# Mining Frequent Similar Patterns on Mixed Data

Ansel Y. Rodríguez-González[1,2], José Francisco Martínez-Trinidad[2], Jesús Ariel Carrasco-Ochoa[2], and José Ruiz-Shulcloper[1]

[1] Advanced Technologies Applications Center (CENATAV), Havana, Cuba
{arodriguez,jshulcloper}@cenatav.co.cu
[2] National Institute of Astrophysics, Optics and Electronics (INAOE), Puebla, México
{ansel,fmartinez,ariel}@inaoep.mx

**Abstract.** Frequent Pattern Mining is an important task due to the relevance of repetitions on data, also it is a fundamental step in the Association Rule Mining. Most of the current algorithms for mining frequent patterns assume that two object subdescriptions are similar if and only if they are equal, but in soft sciences some other similarity functions are used. In this work, we focus on the search of frequent patterns on Mixed Data, incorporating similarity between objects. We propose a novel and efficient algorithm to mine frequent similar patterns for a family of similarity functions that fulfill Downward Closure property and we also propose another algorithm for the remaining families of similarity functions. Some experiments over mixed datasets are done, and the results are compared against the ObjectMiner algorithm.

**Keywords:** data mining, frequent pattern, mixed data, similarity functions.

## 1 Introduction

Frequent Pattern Mining is an important task due to the relevance of repetitions on data, also it is a fundamental step of Association Rule Mining [1]. A *frequent pattern* is a combination of feature values (pattern) of the objects of study, that appears in a data set with frequency not less than a user-specified threshold. According to the application area these patterns could represent the user's profiles, modus operandis, common syndromes, risk factors, etc.

The concept of similarity is usually used in soft sciences, like Medicine, Geology, Sociology, etc., as tool to make decisions. For example, in sociological studies we might consider two persons to be similar in terms of their age if they belong to the same generation, which is equivalent to considering two ages to be similar if the absolute value of their difference is at most 5 years. Also we can consider the relation *"is a"* over the feature *Education* as similarity relationship, for example *Doctor is a Bachelor* and *Doctor is a High School Graduated* and *Bachelor is a High School Graduated*. In this case the similarity is asymmetric.

J. Ruiz-Shulcloper and W.G. Kropatsch (Eds.): CIARP 2008, LNCS 5197, pp. 136–144, 2008.

In the previous examples the similarity was used for comparing values of a feature in the objects of study. However, the similarity can be used to compare complete objects or parts of them. For example, we can consider that two parts of objects are similar if they are similar in all the features (*full matching similarity*) or if they are similar in at least 90% of the features.

Most of the current algorithms for mining frequent patterns assume that two object subdescriptions are similar if and only if they are equal (*full equality*), but in soft sciences some other similarity functions are used. Therefore, as we can see in the following example, when one of these algorithms is applied in these circumstances, some frequent patterns could be lost and thus some information could be mislaid. In real problems like obtaining user's profiles, modus operandis, common syndromes and risk factors, this means that some false conclusions could be arrived or some knowledge could be not found.

*Example 1.* Let $\Omega$ be the mixed data collection shown in Table 1, considering the similarity for features *Age* and *Education* as was mentioned before, the similarity for feature values combinations as full matching similarity, and setting the minimum frequent threshold at 0.5. We have that the following feature value combinations: ($A$=25), ($A$=30), ($E$=*High School*), ($E$=*Bachelor*), ($M$=*No*), ($A$=25, $E$=*High School*), ($A$=25, $M$=*No*), ($E$=*High School*, $M$=*No*) and ($A$=25, $E$=*High School*, $M$=*No*) are frequent patterns. If we had considered the full equality then only ($M$=*No*) would have been a frequent pattern.

**Table 1.** Example 1

| $\Omega$ | Age (A) | Education (E) | Married (M) |
|---|---|---|---|
| $O_1$ | 23 | Bachelor | No |
| $O_2$ | 25 | High School | No |
| $O_3$ | 30 | Doctor | Yes |
| $O_4$ | 30 | Bachelor | No |
| $O_5$ | 45 | Doctor | Yes |

In this work, we focus on the search of frequent patterns on mixed data, incorporating the similarity between objects. We propose an efficient algorithm to solve the problem for the families of similarity functions which hold the Downward Closure property and we also propose another algorithm for the remaining families of similarity functions.

## 2 Related Works

In [1] the search of frequent patterns is introduced and it is limited to collections of binary data. However, data collections commonly contain mixed data, i.e., different kinds of data (numerical and non numerical) in the descriptions of objects are combined. The search of frequent patterns on mixed data was introduced in [2], where a fine partitioning over numerical features and a combining of adjacent intervals are proposed. This approach although reduces the

information loss, does not consider the similarity and therefore, as we point out in the introductory example, frequent patterns could be lost.

Since [2] the search of frequent patterns on mixed data has been addressed following two fundamental approaches: discretizing the domain of the numerical features transforming the problem into a binary pattern mining problem [3]; and using fuzzy set theory concepts to manipulate the values of numerical and non numerical features [4]. The discretization procedures of the numerical features are insufficient to solve the problem because sometimes these transformations are artificial, without considering neither the semantic nor similarity and result in changing the data nature. In practice there are numerical features that can not be discretized, for example, all the features that two feature values are similar if the absolute value of their difference is lesser than a threshold, like in geosciences the Bouguer Anomaly and its gradient [5]. The fuzzy set approach is a better approximation for solving the problem because in the definition of linguistic variables the similarity between values of features is incorporated. However, the fuzzy approach does not allow incorporating this information between combinations of features.

Finally in [6], an algorithm (*ObjectMiner*) incorporating the similarity through the restricted family of similarity functions that holds: if two objects are different respect to a feature combination $S_1$ then they are different respect to all feature combination $S_2$, such that $S_1 \subset S_2$, is proposed.

## 3 Notation and Problem Definition

Let $\Omega = \{O_1, O_2, \ldots, O_n\}$ be a data collection. Each object is described by a set of features $R = \{r_1, r_2, \ldots, r_m\}$ and consists of a tuple $(v_1, v_2, \ldots, v_m)$ where $v_i \in D_i$, the domain of $r_i$ $(1 \leq i \leq m)$. A *subdescription* of an object $O$ for a subset of features $S \subseteq R$ denoted $I|_S(O)$, is the projection of the values of $O$ in terms of the features in $S$. Usually $O[r]$ denotes the projection of the values of $O$ on one feature $r \in R$. Each feature $r_i$ has an associated *comparison criterion* $C_{r_i}(x, y)$ (here we assume a Boolean comparison criterion $C_{r_i} : D_i \times D_i \rightarrow \{0, 1\}$), not necessarily symmetric, to evaluate the similarity between $y$ and $x$ such that $C_{r_i}(x, y) = 1$ means that $y$ is similar to $x$ and $C_{r_i}(x, y) = 0$ otherwise. Two common examples of comparison criteria are:

$$C_r(x, y) = \begin{cases} 1 \text{ if } x = y \\ 0 \text{ otherwise} \end{cases} \qquad C_r(x, y) = \begin{cases} 1 \text{ if } |x - y| \leq \varepsilon \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

Each subset of features $S \subseteq R$, $S \neq \emptyset$ has an associated Boolean *similarity function* [7] $f_S$ between subdescriptions of objects of $\Omega$, not necessarily symmetric and not necessarily the same, based on comparison criteria. Given two subdescriptions $P_1 = I|_S(O)$, $P_2 = I|_S(O')$, with $O, O' \in \Omega$, $f_S(P_1, P_2) = 1$ means that $P_2$ is similar to $P_1$ w.r.t. $S$ and $f_S(P_1, P_2) = 0$ otherwise. $\Gamma$

denotes a set of *similarity functions*, where for each subset of features, a similarity function is given. Two common examples of similarity function sets are:

$$\Gamma = \left\{ f_S | S \subseteq R, f_S(I|_S(O), I|_S(O')) = \left\{ \begin{array}{l} 1 \text{ if } \forall r \in S, C_r(O[r], O'[r]) = 1 \\ 0 \text{ otherwise} \end{array} \right\} \right. \quad (2)$$

$$\Gamma = \left\{ f_S | S \subseteq R, f_S(I|_S(O), I_S(O')) = \left\{ \begin{array}{l} 1 \text{ if } \frac{|\{r \in S | C_r(O[r], O'[r]) = 1\}|}{|S|} \geq k \\ 0 \text{ otherwise} \end{array} \right\} \right. \quad (3)$$

For all $S \subseteq R$, $S \neq \emptyset$, $f_S \in \Gamma$, the *frequency of a subdescription* $I|_S(O)$ *in* $\Omega$ *for* $f_S$ is defined as:

$$freq_{f_S,\Omega}(I|_S(O)) = \frac{|\{O' \in \Omega \ | f_S(I|_S(O), I|_S(O')) = 1\}|}{|\Omega|} \quad (4)$$

We say that $I|_S(O)$ is a $\Gamma$*-frequent subdescription* (*frequent similar pattern*) in $\Omega$ if $freq_{f_S,\Omega}(I|_S(O)) \geq minFreq$, where $f_S$ is the similarity function associated to $S$.

The problem of mining frequent similar patterns on a mixed data collection consists in giving a set of objects $\Omega$ described by a set of $m$ features, one comparison criterion for each feature, a set $\Gamma$ of similarity functions that covers all subsets of features (one for each subset of features) and a minimum frequency threshold $minFreq$, finding all $\Gamma$-frequent subdescriptions in $\Omega$.

## 4 Proposed Algorithms

The universe of all the possible sets of similarity functions is divided into two subsets; using the following property:

*Property 1 (Downward Closure).* Consider as before $\Gamma$ and $R$. We say that $\Gamma$ holds *Downward Closure* iff for all $S_1 \subseteq S_2 \subseteq R$; $S_1 \neq \emptyset$; $O \in \Omega$; $f_{S_1}, f_{S_2} \in \Gamma$: $(freq_{f_{S_1},\Omega}(I|_{S_1}(O)) < minFreq) \Rightarrow (freq_{f_{S_2},\Omega}(I|_{S_2}(O)) < minFreq)$.

### 4.1 STreeDC Algorithm

The Downward Closure property ensures that there are not *no-*$\Gamma$*-frequent subdescription* that can be expanded (by adding a new feature) to a $\Gamma$-frequent subdescription. Thus, given a linear order $\prec$ over the features in $R$, each subset of features $S \subseteq R$ can be expanded as $\hat{S} = S \cup \{r\}$ such that $r \in R$, $\forall l \in S, l \prec r$, if the number of $\Gamma$-frequent subdescriptions with respect to $S$ is greater than zero or $S = \emptyset$. As a result of all the possible expansions from the empty set we obtain all $\Gamma$-frequent subdescriptions. The proposed algorithm named *STreeDC* follows this idea.

In order to facilitate the search of all $\Gamma$-frequent subdescriptions respect to each expansion $\hat{S}$ of the feature set $S$, STreeDC builds a structure called $STree_{\hat{S}}$. Each $STree_{\hat{S}}$ is a tree where each path from the root to a leaf represents a subdescription $P$. In each leaf the number of occurrences of the subdescription

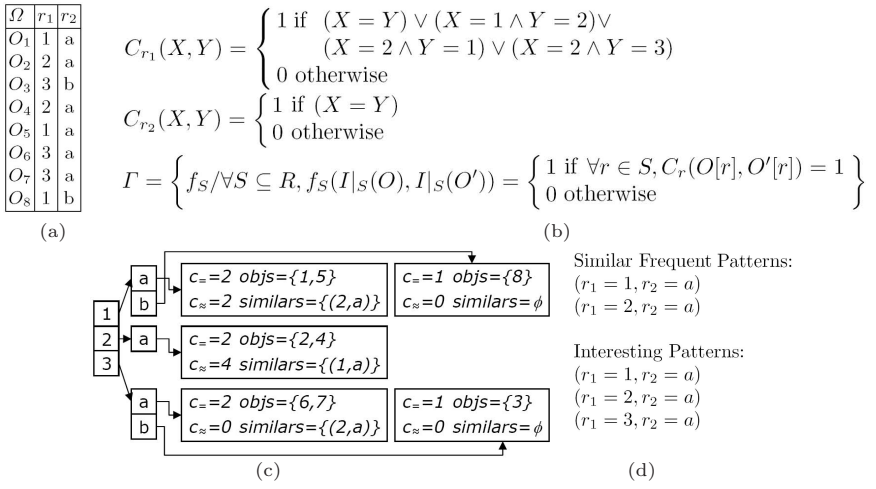


**Fig. 1.** Example of $STree_{\{r_1,r_2\}}$. (a) Collection $\Omega$. (b) Set of similarity functions $\Gamma$ and comparison criteria. (c) $STree_{\{r_1,r_2\}}$ structure. (d) Frequent similar patterns and interesting patterns with respect to the subset of features $\{r_1, r_2\}$ for $minFreq = 0.5$.

**Algorithm 1**: Build STree

**Input**: $STree_S$, $STree_{\{r\}}$: STree structures ($S \cup \{r\}$ is an expansion of $S$)
$minFreq$: minimum support threshold
**Output**: $STree_{\hat{S}}$: STree structure
$STree_{\hat{S}} \leftarrow$empty STree structure
**foreach** *subdescription* $P \in STree_S$ **do**
  **if** $P$ *is* $\Gamma$-frequent *and* $\exists\ P' \in P.similars$, *such that* $P'$ *is* $\Gamma$-frequent **then**
    **foreach** *Object* $O^* \in P.objs$ **do**
      **if** $STree_{\hat{S}}.contain(I|_{\hat{S}}(O^*))$ **then**
        $STree_{\hat{S}}.I|_{\hat{S}}(O^*).c_= \leftarrow STree_{\hat{S}}.I|_{\hat{S}}(O^*).c_= + 1$
      **else**
        $STree_{\hat{S}}.add(O^*)$
**foreach** $P, P' \in STree_{\hat{S}}$, *such that* $P'$ *is similar to* $P$ *with respect to* $S$ *and* $\{r\}$ **do**
  **if** $P'$ *is similar to* $P$ *with respect to* $\hat{S}$ **then**
    $P'.similars \leftarrow P'.similars \cup \{P\}$
**foreach** $P \in STree_{\hat{S}}$ **do**
  **foreach** $P' \in P.similars$, *such that* $P' \neq P$ **do**
    $P'.c_\approx \leftarrow P'.c_\approx + P'.c_=$
**return** $STree_{\hat{S}}$

$P$ in the collection ($P.c_=$), the number of occurrences of similar subdescriptions to $P$ ($P.c_\approx$), a list of subdescriptions of which $P$ is similar [1] ($P.similars$) and the list of objects having a subdescription equal to $P$ ($P.objs$) is stored. In Figure 1 an example of $STree$ is shown.

[1] Observe that $f_{\hat{S}}$ is not necessarily symmetric.

An object $O$ is considered an *interesting object* respect to $S$ if $I|_S(O)$ is a $\Gamma$-frequent subdescription in $\Omega$ or it is similar to a $\Gamma$-frequent subdescription $I|_S(O')$. For building each $STree_{\hat{S}}$, we propose the Algorithm 1, which consists on three steps: I) add to $STree_{\hat{S}}$ all interesting objects respect to $S$ contained in $STree_S$, II) compute the similarity among all subdescriptions stored in $STree_{\hat{S}}$, III) compute the number of occurrences of similar subdescriptions for each subdescription. Finally, in order to find frequent similar subdescriptions in $STree_{\hat{S}}$, it is verified if each subdescription in $STree_{\hat{S}}$ is a $\Gamma$-frequent subdescription.

### 4.2 STreeNDC Algorithm

Unfulfillment of the Downward Closure property eliminates the possibility of pruning the feature subset space, therefore it is necessary to search the $\Gamma$-frequent subdescriptions for all $S \subseteq R$, $S \neq \emptyset$. Notice that steps I and II of *STreeDC* are meaningless in this case.

However, if $STree_S$ with $S \neq \emptyset$ is constructed adding all the objects in the collection instead of applying steps I and II; and the similarity among all pairs of subdescriptions contained in $STree_S$ is computed, this structure contains the information needed to build any $STree_{S'}$, $S' \subseteq S$, $S' \neq \emptyset$. For example, starting from $STree_{\{r_1,r_2\}}$ showed in Figure 1 $STree_{\{r_1\}}$ can be constructed without adding the 8 objects from the collection, as in the step I, but only the 5 subdescriptions contained in $STree_{\{r_1,r_2\}}$, storing in variable $c_=$ of each subdescription $I|_{\{r_1\}}(O)$, the number of occurrences of any subdescription $I|_{\{r_1,r_2\}}(O')$ contained in $STree_{\{r_1,r_2\}}$, such that, $I|_{\{r_1\}}(O') = I|_{\{r_1\}}(O)$. Thus, the number of objects added is reduced as the number of repeated subdescriptions grows. Notice also, that the list of objects associated with each subdescription, is not needed anymore to build a *STree*.

In this case, the proposed algorithm (*STreeNDC*) obtains the $\Gamma$-frequent subdescriptions of all possible reductions of $STree_R$, adding all the objects in the collection. We say that $\check{S} = S - \{r\}$ is a reduction of $S$, $S \subseteq R$, $r \in R$, if $\check{S} \neq \emptyset$ and $\forall l \in S, l \prec r$; and we say that $STree_{\check{S}}$ is a reduction of $STree_S$ if $\check{S}$ is a reduction of $S$ and $STree_{\check{S}}$ is built from $STree_S$. *STreeNDC* algorithm is one efficient solution to the problem of frequent similar pattern mining for collections of objects described by a small set of features.

## 5 Experimentation Results

In this section we report our experimental results and compare *STreeDC* and *STreeNDC* algorithms against the *ObjectMiner* algorithm (provided by the authors of [6]), which is the only previous frequent pattern mining algorithm that allows using a similarity function different from equality. The comparison is done in terms of the execution time and the number of frequent similar patterns for different values of the *minFreq* threshold. Table 2 gives a description of the data collections[2] used in the experiments.

[2] http://archive.ics.uci.edu/ml/datasets.html

**Table 2.** Description of data collections

| Collection | Objects | Numerical Features | Non Numerical Features |
|---|---|---|---|
| *Car Evaluation* | 1728 | 2 | 5 |
| *Contraceptive Method Choice* | 1473 | 2 | 8 |
| *Census* | 32561 | 6 | 9 |
| *Poker Hand* | 1000000 | 5 | 6 |

As comparison criteria for *Age*, *Doors*, *Persons*, *Capital gain* and *Capital loss* the equation (1) right with $\varepsilon = 5, 2, 2, 1000, 1000$ respectively was used, and for the remaining features the equation (1) left was used.

The first experiment (Figure 2) was executed using the similarity function set showed in equation (2), which fulfills the Downward Closure property. Notice that the results of *STreeNDC* were not plotted in Figures 2(c) and 2(d). This is due to the *STreeNDC* high time consuming because of the number of features and objects in the collections. *STreeDC* achieved better performance than the other algorithms for all collections, see Figure 2. This performance was even better than the other algorithms for small values of *minFreq*. The runtime of *STreeDC* is down to 7.1, 5.9, 3.9 y 4.2 times the run time of *ObjectMiner* for the collections respectively.

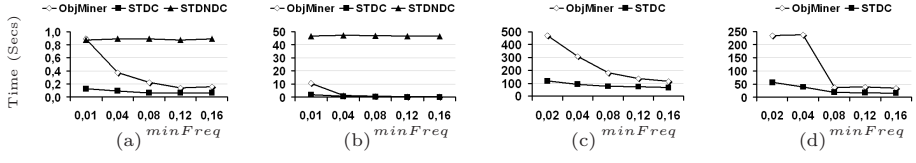


**Fig. 2.** Experiment results using $\Gamma$ which fulfills Downward Closure property. (a) *Car Evaluation*. (b) *Contraceptive Method Choice*. (c) *Census*. (d) *Poker Hand*.

In the second experiment (see Figure 3) as similarity function set the equation (3) with $k = 0.7$, was used. This function does not satisfy the Downward Closure property. As in the first experiment, the results of *STreeNDC* are not plotted in Figures 3(c),(d),(g),(h),(k) and (l). Nevertheless, the behavior of *STreeNDC* for the collections *Car Evaluation* and *Contraceptive Method Choice* (both with few features and few objects) was acceptable. It is worthwhile to underline that the algorithms (*ObjectMiner* and *STreeDC*), which assume that $\Gamma$ fulfills Downward Closure property, can not find all frequent similar patterns. In our experiments, the sets of frequent similar patterns found by *ObjectMiner* were subsets of the frequent similar patterns found by *STreeDC*, for all collections. Notice that *ObjectMiner* loses up to 414 708 $(80, 1\%)$ frequent similar patterns regarding all the existing frequent similar patterns and 210 306 $(67, 1\%)$ w.r.t. frequent similar patterns obtained by *STreeDC* for *Contraceptive Method Choice* collection (Figure 3(f)) and loses up to 4 023 600 $(98, 9\%)$ w.r.t. frequent similar patterns obtained by *STreeDC* for *Census* collection (Figure 3(g)). Another relevant point

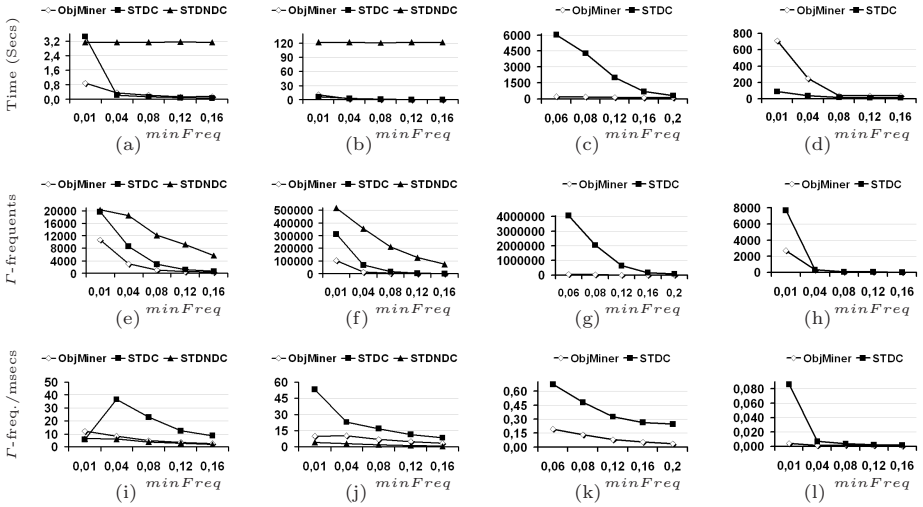


**Fig. 3.** Experiment results using $\Gamma$ which does not fulfill Downward Closure property. (a)(e)(i) *Car Evaluation*. (b)(f)(j) *Contraceptive Method Choice*. (c)(g)(k) *Census*. (d)(h)(l) *Poker Hand*.

is that *STreeDC* in most cases had a better performance, in terms of ratio between $\Gamma$-frequent patterns and runtime. Also, in our algorithms we assume that the similarity functions are not symmetric, and then we evaluate these in both directions. Otherwise, we can reduce the number of similarity function evaluations in half and thus the runtime can be diminished even more.

## 6 Conclusions

In this paper, the importance of use of similarity to mining frequent patterns on mixed data was shown. An efficient structure to store all necessary information about object subdescription and their similarity was presented. Also, a novel and efficient algorithm to mine frequent similar patterns for a family of similarity functions that fulfill Downward Closure property and another algorithm for the remaining families of similarity functions, were proposed. The experimental results have shown a better behavior of our algorithms, in time and number of frequent similar patterns found, than previous work.

## References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: 1993 ACM SIGMOD International Conference on Management of Data, Washington, USA, pp. 207–216 (1993)
2. Srikant, R., Agrawal, R.: Mining Quantitative Association Rules in Large Relational Tables. In: 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Canada, pp. 207–216 (1996)

3. Aouissi, S., Vrain, C., Nortet, C.: QuantMiner: A Genetic Algorithm for Mining Quantitative Association Rules. In: IJCAI 2007, Hyderabad, India, pp. 1035–1040 (2007)
4. Hong, T.P., Lee, Y.C.: An Overview of Mining Fuzzy Association Rules. Fuzzy Sets and Their Extensions: Representation, Aggregation and Models, pp. 397–410. Springer, Heidelberg (2008)
5. Gómez, J., Rodriguez, O., Valladares, S., Ruiz-Shulcloper, J., et al.: Prognostic of Gas-oil Deposits in the Cuban Ophiological Association, Applying Mathematical Modeling. Geophys. Int. 33(3), 447–467 (1994)
6. Dánger, R., Ruiz-Shulcloper, J., Berlanga, R.: Objectminer: A New Approach for Mining Complex Objects. In: ICEIS 2004, Oporto, Portugal, pp. 42–47 (2004)
7. Trinidad, J.M., Shulcloper, J.R., Cortés, M.S.: Structuralization of Universes. Fuzzy Sets and System 112(3), 485–500 (2000)