A METHOD OF NEUTROSOPHIC LOGIC TO ANSWER QUERIES IN RELATIONAL DATABASE

The Neutrosophic Search, an application of Neutrosophic Logic

Smita Rajpal, M. N. Doja, Ranjit Biswas

Abstract.

In this study, the authors propose a new method of intelligent search called **Neutrosophic search** to find the most suitable match for the predicates to answer any imprecise query made by the database users.

It is also to be mentioned that the Neutrosophic-search method could be easily incorporated in the existing commercial query languages of DBMS to serve the lay users better.

Authors suggest a new method called **Neutrosophic-equality Search** to answer the queries of Relational database based on ranks.

I. 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 it is discussed **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 imprecisions in data.

Imprecision can be of many types: **non-matching data values, imprecise queries, inconsistent data, misaligned schemas**, etc.

I. INTRODUCTION – cont.

The quest to manage imprecisions is equal to major driving force in the database community. It 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.

Present implementation includes ranking query answers.

Since our Database is deterministic, the query returns a ranked list of tuples.

Sometimes, we get the empty answers for the user queries in the deterministic database.

I. INTRODUCTION – cont.

For e.g.,

```
Try to buy a house in Seattle,
      Select*
      From Houses
      Where bedrooms = 4
             And style = craftsman
             And district = View Ridge
             And price<400000
```

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** → pare to semantics (deterministic);
- ❑ **Quantitative** → alter the query ranking.

Definition: An imprecise attribute value $t_m(a_i)$ must be specified as a discrete probability distribution over Di, that is $t_m(a_i) = \{(z_j,P_j) \backslash z_j \in D_i \text{and } P_j \in [0, 1]\}$ with $\Sigma P_j = \alpha_{im}$, $0 <= \alpha_{im} <= 1$. $(z_j,P_j) \in f_{vn}(a)$.

Ranking is defined as computing a similarity score between a tuple and the query.

This definition covers both interpretations of null values as well as the usual interpretation of imprecise data.

If $a_{im} = 1$, we certainly know that an attribute value exists and with $a_{im} = 0$, we represent the fact that no value exists for this attribute. In the case of $0 < o_i, < 1$, o_i, it 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_l; : : : ;V_n) be a tuple of cases of the relation scheme R.

For each V_i, let V_l be the set of the v_j = (a_j, l_j, u_j, p_j) such that (a_j ; l_j ; u_j) ∈ V_i, where p_j is the path associated with a_j.

A probabilistic tuple t_0 = (v_l'; : : : ; v_n') is an element of the cartesian product V_l× … ×V_0.

By A_i. l, A_i.u and A_i.p we denote l_j, u_j and p_j associated with a generic value of A_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 domr (A_i) we will denote the set of all values of the attribute A_i in the relation r.

Definition: Probabilistic Database.

A probabilistic database of the database scheme R = {$R_1(A_1)$, : : : ;R_m (A_m)} is a finite set of probabilistic relations r = (r_1,……, r_m), where each r_i is a relation of the scheme $R_i(A_i)$.

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.

[1] Gau, W. L. and Buehrer, D. J., 1993. Vague sets. IEEE Trans. Syst., Man and Cybernetics, 23: 610-614.
[7] Smarandache, F. 2002. A unifying field in logics: Neutrosophic filed. Multiple Valued Logic Int. J., 8: 385-438. www.gallup.unm.edu/~smarandache/eBookNeutrosophics2.pdf

Here, it is proposed a new type of searching techniques 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 intelligent search for match in order to answer imprecise queries of the lay users.

This method is called **neutrosophic search** which is a combination of **neutrosophic-equality search** and **neutrosophic proximity search**.

The 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, the intelligent system will produce answers that closely match the queries constraints. This important issue of closeness can not be addressed with the crisp mathematics. That is why neutrosophic tools are used.

α-Neutrosophic Equality Search.

Consider the Students database as described above.

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 ∈ dom (AGE), there is a membership value $t_{I(x)}(a)$ proposing the degree of equality of this crisp number a with the quantity approximately x and a nonmembership value $f_{I(x)}(a)$ proposing the degree of nonequality.

Thus, in neutrosophic philosophy, every element of dom (AGE) satisfies the predicate AGE = approximately 20 up to certain extent and does not satisfy too, up to certain extent.

But we will restrict ourselves to those members of 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 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 $\alpha \in [0,1]$. A member of a of dom (AGE) is said to be α-Neutrosophic-equal to the quantity approximate x if $a \in I_\alpha(x)$, where $I_\alpha(x)$ is the α-cut of the Neutrosophic number I(x). The degree or amount of this equality is measured by the interval $m_{I(x)}(a) = [t_{I(x)}(a), 1-f_{I(x)}(a)]$. Denote the collection of all such α-neutrosophic-equal members from dom (AGE) by the notation $AGE_\alpha(x)$, which is a subset of dom (AGE). If $AGE_\alpha(x)$ 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 $\beta \in [0,1]$. At β level of choice, for every element a of $AGE_\alpha(x)$, the truth-value $t(p_1, p_2)$ of the matching of the predicate p_1: given by AGE = approximately x with the predicate p_2: AGE = a is equal to the β-value of the interval $m_{I(x)}(a)$.

Neutrosophic Proximity Search.

The notion of α-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.

This is another type of search for finding out a suitable match to answer imprecise queries. In this search, the theory of neutrosophic-proximity relation is used.

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 previously described and a query like

Project (Student_Name)
Where Eye-Color = dark-brown.

The value/data dark-brown is not in the set dom (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 dom (Eye-Color) which may closely match the eyecolor of brown or dark- brown.

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

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

(It may be mentioned here that the condition t_R(dark-brown,e) $\geq \alpha$ does also imply the condition f_R(dark-brown,e) $\leq$ 1- α).

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

At β level of choice, the truth-value t (p_1, p_2) of the matching of the predicate p_1: given by EYE-COLOR = dark-brown with the predicate p_2: AGE = e is equal to the β-value of the interval $m_{dark\text{-}brown}(e)$.

Neutrosophic Search.

The neutrosophic-search of matching is actually a combined concept of 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_1, p_2 and p_3 given by:

- p1: SEX = M,
- p2: EYE-COLOR = dark-brown, and
- p3 : AGE = approximately 20

where p_1 is crisp and p_2, p_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 α-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_1, p_2 and p_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.

A new method was introduced here to answer imprecise queries of the lay users from the databases - details of the databases may not be known to the lay (users).

Neutrosophic set tool was adopted 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 completely new method of answering queries based on neutrosophic logic.

REFERENCES

1. Gau, W. L. and Buehrer, D. J., 1993. Vague sets. IEEE Trans. Syst., Man and Cybernetics, 23: 610-614.
2. Atanassov, K., 1986. Intuitionistic fuzzy sets. Fuzzy Sets System, 20: 87-96.
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.
5. Chiang D., L.R. Chow and N. Hsien, 1997. Fuzzy information in extended fuzzy relational databases. Fuzzy Sets Systems, 92: 1-10.
6. Barbara, D., H. Garcia-Molina and D. Porter, 1992. The management of probabilistic data. IEEE Trans. Knwl. Data Eng., 4; 487-502.
7. Smarandache, F. 2002. A unifying field in logics: Neutrosophic filed. Multiple Valued Logic Int. J., 8: 385-438. www.gallup.unm.edu/~smarandache/eBookNeutrosophics2.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
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 VLBDB Conference, Brighton, England, 71-78.

REFERENCES

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 KnowledgeBased Syst., 4: 103-127.
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 1997.
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 topk query evaluation on probabilistic data. In: Proceedings of ICDE (IEEE International Conference on Data Engineering), 2007, 886–895. www.icde2007.org