Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 813507, 10 pages
http://dx.doi.org/10.1155/2015/813507

Research Article

Alternative Tuples Based Probabilistic Skyline Query Processing in Wireless Sensor Networks

Zhiqiong Wang,[1] Junchang Xin,[2] and Pei Wang[2]

[1]*Sino-Dutch Biomedical and Information Engineering School, Northeastern University, Shenyang 110169, China*
[2]*School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China*

Correspondence should be addressed to Zhiqiong Wang; wangzq@bmie.neu.edu.cn

Received 31 July 2015; Revised 10 December 2015; Accepted 13 December 2015

Academic Editor: Filippo Ubertini

As uncertainty is the inherent character of sensing data, the processing and optimization techniques for Probabilistic Skyline (PS) in wireless sensor networks (WSNs) are investigated. It can be proved that PS is not decomposable after analyzing its properties, so in-network aggregation techniques cannot be used directly to improve the performance. In this paper, an efficient algorithm, called Distributed Processing of Probabilistic Skyline (DPPS) query in WSNs, is proposed. The algorithm divides the sensing data into candidate data (CD), irrelevant data (ID), and relevant data (RD). The ID in each sensor node can be filtered directly to reduce data transmissions cost, since, only according to both CD and RD, PS result can be correctly obtained on the base station. Experimental results show that the proposed algorithm can effectively reduce data transmissions by filtering the unnecessary data and greatly prolong the lifetime of WSNs.

1. Introduction

Recently, it is found that wireless sensor networks (WSNs) have a more and more important impact on the ways to collect and use information from the physical world. With the rapid development of microelectronics technology, communication technology, and the embedded technology, WSNs have become a common concern to industry and academia because of their great commercial prospects and its value of academic research [1–3]. For example, we can prevent forest fires by monitoring the temperature and humidity in real time. Influenced by manifold factors such as hardware devices, sensor technology, communication quality, and the surrounding environment, sensing data collected by sensor nodes are often with inaccurate or low confidence. That is to say, the temperature and humidity data acquired by sensor nodes are not accurate. As uncertainty is an inherent property of sensing data, to some extent, sensing data are uncertain data essentially.

As one of the most important means, multiobjective decision, skyline query [4–8] processing technologies have brought a large number of excellent researches, both in WSNs [9–16] and for uncertain data [17–26]. Considering a wireless sensor network that consists of a large amount of sensor nodes deployed in a geographical region, sensing data are collected by these distributed sensor nodes. Accordingly, there could be multiple sensor nodes deployed in certain zones to promote the precision of uncertain data. As a result, many queries in WSNs that rarely need transmitting every piece of sensing data in the local sensor nodes have been well studied to reduce the communication cost and to speed up the computation [9–16], for instance, sliding window skylines in sensor network [11, 12], continuous skyline monitoring in WSNs [10], probabilistic query of uncertain data streams [18, 19], dynamic (or relative) skylines [25], and distributed uncertain skyline query [26]. Nevertheless, most of these researches are studied under a centralized system setting.

In this paper, an efficient algorithm, called Distributed Processing of Probabilistic Skyline (DPPS) query in WSNs, is proposed. It explores the problem of PS query processing in distributed WSNs, in which there exist alternative tuples. The basic idea is to perform data pruning and aggregation

at sensor node such that only the data required for final processing are transferred to the base station. By comparing the data communication cost of DPPS and Centralized Algorithm (CA) to examine the effectiveness of the DPPS, we also perform sensitivity tests to observe the behavior of examined DPPS under various parameter settings. The result validates our ideas and shows the superiority of our proposal.

In summary, the contributions of this paper are as follows:

(i) The properties of PS have been analyzed, and we prove theoretically that PS query is not decomposable.

(ii) An efficient algorithm, called Distributed Processing of Probabilistic Skyline (DPPS) query in WSNs, is proposed, which reduces the in-network amount of data transmission by filtering the irrelevant data on the sensor nodes.

(iii) Last but not least, the experimental results show that DPPS has advantages of data transmission in WSNs over CA.

The rest of this paper is organized as follows. The related work is introduced briefly in Section 2. Section 3 introduces the important notions and theorems. In Section 4, the DPPS is depicted in detail. And we analyze the performance evaluation of DPPS in Section 5. Finally, the conclusion of this paper is presented in Section 6.

2. Related Work

Here, we review representative work in the areas of (1) skyline query processing in WSNs and (2) skyline query processing on uncertain data.

Skyline Query Processing in Sensor Networks. An extensive number of research works in this area have appeared in the literature [9–16]. Due to the limited energy budget available of sensor nodes, the primary issue is how to develop energy-efficient techniques to reduce communication and energy costs in the networks. In literature [9], Wang et al. analyzed the properties of reverse skyline query and presented a skyband-based approach to tackle the problem of reverse skyline query answering efficiently over WSNs. Chen et al. [10] addressed the problem of continuous skyline monitoring in WSNs and presented a hierarchical threshold-based approach, MINMAX, to minimize the transmission traffic. Two papers in the literature [11, 12] investigated the sliding window skylines in sensor network. The former put forward an energy-efficient algorithm, SWSMA, to continuously maintain sliding window skylines over a wireless sensor network. The algorithm employs tuple filter or grid filter within each sensor to reduce the amount of data to transmit and save the energy consumption as a consequence, while the latter proposed a method EES which uses a mapping function to map the data into a smaller range of integers and carries out the skyline of the mapped set as the mapped skyline filter (MSF). Chen et al. [13] partitioned the entire data set into disjoint subsets and returned the skyline points progressively through examining the subsets one by one. Also, a global filter consisting of some found skyline points in the processed subsets is used to filter out those unlikely skyline points from the rest of subsets for transmission. Shen et al. [14] researched location-based skyline queries in WSNs and raised an energy-efficient approach of Ring-Skyline (RS) which divides the monitoring area into several rings and adopts in-network query processing to reduce energy consumption. In [15], Xin et al. raised an energy-efficient multiskyline evaluation (EMSE) algorithm to evaluate multiple skyline queries effectively in WSNs. EMSE utilizes both global and local optimization mechanisms to eliminate unnecessary data transmission. In literature [16], a new lter-based method, called SKYFILTER, was brought up for skyline query processing. The method provides an enhanced efficiency by reduction of the total wireless communication between sensor nodes.

Skyline Query Processing on Uncertain Data. In literature [17], the bottom-up and top-down algorithms are put forward to process p-skyline queries; a p-skyline contains all the objects whose skyline probabilities are at least p. It can filter the unqualified objects efficiently with the help of the grid-based space division algorithm and weight-counting algorithm. Literature [18, 19] investigated the PS query of uncertain data streams. The former proposed an approach, *candidate list*, to compute a PS on a large number of uncertain tuples within the sliding window, and the later studied the problem of efficiently computing the skyline over sliding windows on uncertain data elements against probability thresholds. The all skyline query problem over discrete uncertain data sets was first researched in [20], in which space splitting algorithm and dominating counting algorithm were raised. In [21], Böhm et al. attempted to model the uncertainty with pdfs (probability density function) and investigated the skyline query over the pdf modeled uncertain data. Additionally, in [22], the objects are indexed with the Gauss-tree in the parameter space to improve the pruning efficiency, where the leaf nodes store the objects with expectation and variance. Ding and Jin [23] first address the distributed uncertain skyline query problem and the DSUD and e-DSUD algorithms were raised to process the queries over tuple-level uncertain data with the processing framework, in which the uncertain tuples are independent of each other. For skyline computation in highly distributed environments, Hose and Vlachou [24] provide a good survey of existing approaches, where the uncertain skyline queries and the open research directions are discussed. The reverse skyline query over uncertain database retrieves all the uncertain objects whose dynamic (or relative) skylines [25] contain a user-specified query object with a probability not less than a user-specified threshold. In [26], efficient exact and approximate algorithms are addressed to tackle this problem that skyline probability computation over uncertain preferences is $\sharp P$-complete.

As opposed to our investigation, these researches either ignored the uncertainty of sensing data or considered no particularity of wireless sensor network environment. All of them failed to solve PS query processing problems effectively in WSNs.

Table 1: The meanings of frequently used symbols.

Symbol	Meanings
t, t_i	Uncertain tuple
U	Universal set of all the uncertain tuples
D	A dimension space with d-dimension
U_i	Subset of universal set U
τ	Set of alternative tuples, we have $U = \{\tau_1, \tau_2, \ldots\}$
τ^t	τ that dominates t
T	A set composed of τ^t
W, W_i	Possible worlds
PW	Set of possible worlds in U, which is the subset of U

3. Preliminaries

3.1. Problem Statement. In this section, some important concepts are defined; also, some theorems are proved to be true. The variable p is the threshold of the Probabilistic Skyline and the meanings of frequently used symbols are listed in Table 1.

Consider a WSN that consists of a lot of sensor nodes deployed in a geographical region. Feature readings (e.g., temperature and humidity) are collected from these distributed sensor nodes. Multiple sensors are deployed at certain zones in order to improve monitoring quality. Figure 1 shows a wireless sensor network (with a two-tier hierarchical topology) that monitors forest temperature and humidity in different zones (denoted as different color). In this network, sensor nodes are grouped into clusters, where cluster heads are responsible for local processing and for reporting aggregated results to the base station. As shown, p_2 and p_6 denote the cluster heads for clusters A and B, correspondingly.

A table is shown in Figure 1, representing a snapshot of temperature and humidity records collected from the sensor network. As shown, each tuple records both possible temperature and humidity corresponding to a location. The confidence value associated with a tuple indicates the existence probability of those particular temperature and humidity. For example, there are two data tuples generated for Location A. The temperature and humidity in these two tuples are both valid (i.e., with measured confidences).

Definition 1 (possible world semantics [23]). We use D to denote a d-dimensional space and U to denote the universal set of all uncertain tuples in the d-dimensional space D. Each tuple has a probability $P(t_i)$ $(0 \leq P(t_i) \leq 1)$ to occur, and v_{ij} $(1 \leq j \leq d)$ denotes the jth dimension value. The tuples that cannot exist at the same time are alternatives. A possible world W is instantiated by taking a set of tuples from the alternative relation.

For example, uncertain tuples t_1 and t_2 in Figure 1 are alternatives. The various dimensions numerical values of t_1 and t_2 indicate the relevant information of the region A. Due to the property of alternative tuples, both of them may occur but cannot occur simultaneously.

The aggregate confidence of τ is the sum of the confidence values of all its alternative tuples; that is, $P(\tau) = \sum_{t \in \tau} P(t)$.

Table 2: An example of possible worlds.

Possible world W	Probability $\Pr(W)$
$W_1 = \{\varnothing\}$	$(1 - 0.9) * (1 - 1) = 0$
$W_2 = \{t_1, t_3\}$	$0.5 * 0.1 = 0.05$
$W_3 = \{t_1, t_4\}$	$0.5 * 0.4 = 0.20$
$W_4 = \{t_1, t_5\}$	$0.5 * 0.1 = 0.05$
$W_5 = \{t_1, t_6\}$	$0.5 * 0.2 = 0.1$
$W_6 = \{t_1, t_7\}$	$0.5 * 0.2 = 0.1$
$W_7 = \{t_2, t_3\}$	$0.4 * 0.1 = 0.04$
$W_8 = \{t_2, t_4\}$	$0.4 * 0.4 = 0.16$
$W_9 = \{t_2, t_5\}$	$0.4 * 0.1 = 0.04$
$W_{10} = \{t_2, t_6\}$	$0.4 * 0.2 = 0.08$
$W_{11} = \{t_2, t_7\}$	$0.4 * 0.2 = 0.08$
$W_{12} = \{t_3\}$	$(1 - 0.5 - 0.4) * 0.1 = 0.01$
$W_{13} = \{t_4\}$	$(1 - 0.5 - 0.4) * 0.4 = 0.04$
$W_{14} = \{t_5\}$	$(1 - 0.5 - 0.4) * 0.1 = 0.01$
$W_{15} = \{t_6\}$	$(1 - 0.5 - 0.4) * 0.2 = 0.02$
$W_{16} = \{t_7\}$	$(1 - 0.5 - 0.4) * 0.2 = 0.02$

For instance, corresponding to location A, $\tau_A = \{t_1, t_2\}$; that is, t_1 and t_2 are alternative tuple instances (or simply called alternatives) of τ_A. Consider $P(\tau_A) = 0.3 + 0.4 = 0.7$. In the same way, we can get that $\tau_B = \{t_3, t_4, t_5, t_6, t_7\}$ and $P(\tau_B) = 0.1 + 0.4 + 0.1 + 0.2 + 0.2 = 1$. The probability of all possible worlds in U is shown in Table 2.

Definition 2 (skyline). Given a set U of uncertain tuples in the d-dimensional space D, a skyline query retrieves tuples in U that are not dominated by any other tuple. For two tuples t_i and t_j in U, tuple t_i dominates t_j (denoted as $t_i \prec t_j$) if it is not worse than t_j in all dimensions $(\forall k \in [1, d],\ v_{ik} \geq v_{jk})$ and better than t_j at least in one $(\exists l \in [1, d],\ v_{il} > v_{jl})$. The probability that t_i dominates t_j is t_i's existing probability denoted as $P(t_i \prec t_j) = P(t_i)$.

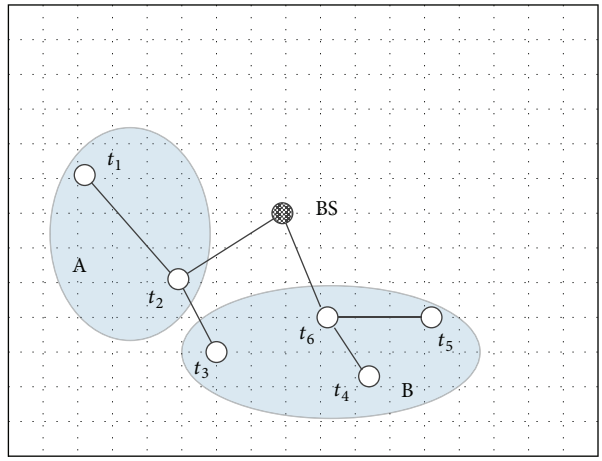
Definition 3 (skyline probability). Given a set U of uncertain tuples in the d-dimensional space D, the set of possible worlds based on set U is denoted in the form of $PW = \{W_1, W_2, \ldots, W_n\}$. We assume that there exit uncertain tuple t and possible world subset $PW'_t = \{W'_1, \ldots, W'_m\} \subseteq PW$, if t and PW satisfy that

(1) for any possible world $W \in PW'_t$, the uncertain tuple t belongs to the skyline of W; that is, $t \in \text{Skyline}(W)$;

(2) for any possible world $W \in PW\text{-}PW'_t$, the uncertain tuple t does not belong to the skyline of W; that is, $t \notin \text{Skyline}(W)$.

Then, we conclude that the skyline probability of an uncertain tuple t is the sum of all the possible worlds' existential probability which are in the subset PW'_t; that is to say, $P_{\text{sky}}(t) = \sum_{W \in PW'_t} \Pr(W)$. For example, $P_{\text{sky}}(t_2) = \Pr(W_7) + \text{pr}(W_{10}) = 0.04 + 0.08 = 0.12$.

Assume that there exist an uncertain tuple t and an alternative tuples set $\tau^t = \{t'_1, t'_2, \ldots\}$ in the universal set



Loc.	Tuples	Temp.	Hum.	Conf.
A	t_1	1	2	0.5
A	t_2	4	3	0.4
B	t_3	2	5	0.1
B	t_4	4	3	0.4
B	t_5	5	5	0.1
B	t_6	3	2	0.2
B	t_7	4	5	0.2

Figure 1: An example of wireless sensor network.

$U = \{\tau_1, \tau_2, \ldots, \tau_m\}$. If there exists $t'_k \in \tau^t$ that dominates t, we can say τ^t dominates t ($\tau^t \prec t$). Then, the probability that τ^t dominates t can be calculated as $P(\tau^t \prec t) = \sum_{t'_k \in \tau^t, t'_k \prec t} P(t'_k)$. We use T to denote the set that is composed of all τ^t in U; that is, $T = \{\tau^t_1, \tau^t_2, \ldots, \tau^t_n\} \subseteq U$. Consequently, the skyline probability of uncertain tuple t is the product of the existent probability $P(t)$ of t and the nonexistent probability $\prod_T (1 - P(\tau^t_i))$ of $\tau^t_i \in T$; that is, $P_{\text{sky}}(t) = P(t) \times \prod_T (1 - P(\tau^t_i))$.

Definition 4 (Probabilistic Skyline). Given a set U of uncertain tuples in the d-dimensional space D and a threshold value p, then the Probabilistic Skyline of U contains all the uncertain tuples in U whose skyline probability is bigger than p, denoted as $\text{PS}(U) = \{t \mid P_{\text{sky}}(t) > p\}$.

3.2. Property Analysis

Theorem 5. *Probabilistic Skyline query is not a decomposable operator.*

Proof of Theorem 5. We first let $t_i.x > t_j.x$ represent the fact that $t_i.x$ is better than $t_j.x$ and let $t_i.y > t_j.y$ represent the fact that $t_i.y$ is better than $t_j.y$. Then, we assume that the set $U = \{t_1, t_2, t_3, t_4, t_5\}$ of uncertain tuples is depicted in Figure 2(a), and the threshold value p is 0.3. We can know that $P_{\text{sky}}(t_1) = 0.09$, $P_{\text{sky}}(t_2) = 0.3$, $P_{\text{sky}}(t_3) = 0.12$, $P_{\text{sky}}(t_4) = 0.072$, and $P_{\text{sky}}(t_5) = 0.4$ by Definition 2. Also, we have the result $\text{PS}(U) = \{t_5\}$ according to Definition 3. Now, let $U = U_1 \cup U_2$, $U_1 = \{t_1, t_2, t_5\}$, illustrated in Figure 2(b), and $U_2 = \{t_3, t_4\}$, shown in Figure 2(c). Similarly, it can be proved that $\text{PS}(U_1) = \{t_5\}$ and $\text{PS}(U_2) = \{t_4\}$. Only by $\text{PS}(U_1) \cup \text{PS}(U_2) = \{t_4, t_5\}$ demonstrated in Figure 2(d), in whatever way, we cannot obtain the result that $\text{PS}(U) = \{t_5\}$; that is to say, $\text{PS} \neq g(\text{PS}(U_1) \cup \text{PS}(U_2))$. Thus, PS query is not a decomposable operator. □

We can know that PS query is not a decomposable operator by Theorem 5; thus, we cannot improve the efficiency of PS queries in WSNs by using in-network computing technology [11, 15] directly.

Next, we will further analyze the properties of the PS query.

Theorem 6. *Given a set U of uncertain tuples in the d-dimensional space D, a tuple $t \in U_i$ and a threshold value p. $U_i = \{\tau_1, \tau_2, \ldots, \tau_m\}$ are the subset of U which contains tuples collected on the ith cluster, and one uses $T_i \subseteq T$ to denote the set that is composed of $\tau^t_k \subseteq U_i$. Thus, t does not belong to the skyline of U when it satisfies the conditions as follows:*

$$P(t) \times \prod_{T_i} \left(1 - P\left(\tau^t_k\right)\right) < p. \tag{1}$$

Proof of Theorem 6. This theorem can be proved by Definitions 2 and 3 directly. □

Theorem 7. *Given a set U of uncertain tuples in the d-dimensional space D, a tuple $t \in U_i$, and a threshold value p, then, t should be excluded when it satisfies the conditions as follows:*

$$\frac{\prod_{T_i} \left(1 - P\left(\tau^t_k\right)\right)}{1 - P\left(\tau^t_x\right)} < p \quad \left(\text{for any } \tau^t_x \subseteq T_i\right). \tag{2}$$

Proof of Theorem 7. Since $\prod_{T_i}(1 - P(\tau^t_k))/(1 - P(\tau^t_x)) < p$, $1 - P(\tau^t_x) \leq 1$, and $P(t) \leq 1$, then it can be deduced that $P_{\text{sky}}(t) = P(t) \times \prod_{T_i}(1 - P(\tau^{t_{\text{new}}}_k)) < p$. Thus, $t \notin \text{PS}(U_i)$ and $t \notin \text{PS}(U)$.

Only the skyline probability of the tuples dominated by t will be affected if we delete t. Suppose t_{new} dominated by t is a tuple in another sensor node which will possibly be interleaved with tuples in U_i at the base station, and let $P_{\text{sky}}(t_{\text{new}})$ indicate the skyline probability of t_{new}. There are two possible cases to consider.

Case 1. t_{new} itself forms a new τ_{new} because the tuples that dominate t must dominate t_{new} as well. Thus, it can be

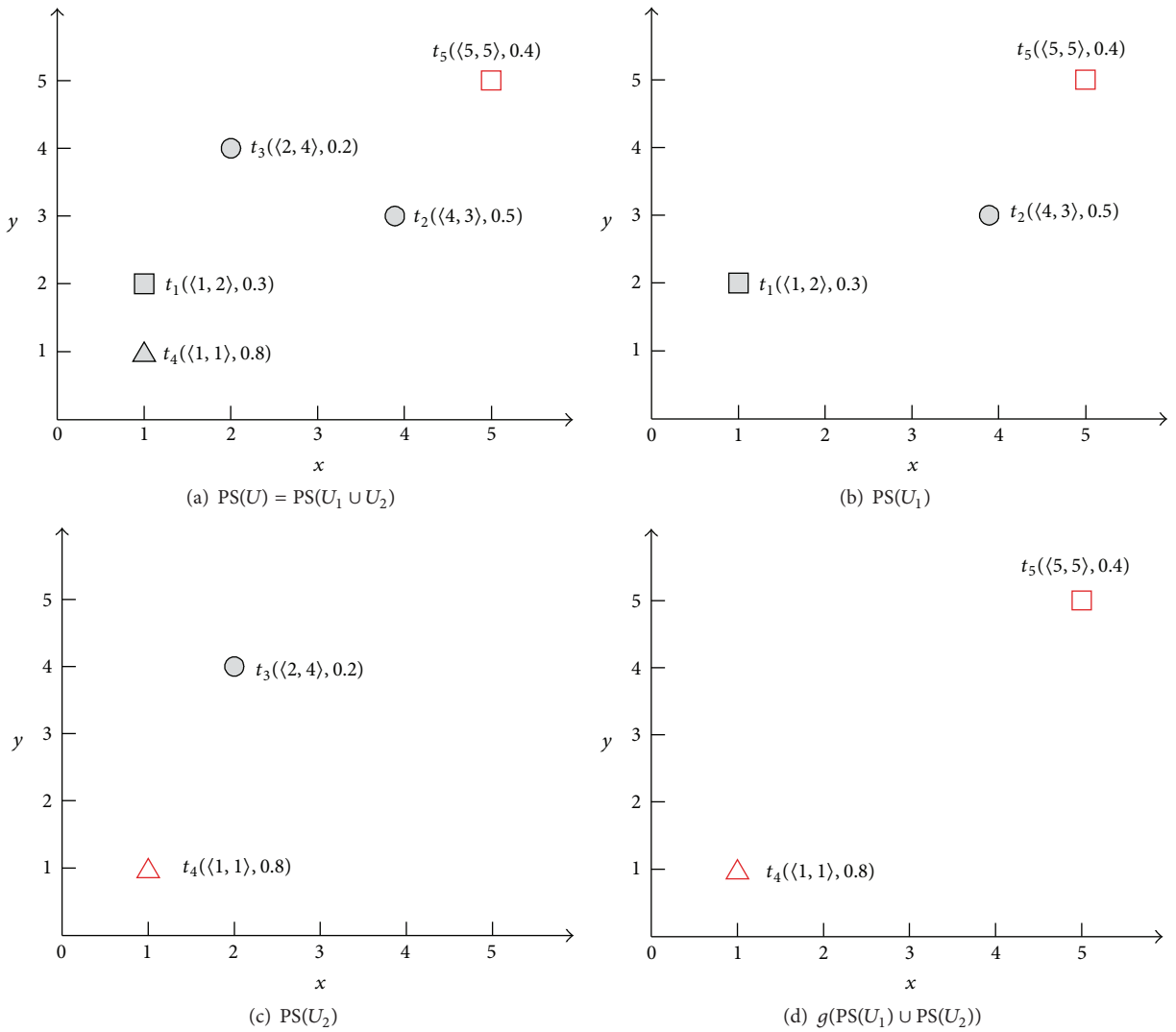


Figure 2: Example of PS query is not decomposable.

deduced that $P_{\text{sky}}(t_{\text{new}}) = P(t_{\text{new}}) \times \prod_T (1 - P(\tau_k^t)) < p$ and t_{new} will not be judged as the skyline tuple by mistake.

Case 2. t_{new} is a member of an existed τ that does exist in U_i named τ_{new}. Due to the mutual exclusiveness of tuple members in τ, t_{new} may appear in a possible world if and only if no other members of τ_{new} coexist in this possible world. By formula (2), it can be proved that $P_{\text{sky}}(t_{\text{new}}) = P(t_{\text{new}}) \times \prod_T (1 - P(\tau_k^{t_{\text{new}}})) < p$. Also, t_{new} will not be judged as the skyline tuple by mistake. □

Theorem 6 pointed out the tuples in the subset U_i that must not belong to the skyline of U clearly; that is, it pointed out the tuples that may be the skyline tuples of U. Theorem 7 evidenced that we can delete the tuples in U_i which will not affect the calculation of the skyline of U. Not all the tuples which do not belong to U_i can be deleted. The tuples that do not satisfy the conditions above will affect the calculation of skyline probability of other tuples, so we should hold them.

4. DPPS Algorithm

In this section, we propose the notions of candidate data, irrelevant data, and relevant data according to Theorems 6 and 7. Next, we take the PS query as a test case to derive candidate data and relevant data meanwhile prune the irrelevant data. Thus, irrelevant data tuples pruned in local sensor nodes will never appear in the final answer set.

Definition 8 (candidate data). In the sensing data subset $U_i \subseteq U$ on sensor node, the tuples which are candidate data (CD) of the Probabilistic Skyline query satisfy the conditions:

$$P(t) \times \prod_T \left(1 - P\left(\tau_k^t\right)\right) > p. \tag{3}$$

```
// input: The message set of child node M_S, the local sensing data R,
//        the threshold value p
// output: The data set S which will be submitted to the parent node
For each element m in M_S Do
    tempCD = tempCD + m.CD;
    tempRD = tempRD + m.RD;
end For
tempCD = tempCD + R;
For each element t in tempCD Do
    temp = 1;
    n = t.getDominatingNumber(tempCD + tempRD); // get the number n of τ^t
    tempT = t.getDominatingT(tempCD + tempRD); // and all τ^t dominate t
    For each τ^t in tempT Do
        calculate P(τ_k^t);                    // get τ_k^t's domination probability
    end For
    If ∏_T(1 − P(τ_k^t))/(1 − P(τ_x^t)) < p Then
        tempCD.Delete(t);                      // delete ID from CD set
    Else If P(t) × ∏(1 − P(τ_k^t)) ≤ p Then
        tempRD.Add(t);                         // transmit RD to RD set from CD set
        tempCD.Delete(t);
    end If
end For
For each element t in tempRD Do
    temp = 1;
    n = t.getDominatingNumber(tempCD + tempRD);    // get the number n of τ^t
    tempT = t.getDominatingT(tempCD + tempRD);     // and all τ^t dominate t
    For each τ^t in tempT Do
        calculate P(τ_k^t);
    end For
    If ∏_T(1 − P(τ_k^t))/(1 − P(τ_x^t)) < p Then
        tempRD.Delete(t);                      // delete ID from RD set
    end If
end For
return S = ⟨tempRD + tempCD⟩;
```

Algorithm 1: Query processing on sensor node.

Definition 9 (irrelevant data). In the sensing data subset $U_i \subseteq U$ on sensor node, the tuples which are irrelevant data (ID) of the Probabilistic Skyline query satisfy the conditions:

$$\frac{\prod_T \left(1 - P\left(\tau_k^t\right)\right)}{1 - P\left(\tau_x^t\right)} < p \quad \left(\text{for any } \tau_x^t \in T\right). \tag{4}$$

Definition 10 (relevant data). In the sensing data subset $U_i \subseteq U$ on sensor node, the tuples which are relevant data (RD) of the Probabilistic Skyline query satisfy the conditions:

$$P(t) \times \prod \left(1 - P\left(\tau_k^t\right)\right) \le p \quad \left(\text{for any } \tau_x^t \in T\right), \quad \frac{\prod_T \left(1 - P\left(\tau_k^t\right)\right)}{1 - P\left(\tau_x^t\right)} \ge p. \tag{5}$$

Algorithm 1 sketches the process of data aggregation, data classification, and the ID filtering on sensor nodes. First, the algorithm merges all the data tuples sent by child nodes. In other words, it merges CD into candidate data set and merges RD into relevant data set (Lines 4–7); second, the algorithm adds the local data tuple to the candidate data set (Line 8); and, then, the skyline probability of each tuple in the candidate data set and relevant data set will be calculated. Meanwhile, the tuples will be classified according to the definitions to removing ID and signing RD and CD (Lines 9–33); in the end, the partial relevant data set and candidate data set will be submitted to the parent node (Line 34).

For data classification in a candidate data set, our algorithm works as follows: first, it initializes the cumulative probability variable (Line 10); second, the value of n is calculated, where n is the number of τ^t that can dominate the tuple t (Line 11); third, it finds out all τ^t that dominate t (Line 12), after which each τ^t's dominant probability is calculated (Lines 13–15). Then, the data tuples are classified based on the definitions above. In this procedure, tuples which are ID are deleted while tuples which are RD are transferred from the candidate data set to the relevant data set (Lines 16–22).

The process of data classification in a relevant data set is similar to the former. At first, the cumulative probability variable is initialized (Line 24); second, the value of n is calculated (Line 25); third, it finds out all τ^t that dominate t (Line 26); next, the dominant probability of each τ^t will be worked out (Lines 27–29); finally, the algorithm deletes t from the relevant data set if it is ID (Lines 30–33).

```
// input: The message set of child node M_S, the threshold value p.
// output: The data set S which will be submitted to the parent node
For each element m in M_S Do
   tempCD = tempCD + m.CD;
   tempRD = tempRD + m.RD;
end For
For each element t in tempCD Do
   temp = 1;
   n = t.getDominatingNumber(tempCD + tempRD);    // get the number n of τ^t
   tempT = t.getDominatingT(tempCD + tempRD);    // and all τ^t dominate t
   For each τ^t in tempT Do
      calculate P(τ_k^t);
   end For
   If ∏_T(1 − P(τ_k^t))/(1 − P(τ_x^t)) ≤ p Then
      tempCD.Delete(t);              // delete ID and RD from CD set
   end If
end For
return S = ⟨tempCD⟩;
```

Algorithm 2: Query processing on base station.

In consideration of the running example in Theorem 5, we assume that the WSN is a two-tier hierarchical topology network. Let tuples t_1, t_2, and t_5 in U_1 be collected by sensor nodes i. In the meantime, let t_3 and t_4 in U_2 be collected by sensor node j. According to Algorithm 1, we can firstly calculate the Local Skyline Probability (denoted as P_{sky_L}) of the tuples and then get the result that $P_{\text{sky}_L}(t_1) = 0.15$, $P_{\text{sky}_L}(t_2) = 0.3$, $P_{\text{sky}_L}(t_5) = 0.4$, $P_{\text{sky}_L}(t_3) = 0.2$, and $P_{\text{sky}_L}(t_4) = 0.64$. Thus, the data classification on node i is that t_1 is ID, t_2 is RD, and t_5 is CD. Similarly, t_3 is ID and t_4 is CD on node j. As a result, tuples t_2, t_5 on node i and t_4 on node j are transmitted to the base station.

The process of query processing on base station is described in detail in Algorithm 2. To begin with, the algorithm merges all the data tuples sent by child nodes; that is to say, it merges CD into the candidate data set and merges RD into the relevant data set (Lines 3–6); second, the skyline probability of each tuple in the candidate data set will be calculated; then, ID are removed from candidate data set (Lines 7–17); finally, the rest data tuples in candidate data set are the final result of PS (Line 18).

For removing ID and RD in a candidate data set, it first initializes the cumulative probability variable (Line 8); second, the value of n is calculated (Line 9); third, it finds out all τ^t that dominates t (Line 10); then, the dominant probability of each τ^t will be calculated (Lines 11–13); last, the tuple which is not CD is removed from the candidate set (Lines 14–17).

For example, on base station, the process of our running example above works as follows: first, tuples t_4 and t_5 are merged in candidate data set; t_2 is merged in RD. Second, we have $P_{\text{sky}_L}(t_4) = 0.24$ and $P_{\text{sky}_L}(t_5) = 0.4$. Third, delete t_4 from candidate data set. Finally, we get the last result that t_5 is the skyline result, which illustrates the correctness and feasibility of our algorithm.

5. Experimental Evaluations

In our experiments, n sensor nodes were generated randomly in a region with an area of n; thus, the average area of each node is 1. The communication radius between two nodes was set to be $2\sqrt{2}$, and the maximum packet transmitted between two nodes was stipulated to be 48 bytes. All the experiments were conducted on a computer with Intel Core i7-3770 CPU 3.40 GHz and 8.00 GB RAM. We conducted our evaluation on the standard test data sets of PS query, in which the probability for each tuple was generated uniformly. The performance of the algorithm is mainly studied on independence data and anticorrelated data.

Three parameters are mainly investigated in our experiments, which are the number of sensor nodes, the dimensions of sensing data, and the threshold value of the PS query. The algorithm adjusted the values of the parameters to minimize the overall data transmission in the network. The overall data transmission is calculated by the communication cost sent by all the sensor nodes in the network; that is, it is calculated by the dimensionality of sensing data × numbers × hop count. The communication costs of DPPS and CA were mainly explored with a number of sensor nodes which range from 600 to 1000, with the default number equaling 600. The dimensions of the sensing data range from 2 to 6 with the default dimension equaling 2. The threshold value of the PS query ranges from 0.1 to 0.3, which is 0.1 by default.

Under the independent and anticorrelation distribution, the data communication cost of DPPS and CA affected by the change of sensor nodes number is shown in Figure 3. In this figure, we found that a large number of sensor nodes lead to more communication cost. The increase speed of DPPS is slower than CA's. As the number of sensing data increases due to the more sensor nodes, the communication cost of CA increases fast. However, the unnecessary sensing

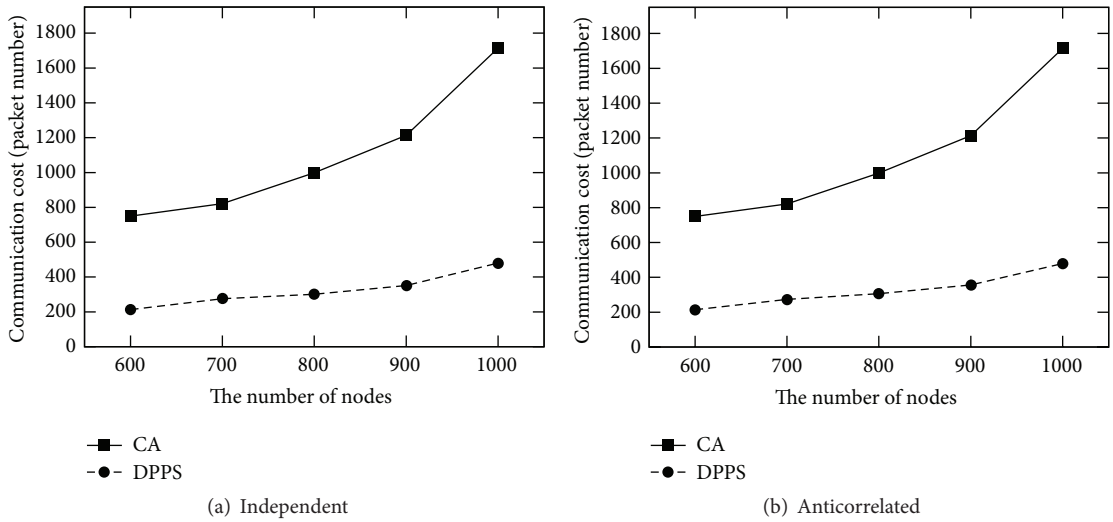


Figure 3: The communication cost influenced by nodes' number.

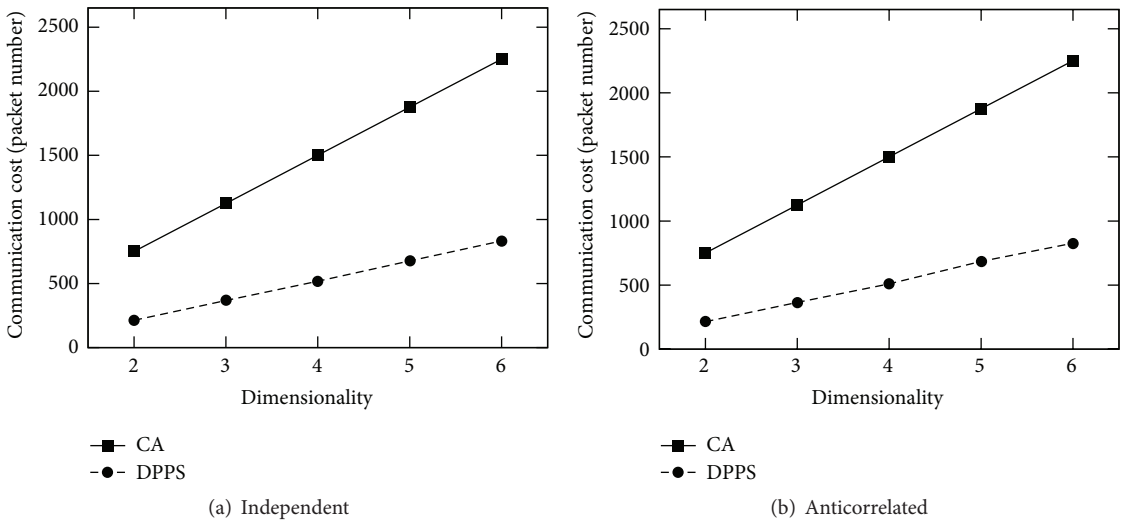


Figure 4: The communication cost influenced by data dimensions.

data are filtered in DPPS which directly leads to a less communication cost and a much slower rate of increasement. The communication cost in independent distribution is close to the one in anticorrelation distribution, which explains that data distribution has less impact on communication cost. In other words, the confidence of sensing data is the primary factor which affects the communication cost.

The data communication cost of both the algorithms, under the two kinds of data distribution, affected by the change of sensor data dimensionality is revealed in Figure 4. Obviously, the bigger the dimensionality is, the more the communication cost is. The reason is that, with the increment of data dimension, the probability of tuples dominated by others is decreased, which led to an increment in the number of skyline tuples and the data communication cost. The communication cost of DPPS is smaller than CA's, which further verified the effectiveness of DPPS. In addition, we can draw a conclusion that it is the confidence of sensing data which plays the primary role in communication cost affection.

Under the two different distributions, the data communication cost of DPPS and CA affected by the change of threshold value is shown in Figure 5. In the figure, we can see that a larger threshold value usually leads to less communication cost. It is intuitive, since the larger the threshold value is, the smaller the PS query result set will be. That actually results in a less communication cost. The communication cost of DPPS is always less than CA's, which proved the effectiveness of DPPS in a very great degree. In a similar way, the results demonstrated the confidence is the primary factor again.

All the results showed that DPPS precedes CA in all changes of sensor node number, the sensing data dimension, and the PS threshold value. It can be widely used in sensor

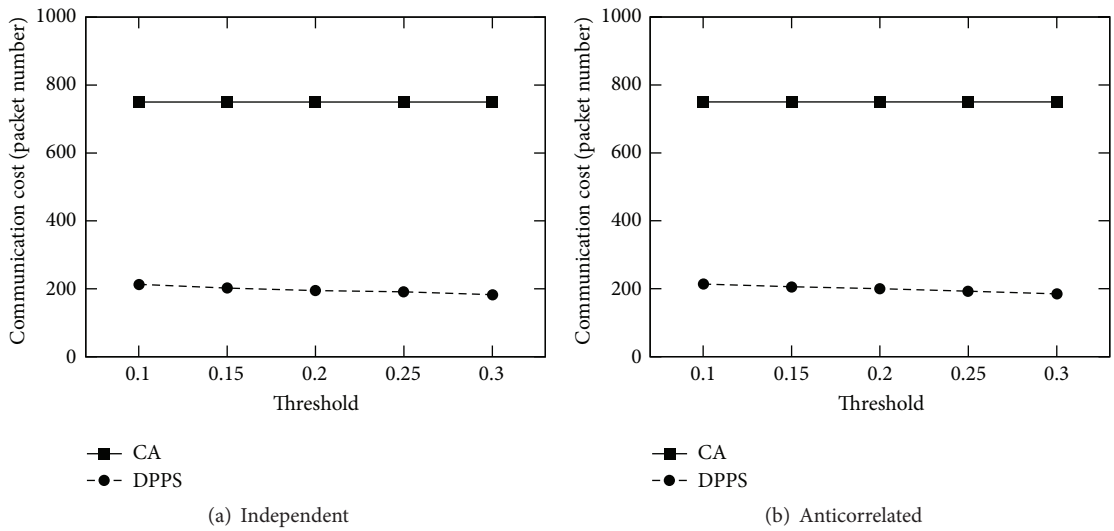


Figure 5: The communication cost influenced by threshold value.

networks since it can improve efficiency and reduce the communication cost significantly.

6. Conclusion

In this paper, we explored deeply the requirements of PS query algorithm in WSNs and summarized the existing problems in the WSNs. According to the characteristics of applications in WSNs, we firstly studied the basic properties of PS query and theoretically proved that the algorithm is not decomposable. Then, an efficient algorithm, Distributed Processing of Probabilistic Skyline (DPPS) query in WSNs, was put forward. DPPS can classify the sensing data on sensor nodes and discard the irrelevant data which will not affect the result of the PS query. Thereby, the DPPS can reduce the data transmission cost significantly in WSNs. Finally, the algorithm was verified by simulation experiments, and the results showed that the performance of DPPS compared with the CA is significantly improved in saving the communication cost in network.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China under Grant nos. 61402089, 61472069, and 61100022, the Natural Science Foundation of Liaoning Province under Grant no. 2015020553, and the Fundamental Research Funds for the Central Universities under Grant nos. N141904001 and N130404014.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[4] S. Börzsönyi, D. Kossmann, and K. Stocker, "The skyline operator," in *Proceedings of the 17th International Conference on Data Engineering (ICDE '01)*, pp. 421–430, April 2001.

[5] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "Progressive skyline computation in database systems," *ACM Transactions on Database Systems*, vol. 30, no. 1, pp. 41–82, 2005.

[6] L. Chen and X. Lian, "Efficient processing of metric skyline queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 3, pp. 351–365, 2009.

[7] Y. Tao and D. Papadias, "Maintaining sliding window skylines on data streams," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 3, pp. 377–391, 2006.

[8] M. Morse, J. M. Patel, and W. I. Grosky, "Efficient continuous skyline computation," *Information Sciences*, vol. 177, no. 17, pp. 3411–3437, 2007.

[9] G. Wang, J. Xin, L. Chen, and Y. Liu, "Energy-efficient reverse skyline query processing over wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 7, pp. 1259–1275, 2012.

[10] H. Chen, S. Zhou, and J. Guan, "Towards energy-efficient skyline monitoring in wireless sensor networks," in *Wireless Sensor Networks*, vol. 4373 of *Lecture Notes in Computer Science*, pp. 101–116, Springer, Berlin, Germany, 2007.

[11] J. Xin, G. Wang, L. Chen, X. Zhang, and Z. Wang, "Continuously maintaining sliding window skylines in a sensor network," in *Advances in Databases: Concepts, Systems and Applications: 12th International Conference on Database Systems for Advanced*

Applications, DASFAA 2007, Bangkok, Thailand, April 9–12, 2007. Proceedings, vol. 4443 of *Lecture Notes in Computer Science*, pp. 509–521, Springer, Berlin, Germany, 2007.

[12] J. Xin, G. Wang, and X. Zhang, "Energy-efficient Skyline queries over sensor network using mapped skyline filters," in *Advances in Data and Web Management: Joint 9th Asia-Pacific Web Conference, APWeb 2007, and 8th International Conference, on Web-Age Information Management, WAIM 2007, Huang Shan, China, June 16–18, 2007. Proceedings*, vol. 4505 of *Lecture Notes in Computer Science*, pp. 144–156, Springer, Berlin, Germany, 2007.

[13] B. Chen, W. Liang, and J. X. Yu, "Energy-efficient skyline query optimization in wireless sensor networks," *Wireless Networks*, vol. 18, no. 8, pp. 985–1004, 2012.

[14] H. Shen, Z. Chen, and X. Deng, "Location-based skyline queries in wireless sensor networks," in *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '09)*, pp. 391–395, IEEE, Hubei, China, April 2009.

[15] J. Xin, G. Wang, L. Chen, and V. Oria, "Energy-efficient evaluation of multiple Skyline queries over a wireless sensor network," in *Database Systems for Advanced Applications*, vol. 5463 of *Lecture Notes in Computer Science*, pp. 247–262, Springer, Berlin, Germany, 2009.

[16] Y. J. Roh, I. Song, J. H. Jeon, K. G. Woo, and M. H. Kim, "Energy-efficient two-dimensional skyline query processing in wireless sensor networks," in *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC '13)*, pp. 294–301, IEEE, Las Vegas, Nev, USA, January 2013.

[17] J. Pei, B. Jiang, X. Lin, and Y. Yuan, "Probabilistic skylines on uncertain data," in *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB '07)*, pp. 15–26, Vienna, Austria, September 2007.

[18] X. Ding, X. Lian, L. Chen, and H. Jin, "Continuous monitoring of skylines over uncertain data streams," *Information Sciences*, vol. 184, no. 1, pp. 196–214, 2012.

[19] W. Zhang, X. Lin, Y. Zhang, W. Wang, and J. X. Yu, "Probabilistic skyline operator over sliding windows," in *Proceedings of the 25th International Conference on Data Engineering (ICDE '09)*, pp. 1060–1071, IEEE, Shanghai, China, April 2009.

[20] M. J. Atallah and Y. Qi, "Computing all skyline probabilities for uncertain data," in *Proceedings of the 28th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '09)*, pp. 279–287, Providence, RI, USA, July 2009.

[21] C. Böhm, F. Fiedler, A. Oswald, C. Plant, and B. Wackersreuther, "Probabilistic skyline queries," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*, pp. 651–660, 2009.

[22] C. Böhm, A. Pryakhin, and M. Schubert, "The gauss-tree: efficient object identification in databases of probabilistic feature vectors," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, p. 9, Atlanta, Ga, USA, April 2006.

[23] X. Ding and H. Jin, "Efficient and progressive algorithms for distributed skyline queries over uncertain data," in *Proceedings of the 30th International Conference on Distributed Computing Systems (ICDCS '10)*, pp. 149–158, IEEE, Genova, Italy, June 2010.

[24] K. Hose and A. Vlachou, "A survey of skyline processing in highly distributed environments," *VLDB Journal*, vol. 21, no. 3, pp. 359–384, 2012.

[25] E. Dellis and B. Seeger, "Efficient computation of reverse skyline queries," in *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB '07)*, pp. 291–302, Vienna, Austria, September 2007.

[26] Q. Zhang, P. Ye, X. Lin, and Y. Zhang, "Skyline probability over uncertain preferences," in *Proceedings of the 16th International Conference on Extending Database Technology and the 16th International Conference on Database Theory (EDBT-ICDT '13)*, pp. 395–405, ACM, Genoa, Italy, March 2013.