Lambda Functions and Approximate Generalized Positive Boolean Dependencies

Huy Nguyen Xuan[1], Van Nguyen Thi[2]

[1] Institute of Information Technology, Vietnam Academy of Science and Technology, Ha Noi , Vietnam
[2] Hanoi Community College, Ha Noi, Vietnam

Correspondence: Van Nguyen Thi, van.cdcd@gmail.com

Communication: received 21 June, revised 9 August, accepted 30 August.
DOI: 10.32913/mic-ict-research.v2022.n2.1099

***Abstract*: The main purpose of the paper is to propose a lambda function and its apply to the concept of measure in comparing tuples of relations. Extend the generalized positive Boolean dependency to obtain a new type of dependency called approximate generalized positive Boolean dependency.**

The results can be applied in constructing more complicated databases, especially allowing extended search capabilities for the real-world data.

Keywords: *Generalized Boolean dependency, approximate generalized positive Boolean dependency, lambda function.*

I. Introduction

he theory of data dependencies play an important role in describing the real world and reflecting the data semantics in the databases. In 1970, Codd proposed the concept of relational database with the data constraints presenting as functional dependencies [5]. This concept is a useful tool for the database analysis and design.

Namely, attribute B is functionally dependent on attribute A in relation r, if each two tuples u and v in r are equal on attribute A then they must be equal on attribute B. For example, if H is the customer's height and S is his shirt size, then the functional dependency $H \longrightarrow S$ requires two people to have the same height to receive the same shirt size. Similarly, the relationship between weight, age, and drug dosage should be understood as dependencies that should be relaxed.

The absolute dependencies above reveal the inadequacy when we have to deal with relations with thousands of tuples, all the while, there are only a few tuples not seem to be functional dependency accepted.

This loses the inherent flexible dependency between attributes. Therefore, from the beginning of the 80s of the last Century, studies have extended the concept of functional dependencies to the approximate dependencies. These dependencies allow us to describe the real world more accurately by replacing equality comparisons with approximate matching on tuples [3, 6, 8].

Along with the development of science and technology, these types of dependencies have been appearing more and more in the fields such as genetics, physics, molecular biology, materials technology, etc. The demand for a mathematical tool to help describe and reflect the loosened data dependence patterns in general and approximations in particular are necessary and natural [7, 9].

In this paper, continue the above research tendency, we are going to expand the general positive Boolean dependencies to obtain a new kind of dependencies, which called the approximate generalized positive Boolean dependencies.

II. Concepts and Notations

Let $U = x_1, \dots, x_n$ be a finite set of Boolean variables that hold the values in the set of logical values $B = 0$ (FALSE), 1 (TRUE). Each Boolean formula can be constructed from U using the logical connectives $\wedge, \vee, \longrightarrow, \neg$, and logic constants 1 and 0. A vector 0/1, $v = (v_1, \dots, v_n)$ in B^n is called *an assignment*. Then each Boolean formula f, $f(v)$ is a value of f *on assignment* v. We denote e is the *unit assignment*, $e = (1, 1, \dots, 1)$. A formula f is called positive if $f(e) = 1$. The formula $X \longrightarrow Y, X, Y \subseteq U$ is called implication. It is clear that every implication formula is positive. Denote by $P(U)$ the set of all positive Boolean formulas over U, and by T_f the truth-table of $f, T_f = \{v \in B^n | f(v) = 1\}$. A set formulas F can be expressed as $F = \wedge\{f | f \in F\}$, so $T_F = \cap\{T_f | f \in F\}$ is the truth-table of F. It is known that $f \longrightarrow g(F \longrightarrow g)$ if and only if $T_f \subseteq T_g(T_F \subseteq T_g)$.

If U is a set of attributes, $A \in U$ and v is a tuple in a relation r on U, then $v.A$ is the value of variable A in tuple v. If $X \subseteq U$, then we define $v.X = \{v.A | A \in X\}$.

By convention of knowledge and database theory, some time, we use the following equivalent notations

A set of attributes (or logical variables) can be written as a sequence of symbols:

$$\{a, b, c\} \equiv abc$$
$$a \wedge b \equiv ab$$
$$a \vee b \equiv a + b$$
$$\neg a \equiv a'$$

Other traditional notations and conventions can be found in the database literature [1, 4, 5].

We assume that each domain V_x of attribute x in U contains at least two values. For each domain V_x, we consider a mapping α_x: $V_x^2 \rightarrow \{0,1\}$ which satisfies the following properties [1, 6]:

$\forall a, b \in V_x$

$A1)$ *Reflexivity* $\alpha_x(a,a) = 1$

$A2)$ *Symmetry* $\alpha_x(a,b) = \alpha_x(b,a)$

$A3)$ *Partiality* $\exists c \in V_x : \alpha_x(a,c) = 0$.

α_x is a partial relation on domain V_x satisfying reflexivity and symmetry.

It is easy to see that equal mapping on V_x, denoted by $=_x$, satisfies the properties $(A1),(A2)$ and $(A3)$.

Each positive Boolean formula f in $P(U)$ with the given comparison α is called *generalized positive Boolean dependency* (GPBD). A pair $p = (U,F)$, where U is a finite set of attributes with the comparisons α on domains, F is a finite set of generalized positive Boolean dependencies is called *relational schema of generalized positive Boolean dependencies*.

Let $p = (U,F)$ be a schema, and r be a relation on U. For each pair of tuples $u = (u_1,u_2,...,u_n), v = (v_1,v_2,...,v_n)$ in r, we set a vector 0/1 $t = \alpha(u,v) = (t_1,t_2,...,t_n) \in B^n$ where the $t_i = \alpha_i(u_i,v_i), 1 \le i \le n$.

We call the set $T_r = \{\alpha(u,v)|u,v \in r\}$ value-table of relation r on the schema p [1, 2, 3, 6].

A relation r on U satisfies a GPBD f (set of GPBDs) and is writen as $r(f)$, ($r(F)$ if $T_r \subseteq T_f (T_r \subseteq T_F)$.

Example 2.1

Consider the relation r describing individuals with the following attributes: bloodline H, father's ADN F and mother's ADN M. We see that two individuals with the same bloodline have the same father's ADN or mother's ADN. We have $H \rightarrow F + M$ is a GPBD. With relation $r(H,F,M)$ containing information about bloodline, we construct the value-table of T_r as follows:

TABLE 2.1
RELATION r WITH ATRIBUTES: BLOODLINE H, FATHER'S ADN: F AND MORTHER'S ADN: M

Tuple	H	F	M
1	h1	f0	m2
2	h2	f1	m1
3	h1	f0	m3
4	h3	f2	m4
5	h2	f3	m1

TABLE 2.2
VALUE-TABLE OF RELATION r

pair of tuples	H	F	M
(1,1)	1	1	1
(1,2)	0	0	0
(1,3)	1	1	0
(2,5)	1	0	1

TABLE 2.3
TRUTH-TABLE OF FUNCITON $H \rightarrow F + M$

H	F	M
1	1	1
1	0	1
1	1	0
0	1	0
0	0	1
0	1	1
0	0	0

Let U be a finite set of attributes. Denote by $REL(U)$ the set of all relations on U, $REL2(U)$ the set of all relations with no more than two tuples on U. Let F be a set of GPBDs and f be a GPBD. We say that

- F implies f by logic and denote as $F \models f$ if $T_F \subseteq T_f$
- F implies f by relation, and denote as $F \vdash f$ if $\forall r \in REL(U) : r(F) \rightarrow r(f)$
- F implies f by no more than two tuple, denote $F \vdash_2 f$ if

$$\forall r \in REL2(U) : r(F) \Rightarrow r(f)$$

Equivalence theorem

Let F be a set of GPBDs and f be a GPBD. The following are equivalent [1, 6]:

$(i) F \models f (logicconsequence)$

$(ii) F \vdash f (relationconsequence)$

$(iii) F \vdash_2 f (two-tuplerelationconsequence)$

The equivalence theorem allows us to determine when a GPBD f is con-sequenced from a set of GPBDs F through formal logical transformations without relying on data.

III. LAMBDA FUNCTION

Let U be an attribute set. For each attribute A in U, we define a mapping λ_A from the value domain of the attribute to the real number space as follows:

Let U be an attribute set. For each attribute A in U, we define a mapping λ_A from the value domain of the attribute to the real number space as follows:

$\lambda_A : V_A \rightarrow \mathbb{R}$

$L1)$:$\forall a, b \in V_A : a = b \Rightarrow \lambda_A(a) = \lambda_A(b)$

$L2)$: $\exists a, b \in V_A$: $\lambda_A(a) \neq \lambda_A(b)$

λ_A is called a quantitative function for the attribute A.

Example 3.1

- Let attribute A be the type string, we define $\forall s \in V_A : \lambda_A(s) = len(s)$. We have,

$\lambda_A(\text{"}Internet\text{"}) = 8$; $\lambda_A(\text{"}e-mail\text{"}) = 6$;
$\lambda_A(\text{""}) = 0$ (the quantification of empty string is 0).

- Let attribute B be the salary coefficient we define $\forall \in V_B : \lambda_B(a) = a$. Then

$$\lambda_B(2.3) = 2.3; \lambda_B(8.0) = 8.0$$

- Let attribute C be the level of satisfaction rank (the customer for an airline company, for example) with the values $V_C = \{verysatisfied, satisfied, normal, notsatisfied\}$ we define

$\lambda_C(verysatisfied) = 4$; $\lambda_C(satisfied) = 3$; $\lambda_C(normal) = 2$; $\lambda_C(notsatisfied) = 1$.

Theorem 3.1

Let λ_A be given, then the function α_A is defined by the following formula (*)

$$\forall a, b \in V_A: \alpha_A(a,b) \overset{def}{\Longleftrightarrow} (\lambda_A(a) = \lambda_A(b)) \quad (*)$$

Satisfy properties $A1, A2$ and $A3$ in the definition of α.

Proof

Suppose $a \in V_A$. Then $\lambda_A(a) = \lambda_A(a)$ therefore $\alpha_A(a,a) = 1$. Reflective properties $A1$ is proven.

Suppose $a, b \in V_A$. We have

$$\alpha_A(b,a) = 1 \overset{def}{\Longleftrightarrow} \lambda_A(a) = \lambda_A(b) \overset{def}{\Longleftrightarrow} \alpha_A(b,a) = 1.$$

The A2 symmetry properties is proven.

According to the definition of the Lambda function, $\exists x, y \in V_A : \lambda_A(x) \neq \lambda_A(y)$, implies $\alpha_A(x,y) = 0$.

The A3 Partiality properties is proven.

The theorem is proven

Measure

The measure d on the set V is a mapping $d : V \times V \to \mathbb{R}^+$ where $\mathbb{R}^+$ is the set of non-negative real numbers, satisfying the following properties : $\forall a, b, c \in V$.

- $Non-negativity : d(a,b) \geq 0,\ d(a,a) = 0$
- $Symmetry : d(a,b) = d(b,a)$
- $Triangle : d(a,b) + d(b,c) \geq d(a,c)$.

Example 3.2

On line (one-dimensional space), the measure between two points x and y representing the distance from x to y is calculated by $d(x,y) = ||x - y||$. In n-dimensional space, the measure between two points $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ is defined over the *Euclid* distance as follows:

$$e(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

The A2 symmetry properties is proven.

Let $p = (U, F)$ be a Lambda functions defined for each attribute. For each tuple $u = (u_1, u_2, ..., u_n) \in r$, we define $\lambda(u)$ as a vector

$$\lambda(u) = (\lambda_1(u_1), \lambda_2(u_2), ..., \lambda_n(u_n))$$

Theorem 3.2

Given a relation r on the attribute set U and Lambda functions on each attribute A in U. Let d be an arbitrary measure in an n-dimensional vector space. Then the following d function is a measure:*

$$\forall u, v \in r : d^*(u,v) = d(\lambda(u), \lambda(v))$$

We call d^* the *inductive measure* from the d measure.

Proof

We proof that the function d^* satisfies the three properties of the measure definition. Indeed, because d is non-negative, then $d^*(u,v) = d(\lambda(u), \lambda(v)) \geq 0$. If $u = v$ then $\lambda(u) = \lambda(v)$, therefore $d^*(u,v) = d(\lambda(u), \lambda(v)) = 0$.

The symmetry property of d^* is directly implied from the symmetry property of the measure d.

Assume u, v and w are three tuples in r. Since d satisfies is a measure, d satisfies the triangular property, we have,

$$d^*(u,v) + d^*(v,w) = d(\lambda(u), \lambda(v)) + d(\lambda(v), \lambda(w)) \geq d(\lambda(u), \lambda(w)) = d^*(u,w).$$

The triangle property of d^* is proven

Theorem is proven.

IV. Some special case of generalized positive Boolean varieties

Let U be an attribute set, $X, Y \subseteq U$, r be a relation on U.

1. Functional dependencies

A $functional dependency$ (FD) on U is a formula $f : X \to Y$. We say the relation $r satisfies$ FD f and write $r(X \to Y)$ if for every pair of tuples u and v in r that are equal on X then they must be equal on Y :

$$r(X \to Y) \overset{def}{\Longleftrightarrow} \forall u, v \in r: u.X = v.X \Rightarrow u.Y = v.Y$$

Thus, FD is a special case of GPDB when the positive Boolean formula is the consequence formula $X \to Y$ and the comparison is equality.

2. Weak dependencies

A formula of the form $f : \wedge X \to \vee Y$ is called weakly derived formula. A weak dependency (WD) on U is a weakly derived formula $f : \wedge X \to \vee Y$ is called weakly derived formula.

We say the relation r satisfies WD f and write $r(\wedge X \to \vee Y)$ if for every pair of tuples u and v in r that are equal on X then they must be equal on some attribute B of Y.

$$r(\wedge X \to \vee Y) \Leftrightarrow (\forall u, v \in r): (u.X = v.X)$$
$$\Rightarrow \exists B \in Y: \ (u.B = v.B)$$

Thus, an WD is a special case of the positive Boolean dependency. We also have:

The *relation r satisfies the weak dependency f* (the set of WDs F) if and only if $T_r \subseteq T_f (Tr \subseteq T_F)$.

In [2], the following results is proven:

Theorem 4.1

textitEach positive Boolean formula is equivalent to a set of weakly derived formulas (in the sense that they all have the same truth-table).

Example 4.1

The positive Boolean formula $f = (a + b) \rightarrow (c \rightarrow d)$ is equivalent to the set of weakly derived formulas $S = \{bc \rightarrow a + d, ac \rightarrow b + d, abc \rightarrow d\}$

Example 4.2

In the bloodline relation $r(H, F, M), B + M$ is weakly dependent on H.

3. Approximate Weak Dependency (AWD)

The concept of approximate functional dependencies was proposed by Jyrky Kivinen et al. in 1995 [7, 8, 9], where the term *approximation* is understood as if at least ε tuples are removed from the relation r then the rest will satisfy the given function dependency. In this paper, a general concept of approximate dependence is proposed on a set of numeric and non-numerical attributes, according to an arbitrary positive Boolean formulas with more general comparisons.

Given an attribute set U with a measure d and a weak derivation formula on $U, f : \wedge X \rightarrow \vee Y$ Let $\varepsilon_1, \varepsilon_2$ are two non-negative real numbers. The approximate weak dependency (AWD) of attribute set Y on attribute set X is an expression of the form:

$$X\ \varepsilon_1 \rightarrow \varepsilon_2\ Y$$

We say that the relation $r(U)$ satisfies the *approximate weak dependence* $X\ \varepsilon_1 \rightarrow \varepsilon_2\ Y$ if u and v in the relation r, the two tuples u and v have

$$\forall u,v \in r\colon d(u.X, v.X) \leq \varepsilon_1 \Rightarrow \exists B \in Y\colon d(u.B, v.B) \leq \varepsilon_2$$

Example 4.3

On the promotional day, a supermarket selling item H has a policy to give each customer a discount of 10 units of the assumed currency T from 1 PM onwards. Information at the cashier is given in the form of the relation r as follows:

TABLE 4.1
RELATION r

ID	Time	Quantity	Price	Total
1	10	5	10	50
2	11	8	10	80
3	12	4	10	40
4	13	5	10	40
5	14	8	10	70
6	15	4	10	30

Consider the following dependencies:

- Functional Dependency:
  $h : Quantity, price \rightarrow Total$
- Weak Dependency :
  $w : Quantity, price \rightarrow Total$
- Approximate Weak Dependency :
  $f : Quantity, price(0) \rightarrow (10)Total$

We can see that two transactions 1 and 4 buy the same quantity but at two different times, then the Total has two different values, so the relation r is not satisfied functional dependency h.

Similar, relation r is not satisfied approximate weak dependency w.

However, relation r satisfies the approximate weak dependency f: two customers buying the same quantity at different times will pay different sums: Total and Total - 10.

We call $p = (U, \varepsilon_1, \varepsilon_2, \lambda, d, F)$ a *relation schema with approximate weak dependencies* (RSAWD), where

- U is a set of n attributes, each of which is equipped with a quantitative Lambda function λ.
- d is an arbitrary measure.
- F is a set of weakly consequence formulas $\wedge X \rightarrow \vee Y, X, Y \subseteq U$
- $\varepsilon_1 \geq 0, \varepsilon_2 \geq 0$ are approximate thresholds.

Given a relation r $RSAWD$, $p = (U, \varepsilon_1, \varepsilon_2, \lambda, d, F)$. For each pair of tuples $u = (u_1, u_2, ..., u_n), v = (v_1, v_2, ..., v_n)$ in r, we define:

$$t(u, v) = (t_1, t_2, ..., t_n)$$
$$t_i = (d(\lambda(u_i), \lambda(v_i)) \leq \varepsilon)?1 : 0; 1 \leq i \leq n$$
$$\varepsilon = min(\varepsilon_1, \varepsilon_2)$$
$$T_r = \{t(u, v) | u, v \in r\}$$

and call T_r the value table of the relation r according to the scheme p.

Theorem 4.2

Let $p = (U, \varepsilon_1, \varepsilon_2, \lambda, d, F)$ and a $RSAWD$ be an AWD. Let r be a relation on U. Then,

(i) The relation r satisfies AWD f if and only if $T_r \subseteq T_f$

(ii) Relation r satisfies the set of $AWDs$ F if and only if $T_r \subseteq T_F$.

Proof

(i) Suppose the relation $r(U)$ satisfies $AWD\ f$. We need to prove $T_r \subseteq T_f$. If r is empty, then since $T_r = \emptyset \subseteq T_f$ so $T_r \subseteq T_f$. Let $\varepsilon = min(\varepsilon_1, \varepsilon_2)$ and $u \in r$. Since $t(u, u) = e \in T_r$. Since f is a positive Boolean formula, then $e \in T_f$. Suppose $t \in T_r$, by the definition of $T_r, \exists u, v \in r : t(u, v) = t$. Since relation r satisfies f, we have $f(t) = 1$, so $t \in T_f$.

In the other hand, suppose $T_r \subseteq T_f$. We need to show that relation r satisfies $AWD\ f$. For every pair of tuples u, v in r, we have $t(u, v) \in T_r$. Since $T_r \subseteq T_f$, then $t(u, v) \in T_f$. This implies $f(t(u, v)) = 1$, that r satisfies AWD f.

(ii) By the definition of T_F we have

$$T_F = \cap\{T_f | f \in F\}$$

According to the proof (i), relation r satisfies the set F if and only if for every f in F we have $T_r \subseteq T_f$. Since T_F is the intersection of $Tf, T_r \subseteq T_F$. Otherwise, if $T_r \subseteq T_F$, then $T_r \subseteq T_f$ for every f in F. By proof (i), r satisfies all AWD f, so relation r satisfies the set of $AWDs$ F. The theorem is proved.

4. Approximate generalized positive Boolean dependency

Definition 4.1

Let U be an attribute set and f be a positive Boolean formula f over U. Let $\varepsilon_1, \varepsilon_2$ be two non-negative thresholds. Approximate generalized positive Boolean dependency (AGPBD) is a set of approximate weak dependencies S equivalent to f with the threshold $\varepsilon_1, \varepsilon_2$.

We say that the relation $r(U)$ satisfies AGPBD f if r satisfies every AWD in S.

Since f is equivalent to S, then $T_f = T_S = \cap\{T_g | g \in S\}$. We have the following result as a corollary of theorem 4.1:

Corollary 4.1

Let $p = (U, \varepsilon_1, \varepsilon_2, \lambda, d, F)$ *be a RSAWD and* f. *Let* r *be a relation on* U. *Then,*

(i) *The relation* r *satisfies AGPBD* f *if and only if* $T_r \subseteq T_f$.

(ii) *Relation* r *satisfies the set AGPBD* F *if and only if* $T_r \subseteq T_F$.

V. Conclusion

The classification and proposal of a common model for types of data dependencies is one of the issues of great interest to big data researchers. With an arbitrary measure, through Lambda functions on the value domain of the attributes, we can evaluate the approximation of tuples in the relation including numeric and non-numeric attributes of a database.

References

[1] Nguyen Xuan Huy (2006), *Logical dependencies in the database*, Statistical Publishing House (Vietnamese).

[2] Nguyen Xuan Huy, Truong Thi Thu Ha (2014), "Equivalence between positive and weak Boolean dependencies in relational databases", *Proceedings of the 17th National Conference: Some selected issues of Information and Communication Technology, Dak Lak, 30-31/10/2014, Science and Technology Publishing House*, ISBN: 978-604-67-0426-3, Hanoi, page.361-365. (Vietnamese)

[3] Nguyen Xuan Huy, Truong Thi Thu Ha, Nguyen Thi Van (2016), "Correlation between approximate functional dependency and positive Boolean dependency in general in relational databases, *Proceedings of the 19th National Conference: Some selected issues of Information and Communication Technology, Science and Technology Publishing House, ISBN: 978-604-67-0781-3, Hanoi, page.361-365. (Vietnamese)*

[4] Berman J., Blok W. J. (1988), "Positive Boolean dependencies", Inf. *Processing Letters, 27*, p.147 – 150.

[5] Codd E. F. (1970), "A Relational Model of Data for Large Shared Data Banks", *CACM 13:6*, 377-387.

[6] Huy Nguyen Xuan, Thanh Le Thi (1992), "Generalized Positive Boolan Dependencies", J. *Inf. Process. Cybern. EIK*, vol. 28, p. 363 – 370.

[7] Jalal Atoum (2009), "Mining Approximate Functional Dependencies from Databases", *European Journal of Scientific Research*, ISSN 1450 – 216X vol. 33 no. 2, p.338 – 346.

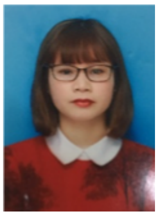
[8] Jyrky Kivinen et al (1995), "Approximate Inference of Funtional Dependencies from Relations", *Journal of Theoretical Computer Science*, Vol.149 Issue 1, p.129 – 149.

[9] Ronald S. King, James J. Legendre (2003), "Discovery of Functional and Approximate Functional Dependencies in Relational Databases", *J. of Appl. Math. and Decision Scienses*, 7(1), 49-59.

[10] Truong Thi Thu Ha (2017), "Correlation between logically dependent classes in the database", *PhD thesis, V-LA2/4042 – Military Technical Institute*. (Vietnamese)

Nguyen Xuan Huy, Institute of Information Technology, Vietnam Academy of Science and Technology Nguyen Xuan Huy received Math BS from the Herzen University, Saint Petersburg (1973), PhD in Computer Science (1980), and Doctor of Sciences in Computer Science (1990), Academy of Sciences, Russia. He is an associate professor and a senior researcher. He is the author of papers and books in computer science and database systems. He held positions in the departments of computer science of the State Thai Nguyen University, and Da Lat University. He is a visiting lecturer of the Ha Noi State University, Ha Noi Posts and Telecommunications Institute of Technology, State University in Ho Chi Minh city, Da Nang State University, Duy Tan University, and Qui Nhon University. Huy was the chairman of the Advisory Board of the Microsoft Project Partners In Learning (PIL) in Viet Nam (2004-2012). He is a member of Vietnam Association for Information Processing (VAIP, 1982), Vietnam Mathematical Society (VMS, 1982)

Email: nxhuy564@gmail.com.



Nguyen Thi Van received the B.S of IT at Hanoi University of Business and Technology, 2011 and M.S. degree in Computer Science at University of Information and Communication Technology, 2014. Currently working at, Hanoi Community College. Research area: Logical dependencies in databases, data models and database

Email: van.cdcd@gmail.com