Evidential Top-*k* Queries Evaluation: Algorithms and Experiments

Fatma Ezzahra Bousnina[1,3](✉), Mouna Chebbah[2],
Mohamed Anis Bach Tobji[2], Allel Hadjali[3], and Boutheina Ben Yaghlane[4]

[1] LARODEC, Institut Supérieur de Gestion, Université de Tunis, Tunis, Tunisie
fatmaezzahra.bousnina@gmail.com
[2] ESEN, Univ. Manouba, Manouba, Tunisie
mouna.chebbah@esen.tn, anis.bach@isg.rnu.tn
[3] LIAS, ENSMA - Université de Poitiers, Poitiers, France
allel.hadjali@ensma.fr
[4] Institut des Hautes Etudes Commerciales, Université de Carthage, Tunis, Tunisie
boutheina.yaghlane@ihec.rnu.tn

Abstract. Top-*k* queries represent a vigorous tool to rank-order answers and return only the most interesting ones. ETop-*k* queries were introduced to discriminate answers in the context of evidential databases. Due to their interval degrees, such answers seem to be difficult to rank-order and to interpret. Two methods of ranking intervals were proposed in the evidential context. This paper presents an efficient implementation of these methods and discusses the experimental results obtained.

Keywords: Evidence theory · Evidential databases
Evidential Top-k queries

1 Introduction

Querying imperfect databases received a lot of importance recently with the emergence of domains like sensor networks, data cleaning, recommendation and recommender systems, etc. Indeed, information generated from this type of applications is obviously pervaded with imperfection (uncertainty, imprecision, ignorance...). That is why, database models that handle imperfect data were introduced (Probabilistic, possibilistic and evidential databases [1,3,5,6,16]). The latter, models several types of imperfect data but also perfect information using theory of belief functions. In database management, querying is a fundamental step. As consequence, multiple types of 'imperfect' queries were introduced. We name the evidential skyline [9], the extended relational queries [1] and the evidential Top-*k* queries [4].

In general, Top-*k* queries are needed in real world applications. For an example, movies, music and books are ordered by the preferred ones, researchers by their H-index, etc. Imperfect top-*k* queries can be very challenging when it comes to their semantics but also when it comes to their practical implementation.

J. Medina et al. (Eds.): IPMU 2018, CCIS 853, pp. 407–417, 2018.
https://doi.org/10.1007/978-3-319-91473-2_35

In this paper, we present two algorithms of evidential Top-k queries: The first named *NaiETop-k* is based on the computation of the preference degrees (called evidential scores) as introduced in [18] and adapted in [4]. The second named *OptETop-k* is based on an optimized version of the preference degree calculation justified by the complementarity property as detailed in [4]. The proposed implementation allows the ranking of all evidential scores and finally, it provides the k most interesting results among all rank-ordered answers.

Table 1 is an example of an evidential table that stores some users' preferences about books: b_1, b_2, b_3, b_4. This relation includes three attributes: The ID which is a unique reader identifier. The *BookRate* that includes the reader's appreciations about one book and/or several books modeled through the belief functions theory (in this context only few researches addressed the issue of preference elicitation using this theory [2,11]). The CL which is a specific attribute to evidential databases that stores intervals of confidence about user's responses.

Table 1. Books appreciations' table: EDB

ID	BookRate	CL
1	b_1 0.3	[0.5; 1]
	$\{b_2, b_3\}$ 0.7	
2	b_2 0.5	[0.3; 0.8]
	b_4 0.5	
3	$\{b_1, b_2, b_3\}$ 1	[1; 1]
4	b_3 1	[0.5; 0.9]

This paper is organized as follows: we recall, in Sect. 2 some basic concepts about the belief functions theory and evidential databases. In Sect. 3, we remind needs and challenges of evidential Top-k queries and we present the mathematical materials to compute and compare *Evidential Scores* and *Preference Degrees*. Section 4 is dedicated to the presentation of proposed algorithms. Experiments and results are shown in Sect. 5. Section 6 is devoted to the conclusion and the future works.

2 Evidence Theory and Evidential Databases

Evidence theory named *the belief functions theory or the Dempster-Shafer theory* [7,8,17], is a powerful tool to model ignorance and to represent uncertain, imprecise and inconsistent information.

In the theory of belief functions, a set $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ is a finite, non empty and exhaustive set of n elementary and mutually exclusive hypotheses related to a given problem. The set Θ is called the *frame of discernment* or *universe of discourse.*

The *power set* $2^{\theta} = \{\varnothing, \theta_1, \theta_2, \ldots, \theta_n, \{\theta_1, \theta_2\}, \ldots, \{\theta_1, \theta_2, \ldots, \theta_n\}\}$ is the set of all subsets of Θ.

A *mass function*, noted m, is a mapping from 2^{Θ} to the interval $[0, 1]$. The *basic belief mass* of an hypothesis x is noted $m(x)$, it represents the belief on the truth of that hypothesis x. A mass function is also called *basic belief assignment* (*bba*). It is formalized such that:

$$\sum_{x \subseteq \Theta} m^{\Theta}(x) = 1 \tag{1}$$

If $m^{\Theta}(x) > 0$, x is called *focal element.* The set of all focal elements is denoted F and the couple $\{F, m\}$ is called *body of evidence.*

The belief function, denoted *bel*, is the minimal degree of support committed exactly to x such that:

$$bel(x) = \sum_{y \subseteq x; y \neq \varnothing} m^{\Theta}(y) \tag{2}$$

The plausibility function, denoted *pl*, is the maximal degree of support committed exactly to x such that:

$$pl(x) = \sum_{y \subseteq \Theta; x \cap y \neq \varnothing} m^{\Theta}(y) \tag{3}$$

An evidential database, denoted EDB, stores different types of data using the belief functions theory as shown in Table 2.

Table 2. The different types of information modeled in the evidential database

Information	Properties	Example
Certain	When the focal element is a singleton with a mass equal to 1 bba is Certain	b_3 1
Probabilistic	When focal elements are singletons bba is Bayesian	b_2 0.5 b_4 0.5
Possibilistic	When focal elements are nested bba is Consonant	b_1 0.2 $\{b_1, b_2\}$ 0.8
Evidential	When none of previous types is present bba is Evidential	$\{b_1, b_2, b_3\}$ 1

Definition 1. *[Compact Evidential Database]*
An EDB *has N objects and A attributes. An* evidential value*, noted V_{la}, is the value of an attribute a ($1 \leq a \leq A$) for an object l ($1 \leq l \leq N$) that represents a basic belief assignment.*

$$V_{la} : 2^{\Theta_a} \rightarrow [0, 1] \tag{4}$$

$$with\ m_{la}^{\Theta_a}(\varnothing) = 0 \quad and \quad \sum_{x \subseteq \Theta_a} m_{la}^{\Theta_a}(x) = 1 \tag{5}$$

The set of focal elements relative to the bba V_{la} is noted F_{la} such that:

$$F_{la} = \{x \subseteq \Theta_a / m_{la}(x) > 0\} \tag{6}$$

A confidence level*, CL, is a specific attribute that includes intervals. Each one represents the confidence about its object l in the evidential database. The confidence level is a pair of belief and plausibility [bel; pl] reflecting the pessimistic and the optimistic degrees of support about each object' existence in the database [1, 14, 15].*

Multiple types of queries can be applied over an EDB like the extended relational operators (select, project, join...) [1,14,15], skyline queries [9,10] and ranking queries [4].

3 Evidential Top-*k* Querying

Top-k queries represent a mighty tool to order queries' results and give only the most interesting answers. Top-k queries were firstly introduced in the multimedia systems [12,13]. They use a score function to rank answers where only results with the highest scores are returned.

Evidential Top-k queries, denoted ETop-k, rank answers using an evidential score function and return the most interesting ones (with the highest scores). Contrary to usual top-k queries that give a ranking based on a score function with precise values, the ETop-k queries give answers based on a score function with intervals. The latter reflect the minimal and the maximal amounts of confidence about each answer.

Definition 2. *[Evidential Score]*
Let R_i be a response generated from processing a query Q over an evidential database EDB of a size N and let $S(R_i)$ be the score function of that answer R_i and $bel(R_i)$ and $pl(R_i)$ are respectively its belief and plausibility in the table, such that:

$$S(R_i) = [bel(R_i); pl(R_i)] \tag{7}$$

$$where \quad bel(R_i) = \frac{\sum_{l=1}^{N} bel_l(R_i) * bel_l}{N}$$

$$pl(R_i) = \frac{\sum_{l=1}^{N} pl_l(R_i) * pl_l}{N}$$

The belief of an answer, $bel(R_i)$, is a disjunction of the response's beliefs in each object of the database. The belief of a response in one object l, denoted bel_l, *is the product of its belief in the attribute and the belief of that object. Same for the plausibility of an answer, $pl(R_i)$. It is the disjunction of the response's plausibilities in each object of the database where the plausibility of a response in one object l, denoted pl_l is the product of its plausibility in the attribute and the plausibility of that object [1,14].*

Example 1. *The Top-k query processed over the evidential database of Table 1 is the following [4]:*

Q: ***SELECT*** *BookRate* ***FROM*** *EDB* ***ORDER BY*** *S(BookRate)* ***LIMIT*** *k;*

Four possible responses are computed using the evidential score as detailed in Definition 2:

- *S(b_1) = [bel(b_1); pl(b_1)] = [0,0375; 0.325]*
- *S(b_2) = [bel(b_2); pl(b_2)] = [0,0375; 0.525]*
- *S(b_3) = [bel(b_3); pl(b_3)] = [0,125; 0.65]*
- *S(b_4) = [bel(b_4); pl(b_4)] = [0,0375; 0.1]*

Often a top-k query processed over an evidential database gives a large number of results. These latter need to be ranked in order to respond to the objective of the given query. In the evidential case, the result is a set of intervals that must be compared. Two methods were introduced to compare interval results in EDBs' context [4,18].

(i) The first method was introduced in [18] and adapted in [4]. It is about computing degrees of preference of two intervals and then compare their results to deduce the rank based on three cases:

Definition 3. *[Preference Degree]*
Let $S(R_i) = [bel_i; pl_i]$ and $S(R_j) = [bel_j; pl_j]$ be two evidential scores. Each one is an interval composed of a belief degree and a plausibility degree. The degree of one interval to be greater than the other one is called a degree of preference *and denoted P.*

The degree of preference that $S(R_i) > S(R_j)$ is defined such that:

$$P(S(R_i) > S(R_j)) = \frac{max(0, pl_i - bel_j) - max(0, bel_i - pl_j)}{(pl_i - bel_i) + (pl_j - bel_j)} \tag{8}$$

The degree of preference that $S(R_i) < S(R_j)$ is defined such that:

$$P(S(R_i) < S(R_j)) = \frac{max(0, pl_j - bel_i) - max(0, bel_j - pl_i)}{(pl_i - bel_i) + (pl_j - bel_j)} \tag{9}$$

The different cases of comparing intervals $S(R_i)$ and $S(R_j)$ are as follows:

- *If $P(S(R_i) > S(R_j)) > P(S(R_j) > S(R_i))$, then $S(R_i)$ is said to be superior to $S(R_j)$, denoted by $S(R_i) \succ S(R_j)$.*

- *If* $P(S(R_i) > S(R_j)) = P(S(R_j) > S(R_i)) = 0.5$, *then* $S(R_i)$ *is said to be indifferent to* $S(R_j)$, *denoted by* $S(R_i) \sim S(R_j)$.
- *If* $P(S(R_j) > S(R_i)) > P(S(R_i) > S(R_j))$, *then* $S(R_i)$ *is said to be inferior to* $S(R_j)$, *denoted by* $S(R_i) \prec S(R_j)$.

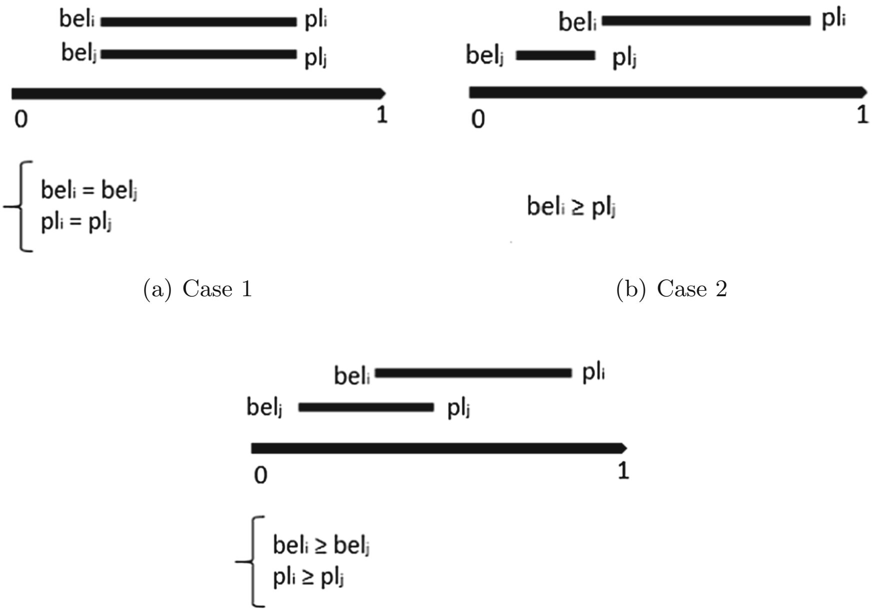
(ii) The second method optimizes the first one using the *complementarity* proof*, results are compared in order to deduce their rank [4]:

Definition 4. *[Optimized Preference Degree]*
Let $S(R_i) = [bel_i; pl_i]$ *and* $S(R_j) = [bel_j; pl_j]$ *be two evidential scores. Every interval is composed of degrees of belief (bel) and plausibility (pl) and P is the calculated preference degree.*

$$P(S(R_i) > S(R_j)) = \frac{max(0, pl_i - bel_j) - max(0, bel_i - pl_j)}{(pl_i - bel_i) + (pl_j - bel_j)} = \lambda \qquad (10)$$

The different cases of comparing intervals $S(R_i)$ *and* $S(R_j)$ *are as follows:*

- *If* $\lambda > 0.5$ *then* $S(R_i) \succ S(R_j)$.
- *If* $\lambda = 0.5$, *then* $S(R_i) \sim S(R_j)$.
- *If* $\lambda < 0.5$ *then* $S(R_i) \prec S(R_j)$.



(a) Case 1

(b) Case 2

(c) Case 3

Fig. 1. Specific cases to deduce evidential scores [4]

Cases that permit to minimize computations before using Definitions 3 or 4 are illustrated in Fig. 1.

Note here the importance of *transitivity*** property detailed in [18] to give the final ranking.

4 Implementation of Evidential Top-*k* Query

As the best of our knowledge, there is no implementation of evidential top-k queries. Indeed, we present in this paper an object-oriented implementation of two methods to rank evidential scores (the evidential intervals). The first method is naive, it consists on computing the preference degree through three steps each time: (a) it computes the preference degree that the first interval is superior to the second one and then (b) it computes the preference degree that the first interval is inferior to the second one. Finally, (c) it compares results and give the partial rank. This algorithm is presented in Table 3.

The second method is an optimization of the first one. Indeed, it consists on computing only in one step the preference degree and then deduce the partial order between two intervals. This algorithm is detailed in Table 4.

Finally, the last order is treated using a sorting algorithm, that ranks all evidential intervals and provide the k most interesting ones. The presented implementations offer two methods of evidential intervals' ranking. Both algorithms use the object-oriented paradigm for its programming benefits.

Table 3. ETop-k naive algorithm

```
Naive Method
Initialization
Tuple a, b ;
begin
if (a.Bel=b.Bel and a.Pl=b.Pl)
return 0;
if (a.Pl<b.Bel)
return -1;
if (b.Pl<a.Bel)
return 1;
if (a.Bel>b.Bel and a.Pl>b.Pl)
return 1;
if (b.Bel>a.Bel and b.Pl>a.Pl)
return -1;
if (score(a,b)>score(b,a))
return 1;
else return -1;
end
```

```
Naive Evidential Top-k Algorithm
Initialization
Integer m;
ArrayList Table;
begin
for (int i ⟵ 0; i<Table.size()-1; i++)
{ m⟵i;
for (int j⟵ i+1; j<Table.size(); j++)
{
if (NaiveMethod(Table.get(j), Table.get(m))=1)
{ m⟵j; }
}
if (NaiveMethod(Table.get(m) ,Table.get(i))=1)
{ Tuple c ⟵ Table.get(i);
Table.set(i,Table.get(m));
Table.set(m,c); }
}
end
```

Table 4. ETop-k optimized algorithm

ETop-k Method

```
Initialization
Tuple a, b ;
begin
if (a.Bel=b.Bel and a.Pl=b.Pl)
return 0;
if (a.Pl<b.Bel)
return -1;
if (b.Pl<a.Bel)
return 1;
if (a.Bel>b.Bel and a.Pl>b.Pl)
return 1;
if (b.Bel>a.Bel and b.Pl>a.Pl)
return -1;
if (score(a,b)>0.5)
return 1;
else return -1;
end
```

Optimized ETop-k Algorithm

```
Initialization
Integer m;
ArrayList Table;
begin
for(int i⟵0; i<Table.size()-1; i++)
{
for (int j⟵i+1; j<Table.size(); j++)
{
if(EtopKMethod(Table.get(j), Table.get(m))=1)
{m⟵j;}
}
if(EtopKMethod(Table.get(m) ,Table.get(i))=1)
{ Tuple c ⟵ Table.get(i);
Table.set(i,Table.get(m));
Table.set(m,c); }
}
end
```

5 Experimental Study

In this section, we evaluate both algorithms from a performance point of view. We used a windows 10 operating system with 2.10 GHz CPU and 4 GB RAM. We also used Java programming language and NetBeans platform.

5.1 Data Sets

We used synthetic data sets with the following parameters (a) N the size of the database, (b) S the evidential score which is an interval of belief and plausibility $[Bel; PL]$ with BEl, PL $\in$ [0;1] and BEL $\leq$ PL[1].

To generate a synthetic evidential database, the used algorithm uses a procedure that generates a synthetic S. Indeed, the procedure computes randomly a fixed number of evidential scores in the interval $[0, 1]$. Then one of the algorithms (naive or optimized) are processed in order to compare intervals. Finally, a sorting function is used to provide the final complete ranking of all intervals. Note that each interval is associated to a specific and unique item in the evidential database. In our example, the item is a specific book.

Experiments showed interesting results from a performance point of view. In fact, we varied the database size parameter (N) from 10 to 3000. The execution time did not exceed 4 min and 50 s for both algorithms. Results are presented in Table 5. Both algorithms showed interesting results. Moreover, *OptETopK* gave better ones as shown in Fig. 2. For example, *OptETopK* ranked 1500 tuples in

[1] Bel and Pl are two functions defined in the object-relational implementation of evidential databases in [5].

Table 5. Impact of the database size for methods: NaiTopK and OptTopK

Tuples number (N)	Execution time (s)	
	NaiETopK method	OptETopK method
10	1	0
50	2	0
100	2	0
200	3	0
300	4	1
500	8	5
800	19	13
1000	33	28
1500	69	60
2000	125	121
3000	279	277

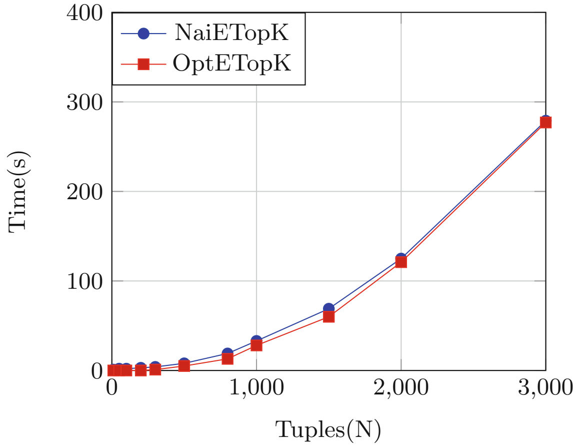


Fig. 2. Comparison of performance of NaiETopK and OptETopK

69 s against 60 s for $NaiETopKNote$. Note that complexity depends also on the intervals' nature generated randomly as detailed theoretically in Sect. 3.

6 Conclusion

Throughout this paper, we presented an implementation of the Evidential Top-k query, *ETop-k*. In fact, we proposed two algorithms *NaiETopK* and *OptETopK*. Both methods showed interesting results when we varied the database size but *OptETopK* showed best performance in practice as shown theoretically in [4].

The proposed implementation is an important achievement of the evidential Top-k querying fitting the semantics of returning the k most credible answers.

Other types of queries, in the evidential context, like aggregation, range, threshold remain as a promising future works.

A Appendix

Proof. *Complementarity:

$$P(S(R_i) < S(R_j)) = \frac{max(0, pl_j - bel_i) - max(0, bel_j - pl_i)}{(pl_i - bel_i) + (pl_j - bel_j)}$$

$$P(S(R_j) < S(R_i)) = \frac{max(0, pl_i - bel_j) - max(0, bel_i - pl_j)}{(pl_i - bel_i) + (pl_j - bel_j)}$$

$$P(S(R_i) < S(R_j)) + P(S(R_j) < S(R_i))$$

$$\begin{aligned} &= \frac{max(0, pl_j - bel_i) - max(0, bel_j - pl_i)}{(pl_i - bel_i) + (pl_j - bel_j)} \\ &+ \frac{max(0, pl_i - bel_j) - max(0, bel_i - pl_j)}{(pl_i - bel_i) + (pl_j - bel_j)} \\ &= \frac{max(0, pl_j - bel_i) - 0 + max(0, pl_i - bel_j) - 0}{(pl_i - bel_i) + (pl_j - bel_j)} \\ &= \frac{pl_j - bel_i + pl_i - bel_j}{pl_i - bel_i + pl_j - bel_j} = 1 \end{aligned}$$

$$P(S(R_i) < S(R_j)) + P(S(R_j) < S(R_i)) = 1$$

Property 1. **Transitivity

Let $S(R_i) = [bel_i; pl_i]$, $S(R_j) = [bel_j; pl_j]$ and $S(R_k) = [bel_k; pl_k]$ be three intervals. If $S(R_i) \succ S(R_j)$ and $S(R_j) \succ S(R_k)$ then $S(R_i) \succ S(R_k)$.

References

1. Bell, D.A., Guan, J.W., Lee, S.K.: Generalized union and project operations for pooling uncertain and imprecise information. Data Knowl. Eng. (DKE) **18**, 89–117 (1996)
2. Yaghlane, A.B., Denœux, T., Mellouli, K.: Elicitation of expert opinions for constructing belief functions. In: Uncertainty and Intelligent, Information Systems, pp. 75–88 (2008)
3. Bousnina, F.E., Bach Tobji, M.A., Chebbah, M., Liétard, L., Ben Yaghlane, B.: A new formalism for evidential databases. In: Esposito, F., Pivert, O., Hacid, M.-S., Raś, Z.W., Ferilli, S. (eds.) ISMIS 2015. LNCS (LNAI), vol. 9384, pp. 31–40. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25252-0_4

4. Bousnina, F.E., Chebbah, M., Bach Tobji, M.A., Hadjali, A., Ben Yaghlane, B.: On top-k queries over evidential data. In: 19th International Conference on Enterprise Information Systems (ICEIS), Porto, Portugal, vol. 1, pp. 106–113 (2017)
5. Bousnina F., Chebbah M., Bach Tobji M., Hadjali A. and Ben Yaghlane B.: Object-relational implementation of evidential databases. In: 1st International Conference on Digital Economy (ICDEc), La Marsa, Tunisia, pp. 80–87 (2016)
6. Cavallo, R., Pittarelli, M.: The theory of probabilistic databases. In: Proceedings of the 13th VLDB Conference, Brighton, UK, pp. 71–81 (1987)
7. Dempster, A.P.: Upper and lower probabilities induced by a multiple valued mapping. Ann. Math. Stat. **38**(2), 325–339 (1967)
8. Dempster, A.P.: A generalization of Bayesian inference. J. R. Stat. Soc. Ser. B **30**, 205–247 (1968)
9. Elmi, S., Benouaret, K., Hadjali, A., Bach Tobji, M.A., Ben Yaghlane, B.: Computing skyline from evidential data. In: Straccia, U., Calì, A. (eds.) SUM 2014. LNCS (LNAI), vol. 8720, pp. 148–161. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11508-5_13
10. Elmi, S., Benouaret, K., HadjAli, A., Bach Tobji, M.A., Ben Yaghlane, B.: Requêtes skyline en présence des données évidentielles. In: Extraction et Gestion des Connaissances (EGC), pp. 215–220 (2015)
11. Ennaceur, A., Elouedi, Z., Lefevre, E.: Multi-criteria decision making method with belief preference relations. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **22**(04), 573–590 (2014)
12. Fagin, R.: Combining fuzzy information from multiple systems. In: 15th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Montreal, Canada, pp. 216–226. ACM (1996)
13. Fagin, R.: Fuzzy queries in multimedia database systems. In: 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Seattle, WA, USA, pp. 1–10. ACM (1998)
14. Lee, S.K.: An extended relational database model for uncertain and imprecise information. In: 18th Conference on Very Large Data Bases (VLDB), Canada, pp. 211–220 (1992)
15. Lee, S.K.: Imprecise and uncertain information in databases: an evidential approach. In: 8th International Conference on Data Engineering (ICDE), Arizona, USA, pp. 614–621 (1992)
16. Prade, H., Testemale, C.: Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. Inf. Sci. **34**(2), 115–143 (1984)
17. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
18. Wang, Y.-M., Yang, J.-B., Dong-Ling, X.: A preference aggregation method through the estimation of utility intervals. Comput. Oper. Res. **32**(8), 2027–2049 (2005)