Extended Functional Dependencies as a Basis for Linguistic Summaries

Patrick Bosc[1], Ludovic Liétard[2], and Olivier Pivert[1]

[1] IRISA/ENSSAT Technopole ANTICIPA BP 447 22305 Lannion Cedex France
{bosc|pivert}@enssat.fr
[2] IRISA/IUT Rue Edouard Branly BP 150 22302 Lannion Cedex France
ludovic.lietard@iut-lannion.fr

Abstract. This paper is concerned with knowledge discovery in databases and linguistic summaries of data. The summaries proposed here allow for a qualitative description of data (instead of the quantitative description given by a probabilistic approach) and they involve linguistic terms to obtain a wider coverage than Boolean summaries. They are based on extended functional dependencies and are situated in the framework of the relational model of data. Such summaries express a meta-knowledge about the database content according to the pattern "for any tuple t in relation R: the more A, the more B" (for instance: the *taller* the player, the *higher* his score in the NBA championship) where A and B are two linguistic terms. In addition, an algorithm to implement the discovery process (which takes advantage of properties of extended functional dependencies) is given. This algorithm is iterative and each tuple is successively considered in order to refine the set of valid summaries.

1 Introduction

Knowledge discovery in databases aims at extracting knowledge from information contained in a database. Many items may be the subject of discovery and among them are linguistic summaries of data [1, 2]. Such summaries are statements of the natural language (such as "*young* people are *well-paid*") and represent properties about the database current content. The objective of this paper is to propose linguistic summaries of data which offer a qualitative description of data and no longer a quantitative one as that given by a probabilistic approach. We also present an algorithm implementing the knowledge discovery process. In this algorithm, tuples are successively accessed in order to refine the set of valid summaries.

The linguistic summaries discussed later are based on extended functional dependencies [3] and are situated in the context of the relational model of data. Intuitively, the extended functional dependency (age, [20, 25]) $\rightarrow$ (salary, [5000, 7000]) means that an age between 20 and 25 imposes a salary between 5000 and 7000. One may point out that this property is too demanding, since a single tuple where age = 24 and salary = 7500 is enough to make it false. In addition, this extension is not linguistic and sharp boundaries (here [20, 25] and [5000, 7000]) may be challenged. Obviously, these aspects are due to the Boolean requirements regarding age and salary (i.e., all or nothing membership functions). Consequently, we consider fuzzy subsets [4] of attribute domains which are more appropriate to express some flexibility. More precisely, we will consider linguistic terms in the summaries which will have a fuzzy set-based interpretation [5]. As an example, when defining the fuzzy sets *young* and *well paid* on the attributes age and salary, the previous extended functional dependency is turned into (age, *young*) $\rightarrow$ (salary, *well paid*) which means

that "*young* people are *well paid*". This type of pattern (extended functional dependencies with graduality) allows for defining linguistic summaries of data. It is worth mentioning that a clear semantics of such a statement calls on establishing a connection between the degrees attached to young and well-paid. This is achieved through extended (fuzzy) implications which generalize the regular one in use in the expression of functional dependencies (FDs).

The paper is organized as follows: section 2 is devoted to fuzzy sets and extended implications; section 3 introduces previous propositions to define linguistic summaries. Section 4 introduces linguistic summaries defined as extended FDs and their properties, and proposes the discovery algorithm.

2 Fuzzy Sets and Extended Implications

Fuzzy sets were proposed by Zadeh [4] in order to define sets having non sharp boundaries. A fuzzy set F on the universe X (a fuzzy subset of X) is defined by a function μ_F which associates a membership degree in [0,1] to each element x of X. When $\mu_F(x) = 0$, x does not belong at all to F and the closer to 1 $\mu_F(x)$, the more x belongs to F.

Set operations have been extended to fuzzy sets according to the following definitions where A and B stand for two fuzzy sets defined over the universe X: i) $\forall x \in X$, $\mu_{A\cap B}(x) = op_1(\mu_A(x), \mu_B(x))$, where op_1 is a triangular norm (associative, commutative, monotonic operator such that $op_1(a, 1) = a$), ii) $\forall x \in X$, $\mu_{A\cup B}(x) = op_2(\mu_A(x), \mu_B(x))$, where op_2 is a triangular co-norm (associative, commutative, monotonic operator such that $op_2(a, 0) = a$). Among the pairs norm/co-norm of operators op_1/op_2, let us mention: $op_1(x,y) = \min(x,y)$; $op_2(x,y) = \max(x,y)$ which will be assumed later.

The implication operator has also been extended and two main families can be distinguished [6]. The extension based on the following definition of the Boolean implication: $(A \Rightarrow B) \Leftrightarrow$ (B is at least as true as A) leads to Rescher-Gaines implication ($a \Rightarrow_{R\text{-}G} b = 1$ if $a \leq b$ and 0 otherwise) which will be used later on.

3 Linguistic Summaries: Previous Approaches

A linguistic summary of data describes the content of the database. Dubois and Prade [1] propose a summary given by a fuzzy set describing the frequency of events knowing a similarity relation over attribute values. Yager and Rasmussen's proposal [2] is more qualitative and the authors propose to summarize a relation R by a sentence of the type "*Q* are *S*" where *Q* is a linguistic quantifier [7] and *S* a property whose satisfaction is gradual. Both *Q* and *S* are defined as fuzzy sets. A summary is associated with a degree T (in [0, 1]) which indicates its validity. When *Q* defines a proportion such as *about half* (resp. a number such as *at most 3*) it is defined by a fuzzy subset of [0,1] (resp. of the naturals) and T is given by:

$$T = \mu_Q\left(\sum_{t\in R}\mu_S(t)/n\right) \qquad \left(\text{resp. } T = \mu_Q\left(\sum_{t\in R}\mu_S(t)\right)\right)$$

where n is the number of tuples in relation R.

Example 1. The fuzzy quantifier *about half* and S the fuzzy set *young* are given by Fig. 1.

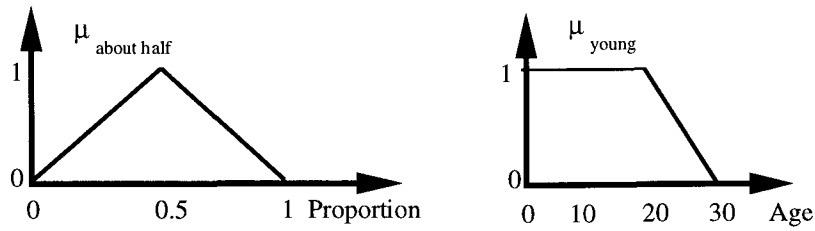

Fig. 1. A definition for *about half* and *young*.

The following relation EMP describes employees:

EMP

Name	Age	Salary	Firm	Country
Durand	25	6 000	Comp. Inc.	France
Dupond	25	7 000	Comp. Inc.	France
James	18	5 000	Softw. Inc.	UK
Peter	33	17 000	Softw. Inc.	UK
Walker	45	28 000	Softw. Inc.	UK
Müller	50	17 000	Netw. Inc.	Germany

The summary "*about half* are *young*" is valid at degree $\mu_Q((0.5+0.5+1)/6) = 2/3$ ♦

It is important to notice that two relations R and R' may lead to a same summary with the same value T even if they convey very different situations (it is just necessary that $\mu_Q(\sum_{t\in R}\mu_S(t)/n) = \mu_Q(\sum_{t\in R'}\mu_S(t)/n)$.

4 Summaries as Extended Functional Dependencies

4.1 Extended Functional Dependencies

A functional dependency (FD) over a relation R(U) is a property denoted $X \rightarrow Y$ ($X, Y \subseteq U$) defined as:

$\forall\ t, t' \in R, (t.X = t'.X) \Rightarrow (t.Y = t'.Y)$.

Example 2. The FD Firm → Country holds in relation EMP of Example 1 ♦

FDs share a number of properties among which augmentation ($X \rightarrow Y$ and $Z \subseteq U \Rightarrow XZ \rightarrow Y$), decomposition ($X \rightarrow Y \Rightarrow \forall\ Z \subseteq Y, X \rightarrow Z$) and union ($X \rightarrow Y$ and

$X \rightarrow Z \Rightarrow X \rightarrow YZ$). These three properties will play a key role in our algorithm aiming at the discovery of linguistic summaries of data (see section 5).

An extension of FDs has been proposed [3] where fuzzy sets (or linguistic labels) are used to characterize attribute values in a relation R. For the sake of simplicity, we consider that X and Y are single attributes. The extended FD $(X, L_i) \rightarrow (Y, L'_j)$, where L_i (resp. L'_j) is a label defined over the domain of X (resp.Y) means that:

- $\forall$ t, t' $\in$ R, if t.X and t'.X rewrite as L_i then t.Y and t'.Y rewrite as L'_j,
- each tuple t of R satisfies the property "the higher $\mu_{L_i}(t.X)$, the higher $\mu_{L'_j}(t.Y)$".

The second statement deals with graduality in order to ensure that the more L_i is t.X, the more L'_j is t.Y. The validity of $(X, L_i) \rightarrow (Y, L'_j)$ is based on:

$$(X, L_i) \rightarrow (Y, L'_j) \text{ holds} \Leftrightarrow \forall t \in R, \mu_{L_i}(t.X) \Rightarrow_{R\text{-}G} \mu_{L'_j}(t.Y),$$

where $\Rightarrow_{R\text{-}G}$ is Rescher-Gaines implication. In other words, $(X, L_i) \rightarrow (Y, L'_j) \Leftrightarrow (\min_{t \in R} [\mu_{L_i}(t.X) \Rightarrow_{R\text{-}G} \mu_{L'_j}(t.Y)] = 1) \Leftrightarrow \forall t \in R, \mu_{L_i}(t.X) \leq \mu_{L'_j}(t.Y)$.

Example 3. When considering the relation EMP of Example 1, the usual FD Age $\rightarrow$ Salary does not hold (due to the first two tuples). However, we may consider the linguistic labels *old*, *high* and *low* defined by the fuzzy sets of Fig. 2 and *young* defined in Fig. 1.

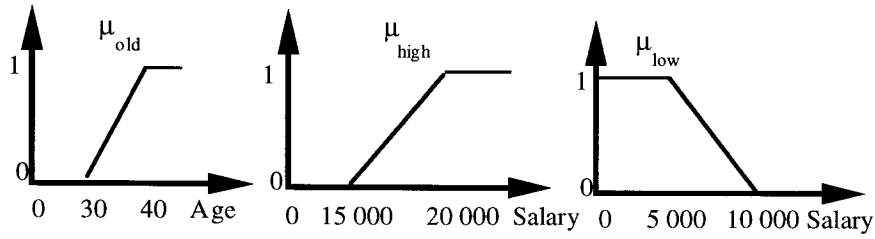

Fig. 2. The linguistic labels *old, high* and *low*

Then, relation EMP can be rewritten (from a conceptual point of view) as:

EMP

Name	Age	Salary	Firm	Country
Durand	*Young* (0.5)	*Low* (0.8)	Comp. Inc.	France
Dupond	*Young* (0.5)	*Low* (0.6)	Comp. Inc.	France
James	*Young* (1)	*Low* (1)	Softw. Inc.	UK
Peter	*Old* (0.3)	*High* (0.4)	Softw. Inc.	UK
Walker	*Old* (1)	*High* (1)	Softw. Inc.	UK
Müller	*Old* (1)	*High* (0.4)	Netw. Inc.	Germany

Each value of Age and Salary is replaced by its label along with the associated membership degree. In this case, the extended FD (age, *young*) $\rightarrow$ (salary, *low*) holds

in EMP, which means that the *younger* an employee, the *lower* his salary. On the other hand, the extended FD (age, *old*) → (salary, *high*) does not hold♦

4.2 Linguistic Summaries

Let us consider a relation R defined over n attributes A_1, A_2, ..., A_n. Each A_i has its domain partitioned into k fuzzy sets $L_{i,k}$ which intersect at degree 0.5 (cf. next example).

Example 4. The domain [0, 50] of Age (A_1) is partitioned into 6 fuzzy sets (k = 6): *around 0* ($L_{1,1}$), *around 10* ($L_{1,2}$), *around 20* ($L_{1,3}$), *around 30* ($L_{1,4}$), *around 40* ($L_{1,5}$) and *around 50* ($L_{1,6}$) as described in Fig. 3. Such a partitioning has the advantage of covering the entire domain. In this context, a linguistic summary which involves two different attributes A_i and A_j is expressed by an extended FD (A_i, $L_{i,u}$) → (A_j, $L_{j,v}$) where u (resp. v) indicates the label associated with A_i (resp. A_j.) (cf. Example 3).

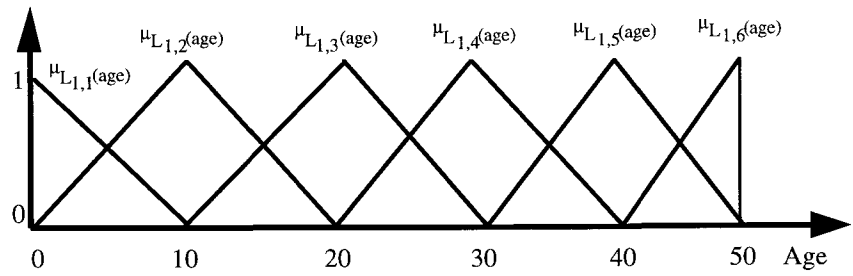

Fig. 3. Labels for the attribute Age♦

The general form of an extended FD is:

$$(A\sigma(1), L\sigma(1),f(1)), ..., (A\sigma(p), L\sigma(p),f(p)) \rightarrow (A_{\sigma(p+1)}, L_{\sigma(p+1),f(p+1)}), ..., (A_{\sigma(q)}, L_{\sigma(q),f(q)}) \quad (1)$$

where σ (resp. f) is a permutation of q values among {1, ..., n} (resp. an application from {1, ..., q} to {1, ..., k}) which characterize attributes (resp. the chosen labels). Without loss of generality we will consider that: 1) an attribute cannot appear several times in left or right part (otherwise the summary is not meaningful) and 2) an attribute cannot appear both in left and right parts. To simplify the notations, in the following, expression (1) will be denoted :

$$(A_1, L_1), ..., (A_p, L_p) \rightarrow (A_{p+1}, L_{p+1}), ..., (A_q, L_q).$$

The validity of this summary over R is given by:

$$\min_{t \in R} [\min(\mu_{L1}(t.A_1), ..., \mu_{Lp}(t.A_p)) \Rightarrow_{R\text{-}G} \min(\mu_{Lp+1}(t.A_{p+1}), ..., \mu_{Lq}(t.A_q))],$$

and the summary is valid iff:

$\forall$ t in R, $\min(\mu_{L1}(t.A_1), ..., \mu_{Lp}(t.A_p)) \leq \min(\mu_{Lp+1}(t.A_{p+1}), ..., \mu_{Lq}(t.A_q))$.

This means that each tuple t in R satisfies the property "t is at least as (L_{p+1} and ... and L_q) as it is (L_1 and ... and L_p)".

It is possible to show that these summaries satisfy the following properties:

- **Augmentation.** The left part of a summary is denoted by LP and its right part by RP. If the summary LP $\rightarrow$ RP is valid then: $\forall$ j the summary [LP, (A_j, L_j)] $\rightarrow$ RP is valid.

- **Decomposition.** If LP $\rightarrow$ RP is valid then $\forall$ RP' $\subseteq$ RP, LP $\rightarrow$ RP' is valid.

- **Union.** We consider two valid summaries LP $\rightarrow$ RP1 and LP $\rightarrow$ RP2 with the same left part, but different right parts RP1 and RP2. The property of union states that the summary LP $\rightarrow$ [RP1, RP2] is valid.

4.3 The Discovery Algorithm

The algorithm intended for the discovery of summaries can be limited to extended FDs with a single attribute in the right part. This is justified by the fact that any valid summary can be decomposed into such summaries (property of decomposition) and is the composition of summaries having a single attribute in their right parts (property of union).

The algorithm is based on an iteration over tuples t of the considered relation (R). It computes the minimal set S of summaries which are valid on R. This set is minimal in the sense that summaries in S have a minimum number of attributes in their left parts. Other valid summaries can be obtained by a left extension of a summary in S (i.e., new attributes with labels are added to the left part according to the property of augmentation).

At the beginning of the algorithm, the set S is empty. Tuples t of relation R are successively processed. The invariant of the loop is: "from S, it is possible to obtain any summary valid on already processed tuples". The progression consists in constructing the set S_t made of summaries which are valid for t and to merge S and S_t in order to obtain a new set S. These two aspects are dealt with in sections 4.3.1 and 4.3.2 and the resulting algorithm is given in 4.3.3.

4.3.1 Principle for the Determination of S_t

For a given tuple t, S_t is made of summaries which are valid on {t} and whose left part involves a single attribute. This is sufficient since: i) any summary which is valid on {t} is the left extension of one of such summaries (next property), ii) the left extension of any such summary is valid on {t} (property of augmentation).

Property. (A_1, L_1), ..., (A_p, L_p) $\rightarrow$ (A_{p+1}, L_{p+1}) is valid on t $\Rightarrow \exists$ i ($1 \leq i \leq p$) such that (A_i, L_i) $\rightarrow$ (A_{p+1}, L_{p+1}) is valid on t.

4.3.2 Merging S and S_t

This merge delivers the new set S. When S is empty (it is the case at the beginning of the algorithm), set S_t is delivered (S = S_t). If S is not empty, the merge necessitates to compute the set S' made of summaries compatible with S and S_t. The

construction of S' needs to access each summary r of S_t: $(A_i, L_{i,u}) \rightarrow (A_j, L_{j,v})$ and to compare r with summaries in S having $(A_j, L_{j,v})$ as a right part (which are acceptable summaries). Three exclusive cases must be investigated:

case 1: if there is no acceptable summary in S, the next summary in S_t is processed.

case 2: if $(r \in S)$ then $S' = S' \cup \{r\}$.

case 3: acceptable summaries which are already a left extension of r are added to S'. Moreover, since any left extension of r is valid for t, and any left extension of an acceptable summary is valid for already processed tuples, it is possible to extend the ones by the others. Consequently, the left part of r is extended using each left part of acceptable summaries (as many extensions as acceptable summaries). The extension is performed only when the acceptable summary does not already refer to A_i (otherwise no extension is possible because the obtained summary would refer to A_i twice and would not be meaningful). The obtained summaries are added to S'.

However, some redundancy may result from the introduction in S' of summaries which are left extensions of summaries already in S' (cf. next example).

Example 5. $S_t = \{(A_1, L_{1,u}) \rightarrow (A_3, L_{3,w}); (A_2, L_{2,v}) \rightarrow (A_3, L_{3,w})\}$, $S = \{(A_2, L_{2,v}) \rightarrow (A_3, L_{3,w})\}$ and $S' = \varnothing$. After an access to $(A_1, L_{1,u}) \rightarrow (A_3, L_{3,w})$, $S' = \{(A_1, L_{1,u}), (A_2, L_{2,v}) \rightarrow (A_3, L_{3,w})\}$ (case 3). After an access to $(A_2, L_{2,v}) \rightarrow (A_3, L_{3,w})$, $S' = \{(A_1, L_{1,u}), (A_2, L_{2,v}) \rightarrow (A_3, L_{3,w}); (A_2, L_{2,v}) \rightarrow (A_3, L_{3,w})\}$ (case 2). The first summary in S' is redundant because it is the extension of the second one♦

4.3.3 The Algorithm

At the beginning of the algorithm, the first tuple t of R is accessed and $S = S_t$. In addition if S is empty after the process of a tuple, the algorithm is stopped and the result is empty (since no summary is valid for already processed tuples, no summary can be expected to be valid on R). We obtain the algorithm:

```
begin
   first tuple t in R is accessed;
   S := determination_of_valid_summaries(t);
   for each remaining tuple t in R do
      St := determination_of_valid_summaries(t);
      S' := ∅;
      for any summary r: (Ai, Li,u) → (Aj, Lj,v) in St do
         if (r ∈ S) then S' := S' ∪ {r};                    /* case 2 */
         else for each summary LP → (Aj, Lj,v) in S do      /* case 3 */
                  if ((Ai, Li,u) ∈ LP) then S' := S' ∪ {LP → (Aj, Lj,v)}
                  elsif there is no k such that (Ai, Li,k) ∈ LP then
                        S' := S' ∪ {LP, (Ai, Li,u) → (Aj, Lj,v)} endif
              enddo
         endif
      enddo
```

```
        S := remove_redundancy(S');
        if S = ∅ then stop endif;
    enddo
end
```

determination_of_valid_summaries(t). This function determines the summaries of the type $(A_i, L_{i,u}) \rightarrow (A_j, L_{j,v})$ which are valid on t. They are characterized by $\mu_{L_{i,u}}(t.A_i) \leq \mu_{L_{j,v}}(t.A_j)$.

remove_redundancy(S'). This function delivers non redundant summaries from S'. A summary is delivered if it is not the left extension of a summary already in S'.

In terms of data accesses, the complexity of this algorithm is linear (each tuple is accessed once). Concerning the function determination_of_valid_summaries(t), it is possible to show that its complexity in the worst case (in terms of implication values to compute) is $(n*k)^2$ where n is the number of attributes and k the number of labels for each attribute. In practise n is less than 20 and k is around 5. The complexity of the computation of S' appears very tricky and its study is beyond the scope of this paper.

Example 6. We consider a relation PLAY which describes professional basket-ball players. The labels are *young*, *tall* and *high* (this last label applies to the number of points scored by the player during a given period).

Player	**(age, *young*)**	**(score, *high*)**	**(height, *tall*)**
Paul	0.9	0.8	0.5
Peter	0.7	0.7	0.9

The algorithm accesses the tuple describing Paul: $S_t = S = \{$(score, *high*) → (age, *young*); (height, *tall*) → (age, *young*); (height, *tall*) → (score, *high*)$\}$. The tuple describing Peter is then accessed: S_t is $\{$(age, *young*) → (score, *high*); (age, *young*) → (height, *tall*); (score, *high*) → (age, *young*); (score, *high*) → (height, *tall*)$\}$. The merging of S and S_t delivers the new set S = $\{$(age, *young*), (height, *tall*) → (score, *high*); (score, *high*) → (age, *young*)$\}$. Relation PLAY satisfies "the *younger* and the *taller* a player, the *higher* his score" and "the *higher* the score, the *younger* the player"♦

5 Conclusion

This paper has introduced a new type of summary for knowledge discovery in databases. These summaries offer a qualitative description of the database content and are based on extended FDs. They are situated in the framework of the relational model of data and follow the pattern: "∀ t in R, the more A it is, the more B it is" where A and B are conjunctions of linguistic labels (*young*, *well paid*, ...) defined by fuzzy sets. In this approach, the properties of augmentation, union and decomposition valid on FDs still hold on extended FDs.

The proposed algorithm takes advantage of these properties and assumes that attribute domains are partitioned by fuzzy sets corresponding to linguistic labels. It delivers a minimal set of valid summaries of the form (A_1, L_1), ..., $(A_p, L_p) \rightarrow (A_{p+1}, L_{p+1})$, ..., (A_q, L_q) which means that each tuple t in R satisfies the property "t is at least as (L_{p+1} and ... and L_q) as it is (L_1 and ... and L_p)".

In the near future, we aim at experimenting the proposed algorithm with user-defined linguistic labels.

References

1. D. Dubois, H. Prade (1993). On data summarization with fuzzy sets. *Proceedings of the 5th IFSA Congress*, Seoul (Korea), 465-468.
2. D. Rasmussen and R.R. Yager (1996). Summary SQL - A flexible fuzzy query language. *Proceedings of the Flexible Query-Answering Systems Workshop (FQAS'96)*, Roskilde (Denmark), 1-18.
3. P. Bosc, L. Liétard and O. Pivert (1997). Gradualité, Imprécision et Dépendances Fonctionnelles. In *13ème Journées Bases de Données Avancées*, Grenoble (France), 391-413.
4. L.A. Zadeh (1965). Fuzzy sets. *Information and Control*, 8, 338-353.
5. L.A Zadeh (1981). Test-score semantics for natural languages and meaning representation via PRUF. In *Empirical semantics*, (B.B. Rieger ed.), Brockmeyer, Bochum, 1, 281-349.
6. D. Dubois, H. Prade (1985). A review of fuzzy set aggregation connectives. *Information Sciences*, 36, 85-121.
7. L.A. Zadeh (1983). A computational approach to fuzzy quantifiers in natural languages. *Computer Mathematics with Applications*, 9, 149-183.