Indexing Uncertain Categorical Data over Distributed Environment

Adel Benaissa, Salima Benbernou, Mourad Ouziri, Soror Sahri

Indexing Uncertain Categorical Data over Distributed Environment

Adel Benaissa, Salima Benbernou, Mourad Ouziri and Soror Sahri

Universite Paris Descartes, France

Abstract

Today, a large amount of uncertain data is produced by several applications where the management systems of traditional databases including indexing methods are not suitable to handle such type of data. In this paper, we propose an inverted based index method for efficiently searching uncertain categorical data over distributed environments. We address two kinds of query over the distributed uncertain databases, one a distributed probabilistic thresholds query, where all results satisfying the query with probabilities that meet a probabilistic threshold requirement are returned, and another a distributed top k-queries, where all results optimizing the transfer of the tuples and the time treatment are returned.

Keywords: Uncertain database, indexating, distributed environment, top-k query, query optimization, threshold query.

1. Introduction

Nowadays, real world applications producing large amounts of uncertain data have increasingly flourished. For instance, data collected from sensor network, web extraction, data integration, data cleaning [1, 2, 3, 4]. The exact value of the data is often unknown, but may be selected from a reasonable number of alternatives. Hence, due to the importance of uncertain data for those large number of applications, there has been significant recent interest in database support for uncertain data. However, Current database management systems do not provide a convenient means for managing uncertain data with traditional indexing structure. Consequently, existing work in this area provides new models for uncertain data, prototype implementations, and efficient semantic query processing algorithms. Possible world has been largely adapted to model semantically uncertain databases, where each possible world corresponds to a single deterministic data instance [2]. Moreover, Large body of those works has considered the uncertainty of data in database in two ways, *uncertain tuple* and *uncertain attribute*. For uncertain attribute [1], the value of a tuple is not known precisely and is modeled as a set or range of possible values with associated probabilities. For uncertain tuple, the presence of an entire tuple within a relation is uncertain and a probability is attached to the tuple. Many cases where uncertainty arises are distributed in nature, e.g., distributed sensor networks, multiple data sources for information integration. However, very little work has addressed the indexation of distributed uncertain data. One can cite [5], it is proposed a novel approach of processing uncertain top-k query in large-scale P2P networks, [6] addresses the problem of entity resolution over probabilistic data. In this paper, we address the problem of indexing uncertain categorical data over distributed database. We assume there exits an uncertainty in an attribute. We propose an approach using (1) a local index on each site based on the inverted index introduced first in [7], (2) a global index that summarizes each distributed uncertain database (each site). We show that the global index supports some types of queries including probabilistic threshold query and top-k query. The proposed global index uses a pruning algorithm in order to reduce the communication and optimizes the processing time. A fundamental problem we address is to retrieve the global top-k tuples from all distributed nodes with minimum communication cost.

The key contributions of this paper are as follows:

- We propose a distributed index structure for uncertain data over distributed database. We suggest an inverted based local index and a global index summarizing each distributed sites.
- We propose pruning algorithms to answer queries using the proposed index. We propose two types of queries to be processed over the global index including probabilistic threshold query and probabilistic top-k query.
- We propose methods to reduce the communication cost between distributed sites when querying uncertain data over distributed sites.

1.1. Motivation

The sector of bovine breeding knows colossal losses due to the transmission of viral diseases affecting the cows on the farms. The breeders as well as the insurers are concerned with such losses. To minimize the losses generated by viral diseases, the protector service of bovine breeding puts online consultations for the veterinary surgeons based, on one hand on the symptoms described by the breeders which are uncertain and in the other hand on the diagnosis provided by the veterinary surgeons them-

T_{id}	weight	illness
$T_{1,1}$	700	$\{(da, 0.7); ds(0.3)\}$
$T_{1,2}$	710	$\{(da, 0.8); ds(0.2)\}$
$T_{1,3}$	790	$\{(nc, 1)\}$
$T_{1,4}$	725	$\{(nc, 0.9); (ds, 0.1)\}$

T_{id}	weight	illness
$T_{2,1}$	700	$\{(da, 0.2); ds(0.8)\}$
$T_{2,2}$	710	$\{(da, 0.9); ds(0.1)\}$
$T_{2,3}$	790	$\{(nc, 0.85); (ds, 0.15)\}$
$T_{2,4}$	725	$\{(nc, 0.9); (ds, 0.1)\}$

T_{id}	weight	illness
$T_{3,1}$	749	$\{(mc, 0.8); nc(0.2)\}$
$T_{3,2}$	645	$\{(mc, 1\}$
$T_{3,3}$	801	$\{(nc, 1)\}$
$T_{3,4}$	799	$\{(nc, 0.5); (mc, 0.5)\}$

T_{id}	weight	illness
$T_{4,1}$	711	$\{(mc, 0.18); nc(0.82)\}$
$T_{4,2}$	745	$\{(nc, 0.9); (mc, 0.1)\}$
$T_{4,3}$	901	$\{(nc, 0.85); (mc, 0.15)\}$
$T_{4,4}$	799	$\{(nc, 0.95); (mc, 0.05)\}$

Table 1: exemple of distributed uncertain databases

selves during their visits to the farms. For instance Table 1 gives more than three distributed sites (farms) with an uncertain attribute "Illness". $T_{1,1}$ means that the farms 1 has a cow with identity $T_{1,1}$ with weight 700 and its illness may be "acute diarrhoea (da)" with probability 0.5 and "simple diarrhoea (ds)" with probability 0.5. $T_{2,1}$ means that the farms 2 has a cow with identity $T_{2,1}$ with weight 719 and its illness may be "mad cow (mc)" with probability 0.4 and "normal cow (nc)" with probability 0.6.

The final report of diagnosis made by the veterinary surgeons for every farm, will provide which farm is affected by the disease. The aim is not to query the whole distributed farms, but only those affected by the disease The service wishes to locate which farm is affected by the disease in order to make prevention. The purpose is to provide an approach of uncertain interrogations of uncertain data over distributed farms.

The rest of the paper is organized as follows: section 2 presents the problem definition. Section 3 discusses the proposed framework to model the distributed indexing. Section 4 provides the query types to be processed on the distributed index. The related work is presented in section 5, and finally conclude the paper in section 6.

2. Problem Definition

Uncertainty in the database can be modeled by two viewpoints, the first one affects a probability value to each tuple called (tuple uncertainty) and the second affects a probability of one attribute. For the sake of simplicity, in this paper, we limit our study to uncertain database with a single uncertain attribute that are drawn from categorical domains over distributed environment.

2.1. Distributed environment

Given n distributed sites $S = \{S_1, S_2, ..., S_n\}$, each site holds a set of an uncertain relation $R = \{R_1, R_2, ..., R_n\}$. We assume that the R_i have the same schema in each $S_i, 1 < 1 \leq n$.

2.2. Categorical domain

Following the definition in [7], $R_i.a$ is a particular attribute in relation R_i which is uncertain. $R_i.a$ takes values from the categorical domain D with cardinality $|D| = N_i$. For a traditional (certain) relation, the value of an attribute a for each tuple $t.a$, would be a single value in D. In the case of an uncertain relation, $t.a$ is a probability distribution over D instead of a single value. Let $D = \{d_1, d_2, ..., d_N\}$, then $t.a$ is given by the probability distribution $Pr(t.a = d_i)$ for all values of $i \in 1, ..., N$.

Come back to our motivation example, the illnesses that can affect the cows is in the categorical domain of $Illness = \{da, ds, mc, nc, ...\}$, where MC means mad cow, FS fever acurate....

The problem of indexing uncertain data arises frequently in the context of several application domains such as moving trajectories or sensor data [8, 9]. In such cases, the data is updated only periodically in the index, and therefore the current attribute values cannot be known exactly; it can only be estimated. There are different kinds of queries to be processed over the distributed sites using the index structures. Next, we define the type we are interested in this paper.

2.3. Semantic query

In this paper, we are interested in process probability threshold query, and top-k query over distributed uncertain database.

Probabilistic threeshold query (PTQ)

Given a relation R, and a an uncertain attribute and threshold probability $\tau, \tau > 0$ The query returns all tuples t from R, along with probability values, such that the probability value $Pr(q = t.a) > \tau$. For instance:
SELECT T_{id} FROM S1 WHERE ilness="mc" with $Pr(q = t.a) > 0.5$.

If we adjust the threshold probability in a dynamic way, we can obtain the top-k highest tuples that have the same illness.

Probabilistic top-k query (Ptop-k)

Given a Relation R, and an uncertain attribute a, the top-k query returns the most promising attribute that have the highest probability in relation R

3. Indexing distributed sites

Straightforward approach to quering a distributed database is to broadcast the query to all sites and get the response from them. The drawback of this approach is, that some sites can have no answer for the uncertain query. Then, the system will spend a lot of time waiting the sites which are not qualified to give a response. Moreover, the communication between peers can cause a server bottleneck and bandwich consummation. Furthermore, the top-k query in a distributed environment can consume bandwich and is costly in time processing, because all sources send their k tuples to the query coordinator.
In this section, we discuss the first contribution, including a global index named GII(Global Inverted Index) based on the one presented in [7]; the aim of the index is to query only the site from which we can get an answer. The proposed index allows a pruning phase by consulting it. Notice that, the performance networking is out of the scope of the paper. This section section will show :

- Local indexing model: the categorical data in each source are indexed using the inverted index proposed in [7].
- Global indexing model: the main advantage of the global indexing is that some of the sites are not involved in the processing of a given query, only the appropriate sites are triggered.

3.1. Uncertain Local Indexing

In this part of our framework in the local site, we adopt the inverted index proposed by Singh &al in [7]. The basic technique for inverted list consists in making a list for each word in the text, the structure is organized in two parts, a vocabulary for the relevant word and their occurrences in the text.

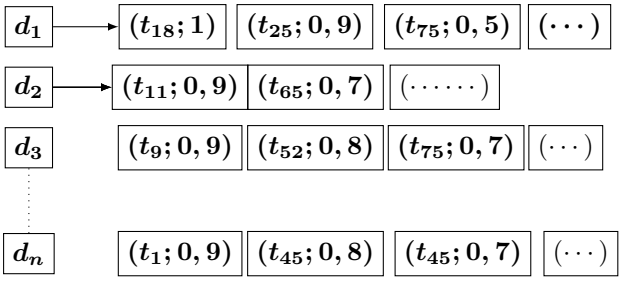

Figure 1: Inverted index in Local Site S_1

For each value $d \in D$ the uncertain attribute $R.a$, it is stored in the list of inverted index a set of pairs including the Tid tuple of R and the probability of $R.a$ attached to this T_{id}. So the component of the list of a given $d_i \in D$ is a pair of (t_{id}, P_i), this list is organized in decreasing order. In practice such a list is organized in the memory with a dynamic structure like b+ tree. for this type of index Singh & al [7] have proposed three heuristics for searching in list.
Figure[1] depicts an example of inverted index in local. For instance the value $\boldsymbol{d_1}$ is found in t_{18} with probability 1, and in tuple t_{25} with probability 0.9, and in tuple t_{75} with probability 0.5, and so on. The same reasoning with the rest of data d_i. Each list organizes the pairs in a descending order.

3.2. Uncertain Global Indexing

The global index is a distributed index called GII is stored in the coordinator site. The main steps adopted to build this Global index are:

- Collect the representative of each database
- For each categorical data, the index stores a list of pairs that represents the source of this data and its maximal probability in this source stored in the local index.

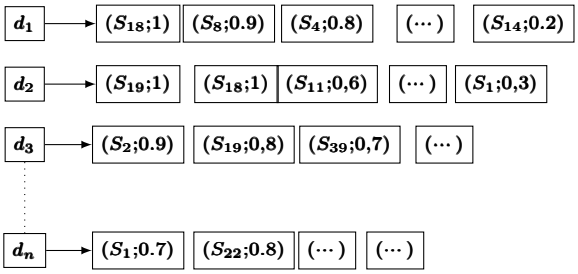

Figure 2: Global inverted Index (GII)

For the structure of GII in Figure [2], we have adopted the same structure as the one used in the inverted index for local uncertain database. The difference is that, the list of pairs in GII is the id_{Site} where the data d exists and its maximal probability in this site. so the proposed structure of the global index is:
for each $d_i \in D$ and $S_i \in S$ the global index stores a list of pairs (S_i, P_{max}) for each value of d_i. For instance, in Figure[2], the value d_1 is located in the site S8 with the maximal probability equal to 1, in the site S8 with maximal probability 0.9, and so on.
In the next section we discuss how to provide two kinds of query over uncertain distributed sites uing the proposed structure of the index.
In the next section we discuss how to provide two kinds of query over uncertain distributed sites in the proposed structure of the index

4. Query processing

For the simple reason, we had chosen a coordinator site to implement the global index and all the queries have been executed from this site.

A straightforward approach to quering a distributed uncertain database system is to broadcast a query to all sites from the coordinating site, each site will execute a query locally and send the response to the site coordinator. The latter will integrate all the responses received from the sites. The drawback of this approach is that querying some sites that will not give any answers will engender a time cost to get the response. Hence, we propose two algorithms for querying uncertain database system over a distributed environment called probabilistic threshold-query algorithm and probabilistic top-k query algorithm.

4.1. Probalistic threshold query processing

The aim of our algorithm depicted in algorithm1 is to eliminate the sites that are not concerned by the query:

input : Query Q,GII index,τ threshold
output: **thr_response()**

$thr\_response = \{\}$;

$Q_{answer} = \{\}$

Execute query Q in coordinator site;

Goes through the index GII;

for *each Source in index GII* **do**
  if $p_{max} > \tau$ **then**
    Execute query Q ;
    $Q_{answer} \leftarrow Answer\_of\_Q$
  end
  $thr\_respnse \leftarrow trh\_response \cup Q_{answer}$
end
return(**thr_response**)

Algorithm 1: Threshold Query Algorithm

- The first step of this algorithm is to execute the query in the coordinator site,
- After the algorithm goes through the global index to get only the source we are sure we can get the attribute by considering its probability greater than a threshold τ defined in the query ($p_{max} > \tau$). All sites that can not satisfy this condition will not be concerned by the query Q.
- In the last step of our algorithm the query will be executed on the site which satisfy the threshold probability, and the coordinator will integrate the results obtained from all the sites.

Example 4.1 *We illustrate the main steps of the threshold algorithm through the previous example. Let us consider the following query on the distributed relation R in table[1]:*

$Q_1 : SELECT * FROM\ R\ WHERE\ illness =' da'\ AND \tau = 0.5$
First, from the GII (shown in Figure[2]), the distributed pruning returns S_1 and S_3 nodes that correspond to the illness attribute value (i.e. da) with a probability above 0.5. Q_1 is then executed on S_1 and S_3. Finally, the local results are merged on the site where $Q1$ is issued.

4.2. Ptop-k query processing

The naive approach to get top-k answers over distributed uncertain database is to get all top-k answers from each site and integrate them into the coordinator site and final rank the result to obtain top-k answers in distributed systems. The drawback of this approach is to get some results that are not contained in the final response which engenders a high time and communication cost between coordinator site and all sites.

input : Query Q,GII index,τ threshold, K
output: **topk_response()**

$topk\_response = \{\}$;
Execute query Q in coordinator site;
Travers the index GUI;
$Li \leftarrow$ liste of site from GII
for *each Source in liste Li* **do**
  explore Local index;
  $Lp_{min} \leftarrow$ get the probability of k_{th} of site from Local index ;
end
$\tau \leftarrow$ Max of Lp_{min}

Execute Threshold alghorithm (τ)

Top-k query$\leftarrow$ Ranking the global results and choose the k first Tuple
return(**topk_response**)

Algorithm 2: topk Algorithm

Our approach is conducted in the algorithm depicted in Algorithm2 as follows:

- The same processing is performed as for threshold queries. Let $S_{Q2} = \{S_1, S_2, ..., S_m\}$ where $m \leq n$, be the list of the distributed sites where $Q2$ should be executed. The query node S_c broadcasts the query $Q2$ to those nodes. Notice that GII allows a distributed pruning.
- On each site S_i in S_Q from the previous step, the algorithm explores its local index LUI. The aim is to search the k^{th} probability P_i (i.e. the maximal probability in the LUI inverted list) from each selected site. Then, this probability value is sent to the query node where a threshold list $ThList$ stores all the probability values sent by the distributed sites. The maximal probability, $\delta = Max_{i \in [1,m]} P_i$, is considered as the new probability threshold and a new pruning is performed from the global index GII according to δ. Indeed, for each pair (S_i, P_{max})

from GII inverted Index, if $P_{max} \lesssim \delta$, then the corresponding site S_i is pruned. The result of this step is a new list S_{Q_δ} of distributed sites to execute Q, where $S_{Q_\delta} = \{S_1, S_2, ..., S_t\}$ where $t \lesssim m$. Consequently, the query node S_c sent the query Q to th sites in S_{Q_δ}.

- On each node S_i in S_{Q_δ}, the algorithm executes Q. It explores each local index LUI and gets the tuples form the corresponding inverted list of the uncertain attribute value. Each node ranks its own tuples in decreasing probability order, stopping when no more tuples are likely to satisfy k. Then the ranked k^{th} tuples are sent to the query node.
- On the query node, the algorithm computes the final query result. Another ranking is performed from the whole top-k tuples of each node.

Example 4.2 *Let us consider the following query on the distributed relation R of table[1]:* $Q2$: $SELECT * FROM\ S\ WHERE\ illness =' da' limit 100$

- *First, the GII of the distributed relation R is explored to find the value* ' da ' *of the uncertain attribute illness. Only* S_1 *and* S_2 *nodes are returned, as the value* ' da ' *constitutes an entry in the GII. Notice that* S_3 *and* S_4 *are pruned.*
- *Next,* $Q2$ *is executed on* S_1 *and* S_2 *nodes. From their local indexes LUIs, the algorithm gets the top 100th probability values (ie. the maximal probability) corresponding to each LUI entry of da value. Those probability values are sent to the query node and the maximal one is defined as a new probability threshold. With this new threshold, the GUI is explored and the nodes satisfying it are selected.*
- $Q2$ *is executed on the selected sites by exploring their LUIs. At this step, the algorithm gets the tuple identifiers that satisfy* $Q2$*, and ranks the corresponding tuples in decreasing order of their probability.*
- *Finally, on the query node, only the top 100 tuples are ranked from the whole candidate tuples coming from local sites.*

The proposed algorithm gives an effective top-k tuples with minimum cost communication and reduces the transferring tuple between coordinator site and the other sites. it executes the process in two round communications between the coordinator site and the other sites.

5. Related Work

Uncertain data have received special attention from the database community. Indeed, since the theory of probabilistic databases, considered as a generalization of relational databases, was outlined [1], numerous research projects have treated data uncertainty [1, 2] Researchers have investigated traditional databases taking into account the uncertain aspect of data, like modeling uncertainty [1, 10, 11], query evaluation [2, 12], indexing, query processing against relational and spatial uncertain data [7, 8, 9, 13].

In the context of centralized uncertain databases, Shing et al [7] had proposed two index structures to search and manage uncertain categorical data. The first index, called probabilistic inverted index, is based on inverted list. And, the second, Probabilistic Distribution Rtree, is based on the Rtree structure. In our approach we have dealt with the first index structure.
In a distributed environment, many approaches have been studied the topk query and ranking distributed uncertain databases [4, 14, 6, 8], Sun etal [8] have proposed a novel definition of a top - k query across P2P networks, and a global Q-tree index based on Q-tree structure was used. The index summarized all indices of the peers of the P2P network, where each pair had a local index based on the same structure of the global index. Also, they have proposed a spatial pruning algorithm to reduce communication costs between peers and minimize data transfer.

This approach deals with the uncertain object and the main algorithm requires several phase communication between the peers.Second, Li et al [14] proposed an approach where the tuples in local site are sorted based on expected rank. Queries are executed from a central server that accesses to the tuples of all sites in order to construct a global rank. The server maintains a priority queue to store each sites tuple with their ranks. Then it initializes the priority queue with the first tuple from each site. The global rank of each tuple t_{ij} (from site S_i ranked j_{th}) in priority queue is computed by broadcasting this tuple to all other nodes, except Sender. The server gets the next tuple from Sender along with its expected rank. When the rank of the first tuple in the queue is higher than or equal to kth element rank, the process terminates. The drawback of this approach is concerned by transmission bandwidth tradingoff latency. The last approach, proposed by AbdulAzeem et al [4] , is implemented as a framework for ranking a distributed uncertain database. The approach is based on three layers : query layer,ranking layer an monitoring layer. Indeed, after broadcasting the query to all the nodes, these nodes send the list of their candidates topk to the query node who will rank the received list and maintains the score of kth as minimum bound. In the second phase, the query node informs the other nodes to recalculate their topk. The drawback of this approach is to in the ranking layer all the nodes cause transmission bandwidths and send their list of promising tuples. Even if they will not be taken in the final result. To the best of our knowledge, our

work is the first to address the problem of indexing uncertain categorical data in distributed environments

6. Conclusion

In this paper, we addressed the problem of indexing and query processing over uncertain categorical data in distributed environments. We proposed an approach to query uncertain data over distributed database. It is presented at local and global index based on the inverted index approach. In this paper, we considered two types of query over the structure of the provided indexes, for that two algorithms are suggested to handle probabilistic threshold queries and probabilistic top- k query. Our proposed algorithms perform distributed pruning, and then allow minimum communication and minimize processing costs. A prototype of the approach is ongoing.

References

[1] Daniel Barbará, Hector Garcia-Molina, and Daryl Porter. The management of probabilistic data. *IEEE Trans. Knowl. Data Eng.*, 4(5):487–502, 1992.

[2] Nilesh N. Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. [15], pages 864–875.

[3] Yinian Qi, Rohit Jain, Sarvjeet Singh, and Sunil Prabhakar. Threshold query optimization for uncertain data. In Ahmed K. Elmagarmid and Divyakant Agrawal, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010*, pages 315–326. ACM, 2010.

[4] Yousry M. AbdulAzeem, Ali I. El-Desouky, and Hesham A. Ali. A framework for ranking uncertain distributed database. *Data Knowl. Eng.*, 92:1–19, 2014.

[5] Yongjiao Sun, Ye Yuan, and Guoren Wang. Top-k query processing over uncertain data in distributed environments. *World Wide Web*, 15(4):429–446, 2012.

[6] Naser Ayat, Reza Akbarinia, Hamideh Afsarmanesh, and Patrick Valduriez. Entity resolution for probabilistic data. *Inf. Sci.*, 277:492–511, 2014.

[7] Sarvjeet Singh, Chris Mayfield, Sunil Prabhakar, Rahul Shah, and Susanne E. Hambrusch. Indexing uncertain categorical data. In Rada Chirkova, Asuman Dogac, M. Tamer Özsu, and Timos K. Sellis, editors, *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15-20, 2007*, pages 616–625. IEEE, 2007.

[8] Reynold Cheng, Yuni Xia, Sunil Prabhakar, Rahul Shah, and Jeffrey Scott Vitter. Efficient indexing methods for probabilistic threshold queries over uncertain data. In Nascimento et al. [15], pages 876–887.

[9] Jens Henrik Hosbond, Simonas Saltenis, and Rasmus Ørtoft. Indexing uncertainty of continuously moving objects. In *14th International Workshop on Database and Expert Systems Applications (DEXA'03), September 1-5, 2003, Prague, Czech Republic*, pages 911–915. IEEE Computer Society, 2003.

[10] Sarvjeet Singh, Chris Mayfield, Rahul Shah, Sunil Prabhakar, Susanne E. Hambrusch, Jennifer Neville, and Reynold Cheng. Database support for probabilistic attributes and tuples. In Gustavo Alonso, José A. Blakeley, and Arbee L. P. Chen, editors, *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, México*, pages 1053–1061. IEEE, 2008.

[11] Sarvjeet Singh, Chris Mayfield, Sagar Mittal, Sunil Prabhakar, Susanne E. Hambrusch, and Rahul Shah. Orion 2.0: native support for uncertain data. In Jason Tsong-Li Wang, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1239–1242. ACM, 2008.

[12] Sarvjeet Singh, Chris Mayfield, Rahul Shah, Sunil Prabhakar, and Susanne E. Hambrusch. Query selectivity estimation for uncertain data. In Bertram Ludäscher and Nikos Mamoulis, editors, *Scientific and Statistical Database Management, 20th International Conference, SSDBM 2008, Hong Kong, China, July 9-11, 2008, Proceedings*, volume 5069 of *Lecture Notes in Computer Science*, pages 61–78. Springer, 2008.

[13] Pankaj K. Agarwal, Siu-Wing Cheng, Yufei Tao, and Ke Yi. Indexing uncertain data. In Jan Paredaens and Jianwen Su, editors, *Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009, June 19 - July 1, 2009, Providence, Rhode Island, USA*, pages 137–146. ACM, 2009.

[14] Feifei Li, Ke Yi, and Jeffrey Jestes. Ranking distributed probabilistic data. In Ugur Çetintemel, Stanley B. Zdonik, Donald Kossmann, and Nesime Tatbul, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, Providence, Rhode Island, USA, June 29 - July 2, 2009*, pages 361–374. ACM, 2009.

[15] Mario A. Nascimento, M. Tamer Özsu, Donald Kossmann, Renée J. Miller, José A. Blakeley, and K. Bernhard Schiefer, editors. *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*. Morgan Kaufmann, 2004.