16th World Congress of the International Fuzzy Systems Association (IFSA)
9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT)

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 incuding indexing methods are not suitable to handle such type of data. In this paper, we propose an inverted based index method for effciently searching uncertain categorical data over distributed environments. We adress two kinds of query over the distributed uncertain databases, one a distributed probabilistic thresholds query, where all results sastisfying the query with probablities that meet a probablistic 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, 5, 6]. 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, 3]. 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 [7], it is proposed a novel approach of processing uncertain top-k query in large-sacle P2P networks, [8] 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 inverted index introduced first in [9], (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, egality query , top-k query. The proposed global index uses a prunning 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 propese a distributed index structure for uncertain data over distributed database. We suggest an inverted based local index and a global index summarizing each distributed site.
- We propose prunning 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 by 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 uncertains and in the other hand on the diagnosis

T_{id}	weight	illness
$T_{1,1}$	700	$\{(da, 0.5); ds(0.5)\}$
$T_{1,2}$	710	$\{(ds, 0.2); ds(0.8)\}$
$T_{1,3}$	790	$\{(nc, 1)\}$
....	..	...
$T_{1,n}$	725	$\{(nc, 0.9); (ds, 0.1)\}$

T_{id}	weight	illness
$T_{2,1}$	719	$\{(mc, 0.4); nc(0.6)\}$
$T_{2,2}$	780	$\{(mc, 0.9); nc(0.1)\}$
$T_{2,3}$	725	$\{(nc, 1)\}$
....	.	.....
$T_{1,n}$	688	$\{(nc, 0.9); (mc, 0.1)\}$

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

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

Table 1: exemple of distributed uncertain databases

provided by the veterinary surgeons themselves during their visits to the farms. For instance Table 1 gives more than three distributed sites (farms) with an uncertain attribute "Ilness". $T_{1,1}$ means that the farms 1 has a cow with identity $T_{1,1}$ with weight 700 and its ilness may be "acute diarrhoea (da)" with probabilit 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 ilness may be "mad cow (mc)" with probabilit 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 desease. The aim is not to query the whole distributed farms but only those affected by the desease. The service whishes 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 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 database can be modeled by two view points, the first one affects a probability value to each tuple called (tuple uncertainty)and the second affects a probability to one attibute. 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 [9], $R_i.a$ 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 ilnesses that can affect the cows is in the categorical domain of $Ilness = \{MC, FS, FA, ...\}$, 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 [10, 11]. 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 to process probability threshold query, and top-k query over distributed database.

Probabilistique threeshold query (PTQ)

Given a relation R, and a an uncertain attribute and threshoold 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.

Probabilistique 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 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, the site can have no answer for the uncertain query. Then, the system will spend a lot of time waiting the site which is not qualified to give a response. Moreover, the communication between peers can cause a server bottleneck and bandwich consommation. Furthermore, the top-k query in a distributed environment can consume bandwich and is costly in time processsing, 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 based on the one presented in [9]; the aim ouf 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 [9].
- 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 appropriarte 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 [9]. The basic technique for inverted list consists in making a list for each word in the text, the structure is organized on two parts, a vocabulary for the relevant word and their occurrences in the text. For each value d of uncertain data 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 [9] have proposed three heuristics for searching in list.

Figure 3.1 depicts an exemple of inverted index in local. For instance the value $\boldsymbol{d_1}$ is found in (t_{18} with

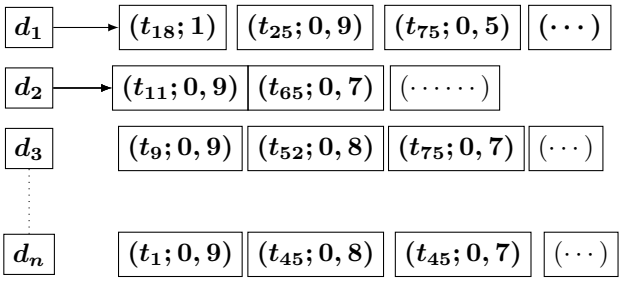

Figure 1: inverted index in Local Site

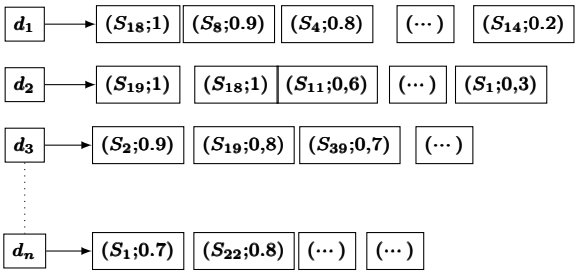

Figure 2: Global inverted Index (UII)

probability 1, and in tuple (t_{25} with probability 0.9, and in tuple (t_{75} with probability 0.5, and os on. The same reasoning with the rest of data di.

Each list organizses the pairs in a descendant order.

3.2. Uncertain Global Indexing

The global index is a distributed index called UII 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.

For the structure of UII in Figure 3.2, we have adopted the same stucture as the one used in inverted index for local uncertain database. The difference is that the list of paire in UII is the id site where the data d exists and its maximal probabilty in local 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 di.

For instance, in Figure 3.2, the value d1 is located in the site S8 with the maximal probiblity 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 index.

4. Query processing

For the simplicity reason, we had choosen a cordinator 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 coordinator site, each site will execute a query locally and send the response to the cordinator site. The latter will integrate all the responses received from the sites. The drawback of this approach is that quering some sites that will not give any anwers will engeder a time cost to get the response. Hence, we have proposed two alghorithm to quering uncertain database system over distributed environment called prabobilsitic threshold-query algorithm and probabilistic top-k query alghorithm.

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:

- 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 consedering its probability greater than a threshold τ defined in the query ($p_{max} > \tau$). All sites that can not staisfy this condition will not be concerned by the query Q.
- In the last step of our alghorithm the query will be executed in the site which statisfy the threshold probability, and the coordinator will integrate the results obtained from all the sites.

As an exemple, in table[1] we suppose that we whould have all cows that are affected by mc illness with probability(p>04), then only the sites S_1 and $S3$ will be concerned by the query Q, the other sites are not because the threshold is greater to the maximal probability of given illness of those sites.

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 in the coordinator site and finally rank the result to obtain top-k answers in distributed system. 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,*UII* index,τ threshold
output: **threshold_response()**

$thrshold\_response = \{\}$;
Execute query Q in coordinator site;
Goes through the index UII;
for *each Source in index UII* **do**
  if $p_{max} > \tau$ **then**
    Execute query Q ;
  end
  $thrshold\_respnse \leftarrow threshold\_response$
end
return(**threshold_response**)

Algorithm 1: Threshold Query Algorithm

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

- we introduce a threshold probability that can prunne some sources which can not give answers to the query,
- we explore the UII index to get the list of sources that we are sure they can send our top-k,
- the top-k query will be excuted in all sites in this of the list to get the probability of k-th tuple of all, and send it to the coordinator site,
- the cordinator site defines the maximal probability got from the source as the threshold,
- the query is rewritten with a given threshold
- the probabilsitic threshold query algorithm is executed
- the site concerned by the query will send the answers to query site to integrate all answers and will rank them to get the top-k promisiong tuple.

The proposed alghorithm gives an eeffectivness top-k tuples with minimum cost communication and reduces the transfering tuple betwwen coordinator site and the other sites. it executes the process in two round communications between the query site and the other sites.

5. Related Work

Due to the importance of data uncertainty, the communoty of database society has studied the uncertain database since the eighties last century. they have reinvestigated under uncertain symantique an number of technical issues in traditional databases like modeling uncertainty [1, 12, 13], query evaluation [3, 2, 14], indexing, query processing against relational and spatial uncertain data [9, 10, 11, 15]. in the contex of central uncertain databases shing &al [9] had proposed for shearshing and managing uncertain catÃ©gorical data tow index structure, the first one called probabilistic inverted index based on inverted liste and the second Probabilistic Distribution Rtree based on and Rtree structure, in our approach we have deal wiht the first

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

topk_response = {};
Execute query Q in coordinator site;
Travers the index GII;
$Li \leftarrow$ liste of site from GII
for *each Source in liste Li* **do**
  execute query Q;
  $Lp_{min} \leftarrow$ get the probability of k_{th} of site ;
end
$\tau \leftarrow$ Max of Lp_{min}
Rewrite Query Q w.t.r τ
threshold_query(τ)
Top-k query$\leftarrow$ the k first of Threshold_query()
return**(topk_response)**

Algorithm 2: Top-k Query Algorithm

index structure. in the contex of distributed environement there has been many approach studied the topk query and rankin distriubuted uncertain databases [6, 16, 5, 8],in the appraoche of Li&al [16] the tuples in local site are sorted based on expected rank, the query query are excuted from a central server that he accesses tuples from all site in order ranking then he maintains a priority queue to store each sites tuples with their ranks. Then it initializes the priority queue with first tuple from each site. The global rank of each tuple t_{ij} (from site Si ranked j_{th}) in priority queue is computed by broadcasting this tuple to all other nodes except Sender The query nodegets the next tuple from Sender along with its expected rank. When the rank of the first tuple in the queue is higher than or equalkth element rank the process terminates. the drawback of this approach is concerned by transmission bandwidth tradingoff latency

the second approach proposed by AbdulAzeem &al [6] ranking query or topk, AbdulAzeem &al [6] is a framework for ranking a distributed uncertain database, their main approach are executed in three layer (query,ranking an monitoring) after broadcasting the query to all the node , the nodes send the liste of their candidates topk to the query node who will rank the received liste and maintain the score of kth as minimum bound and informe in the seconde phase the other nodes, to recalculating their topk the droewback of this approach is to in the rankin layer all the nodes cause transmission bandwidth and all the node are send their liste of prommissing tuple.Even if they will not be taken in the final result. to the best of our knowledge, ours is the first work to address the problem of indexing uncertain categorical data in distributed environement

6. Conclusion

In this paper we proposed an approach to query uncertain data over distributed database. It is presented a local and global index based on inverted index approach. IN this paper we considered two types of query over over the structure of the provided indexes, for that two algorithms are suggested to handle probabilistic threshold query and probabilistic top- k query. A protype 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 Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. *The VLDB Journal*, 16(4):523–544, October 2007.

[3] Nilesh Dalvi and Dan Suciu. Management of probabilistic data: foundations and challenges. In *PODS '07: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–12, New York, NY, USA, 2007. ACM Press.

[4] Yinian Qi, Rohit Jain, Sarvjeet Singh, and Sunil Prabhakar. Threshold query optimization for uncertain data. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 315–326, New York, NY, USA, 2010. ACM.

[5] Qiming Fang and Guangwen Yang. Efficient top-k query processing algorithms in highly distributed environments. *JCP*, 9(9):2000–2006, 2014.

[6] 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.

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

[8] Naser Ayat, Reza Akbarinia, Hamideh Afsarmanesh, and Patrick Valduriez. Entity resolution for distributed probabilistic data. *Distributed and Parallel Databases*, 31(4):509–542, 2013.

[9] Sarvjeet Singh, Chris Mayfield, Sunil Prabhakar, Rahul Shah, and Susanne E. Hambrusch. Indexing uncertain categorical data. In *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15-20, 2007*, pages 616–625, 2007.

[10] Reynold Cheng, Yuni Xia, Sunil Prabhakar, Rahul Shah, and Jeffrey Scott Vitter. Efficient indexing methods for probabilistic threshold queries over uncertain data. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, Au-*

gust 31 - September 3 2004, pages 876–887, 2004.

[11] 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, 2003.

[12] Sarvjeet Singh, Chris Mayfield, Rahul Shah, Sunil Prabhakar, Susanne Hambrusch, Jennifer Neville, and Reynold Cheng. Database support for probabilistic attributes and tuples. In *ICDE '08: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*.

[13] Sarvjeet Singh, Chris Mayfield, Sagar Mittal, Sunil Prabhakar, Susanne Hambrusch, and Rahul Shah. Orion 2.0: native support for uncertain data. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1239–1242, New York, NY, USA, 2008. ACM.

[14] Sarvjeet Singh, Chris Mayfield, Rahul Shah, Sunil Prabhakar, and Susanne Hambrusch. Query selectivity estimation for uncertain data. In *SSDBM '08: Proceedings of the 20th international conference on Scientific and Statistical Database Management*, pages 61–78, Berlin, Heidelberg, 2008. Springer-Verlag.

[15] Pankaj K. Agarwal, Siu-Wing Cheng, Yufei Tao, and Ke Yi. Indexing uncertain data. In *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, 2009.

[16] Feifei Li, Ke Yi, and Jeffrey Jestes. Ranking distributed probabilistic data. In *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, 2009.