

# Energy-Efficient Skyline Query Processing and Maintenance in Sensor Networks

Weifa Liang  
Australian National University  
wliang@cs.anu.edu.au

Baichen Chen  
Australian National University  
baichen@cs.anu.edu.au

Jeffrey Xu Yu  
Chinese Univ. of Hong Kong  
yu@se.cuhk.edu.hk

## ABSTRACT

The skyline query, as an important operator in databases for multi-preference analysis and decision making, has received much attention recently due to its wide application backgrounds. In this paper, we consider the skyline query problem in Wireless Sensor Network with an objective to maximize the network lifetime by proposing filter-based distributed algorithms for skyline evaluation and maintenance. We also conduct preliminary experiments to evaluate the performance of the proposed algorithms. The experimental results demonstrate that the proposed algorithms significantly outperform existing algorithms on various datasets.

## Categories and Subject Descriptors

H.2.4 [Database Management]: Query processing

## General Terms

Algorithms, Experimentation, Performance

## Keywords

skyline, filtering algorithm, sensor network, network lifetime

## 1. INTRODUCTION

As one of the popular queries in modern databases, the skyline query has received much attention recently by the database community due to its wide applications in multi-criteria decision making. Most previous studies [4, 3] for the skyline problem are based on centralized databases and distributed computing environment. Recently some effort has been taken for skyline query evaluation and maintenance in wireless sensor networks (WSNs), such as [2, 1, 5]. In this paper we focus on skyline query evaluation and maintenance in WSNs by devising energy-efficient evaluation and maintenance algorithms for skyline queries to maximize the network lifetime. To achieve that, we invent a new concept - the local skyline certificate, which will be used to find a global filter. Based on the global filter, we then devise filter-based, distributed evaluation and maintenance algorithms for skyline queries.

## 2. PRELIMINARIES

We consider a wireless sensor network consisting of  $n$  sensor nodes randomly deployed in a region of interest, each

measuring a certain number  $d$  of numeric attributes and a base station with unlimited energy supply serves as the gateway between the sensor network and the users. We assume that the transmission range of each sensor node is identical. Two sensor nodes are the *neighboring nodes* if they are within the transmission range of each other. The amounts of transmission and reception energy by transmitting or receiving a message containing  $k$  readings are  $\rho_t + R * d * k$  and  $\rho_r + r_e * d * k$  respectively, where  $\rho_t$  and  $\rho_r$  are the energy consumptions of handshaking to build the communication channels between the sender and receiver and transmitting and receiving the message header. For different messages the values of  $\rho_t$  and  $\rho_r$  are identical. Given a sensor network, each sensor node generates a data point at a time step. A data point consists of  $d$  attributes. The dataset  $P$  contains all the points and the *skyline query* on  $P$  is to return the points that are not dominated by any other points in  $P$ .

We assume that each node  $v$  contains a set  $S(v)$  of points. We refer to  $S(v)$  as the *snapshot of data points* during a given time interval. The entire set  $P = \cup_{v \in V} S(v)$ , where  $V$  is the set of sensor nodes. Assume that a routing tree rooted at the base station  $r$  is built. A simple merge-based algorithm for skyline queries proceeds as follows. Each node  $v \in V$  computes its local skyline  $LSK(v)$  on set  $S(v)$ , followed by computing the skyline  $SK(v)$  on set  $\cup_{u \in C(v)} SK(u) \cup LSK(v)$  which is equivalent to computing the skyline on set  $\cup_{u \in desc(v)} S(u)$ , where  $C(v)$  is the child set of  $v$  and  $desc(v)$  is the set of descendants of  $v$  in the routing tree. The skyline merge-based algorithm performs in-network skyline merging operation at each node and finally the root node  $r$  computes a set  $SK(r)$ , which is the global skyline. We refer to this algorithm as algorithm `Sky_Merge`( $n, S(v), r$ ), which will serve for the benchmark purpose.

## 3. SKYLINE EVALUATION ALGORITHM

In this section a filter-based algorithm for skyline query evaluation is proposed. It first finds a global skyline filter consisting of potential global skyline points, then finds a proper subset of global skyline on set of non-filtered points through in-network processing by using the global skyline filter, finally removes those dominated points by the points in the found proper subset from the global skyline filter. The final global skyline consists of the remaining global filter points and the found global skyline points.

### 3.1 Finding a skyline certificate

The *certificate* of a skyline is a subset of the skyline. We will make use of the skyline certificate to find a global sky-

line filter. Specifically, let  $SK(S)$  be the skyline on set  $S$ , we choose  $k = \max\{\lfloor \alpha * |SK(S)| \rfloor, 1\}$  points from  $SK(S)$  as the certificate of  $SK(S)$ , where  $\alpha$  is a constant with  $0 < \alpha \leq 1$ . We devise an approximation algorithm for finding a certificate of size  $k$  for a local skyline at node  $v$  whose points can dominate as many points as possible. The algorithm consists of  $k$  iterations, and within each iteration, a skyline point is chosen if it dominates the maximum number of points in the local skyline that are not dominated by the points in the current certificate.

### 3.2 Evaluation algorithm for skyline queries

We now propose the filter-based algorithm for skyline evaluation. We start by finding the global skyline filter. Each node  $v$  first computes the local skyline  $LSK(v)$  on set  $S(v)$ . If  $v$  is the leaf node,  $v$  computes the certificate  $cerf(v)$  of  $LSK(v)$  and forwards it to its parent.  $LSK'(v) = LSK(v) - cerf(v)$  is the set of the remaining local skyline points. Otherwise ( $v$  is the internal node),  $v$  computes the skyline  $SK(v)$  on set  $LSK(v) \cup_{u \in C(v)} filter(u)$  where  $u$  is a child node of  $v$ , and  $LSK'(v) = SK(v) - cerf(v)$ . As a result, the root  $r$  computes the skyline  $SK(r)$  on set  $\bigcup_{u \in C(r)} filter(u)$ .  $SK(r)$  is then used as the Global Skyline Filter, referred to as  $GSF(P) = SK(r)$ .  $GSF(P)$  is broadcast to each node  $v$  in the sensor network, and node  $v$  removes the dominated points by the points in  $GSF(P)$  from  $LSK'(v)$  for every  $v \in V$ . Denote by  $LSK'(v)$  be the set of  $LSK'(v)$  after use of the points in  $GSF(P)$  for filtering.

What followed is to find a proper subset of the global skyline. We apply the simple merge-based algorithm on the set of non-filtered points  $\bigcup_{v \in LSK'(v)}$  by traversing the nodes in the routing tree in a bottom-up fashion to find the global skyline  $SK'(r)$  at the root  $r$ .

We finally proceed the cleaning up stage by removing those dominated points by the points in  $SK'(r)$  from  $GSF(P)$ . Let  $GSF'(P)$  be the set of remaining points in  $GSF(P)$ . Then, the global skyline  $SK(P)$  on set  $P$  is  $SK'(r) \cup GSF'(P)$ .

## 4. SKYLINE MAINTENANCE ALGORITHM

We assume that the point updating and skyline evaluation can be done within a time step. Suppose  $SK(P_t)$  is the global skyline on set of valid points in the sensor network at time step  $t$ ,  $CSK(v)_t$  is the set of candidate skyline points at node  $v$ ,  $ADD(v)_t$  is the set of the newly generated points at node  $v$ , and  $LSK(v)_t$  is the local skyline on the points of node  $v$  at time step  $t$ .  $Remove(S)_t$  is referred to as the expired points in set of points  $S$  at time step  $t$ . The skyline maintenance algorithm at time step  $t$  consists of two stages: find the new global skyline, followed by updating the global skyline filter at each node, using the new global skyline.

In the first stage, each node  $v$  first removes the expired points and the points dominated by  $ADD(v)_t$  from  $CSK(v)_t$ . Then  $v$  computes the local skyline  $LSK(v)_t$  on set  $CSK(v)_t \cup ADD(v)_t$ , removes the points that are dominated by  $SK(P_{t-1})$  from  $LSK(v)_t$  and appends  $ADD(v)_t$  to the updated  $CSK(v)_t$ . The simple merge-based algorithm is then applied to the routing tree to find the new global skyline on set  $\bigcup_{v \in V} LSK(v)_t$ . In the end the base station  $r$  computes the skyline  $SK(r)_t$  on set  $\bigcup_{u \in C(r)} SK(u)_t \cup CSK(r)_t$ . The global skyline  $SK(P_t)$  on set  $P_t$  is then  $SK(P_t) = SK(r)_t \cup SK(P_{t-1}) - Remove(SK(P_{t-1}))_t - \{p \mid q \prec p, p \in SK(P_{t-1}), q \in SK(r)_t\}$ . What followed is to broadcast  $DSK_t = SK(r)_t -$

$SK(P_{t-1})$  to every node in the network. The global filter at each node is updated, i.e,  $GSF(v)_t = SK(P_t) = DSK_t \cup SK(P_{t-1}) - Remove(SK(P_{t-1}))_t - \{p \mid q \prec p, p \in SK(P_{t-1}), q \in DSK_t\}$ .

## 5. PERFORMANCE EVALUATION

We evaluate the performance of proposed algorithms in terms of the total energy consumption, the maximum energy consumption, and the network lifetime. Assume that 500 sensor nodes are deployed in a  $100m \times 100m$  by the NS-2 simulator. The transmission range of each sensor is  $10m$  and the initial energy capacity of each sensor is  $10^6 mJ$ . We set  $R = 0.0144 mJ$ ,  $r_e = 0.00576 mJ$ ,  $\rho_t = 0.1152 mJ$  and  $\rho_b = 0.04608 mJ$ . Assume that there are 1,000 randomly generated points at each node initially. The dimensionality of data points varies from 2 to 5. The certificate size of a local skyline at node  $v$  in the proposed algorithm **Filter- $\alpha$**  is  $k = \max\{\lfloor \alpha * |SK(v)| \rfloor, 1\}$  and  $\alpha = 0.2$ . We refer to the proposed dynamic filter algorithm by Huang et al [2] as algorithm **DF**, the single (tuple) point filter algorithm and the grid index filter algorithm by Xin et al [5] as algorithm **TF** and algorithm **GI**, respectively. It can be seen from Fig. 1 that algorithm **Filter- $\alpha$**  outperforms all the other algorithms significantly on all dimensional datasets. With the growth of dimensionality, the total and maximum energy consumptions by the simple merge-based algorithm are at least twice of the amounts by algorithm **Filter- $\alpha$** , as shown in Figures 1(a) and (b). Meanwhile, Fig. 1(c) indicated that the ratio of network lifetime of algorithm **Filter- $\alpha$**  to the best existing algorithm **TF** increases from 1.1 to 1.7 with the increase of the dimensionality  $d$  from 2 to 5.

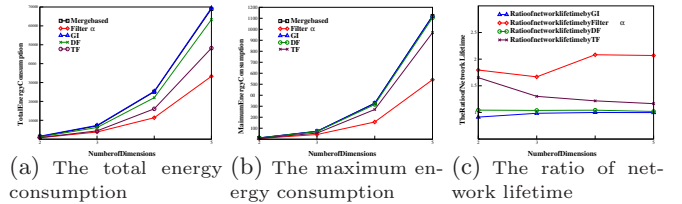


Figure 1: The performance of different evaluation algorithms on various datasets

## 6. CONCLUSIONS

In this paper we first devised a novel, distributed evaluation algorithm for skyline query evaluation on data snapshot. We then proposed an energy-efficient maintenance algorithm for skyline maintenance within sliding windows. The experimental results demonstrated that the proposed algorithms significantly outperform existing algorithms on various datasets.

## 7. REFERENCES

- [1] H. Chen, S. Zhou and J. Guan. Towards energy-efficient skyline monitoring in wireless sensor networks. *Proc. of European Workshop on Wireless Sensor Networks*, 2007.
- [2] Z. Huang, et al. Skyline queries against mobile lightweight devices in MANETs. *Proc of ICDE*, 2006.
- [3] K.C. Lee, B. Zheng, H. Lu and W-C. Lee. Approaching the skyline in Z order. *Proc of VLDB*, 2007.
- [4] D. Papadias, Y. Tao, G. Fu and B. Seeger. An optimal and progressive algorithm for skyline queries. *Proc of SIGMOD*, 2003.
- [5] J. Xin, G. Wang, L. Chen, X. Zhang, and Z. Wang. Continuously maintaining sliding window skyline in a sensor network. *Proc of DASFAA*, 2007.