Sampling-Based Approximate Skyline Query in Sensor Equipped IoT Networks

Ji Li, Akshita Maradapu Vera Venkata Sai, Xiuzhen Cheng, Wei Cheng, Zhi Tian, and Yingshu Li*

Abstract: The ever increasing requirements of data sensing applications result in the usage of IoT networks. These networks are often used for efficient data transfer. Wireless sensors are incorporated in the IoT networks to reduce the deployment and maintenance costs. Designing an energy efficient data aggregation method for sensor equipped IoT to process skyline query, is one of the most critical problems. In this paper, we propose two approximation algorithms to process the skyline query in wireless sensor networks. These two algorithms are uniform sampling-based approximate skyline query and Bernoulli sampling-based approximate skyline query. Solid theoretical proofs are provided to confirm that the proposed algorithms can yield the required query results. Experiments conducted on actual datasets show that the two proposed algorithms have high performance in terms of energy consumption compared to the simple distributed algorithm.

Key words: data aggregation; sampling; IoT networks

1 Introduction

An increase in the urban population has made everyday sustainability an uphill problem leading to new challenges like, constant power supply, public safety, disaster prediction, and traffic maintenance. Deploying smart cities has become inevitable to address these challenges. Many cities have adopted various technologies, and are working towards smart city development[1–3]. Although these applications are designed for urban living, they should also be incorporated in rural areas, so that more resources could be preserved. Smart city networks should collect sensory data from all over the city in order to provide better information. Therefore, to collect such well spread data and interpret information at the city level, they exploit various sensor-equipped devices in the city[4,5].

Nowadays, all the devices from smart home systems to intelligent transport systems are well connected by the means of internet, and such a network is called Internet-Of-Things (IoT)[6–8]. With an increase in the network’s scale, our need for intelligent devices emerges along with the need to sense various activities for the convenient living of city dwellers. The main objective of IoT is to reduce cost and provide faster access to the sensory data[9–13]. However, the primary challenge is that the deployment of IoT network is expensive due to its requirement of large number of sensing devices[14]. Moreover, it is also critical to collect data autonomously and provide intelligent methods that address the issues of dynamic traffic, accommodating new services, channel conditions, and ever-increasing user requirements.

Sensors are the building blocks for many IoT devices. Using sensors as a communication media helps us resolve many issues, like the ones discussed above.
IoT devices do not need external infrastructure for communication. In previous works, researchers have studied the issues of routing, topology control, and time synchronization in networks that include sensors for communication\cite{15–18}.

Although using sensors reduces the communication cost, it causes the problem of processing cost. Sensors collect data for an extended period of time, thus there is a massive amount of data to be processed at a single node for information retrieval. Therefore increasing the processing cost to solve this problem, we need data aggregation at the sensor level. When sensory data is aggregated, just the aggregated partial data can be sent into the network instead. However, this raises the energy consumption issue as the aggregation costs more energy and the sensors are not equipped with huge amounts of power supply. According to Ref. [19], cost of transmitting one bit of sensory data using wireless link is equivalent to the cost of executing 1000 instructions. Therefore, reducing data transmission is one of the mainstream methods to decrease the energy consumption in IoTs. Hence, it is critical to design energy efficient data aggregation algorithms for sensor-equipped IoT networks. A method of aggregation named skyline queries filters results to keep only those data that are no worse than other data and are essential for a given network. For example, in smart cities applications, skyline query can be used to find hotels which are cheap and close to the beach or close to a wireless link is equivalent to the cost of executing 1000 instructions. In the field of environment monitoring, skyline query results can be used to find potential pollution sources\cite{21–22}. Therefore, an energy efficient data aggregation model should accommodate this query in its development. In practice, exact query results are not always necessary. On the other hand, approximate query results may be acceptable for energy conservation\cite{24, 25}. Therefore, in this paper, we propose two algorithms to process approximate skyline queries in IoTs. These algorithms are based on uniform sampling and Bernoulli sampling, respectively. The proposed algorithms can return the δ-approximate skyline query results. In summary, the main contributions of this paper can be summarized as follows.

1. Mathematical estimators for the skyline aggregation operations are provided.
2. Mathematical methods to determine the required sample size and sample probability for calculating δ-approximate skyline are designed.
3. Distributed algorithms for approximate skyline query are provided. Additionally, the energy costs of these two algorithms are analyzed.

4. Simulation results are presented which indicate that the proposed algorithms have high performance in terms of energy consumption.

The rest of the paper is organized as follows. Section 2 defines the approximate skyline query problem. Section 3 provides the mathematical proof for the δ-approximate skyline query algorithms. Section 4 explains the proposed δ-approximate skyline query algorithms. Section 5 shows the simulation results. The related works are discussed in Section 6. Section 7 concludes the whole paper.

2 Problem Definition

Assume that we have a sensor-equipped IoT network with \( n \) sensor nodes, \( s_{t1} \) is the sensory value of node \( i \) at time \( t \), and \( S_t = \{s_{t1}, s_{t2}, \ldots, s_{tn}\} \) is the set of all the sensory data in the network at time \( t \). Moreover, each sensory data \( s_i \) has \( d \) dimensions, \( s_i, D_1, s_i, D_2, \ldots, s_i, D_d \). In this paper, we assume that the data is distributed randomly in the network, and the spatial and temporal correlation for the sensory data is ignored. Moreover, we also assume that the different dimensions of the sensory data are independent of each other.

In this paper, we focus on the skyline query operations on \( S_t \), whose definition is as follows.

**Definition 1** (cominate) For two given sensory data readings \( s_{t1} \) and \( s_{t2} \) with \( d \) dimensions, we define \( s_{t1} \) dominates \( s_{t2} \), which is written as \( s_{t1} \rightarrow s_{t2} \) as follows:

\[
\begin{align*}
& s_{t1}, D_i \geq s_{t2}, D_i, \quad \forall 1 \leq i \leq d, \\
& \exists 1 \leq i \leq d \quad s_{t1}, D_i > s_{t2}, D_i.
\end{align*}
\]

**Definition 2** (skyline) For a given sensory data set \( S_t \), the skyline of \( S_t \) is defined as

\[
\text{Skyline}(S_t) = \{s \in S_t | \forall v \in S_t, v \rightarrow s\}.
\]

An example of the skyline query in a 2-dimension data set with 14 readings is shown in Fig. 1. The black dots are used to denote the skyline set. We can see that no sensory data in the data set can dominate the black nodes.

A naive method that solves the skyline query problems has four main steps:

1. All the nodes in the network are organized into an aggregation tree.
2. The sink node broadcasts the skyline query operation in the network.
3. All the nodes in the network submits its sensory data along the aggregation tree.
(4) During the data transmission, the intermediate nodes in the aggregation tree aggregates the partial results. However, the above method will lead to a huge communication cost and computation cost for calculating exact aggregation result. Therefore, we propose the definition of δ-approximate result for the skyline query operation.

**Definition 3** (δ-approximate skyline) For a given sensory data set $S_t$, let $\text{Max}(S_t, D_i)$ denote the maximum value of the sensory data in the $i$-th dimension, then $\text{Skyline}(S_t)$ is called the δ-approximate skyline of the sensory data set $S_t$ if

$$\Pr(\text{Max}(S_t, D_i) = \text{Max}(\text{Skyline}(S_t), D_i)) \geq 1 - \delta,$$

where $\forall 1 \leq i \leq d$.

According to Definition 3, we can see that for a δ-approximate skyline, in each dimension, we need to make sure that the actual maximum value exists with a probability of at least $1 - \delta$. Moreover, in order to calculate the required sample size and sample probability to calculate the δ-approximate skyline, we need to make use of the definition of δ-estimator proposed in Ref. [26]. Let $I_t$ and $\tilde{I}_t$ be the exact aggregation result and approximate aggregation result of $S_t$ at time $t$, respectively. The definition of the δ-estimator is as follows.

**Definition 4** (δ-estimator) For any $\delta$ $(0 < \delta \leq 1)$, $\tilde{I}_t$ is called the δ-estimator of $I_t$ if $Pr(\tilde{I}_t \neq I_t) \leq \delta$. According to Definition 3, the problem of computing δ-approximate skyline is defined as follows.

**Input:** (1) A sensor-equipped IoT network with $n$ nodes; (2) the sensory data set $S_t$; and (3) $\delta$ $(0 \leq \delta \leq 1)$.

**Output:** δ-approximate skyline query result.

# 3 Preliminary

In this paper, we use two sampling techniques to sample the sensory data, which are uniform sampling and Bernoulli sampling. The preliminaries of calculating the approximate skyline, required sample size, and required sample probability are presented in the following subsections.

## 3.1 Uniform Sampling based Approximate Skyline (USAS) query

Let $u_1, u_2, \ldots, u_m$ denote $m$ simple random samplings with replacement from the sensory data set $S_t$, and $\text{Max}(S)$ denote the maximum value of sensory data set $S$. Let $U(m) = \{u_1, u_2, \ldots, u_m\}$ denote a uniform sample of $S_t$ with sample size $m$, then we have the following theorem which is proposed in Ref. [26].

**Theorem 1** Let $n_{\text{Max}S_t}$ denote the number of appearances of the maximum value, then we have

$$\Pr(\text{Max}(U(m)) \neq \text{Max}(S_t)) = \left(1 - \frac{n_{\text{Max}S_t}}{n}\right)^m.$$

Let $\text{Skyline}(S_{t})_{\text{USAS}}$ denote the uniform sampling-based estimator of exact result $\text{Skyline}(S_{t})$. Then $\text{Skyline}(S_{t})_{\text{USAS}}$ is defined as

$$\text{Dis}(S_{t})_{\text{USAS}} = \text{Skyline}(U(m)).$$

Based on Theorem 1, we have the following theorem.

**Theorem 2** $\text{Skyline}(S_{t})_{\text{USAS}}$ is a δ-approximate skyline of $S_t$ if

$$m \geq \frac{\ln(1 - d\sqrt{1 - \delta})}{\ln(1 - n_{\text{min}}/n)},$$

where $n_{\text{min}}$ is the number of appearances of the least appearing data.

**Proof** Based on the condition, we have

$$m \ln(1 - n_{\text{min}}/n) \leq \ln(1 - d\sqrt{1 - \delta}),$$

$$(1 - n_{\text{min}}/n)^m \leq 1 - d\sqrt{1 - \delta},$$

$$1 - (1 - n_{\text{min}}/n)^m \geq d\sqrt{1 - \delta},$$

$$(1 - (1 - n_{\text{min}}/n)^m)^d \geq 1 - \delta.$$

According to Theorem 1, we can see that for each dimension, the probability for its maximum value to appear in the final result is at least $1 - (1 - n_{\text{min}}/n)^m$. Now the condition $(1 - (1 - n_{\text{min}}/n)^m)^d \geq 1 - \delta$ is satisfied, which in turn satisfies Definition 3. Then this theorem is proved.

## 3.2 Bernoulli Sampling Approximate Skyline (BSAS) query

Let $B(q) = \{b_1, b_2, \ldots, b_{B(q)}\}$ denote a Bernoulli sample of sensory data set $S_t$ with sample probability $q$. Using the similar strategy in Ref. [26], we have the following theorem.
Theorem 3  Let \( n_{\text{Max}} \) denote the number of appearances of the maximum value, then we have
\[ \Pr(\text{Max}(B(q)) \neq \text{Max}(S_t)) = (1 - q)^{n_{\text{Max}}}. \]

Let \( \text{Skyline}(S_t)_b \) denote the Bernoulli sampling-based estimator of exact result \( \text{Skyline}(S_t) \). Then \( \text{Skyline}(S_t)_b \) is defined as
\[ \text{Skyline}(S_t)_b = \text{Skyline}(B(q)). \]

Based on Theorem 3, we have the following theorem.

Theorem 4  \( \text{Skyline}(S_t)_b \) is a \( \delta \)-approximate skyline of \( S_t \) if
\[ q \geq 1 - n_{\text{min}} \sqrt{1 - \sqrt{1 - \delta}}, \]
where \( n_{\text{min}} \) is the number of appearances of the least appearing data.

Proof  According to the condition, we have
\[
1 - q \leq n_{\text{min}} \sqrt{1 - \sqrt{1 - \delta}}, \\
(1 - q)^{n_{\text{min}}} \leq 1 - \sqrt{1 - \delta}, \\
1 - (1 - q)^{n_{\text{min}}} \geq \sqrt{1 - \delta}, \\
(1 - (1 - q)^{n_{\text{min}}})^d \geq 1 - \delta.
\]

According to Theorem 3, we can see that for each dimension, the probability of its maximum value appearing in the final result is at least \( 1 - (1 - q)^{n_{\text{min}}} \). Now the condition \( (1 - (1 - q)^{n_{\text{min}}})^d \geq 1 - \delta \) is satisfied, which in turn satisfies Definition 3. Then this theorem is proved.

4  \( \delta \)-Approximate Skyline Query Algorithms

The theorems mentioned in Section 3 are about calculating the required sampling size and sampling probability according to a given \( \delta \). However, we still have the following critical problems:

(1) How does the sink node broadcast the sampling information in the network.
(2) How to sample the sensory data.
(3) How to aggregate and transmit the partial aggregation results.

The method to solve the above mentioned problems will be introduced in the following two subsections.

4.1 USAS query algorithm

When the sample size \( m \) is calculated using the theorems in Section 3.1, the sink node needs to broadcast the sample information to each node in the network. Obviously, there is a simple method to sample the sensory data, and the basic process is introduced as follows:

1. The sink generates \( m \) random numbers in \( \{1, 2, \ldots, n\} \) and broadcasts these random numbers.
2. For each sensor node, if its ID belongs to the \( m \) numbers, the node will send its sensory data to the sink node.

However, in the above algorithm, the first step has an energy cost, due to the transmission of a large amount of sampling information. In order to further reduce the energy cost, we divide the whole network into \( k \) disjoint clusters \( C_1, C_2, \ldots, C_k \). All the cluster-heads in the network are organized as a minimum hop-count spanning tree rooted at the sink node. An example of the network topology is shown in Fig. 2. We can see that all the cluster-heads are organized as a minimum hop-count spanning tree. Since the communication range of a sensor node may be limited, some ordinary nodes may also be included in the spanning tree to ensure network connectivity. We adopt the uniform sampling algorithm proposed by Ref. [28], which is described as follows:

1. The sink generates random numbers \( Y_i \) satisfying \( \Pr(0 = l) = |C_l|/n (1 \leq i \leq m) \).
2. Let \( m_l \) be the sample size of \( C_l \). Calculate \( m_l \) by \( m_l = \lfloor |Y_l|/n \rfloor \).
3. The sink node sends the sample size \( \{m_l | 1 \leq l \leq k\} \) to each cluster head.
4. Then each cluster-head samples the sensory data in the cluster using the above naive sampling algorithm.

When the cluster head of the \( l \)-th cluster receives the sensory data \( U(m_l) \), it calculates the partial aggregation result \( \text{Skyline}(U(m_l)) \). The partial aggregation result \( \text{Skyline}(U(m_l)) \) is transmitted along the spanning tree to the sink node. In order to further reduce the transmission cost, the intermediate nodes aggregate the received partial results while transmitting the data. The above process is explained in Algorithm 1.

According to the analysis in Section 3.1, we have
\[
m = \begin{bmatrix} \ln(1 - \sqrt{1 - \delta}) \\ \ln(1 - n_{\text{min}}/n) \end{bmatrix}.
\]

Fig. 2  Network topology example.
Algorithm 1  USAS query algorithm

Input: $\delta$

Output: $\delta$-approximate skyline result

1: $m = \left\lceil \frac{\ln(1 - \sqrt{1 - \delta})}{\ln(1 - \frac{n_{\min}}{n})} \right\rceil$
2: generate $Y_i$ following $Pr(Y_i = l) = \frac{C_l}{\sum_{l=1}^{k} C_l}$
3: $m_j = |\{Y_i | Y_i = l\}|$ (1 \(\leq i \leq m, 1 \leq l \leq k\)), the sink sends $m_j$ to each cluster head by multi-hop communication
4: for each cluster-head of the clusters $C_i$ (1 \(\leq l \leq k\)) do
5: generate random numbers $k_1, k_2, \ldots, k_{m_l}$ then broadcast inside the cluster
6: end for
7: for each cluster member of $C_i$ (1 \(\leq l \leq k\)) do
8: send sensory value to cluster head if its ID $\in \{k_1, k_2, \ldots, k_{m_l}\}$
9: end for
10: for each cluster-head of the clusters $C_i$ (1 \(\leq l \leq k\)) do
11: receive sample data $U(m_j)$ and calculate partial result $Skyline(U(m_j))$
12: end for
13: for each node $j$ in the spanning tree do
14: if $j$ is the leaf node then
15: send Skyline, to its parent node
16: else
17: receive partial results $Skyline_{j_1}, Skyline_{j_2}, \ldots, Skyline_{j_c}$ from its children
18: $Skyline_j = Skyline(\bigcup_{i=1}^{c} Skyline_{j_i})$
19: if $j$ is the sink node then
20: return $Skyline_j$
21: else
22: send Skyline, to its parent node
23: end if
24: end if
25: end for

Therefore, we have

$$m = O\left(\frac{1}{\ln(1 - \sqrt{1 - \delta})}\right).$$

In practice, we can assume $|Skyline_j|$ to be a constant. According to Ref. [28], the communication and energy cost of the USAS query algorithm is

$$O\left(\frac{1}{\ln(1 - \sqrt{1 - \delta})}\right).$$

4.2 BSAS query algorithm

Unlike the USAS query algorithm, the sampling information of BSAS query algorithm utilizes only the sampling probability $q$. Moreover, Bernoulli-based method provides a method for each node to do the sampling independently. Therefore, the following method is used in the BSAS query algorithm to perform sampling, and there is no need to divide the network into clusters.

1: Sink node broadcasts the sampling probability.
2: Each node in the network generates a random number $rand \in [0, 1]$ and submits its sensory data to the parent node if $rand < q$.

When the intermediate nodes receive the submitted sensory data, they will calculate the partial aggregation results, and transmit them along the spanning tree. During the process of transmitting partial aggregation results to the sink node, the intermediate nodes aggregate the received partial results. The above process is explained in detail in Algorithm 2. The symbol $j$.data is used to denote the sensory data of node $j$.

According to the analysis in Section 3.2, for the sample probability $q$, we have

$$q = 1 - \frac{n_{\min}}{\sqrt{1 - \delta}}.$$

Similarly, the communication and energy cost of the BSAS query algorithm is $O(n - n_{\min})$.

5 Simulation Result

To evaluate the proposed algorithms, we have simulated a network with 1000 nodes. These nodes are randomly distributed in a rectangular region of size 300 m \(\times\) 300 m and the sink is located in the center of the region. For the USAS algorithm, we use the following strategy to form the clusters:

Algorithm 2  BSAS query algorithm

Input: $\delta$

Output: $\delta$-approximate skyline result

1: $q = 1 - (1 - (1 - \delta)^{n_{\max}}/n^{1/n_{\max}}$
2: sink node broadcasts $q$ in the network
3: for each leaf node $j$ in the spanning tree do
4: if $rand < q$ then
5: send its own sensory data to its parent node
6: end if
7: end for
8: for each non-leaf node $j$ in the spanning tree do
9: receive partial results $Skyline_{j_1}, Skyline_{j_2}, \ldots, Skyline_{j_c}$ from its children
10: $Skyline_j = Skyline(\bigcup_{i=1}^{c} Skyline_{j_i})$
11: if $rand < q$ then
12: $Skyline_j = Skyline(Skyline_j \cup \{j$.data$\})$
13: end if
14: if $j$ is the sink node then
15: return $Skyline_j$
16: else
17: Send Skyline, to its parent node
18: end if
19: end for
(1) Divide the whole region into $10 \times 10$ grids.
(2) All the nodes in the same grid are grouped into one cluster.
(3) For each cluster, randomly select a node among all the nodes in the same grid as the cluster-head.

According to Ref. [29], for each node, the energy cost to send and receive one byte is set as 0.0144 mJ and 0.0057 mJ, respectively. According to the experiment results in Ref. [30], the communication radius of each sensor node is set to be $30\sqrt{2}$ m, which allows every sensor node to communicate with its cluster-head by a one-hop message.

The sensory data set is generated based on an actual data set named AADF Data Traffic Counts Metadata[31]. The default parameters of the simulation are summarized in Table 1.

### 5.1 USAS query algorithm

The first group of simulations is about the relationship between $\delta$ and the required sample size. The results are presented in Fig. 3. Two groups of results with different values of $n/n_{min}$ are listed. These results indicate that the sample size increases with the decline of $\delta$. Moreover, we can see that the sample sizes are much smaller than the size of the network. For example, when $\delta = 0.01$, the sample size is about 105 for deriving $\delta$-approximate skyline if $n/n_{min} = 15$, which indicates that we just need to sample about 10% sensory data from the network to calculate an approximate skyline. Therefore, our USAS query algorithm saves a huge amount of energy since it only needs a little amount of sensory data to be sampled and transmitted.

The second group of simulations is about the relationship between $\delta$ and the energy cost. The results are shown in Fig. 4. These results show that the energy cost increases as $\delta$ decreases, since more data needs to be sampled.

The third group of simulations is to compare the energy cost between the USAS query algorithm and the simple distributed algorithm. The basic idea of the simple distributed algorithm is as follows:

1. Collect all the raw sensory data.
2. Aggregate the partial results during the transmission.

From Fig. 5, we can see that the simple distributed algorithm can always achieve accurate results. For the USAS query algorithm, we set $\delta = 0.1$, $n/n_{min} = 15$, and the network size changes from 500 to 1500. The results are listed in Fig. 5. We can see that for both the USAS query algorithm and the simple distributed algorithm, the energy cost increases with an increase

| Parameter          | Value       |
|--------------------|-------------|
| Network size $n$   | 1000        |
| Rectangular region size | $300 \times 300$ m |
| Grid size          | $30 \times 30$ m |
| Energy cost to send one byte | 0.0144 mJ |
| Energy cost to receive one byte | 0.0057 mJ |
| $\delta$           | 0.01        |
| $n_{min}$          | 67          |

![Table 1 Default parameters.](image)

**Fig. 3** Relationship between $\delta$ and the sample size.

**Fig. 4** Relationship between $\delta$ and the energy cost for the USAS query algorithm.

**Fig. 5** Energy cost comparison between the USAS query algorithm and the simple distributed algorithm.
in the network size. For the same network size, the USAS query algorithm has a much lower energy cost compared with the simple distributed algorithm. This is because, for the USAS query algorithm, only a small number of nodes need to submit their sensory data. This indicates that the USAS query algorithm has much better performance in energy consumption although it returns approximate skyline query results. Moreover, we can see that with an increase in the network size, the energy cost of the simple distributed algorithm proliferates. On the other hand, the energy cost of the USAS query algorithm remains the same for most of the time. The above phenomenon indicates that the USAS query algorithm has better performance for large scale networks.

The fourth group of simulations is about the relationship between \( d \) and the required sample size, where \( d \) is the number of dimensions of the sensory data. The results are shown in Fig. 6. Two groups of results with different values of \( n/n_{\text{min}} \) are listed for comparison. These results indicate that the sample size increases with the increase of \( d \). Similarly, we can see that the sample sizes are much smaller than the size of the network, which indicates that the proposed USAS query algorithm has high performance in terms of energy cost with different number of sensory data dimensions.

The fifth group of simulations is about the relationship between \( d \) and the energy cost. The results are shown in Fig. 7. Two groups of results with different values of \( n/n_{\text{min}} \) are listed for comparison. These results show that the energy cost increases with an increase in \( d \), as more data need to be sampled. Moreover, for the same value of \( d \), the simulation with less \( n/n_{\text{min}} \) has lower energy cost.

### 5.2 BSAS query algorithm

The first group of simulations is about the relationship between \( \delta \) and the required sample probability. These results are shown in Fig. 8. The results indicate that the sample probability increases as \( \delta \) decreases. Moreover, the required sample probabilities are much smaller than 1. For example, when \( \delta = 0.01 \), the sample probability is about 0.1 when \( n_{\text{min}} = 67 \). Therefore, the proposed BSAS query algorithm saves a huge amount of energy.

The second group of simulations is to show the relationship between \( \delta \) and the energy cost. The results are shown in Fig. 9. We can observe from these results...
that the energy cost increases as $\delta$ decreases, as more data are sampled.

The third group of simulations is to compare the energy cost between the BSAS query algorithm and the simple distributed algorithm. For the BSAS query algorithm, we set $\delta = 0.1$ and $n_{min} = 67$. The network size changes from 500 to 1500 nodes. The results are listed in Fig. 10. We can see that for the same network size, the energy cost of the BSAS query algorithm is much lower than that of the simple distributed algorithm. This indicates that the BSAS query algorithm has high performance in terms of energy consumption. Moreover, we can also see that the BSAS query algorithm has even better performance in large-scale networks.

The fourth group of simulations is to compare the energy cost between the BSAS query algorithm and the USAS query algorithm. In this group of simulation, we set $\delta = 0.1$, $n/n_{min} = 15$, and the communication radius to 60 m. The results are presented in Fig. 11. From these results, we can see that for both the USAS query algorithm and the BSAS query algorithm, the energy cost increases as the network size increases. Moreover, the BSAS query algorithm has lower energy cost when the network size is small, while the USAS query algorithm has lower energy cost when the network size is large. From the above results, we can see that the BSAS query algorithm has the following advantages:

1. The BSAS query algorithm can be used in unclustered networks.

2. The BSAS query algorithm has lower energy cost in small-scale networks.

While on the other hand, the uniform sampling algorithm is good for large-scale clustered networks.

The fifth group of simulations is about the relationship between $d$ and the required sample probability, where $d$ is the number of dimensions for the sensory data. These results are shown in Fig. 12. Two groups of results with different values of $n/n_{min}$ are listed for comparison. The results indicate that the sample probability increases as $d$ increases. Moreover, the required sample probabilities are much smaller than 1. Therefore, the proposed BSAS query algorithm has high performance in terms of energy cost. For the same value of $d$, the simulation with less $n/n_{min}$ has lower required sampling probability.

The sixth group of simulations is about the relationship between $d$ and the energy cost. The results are shown in Fig. 13. Two groups of results with different values of $n/n_{min}$ are listed for comparison. These results show that the energy cost increases as $d$ increases, as more data needs to be sampled. Moreover, for the same value of $d$, the simulation with less $n/n_{min}$ has lower energy cost.

6 Related Work

The sampling technique has been widely used in many fields like, quantile calculation, data collection, and top-k query. For example, Ref. [32] proposed an approximate algorithm to calculate the quantiles in wireless sensor networks. This algorithm reduces energy cost by using
the sampling technique. Reference [33] develops ASAP, an adaptive sampling approach for energy-efficient periodic data collection in sensor networks. Reference [34] uses samples of past sensory data to formulate an optimization problem to approximate top-k queries. However, all the above techniques cannot be used in our problem directly, as the above operations differ a lot from the skyline query.

Skyline query in wireless sensor networks has been widely studied in many existing works, such as Refs. [20] and [35]. In Ref. [20], the author considers the problem of efficient multi-source skyline query processing in road networks. Three different query processing algorithms are proposed and evaluated. Extensive experiments using large actual road network datasets have been conducted, which show that the proposed algorithm has high performance. In Ref. [35], the author proposes a solution that satisfies the three desiderata. The proposed method can control the amount of query forwarding, limit the number of peers involved and the amount of messages transmitted. Experiments on real and synthetic datasets confirm the proposed method has high performance.

Data aggregation in IoT are widely studied in many existing works, such as Refs. [36] and [37]. In Ref. [36], the author proposed a uniform sampling-based approximate aggregation algorithms for four different aggregation operations. Solid theoretical proofs are included, which confirm that the final aggregation results satisfy the required conditions. In Ref. [37], the author proposed Bernoulli sampling-based approximate algorithm to do frequency query in dynamic networks. Simulation results show that the algorithms proposed in both Refs. [36] and [37] have high performance in terms of energy cost.

7 Conclusion

In this paper, two δ-approximate algorithms for the skyline query in sensor equipped IoT networks are proposed. These algorithms are based on the uniform sampling and Bernoulli sampling, respectively. We also proposed mathematical estimators for both algorithms. Moreover, we have derived the values for the required sample size and the required sample probability, that satisfy the specified failure probability requirements of the final result. Finally, a uniform sampling-based approximate skyline query algorithm and a Bernoulli sampling-based algorithm approximate skyline query algorithm are provided.

Experiments are conducted for varying the δ values and network sizes. The performance of the proposed algorithms are then compared to the simple distributed query method. The simulation results indicate that the proposed algorithms have high performance in terms of energy cost.

Acknowledgment

This work was partly supported by the National Science Foundation of USA (Nos. 1741277, 1741287, 1741279, 1851197, and 1741338).

References

[1] A. Cocchia, Smart and digital city: A systematic literature review, in Smart City. New York, NY, USA: Springer, 2014, pp. 13–43.
[2] R. G. Hollands, Will the real smart city please stand up? intelligent, progressive or entrepreneurial? City, vol. 12, no. 3, pp. 303–320, 2008.
[3] Z. Cai and Z. He, Trading private range counting over big IoT data, in Proc. of the 39th IEEE International Conference on Distributed Computing Systems, Dallas, TX, USA, 2019, pp. 144–153.
[4] A. Belhassen and H. Wang, Trajectory big data processing based on frequent activity, Tsinghua Science and Technology, vol. 24, no. 3, pp. 317–332, 2019.
[5] L. Liu, X. Chen, Z. Lu, L. Wang, and X. Wen, Mobile-edge computing framework with data compression for wireless network in energy internet, Tsinghua Science and Technology, vol. 24, no. 3, pp. 271–280, 2019.
[6] Y. Yuan, F. Li, D. Yu, J. Yu, W. Lv, and X. Cheng, Fast fault-tolerant sampling via random walk in dynamic networks, in Proc. of the 39th IEEE International Conference on Distributed Computing Systems, Dallas, TX, USA, 2019, pp. 536–544, 2019.
[7] H. Jin, N. Wang, D. Yu, Q.-S. Hua, X. Shi, and X. Xie, Core maintenance in dynamic graphs: A parallel approach based on matching, IEEE Transactions on Parallel and Distributed Systems, vol. 29, no. 11, pp. 2416–2428, 2018.
Tsinghua Science and Technology, April 2021, 26(2): 219–229

[8] D. Yu, L. Ning, Y. Zou, J. Yu, X. Cheng, and F. C. Lau, Distributed spanner construction with physical interference: Constant stretch and linear sparseness, *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2138–2151, 2017.

[9] Z. Cai and X. Zheng, A private and efficient mechanism for data uploading in smart cyber-physical systems, *IEEE Transactions on Network Science and Engineering*, doi: 10.1109/TNSE.2018.2830307.

[10] S. Cheng, Z. Cai, J. Li, and H. Gao, Extracting kernel dataset from big sensory data in wireless sensor networks, *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 4, pp. 813–827, 2016.

[11] S. Cheng, Z. Cai, J. Li, and X. Fang, Drawing dominant dataset from big sensory data in wireless sensor networks, in *Proc. of IEEE Conference on Computer communications*, Hong Kong, China, 2015, pp. 531–539.

[12] D. Yu, Y. Zhang, Y. Huang, H. Jin, J. Yu, and Q.-S. Hua, Exact implementation of abstract MAC layer via carrier sensing, in *Proc. of IEEE Conference on Computer Communications*, Honolulu, HI, USA, 2018, pp. 1196–1204.

[13] X. Gong, Q.-S. Hua, L. Qian, D. Yu, and H. Jin, Communication-efficient and privacy-preserving data aggregation without trusted authority, in *Proc. of IEEE Conference on Computer Communications*, Honolulu, HI, USA, 2018, pp. 1250–1258.

[14] S. Cheng, Z. Cai, and J. Li, Curve query processing in wireless sensor networks, *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5198–5209, 2014.

[15] Z. Cai, G. Lin, and G. Xue, Improved approximation algorithms for the capacitated multicast routing problem, in *Proc. of International Computing and Combinatorics Conference*, Berlin, Germany, 2005, pp. 136–145.

[16] Z. Cai, Z.-Z. Chen, and G. Lin, A 3.4713-approximation algorithm for the capacitated multicast tree routing problem, *Theoretical Computer Science*, vol. 410, no. 52, pp. 5415–5424, 2009.

[17] Z. Cai, R. Goebel, and G. Lin, Size-constrained tree partitioning: Approximating the multicast k-tree routing problem, *Theoretical Computer Science*, vol. 412, no. 3, pp. 240–245, 2011.

[18] J. Elson and D. Estrin, Time synchronization for wireless sensor networks, in *Proc. of the 2001 International Parallel and Distributed Processing Symposium*, San Francisco, CA, USA, 2001.

[19] J. Li and J. Li, Data sampling control, compression and query in sensor networks, *International Journal of Sensor Networks*, vol. 2, nos. 1&2, pp. 53–61, 2007.

[20] K. Deng, X. Zhou, and H. Tao, Multi-source skyline query processing in road networks, in *Proc. of IEEE 23rd International Conference on Data Engineering*, Istanbul, Turkey, 2007, pp. 796–805.

[21] Z. Peng and C. Wang, Member promotion in social networks via skyline, *World Wide Web*, vol. 17, no. 4, pp. 457–492, 2014.

[22] C. Wang, C. Wang, G. Guo, X. Ye, and S. Y. Philip, Efficient computation of g-skyline groups, *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 4, pp. 674–688, 2017.

[23] H. Chen, S. Zhou, and J. Guan, Towards energy-efficient skyline monitoring in wireless sensor networks, in *Proc. of European Conference on Wireless Sensor Networks*, Istanbul, Turkey, 2007, pp. 101–116.

[24] Z. He, Z. Cai, S. Cheng, and X. Wang, Approximate aggregation for tracking quantiles and range countings in wireless sensor networks, *Theoretical Computer Science*, vol. 607, pp. 381–390, 2015.

[25] G. Hartl and B. Li, Infer: A Bayesian inference approach towards energy efficient data collection in dense sensor networks, in *Proc. of 25th IEEE International Conference on Distributed Computing Systems*, Columbus, OH, USA, 2005, pp. 371–380.

[26] J. Li, M. Siddula, X. Cheng, W. Zheng, Z. Tian, and Y. Li, Sampling-based $\delta$-approximate data aggregation in sensor equipped IoT networks, in *Proc. of International Conference on Wireless Algorithms, Systems, and Applications*, Tianjin, China, 2018, pp. 249–260.

[27] R. Lachowski, M. E. Pellenz, M. C. Penna, E. Jamhour, and R. D. Souza, An efficient distributed algorithm for constructing spanning trees in wireless sensor networks, *Sensors*, vol. 15, no. 1, pp. 1518–1536, 2015.

[28] S. Cheng and J. Li, Sampling-based $(\epsilon, \delta)$-approximate aggregation algorithm in sensor networks, in *Proc. of IEEE International Conference on Distributed Computing Systems*, Montreal, Canada, 2009, pp. 273–280.

[29] MPR-Mote Processor Radio Board User’s Manual, Crossbrow Inc., Milpitas, CA, USA, 2004.

[30] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, Performance measurements of motes sensor networks, in *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Venice, Italy, 2004, pp. 174–181.

[31] Statistics and data about the vehicle miles travelled by vehicle type, road category and region, https://www.gov.uk/government/collections/road-traffic-statistics, 2013.

[32] Z. Huang, L. Wang, K. Yi, and Y. Liu, Sampling-based algorithms for quantile computation in sensor networks, in *Proc. of International Conference on Management of Data*, Athens, Greece, 2011, pp. 745–756.

[33] B. Gedik, L. Liu, and S. Y. Philip, ASAP: An adaptive sampling approach to data collection in sensor networks, *IEEE Transactions on Parallel and distributed systems*, vol. 18, no. 12, pp. 1766–1783, 2007.

[34] A. S. Silberstein, R. Braynard, C. Ellis, K. Munagala, and J. Yang, A sampling-based approach to optimizing top-k queries in sensor networks, in *Proc. of the 22nd International Conference on Data Engineering*, Atlanta, GA, USA, 2006, pp. 68–68.

[35] S. Wang, B. C. Ooi, A. K. Tung, and L. Xu, Efficient skyline query processing on peer-to-peer networks, in *Proc. of IEEE 23rd International Conference on Data Engineering*, Providence, RI, USA, 2007, pp. 1126–1135.

[36] J. Li, S. Cheng, Z. Cai, J. Yu, C. Wang, and Y. Li, Approximate holistic aggregation in wireless sensor
networks, ACM Transactions on Sensor Networks, vol. 13, no. 2, pp. 1–11, 2017.

[37] J. Li, S. Cheng, Z. Cai, Q. Han, and H. Gao, Bernoulli sampling-based \((\epsilon, \delta)\)-approximate frequency query in mobile ad hoc networks, in Proc. of the 10th International Conference on Wireless Algorithms, Systems, and Applications, Qufu, China, 2015, pp. 315–324.

Ji Li received the PhD degree from Georgia State University, USA in 2018. He received the BS degree from the School of Computer Science and Technology, Heilongjiang University, China, in 2012. He is currently an assistant professor in the College of Computing and Software Engineering, Kennesaw State University, USA. His research focuses on mobile crowd sensing and big data management in IoT networks.

Akshita Maradapu Vera Venkata Sai received the bachelor degree of technology from Gandhi Institute of Technology and Management University, India and the MS degree from Georgia State University, USA, in 2016 and 2017, respectively. She is currently pursuing the PhD degree with the Department of Computer Science, Georgia State University, USA. Her research interests include mobile social networks, security and privacy, deep learning, and big data mining.

Xiuzhen Cheng received the MS and PhD degrees from the University of Minnesota Twin Cities, Minneapolis, MN, USA, in 2000 and 2002, respectively. She is a professor with the Department of Computer Science, the George Washington University, Washington, DC, USA. She was a program director for the National Science Foundation from April to October in 2006 (full time) and from April 2008 to May 2010 (part time). She has published more than 170 peer-reviewed papers. Her current research interests include privacy-aware computing, wireless and mobile security, dynamic spectrum access, mobile handset networking systems (mobile health and safety), cognitive radio networks, and algorithm design and analysis. She has served on the editorial board of several technical journals (e.g., IEEE Transactions on Parallel and Distributed Systems and IEEE Wireless Communications) and the technical program committee of various professional conferences/workshops (e.g., IEEE Conference on Computer Communications, IEEE International Conference on Distributed Computing Systems, IEEE International Conference on Communications, and IEEE/ACM International Symposium on Quality of Service). She also has chaired several international conferences (e.g., IEEE Conference on Communications and Network Security and International Conference on Wireless Algorithms, Systems, and Applications).

Wei Cheng received the BS and MS degrees from the National University of Defense Technology, Changsha, China, in 2002 and 2004, respectively, and the PhD degree from the George Washington University, Washington, DC, USA, in 2010. He is currently an assistant professor with Virginia Commonwealth University, Richmond, VA, USA. He was a post-doctoral scholar with the University of California at Davis, Davis, CA, USA. His current research interests include wireless networks, cyber-physical networking systems, and algorithm design and analysis. In particular, he is interested in localization, security, fog computing, and smart cities. He is a member of the ACM.

Zhi Tian received the BEng degree from the University of Science and Technology of China, Hefei, China in 1994, and the MS and PhD degrees from George Mason University, Fairfax, VA, USA in 1998 and 2000, respectively. She is a professor in the Electrical and Computer Engineering Department of George Mason University, Fairfax, VA, USA, as of January 2015. Prior to that, she was on the faculty of Michigan Technological University from 2000 to 2014. Her research interests lie in statistical signal processing, wireless communications, and wireless sensor networks. She is an IEEE fellow. She is an elected member of the IEEE Signal Processing for Communications and Net-working Technical Committee and a member of the BigData Special Interest Group IEEE Signal Processing Society. She served as an associate editor for IEEE Transactions on Wireless Communications and IEEE Transactions on Signal Processing. She is a distinguished lecturer of the IEEE Vehicular Technology Society from 2013 to 2017 and the IEEE Communications Society from 2015 to 2016.

Yingshu Li received the BS degree from the Department of Computer Science and Engineering, Beijing Institute of Technology, Beijing, China in 2001, and the MS and PhD degrees from the Department of Computer Science and Engineering, University of Minnesota Twin Cities, Minneapolis, MN, USA, in 2003 and 2005, respectively. She is currently an associate professor with the Department of Computer Science, Georgia State University, Atlanta, GA, USA. Her research interests include wireless networking, sensor networks, sensory data management, social networks, and optimization. She received an NSF CAREER Award.