Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 12, December-2020

Extracting Fuzzy Functional Dependencies using Information Theoretic Measures

Dr. K. Amshakala
Associate Professor, Coimbatore Institute of Technology,
Coimbatore, Tamil Nadu, India

***Abstract*:- In today's information age, data is the most valuable asset that an organization can possess. In a typical business organization, data is generated in multiple systems and it has become essential to extract meaningful information from the raw data collected from these multiple data sources to set the stage for business analysis and decision making. Extracting dependency constraints from databases is also crucial for data management and database reverse engineering. All the dependency constraints are not enforced when the database is modeled. Many conceptual dependencies are hidden in data values and are to be extracted explicitly. The extraction process is called dependency discovery which aims to find all dependencies satisfied by the existing data. It is not easy to discover perfect functional dependencies in a database because one single exception in equality of attribute values violates the dependency. But indeed, if the number of exceptions is not very high, such functional dependencies with exceptions may represent some interesting patterns hidden in data. Functional Dependencies that include attributes whose domain is quantified using fuzzy logic are called as fuzzy functional dependencies. This paper describes an information theory based method to extract fuzzy functional dependencies.**

Keywords:- Fuzzy Logic, Fuzzy Functional Dependency, Dependency Discovery, Information Theory

I. INTRODUCTION

FDs with more expressiveness are required to specify constraints in real-world data that are often imprecise or non-deterministic. All real data cannot be precise because of their fuzzy nature. In general, based on the data type of an attribute domain, attributes are classified as either crisp or fuzzy. An attribute with precise data value is called crisp attribute. For example, Name, City and so on. are attributes with crisp values. An attribute with its data values expressed as fuzzy set is called a fuzzy attribute. For example, Age, Salary, Price, Grade and so on. are fuzzy attributes. Consequently, for comparing attributes of a relation that has both crisp and fuzzy data, typical equality logic is not suitable. Fuzzy set theory and fuzzy logic proposed by Zadeh(1968) provide mathematical framework to deal with imprecise information.

Fuzzy Functional Dependencies

Extracting fuzzy functional dependencies (FFDs) helps us to extract meaningful fuzzy rules from the dataset that are hidden otherwise. Fuzzy rules help in matching attributes using similarity metrics rather than equality functions and can be used as matching rules in entity matching on uncertain data. Such fuzzy rules are exploited in decision making and are also used in medical field for analyzing various test reports.

Preliminaries

The basic definitions required to understand fuzzy functional dependencies are discussed in this Section.

Fuzzy Logic

Fuzzy logic is a form of many-valued logic, which deals with reasoning that is approximate rather than fixed and exact. In contrast to the traditional logic theory, where binary sets have two-valued logic: true or false, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false.

Fuzzy Set

Fuzzy sets are sets whose elements have degrees of membership. A fuzzy set is a pair (U, m), where U is a set and m is a function, mapping every element of U with a value in the interval 0 to 1 i.e U(x) →[0,1]. For each x $\in$ U, the value m(x) is called the degree of membership of x in (U, m). For a finite set U={x_1,x_2,...x_n}, the fuzzy set (U, m) is often denoted by { m(x_1)/x_1, ...m(x_n)/x_n }.

Each element of the fuzzy set has an associated degree of membership based on the membership function linked with the attribute domain. For any set X, the membership function usually denoted by $\mu(X)$ is any function from X to the real unit interval [0,1]. For an element x of X, the value $\mu_X(x)$ is called the membership degree of x in the fuzzy set X.

Membership Degree

The membership degree $\mu_X(x)$ quantifies the grade of membership of the element x to the fuzzy set X. The degree of membership is a real number between zero and one, and measures the extent to which the element belongs to the fuzzy set. The value 0 means that x is not a member of the fuzzy set and the value 1 means that x is fully a member of the fuzzy set. The values between 0 and 1 characterize fuzzy members, which belong to the fuzzy set only partially. A typical membership function is shown in Figure 1.

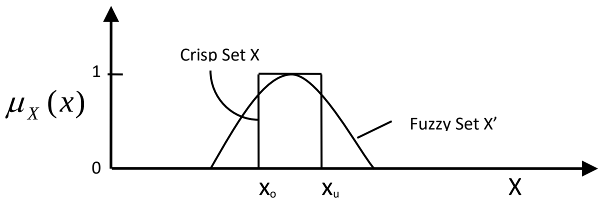

Figure 1. Fuzzy Membership Function

Fuzzification

The process of transforming crisp values into grades of membership for each fuzzy set is called fuzzification. In a relational table fuzzy attributes are also represented as crisp values. A fuzzy attribute is represented using multiple fuzzy sets based on the linguistic variables that are relevant to the attribute domain. Fuzzy sets are associated with membership functions and they allow the fuzzification of the crisp values of the attributes by estimating the degree of membership with respect to a fuzzy set. Two elements of a fuzzy set are called nearer only if their membership degree is above a specific threshold. For example, two elements $x_1, x_2 \in X$ are said to be $\theta - nearer$ if $\mu_X(x_1)$ and $\mu_X(x_2) \geq \theta$ where θ represents the membership threshold. If the two elements x_1 and x_2 are $\theta - nearer$ then they are said to be similar and this similarity is represented as $x_1 \approx_\theta x_2$.

Usually, a functional dependency, denoted by $X \rightarrow Y$, expresses that a function exists between the two sets of attributes X and Y, and it can be stated as follows: for any pair of tuples t_1 and t_2, if t_1 and t_2 share a common value on X, they also have the same value on Y. Such a statement can be extended along different lines and fuzzy sets have been used in various ways, among which:

- The universal quantifier "for any pair" is weakened into "almost all".
- The strict equality is replaced by a resemblance relation.
- Precise values are rewritten using linguistic labels related to the attribute.
- The values taken by the sets of attributes X and Y may be imprecise.

It appeared that these extended functional dependencies were not really able to capture redundancy. Hence, they are not interesting for database modeling, but could be used to represent rules or properties in the context of data mining.

Fuzzy Functional Dependency

A fuzzy functional dependency, denoted by $X \rightarrow_\theta Y$, is said to exist, if whenever $t_1[X] \approx_\theta t_2[X]$, it is also the case that $t_1[Y] \approx_\theta t_2[Y]$ where θ represents the membership threshold and the operator $\approx_\theta$ is used to indicate that $t_1[x]$ is $\theta - nearer$ to $t_2[x]$ and $t_1[Y]$ is $\theta - nearer$ to $t_2[Y]$.

Although various forms of FFDs have been proposed for fuzzy databases, they stressed upon theoretical perspective and only a few mining algorithms are given. The FFD discovery method (DDFFD) proposed by Wang et al (2010) validates and incrementally searches for FFDs from similarity-based fuzzy relational databases. For a given pair of attributes, the validation of FFDs is based on fuzzy projection and fuzzy selection operations. In the proposed Information theory based FFD discovery method(ITFFD), the presence of fuzzy FDs is discovered by computing entropy for the fuzzy columns, which does not require equivalent class refinements used in the dynamic FFD discovery method(Wang et al 2010).

II. RELATED WORKS

Traditional FDs capture data dependencies between attributes that take values from crisp domains. Fuzzy functional dependencies are used to capture the semantics of similarity relationships between fuzzy attributes. The authors Duki & Avdagic (2005) have discussed on computing fuzzy data constraints by using fuzzy calculus. They proposed a set of sound and complete inference rules for fuzzy functional dependencies and examine the lossless join problem of fuzzy relations. A significant body of research in the area of fuzzy database modeling has been developed over the past thirty years and tremendous gain is hereby accomplished in this area.

Various fuzzy database models (e.g., relational and object-oriented databases) have been proposed, and some major issues related to these models have been investigated by Ma & Yan (2008). They provided a brief literature review on fuzzy database models and Mitra et al (2002) gave a detailed survey of soft computing techniques used for data mining. Another interesting issue was the search for functional dependencies in fuzzy relational databases(Sozat &Yazici 2001, Raju & Majumdar 1988). Several forms of fuzzy functional dependencies in relational databases were defined by Bosc et al (1994), Hartileb (2006), Buckles & Petry (1982), Cordero et al (2010), Dutta et al (2009), Duki & Avdagic (2005), Raju & Majumdar (1988), Al-Hamouz & Biswas, (2006). Fuzzy approximate dependencies are discussed by Berzal et al (2005). A complete axiomization of fuzzy functional and multi-valued dependencies is discussed by Ma & Yan (2008). Fuzzy rule generation and reasoning are discussed by Chen & Lee(2003), Makrehchi (1995), Chen & Huang (2003), Wang & Mendel (1992), Zadeh (1997). Application of fuzzy functional dependencies in approximate query answering has examined by Intan &Mukaidono (2000).

In the work proposed by Buckles & Petry (1982), fuzzy similarity relations are attached to attribute domains to model inter-changeability between values. Since the beginning of the eighties, several groups of researchers have been working on the application, to database management, of methods based on fuzzy sets along with the possibility theory for the treatment of imprecision and uncertainty and the handling of properties

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 12, December-2020

whose satisfaction is a matter of degree. Discovering functional dependencies in similarity based fuzzy relational databases is discussed by Wang et al (2008). Based on the concept of tuple partitions, Wang et al (2008) proposed an incremental data mining algorithm to discover fuzzy functional dependencies from similarity-based fuzzy relational databases. Wang et al (2010) has discussed a dynamic discovery method, which extracts Fuzzy FDs from dynamically growing datasets. The proposed information theory based approach uses entropy to capture the probability distribution of attribute values in a single value, which does not involve computation of minimal cover or set closure to discover functional dependencies.

III. INFORMATION THEORY BASED FFD DISCOVERY METHOD

Fuzzy functional dependencies are used to capture the semantics of similarity relationships between fuzzy attributes. Certain inter-attribute dependencies may be fuzzy in nature and can not be expressed using crisp attributes. Consider a relation U(A,B,C) that includes a fuzzy attribute A and crips attributes B, C. Let it be assumed that the attribute A with crisp domain is fuzzified into n different fuzzy sets and as a result the fuzzy columns fA_1, fA_2... fA_n. are added to the relation. The entropy of the fuzzy attribute is computed by finding the entropy of each fuzzy column and summing them as shown in Equation 1.1.

$$H(fA) = \sum_{i=1}^{n} H(fA_i) \qquad (1.1)$$

The joint probability between a fuzzy attribute A and a crisp attribute B is computed by finding partial joint entropy between the crisp attribute B and each of the fuzzy columns separately. The summation of partial entropies gives the joint entropy between A and B as shown in Equation 1.2.

$$H(AB) = \sum_{i=1}^{n} H(fA_i B) \qquad (1.2)$$

After computing attribute entropy and joint entropy between attributes, a level-wise search through the attribute semi-lattice is carried out to discover FFDs by checking the presence of FFD using Theorem 1 stated in Chapter 1. Mining of fuzzy functional dependencies is carried out step by step as described below.

ITFFD Algorithm

Input: A relational instance r(U) with n attributes and m tuples,
Membership functions $f_1, f_2, f_3 \ldots f_n$,
Membership threshold θ
Output: Set of FFDs holding in r(U)
Procedure
Begin
1. for each fuzzy attribute fa_i i=1,2..k
   fac_i[] ← Fuzzyfication(fa_i, f_i);
2. for each crisp attribute i= 1,2..m-k
   Compute_entropy();
3. for each fuzzy column added in step one,
   Compute_fuzzyEntropy(θ);
4. for each fuzzy attribute A and a crisp attribute B
   H(AB←Compute_jointEntropy(AB);
   if (H(AB) = = H (B))
   then FFD= FFD U (B $\rightarrow_\theta$ A)

End

Fuzzyfication(fa_i, f_i) is the process of converting fuzzy attribute fa to a set of fuzzy columns represented as fac_i based on the membership function f_i. The procedure Compute_entropy( ) computes the entropy of the crisp attributes and the procedure Compute_fuzzyEntropy(θ) computes the entropy of fuzzy columns. The values in the fuzzy columns with membership degree greater than or equal to the membership threshold θ are treated as equal. The Compute_jointEntropy(AB) function computes the joint entropy of AB using Equation 1.2.

In step 4, the presence of FFD is checked using the equality H(AB) = H(B). When entropy is used to check the presence of functional dependency, set comparisons of equivalence classes are not required. For any two attributes, only their joint entropy and attribute entropy are to be compared to check the presence of functional dependencies. This reduces the computation time.

IV. ILLUSTRATION

For example, Table 1.1 shows the height of successful players of different sports. The height attribute of the Table may be fuzzified by associating linguistic variables like short, medium and tall which are quantified using the trapezoidal membership functions shown in Figure 1.

Table 1 Sample Relation Showing Height of Successful Players

Tuple	Player_Name	Sports_Name	Height
T1	Ivo Karlovic	Tennis	208cm
T2	Mike Mentzer,	Weightlifting	176 cm
T3	Michael Phelps	Swimming	194 cm
T4	John Isner	Tennis	206cm
T5	Mario Lemieux	Ice Hockey	180 cm
T6	Juan Martín del Potro	Tennis	198 cm
T7	Brett Kimmorley	Rugby league	173 cm
T8	Franco Columbu,	Weightlifting	165 cm
T9	Mario Ancic,	Tennis	196cm
T10	Michael Grob	Swimming	201 cm
T11	Shawn Ray,	Weightlifting	170 cm
T12	Chris Pronger	Ice Hockey	180 cm
T13	Andrew Johns	Rugby league	174 cm
T14	Alexey Lesukov,	Weightlifting	168 cm
T15	Marin Cilic	Tennis	198 cm
T16	Brett Hodgson	Rugby league	175 cm
T17	Wayne Gretzky	Ice Hockey	183 cm
T18	Billy Slater	Rugby league	176 cm

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 12, December-2020

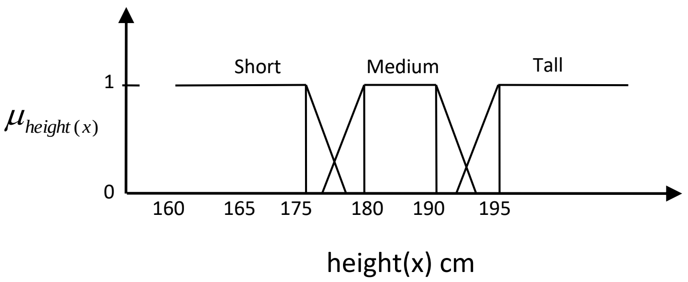

Figure 2. Trapezoidal Membership Function for Height

The membership function used to assign membership degree to various values of height along the fuzzy dimensions Tall, Short, Medium is shown in Table 2.

Table 2 Trapezoidal Function Furnished in Figure 2

Membership Function	Membership degree	Value range
f(x, a, b, c, d)	0	x < a and x > d
	(x - a) / (b - a)	a ≤ x ≤ b
	1	b < x < c
	(d - x) / (d - c)	c ≤ x ≤d
Tall(x)	0	height(x) ≤ 191
	(height (x)-191)/5	191 < height(x) < 195
	1	> 195
Medium(x)	0	height(x) < 177 and height(x) >194
	(height (x)-177)/3	177 ≤ height(x) < 180
	1	180 ≤ height (x) ≤190
	194-height(x)/3	191 ≤ height(x) ≤ 194
Short(x)	0	height(x) < 160 and height(x) >178
	(height (x)-176)/3	175 < height(x) ≤ 178
	1	height (x) ≤ 175

The projection of the actual relational table over the attribute height, fuzzified using the membership function is shown in Table 3. The membership threshold θ to qualify data values as identical can be fixed as any value greater than 0.5.

Table 3. Projection of Relational Table Shown in Table 1 on Linguistic Variables Short, Medium and Tall

Tuple	Sports_Name	Height	μ_{tall} (Height)	μ_{medium} (Height)	μ_{short} (Height)
T1	Tennis	208cm	1	0	0
T2	Weight lifting	178 cm	0	0.33	0.66
T3	Swimming	194 cm	0.75	0	0
T4	Tennis	206cm	1	0	0
T5	Ice Hockey	180 cm	0	1	0
T6	Tennis	198 cm	1	0	0
T7	Rugby league	173 cm	0	0	1
T8	Weight lifting	165 cm	0	0	1
T9	Tennis	196cm	1	0	0
T10	Swimming	201 cm	1	0	0
T11	Weight lifting	170 cm	0	0	1
T12	Ice Hockey	180 cm	0	1	0
T13	Rugby league	174 cm	0	0	1
T14	Weight lifting	168 cm	0	0	1
T15	Tennis	198 cm	1	0	0
T16	Rugby league	175 cm	0	0	1
T17	Ice Hockey	183 cm	0	1	0
T18	Rugby league	178 cm	0	0.33	0.66

The Table is partitioned into equivalence classes that include tuple IDs of those tuples that qualify as equal along different linguistic variables associated with the attribute. The relational table is also partitioned based on crisp data values over the crisp attribute Sports_Name and with each linguistic dimension separately Partitioning of tuples are done by sequentially checking the tuples, but by using hash table. Usage of hash table helps in getting the frequency count of every distinct value in a particular column faster using which entropy of the column is computed.

$\prod_{(Sports\_Name)}(PLAYERS)$ = {{ T1, T4, T6,T9, T15} , { T2, T8, T11, T14},{T3,T10}, { T5,T12,T17}, { T7,T13,T16,T18} }

$\prod_{(Sports\_Name,short(Height(x))}(PLAYERS)$ ={{T2, T8, T11, T14}, { T7,T13,T16,T18}}

$\prod_{(Sports\_Name,Medium(Height(x))}(PLAYERS)$ = { { T5,T12,T17} }

$\prod_{(Sports\_Name,Tall(Height(x))}(PLAYERS)$ = {{ T1, T4, T6,T9, T15}, {T3,T10} }

Consider the relational Table PLAYERS and the projected relational table given in Table 2 and Table 3 respectively. Let us take θ = 0.6. The fuzzy attribute entropy and joint entropy of the attributes are computed using Equation 1.1 and 1.2 respectively.

H(Sports_Name) = 2.257

H(Sports_Name,Tall(Height(x)) = 0.863

H(Sports_Name,Medium(Height(x))= 0.430

H(Sports_Name , Short(Height(x)) = 0.964

H (Sports_Name , Height (x)) = 0.863 + 0.430 + 0.964 = 2.257.

It is seen that H(Sports_Name) is equal to H(Sports_Name, Height (x)) and this equality in entropy values indicate that Sports_Name $\rightarrow_{\theta}$ Height with a

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 12, December-2020

degree of 0.6. The following fuzzy rules could be derived from this fuzzified functional dependency.

Fuzzy Rule 1: All successful tennis and swimming players are tall.

Fuzzy Rule 2 : All successful Weightlifting and Rugby league players are short.

Fuzzy Rule 3: All successful Ice Hockey players are medium in height.

V. EXPERIMENTAL RESULTS

The algorithms ITFFD and DDFFD were implemented using java and tested on the sports Table, which includes values collected from Wikipedia. The sports Table shown in Table 1 is extended with the attributes like Age, Weight, and Country etc. to create a dataset of large arity. There are about 1K records in the Table. The data records are duplicated to create data sets of large size. The experiments were run on Intel® core™ i5 Duo CPU at 1.6 GHz speed with 8 GB RAM.

Figure 3 shows the precision of the results produced by the ITFFD algorithm. Precision increases as the number of records increases, because more number of records contributes to the qualifying FFDs. The experiment is repeated by varying the membership threshold between 0.6 and 0.9.

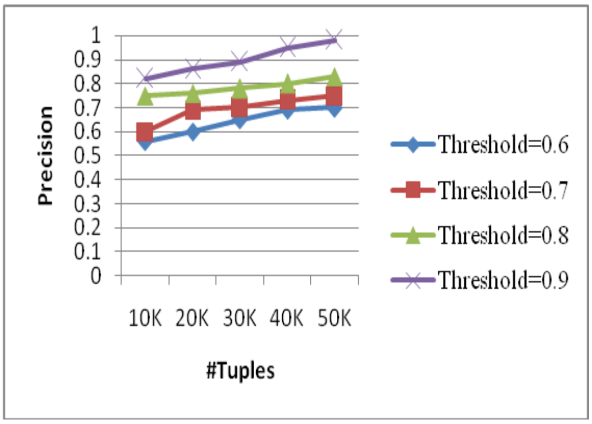

Figure 3 Precision Vs Number of Tuples

When the membership threshold increases, the accuracy of the detected rules also increases. When the membership threshold is kept at 0.9, the fuzzified attribute is almost equivalent to crisp attribute and hence contributes to higher accuracy of the results. Figure 4 shows the numbers of FFDs extracted from datasets of different sizes. As the size of the dataset increases, the number of FFDs decreases. This is due to the fact that, the new tuples inserted may invalidate the FFDs discovered. When the membership threshold increases, the number of FFDs discovered decreases.

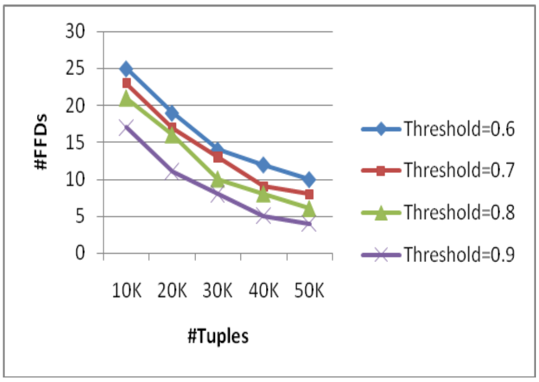

Figure 4 Number of FFDs Vs Number of Tuples

Figure 5 shows the execution time in milliseconds taken by the algorithm for both fuzzification and for extracting fuzzy functional dependencies.

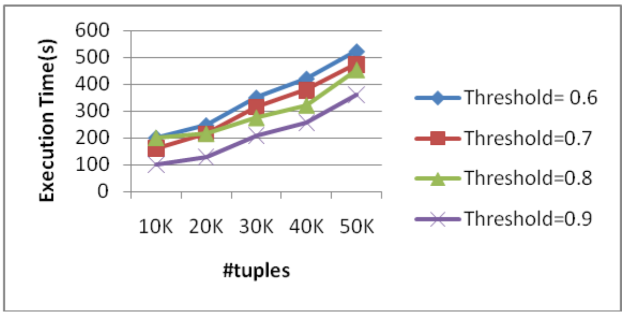

Figure 5 Execution Time Vs Number of Tuples

The execution time of the algorithm decreases, when the threshold increases. The number of FFDs to be verified decreases as the membership threshold increases and hence the algorithm takes lesser time than that with lower threshold values. The execution time taken by the proposed ITFFD approach and the recently proposed DDFFD method of Wang et al (2010) when the number of tuples is varied from 10K to 50 K records is shown in Figure6.

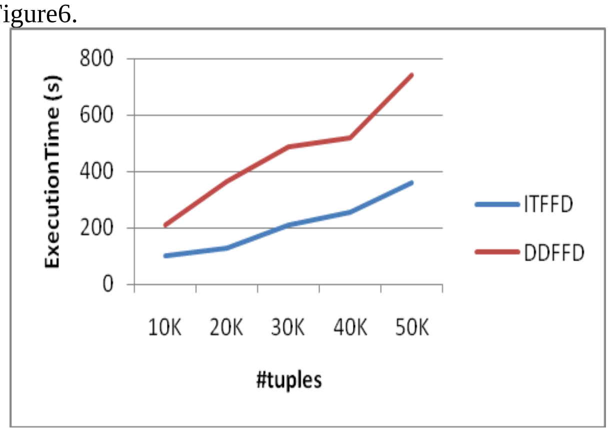

Figure 6 Execution Time- ITFFD Vs DDFFD

It is seen from Figure 6 that the time taken by the proposed approach is 40% less on average compared to the time taken by DDFFD method. Computing equivalence classes and their refinement is not required by the proposed approach and hence takes much lesser time to discover FFDs.

CONCLUSION

The extensions of traditional FDs with fuzzy logic helps to capture more semantics from data in the form of rules. An algorithmic approach is discussed, that dictates a step by step procedure to extract fuzzy functional dependencies from data sets. The experimental results show that the proposed approach discovers FFDs faster, because of using effective pruning rules to reduce the dependency mining search space.

REFERENCES

[1] Dukić, N & Avdagić, Z 2005, 'Fuzzy functional dependency and the Resolution Principle', Informatica, vol.16, no.1, pp. 45-60.
[2] Dutta, Ashit Kumar, Al-adhaileh, Mosleh Hmoudab, Ranjit Biswas, 2009, 'A Method of Intuitionist Fuzzy Functional Dependencies in Relational Databases', European Journal of Scientific Research, vol.29, no.3, pp.415-425.
[3] Cordero, P, Enciso, M, Mora, A & de Guzmán, IP 2010, 'A Complete Logic for Fuzzy Functional Dependencies over t-norms', In XV Congreso Espanol Sobre Technologies Y Logica Fuzzy (ESTYLF).
[4] Chaudhuri, S, Ganti, V & Motwani, R 2005, 'Robust Identification of Fuzzy Duplicates', in Proc. 21st IEEE Conf. on Data Engineering, pp.865-876.
[5] Frank, A & Asuncion, A 2010, UCI machine learning repository.
[6] Giannella, C & Robertson, E 2004, 'On Approximation Measures for Functional Dependencies' Information Systems, vol. 29, no.6, pp.483-507.
[7] Hartlieb, B 2006, 'Functional Dependencies in Fuzzy Databases', 21st Computer Science Seminar, Vol. 2, pp. 363-372.
[8] Intan, R & Mukaidono, M 2000, 'Fuzzy Functional Dependency and its Application to Approximate Data Querying', International Database Engineering and Applications Symposium, pp. 47-54.
[9] Liu, J, Li, J, Liu, C & Chen, Y 2012, 'Discover Dependencies from Data—A Review', IEEE Trans. on Knowledge And Data Engineering, vol.24, no.2, pp.251-264.
[10] Lopes, S, Petit, JM & Lakhal, L 2002, ' Functional and Approximate dependency Mining: Database and FCA points of view', Journal of Experimental and Theoretical Artificial Intelligence, vol.142, no.3, pp.93-114.
[11] Ma, ZM & Yan, L 2008, 'A Literature Overview of Fuzzy Database Models', Journal of Information Science and Engineering, vol.24, no.1, pp.189-202.
[12] Wang, LX & Mendel, JM 1992, 'Generating Fuzzy Rules by Learning from Examples', IEEE Trans. on Systems, Man and Cybernetics, vol.22, no.6, pp.1414-1427.
[13] Wang, S. L., Shen, J. W., & Hong, T. P. 2010, 'Dynamic Discovery of Fuzzy Functional Dependencies Using Partitions', Intelligent Soft Computation and Evolving Data Mining: Integrating Advanced Technologies, vol.44,pp.44-60.
[14] Wang, SL, Shen, JW & Hong, TP 2008, 'Incremental Discovery of Fuzzy Functional Dependencies', Handbook of Research on Fuzzy Information Processing in Databases, pp.615-633.
[15] Yao, YY, Wong, SM & Butz, CJ 1999, 'On Information-Theoretic Measures of Attribute Importance', LNCS Springer, vol.1574, pp.133-137.
[16] L Jekov L. Jekov, P. Cordero,M. Enciso (June 2017) ,"Fuzzy functional dependencies", Fuzzy Sets and Systems.
[17] MiljanVučetića, MiroslavHudecbc, BoškoBožilovića, (February 2020), Fuzzy functional dependencies and linguistic interpretations employed in knowledge discovery tasks from relational databases, Engineering Applications of Artificial Intelligence , Volume 88.