Title	A statistics-based sensor selection scheme for continuous probabilistic queries in sensor networks
Author(s)	Han, S; Chan, E; Cheng, R; Lam, KY
Citation	The 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2005), Hong Kong, China, 17-19 August 2005. In Proceedings of the 11th RTCSA, 2005, p. 331-336
Issued Date	2005
URL	http://hdl.handle.net/10722/129582
Rights	IEEE International Conference on Embedded and Real-Time Computing Systems and Applications. Copyright © IEEE.

A Statistics-Based Sensor Selection Scheme for Continuous Probabilistic Queries in Sensor Networks

Song Han[1], Edward Chan[1], Reynold Cheng[2], and Kam-Yiu Lam[1]

Department of Computer Science[1]
City University of Hong Kong
Tat Chee Avenue, Kowloon Tong
HONG KONG
{shan00, csedchan, cskylam}@cityu.edu.hk

Department of Computer Sciences[2]
Purdue University
West Lafayette, IN 47907
USA
ckcheng@cs.purdue.edu

Abstract

An approach to improve the reliability of query results based on error-prone sensors is to use redundant sensors. However, this approach is expensive; moreover, some sensors may malfunction and their readings need to be discarded. In this paper, we propose a statistical approach to decide which sensors to be used to answer a query. In particular, we propose to solve the problem with the aid of continuous probabilistic query (CPQ), which is originally used to manage uncertain data and is associated with a probabilistic guarantee on the query result. Based on the historical data values from the sensors, the query type, and the requirement on the query, we present methods to select an appropriate set of sensors and provide reliable answers for aggregate queries. Our algorithm is demonstrated in simulation experiments to provide accurate and robust query results.

1. Introduction

Recent advances in sensor technology have made it possible to develop low-cost sensors, so that large wireless sensor networks with thousands of sensors are well within the realm of reality, and these large sensor networks can support many new applications.

One problem with sensor based monitoring is that the readings are noisy and error-prone [NN04]. A solution is to use multiple sensors to monitor the same region. However, this will increase the consumption of scarce network bandwidth. Also, since some sensors may not work properly, they may generate abnormal readings that skew the average value.

In this paper, we focus on selecting the right set of sensors for multiple sensor aggregation in order to obtain data values that are precise enough to meet the probabilistic requirement of the queries. We partition the sensor network into regions and propose an approach to determine (1) the sampling period for each region adaptively; (2) the sample size and the set of sensors for multiple sensor aggregation within a region at a certain sampling time and (3) the set of regions to be used to obtain the query result while meeting the associated accuracy requirements. This paper is organized as follows. In Section 2 we discuss related works. Section 3 describes the wireless sensor model as well as the underlying sensor data and query models. In Section 4 we present our algorithms that solve the problems of sensor selection while satisfying the prescribed accuracy requirements for a continuous probabilistic query. The performance of our algorithms is studied using simulation experiments and results are discussed in Section 5. Section 6 concludes the paper.

2. Related Works

Researchers have only started to consider the effect of data uncertainty in sensor networks recently. The issues of data uncertainty and probabilistic queries are studied extensively in [CKP03]. Unlike our paper which assumes a wireless sensor network environment, their system model is simple and assumes the host communicates directly with every sensor source, and their method of reducing uncertainty is by sampling hot items more frequently. Our approach, on the other hand, selects appropriate sensors to improve reliability in sensor readings. Also, unlike our paper, they do not study *continuous queries*, and do not allow users to specify probabilistic requirements, which can be seen as a quality guarantee on query results.

The problem of selecting appropriate sensors in a wireless environment is usually framed in the context of improving accuracy in location tracking. In [EFP03] and [LRZ03], mutual information between the distribution of an object's location and the predicted location observed by a sensor is used to compute the information gain due to the sensor. The sensor with the highest information gain is selected to reduce the uncertainty of the sensor reading. Another scheme, based on entropy-based selection heuristics, is claimed to be computationally more efficient than the above mutual-information-based methods [WYPE04]. In a previous paper [LCLC04], we proposed sensor selection algorithms for common types of continuous queries with data uncertainty requirements. While that work represents the first comprehensive work

Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05)
1533-2306/05 $20.00 © 2005 **IEEE**

on *query-based* sensor selection methods, it assumes the regions' values are stable and no experimental results are included. In this paper, we assume the region's value changes continuously, made many refinements to our previous work, and present simulation results to demonstrate the effectiveness of our approach.

3. System Model and Query Model

In this section we briefly describe the underlying system model and the query model. The wireless sensor system model consists of a *base station* (BS) and a collection of` *sensor nodes*. It is assumed that the system environment is divided into a number of *regions*, each of which consists of a node with high computational capability, called *coordinator node* that manages nodes in the same region. BS is responsible for communication between the coordinator nodes and the users of the system and it communicates with the coordinator nodes through a low bandwidth wireless network and may require the relay of other sensor nodes and coordinator nodes. We assume that BS knows the distribution and connections of the coordinator nodes and what sensor data items are represented by each sensor node. Figure 1 illustrates the overall system structure.

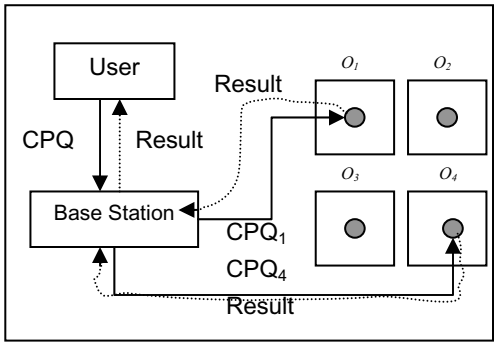

Figure 1: CPQ Processing in the System

A continuous probabilistic query is submitted by users to the sensor network system for the purposes of continuous monitoring and event detection. We can formally define a *Continuous Probabilistic Query (CPQ)* as a probabilistic query repeatedly executed over the time interval [*begin_time*, *end_time*] on objects O_1, O_2... O_n. The answers produced satisfy the *CPQ* with some probability specified by users. A Continuous Probabilistic Sub-Query *i*, denoted as CPQ_i, is a sub-query of *CPQ* executed during the interval {*begin_time*, *end_time*}.It accesses item O_i in the list of objects specified by *CPQ*. It returns to *CPQ* a Gaussian distribution $N(u_i, \sigma_i)$ of O_i.

In this paper, we concentrate on aggregate queries, such as MIN / MAX aggregate queries which return the object that contains the minimum (maximum) value among objects O_1, O_2... O_n with probability guaranteed to be larger than a threshold value *P*.

When the base station receives a *CPQ*, it determines the set of data items required by the *CPQ* according to the required regions of the query and which coordinator nodes are responsible for generating the required data items. The base station then breaks down the *CPQ* into sub-queries $\{CPQ_1, CPQ_2 \ldots CPQ_n\}$. Each sub-query CPQ_i is then sent to the coordinator node, which is responsible for reading O_i and generating a Gaussian distribution for the reading of O_i to describe its distribution. Each coordinator sends its results back to the base station, which then computes the final result and sends it back to the user. Figure 1 illustrates an example of a *CPQ* executing on objects O_1 and O_4 under our system model. The *CPQ* submitted by the user is broken down into two sub-queries, CPQ_1 and CPQ_4, which access regions O_1 and O_4 respectively. The results from coordinators for O_1 and O_4 are sent to the Base Station, which subsequently returns the result to the user.

4. Statistics-based Sensor Selection Scheme

Accessing more sensors can improve the reliability of query results at the expense of an increased aggregation workload. Our goal is to meet the probabilistic requirement of a continuous query using the minimum number of sensors for generating the value of a data item required by a query. *Specifically, what we want to solve is to determine the sampling period for different regions and to determine the set of sensors to participate in sampling for aggregation of the values at the time when a certain region is sampled.*

4.1 Computing a Region's Initial Statistical Properties

In this step, for each region, we calculate its initial statistical properties including the expected value and estimated population variance. The population variance for each region is kept constant during the query period while the expected value will vary at different sampling time as the region's value may change continuously, but it can be evaluated similarly using the selected sensor set described in section 4.5.

For each region R_i required by a sub-query CPQ_i, the coordinator node identifies the set of sensors S_i which are responsible for generating values for R_i. Then it sends out data request messages to all these sensors. Each sensor responds to the request message by returning its latest sampled data value of R_i to the coordinator. The received data values from each sensor are first buffered and the mean values are calculated by the coordinator until a pre-determined waiting time has expired. If the variance of the values from a sensor is higher than a pre-defined threshold, it is assumed that the sensor is either currently located at a high-noise environment or it is currently in an abnormal state. The sensor will be marked as abnormal and the coordinator will not consider it for further processing.

Based on the variances of the values from all the sensors selected, σ_i^P, the population variance for the region R_i can be estimated as their average and will play an important role in the following calculation of the maximum

allowed variance for different kinds of queries and the sampling size in section 4.4 and 4.5 respectively.

4.2 Adaptive Sampling Period

Our system model assumes the value of the region changes continuously. To obtain accurate results for a continuous query, each region needs to be sampled periodically. However the simplistic approach of using a fixed sampling interval for each region can consume excessive bandwidth if the sampling interval is too small while accuracy could suffer if itl is too large.

In this section, we propose sampling with an adaptive sampling period. In this scheme, a region will only be sampled when its value is predicted to affect the result of the query. The key to change the sampling interval is to increase the sampling period for the regions whose values have little effect on the query result, and in this paper, we will focus on the scheme for the MAX and MIN queries.

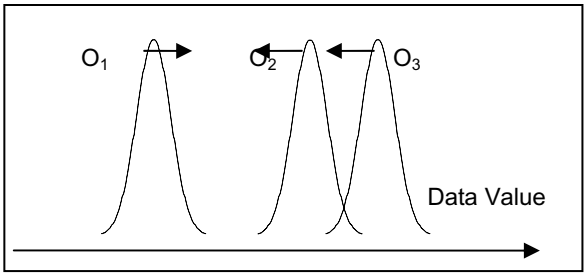

Figure 2: Different significances in region sampling

Here we use the MAX query as an example to illustrate this idea. Figure 2 illustrates the different effect of the regions' values to the query result. The effect from region O_3 is larger than that of O_2, which in turn is larger than that of region O_1. So the sampling period of O_1 will be larger than O_2 while O_3 should have the smallest sampling interval to maintain the accuracy of the result. We now demonstrate how to calculate the sampling period for each region by first introducing the concept of Predicted Sampling Time.

The **Predicted Sampling Time (PST)** of a region is the time when the value of that region will affect the result of a query according to the predicted change rate in the regions' value.

Assume the value of region O_i has a distribution $N(\mu_i, \sigma_i)$ and the region with the largest value O_{max} has the distribution $N(\mu_{MAX}, \sigma_{MAX})$. The rates of change in the values of these two regions are v_i and v_{max} respectively. We also assume when the difference of the two regions' values exceeds $3\times(\sigma_{MAX} + \sigma_i)$, the result of the query will be affected. Then the predicted sampling time for region i can be calculated as:

$$PST_i = \frac{Max((\mu_{\max} - \mu_i - 3\cdot(\sigma_{\max} + \sigma_i)), 0)}{v_i - v_{\max}}$$

The reason why we select $3\times(\sigma_{MAX} + \sigma_i)$ as the threshold is to ensure that the probability is less than *0.3%* when one region's value will be inside the *99.7%* confidence interval of the other region's value.

Given that the actual rate of change in a region's value may be different from the predicted one, setting the sampling period to be the PST can easily produce an incorrect query result. So in the calculation we only use a fraction of the PST, which we call **prediction factor (PF)** to reduce the effect of the prediction process. Its calculation for each region follows:

1. $T_{old} = PST_{last}$ – (Current Sampling Time – Last Sampling Time).
2. $PST_{error} = (PST_{new} - T_{old}) / T_{old}$.
3. if $(PST_{error} > 0)$ $PF = PF + \Delta\sigma$
4. if $(PST_{error} < 0)$ $PF = PF - \Delta\sigma$

Figure 3: Calculation of the Prediction Factor

With the help of the PST and the Prediction Factor, we can calculate the next sampling period for each region as below, and SP_{min} and SP_{max} are the minimum sampling period and the maximum sampling period respectively.

$$Sampling\ \ Period_i = \begin{cases} SP_{\min} & if \quad PST_i = 0 \\ PST_i \cdot PF_i & if \quad PST_i > 0 \\ SP_{\max} & if \quad PST_i < 0 \end{cases}$$

4.3 Region Selection

Based on the adaptive sampling period decision scheme, we predict roughly the potential regions which will affect the query result and take part in the evaluation at sampling time T. Intuitively; we can calculate the query result using the information from all the regions in the system. However, in order to reduce computational overhead, it is possible to eliminate some regions from the calculation because their impact on the query result is very small. In this section we illustrate this idea by showing how to minimize the set of regions for MAX and MIN queries.

Assume there are N regions in the system and the *sampling distribution* from each region is $N(\mu_i,\sigma_i)$ $(i = 1...N)$. Here we define $\mu_m = Max(\mu_i)$ $(i = 1...N)$ and $X \sim N(\mu_m,\sigma_m)$. In this region selection step, we compare X with all the other distributions to test, with a pre-determined level of significance α, whether the information for that region should be included into the calculation. Suppose the distribution of the region for testing is $Y \sim N(\mu_i,\sigma_i)$ $(i = 1...N, i\neq m)$ and the population variances for all the regions are unknown but identical, and the sampling size of X and Y are n_1 and n_2 respectively.

Now we consider the hypotheses testing:
$H_0: \mu_1 - \mu_2 = 0$, $H_1: \mu_1 - \mu_2 \neq 0$ and we introduce

$$T = \frac{\overline{X} - \overline{Y} - (\mu_1 - \mu_2)}{S_w\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

where $S_w^2 = \frac{n_1 S_1^2 + n_2 S_2^2}{n_1 + n_2 - 2}$, $S_1^2 = \frac{1}{n_1}\sum_{i=1}^{n_1}(X_i - \overline{X})^2$ and $S_2^2 = \frac{1}{n_2}\sum_{i=1}^{n_2}(Y_i - \overline{Y})^2$

From statistics, we know that $T \sim t(n_1 + n_2 - 2)$ and given the level of significance α, , we get the rejection region $W = (-\infty, -t_{\frac{1}{2}\alpha}(n_1+n_2-2)\ ] \cup [\ t_{\frac{1}{2}\alpha}(n_1+n_2-2), +\infty)$

If the result is inside the rejection region, we will eliminate this region. By repeating this for all the regions we can reduce the number of regions to be considered in the calculation of the query result.

4.4 Deriving Maximum Allowed Variance

The impact of errors in sampled data values on the query result depends on the query type. In MAX / MIN queries, the data being queried are aggregated from multiple sensors from the same regions by the coordinator nodes. Thus it is reasonable to assume that the data values follow normal distributions with specific means and variances. Basically, the sub-query CPQ_i executed at each coordinator returns a normal distribution of its sensor reading to the base station. The rest of this section describes how the maximum variance allowed for each region comes into play.

One important observation about MAX/MIN queries is: as the variance of the sampled data values decreases, the maximum and minimum become more distinct. In the example of two data values, the probability of O_1 being the maximum data object is:

$$p_1 = \int_{-\infty}^{+\infty} f_1(s) \cdot \left(\int_{-\infty}^{s} f_2(t)dt \right) ds$$

Where $f_1(s)$ and $f_2(t)$ are the probability density functions for O_1 and O_2 respectively. It can be seen from Figure 4 that the variance decreases with increases in p_1. It is consistent with the fact that O_1 is more likely to be the maximum. For the case of multiple data values, suppose the size of the calculation set is N, the probability of O_i being the maximum is:

$$P_i^{MAX} = \prod_{j=1 \wedge j \neq i}^{N} P\{O_i > O_j\} = \prod_{j=1 \wedge j \neq i}^{N} \left(\int_{-\infty}^{+\infty} f_i(s) \cdot \left(\int_{-\infty}^{s} f_j(t)dt \right) ds \right)$$

where the probability density functions are:

$$f_i(s) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(s-\mu_i)^2}{2\sigma_i^2}} \text{ and } f_j(t) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(t-\mu_j)^2}{2\sigma_j^2}}$$

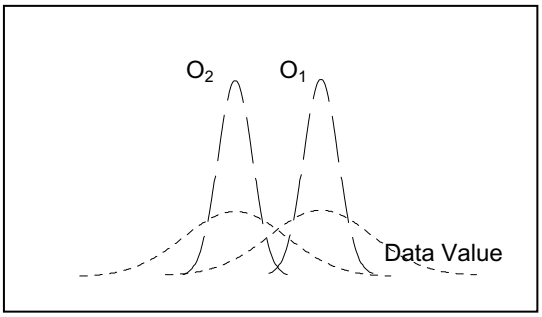

Figure 4: MAX and MIN queries

The algorithm below finds the maximum allowed variance for each region to satisfy the requirement that the probability of the region holding the maximum or minimum value is larger than P.

1. Set σ_{req}'s as σ_i^P for each region.
2. If (Type is MAX) $P = \prod_{j=1 \wedge j \neq i}^{N} \left(\int_{-\infty}^{+\infty} f_i(s) \cdot \left(\int_{-\infty}^{s} f_j(t)dt \right) ds \right)$

   else $P = \prod_{j=1 \wedge j \neq i}^{N} \left(\int_{-\infty}^{+\infty} f_i(s) \cdot \left(1 - \int_{-\infty}^{s} f_j(t)dt \right) ds \right)$
3. Find $k_{\max}$, the index of the max $\frac{\partial}{\partial \sigma_k} P(\sigma_1, \sigma_2, ..., \sigma_n)$
4. Adjust variance requirement of the k_{max}^{th} sensor: $\sigma_{k_{\max}} = \sigma_{k_{\max}} - \Delta\sigma$
5. Repeat 2 to 4 until $P(\sigma_1, \sigma_2, ..., \sigma_n) \geq P\%$
6. Return $\sigma_1, \sigma_2, \ldots, \sigma_n$, as σ_{req}'s

Figure 5: Algorithm for determining the MAV

Although this algorithm will be executed in the base station which is supposed as a powerful PC, one disadvantage is that there is a loop from step 2 to step 4. The time to execute the steps in the loop depends on the number of regions taking part in the calculation and the step length $\Delta\sigma$. If the step length is large, the query accuracy cannot be guaranteed, while if it is small, the process will consume a long time which is a vital disadvantage in a real time system. An important observation about k_{max} is that it always lies in the regions with top values, because only the regions with top values will affect the query result greatly. It is possible to derive an optimization algorithm such that all the variances σ_{req}'s are constants while just letting the top two regions' variances be variables. In this way, with a suitable optimization condition, we get the variances satisfying the query condition efficiently. Suppose the regions with top two values satisfy $S_1 \sim N(\mu_1, \sigma_1)$ and $S_2 \sim N(\mu_2, \sigma_2)$ respectively. We define:

$$F(\sigma_1, \sigma_2) \equiv \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{1}{2}(\frac{s-\mu_1}{\sigma_1})^2} \left(\frac{1}{2} + \Phi\left(\frac{s-\mu_2}{\sigma_2}\right) \right) ds$$

$$P_i^{MAX} = F(\sigma_1, \sigma_2) \cdot \prod_{j=1 \wedge j \neq i, j \neq k}^{N} \left(\int_{-\infty}^{+\infty} f_i(s) \cdot \left(\int_{-\infty}^{s} f_j(t)dt \right) ds \right)$$

Here we suppose $\prod_{j=1 \wedge j \neq i, j \neq k}^{N} \left(\int_{-\infty}^{+\infty} f_i(s) \cdot \left(\int_{-\infty}^{s} f_j(t)dt \right) ds \right)$ is a constant C (in fact, with the increase of σ_1 and σ_2. C will increase, but the increase will be small). Our goal is to get the σ_1 and σ_2 when

$$P_i^{MAX} = F(\sigma_1, \sigma_2) \times C = P$$

where P is the accuracy requirement for the query.

Considering the condition that the sum of sensors to be selected should be smallest, we need to minimize

$$O(\sigma_1, \sigma_2) = N_{s1} + N_{s2} = \frac{(\sigma_1^P)^2}{\sigma_1^2} + \frac{(\sigma_2^P)^2}{\sigma_2^2}$$

and $O(\sigma_1, \sigma_2) \geq 2 \cdot \frac{\sigma_1^P \sigma_2^P}{\sigma_1 \sigma_2}$ only when $\sigma_2 = \frac{\sigma_1 \sigma_2^P}{\sigma_1^P}$

Here N_{s1} and N_{s2} are the sampling size for region 1 and region 2 respectively, and details about calculating the sampling size can be found in section 4.5.

With the simple approximations of $\Phi(x)$ and the relationship between σ_1 and σ_2, function $F(\sigma_1, \sigma_2)$ is changed to a uni-variable function F (σ_1).

$$\Phi(x) \approx \begin{cases} 0.1x(4.4-x) & for \quad 0 \le x \le 2.2 \\ 0.49 & for \quad 2.2 < x < 2.6 \\ 0.50 & for \quad x \ge 2.6 \end{cases}$$

With the condition $P_i^{MAX} = F(\sigma_1, \sigma_2) \times C$, then we can use dichotomy method to get the σ_1 satisfying:

$$\begin{cases} P_i^{MAX} > F(\sigma_1 - \Delta\sigma) \cdot C \\ P_i^{MAX} < F(\sigma_1) \cdot C \end{cases}$$

In this way, it will be more efficient to evaluate the required variances because only parts of the regions will take part in the evaluation and more importantly, we can get rid of the step 3 in the previous algorithm in which we have to find the region whose impact to the query result is the maximum one.

4.5 Determining Sample Size and the Set of Sensors

In this step, based on the information of the variance σ_{req} transmitted from the base station, the coordinator node in each region determines the sample size and the set of sensors to be sampled to meet the confidence requirement of data being queried.

We first determine the sample size. Suppose the sample size is n_s and the approximate mean value is $\overline{S} = \frac{1}{n_s}\sum_{i=1}^{n_s} s_{k_i}$, where $1 \le k_i \le n$ and $k_i \ne k_j$ for all $i \ne j$. We know that if all S_{ki} follow an identical distribution $N(\mu,\sigma^2)$, then $\overline{S}$ follows the normal distribution $N(\mu,\sigma^2/n_s)$, where μ is the expected value and σ is the region's estimated population variance calculated in section 4.1. To satisfy the accuracy requirement, we need to choose an n_s value satisfying the constraint $\sigma/\sqrt{n_s} \le \sigma_{req}$. So we set the sample size as

$$n_s = \lceil \sigma^2 / \sigma_{req}^2 \rceil.$$

Next we determine the set of sensors to be sampled. For each sensor, we calculate the difference between the sensor data and the expected value for the region and set it as the criteria for sorting the sensors in each region for selection.

$$d_i = s_i - E(s)$$

We sort the sensors in ascending order of d_i. At each sampling time, with a certain variance σ_{req}, the coordinator will calculate the sampling size n_s and select the top n_s sensors to sample.

Since the selected sensor could be in an error state during the sampling period, so at each sampling time, when the coordinator collects all the possible sensor data with a small delay, it re-calculates d_i to see whether a sensor's value exceeds the expectation a pre-fixed threshold, if the threshold is exceeded, we assume that the sensor is in error at the sampling time, and the coordinator will send a new request information to other sensor candidates in the sorted list and it will also re-sort the sensor list for the further use.

5. Performance Evaluation

Parameter	Baseline Value
Continuous query length	1000 sec
Initial calculation period	100 sec
Sensor sampling interval	5 sec
Accuracy Requirement	95%
Variance Change Step ($\Delta\sigma$)	0.3
Number of regions	1 ~ 4
No. of sensors in each region	U [100,150]
region value's difference	2% ~ 10%
Sensor error variance range	5% ~ 25%

Table 1: Experiment Parameters

In this section we evaluate the performance of our scheme using a number of simulation experiments. We compare our scheme with a baseline method where sensors are selected randomly. In Figure 6, we demonstrate the percentage of the sensors selected based on our algorithm. For simplicity, we assume there are two regions in the system in considering a MAX / MIN query and the actual value for these two regions do not change. The model of the sensor's value at time t, S^t is defined as follows:

$$\begin{cases} S^t = S^t_{actual} + error^t \\ error^t \sim N(0,\sigma) \\ \sigma = \lambda \cdot S^t_{actual} \end{cases}$$

S^t_{actual} is the region's actual value at time t and $error^t$ satisfy normal distribution $N(0, \sigma)$. λ is called the sensor's *Error Variance Percentage* which is the percentage of the region's actual value and decides the size of the σ, in this experiment we suppose it satisfies uniformly distributed *U [0%, 15%]*. It can be seen from Figure 6 that the sensor selection percentage is sensitive to the difference between the two regions' value. The selection percentage can be reduced dramatically to 5% of the total number of sensors when the difference is about 10% of the region's value. On the other hand, even when the difference is as small as 2%, we still can reduce the selected percentage to 65%. More importantly, accuracy of the query result is maintained. Figure 7 shows that our scheme outperforms random sensor selection consistently; it is able to maintain the required accuracy requirement of 95% even though the difference of the regions values is only 2%. Figure 8 demonstrates that while accuracy for random selection of sensors decreases with increasing sensor error variance, the accuracy for our scheme is basically unaffected by the sensor' error variance because it only selects the sensors whose values are closed to the

expected values of the regions. Of course there is an additional cost when the sensors' error variance increases. This is illustrated in Figure 9, which shows the relationship between the percentage of sensors selected and the sensor's Error Variance Percentage. As expected, a region's selection percentage increases with the sensors' error variance.

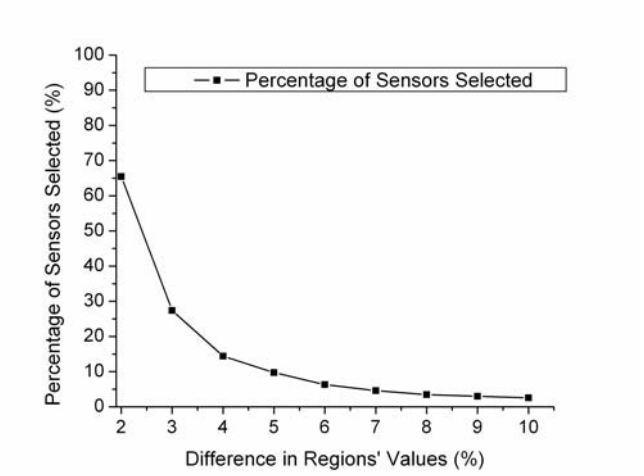

Figure 6: Percentage of Sensor Selected vs. Difference in Regions' Values

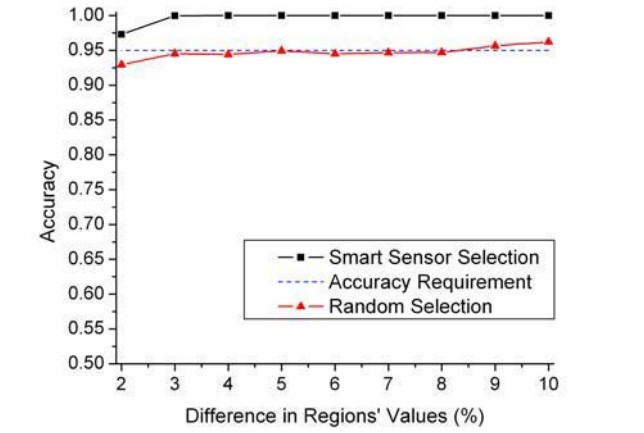

Figure 7: Accuracy vs. Difference in Regions' Values

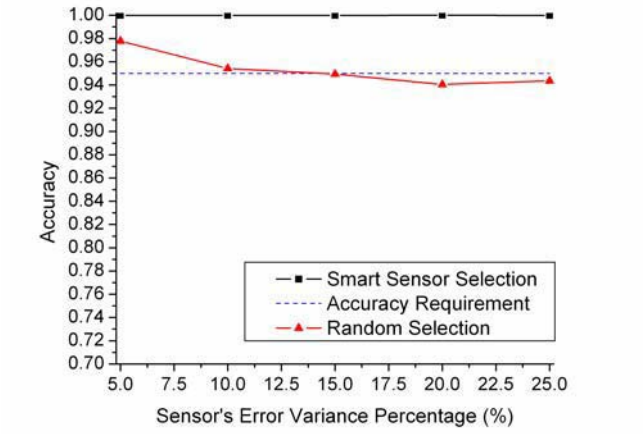

Figure 8: Accuracy vs. Sensor Error Variance Percentage

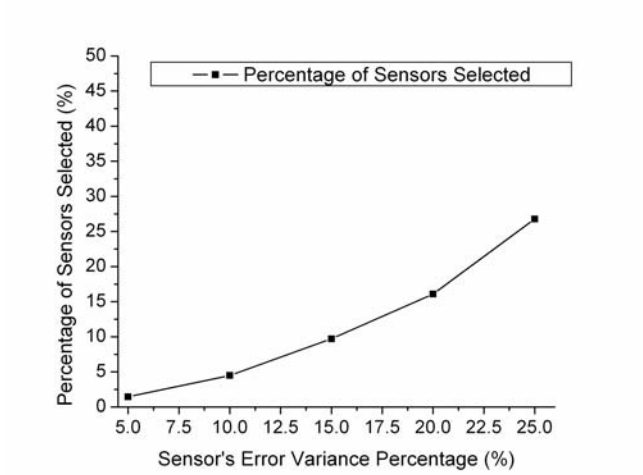

Figure 9: Percentage of Sensors Selected vs. Sensor Error Variance

6. Conclusion

With prices of sensors continuously dropping, we expect that more applications will deploy large sensor networks for monitoring purposes. In this paper, we exploit the availability of low-cost sensors and develop a comprehensive scheme that selects appropriate sensors to provide reliable query results. We devise a probabilistic approach to select sensors intelligently to efficiently execute CPQs for these common aggregate query types. Our simulation results show that we meet the required accuracy requirements with a much smaller set of sensors than random selection of sensors.

Acknowledgement

This work was supported in part by a strategic research grant from City University of Hong Kong [Project No.7001472].

Reference

[CKP03] R. Cheng, D. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data,* June 2003.

[EFP03] E. Ertin, J. Fisher, and L. Potter. Maximum mutual information principle for dynamic sensor query problems. In *Proc. IPSN'03, Palo Alto, CA,* April 2003.

[LCLC04] K. Y. Lam, R. Cheng, B.Y. Liang and J. Chau. Sensor Node Selection for Execution of Continuous Probabilistic Queries in Wireless Sensor Networks. In *Proc. of ACM 2nd Intl Workshop on Video Surveillance and Sensor Networks*, New York, USA, Oct 2004.

[LRZ03] J. Liu, J. Reich, and F. Zhao. Collaborative in-network processing for target tracking. In *EURASIP JASP: Special Issues on Sensor Networks*, 2003(4):378-391, March 2003.

[NN04] D. Niculescu and B. Nath. Error characteristics of ad hoc positioning systems. In *Proc. ACM Mobihoc*, Tokyo, Japan, May 2004.

[WYPE04] H. Wang, K. Yao, G. Pottie, and D. Estrin. Entropy-based Sensor Selection Heuristic for Localization. In *Proc. IPSN'04,* April 2004.