frac{p(x|\theta)p(\theta)}{\int_\Psi p(x|\Psi)p(\Psi)d\Psi} \quad (1)$$

3.1 Single-Attribute Sensors

Sensors of this class have only one attribute. Due to occurrence of random errors the observed value of the attribute o will be noisy, i.e., it will be higher or lower than the true value t. As discussed in Section 2, the random error is normally distributed (Gaussian) with zero mean and a known standard deviation $\sim N(0, \delta^2)$. Therefore, the true value t follows a Gaussian distribution centered around a mean $\mu = t$ and with variance δ^2, i.e., $p(o|t) \sim N(t, \delta^2)$. We apply Bayes' theorem to obtain a more accurate uncertainty model (posterior pdf) for t, $p(t|o)$. In particular, we combine the observed value o, error model $\sim N(0, \delta^2)$, and the prior knowledge of the true reading distribution $p(t)$ as follows.

$$p(t|o) = \frac{p(o|t)p(t)}{p(o)} \quad (2)$$

Equations 3,4 shows the computation when the reading of a specific sensor s is known to follow a Gaussian distribution with mean μ_s and standard deviation σ_s, i.e., $t \sim N(\mu_s, {\sigma_s}^2)$ (prior). Specifically, by applying Bayes' theorem and using some properties of the Gaussian distribution we conclude that the posterior probability $p(t|o)$ also follows a Gaussian distribution $N(\mu_t, {\sigma_t}^2)$ [4, 11]. In general, we do not restrict the prior distribution of the true reading, t, to a specific class of distributions. However, Gaussian distributions have certain attractive properties which makes them a good choice for modeling priors. In particular, they yield another Gaussian posterior distribution with easily computed parameters which enables performing the cleaning at the sensor level, they are known to be analytically tractable, they are useful for query processing and yield closed form solutions as we will show in Section 4, and finally they also have the maximum entropy among all distributions [9].

$$\mu_t = \frac{\delta^2}{{\sigma_s}^2 + \delta^2}\mu_s + \frac{{\sigma_s}^2}{{\sigma_s}^2 + \delta^2}o \quad (3)$$

$${\sigma_t}^2 = \frac{{\sigma_s}^2.\delta^2}{({\sigma_s}^2 + \delta^2)} \quad (4)$$

Example Let us obtain the uncertainty model of a temperature sensor at a specific time instance. Assume that our prior knowledge is that the temperature r follows a Gaussian distribution, and it is most likely 9 degrees with standard deviation of 4, i.e., $r \sim N(\mu_s = 9, {\sigma_s}^2 = 4^2)$. Further assume that the noise at this sensor is known to have a standard deviation of 10; noise $\sim N(0, \delta^2 = 10^2)$. If the reported noisy temperature is 15 then, using equations 3 and 4, we obtain a mean ≈ 9.8 and a standard deviation ≈ 3.7 of the posterior distribution for the true unknown temperature, $p(t|o) \sim N(9.8, 3.7^2)$ as shown in Figure 3.

In order to prove the effectiveness of our approach in reducing uncertainty, we compute the Bayesian mean squared error, $E[(t - \hat{t})^2]$ for the resultant posterior, where $t, \hat{t}$ are the true unknown reading, and the posterior mean, respectively. We then compare it with the case when no prior knowledge is utilized which is the straightforward approach, discussed in Section 2. The error (uncertainty)

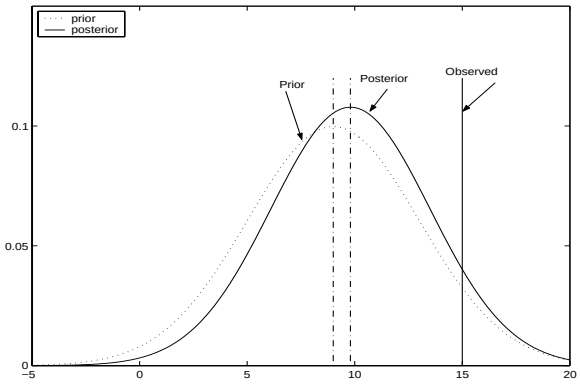

Figure 3: The resultant uncertainty model of the true temperature (posterior) and the observed erroneous reading.

in the resultant posterior equals ${\sigma_t}^2 = \delta^2.(\frac{{\sigma_s}^2}{{\sigma_s}^2+\delta^2})$ (please refer to [15] for details). This amount is less than δ^2, the error (uncertainty) when no prior is utilized. Moreover, when the variance of the prior becomes very small as compared to the variance of the noise, or in other words, when the prior becomes very strong, the error of the posterior becomes smaller and the uncertainty is further reduced. Consequently, our resultant uncertainty model becomes far more accurate than the no-prior case. Equation 3 also illustrates an interesting fact. It shows that our approach in general compromises between the prior knowledge and the observed noisy data. When the sensor becomes less noisy, its observed reading becomes more important and the model depends more on it. At very high noise levels, the observed reading could be totally ignored.

3.2 Multi-Attributes Sensors

We now extend our approach to the case of multi-attributes sensors. We assume that the random errors on the attributes are independent and normally distributed, but not necessarily identical, i.e., the random error e_i on attribute $s.a_i$ is normal with zero mean and a known standard deviation $\sim N(0, {\delta_i}^2), i = 1 \cdots m$, where m is the number of attributes. The observed noisy readings of the attributes are represented by the column vector $O_{m\times 1}$ while the unknown true readings are represented by the column vector $T_{m\times 1}$. Based on our error model, we assess that T follows a multivariate Gaussian distribution centered around an $m-$component mean column vector $\mu = T$ and with an $m \times m$ covariance matrix Σ, i.e., $p(O|T) \sim N(T, \Sigma)$. The prior knowledge in this case is a multivariate distribution $p(T)$ that models the prior knowledge of the true readings and the correlation between the attributes appropriately.

$$\Sigma = \begin{pmatrix} {\delta_1}^2 & 0 & \cdots & 0 \\ 0 & {\delta_2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & {\delta_m}^2 \end{pmatrix} \quad (5)$$

Similar to the single attribute case, the posterior pdf of T, $p(T|O)$ is computed using Bayes' theorem. For example, assume that the readings of a specific sensor s follow a multivariate Gaussian distribution with mean vector μ_s and a covariance matrix Σ_s, i.e., the prior $T \sim N(\mu_s, \Sigma_s)$, $p(O|T)$ follows a Gaussian distribution centered around T with a covariance matrix Σ described by equation 5, i.e., $p(O|T) \sim N(T, \Sigma)$. Using Bayes' theorem and the properties of Gaussian distributions, the multivariate posterior pdf, $p(T|O)$, is also Gaussian $\sim N(\mu_T, \Sigma_T)$, where μ_T and Σ_T are computed as follows.

$$\mu_T = \mu_s + \Sigma_s[\Sigma_s + \Sigma]^{-1}(O - \mu_s) \quad (6)$$

$$\Sigma_T = \Sigma_s - \Sigma_s[\Sigma_s + \Sigma]^{-1}{\Sigma_s}^t \quad (7)$$

The terms $\Sigma_s[\Sigma_s + \Sigma]^{-1}$, Σ_T will be computed off-line. They need not be recomputed at every time instance as long as the prior does not change. Moreover, if the attributes are known to be uncorrelated, the covariance matrices Σ_s, Σ_T will be diagonal and the computations will be further simplified. In general, if the attributes are known to be uncorrelated, the multivariate case will reduce to m individual single attribute cases where the uncertainty associated with each attribute can be obtained independently of the other attributes. In fact the multivariate posterior pdf in this case is the product of the individual posteriors. Correlation between attributes, when exists, however, usually leads to more accurate models.

4. EVALUATION OF QUERIES

In this section, we highlight the major differences between evaluation of queries over data of uncertainty models and data of single points. We also give a classification of queries that we consider in this paper. We then present algorithms for evaluation of queries over sensor data, presented using our proposed uncertainty models discussed in Section 3. These algorithms are used in the processing module of our framework over the output of the cleaning module and at the database level. In what follows, for simplicity of notation, we use the term $p_{s_i}(t)$ to describe the uncertainty model in the univariate case, i.e., the posterior distribution, $p(t|o)$, of sensor s_i. For the multi-attribute case, we will use the term $p_{s_i}(T) = p_{s_i}(t_1, t_2, ..., t_m)$ to refer to $p(T|O)$ of sensor s_i.

Based on our uncertainty models, the reading of each noisy sensor at a specific time instance is considered a random variable (r.v.) described by the posterior pdf of the sensor and not necessarily by a single point with probability 1. Therefore, traditional query evaluation algorithms that assume single points cannot be used for evaluation over noisy sensors. Another significant difference between queries over exact data (single points) and over noisy sensor data (uncertainty models) is illustrated by the following example.

Example Consider the scenario where we have noisy temperature sensors. A user poses the following query to the system at a specific time instance, "return the maximum reading of those sensors that record a temperature ≥ 50F". If the data is exact (no noise) then the system will have a single reading of each sensor, i.e., the true reading of each sensor will be known exactly and equals to its observed reading with probability 1. Consequently, the system can check whether each sensor satisfies the provided predicate or not. It then returns the maximum of those sensors that satisfy the predicate. Now consider noisy sensors in our framework, the processing module does not have a single estimate of the true reading of each sensor. It only has a pdf that represents the "possible" values of the true reading. In order to determine whether or not a specific sensor satisfies the given predicate, the processing module can compute the probability that each sensor satisfies the predicate using its posterior pdf. However, when the probability is less than 1, which is highly expected, the module will be "uncertain" whether the sensor satisfies the predicate or not. Even though there is a high chance that a specific sensor satisfies the query as its probability approach 1, e.g., 0.85, neither the processing module nor any person can decide for sure. Therefore, there is no answer to this predicate and consequently we do not know which sensor is the maximum. In order to overcome this difficulty without violating any statistical rules, we propose modifying predicate queries by rephrasing them as "return the maximum value of those sensors that have at least a $c\%$ chance of recording at a temperature ≥ 50F". We call c the "confidence level", and it is defined by users as part of their queries. Intuitively, in presence of uncertainty users must play a role in determining which answers are considered acceptable and which answers should be rejected due to the lack of confidence. Following our reasoning, the processing module can now filter out all those sensors that have a probability less than $\frac{c}{100}$ of satisfying the query. It then computes the maximum over the remaining sensors. This leave the problem of computing the maximum over a pdf which we will discuss shortly.

Confidence Level (c)**:** The confidence level or the acceptance threshold c is a user-defined parameter that reflects the desired user's confidence. In particular, any sensor with probability $p < \frac{c}{100}$ of satisfying the given predicate should be excluded from the answer to the posed query.

4.1 Classification Of Queries

Classes of queries related to sensor networks have been identified broadly as traditional SQL-like queries and aggregates [19, 8, 2] or probabilistic range queries for moving objects [28]. In this paper, we follow a classification of queries similar to the former case. However, we do not claim that queries covered in this section form a complete set of possible queries on sensors, nor do we claim that this is the best classification of queries for all applications. Our objective is rather to cover a wide range of queries posed to domain of sensors, and we present a classification that simplifies our discussions. The algorithms that we introduce here are based on statistical approaches for computing functions over one or more random variables; e.g., summing of random variables, computing order statistics of a set of random variables, and so on. We assume that queries, posed to our system, will take on one of the following three forms.

- "What is the reading(s) of sensor x?". We call queries of this form Single Source Queries (SSQ).
- "Which sensors have at least $c\%$ chance of satisfying a given predicate?". We call queries of this form Set Non-Aggregate Queries (SNAQ), since no aggregation is involved.
- "On those sensors which have at least $c\%$ chance of satisfying a given predicate, what is the value of a given aggregate?". The aggregate can be a summary aggregate such as SUM, AVG, and COUNT aggregates or an exemplary aggregate such as MIN, MAX aggregates. We call the former Summary Aggregate Queries (SAQ), and the latter Exemplary Aggregate Queries (EAQ). This classification of aggregate queries into summary and exemplary has been extensively used among the database community, e.g., in [19]. The predicate can be empty. In this case, all the sensors in the field will be considered for aggregation.

4.2 Evaluation of SSQ

This class of queries returns the value(s) of the attribute(s) of a specific sensor and no aggregation is involved. We propose two approaches for answering this class of queries. The first one is based on computing the expected value of the probability distribution. For the single attribute case, the output of this approach is $E_{s_i}(t) = \int_{-\infty}^{\infty} t p_{s_i}(t) dt$ of the queried sensor s_i. The second approach is based on computing the $p\%$ confidence interval of $p_{s_i}(t)$. The confidence factor p can be user-defined with a default value equals to 95. The confidence interval is computed using Chebychev's inequality [7], as follows.

$$P(|t - \mu_{s_i}| < \epsilon) \geq 1 - \frac{{\sigma_{s_i}}^2}{\epsilon^2} \quad (8)$$

Where μ_{s_i}, σ_{s_i} are the mean and the standard deviation of $p_{s_i}(t)$, $\epsilon > 0$. In order to compute ϵ we set $(1 - \frac{{\sigma_{s_i}}^2}{\epsilon^2})$ to p and solve. The resultant $p\%$ confidence interval on the attribute will be $[\mu_{s_i} - \epsilon, \mu_{s_i} + \epsilon]$. The two approaches are extended to the multi-attribute sensors in a straightforward way. We first compute the marginal pdf of each attribute t_j of the queried sensor, s_i, $p_{s_i}(t_j)$. We then output the expected reading of each attribute, or the $p\%$ confidence interval on each attribute t_j by applying Chebychev inequality, on its marginal pdf.

4.3 Evaluation Of SNAQ

This class of queries returns the set of sensors that satisfy a given user-defined predicate. We assume that the predicates are simple range queries on one or more attributes. However, our general algorithm outlined below extends naturally to complex conditions with mixes of AND and OR as well as conditions that involve more than one attribute, etc., using a traditional statistical approach for computing the probability on complex conditions.

For the single attribute case, assume that the given range $R = [l, u]$ is specified by lower and upper bounds on the attribute value, l, u, respectively. The answer to the SNAQ will be the eligible set $S_R = \{s_i\}$ of those sensors with probability $(p_i > \frac{c}{100})$ of being inside the specified range R, where $p_i = \int_l^u p_{s_i}(t)dt$ along with their "confidence", p_i.

The fundamental difference in the multi-attribute case is that the range will be specified by several intervals on some of the attributes (a region) rather than a single interval $[l, u]$. Assume, without loss of generality, that the attributes involved in the original given range, R, are $\{t_1, t_2, \cdots, t_k\}$. We start by computing the marginal density distribution of those attributes, $p_{s_i}(t_1, t_2, \cdots, t_k)$. We then compute the eligible set S_R of those sensors s_i with probability $(p_i > \frac{c}{100})$ of being inside the specified range. Figure 4 summarizes the steps of obtaining the eligible set S_R for the multi-attribute case. If the reading(s) of the sensor is also required we compute it over the eligible set and using the algorithms of SSQ.

---

input c, S, R, **output** S_R
$S_R \leftarrow \{\}$
if predicate is empty **then** $S_R \leftarrow S$
else for $i = 1$ **to** $|S|$ **do**
  $p_{s_i}(t_1, \cdots, t_k) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} p_{s_i}(t_1, \cdots, t_m) dt_{k+1} \cdots dt_m$
  $p_i = \int \cdots \int_R p_{s_i}(t_1, t_2, \cdots, t_k) dt_1 \cdots dt_k$
  if $(p_i > \frac{c}{100})$ **then** $S_R = S_R \cup s_i$
return S_R

Figure 4: Computing S_R for multi-attributes sensors.

As an example, consider the scenario of Figure 1(b). Assume that the output of the cleaning module is that the reading of each sensor is uniformly distributed over the depicted squared uncertainty regions. The probabilities of the items being inside the given range are $(item1, 0.6), (item2, 1), (item3, 0.05), (item4, 0.85)$. If the user-defined confidence level is $c = 50\%$, which is a reasonable confidence level, then only item 3 will be thrown away. This coincides with the correct answer over the true unknown readings, and is also more accurate than the answer on the noisy (uncleaned) readings.

Our approach for obtaining the eligible set bears similarities with [8] for dealing with uncertainty in data due to lag of instantaneous updates. The major differences in our approach lie in excluding all sensors with probability $< \frac{c}{100}$ from the set, generalization to the multi-attributes case, and introducing algorithms to output the reading of each sensor, if required.

4.4 Evaluation Of SAQ

The aggregate functions that fall under this category are SUM, COUNT, and AVG queries. Before evaluating the aggregate, we obtain the eligible set S_R of those sensors that satisfy the given predicate, using algorithms of SNAQ. If the predicate is empty then all the sensors are considered in the aggregation, $S_R = S$.

To compute the SUM aggregate, we utilize a statistical approach for computing the sum of independent continuous random variables (convolution) since our uncertainty models form a set of independent continuous r.v. To sum $|S_R|$ sensors, we perform the convolution on two sensors and then add one sensor to the resultant sum (also a r.v.) repeatedly till the overall sum is obtained. Assume that the sum $Z = s_i + s_j$ of two uncertainty models of sensors s_i, s_j, is required. If the pdfs of these two sensors are $p_{s_i}(t), p_{s_j}(t)$, respectively, then the pdf of Z is computed using Equation 9 [7]. The expected value of the overall sum or a 95% confidence interval can then be computed and output as the answer similar to SSQ.

$$p_Z(z) = \int_{-\infty}^{\infty} p_{s_i}(x) p_{s_j}(z - x) dx \quad (9)$$

Computing the COUNT query reduces to output $|S_R|$ over the given predicate. Finally, the answer of the AVG query equals the answer of the SUM query divided by the answer of the COUNT query, over the given predicate. The algorithms of the multi-attribute case are analogous after computing S_R and marginalizing over the attribute involved in the aggregation.

4.5 Evaluation Of EAQ

This class includes the MIN and the MAX queries. Similar to the summary aggregate queries, we start by evaluating the eligible set S_R and then perform the aggregation over sensors in S_R. The MIN of m sensors in S_R is then computed as follows (MAX query is analogous).

Let the sensors $s_1, s_2, \ldots, s_m$ be described by their pdfs $p_{s_1}(t), \ldots, p_{s_m}(t)$, respectively, and their cumulative distribution functions (cdfs) $P_{s_1}(t), \ldots, P_{s_m}(t)$, respectively. Let the random variable $Z = min(s_1, s_2, \ldots, s_m)$ be the required minimum of these independent continuous r.vs. The cdf, pdf of Z, $P_Z(z), p_Z(z)$ are computed using Equations 10, 11, respectively [7]. This algorithm also generalizes to the multivariate case. Nevertheless, other order statistics such as Top-K, Min-K, and median are computed in a similar manner.

$$\begin{aligned} P_Z(z) &= prob(Z \leq z) = 1 - prob(Z > z) \\ &= 1 - prob(s_1 > z, s_2 > z, \ldots, s_m > z) \\ &= 1 - (1 - P_{s_1}(z)) \cdots (1 - P_{s_m}(z)) \end{aligned} \quad (10)$$

$$p_Z(z) = -\frac{d}{dx}(1 - P_{s_1}(z))(1 - P_{s_2}(z)) \cdots (1 - P_{s_m}(z)) \quad (11)$$

4.6 Discussion

The above algorithms involve several integrals that are not usually guaranteed to yield a closed form solution for all families of distributions. However, there are specific formulas for computing these integrals easily in the case of Gaussian distributions. For

example, the marginal pdf of Gaussian is also a Gaussian, so is the sum of Gaussians (and consequently the AVG) [7]. Evaluation of SSQ simply reduces to the mean parameter μ of the Gaussian uncertainty model in the single attribute case, and to the m-component mean vector in the multi-attributes case. For other families of distributions, where no known closed form solution exists, the integrals will be approximated by another suitable distribution. These approximations will be stored in a repository at the query processing module. This means that a large part of the computation will be performed off-line and reused when needed, e.g., by changing the parameters in pre-computed parametric formulas. It is important to distinguish between the approximations in this case, where the answer to the query is computed exactly over the uncertainty models, and the case where these models are used to produce a single point estimation, either to simplify the computations or to be used for in-network query evaluation as we discussed in Section 2. If traditional evaluation algorithms are to be used on the latter case, then the answer itself will be only an approximation and explicit error bounds should be provided. Furthermore, there is no justification for shrinking the uncertainty to a single point, from a statistical perspective. Such a simplification, however, may work well for some types of sensor data and experiments need to be run for investigation. We plan to further study these issues by experimenting with different types of sensor data.

5. EXPERIMENTAL EVALUATION

This section presents a preliminary evaluation of our framework using synthetic data. We are currently building a prototype of our framework and more evaluations using this prototype will be reported in the future. We simulated the readings of 1000 single attribute sensors at a specific time instance using MATLAB. The data was drawn evenly from 5 non-overlapping clusters of data by generating 200 readings randomly from a Gaussian distribution centered around the cluster mean with variance of 100. The cluster means were 1000, 2000, 3000, 4000, and 5000, respectively. These readings represent the unknown true readings of the sensors in our experiment. We used the distribution of the cluster that generated the true reading as the sensor's prior. It is important to notice that the we did not utilize our knowledge of the exact true reading in the prior, rather, we used the distribution of the whole cluster as the prior. In fact our results would have been even better if we used a prior centered around the true reading. We generated the noisy data by adding random noise to each sensor reading, we call this the *raw data*. The noise was generated from a Gaussian distribution with 0 mean. We repeated our experiment at different noise levels (standard deviations) ranging from 5 to 50 with a step of 5. At each level we obtained the posterior distribution of the readings using our proposed approach, we call this *Bayesian data*. We generated random range queries as predicates for evaluation. We repeated for 500 predicates at each noise level and obtained the average error in each case. Due to the limited space we only discuss the single attribute case, however the results for multiple attribute case are comparable. We also include evaluations of some of the queries that we presented above.

SSQ We measured *the mean squared error* between all the true readings and the raw data, and between all the true readings and the Bayesian data at each noise level. Bayesian data was estimated using the mean of the posterior distribution. We repeated the experiment several times with different prior distributions by varying the width of the prior distribution (standard deviation). Our objective is to illustrate the compromise between using the prior and the noisy reading, which we discussed in Section 3, as the noise level increases and to show the error in each case. As shown in Figure 5, our approach indeed reduces the uncertainty of noisy data and therefore yields far less errors. This coincides with the brief analysis of errors that we presented in Section 3.1; since the raw noisy data represents the data of the straightforward approach for modeling uncertainty (by estimating each reading using the mean of its Gaussian distribution). As the noise level increases, the resultant uncertainty model of each sensor reduces to that of its prior and then remains unchanged even when the noise further increases (Equation 3), and hence, the error reduces to the uncertainty associated with the prior. This fact is also illustrated in the figure where the curves of Bayesian data flatten with the large increase in noise.

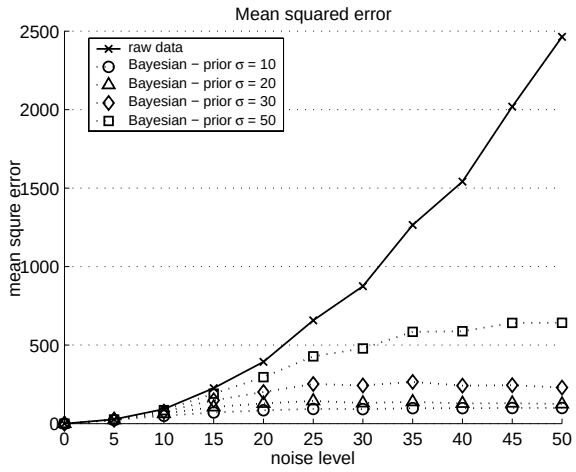

Figure 5: Mean Squared Error at different prior distributions.

SNAQ As we mentioned before, 500 predicates were generated at random at each noise level. The standard deviation of the prior was set to 10 in this experiment. We computed both the set of sensors that satisfy the predicate in case of raw data and in case of Bayesian data and compared them to the true set. Our error metrics in this class are the *Precision* and the *Recall* of the result with respect to the true data. Precision and Recall are *relevance* metrics that are widely used in Information Retrieval [11]. Recall represents the fraction of relevant objects that are retrieved in the answer relative to the total number of true relevant objects in the data set, while Precision represents the fraction of retrieved objects that are relevant. It is clear that methods with high Recall and Precision are favorable since high Recall means low false negatives while high Precision means low false positives. We repeated the experiment for different reasonable confidence levels. As shown in Figure 6, our approach maintains fairly high precision and recall at different confidence levels even in the presence of high levels of noise since the uncertainty in the data is effectively reduced.

SAQ In this class of queries we were interested in *the mean absolute error* of the computed aggregate compared to the true answer (computed on the true data). Similar to the SNAQ class, we generated 500 predicates at random at each noise level to represent the aggregation predicate. We computed the aggregate function in case of raw noisy data and Bayesian data using our proposed algorithms. The standard deviation of the prior was set to 10 while the user-defined confidence was set to 0.5 which is a fairly typical scenario. We evaluated COUNT, SUM aggregates as shown in Figures 7, 8, respectively. The more accurate uncertainty models of our framework yield smaller errors compared to the raw data. The difference in the performance becomes very clear as the noise level increases.

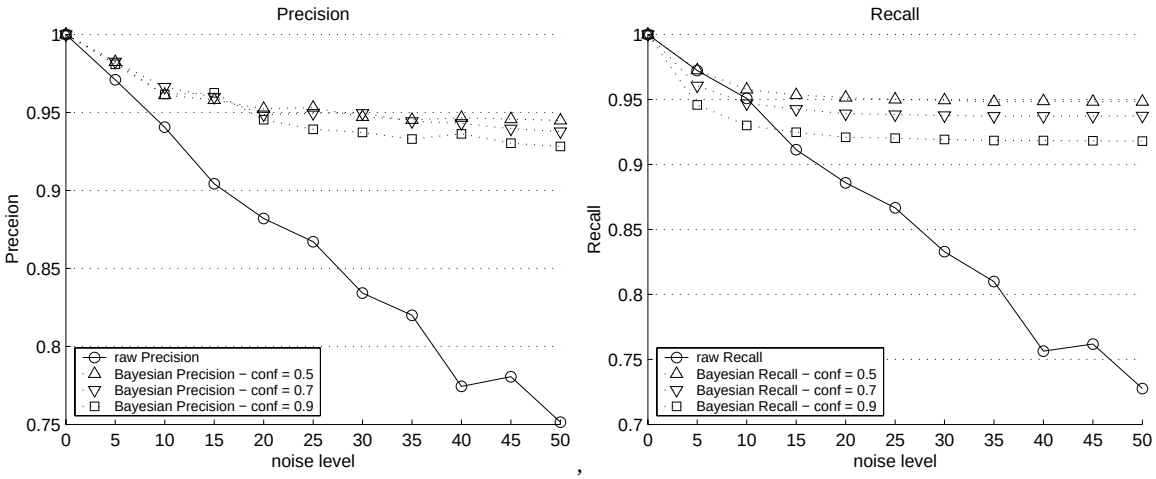

Figure 6: Precision and Recall at different confidence levels.

6. RELATED WORK

Recently, there has been a tremendous amount of work reported in the area of sensor networks, both static sensors and sensors on moving objects. For example, some research has focused on data centric approaches, routing, storage of sensor data, and fault tolerance [25, 24, 14, 13, 5, 16]. In general, in-network processing was proved to be more energy-efficient theoretically and experimentally [19, 30, 13], since valuable communication energy is saved. This motivated the recent work on computing aggregates in sensor networks by processing the query in-network hierarchically, in a distributed fashion [19, 30], and on designing and implementing database functionality [20, 12]. Also, generic architecture for queries over streaming sensors has been proposed in [18]. All these research efforts take into consideration the severe resource constraints of sensor networks, especially, energy and communication constraint, and their unattended deployment potentially in harsh environments. However, this work does not deal with uncertainty in sensor data due to random noise. Data obtained from sensor networks in this research is assumed to be precise and noise-free. Compared to this work, our focus is on noisy sensors. We propose a framework for obtaining accurate models for the true unknown readings of noisy sensors and for querying these models.

General modeling of sensor streams and defining abstractions to represent sensor networks as databases were studied by Gehrke *et al.* as part of their Cougar project [2, 3, 29]. They have also studied indexing and retrieval of noisy sensors in GADT [10]. Specifically, they proposed abstract data types (ADT) and data structures for "indexing" noisy sensors that are represented as pdfs. Our focus, on the other hand, is on reducing the uncertainty associated with noisy sensors, i.e., computing accurate pdfs that represent sensor data, and on general algorithms for computing answers to queries over uncertainty models. Indexing techniques in GADT can then be used over our resultant more accurate uncertainty models. Uncertainty in sensor databases due to lag of updates has been addressed recently in [8]. Due to continuous changes in sensor values and limited network bandwidth and energy, the database state may lag the state of the real world, and therefore, the data inside the database is considered just an estimate of the actual data. The authors assume probabilistic uncertainty models for this problem, i.e., a pdf over a range that is guaranteed to include the current value. Their work, however, does not deal with erroneous noisy data or with reducing their uncertainty as we do. Some of our proposed algorithms for query evaluation bear similarities with their proposed approach since both rely on statistical rules for computing functions over random variables. However, we extended some of their algorithms by defining confidence levels and justifying their use and by proposing algorithms for computing aggregates subject to aggregation conditions (range queries). They consider single attribute sensors only while we also generalized to sensor of multiple attributes. Wolfson *et al.* have studied the problem of uncertainty in the trajectory of moving objects due to lack of perfect tracking of the continuous motion and network delays [22, 28]. Compared to their work, our focus is on reducing uncertainty of the "reported" inaccurate sensor readings, i.e., dealing with inaccuracy and not location prediction. Moving objects can also benefit from our approach. In particular, dynamic priors can be computed (e.g., using information about location and speed of the object, and traffic conditions) and then used for reducing uncertainty of the reported locations. This particular case of reducing uncertainty using dynamically changing priors and Bayes' rule indeed resembles the "measurement step" of Kalman filters [17]. In addition, unlike our proposed algorithms for different classes of queries, the authors focus on range queries, basically due to the nature of their application.

Calibration errors have been addressed very recently in [6]. The authors proposed a post deployment calibration technique. In particular, they derive relative calibration relationships between the sensors by utilizing temporal correlation between co-located sensors and then follow this step by an optimization algorithm. The authors focus only on calibration and assume that any random noise has been suppressed. Our focus however is on dealing with random noise in sensor data and not on calibration errors.

Finally, Bayes-based approaches have been used in literature in the fields of statistics, machine learning, data mining, pattern recognition and estimation theory [21, 11, 27, 9, 15]. In this paper, we utilized a Bayesian approach for obtaining accurate estimates of the true unknown sensor readings. The novelty of our work lies in designing an overall framework that utilizes Bayes' rule and illustrating how to be used for online cleaning of noisy sensor data either at the sensor level or at the base-station. In addition, based on our proposed uncertainty model, we also introduce several algorithms for answering queries over uncertain data. Uncertainty, in general, has received attention in literature especially uncertainty due to incomplete information. Parsons surveys most of the work done in this area from both AI and database perspectives [23]. Uncertainty

has been handled using fuzzy logic and fuzzy theory. However, our focus is on noise which cannot be handled using such an approach since there is no fuzziness or vagueness involved [10].

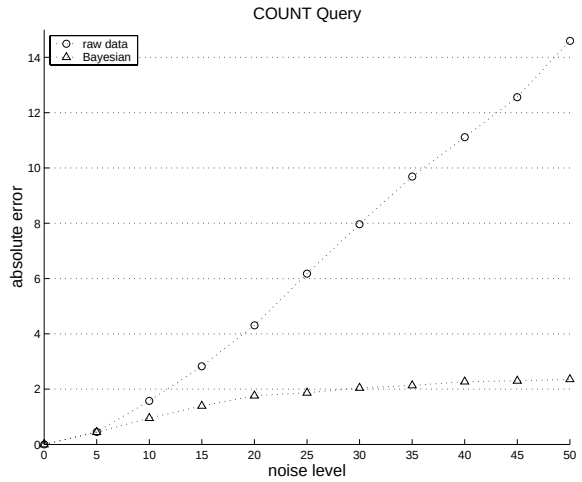

Figure 7: Mean Absolute Error of the COUNT query.

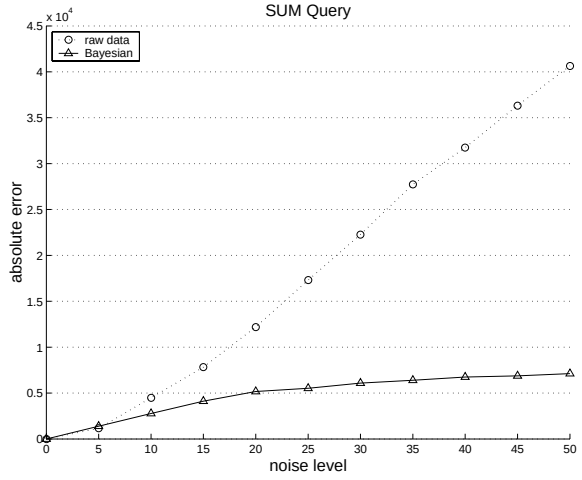

Figure 8: Mean Absolute Error of the SUM query.

7. CONCLUSIONS AND FUTURE WORK

We have highlighted the importance of handling noise in sensor networks. We introduced a framework for cleaning and querying noisy sensors. In particular, we presented a Bayesian approach for reducing the uncertainty associated with noisy sensor data in an online fashion. Bayesian approaches are popular in literature. However, they have not been used for noise cleaning in wireless sensor networks before. The novelty of our work lies in designing a framework that utilizes Bayes' theorem to reduce uncertainty and in illustrating how it is used for online cleaning of noisy sensor data, either at the sensor or at the database levels. Nevertheless, based on our proposed uncertainty models and using a statistical approach, we introduced several algorithms for answering a wide range of traditional database queries over noisy sensors. We also presented a preliminary evaluation of our framework using synthetic data. Other challenges in this area as well as our future work directions can be summarized as follows.

- We are currently building a prototype for our framework in order to explore the real deployment issues. Sensors have different capabilities, noise characteristics and behavior, and therefore, the prototype is needed for further experimentation and characterization. More evaluations from our prototype will be reported in the future.

- Handling other data cleaning problems that cause uncertainty in sensor data such as missing values and outliers. These sources of uncertainty are also common in data obtained from wireless sensor networks. They may severely impact the answer to users' queries as well. Wireless sensors are becoming very pervasive. New applications are emerging every day that rely on these sensors for decision-making; e.g., the perishable items scenario that we introduced from industrial engineering. The future of wireless sensors therefore lies in reasoning about and solving these problems "efficiently", in terms of the available resources, and "online". Existing research on missing values in sensor networks either focused on providing low-level networking solution such as [30], or customized solutions that work for specific applications such as [19]. In both cases, the problem persists though less severely. Hence, a general purpose solution for this problem as well as other sources of uncertainty is needed.

- We discussed only simple traditional database queries in this paper. Addressing more complicated queries as well as optimization issues are part of our future work. In general, an in depth exploration of different future sensor applications along with their potential queries is an important research direction.

- Generalizations to heterogenous sensors and to sampling are challenging problems. Readings obtained from dense sensor network are sometimes highly redundant. In some cases, they may be complementary to each other. Therefore, queries can be evaluated only on a sample of the sensors. However, the sensors in the network may not be homogeneous. They are indeed expected to differ in their remaining energy, storage, processing, and noise effect. A repository is therefore needed at the database system to store metadata about the capabilities and the limitations of each sensor. The database system should be able to turn the sensors on/off or control their rate using proxies, similar to the ones proposed in [18]. The underlying networking functionality should allow for such a scenario.

  Users may define specific quality requirements on the answer to their queries, as part of the query, e.g., a confidence level, the number of false positives/negatives, etc. The challenge then is how to minimize the number of redundant sensors, used unnecessarily to answer a specific query while (1) meeting the given quality level (e.g., confidence) and (2) "best" utilizing the resources of the sensors. We plan to extend our framework to support this scenario. The sample size may need to be increased or specific more accurate sensors may have to be turned on in order to meet the given user's expectations. The sampling methods may have to be changed over the time (random, systematic, stratified, etc.). In general, this introduces another cost factor in decision making and actuation, query optimization and evaluation, and resource consumption. Unfortunately, a large part of existing work on query processing in sensor networks has only focused on homogeneous clean data from all sensors [18, 19, 12], even though there are three dimensions of possible sensor data: homogeneity, uncertainty, and sampling.

8. ACKNOWLEDGEMENT

We would like to thank Professor David Madigan for helpful discussions on probabilistic query evaluation. This research work was supported in part by DARPA under contract number N-666001-00-1-8953 and NSF grant ANI-0240383.

9. REFERENCES

[1] Mica motes: Crossbow technology, inc. Tech. rep. http://www.xbow.com.

[2] BONNET, P., GEHRKE, J., AND SESHADRI, P.Querying the physical world.*IEEE Personal Communications Magazine, Special issue on Networking the Physical World* (October 2000).

[3] BONNET, P., GEHRKE, J., AND SESHADRI, P.Towards sensor database systems.In *Proceedings of the Second International Conference on Mobile Data Management* (January 2001).

[4] BOX, G. E. P., AND TIAO, G. C.*Bayesian Inference In Statistical Analysis*.Addison-Wesley Publishing Company, Inc., 1973.

[5] BRAGINSKY, D., AND ESTRIN, D.Rumor routing algorithm for sensor networks.In *Proceedings of ACM WSNA'02* (2002).

[6] BYCHKOVSKIY, V., MEGERIAN, S., ESTRIN, D., AND POTKONJAK, M.A collaborative approach to in-place sensor calibration.In *Proceedings of IPSN'03* (2003).

[7] CASELLA, G., AND BERGER, R. L.*Statistical Inference*.Duxbury Press, Belmont, California, 1990.

[8] CHENG, R., KALASHNIKOV, D. V., AND PRABHAKAR, S.Evaluating probabilistic queries over imprecise data.In *Proceedings of ACM SIGMOD* (June 2003).

[9] DUDA, R. O., HART, P. E., AND STOCK, D. G.*Pattern Classification*, second ed.John Wiley and Sons, Inc., 2001.

[10] FARADJIAN, A., GEHRKE, J. E., AND BONNET, P.GADT: A probability space ADT for representing and querying the physical world.In *Proceedings of ICDE 2002* (2002).

[11] HAND, D., MANNILA, H., AND SMYTH, P.*Principles Of Data Mining*.The MIT Press, 2001.

[12] HELLERSTEIN, J. M., HONG, W., MADDEN, S., AND STANEK, K.Beyond average: Towards sophisticated sensing with queries.In *Proceedings of IPSN'03* (2003).

[13] INTANAGONWIWAT, C., ESTRIN, D., GOVINDAN, R., AND HEIDEMANN, J.Impact of network density on data aggregation in wireless sensor networks.In *Proceedings of the 22nd International Conference on Distributed Computing Systems* (2002).

[14] INTANAGONWIWAT, C., GOVINDAN, R., AND ESTRIN, D.Directed diffusion: A scalable and robust communication paradigm for sensor networks.In *Proceedings of 6th ACM/IEEE MobiCom'00* (2000).

[15] KAY, S.*Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*.Prentice Hall, 1993.

[16] KRISHNAMACHARI, B., AND IYENGAR, S.Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks.*IEEE Transactions on Computers (to appear)* (2003).

[17] LEWIS, F. L.*Optimal Estimation: With an Introduction to Stochastic Control Theory*.John Wiley and Sons, Inc., 1986.

[18] MADDEN, S., AND FRANKLIN, M. J.Fjording the stream: An architecture for queries over streaming sensor data.In *Proceedings of ICDE* (2002).

[19] MADDEN, S., FRANKLIN, M. J., AND HELLERSTEIN, J. M.TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks.In *Proceedings of 5th Annual Symposium on operating Systems Design and Implementation (OSDI)* (December 2002).

[20] MADDEN, S. R., FRANKLIN, M. J., HELLERSTEIN, J. M., AND HONG, W.The design of an acqusitional query processor for sensor networks.In *Proceedings of ACM SIGMOD* (2003).

[21] MITCHELL, T.*Machine Learning*.McGraw Hill, 1997.

[22] OURI WOLFSON, PRASAD SISTLA, S. C., AND YESHA, Y.Updating and querying databases that track mobile units.*Distributed and parallel databases 7*, 3 (March 1999), 257–387.

[23] PARSONS, S.Current approaches to handling imperfect information in data and knowledge bases.*Knowledge and Data Engineering 8*, 3 (1996), 353–372.

[24] RATNASAMY, S., ESTRIN, D., GOVINDAN, R., KARP, B., SHENKER, S., YIN, L., AND YU, F.Data-centric storage in sensornets.In *Proceedings of ACM WSNA'02* (2002).

[25] RATNASAMY, S., KARP, B., YIN, L., YU, F., ESTRIN, D., GOVINDAN, R., AND SHENKER, S.GHT: A Geographic Hash Table for data-centric storage.In *Proceedings of ACM WSNA'02* (2002).

[26] SADAGOPAN, N., KRISHNAMACHARI, B., AND HELMY, A.The acquire mechanism for efficient querying in sensor networks.In *Proceedings of IEEE SNPA'03* (May 2003).

[27] WITTEN, I. H., AND FRANK, E.*Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations*.Morgan Kaufmann, 2000.

[28] WOLFSON, O., SISTLA, P., CHAMBERLAIN, S., AND YESHA, Y.The geometry of uncertainty in moving objects databases.In *Proceedings of International conference on EDBT* (2002).

[29] YAO, Y., AND GEHRKE, J. E.The cougar approach to in-network query processing in sensor networks.*SIGMOD Record 31*, 3 (September 2002).

[30] ZHAO, J., GOVINDAN, R., AND ESTRIN, D.Computing aggregates for monitoring wireless sensor networks. In *Proceedings of IEEE SNPA'03* (2003).