Wireless Sensor Network, **2010, 2, 675-682**
doi:10.4236/wsn.2010.29081 Published Online September 2010 (http://www.SciRP.org/journal/wsn)

Approximate Continuous Aggregation via Time Window Based Compression and Sampling in WSNs

Lei Yu, Jianzhong Li, Siyao Cheng
Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
E-mail: {*yulei*2008,*lijzh*}*@hit.edu.cn, csyhit@*126*.com*
Received July 4, 2010; *revised August* 7, 2010; *accepted September* 10, 2010

Abstract

In many applications continuous aggregation of sensed data is usually required. The existing aggregation schemes usually compute every aggregation result in a continuous aggregation either by a complete aggregation procedure or by partial data update at each epoch. To further reduce the energy cost, we propose a sampling-based approach with time window based linear regression for approximate continuous aggregation. We analyze the approximation error of the aggregation results and discuss the determinations of parameters in our approach. Simulation results verify the effectiveness of our approach.

Keywords: Approximate Aggregation, Continuous Aggregation, Sampling, Sensor Network

1. Introduction

Wireless sensor networks (WSNs) offer a powerful and efficient approach for monitoring and collecting information in a physical environment. To extract the summary information about the monitored environment, the aggregations of sensed data, such as sum and average, are common interesting queries for users. Therefore, a lot of algorithms and protocols for aggregate query processing in WSNs are proposed [1-8].

The existing works addressed two types of aggregate queries which include exact and approximate aggregate queries. The exact aggregate query requires all the sensed data to be involved in aggregation computation to obtain the exact aggregation results [1,2]. However, the exact aggregate query processing often incurs great energy consumption and is also very sensitive to the packet loss and node failure during the data aggregation. Considering the approximate aggregation results would be enough to reflect the information of the environment, approximate aggregate query processing is addressed to save energy and achieve robustness against the failure of the links and nodes [3-8]. In the research of the approximate aggregate query processing in WSNs, sampling is widely used as a powerful and energy-efficient technique to obtain the statistical information of the environment. A number of sampling based schemes have been proposed for approximate query processing in WSNs [8-10].

In the applications of WSNs such as monitoring air pollution and water quality, the users are often interested in understanding how the environment changes over time and observing data trend in a time window. In such cases, continuous aggregation of sensed data is usually required. In a continuous aggregation, the query aggregation period is divided into epochs and one aggregate answer is provided at each epoch. The existing aggregation schemes usually compute every aggregation result in a continuous aggregation either by a complete aggregation procedure [1,2-4,7] or by partial data update [8] at each epoch. However, the users, who are interested in the time-evolving characteristic of aggregation results, are more concerned about the data trend rather than each individual accurate aggregation result. On the other hand, the communication cost of the existing schemes could be substantial, especially for continuous query with a short epoch and a long period. Motivated by such circumstances, we propose a sampling-based approach with time window based compression for approximate continuous aggregation.

Our approach leverages the batch-based design to compute a period of aggregation results at one time. While giving a series of good approximate aggregation results to provide accurate data trend information, it achieves greater energy-savings than the existing approaches by avoiding individual computation cost of every epoch. In our approach, the combination of data compression and sampling techniques is exploited. A small portion of sensor nodes transmit to the base station

(BS) a compact description of their sensor readings during a time window. The BS computes approximation aggregation results of every epoch in this time window. In this paper, linear regression modeling is adopted by sensor nodes to compress their sensor data in a time window. We analyze the approximation error of the aggregation results and discuss the determinations of parameters in our approach.

The rest of the paper is organized as follows. We present our approach and approximation error analysis in Section 2. We discuss the determination of parameters in our approach in Section 3. Simulation results are presented in Section 4. Finally, we conclude this paper in Section 5.

2 Approximate Continuous Aggregations

2.1 System Model and Time Window Based Framework

We assume a multi-hop sensor network with *N* number of sensor nodes. The BS knows *N*. All the sensor nodes and the base station are loosely time synchronized. Each node has the same communication radius R_c. We assume a continuous querying environment for sensor networks. For a continuous aggregation query, the base station initially disseminates a query into the network, consisting of the epoch duration, the lifetime of the query evaluation and a sampling ratio ϱ.

During the period of a continuous aggregation query, aggregation computation is conducted at time intervals. Each time interval consists of l number of successive epochs. The BS computes the aggregation result of every epoch in a time interval at one time. Such a time interval is referred to as time window and represented by $[t+1, t+l]$. l is the time window size. Let $Ag_{t+1}, \ldots, Ag_{t+l}$ denote the aggregation results from l successive epochs $t+1, \ldots, t+l$.

In the network, the aggregation computation involves sampling sensor nodes that participate in answering the aggregation query, and collecting a compressed representation of sensor readings within a time window from each sampled node.

After receiving the query from the BS, each sensor node u generates a random number rn_u in the range of [0, 1). If $rn_u \le \varrho$, u is sampled for the aggregation query, otherwise u is not sampled. Let $S = \{s_i \mid 1 \le i \le m\}$ (m is the sample size) be the set of sampled nodes. At the end of a time window $[t+1, t+l]$, each node $s_i \in S$ generates a compressed representation M_i of its sensing readings $\{r_{i,t+1}, r_{i,t+2} \ldots, r_{i,t+l}\}$ that contributes to the aggregation in the time window $[t+1, t+l]$. The generation of M_i depends on the specific data compression method we adopted. After that, s_i transmits M_i to the BS. The BS reconstructs the sensor readings of every sampled node s_i by M_i, denoted by $\{\hat{r}_{i,t+1}, \hat{r}_{i,t+2}, \ldots, \hat{r}_{i,t+l}\}$, and computes an approximation answer $\widehat{Ag}_k (t+1 \le k \le t+l)$ for a specific aggregation query.

Definition 1. ((ε, δ)*-approximation aggregation*): Let A_k be a true aggregation result of epoch k, $\widehat{Ag}_k$ is called as (ε, δ) -approximation of A_k, if $\Pr(|\widehat{Ag}_k - A_k| \ge \varepsilon) \le \delta$.

2.2. Modeling Sensor Data with Error Constraint

In our framework, a sample is not a single sensor reading but a compressed representation of the sensor readings, which enables a sensor node to transmit its sensing readings in a time window with less communication cost. It can be built by either lossy or lossless compression methods.

Considering the inherent redundancy of sensor data and the fundamental limit of lossless compression in information theory, we use a data modeling approach, linear regression, to achieve a lossy compression of sensor readings. Linear regression has been widely used to characterize data in sensor networks and answer aggregation queries [11-13]. On this basis, lossless compression methods always can be used for any possible further size reduction. Nevertheless, we note that our framework does not depend on any particular compression method. However, data compression with linear regression modeling would introduce errors in the reconstructed data. Therefore, we put error constraints on the modeling process in our approach. If sampled nodes find that the variance of error incurred by modeling exceeds some threshold σ_T^2, referred to as *error constraint*, they choose to transmit their original data. Otherwise, model parameters including error variance are transmitted.

2.2.1. Linear Regression Model

Regarding the sensor readings $r_{t+1}, \ldots, r_{t+l}$ of a node in each time window $[t+1, t+l]$ as a function of the sequence number from 1 to l, a linear regression model [14] for these sensor readings is built in the following form

$$\mathbf{R} = \mathbf{X}\Theta + \xi \tag{1}$$

where $\mathbf{R} = (r_{t+1}, \ldots, r_{t+l})^T$, $\Theta = (\theta_0, \theta_1, \ldots, \theta_p)^T$,

$$\mathbf{X} = \begin{pmatrix} h_0(1) & h_1(1) & \ldots & h_p(1) \\ h_0(2) & h_1(2) & \ldots & h_p(2) \\ \vdots & \vdots & \vdots & \vdots \\ h_0(l) & h_1(l) & \ldots & h_p(l) \end{pmatrix} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ 1 & 2 & \ldots & 2^p \\ \vdots & \vdots & \vdots & \vdots \\ 1 & l & \ldots & l^p \end{pmatrix},$$

$\xi = (\varepsilon_{t+1}, \ldots, \varepsilon_{t+l})^T$.

In the model, $\{h_i(x) \mid h_i(x) = x^i, 0 \le i \le p\}$ are the set of basis functions, $\theta_0, \theta_1, \ldots, \theta_p$ are regression coefficients, and ξ is a random error vector. Besides, the time window size l is larger than $p+1$. According to Gauss-Markov conditions [14], we also have $E(\varepsilon_i) = 0$, $Var(\varepsilon_i) = \sigma^2$ and $Cov(\varepsilon_i, \varepsilon_j) = 0$ where $i \neq j$, $i, j \in \{t+1, \ldots, t+l\}$.

By the least square estimate, the estimation of regression coefficients, denoted by $\widehat{\Theta} = (\hat{\theta}_0, \hat{\theta}_1, \ldots, \hat{\theta}_p)$, can be computed by solving the following matrix equation, using, for example, Gaussian elimination:

$$\mathbf{A}\widehat{\Theta} = \mathbf{b} \tag{2}$$

where $\mathbf{A} = \mathbf{X}^T\mathbf{X}$, $\mathbf{b} = \mathbf{X}^T\mathbf{R}$.

Once determining l and p, we can see that the matrices $\mathbf{X}$ and $\mathbf{A}$ do not change with $\mathbf{R}$, so they just need to be computed only once for an aggregation query.

2.2.2. Error Variance and Data Reconstruction

Besides computing regression coefficients $\widehat{\Theta}$, each sampled node also needs to estimate the variance of the errors, denoted by σ^2, to decide whether to transmit original data or regression coefficients.

Under Gauss-Markov conditions [14], an unbiased estimator of error variance σ^2 can be computed by

$$\widehat{\sigma}^2 = \frac{(\mathbf{R} - \mathbf{X}\widehat{\Theta})^T(\mathbf{R} - \mathbf{X}\widehat{\Theta})}{l - p - 1} \tag{3}$$

Given an error constraint σ_T^2, if $\widehat{\sigma}^2 \le \sigma_T^2$, the node transmits $p+1$ number of regression coefficients $\widehat{\Theta} = (\hat{\theta}_0, \hat{\theta}_1, \ldots, \hat{\theta}_p)$ and $\widehat{\sigma}^2$ to the base station. Otherwise, it transmits l number of original sensor readings.

By the regression coefficients of $\widehat{\Theta}$ received from a sampled node, the BS can reconstruct its sensor readings $\widehat{\mathbf{R}} = (\hat{r}_{t+1}, \ldots, \hat{r}_{t+l})$ in the time window by

$$\widehat{\mathbf{R}} = \mathbf{X}\widehat{\Theta} \tag{4}$$

where $\mathbf{X}$ can be pre-computed by the BS with l and p.

In the rest of this paper, we regard both the original readings and the regression coefficients as model parameters and do not distinguish them. A sample transmitted by a sampled node s_i is denoted by $M_i = (\widehat{\Theta}_i, \widehat{\sigma}_i^2)$. When $M_i = (\widehat{\Theta}_i, 0)$, M_i represents the original sensor readings.

2.3. Approximate Aggregation

2.3.1. Aggregation Estimation

At the end of each time window, the BS waits for the arrivals of all samples for some time t_w. The waiting time t_w should be larger than the maximum time needed for the message delivery from the samples node to the BS.

After reconstructing sensor readings $\{\hat{r}_{i,k} \mid 1 \le i \le m\}$ of sampled nodes $\{s_i \mid 1 \le i \le m\}$ at epoch k in a time window $[t+1, t+l]$ by Formula (4), the approximation aggregation result $\widehat{Ag}_k$ of epoch k $(t+1 \le k \le t+l)$ can be obtained by

$$\widehat{Ag}_k = F(\hat{r}_{1,k}, \hat{r}_{2,k}, \ldots, \hat{r}_{m,k}) \tag{5}$$

where F is the estimator function of aggregation results. Now we specifically discuss how to estimate the results of aggregation queries including Average and Sum respectively.

Average Average aggregation is estimated by

$$\hat{A}_k^a = \frac{1}{m}\sum_{i=1}^{m} \hat{r}_{i,k} \tag{6}$$

Sum Sum aggregation is estimated by

$$\hat{A}_k^a = N\hat{A}_k^a = \frac{N}{m}\sum_{i=1}^{m} \hat{r}_{i,k} \tag{7}$$

2.3.2. Approximation Error Analysis

Let $\hat{\varepsilon}_{i,k} = r_{i,k} - \hat{r}_{i,k}$. If the estimator function F is a linear function, Formula (5) can be rewrote as

$$\begin{aligned} \widehat{Ag}_k &= F(r_{1,k} - \hat{\varepsilon}_{1,k}, r_{2,k} - \hat{\varepsilon}_{2,k}, \ldots, r_{m,k} - \hat{\varepsilon}_{m,k}) \\ &= F(r_{1,k}, r_{2,k}, \ldots, r_{m,k}) - F(\hat{\varepsilon}_{1,k}, \hat{\varepsilon}_{2,k}, \ldots, \hat{\varepsilon}_{m,k}) \end{aligned} \tag{8}$$

where $r_{i,k}$ is the original data of epoch k and $\hat{\varepsilon}_{i,k}$ is the residual in the linear regression model (1) of node s_i.

Then, the approximation error of $\widehat{Ag}_k$ to the exact aggregation result A_k of epoch k is

$$\begin{aligned} &|\widehat{Ag}_k - A_k| \\ &= |F(r_{1,k}, r_{2,k}, \ldots, r_{m,k}) - A_k - F(\hat{\varepsilon}_{1,k}, \hat{\varepsilon}_{2,k}, \ldots, \hat{\varepsilon}_{m,k})| \\ &\le \underbrace{|F(r_{1,k}, r_{2,k}, \ldots, r_{m,k}) - A_k|}_{\text{sampling estimation error}} + \underbrace{|F(\hat{\varepsilon}_{1,k}, \hat{\varepsilon}_{2,k}, \ldots, \hat{\varepsilon}_{m,k})|}_{\text{modeling estimation error}} \end{aligned} \tag{9}$$

where $|F(r_{1,k}, \ldots, r_{m,k}) - A_k|$ is the estimation error with original data samples, referred to as *sampling estimation error*, and $|F(\hat{\varepsilon}_{1,k}, \ldots, \hat{\varepsilon}_{m,k})|$ is referred to as *modeling estimation error*. The above result indicates the approximation error consists of two types of errors including sampling estimation error and modeling estimation error. Because these two errors separately rely on different factors such as the sample size or the number of regression coefficients, we regard them as two independent random variables.

Now we specifically analyze the approximate error of Average and Sum. Let A_k^a and A_k^s be the exact average and sum result of epoch k $(t+1 \le k \le t+l)$ respectively, *i.e.*, $A_k^a = \frac{1}{N}\sum_{i=1}^{N} r_{i,k}$ and $A_k^s = \sum_{i=1}^{N} r_{i,k}$. By Formula (8), we have

$$\hat{A}_k^a = \frac{1}{m}\sum_{i=1}^{m} r_{i,k} - \frac{1}{m}\sum_{i=1}^{m} \hat{\varepsilon}_{i,k}$$

As we can see, $\hat{A}_k^a$ is a linear combination of two random variables R_k and Z_k, $R_k = \frac{1}{m}\sum_{i=1}^{m} r_{i,k}$ and $Z_k = \frac{1}{m}\sum_{i=1}^{m} \hat{\varepsilon}_{i,k}$.

According to the linear regression theory, under Gauss-Markov conditions, the residual $\hat{\varepsilon}_{i,k}$ follows a normal distribution $N(0, \sigma_i^2(1-p_{k'k'}))$ where σ_i^2 is the error variance in the linear model at node s_i, $k' = k - t$ and $p_{k'k'}$ is the k'-th element on the principal diagonal of matrix $P_X = X(X^T X)^{-1} X^T$. Considering that $\hat{\sigma}_i^2$ in M_i is an unbiased estimator of σ_i^2, we have

$$Z_k \sim N(0, \frac{1-p_{k'k'}}{m^2}\sum_{i=1}^{m} \hat{\sigma}_i^2) \quad (10)$$

Since R_k is the mean of original data samples, according to the general results in the sampling theory [15], we have the following results

$$E(R_k) = A_k^a$$

$$Var(R_k) = \frac{S_k^2}{m}(1-\frac{m}{N})$$

$$S_k^2 = \frac{1}{N-1}\sum_{i=1}^{N}(r_{i,k} - A_k^a)^2$$

Confidence Interval:

$$\Pr\left(R_k - \varphi_{\frac{\alpha}{2}} s_k \sqrt{\frac{1-f}{m}} \le A_k^a \le R_k + \varphi_{\frac{\alpha}{2}} s_k \sqrt{\frac{1-f}{m}}\right) = 1-\alpha, \quad (11)$$

where $f = m/N$, $\varphi_{\frac{\alpha}{2}}$ is the upper $\alpha/2$ point on the standard normal distribution, $s_k^2 = \frac{1}{m-1}\sum_{i=1}^{m}(r_{i,k} - R_k)^2$ is the (unbiased) sample variance and is an unbiased estimator of the population MSE(Mean Square Error) S_k^2.

By the above discussions, we have the following results

Lemma 1. *Under Gauss-Markov conditions*,

$$\mathrm{E}(\mathrm{r}\, E(r_{i,k}\hat{\varepsilon}_{j,k}) = \begin{cases} 0, & i \ne j \\ \sigma_i^2(1-p_{k'k'}) & i = j \end{cases}$$

Proof. If $i \ne j$, since the samples $r_{i,k}$ and $r_{j,k}$ are assumed to be independent random variables in the sampling theory, $r_{i,k}$ and $\hat{\varepsilon}_{j,k}$ are independent and we have

$$E(r_{i,k}\hat{\varepsilon}_{j,k}) = E(r_{i,k})E(\hat{\varepsilon}_{j,k}) = 0.$$

If $i = j$, according to the linear regression theory, we know $\hat{\varepsilon}_{i,k}$ and $\hat{\theta}_{i,x}$ $(0 \le x \le p)$ are independent, thus $E(\hat{\theta}_{i,x}\hat{\varepsilon}_{i,k}) = E(\hat{\theta}_{i,x})E(\hat{\varepsilon}_{i,k}) = 0$. Then, we have

$$E(r_{i,k}\hat{\varepsilon}_{i,k}) = E[(\hat{\theta}_{i,0} + \hat{\theta}_{i,1}k' + \hat{\theta}_{i,2}k'^2 + \ldots + \hat{\theta}_{i,p}k'^p + \hat{\varepsilon}_{i,k})\hat{\varepsilon}_{i,k}]$$
$$= E(\hat{\varepsilon}_{i,k}^2) = \sigma_i^2(1-p_{k'k'})$$

Theorem 1. *Let* $\hat{s}_k^2 = \frac{1}{m-1}\sum_{i=1}^{m}(\hat{r}_{i,k} - \hat{A}_k^a)^2$ *and* $k' = k - t$. *Then*

$$E(\hat{s}_k^2) = S_k^2 - \frac{1-p_{k'k'}}{N}\sum_{i=1}^{N}\sigma_i^2. \quad (12)$$

Proof. It can be easily shown that

$$\hat{s}_k^2 = \frac{1}{m(m-1)}\sum_{\substack{i=1 \\ i<j}}^{m}\sum_{j=1}^{m}(\hat{r}_{i,k} - \hat{r}_{j,k})^2$$
$$= \frac{1}{2m(m-1)}\sum_{\substack{i=1 \\ i\ne j}}^{m}\sum_{j=1}^{m}(\hat{r}_{i,k} - \hat{r}_{j,k})^2$$

$$S_k^2 = \frac{1}{N(N-1)}\sum_{\substack{i=1 \\ i<j}}^{N}\sum_{j=1}^{N}(r_{i,k} - r_{j,k})^2$$
$$= \frac{1}{2N(N-1)}\sum_{\substack{i=1 \\ i\ne j}}^{N}\sum_{j=1}^{N}(r_{i,k} - r_{j,k})^2$$

For each pair $(\hat{r}_{i,k}, \hat{r}_{j,k})$ $(i \ne j, 1 \le i, j \le N)$, the probability that they are both being reconstructed due to the corresponding nodes (i, j) being sampled, is $m(m-1)/(N(N-1))$. Then, with Lemma 1, we have

$$E(\hat{s}_k^2) = \frac{1}{2m(m-1)}E(\sum_{\substack{i=1 \\ i\ne j}}^{m}\sum_{j=1}^{m}(\hat{r}_{i,k} - \hat{r}_{j,k})^2)$$
$$= \frac{1}{2m(m-1)}\sum_{\substack{i=1 \\ i\ne j}}^{N}\sum_{j=1}^{N}E((\hat{r}_{i,k} - \hat{r}_{j,k})^2)\frac{m(m-1)}{N(N-1)}$$
$$= \frac{1}{2N(N-1)}\sum_{\substack{i=1 \\ i\ne j}}^{N}\sum_{j=1}^{N}E((\hat{r}_{i,k} - \hat{r}_{j,k})^2)$$
$$= \frac{1}{2N(N-1)}\sum_{\substack{i=1 \\ i\ne j}}^{N}\sum_{j=1}^{N}((r_{i,k} - r_{j,k})^2 - E(\hat{\varepsilon}_{i,k}^2) - E(\hat{\varepsilon}_{j,k}^2))$$
$$= S_k^2 - \frac{1}{N}\sum_{i=1}^{N}E(\hat{\varepsilon}_{i,k}^2) = S_k^2 - \frac{1-p_{k'k'}}{N}\sum_{i=1}^{N}\sigma_i^2$$

By replacing S_k^2 by s_k^2, $E(\hat{s}_k^2)$ by $\hat{s}_k^2$, and σ_i^2 by $\hat{\sigma}_i^2$ in Formula (13), we can estimate s_k^2 by $\hat{s}_k^2 + \frac{1-p_{k'k'}}{N}\sum_{i=1}^{N}\hat{\sigma}_i^2$. However, $\sum_{i=1}^{N}\hat{\sigma}_i^2$ can not be obtained since sampling all nodes is prohibitive in our approach. Thus, we use an upper bound of s_k^2, denoted by $(s_k^*)^2$, and estimate it by $\hat{s}_k^2 + (1-p_{k'k'})\sigma_T^2$ due to $\hat{\sigma}_i^2 \le \sigma_T^2$.

Theorem 2.

$$\Pr\left(|\hat{A}_k^a - A_k^a| \le \varphi_{\frac{\alpha_r}{2}} s_k^* \sqrt{\frac{1-f}{m}} + \frac{1}{m}\varphi_{\frac{\alpha_z}{2}}\sqrt{(1-p_{k'k'})\sum_{i=1}^{m}\hat{\sigma}_i^2}\right) \quad (13)$$
$$\ge (1-\alpha_r)(1-\alpha_z)$$

where $f = \frac{m}{N}$, $s_k^* = \sqrt{\hat{s}_k^2 + (1-p_{k'k'})\sigma_T^2}$, $\varphi_{\frac{\alpha_r}{2}}$ *and* $\varphi_{\frac{\alpha_z}{2}}$ *are respectively the upper* $\alpha_r/2$, $\alpha_z/2$ *point on the standard normal distribution.*

Proof. Define the events A, B and C respectively as

$$A: \ |\hat{A}_k^a - A_k^a| \le \varphi_{\frac{\alpha_r}{2}} s_k^* \sqrt{\frac{1-f}{m}} + \frac{1}{m}\varphi_{\frac{\alpha_z}{2}} \sqrt{\sum_{i=1}^{m} \hat{\sigma}_i^2}$$

$$B: \ |R_k - A_k^a| \le \varphi_{\frac{\alpha_r}{2}} s_k^* \sqrt{\frac{1-f}{m}}$$

$$C: \ |Z_k| \le \frac{1}{m}\varphi_{\frac{\alpha_z}{2}} \sqrt{(1-p_{k'k'})\sum_{i=1}^{m} \hat{\sigma}_i^2}$$

Because $s_k^{*2} \ge s_k^2$, by Formula (12) we have

$$\Pr(B) \ge \Pr(|R_k - A_k^a| \le \varphi_{\frac{\alpha_r}{2}} s_k \sqrt{\tfrac{1-f}{m}}) = 1-\alpha_r$$

Since $Z_k \sim N(0, \frac{1-p_{k'k'}}{m^2}\sum_{i=1}^{m}\hat{\sigma}_i^2)$,

$$\Pr\left(|Z_k| \le \frac{1}{m}\varphi_{\frac{\alpha_z}{2}} \sqrt{(1-p_{k'k'})\sum_{i=1}^{m} \hat{\sigma}_i^2}\right) = 1-\alpha_z$$

By Formula (9), we have $|\hat{A}_k^a - A_k^a| \le |R_k - A_k^a| + |Z_k|$. When inequalities B and C are satisfied, A must hold. Because sampling and modeling errors are independent random variables, so B and C are independent events. Then, we have

$$\Pr(A) \ge \Pr(BC) = \Pr(B)\Pr(C) \ge (1-\alpha_r)(1-\alpha_z)$$

Let $\varepsilon_k = \varphi_{\frac{\alpha_r}{2}} s_k^* \sqrt{\frac{1-f}{m}} + \frac{1}{m}\varphi_{\frac{\alpha_z}{2}} \sqrt{(1-p_{k'k'})\sum_{i=1}^{m}\hat{\sigma}_i^2}$.

Since $\hat{A}_k^s = N\hat{A}_k^a = NR_k - NZ_k$, we can easily derive the following results from the above analysis of average:

$$\Pr\left(|\hat{A}_k^s - A_k^s| \le N\varepsilon_k\right) \ge (1-\alpha_r)(1-\alpha_z) \qquad (14)$$

Here Formulas (13) and (14) give the approximation error ε_k ($N\varepsilon_k$) of Average (Sum) aggregation with the probability guarantee $(1-\alpha_r)(1-\alpha_z)$.

3. Parameter Determination

From Formulas (13) and (14) we can see that with given the probability guarantee, *i.e.*, α_r and α_z, the approximation error depends on the error constraint σ_T^2 and the sample size m. In this section we discuss the selection of their values with the desired error bound for ε_k by users, denoted by ε_T.

3.1. Error Constraint σ_T^2

As shown in Formula (3), $\hat{\sigma}_i$ indicates the average error for the data reconstructed in a time window. Thus, σ_T specifies the maximum degree of the average error that the user can tolerate for the reconstructed data. A larger σ_T would allow larger errors in the reconstructed data and may enlarge the approximation error. On the other hand, a larger σ_T gives the sampled nodes more chances to transmit their model parameters instead of their original data and further reduce the communication cost. Thus, the trade-off exists between communication cost and approximation error.

Here we provide one possible solution to determine σ_T^2. During the first time window of aggregation, all sampled nodes transmit their original data to the BS. The BS fits the specified model to these data and computes the modeling errors $\{\hat{\sigma}_i^2 \mid 1 \le i \le m\}$ for all sampled nodes. A histogram is computed to count the number of error values falling into each bin, which reflects the quality of data modeling for the sensor network. According to this frequency distribution, the user can select a value of σ_T^2 as large as possible while ensuring an acceptable approximation error. Finally, the BS broadcasts σ_T^2 to the sensor network and each sensor node works on the new error variance constraint. This procedure could be conducted reactively when substantial sampled nodes start to continuously transmit their original data, which indicates the changes of the nature of data in the sensor network.

In our experiments on real data set, we show linear regression well characterizes the sensor data and incur few original data transmissions even with a small error variance constrain.

3.2. Sampling Ratio ϱ

From Formulas (13) and (14), a larger sample size m enables a smaller approximation error.

It is easily shown that we can relax ε_k to

$$\varepsilon_k = \varphi_{\frac{\alpha_r}{2}} s_k^* \sqrt{\tfrac{1-f}{m}} + \tfrac{1}{\sqrt{m}}\varphi_{\frac{\alpha_z}{2}}\sigma_T$$

without changing the inequality relationship with the probability guarantee $(1-\alpha_r)(1-\alpha_z)$ in Formulas (13) and (14). We consider the least sample size to satisfy $\varepsilon_k \le \varepsilon_T$ for any k in $[t+1, t+l]$. With an approximation of $f = m/N \to 0$ (for relative small sample size and large population), we have

$$\tfrac{1}{\sqrt{m}}\varphi_{\frac{\alpha_r}{2}} s_k^* + \tfrac{1}{\sqrt{m}}\varphi_{\frac{\alpha_z}{2}}\sigma_T \le \varepsilon_T$$

which should hold for any k in $[t+1, t+l]$. Then, we can obtain the least sample size m_k required by epoch k in the time window $[t+1, t+l]$ to ensure ε_k is less than a threshold ε_T

$$m_k = \left(\frac{\varphi_{\frac{\alpha_r}{2}} s_k^* + \varphi_{\frac{\alpha_z}{2}}\sigma_T}{\varepsilon_T}\right)^2, \quad t+1 \le k \le t+l \qquad (15)$$

For each epoch k in $[t+1, t+l]$, if the BS finds $m < m_k$, it can issue another sampling request to obtain $m_k - m$ samples. However, s_k^* cannot be obtained before sampling, we give the following estimation if the

upper bound r_{max} and lower bond r_{min} of sensor readings are known

$$s_k^* \approx \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(\frac{r_{\max}-r_{\min}}{2})^2} \quad (16)$$

We can obtain an estimation of the required sample size, denoted by m_r, for all epochs in the time window $[t+1, t+l]$ by inserting Formula (15) into Formula (16). The sampling ratio ϱ is set to be not less than m_r/N.

3.3. Time Window Size *l*

When all sampled nodes transmit their original data, the approximation error includes only the sampling estimation error and no modeling estimation error. Thus, the aggregation computation with original data needs a less sample size than with the compressed data by modeling to achieve the same approximation error. Let m_o be the sample size needed to obtain (ε_T, δ) -approximation aggregation by collecting the original data, then $\varphi_{\frac{\delta}{2}} s_k \sqrt{\frac{1-f_o}{m_o}} = \varepsilon_T$ where $f_o = \frac{m_o}{N}$. With the approximation of $f_o = m_o / N \to 0$,

$$m_o = (\varphi_{\frac{\delta}{2}} s_k)^2 / \varepsilon_T^2$$

On the other hand, we have

$$\delta = 1-(1-\alpha_r)(1-\alpha_z) \Rightarrow \delta > \alpha_r \Rightarrow \varphi_{\frac{\delta}{2}} < \varphi_{\frac{\alpha_r}{2}}$$

As above, we also have $\varphi_{\frac{\delta}{2}} < \varphi_{\frac{\alpha_z}{2}}$.

According to the above discussion on sampling ratio, we have

$$\frac{m_t}{m_o} = \frac{(\varphi_{\frac{\alpha_r}{2}} s_k^* + \varphi_{\frac{\alpha_z}{2}} \sigma_T)^2 / \varepsilon_T^2}{(\varphi_{\frac{\delta}{2}} s_k)^2 / \varepsilon_T^2} = \frac{(\varphi_{\frac{\alpha_r}{2}} s_k^* + \varphi_{\frac{\alpha_z}{2}} \sigma_T)^2}{(\varphi_{\frac{\delta}{2}} s_k)^2}$$

$$\geq \frac{(\varphi_{\frac{\delta}{2}} s_k^* + \varphi_{\frac{\delta}{2}} \sigma_T)^2}{(\varphi_{\frac{\delta}{2}} s_k)^2} \geq \frac{(s_k^* + \sigma_T)^2}{s_k^2} \quad (17)$$

$$\geq \frac{(s_k + \sigma_T)^2}{s_k^2} = (1+\frac{\sigma_T}{s_k})^2$$

Without data modeling compression, the aggregation requires $m_o(l+1)$ original data transmissions for a time window to achieve the approximation error ε_T. With the data modeling compression, our scheme requires $m(p+2)$ data transmissions to achieve the approximation error ε_T. To achieve energy savings, we should have $m_o(l+1)/m_t(p+2) > 1$, then

$$l+1 > (p+2)\frac{m_t}{m_o} \geq (p+2)(1+\frac{\sigma_T}{s_k})^2$$

In the case of $\sigma_T < S_k$, $1+\frac{\sigma_T}{s_k} < 2$, we could set $l+1 > 4(p+2)$.

4. Simulation Evaluation

To measure the performance of our secure aggregation scheme, we simulate a sensor network based on the data from a real world deployment with 54 sensor nodes (ID from 1 -54) in the Intel Research lab, which includes a trace of sensor readings collected between February and April,2004 , node location and network connectivity information. The sensors collected time-stamped humidity, temperature and voltage values in 31 second intervals. We use the first 2000 epochs of the data set in the day 03/08 with the largest size among all days and assume a continuous aggregation query on the temperature attribute during this period. The periodic aggregation is conducted on it with a time window size $l = 40$, *i.e.*, 50 time windows. In the linear regression model, we let $p = 3$.

To show the performance of linear regression model for describing sensor data, we investigate the distribution of error variance and its impact on data transmission for all sensor nodes.

Figure 1 shows temperature readings (in degrees Celsius) of 52 sensor nodes in 2000 successive epochs, which are used for our simulation. **Figure 2** shows the error variance of linear regression model in every time window for all sensor nodes. For all time windows, all the sensor nodes have error variances less than 0.2. **Figure 3** shows under different choice of error constraint σ_T^2, the number of sensor nodes which has a larger modeling error variance than σ_T^2 in each time window. We can conclude most of sensor nodes at most of time windows are consistent with variance constraint. When $\sigma_T^2 > 0.07$, less than 10% of sensor nodes exceed σ_T^2; when $\sigma_T^2 > 0.1$, the number decreases to 2% .Our experiment indicates only a small portion of sampled node will transmit their original data.

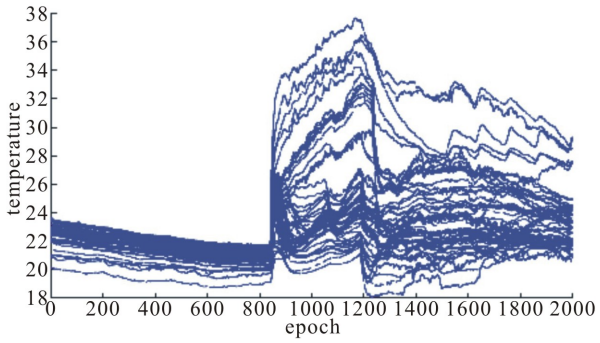

Figure 1. Temperature readings (in degrees Celsius) of 52 sensor nodes in 2000 successive epochs (excluding two nodes with incomplete data and one node with abnormal data).

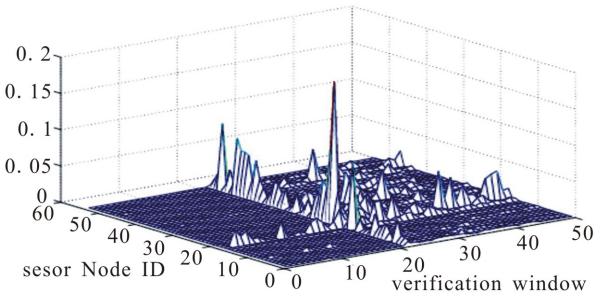

Figure 2. The error variances of linear regression model in all sensor nodes for each time window.

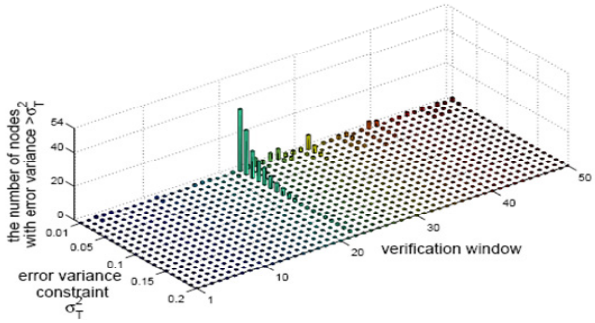

Figure 3. the number of sensor nodes with error variance $> \sigma_T^2$ in each time window.

Assuming $\varrho = 0.4$ and $\sigma_T^2 = 0.2$, **Figure 4** shows the difference between the average aggregation result estimated by sampling with time window based compression and the aggregation result estimated by sampling with original data transmission in every epoch. As we can see, the difference is in [-0.2, 0.25]. It indicates that our approach with data compression can obtain the estimation of average aggregations close to those obtained by the approach without data compression, even when the sample size is the same. It also indicates our approach achieves the energy efficiency while obtaining the approximate estimations, since in each time window only five numbers are sent from each sampled node.

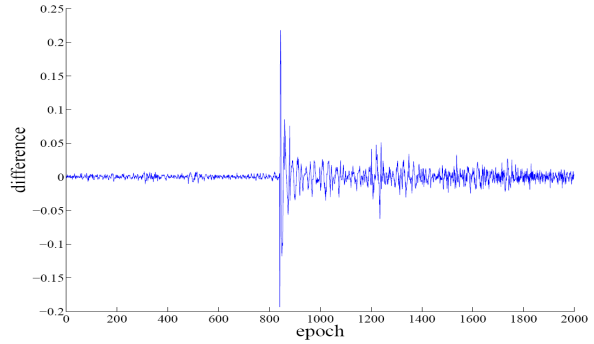

Figure 4. The difference between two average aggregation results respectively estimated by the approaches with and without data compression in every epoch.

5. Conclusions

In this paper we propose a sampling-based approach with time window based linear regression for approximate continuous aggregation. The approximation error of the aggregation results is analyzed. The determination of parameters in our approach is also discussed. By simulation results on real data set we verify the effectiveness of our approach.

5. References

[1] S. Madden, M. J. Franklin, J. M. Hellerstein and W. Hong, "The Design of an Acquisitional Query Processor for Sensor Networks," *Proceedings of International Conference on Management on Data*, California, 2003, pp. 491-502.

[2] S. Madden, M. J. Franklin, J. M. Hellerstein and W. Hong, "TAG: A Tiny Aggregation Service for Ad Hoc Sensor Networks," *Proceedings of the* 5*th Symposium on Operating Systems Design and Implementation*, New York, 2002, pp. 131-146.

[3] J. Considine, F. Li, G. Kollios and J. Byers, "Approximate Aggregation Techniques for Sensor Databases," *International Conference on Data Engineering*, Boston, 2004, pp. 449-460.

[4] S. Nath, P. B. Gibbons, S. Seshan and Z. R. Anderson, "Synopsis Diffusion for Robust Aggregation in Sensor Networks," *Sensys*, 2004, pp. 250-262.

[5] G. Cormode, M. N. Garofalakis, S. Muthukrishnan and R. Rastogi, "Holistic Aggregates in a Networked World: Distributed Tracking of Approximate Quantiles," *Proceedings of International Conference on Management on Data*, 2005, pp. 25-36.

[6] A. Deligiannakis, Y. Kotidis and N. Rossopoulos, "Processing Approximate Aggregation Queries in Wireless Senor Networks," *Information Systems*, Vol. 31, No. 8, 2006, pp. 770-792.

[7] A. Manjhi, S. Nath and P. B. Gibbons, "Tributaries and Deltas: Efficient and Robust Aggregation in Sensor Network Streams," *ACM SIGMOD*, ACM Press, 2005, pp. 287-298.

[8] S. Y. Cheng, J. Z. LI, Q. Q. Ren and L. Yu, "Bernoulli Sampling Based (epsilon, delta)-Approximate Aggregation in Larger-Scale Sensor Networks," *IEEE International Conference on Computer Communications*, California, 2010, pp. 1181-1189.

[9] S. Lin, B. Arai, D. Gunopulos and G. Das, "Region Sampling: Continuous Adaptive Sampling on Sensor Networks," *IEEE International Conference on Data Engineering*, Cancun, 2008, pp. 794-803.

[10] B. Bash, J. Byers and J. Considine, "Approximately Uni-

form Random Sampling in Sensor Networks," *Proceedings of 1st Workshop on Data Management in Sensor Networks*, August, 2004.

[11] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin and S. Madden, "Distributed Regression: An Efficient Framework for Modeling Sensor Network Data," *ACM/IEEE IPSN*, 2004, pp. 1-10.

[12] W. Xue, Q. Luo, L. Chen and Y. Liu, "Contour Map Matching for Event Detection in Sensor Networks," *SIGMOD*, New York, 2006, pp. 145-156.

[13] H. Gupta, V. Navda, S. R. Das and V. Chowdhary, "Efficient Gathering of Correlated Data in Sensor Networks," *MobiHoc*, New York, 2005, pp. 402-413.

[14] A. Sen and M. Srivastava, "Regression Analysis: Theory, Methods, and Applications," Springer-Verlag, New York, 1990.

[15] Z. Govindarajulu, "Elements of Sampling Theory and Methods," Prentice Hall, New Jersey, 1999.

