

A methods of neutrosophic logia to answer queries in Relational database

(1) Smita Rajpal, (2) M.N. Doja and (1) Ranjit Biswas

(1) Institute of Technology and Management, Gurgaon

(2) Jamia Millia University, New Delhi

Corresponding Author: Smita Rajpal, Institute of Technology and Management, Gurgaon

INTRODUCTION

Today Databases are Deterministic. An item belongs to the database is a probabilistic event, or a [tuple](#) is an answer to the query is a probabilistic event and it can be extended to all data models. Here we will discuss probabilistic relational data. Probabilistic relational Data are defined in two ways, Database is deterministic and Query answers are probabilistic or Database is probabilistic and Query answers are probabilistic.

Probabilistic relational databases have been studied from the late 80's until today. But today Application Need to manage imprecision's in data. Imprecision can be of many types: non-matching data values, imprecise queries, inconsistent data, misaligned schemas, etc.

The quest to manage imprecision's is equal to major driving force in the database community is the Ultimate cause for many research areas: data mining, semistructured data, and [schema matching](#), [nearest neighbor](#). Processing probabilistic data is fundamentally more complex than other data models. Some previous approaches sidestepped complexity. Now our implementation includes Ranking query answers. Since our Database is deterministic, The query returns a ranked list of tuples But our User interested in top-k answers. Sometimes we get the empty answers for the user queries in the deterministic database. For e.g.,

Here our Database will fail to Answer because of the imprecision in the query. But Using Ranking query using the [neutrosophic logic](#) we will get the answer. So to Answer this we must know the type of imprecision

Definition Ranking: Ranking is defined as Computing a [similarity score](#) between a tuple and the query,

Consider the query

Q = SELECT*

From R

Where A1= v1 and ... and Am = vm

Query is a vector: Q = (v1, ..., vm)

Tuple is a vector: T = (u1, ..., um)

Consider the applications: personalized search engines, shopping agents, logical user profiles, soft catalogs.

To answer the queries related with the above application Two approaches are given:

- * Qualitative \rightarrow Pare to semantics (deterministic)
- * Quantitative \rightarrow alter the query ranking

Definition: An imprecise attribute value $[t.sub.m]$ ($[a.sub.i]$) must be specified as a [discrete probability distribution](#) over D_i , that is $[t.sub.m]([a.sub.i]) = \{([z.sub.j], [P.sub.j]) \mid [z.sub.j] \in [D.sub.i] \text{ and } [P.sub.j] \in [0, 1]\}$ with $\sum [P.sub.j] = [\alpha.sub.im]$, $0 < [\alpha.sub.im] \leq 1$. ($[z.sub.j], [P.sub.j] \in [f.sub.vn](a)$).

This definition covers both interpretations of null values as well as the usual interpretation of imprecise data: If $[a.sub.im] = 1$, we certainly know that an attribute value exists and with $[a.sub.im] = 0$, we represent the fact that no value exists for this attribute. In the case of $0 < [o.sub.i] < 1$, $[o.sub.i]$, gives the probability that an attribute value exists: For example, someone who is going to have a telephone soon gave us his number, but we are not sure if this number is valid already. With imprecise values specified this way, their probabilistic indexing weight can be derived easily.

Definition probabilistic tuples: Let $R(A)$ be a relation scheme and let $t = ([V.sub.1]; \dots; [V.sub.n])$ be a tuple of cases of the relation scheme R . For each V_i , let $[V.sub.I]$ be the set of the $[v.sub.j] = ([a.sub.j], [l.sub.j], [u.sub.j]; [p.sub.j])$ such that $([a.sub.j]; [l.sub.j]; [u.sub.j]) \in [V_i]$, where $[p.sub.j]$ is the path associated with a_j . A probabilistic tuple $[t.sub.0] = ([v'.sub.1]; \dots; [v'.sub.n])$ is an element of the [cartesian product](#) $[V.sub.1] \times \dots \times [V.sub.n]$.

By $[A.sub.i].l$, $[A.sub.i].u$ and $A_i.p$ we denote $[l.sub.j]$, $[u.sub.j]$ and $[p.sub.j]$ associated with a generic value of $[A.sub.i]$ in a given probabilistic tuple, respectively.

Definition: Probabilistic relation: A probabilistic relation r of the scheme $R(A)$ is a [finite set](#) of probabilistic tuples of $R(A)$. By $\text{domr}([A.sub.i])$ we will denote the set of all values of the attribute $[A.sub.i]$ in the relation r .

Definition: Probabilistic database: A probabilistic database of the database scheme $R = \{[R.sub.1]([A.sub.1]), \dots; [R.sub.m]([A.sub.m])\}$ is a finite set of probabilistic relations $r = ([r.sub.1], \dots, [r.sub.m])$, where each r_i is a relation of the scheme $[R.sub.i]([A.sub.i])$.

In order to avoid probabilistic ambiguities we assume that in each initial relation there cannot be identical tuples.

So the failure of the [RDBMS](#) due to the presence of imprecise constraints in the query

[predicate](#) which can not be tackled due to the limitation of the grammar in standard query languages which work on crisp environment only. But this type of queries is very common in business world and in fact more frequent than grammatical-queries, because the users are not always expected to have knowledge of [DBMS](#) and the query languages.

Consequently, there is a genuine necessity for the different large size organizations, especially for the industries, companies having world wide business, to develop such a system which should be able to answer the users queries posed in natural language, [irrespective of](#) the QLs and their grammar, without giving much botheration to the users. Most of these type of queries are not crisp in nature and involve predicates with fuzzy (or rather vague) data, fuzzy/vague hedges (with concentration or [dilation](#)). Thus, this type of queries is not strictly confined within the domains always. The corresponding predicates are not hard as in crisp predicates. Some predicates are soft because of vague/fuzzy nature and thus to answer a query a hard match is not always found from the databases by search, although the query is nice and very real and should not be ignored or replaced [according to](#) the business policy of the industry. To deal with uncertainties in searching match for such queries, [fuzzy logic](#) and rather vague logic (1) and Neutrosophic logic by Smarandache (7) will be the appropriate tool.

In this study we propose a new type of searching techniques called as neutrosophic search which is a combination of -Neutrosophic-equality search and neutrosophic [proximity search](#) by using [Neutrosophic set](#) theory to meet the predicates posed in natural language in order to answer imprecise queries of the users. Thus it is a kind of an intelligent search for match in order to answer imprecise queries of the lay users. We call this method by Neutrosophic search which is a combination of -Neutrosophic-equality search and neutrosophic proximity search.

Our method, being an intelligent soft-computing method, will support the users to make and find the answers to their queries without iteratively refining them by trial and error which is really boring and sometimes it seriously effects the interest (mission and vision) of the organization, be it an industry, or a company or a hospital or a private academic institution etc. to list a few only out of many. Very often the innocent (having a lack of DBMS knowledge) users go on refining their queries in order to get an answer. The users are from different corner of the academic world or business world or any busy world. For databases to support imprecise queries, our intelligent system will produce answers that closely match the queries constraints, if des not exactly. This important issue of closeness can not be addressed with the crisp mathematics. That is why we have used the Neutrosophic tools.

Theory of neutrosophic set: In the real world there are vaguely specified data values in many applications, such as sensor information, Robotics etc. [Fuzzy set](#) theory has been proposed to handle such vagueness by generalizing the notion of membership in a set. Essentially, in a Fuzzy Set (FS) each element is associated with a point-value selected from the [unit interval](#) [0,1], which is termed the grade of membership in the set. A Vague Set (VS), as well as an Intuitionistic Fuzzy Set (IFS), is a further generalization of an FS. Now take an example, when we ask the opinion of an expert about certain statement, he or she may say that the possibility that the statement is true is between 0.5 and 0.7 and the statement is false is between 0.2 and 0.4 and the degree that he or she is not sure is between 0.1 and 0.3. Here is another example, suppose there are 10 voters during a voting process. In time t1, three vote yes, two vote no and five are

undecided, using neutrosophic notation, it can be expressed as $x(0.3,0.5,0.2)$; in time t_2 , three vote yes, two vote no, two give up and three are undecided, it then can be expressed as $x(0.3,0.3,0.2)$. That is beyond the scope of the intuitionistic fuzzy set. So, the notion of neutrosophic set is more general and overcomes the aforementioned issues. In neutrosophic set, [indeterminacy](#) is quantified explicitly and truth membership, indeterminacy-membership and falsity-membership are independent. This assumption is very important in many applications such as information fusion in which we try to combine the data from different sensors. Neutrosophy was introduced by Smarandache (7).

Neutrosophic set is a powerful general formal framework which generalizes the concept of the classic set, fuzzy set (2), Vague set (1) etc.

A neutrosophic set A defined on universe U . $x = x(T, I, F) A$ with T, I and F being the real standard or non-standard subsets of $]0-,1+[$, T is the degree of truth-membership of A , I is the degree of indeterminacy membership of A and F is the degree of falsity-membership of A .

Definition: A Neutrosophic set A of a set U with $[t.sub.A](u)$, $[f.sub.A](u)$ and $[I.sub.A](u)$, [for all] u [member of] U is called the -Neutrosophic set of U , where [member of] $[0,1]$.

Definition: A Neutrosophic number (NN) is a Neutrosophic set of the set R of real numbers.

Operations with neutrosophic sets: We need to present these set operations in order to be able to introduce the neutrosophic connectors.

Let S_1 and S_2 be two (unidimensional) real standard or non-standard subsets, then one defines.

Addition of sets: $S_1 [direct sum] S_2 = \{x|x = s_1+s_2, \text{ where } s_1[\text{member of}]S_1 \text{ and } s_2[\text{member of}]S_2\}$, with $\inf S_1 [direct sum] S_2 = \inf S_1 + \inf S_2$, $\sup S_1 [direct sum] S_2 = \sup S_1 + \sup S_2$ and as some particular cases, we have $\{a\} [direct sum] S_2 = \{x|x = a+s_2, \text{ where } s_2[\text{member of}]S_2\}$ with $\inf \{a\} [direct sum] S_2 = a + \inf S_2$, $\sup \{a\} [direct sum] S_2 = a + \sup S_2$.

Subtraction of sets: $S_1 S_2 = \{x|x = s_1-s_2, \text{ where } s_1[\text{member of}]S_1 \text{ and } s_2[\text{member of}]S_2\}$.

For real positive subsets (most of the cases will fall in this range) one gets $\inf S_1 S_2 = \inf S_1 - \sup S_2$, $\sup S_1 S_2 = \sup S_1 - \inf S_2$ and as some particular cases, we have $\{a\} S_2 = \{x|x = a-s_2, \text{ where } s_2[\text{member of}]S_2\}$, with $\inf \{a\} S_2 = a - \sup S_2$, $\sup \{a\} S_2 = a - \inf S_2$.

Multiplication of sets: $S_1 [direct sum] S_2 = \{x|x = s_1.s_2, \text{ where } s_1[\text{member of}]S_1 \text{ and } s_2[\text{member of}]S_2\}$.

For real positive subsets (most of the cases will fall in this range) one gets $\inf S_1 [direct sum] S_2 = \inf S_1 . \inf S_2$, $\sup S_1 [direct sum] S_2 = \sup S_1 [direct sum] \sup S_2$ and, as some particular cases, we have $\{a\} [direct sum] S_2 = \{x|x = a [direct sum] s_2, \text{ where } s_2[\text{member of}]S_2\}$, with $\inf \{a\} [direct sum] S_2 = a * \inf S_2$, $\sup \{a\} [direct sum] S_2 = a [direct sum] \sup S_2$.

Division of sets by a number: Let $k[\text{member of}]R^*$ then $S_1/k = \{x|x = s_1/k, \text{ where } s_1[\text{member}$

of S_1 }.

Neutrosophic logic connectors: One uses the definitions of [neutrosophic probability](#) and neutrosophic set operations. Similarly, there are many ways to construct such connectives according to each particular problem to solve; here we present the easiest ones: One notes the neutrosophic logic values of the propositions A1 and A2 by $NL(A1) = (T1, I1, F1)$ and $NL(A2) = (T2, I2, F2)$ respectively.

For all neutrosophic logic values below: if, after calculations, one obtains numbers <0 or >1 , one replaces them $[.sup.-]0$ or $[1.sup.+]$ respectively.

Negation: $NL([\text{logical not}]A1) = (\{[1.sup.+]\}_T1, \{[1.sup.+]\}_I1, \{1+\}_F1)$

Conjunction: $NL(A1[\text{and}]A2) = (T1[\text{direct sum}]T2, I1[\text{direct sum}]I2, F1[\text{direct sum}]F2)$. (And, in a similar way, generalized for n propositions.)

Implication: $NL(A1[\text{left and right arrow}]A2) = (\{[1.sup.+]\}_T1[\text{direct sum}]T2, \{[1.sup.+]\}_I1[\text{direct sum}]I2, \{[1.sup.+]\}_F1[\text{direct sum}]F2)$.

Neutrosophic relation: A neutrosophic relation R on scheme $[\text{summation}]$ is any subset of $([\text{summation}]) \times [0, 1] \times [0, 1]$. For any t $[\text{member of}] ([\text{summation}])$, we shall denote an element of R as $[\text{??}]t, [R(t).sup.+], [R(t).sup.-] [\text{??}]$, where $[R(t).sup.+]$ is the belief factor assigned to t by R and $[R(t).sup.-]$ is the doubt factor assigned to t by R. Let $V([\text{summation}])$ be the set of all neutrosophic relations on $[\text{summation}]$.

Consistent neutrosophic relation: A neutrosophic relation R on scheme $[\text{summation}]$ is consistent if $R [(t).sup.+]+[R(t).sup.-] [\text{less than or equal to}]1$, for all t $[\text{member of}] ([\text{summation}])$. Let $C([\text{summation}])$ be the set of all consistent neutrosophic relations on $[\text{summation}]$. R is said to be complete if $[R(t).sup.+]+[R(t).sup.-] [\text{greater than or equal to}]1$, for all t $[\text{member of}] ([\text{summation}])$. If R is both consistent and complete, i.e., $[R(t).sup.+]+[R(t).sup.-] = 1$, for all t $[\text{member of}] ([\text{summation}])$, then it is a total neutrosophic relation and let $T([\text{summation}])$ be the set of all total neutrosophic relations on $[\text{summation}]$.

A note on interval mathematics: Dealing with the mathematics of Neutrosophic set theory, the crisp theory of interval mathematics is sometimes useful. In this section, we [recollect](#) some basic notions of interval mathematics. For our purpose in this paper, we need to consider intervals of non-negative real numbers only.

Let $[I.sub.1] = [a,b]$ and $[I.sub.2] = [c,d]$ be two intervals of non-negative real numbers. A point valued non-negative real number r also can be viewed, for the sake of arithmetic, as an interval $[r,r]$.

Some [algebraic](#) operations:

* Interval Addition: $[I.sub.1] + [I.sub.2] = [a+c, b+d]$

- * Interval [Subtraction](#): $[I.sub.1] - [I.sub.2] = [a-c, b-d]$
- * Interval Multiplication: $[I.sub.1] * [I.sub.2] = [ac, bd]$
- * Interval Division: $[I.sub.1]/[I.sub.2] = [a/d, b/c]$, when $c, d \neq 0$
- * [Scalar Multiplication](#) : $k \cdot [I.sub.1] = [ka, kb]$

Ranking of intervals: Intervals are not ordered. [Owing to](#) this major weakness, there is no universal method of ranking a finite (or infinite) number of intervals. But in real life problems dealing with intervals, we need to have some tactic to rank them in order to arrive at some conclusion. We will now present a method of ranking of intervals, which we shall use in our work here in subsequent sections. We consider a decision maker (or any intelligent agent like a company manager, a factory supervisor, an intelligent robot, an intelligent network, etc.) who makes a pre-choice of a decision parameter [member of] $[0,1]$. The intervals are to be ranked once the decision-parameter is fixed. But ranking may differ if the pre-choice is renewed.

Definition: α -value of an interval: Let $J = [a, b]$ be an interval. The α -value of the interval J is a non-negative real number J_α , given by $J_\alpha = (1-\alpha) \cdot a + \alpha \cdot b$.

Clearly, $0 \leq J_\alpha \leq 1$ and for $\alpha = 0$ $J_\alpha = a$, which signifies that the decision-maker is pessimistic and also for $[\alpha] = 1$ $[J.sub.[\alpha]] = b$ which signifies that the decision-maker is optimistic. For $[\alpha] = 0.5$ it is the arithmetic-mean to be chosen usually for a moderate decision.

Comparison of two or more intervals we will do here on the basis of α -values of them. If the value of $[\alpha]$ is renewed, the comparison-results may change. The following definition will make it clear. Now Author is proposing α -Neutrosophic-equality search.

$[\alpha]$ -Neutrosophic equality search: Consider the Students database as described in section-1. Consider a normal type of query like Project (Student_Name) Where AGE = approximately 20.

The standard [SQL](#) is unable to provide any answer to this query as the search for an exact match for the predicate will fail. The value approximately 20 is not a precise data. Any data of type approximately x , little more than x , slightly less than x , much greater than x etc., are not precise or crisp, but they are Neutrosophic numbers (NN). Denote any one of them, say the neutrosophic number approximately x by the notation $I(x)$. We know that a Neutrosophic number is a Neutrosophic Set of the real numbers. Clearly for every member a [member of] $\text{dom}(AGE)$, there is a membership value $[t.sub.I(x)](a)$ proposing the degree of equality of this crisp number a with the quantity approximately x and a non-membership value $[f.sub.I(x)](a)$ proposing the degree of non-equality. Thus, in neutrosophic philosophy of samarandech, every element of $\text{dom}(AGE)$ satisfies the predicate AGE = approximately 20 upto certain extent and does not satisfy too, upto certain extent. But we will restrict ourselves to those members of $\text{dom}(AGE)$ which are α -neutrosophic-equal, the concept of which we will define below. Any imprecise predicate of type AGE = approximately 20, or of type AGE = young (where the attribute value young is not a member of the $\text{dom}(AGE)$), is to be called by Neutrosophic-predicate and a query involving Neutrosophic-predicate is called to be a Neutrosophic-query.

Definition: Consider a choice-parameter $[\text{member of}][0,1]$. A member a of $\text{dom}(\text{AGE})$ is said to be $[\alpha]$ -Neutrosophic-equal to the quantity approximate x if a $[\text{member of}] I(x)$, where $I(x)$ is the α -cut of the Neutrosophic number $I(x)$. The degree or amount of this equality is measured by the interval $[\text{m.sub.}I(x)](a) = [[\text{t.sub.}I(x)](a), [1-\text{f.sub.}I(x)](a)]$. Denote the collection of all such $[\alpha]$ -neutrosophic-equal members from $\text{dom}(\text{AGE})$ by the notation $\text{AGE}(x)$, which is a subset of $\text{dom}(\text{AGE})$. If $\text{AGE}(x)$ is not a null-set or singleton, then the members can be ranked by ranking their corresponding degrees of equality.

Definition: Consider a choice value $[\text{member of}][0,1]$. At level of choice, for every element a of $\text{AGE}(x)$, the truth-value $t([\text{p.sub.}1], [\text{p.sub.}2])$ of the matching of the predicate $[\text{p.sub.}1]$: given by $\text{AGE} = \text{approximately } x$ with the predicate $[\text{p.sub.}2]$: $\text{AGE} = a$ is equal to the α -value of the interval $[\text{m.sub.}I(x)](a)$.

Neutrosophic- proximity search: The notion of $[\alpha]$ -neutrosophic-equality search as explained above is appropriate while there is an Neutrosophic-predicate in the query involving NNs. But there could be a variety of vague predicates existing in a Neutrosophic query, many of them may involve Neutrosophic hedges (including concentration/dilation) like good, very good, excellent, too much tall, young, not old, etc. In this section we present another type of search for finding out a suitable match to answer imprecise queries. In this search we will use the theory of neutrosophic-proximity relation (4), (5). We know that a neutrosophic-proximity relation on a universe U is a neutrosophic relation on U which is both neutrosophic-reflexive and neutrosophic-symmetric.

Consider the Students database as described in section-1 and a query like Project (Student_Name) Where Eye-Color = dark-brown.

The value/data dark-brown is not in the set $\text{dom}(\text{Eye-Color})$. Therefore a crisp search will fail to answer this. The objective of this research work is to overcome this type of drawbacks of the classical SQL. For this we notice that there may be one or more members of the set $\text{dom}(\text{Eye-Color})$ which may closely match the eye-color of brown or dark-brown.

Consider a new universe given by $W = \text{dom}(\text{EYE-COLOR}) \cup \{\text{dark-brown}\}$.

Propose a Neutrosophic-proximity relation R over W . Choose a decision-parameter $[\text{member of}][0, 1]$. We propose that search is to be made for the match e $[\text{member of}]\text{dom}(\text{EYE-COLOR})$ such that $[\text{t.sub.}R](\text{dark-brown}, e) \geq [\alpha]$.

(It may be mentioned here that the condition $[\text{t.sub.}R](\text{dark-brown}, e)$ does also imply the condition $[\text{f.sub.}R](\text{dark-brown}, e) \leq 1 - [\alpha]$).

We say that e is a close match with dark-brown with the degree or amount of closeness being the interval $[\text{m.sub.}dark-brown](e)$ given by $[\text{m.sub.}dark-brown](e) = [[\text{t.sub.}R](\text{dark-brown}, e), 1 - [\text{f.sub.}R](\text{dark-brown}, e)]$.

At $[\beta]$ level of choice, the truth-value $t([\text{p.sub.}1], [\text{p.sub.}2])$ of the matching of the predicate

[p.sub.1]: given by EYE-COLOR = dark-brown with the predicate [p.sub.2]: AGE = e is equal to the [beta]-value of the interval [m.sub.dark-brown](e).

Neutrosophic-search: In this section we will now present the most generalized method of search called by Neutrosophic-search. The Neutrosophic-search of matching is actually a combined concept of [alpha]-neutrosophic-equality search, neutrosophic-proximity search and crisp search.

For example, consider a query like Project (Student_Name)

Where (Sex = M, Eye-Color = dark-brown, Age= approximately 20).

This is a neutrosophic-query.

To answer such a query, matching is to be searched for the three predicates [p.sub.1], [p.sub.2] and [p.sub.3] given by:

* [p.sub.1]: SEX = M,

* [p.sub.2]: EYE-COLOR = dark-brown and

* [p.sub.3] : AGE = approximately 20

where [p.sub.1] is crisp and [p.sub.2], [p.sub.3] are neutrosophic(imprecise).

Clearly, to answer this query the proposed neutrosophic search method is to be applied, because in addition to crisp search, both of [alpha]-neutrosophic-equality search and neutrosophic-proximity search will be used to answer this query. The truth-value of the matching of the conjunction p of [p.sub.1], [p.sub.2] and [p.sub.3] will be the product of the individual truth values, (where it is needless to mention that for crisp match the truth-value will be exactly 1). There could be a multiple number of answers to this query and the system will display all the results ordered or ranked according to the truth-values of p.

It is obvious that the neutrosophic-search technique for predicate-matching reduces to a new type of fuzzy-search technique as a special case.

CONCLUSION

In this study, we have introduced a new method to answer imprecise queries of the lay users from the databases (details of the databases may not be known to the lay (users)). We have adopted Neutrosophic set tool to solve the problem of searching an exact match or a close match (if an exact match is not available) of the predicates so that we will be able to get the answer of evidence for you (i.e., exact/truth match) and evidence against you (i.e., false match) and the undecidability (i.e., indeterminacy) This is a complete new Method of Answering Queries based on Neutrosophic logic.

REFERENCES

- (1.) Gau, W.L. and D.J. Buehrer, 1993. Vague sets. [IEEE](#) Trans. Syst., Man and Cybernetics, 23:610-614. ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=5923&isYear=1993&count=32&page=1
- (2.) Atanassov, K., 1986. Intuitionistic fuzzy sets. Fuzzy Sets System, 20: 87-96. linkinghub.elsevier.com/retrieve/pii/016501149400286G
- (3.) Atanassov, K., 2000. Intuitionistic Fuzzy Sets: Theory and Applications. Physica-Verlag, [New York](#). ISBN 37908 1425 3
- (4.) Bustince, H. and P. Burillo, 1996. Vague sets are intuitionistic fuzzy sets. Fuzzy Sets Syst., 79: 403-405. portal.acm.org/citation.cfm?id=241549
- (5.) Chiang D., L.R. Chow and N. Hsien, 1997. Fuzzy information in extended fuzzy relational databases. Fuzzy Sets Systems, 92: 1-10. www.elsevier.com/locate/fss
- (6.) Barbara, D., H. Garcia-Molina and D. Porter, 1992. The management of probabilistic data. IEEE Trans. Knowl. Data Eng., 4; 487-502.
- (7.) Smarandache, F. 2002. A unifying field in logics: Neutrosophic field. Multiple Valued Logic Int. J., 8: 385-438. www.gallup.unm.edu/~smarandache/eBook-Neutrosophics2.pdf
- (8.) Biazzo, V. and A. Gilio, 2000. A generalization of the [fundamental theorem](#) of [de Finetti](#) for imprecise [conditional probability](#) assessments. Int. J. Approximate Reasoning. Volume 24, Number 2-3, MAY 2000 linkinghub.elsevier.com/retrieve/pii/S0888613X00000384
- (9.) Biazzo V., A. Gilio and G. Sanfilippo, 1999. Efficient Coherence Checking and Propagation of [Imprecise Probability](#) Assessments. In Proceedings IPMU-2000.
- (10.) Cavallo, R. and M. Pittarelli, 1987. The theory of probabilistic database. In Proceedings of the 13th VLDB Conference, Brighton, England, Page numbers 71-78.
- (11) Codd, E.F., 1979. Extending the database [relational model](#) to capture more meaning. ACM Trans. Database Syst., 4: 394-405.
- (12) Coletti, G., 1994. Coherent numerical and [ordinal](#) probabilistic assessments. IEEE Trans. Syst. Man Cybernetics, 24: 1747-1754. www.ieeexplore.ieee.org
- (13.) Coletti, G. and R. Scozzafava, 1996. Characterization of coherent conditional probabilities as a tool for their assessment and extension. J. Uncertainty, Fuzziness Knowledge-Based Syst., 4: 103-127. portal.acm.org/citation.cfm?id=766838
- (14.) G. Coletti and R. Scozzafava, Exploiting zero probabilities, in: Proc. of EUFIT '97, Aachen, Germany (ELITE foundation, 1997) pp. 1499-1503. 5th European Congress on Intelligent Techniques and Soft Computing, September 08. -11, 1997.

www.eufit.org/proceedings/97/volume_2.html

(15.) Dey, D. and S. [Sarkar](#), 1996. A probabilistic relational model. ACM Trans. Database Syst., 21: 394-405.

(16.) Re, C., N. Dalvi and D. Suciu, 2007. Efficient top-k query evaluation on probabilistic data. In: Proceedings of [ICDE](#)(IEEE International Conference on Data Engineering),2007. pages 886-895. www.icde2007.org