INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

DISPENSATION OF PROBABILISTIC TOP-K QUERIES IN DISTRIBUTED WIRELESS SENSOR NETWORK

T.S.Venkatesh, E.Ravi Kumar, Dr.Sai Satyanarayana Reddy

[1]*M.Tech, CSE, LBRCE, Mylavaram, venkatesh.tadikonda6@gmail.com*
[2]*Associate Professor, CSE, LBRCE, Mylavaram, India, ravikumar.e@gmail.com*
[3] *Professor, CSE, LBRCE, Mylavaram, India, saisn90@gmail.com*

Abstract—In this paper, I introduce the notion of sufficient set and necessary set for distributed processing of probabilistic top-k queries in cluster-based wireless sensor networks. These two concepts have very nice properties that can facilitate localized data pruning in clusters. Accordingly, I develop a suite of algorithms, namely, sufficient set-based (SSB), necessary set-based (NSB), and boundary-based (BB), for intercluster query processing with bounded rounds of communications. Moreover, in responding to dynamic changes of data distribution in the network, I develop an adaptive algorithm that dynamically switches among the three proposed algorithms to minimize the transmission cost. I show the applicability of sufficient set and necessary set to wireless sensor networks with both two-tier hierarchical and tree-structured network topologies. Experimental results show that the proposed algorithms reduce data transmissions significantly and incur only small constant rounds of data communications. The experimental results also demonstrate the superiority of the adaptive algorithm, which achieves a near-optimal performance under various conditions.

1 INTRODUCTION

Wireless sensor networks are revolutionizing the ways to collect and use information from the physical world.
This new technology has resulted in significant impacts on a wide array of applications in various fields, including military, science, industry, commerce, transportation, and health-care. However, the quality of sensors varies significantly in terms of their sensing precision, accuracy, tolerance to hardware/external noise, and so on. For example, studies show that the distribution of noise varies widely in different photovoltaic sensors [1], precision and accuracy of readings usually vary significantly in humidity sensors [2], and the errors in GPS devices can be up to several meters [3]. Thus, sensor readings are inherently uncertain [4], [5], [6]. To facilitate management of uncertain data, researches on probabilistic databases have received renewed attentions in the past few years. Most of the recent works on probabilistic data modeling propose to associate a confidence (in form of probability) with a data record/tuple to capture the data uncertainty and thus carry a possible worlds semantic [7], [8], [9].1 Accordingly, system issues such as indexing techniques [10], [11] and query processing [8], [6], [12], [13], [14] have been examined. Nevertheless, they have mostly been studied under a centralized system setting. In this paper, I explore the problem of processing probabilistic top-k queries in distributed wireless sensor networks. Here, we first use an environmental monitoring application of wireless sensor network to introduce some basics of probabilistic databases. Running example. Consider a wireless sensor network that consists of a large number of sensor nodes

deployed in a geographical region. Feature readings (e.g., moisture levels or speed of wind gust) are collected from these distributed sensor nodes. Due to sensing imprecision and environmental interferences, the sensor readings are usually noisy. Thus, multiple sensors are deployed at certain zones in order to improve monitoring quality. In this network, sensor nodes are grouped into clusters, within each of which one of sensors is selected as the cluster head for performing localized data processing. By using statistic methods (e.g., [5]), a cluster head may generate a set of data tuples for each zone within its monitored region.2 In this example, we assume that each tuple is comprised of tuple id, zone, a derived possible attribute value, along i th a confidence that serves as a measurement of data uncertainty.

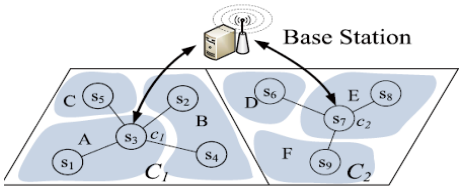

Fig 1. A wireless sensor network. There are six zones, denoted as A, B; . . . ; F, which are organized into two clusters C1 and C2, with the corresponding cluster heads c1 (i.e., s3) and c2 (i.e., s7).

Thus, the data tuples corresponding to the same zone collectively represent the probabilistic distribution of derived possible values for the zone. Since the existences of possible values in these tuples are exclusive to each other, they naturally form a logical tuple, called x-tuple.3 a wireless sensor network (with a two-tier hierarchical topology) that monitors the speed of wind gust in different zones. In this network, sensor nodes are grouped into clusters, where cluster heads are responsible for local processing and to report aggregated results to the base station. As shown, c1 and c2 denote the cluster heads for clusters C1 and C2.

2 PRELIMINARIES

In this section, we first define the PT-Top k query, introduce the centralized algorithms for its query processing, and review some related works.

2.1 System Model and Problem Definition

In this paper, we assume that a wireless sensor network consisting of a base station and N sensor nodes is deployed in a monitoring field. With a continuous power supply, the base station serves as the data collection and processing center to the external users and applications. I assume M clusters are formed and a cluster head is selected for each cluster. The whole sensor network can be logically treated as a distributed uncertain database. Sensor readings are collected by cluster heads and transformed into uncertain data, which collectively form as an uncertain table T.

Table T:

Tid	Loc.	Speed	Conf.	
t_1	A	89 km/h	0.9	T_1 (w.r.t. Cluster C_1)
t_2	B	83 km/h	0.1	
t_3	A	81 km/h	0.1	
t_4	B	80 km/h	0.45	
t_5	C	75 km/h	0.5	
t_6	D	99 km/h	0.1	T_2 (w.r.t. Cluster C_2)
t_7	E	94 km/h	0.5	
t_8	E	90 km/h	0.1	
t_9	F	85 km/h	0.2	

$\tau_A = \{t_1, t_3\}, \tau_B = \{t_2, t_4\}, \tau_C = \{t_5\}, \tau_D = \{t_6\}$
$\tau_E = \{t_7, t_8\}$ and $\tau_F = \{t_9\}$

Fig 2. A global view of wind gust records.

In other words, the content of T, i.e., a set of uncertain tuples (t1; t2; . . . ; tN), is distributed in cluster heads within the sensor network. To capture the data uncertainty, each tuple t 2 T is associated with a confidence P(t) > 0, i.e., the

uncertain table T carries a possible world semantics [7], [8], [9]. A set of x-relation rules are derived for tuples reflecting the same monitored phenomenon (e.g., wind speed in location A). Thus, only one tuple in the same x-relation rule may appear in a possible world. Note that, we assume sensor nodes within the same location are clustered together. Thus x-relation rules, which are derived by cluster heads, are only Applicable to tuples in the same cluster. In other words, tuples in the same x-relation rule are correlated and corresponding to the same location. We assume tuples with different x-relation rules are independent and corresponding to different locations. Moreover, we assume that each tuple in T involves in one and only one x-relation rule. An x-relation rule is specified in the form of T = (t1; t2; . . . ; tm), where m = |T |and $t_i \ (1 \leq i \leq m)$ are alternative tuples, e.g., TA= (t1; t3). The collection of alternative tuples in the same rule can be seen as one logical tuple and thus is called an x-tuple. Let Y (T) = (T1, T2…., Tg) (where g is the total number of x-tuples in T) be the set of x-relation rules specified on T. The aggregate confidence of an x-tuple _ is the sum of the confidence values of all its alternative tuples, i.e.,

$$P(\tau) = \sum_{t \in \tau} P(t).$$

Let W denote a possible world which consists of a subset of tuples in T and w denote the set of all possible worlds. The probability that $W \in \mathcal{W}$ exists is

$$P(W) = \prod_{(\tau \in \Upsilon(T)) \wedge (\tau \cap W = \{t\})} P(t) \prod_{(\tau \in \Upsilon(T)) \wedge (\tau \cap W = \emptyset)} (1 - P(\tau)).$$

Since at most one tuple of each x-tuple exists in any W 2 W and tuples belonging to different x-tuples are independent of each other, the first and second terms above capture the probabilities of the x-tuples that do and do not appear in$W \in \mathcal{W}$, respectively. As mentioned, I use PT-Top k [13] as a test case for our proposed ideas; we first define the top-k probability of a tuple and then define the PT-Top k query.

Definition 1 (Top-k Probability)

Let AW denote the top-k answer set in a possible world W. Given a tuple $t \in T$, the top k probability of t is the aggregate probability of t being in the top-k answers over all$W \in \mathcal{W}$, i.e,

$$P_{topk}(t) = \sum_{W \in \mathcal{W}, t \in A_W} P(W).$$

For example, consider a PT-Top k query with k = 2 and p = 0:5 on data set T1 in Fig. 2. I can calculate Ptop(t1) = P(W1) + P(W2) + P(W3) + P(W4) + P(W5) = 0:9 and Ptop k(t5) = P(W6) + P(W12) = 0:225.
Definition 2 (Probabilistic Threshold Top-k Query (PT-Top k) [13]).

Given a probability threshold p (0 < p <_ 1), PT-Top k finds the set of tuples whose top-k probabilities are at least p.

2.2CentralizedPT-Topk Query Processing

In this section, I present a general approach for processing PT-Top k queries in a centralized uncertain database, which provides a good background for the targeted distributed processing problem. Given an uncertain table T, we first sort T in accordance with the ranking function f such that
$t_1 \prec_f t_2 \prec_f \cdots \prec_f t_n$. .

The query answer can be obtained by examining the tuples in descending ranking order from the sorted table (which is still denoted as T for simplicity). I can easily determine that the highest ranked k tuples are definitely in the answer set as long as their confidences are greater than p since their qualifications as PT-Top k answers are not dependent on the existence of any other tuples. Nevertheless, the qualification of tuple tkþ1 as a PT-Top k answer is dependent on 1) whether there exist some possible worlds where the tuples in front of t k+1 belong to less than k x-tuples; and 2) whether the aggregated confidence of t k+1 over these possible worlds are greater than p. If the answer is positive, then tuple t k+1 is included in the answer set and tuple t k+2 is examined. This process continues until there are no more qualified tuples left to be examined. To shed more light on probabilistic PT-Top k query

processing, let's consider the top-k probability for tuple ti , where i > k. Based on (1), Ptop k(ti) is the sum of the probabilities that ti ranks higher than k in a possible world. In other words, the top-k probability of ti is the sum of possible world probabilities with less than k tuples ranking higher than ti.

Let r i-1;[T]denote a probability of possible worlds in which l of the highest i -1 tuples in T (i.e., those placed in front of ti) do exist. I use r i-1,l for simplicity when it causes no confusion. The top-k probability of ti is as follows:

$$P_{topk}(t_i) = P(t_i) \cdot \sum_{l=0}^{k-1} r_{i-1,l}[T].$$

The above computation involves only the first i - 1

tuples in T except for those in the same x-relation with ti. Thus, I use D ti to denote {tx|tx $\in$ T; 1 < <_ x< _ i- 1 – {tx|ti $\in$ T ∧ tx$\in$ T}.

$$P_{topk}(t_i) = P(t_i) \cdot \sum_{l=0}^{k-1} r_{i-1,l}[D_{t_i}].$$

2.3 RELATED WORK

Here, I review representative work in the areas of 1) top-k query processing in wireless sensor networks, and 2) top-k query processing on uncertain data.

Top-k query processing in sensor networks. An extensive number of research works in this area has appeared in the literature [21], [24], [25], [26]). Due to the limited energy budget available at sensor nodes, the primary issue is how to develop energy-efficient techniques to reduce communication and energy costs in the networks. TAG [21] is one of the first studies in this area. By exploring the semantics of aggregate operators (e.g., sum, avg, and top-k),
In-network processing approach is adopted to suppress redundant data transmissions in wireless sensor networks. Approximate-based data aggregation techniques have also been proposed [27], [25]. The idea is to tradeoff some data quality for improved energy efficiency. Silberstein et al. develop a sampling-based approach to evaluate approximate top-k queries in wireless sensor networks [26]. Based on statistical modeling techniques, a model-driven approach was proposed in [5] to balance the confidence of the query answer against the communication cost in the network. Moreover, continuous top-k queries for sensor networks have been studied in [28] and [29]. In addition, a distributed threshold join algorithm has been developed for top-k queries [24]. These studies, considering no uncertain data, have a different focus from our study.

Top-k query processing on uncertain data. While research works on conventional top-k queries are mostly based on some deterministic scoring functions, the new factor of tuple membership probability in uncertain databases makes evaluation of probabilistic top-k queries very complicated since the top-k answer set depends not only on the ranking scores of candidate tuples but also their probabilities [8]. For uncertain databases, two interesting top-k definitions (i.e., U-Top k and U-k Ranks) and A*-like algorithms are proposed [17]. U-Top k returns a list of k tuples that has the highest probability to be in the top-k list over all possible worlds. U-k Ranks returns a list of k tuples such that the I th record has the highest probability to be the i th best record in all possible worlds.

In [13], PT-Top k query, which returns the set of tuples with a probability of at least p to be in the top-k lists in the possible worlds, is studied. Inspired by the concept of dominate set in the top-k query, an algorithm which avoids unfolding all possible worlds is given.

3 INTRA CLUSTER DATA PRUNING

In a cluster-based wireless sensor network, the cluster heads are responsible for generating uncertain data tuples from the collected raw sensor readings within their clusters. To answer a query, it's natural for the cluster heads to

prune redundant uncertain data tuples before delivery to the base station in order to reduce communication and energy cost. The key issue here is how to derive a compact set of tuples essential for the base station to answer the probabilistic top-k queries. This is a very challenging issue for the following reasons: 1) the interplay of probability and ranking due to the semantic of probabilistic top-k queries; and 2) the lack of global knowledge to determine the probability and ranking of candidate tuples locally at cluster heads. In this section, we propose the notion of sufficient set and necessary set, and describe how to identify them from local data sets at cluster heads. Next, we use the PT-Top k query as a test case to derive sufficient set and necessary set and show that the top-k probability of a tuple t obtained locally is an upper bound of its true top-k probability. Thus, data tuples excluded from the sufficient sets and necessary sets in local clusters will never appear in the final answer set.

3.1 Definition of Sufficient and Necessary Sets

It would be beneficial if cluster heads are able to find the minimum sets of their local data tuples that are sufficient for the base station to answer a given query. Ideally, sufficient set is a subset of the local data set. Data excluded from the sufficient set, no matter which clusters they reside, will never be included in the final answer set nor involved

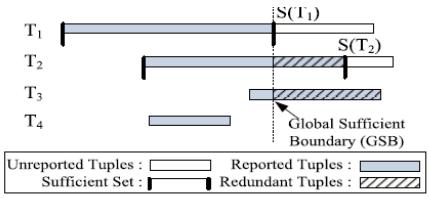

Fig 3. Data distribution over the wireless sensor network

in the computation of the final answer set. Here, we define the sufficient set more formally as follows:

Definition 3 (Sufficient Set).

Given an uncertain data set T_i in cluster C_i, if there exists a tuple $t_{sb} \in T_i$ (called sufficient boundary) such that the tuples ranked lower than t_{sb} are useless for query processing at the base station, then the sufficient set of T_i, denoted as $S(T_i)$, is a subset of T_i as specified below:

$$S(T_i) = \{t | t =_f t_{sb} \text{ort} \prec_f t_{sb}\},$$

Where f is a given scoring function or ranking

Note that a sufficient boundary may not exist for a given data set; then I consider the whole data set as a sufficient set and will discuss it in more detail later.

Theorem 1. Given M clusters in a sensor network, collecting $S(T_i)$ or T_i (when $S(T_i)$ does not exist for C_i) from clusters C_i where i = 1..M is sufficient for the base station to answer a query.

Proof. The proof is sketched below. Meanwhile, I use Fig.3 for illustration. Suppose there are $M (= M_1 + M_2)$ clusters within the wireless sensor network, where M_1 clusters have sufficient sets, while M_2 clusters do not have one (e.g., $M_1 = 2; M_2 = 2$ in Fig. 3). I call the highest ranked sufficient boundary among all clusters the global sufficient boundary (GSB). Based on Definition 3, any tuple ranked lower than GSB is not in the final answer. Thus, only tuples ranked higher than GSB are of interest to the base station. We consider three cases to prove that all the tuples ranked higher than GSB are in the base station: 1) the clusters without sufficient sets (e.g., T_3 and T_4 in Fig. 4)—all tuples are transmitted; 2) the cluster holding the highest sufficient boundary GSB (i.e., $S(T_1)$)—we can simply agree that all the necessary data tuples are sent to the base station; and 3) any cluster holding a sufficient boundary lower than GSB (i.e., T_2)—all the tuples ranked higher than GSB are safely contained within their sufficient sets and delivered to the base station. Hence, it is sufficient for cluster heads $c_i (i = 1..M)$ to deliver $S(T_i)$ or T_i (if $S(T_i)$ does not exist) for top-k query processing.

Definition 4 (Necessary Set). Given a local data set T_i in cluster C_i, assume that A_i is the set of locally known candidate tuples for the final answer and t_{nb} (called necessary boundary) is the lowest ranked tuple Z in A_i.

The necessary set of T_i, denoted as $N(T_i)$, is

$$N(T_i) = \{t | t \in T_i, t \prec_f t_{nb}\}.$$

Note that, at the base station, for a given cluster, the probability computation of some candidate tuples from other clusters may still require data tuples outside its necessary set. In other words, while the data tuples in the necessary sets include the final answer set, they may not be sufficient to determine the final answer set. In the following theorem, we depict a relationship between the sufficient set and necessary set.

Theorem 2. Given a data set T_i in cluster C_i where the sufficient set exists, the necessary set $N(T_i)$ is a subset of the sufficient set $S(T_i)$, i.e.,

$$N(T_i) \subseteq S(T_i).$$

Proof. I prove the theorem by contradiction. Suppose we have a necessary set denoted by $N(T_i)$ which is a proper superset of the sufficient set $S(T_i)$

$$(\text{i.e., } N(T_i) \supset S(T_i)).$$

Let the sufficient boundary be $SB(T_i)$ and the necessary boundary be $NB(T_i)$, then $SB(T_i)$ $NB(T_i)$. According to Definition 3, we know that any tuples ranked lower than $SB(T_i)$ can be safely pruned from transmission. On the other hand, if $NB(T_i)$ can be pruned, it cannot be part of the necessary set $N(T_i)$. This contracts our assumption. Therefore, $NB(T_i)$. Ranks higher than $SB(T_i)$. According to the definitions of sufficient set and necessary set, we conclude that the necessary set must be a subset of the sufficient set, if the sufficient set exists.

4 INTERCLUSTER QUERY PROCESSING

In this section, using the notion of sufficient and necessary sets as a basis, we propose three distributed algorithms for processing probabilistic top-k queries in wireless sensor networks, namely 1) Sufficient Set-based method; 2) Necessary Set-based method; and 3) Boundary-based method. Here we focus on addressing the communication overhead which is critical for wireless sensor networks and their applications. For simplicity, we logically assume single-hop transmission in both intracluster and intercluster communications. Nevertheless, our algorithms are not restricted to this assumption and can be extended for the multihop communications. As long as the base station receives all the candidate data tuples and supplementary tuples, we are able to compute the final answer with a generic centralized algorithm, e.g., those in [17], [13], [18], and [19]. Note that the supplementary tuples refer to the unqualified data tuples needed for computing the confidence probability of final answer.

4.1 SUFFICIENT SET-BASED ALGORITHM

An intuitive way for in-network data processing is toTransmit the sufficient set to base station. As indicated in Theorem 1, the tuples not included in the sufficient set neither have top-k probability higher than p nor affect the top-k probability of qualified tuples in the final answer. Thus, they are subject to pruning. Consequently, the SSB algorithm, as illustrated, consists of only one communication phase from cluster heads to the base station. After collecting data tuples from its cluster, a cluster head computes the sufficient set from the local collected tuples and sends it to the base station. Note that if a sufficient set cannot be obtained, all the local data tuples are transmitted. After receiving the transmitted data tuples from all the cluster heads, the base station computes the query answer by a centralized algorithm. Detailed SSB algorithm is presented in Algorithm 1.

Algorithm 1. SSB ALGORITHM

1: /* Cluster Head ci Side */
2: Compute the sufficient boundary SB(Ti) of Ti
3: if SB(Ti) exists then
4: $S(T_i) \leftarrow \{t | t \preceq_f SB(T_i) \wedge t \in T_i\}$
5: $T_i' \leftarrow S(T_i)$

6: else
7: $T_i' \leftarrow T_i$
8: end if
9: Deliver T_i' to the base station

1: /* Base Station Side */
2: Collect T0 i from all c_i $(1 \leq i \leq M)$
3: $T' \leftarrow \cup_{1 \leq i \leq M} T_i'$
4: Execute centralized algorithm over T'

4.2 NECESSARY SET-BASED ALGORITHM

The necessary set contains only 1) locally qualified tuples that have local top-k probability higher than p and 2) supplementary tuples ranked higher than those in (1) (Because they are needed to compute top-k probabilities of these qualified tuples). However, even though all the tuples that potentially have top-k probabilities higher than p are included in the necessary set, calculating their global top-k probabilities may still need to access some additional supplementary tuples.

Therefore, NSB may have two phases when these additional supplementary tuples are needed. After receiving all the necessary sets, the base station merges all the received tuples into a table $T' = \cup_{i=1}^{M} N(T_i)$, e, and finds the necessary boundary of T' (i.e., $NB(T')$ —called the global boundary (GB)). Consider T0 as a location data set. Based on Lemma 1, any tuple ranked lower than GB is not part of the final result. If GB is ranked higher than the highest ranked necessary boundary (i.e., NBhighest), we can conclude that all the necessary data have been delivered to the base station; thus, the base station computes the final answer and stop here. Otherwise, entering the second phase, the base station sends the GB back to the cluster heads, which return the supplementary data tuples ranked between its local necessary boundary and GB. Eventually, the base station computes the final answer. Algorithms running in cluster head ci and the base station are
shown in Algorithm 2.

Algorithm 2. NSB ALGORITHM

1: /* Cluster Head ci Side */
2: Compute the necessary boundary NB(Ti) of Ti
3: $N(T_i) \leftarrow \{t | t \preceq_f NB(T_i) \wedge t \in T_i\}$
4: Deliver N(Ti) to the base station
5: if Receive GB from the base station than
6: $N'(T_i) \leftarrow \{t | t \preceq_f \text{GB} \wedge t \in [T_i - N(T_i)]\}$
7: Send $N'(T_i)$ to the base station
8: end if

1: /* Base Station Side */
2: Collect N(Ti)from all $c_i (1 \leq i \leq M)$
3: $T' \leftarrow \cup_{1 \leq i \leq M} N(T_i)$
4: Compute the global necessary boundary (GB)
Over T'
5: if $\text{GB} \preceq_f NB(T_i)$, where $\forall i$ of $1 \leq i \leq M$ than
6: Execute centralized algorithm over T'
7: else
8: Broadcast GB to cluster heads
9: Once collect all N0ðTiÞ from all ci
10: $T' \leftarrow T' \cup_{1 \leq i \leq M} N'(T_i)$
11: Execute centralized algorithm over T'

12: end if

4.3 BOUNDARY-BASED ALGORITHM

Instead of directly delivering data tuples to the base station, the boundary-based method first delivers the local knowledge in clusters, in the form of sufficient boundary and necessary boundary, to the base station in order to facilitate a refined global data pruning among clusters later. As shown, each cluster head First computes its sufficient and necessary sets and sends the boundaries to the base station. After receiving all the sufficient and necessary boundaries, the base station computes the global boundary as follows: Let SBhighest denote the highest ranked sufficient boundary and NBlowest denote the lowest ranked necessary boundary. Based on the property of sufficient boundary, we know that any tuple ranked lower than SBhighest is not required for query processing. Meanwhile, based on the property of necessary boundary, we know that all tuples ranked higher than NBlowest may be needed for query processing. Therefore, we return the higher boundary between SBhighest and NBlowest as GB to the cluster heads. In the second phase, all the data tuples ranked higher than GB are transmitted to the base station, which runs a centralized PT-top k algorithm to compute the final answer. Detailed algorithm is presented in Algorithm 3. Its correctness is shown by Theorem 5.

Algorithm 3. BB ALGORITHM

1: /* Cluster Head ci Side */
2: Compute NB(Ti) and SB(Ti) of Ti
3: Send NB(Ti) and SB(Ti) to the base station
4: Receive GB from the base station
5: $T_i' \leftarrow \{t | (t \preceq_f GB) \wedge (t \in T_i)\}|$
6: Deliver T' i to the base station.

5 EXTENSIONS FOR TREE-STRUCTURED NETWORK TOPOLOGY

To perform in-network query processing, a routing tree is often formed among sensor nodes and the base station. A query is issued at the root of the routing tree (i.e., the base station) and propagated along the tree to all sensor nodes. Although the concepts of sufficient set and necessary set introduced earlier are based on two-tier hierarchical sensor networks, they are applicable to tree-structured sensor network. Here, we devise a family of algorithms, namely SSB-T, NSB-T, and NSB-T-Opt, for efficient probabilistic top-k query processing in a tree-structured sensor network Similarly, we assume tuples belonging to the same x-tuple are obtained in a single sensor node.
SSB-T. Similar to SSB, SSB-T is a one-round query processing algorithm.

The basic idea is to deliver the sufficient sets from leaf nodes, along the routing path, back to the root node. More specifically, after the query message is propagated to all sensor nodes, sensor nodes respond as follows: for a leaf sensor node ni, it computes the sufficient set based on Di, denoted as Si, for delivery to its parent. On the other hand, for a non leaf node nj, it collects sufficient sets sent by its children , which along with Dj are used to determine the sufficient set Si for delivery to its parent. When the base station receives all sufficient sets sent by its children. It directly deduces the probabilistic top-k result by running a centralized algorithm. Similar to SSB, the correctness of SSB-T can be proved. NSB-T. NSB-T is a two-round query processing algorithm, where in the first round necessary sets are collected along the routing tree, from the leaf nodes to the root (i.e., base station). In the second round, the necessary boundary derived at the root is returned to all sensor nodes to obtain additional supplementary tuples for computing the final query result.

In the first round, after the query message is propagated to all sensor nodes, NSB-T algorithm runs as follows: for a leaf node ni, it computes the necessary set based on Di, denoted as Ni, for delivery to its parent. On the other hand, for a non leaf node nj, it collects the necessary sets sent by its children (i.e., [nx2childðnjÞNx), which along with Dj are used to determine the necessary set Nj for delivery to its parent. When the base station receives all necessary sets sent by its children, it calculates a global necessary set, and sends back the global necessary boundary (GNB) to all sensor nodes for acquiring supplementary tuples as needed in the second round. After all supplementary data is

collected; the base station deduces the probabilistic top-k result by running a centralized algorithm. Similar to NSB, the correctness of NSB-T can be proved.

6 CONCLUSIONS

In this paper, I propose the notion of sufficient set and necessary set for efficient in-network pruning of distributed uncertain data in probabilistic top-k query processing. Accordingly, we systematically derive sufficient and necessary boundaries and propose a suite of algorithms, namely SSB, NSB, and BB algorithms, for in-network processing of PT-Top k queries. Additionally, we derive a cost model on communication cost of the three proposed algorithms and propose a cost-based adaptive algorithm that adapts to the application dynamics. Although our work in this paper is based mainly under the setting of two-tier hierarchical network, the concepts of sufficient set and necessary set are universal and can be easily extend to a network with tree topology. The performance evaluation validates our ideas and shows that the proposed algorithms reduce data transmissions significantly. While focusing on PT-Top k query in this paper, the developed concepts can be applied to other top-k query variants. We plan to develop algorithms to support other probabilistic top-k queries in the future.

REFERENCES

[1] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak, "A Collaborative Approach to in-Place Sensor Calibration," Proc. Second Int'l Conf. Information Processing in Sensor Networks (IPSN),
pp. 301-316, 2003.

[2] http://www.veriteq.com/, 2012.

[3]http://www.dashdyno.net/tech/GPS/gps_location.html, 2012.

[4] E. Elnahrawy and B. Nath, "Poster Abstract: Online Data Cleaning in Wireless Sensor Networks," Proc. First Int'l Conf. Embedded Networked Sensor Systems (SenSys '03), pp. 294-295, 2003.

[5] A. Deshpande, C. Guestrin, S.R. Madden, J.M. Hellerstein, and W. Hong, "Model-Driven Data Acquisition in Sensor Networks," Proc. 13th Int'l Conf. Very Large Data Bases (VLDB '04), pp. 588-599, 2004.

[6] R. Cheng, D.V. Kalashnikov, and S. Prabhakar, "Evaluating Probabilistic Queries over Imprecise Data," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '03), pp. 551-562, 2003.

[7] S. Abiteboul, P. Kanellakis, and G. Grahne, "On the Representation and Querying of Sets of Possible Worlds," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '87), pp. 34- 48, 1987.

[8] N. Dalvi and D. Suciu, "Efficient Query Evaluation on Probabilistic Databases," Proc. 30th Int'l Conf. Very Large Data Bases (VLDB '04), pp. 864-875, 2004.

[9] A.D. Sarma, O. Benjelloun, A. Halevy, and J. Widom, "Working Models for Uncertain Data," Proc. 22nd Int'l Conf. Data Eng. (ICDE '06), p. 7, 2006.

[10] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J.S. Vitter, "Efficient Indexing Methods for Probabilistic Threshold Queries over Uncertain Data," Proc. 30th Int'l Conf. Very Large Data Bases (VLDB '04), pp. 876-887, 2004.

[11] Y. Tao, R. Cheng, X. Xiao, W.K. Ngai, B. Kao, and S. Prabhakar, "Indexing Multi-Dimensional Uncertain Data with Arbitrary Probability Density Functions," Proc. 31st Int'l Conf. Very Large Data Bases (VLDB '05), pp. 922-933, 2005.

[12] C. Re, N. Dalvi, and D. Suciu, "Efficient Top-k Query Evaluation on Probabilistic Data," Proc. Int'l Conf. Data Eng. (ICDE '07), pp. 896-905, 2007.

[13] M. Hua, J. Pei, W. Zhang, and X. Lin, "Ranking Queries on Uncertain Data: A Probabilistic Threshold Approach," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '08), 2008.

[14] F. Li, K. Yi, and J. Jestes, "Ranking Distributed Probabilistic Data," Proc. 35th SIGMOD Int'l Conf. Management of Data (SIGMOD '09), 2009.

[15] H.W. Rabiner, C. Anantha, and B. Hari, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," Proc. 33rd Hawaii Int'l Conf. System Sciences (HICSS '00), 2000.

[16] Y. Diao, D. Ganesan, G. Mathur, and P.J. Shenoy, "Rethinking Data Management for Storage-Centric Sensor Networks," Proc. Conf. Innovative Data Systems Research (CIDR '07), pp. 22-31, 2007.

[17] M.A. Soliman, I.F. Ilyas, and K.C. Chang, "Top-k Query Processing in Uncertain Databases," Proc. Int'l Conf. Data Eng. (ICDE '07), 2007.

[18] C. Jin, K. Yi, L. Chen, J.X. Yu, and X. Lin, "Sliding-Window Top-k Queries on Uncertain Streams," Proc. Int'l Conf. Very Large Data
Bases (VLDB '08), 2008.

[19] G. Cormode, F. Li, and K. Yi, "Semantics of Ranking Queries for Probabilistic Data and Expected Ranks," Proc. IEEE Int'l Conf. Data Eng. (ICDE '09), 2009.

[20] P. Cao and Z. Wang, "Efficient Top-k Query Calculation in Distributed Networks," Proc. 23rd Ann. ACM Symp. Principles of Distributed Computing (PODC), pp. 206-215, 2004
.
[21] S. Madden, M.J. Franklin, J. Hellerstein, and W. Hong, "TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks," Proc. Fifth Symp. Operating Systems Design and Implementation (OSDI '02),
2002.

[22] Y. Xu, W.-C. Lee, J. Xu, and G. Mitchell, "Processing Window Queries in Wireless Sensor Networks," Proc. IEEE 22nd Int'l Conf. Data Eng. (ICDE '06), 2006.

[23] M. Ye, X. Liu, W.-C. Lee, and D.L. Lee, "Probabilistic Top-k Query Processing in Distributed Sensor Networks," Proc. IEEE Int'l Conf. Data Eng. (ICDE '10), 2010.

[24] D. Zeinalipour-Yazti, Z. Vagena, D. Gunopulos, V. Kalogeraki, V. Tsotras, M. Vlachos, N. Koudas, and D. Srivastava, "The Threshold Join Algorithm for Top-k Queries in Distributed Sensor Networks," Proc. Second Int'l Workshop Data Management for Sensor Networks (DMSN '05), pp. 61-66, 2005.

[25] A. Sharaf, J. Beaver, A. Labrinidis, and K. Chrysanthis, "Balancing
Energy Efficiency and Quality of Aggregate Data in Sensor
Networks," Int'l J. Very Large Data Bases, vol. 13, no. 4, pp. 384-403,
2004.