

Knowledge Discovery in Fuzzy Databases Using Attribute-Oriented Induction

Rafal A. Angryk, Frederick E. Petry
Electrical Engineering and Computer Science Department
Tulane University
New Orleans, LA 70118
USA
{angryk, fep}@eecs.tulane.edu

Abstract

In this paper we analyze an attribute-oriented data induction technique for discovery of generalized knowledge from large data repositories. We employ a fuzzy relational database as the medium carrying the original information, where the lack of precise information about an entity can be reflected via multiple attribute values, and the classical equivalence relation is replaced with relation of the fuzzy proximity. Following a well-known approach for exact data generalization in the ordinary databases [1], we propose three ways in which the original methodology can be successfully implemented in the environment of fuzzy databases. During our investigation we point out both the advantages and the disadvantages of the developed tactics when applied to mine knowledge from fuzzy tuples.

1. Introduction

The majority of current works on data mining describes the construction or application of algorithms performing complex analyses of stored data. Despite the predominant attention on this phase of analysis, because of the extensive volume of data in databases, techniques allowing conversion of raw data into condensed representations has become a practical necessity in many data-mining projects [2-3].

Attribute-Oriented Induction (AOI) [1, 4-12] is a descriptive database mining technique allowing such a transformation. It is an iterative process of grouping of data, enabling hierarchical transformation of similar itemsets stored originally in a database at the low (primitive) level, into more abstract conceptual representations. It allows compression of the original data set (i.e. initial relation) into a generalized relation, which provides

concise and summarative information about the massive set of task-relevant data.

To take advantage of computationally expensive analyses in practice it is often indispensable to start by pruning and compressing the voluminous sets of the original data. Continuous processing of the original data is excessively time consuming and might be expendable, if we are actually interested only in information on abstraction levels much higher than directly reflected by the technical details stored usually in large databases (e.g. serial numbers, time of transactions with precision in seconds, detailed GPS locations, etc.). Simultaneously, the data itself represents information at multiple levels (e.g. Tulane University represents an academic institutions of Louisiana, the university is located in the West South, which is a part of the North American educational system, etc.), and is naturally suitable for generalization (i.e. transformation to a preferred level of abstraction).

Despite the attractive myth of fully automatic data-mining applications, detailed knowledge about the analyzed areas remains indispensable in avoiding many fundamental pitfalls of data mining. As appropriately pointed out in [6], there are actually three foundations of effective data mining projects: (1) the set of data relevant to a given data mining task, (2) the expected form of knowledge to be discovered and (3) the background knowledge, which usually supports the whole process of knowledge acquisition.

Generalization of database records in the AOI approach is performed on an attribute-by-attribute basis, applying a separate concept hierarchy for each of the generalized attributes included in the relation of task-relevant data. The concept hierarchy, which in the original AOI approach is considered to be a part of background knowledge (which makes it the third element of the above mentioned primitives), is treated as an indispensable and crucial element of this data mining technique. Here we investigate the character of knowledge available in the similarity and proximity relations of fuzzy databases and analyze possible ways of its application to the generalization of the information originally stored in fuzzy tuples.

In the next sections we introduce attribute-oriented induction, and briefly characterize crisp and fuzzy approaches to the data generalization; we will also discuss the unique features of fuzzy database schemas that were utilized in our approach to attribute-oriented generalization. In the third part we will present three techniques allowing convenient generalization of records stored in fuzzy databases. The increase in efficiency of these methods over the originally proposed solutions is achieved by taking full advantage of the knowledge about generalized domains stored implicitly in fuzzy database models. Then we will propose a method that allows

multi-contextual generalization of tuples in the analyzed database environments. Finally we will also introduce a simple algorithm allowing generalization of imprecise information stored in fuzzy tuples.

2. Background

2.1. Attribute-Oriented Induction

AOI is a process transforming similar itemsets, stored originally in a database, into more abstract conceptual representations. The transformation has an iterative character, based on the concept hierarchies provided to data analysts. Each concept hierarchy reflects background knowledge about the domain which is going to be generalized. The hierarchy permits gradual, similarity-based, aggregation of attribute values stored in the original tuples. Typically, the hierarchy is built in a bottom-up manner progressively increasing the abstraction of the generalization concepts introduced at each new level (ideally a *0-abstraction-level*, located at the bottom of the hierarchy, includes all attribute values, which occurred in the mined dataset for the particular attribute). To guarantee the reduction of the size of original dataset, each new level of hierarchy has a more coarse-grained structure. In other words, at each new level, there is a smaller number of descriptors, but they have a broader (i.e. general) character. The new concepts, though reduced in number, because of their more general meaning are still able to represent all domain values from the lower abstraction level.

Typically in the AOI technique we initially retrieve a set of task-relevant data (e.g. using a relational database query). Then, we gradually perform actual generalization on the dataset by replacing stored attribute values with more general descriptors. This replacement has an iterative character, where at each stage we are replacing low-level values only with their direct abstracts (i.e. the concepts which are placed at the next abstraction level in the concept hierarchy). After each phase of replacements we merge the tuples that appear identical at the particular abstraction level. To preserve original dependencies among the data at each stage of AOI, we have to accumulate a count of the merged tuples. To be able to keep track of the number of original records that are represented by identical abstract concept, we extend the generalized relation with a new attribute COUNT. At each stage of induction the number stored in the field COUNT informs us about the number of original tuples (i.e. *votes*), which are characterized by the particular abstract entity.

The AOI approach seems to be much more appropriate for detailed analysis of datasets than commonly used simplified, non-hierarchical summarization. The hierarchical character of induction provides analysts with the opportunity to view the original information at multiple levels of abstraction, allowing them to progressively discover interesting data aggregations. In contrast to the flat summarization a gradual process of AOI through concept hierarchies allows detailed tracking of all records and guarantees discovering significant clusters of data at the lowest abstraction level of their occurrence. For that reason the analysts, who use the AOI technique, are able to avoid unnecessary loss of information due to over-generalization. Moreover, in this approach the tuples, which did not achieve significant aggregation at a low abstraction level, rather than being removed from analysis, are gradually further aggregated and so given a chance to reach a meaningful count at one of higher abstraction levels.

Originally the hierarchical grouping proposed by Han and his co-workers [4] was based on tree-like generalization hierarchies, where each of the concepts at the lower level of the generalization hierarchy was allowed to have just one abstract concept at the level directly above it. An example of such a concept hierarchy characterizing a generalization of students' status (*{master of art, master of science, doctorate}* \hat{I} *graduate*, *{freshman, sophomore, junior, senior}* \hat{I} *undergraduate*, *{undergraduate, graduate}* \hat{I} *ANY*) is presented in Figure 1, which appeared in [6].

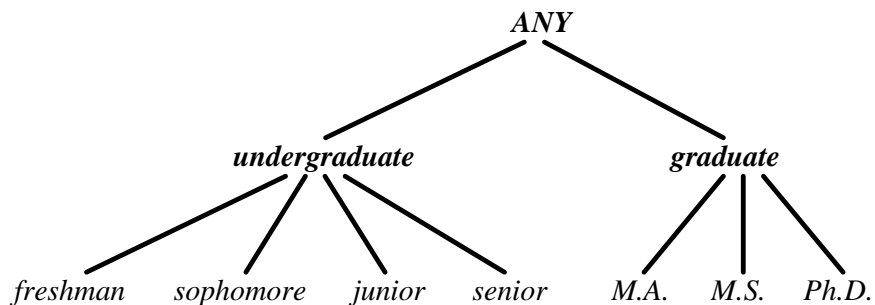


Fig. 1. A concept tree for attribute STUDENT_STATUS [6].

Han [5-6] developed a few elementary guidelines (strategies), which are the basis of effective attribute-oriented induction for knowledge discovery purposes:

1. **Generalization on the smallest decomposable components.** Generalization should be performed on the smallest decomposable components (or attributes) of a data relation.

2. **Attribute removal.** If there is a large set of distinct values for an attribute but there is no higher level concept provided for the attribute, the attribute should be removed in the generalization process.
3. **Concept tree ascension.** If there exists a higher-level concept in the concept tree for an attribute value of a tuple, the substitution of the value by its higher-level concept generalizes the tuples. Minimal generalization should be enforced by ascending the tree one level at a time.
4. **Vote propagation.** The value of the vote of a tuple should be carried to its generalized tuple and the votes should be accumulated when merging identical tuples in generalization.

The work initiated by Han and his co-researchers [4-8] was extended further by Hamilton, Hilderman with their co-workers [9-11], and also by Hwang and Fu in [12]. The attribute-oriented induction methods have been implemented in commercially used systems (DBLearn/DB-Miner [10] and DB-Discover [11]), and applied to a number of research and commercial databases to produce interesting and useful results [5].

Fuzzy hierarchies of concepts were applied to tuples summarization in late nineties by four groups of independent researchers. Lee and Kim [13] used ISA hierarchies, from area of data modeling, to generalize database records to more abstract concepts. Lee [14] applied fuzzy generalization hierarchies to mine generalized fuzzy quantitative association rules. Cubero, Medina, Pons and Vila [15] presented fuzzy gradual rules for data summarization and Raschia, Ughetto and Mouaddib [16-17] implemented SaintEtiq system for data summarization through extended concept hierarchies.

A fuzzy hierarchy of concepts reflects the degree with which a concept belongs to its direct abstract. In addition, more than one direct abstract of a single concept is allowed during fuzzy induction. Each link ℓ in the fuzzy concept hierarchy is a bottom-up directed arc (edge) between two nodes with a certain weight assigned to it. Such a structure reflects a fuzzy induction triple $(c^k, c^{k+1}, \mathbf{m}_{c^k, c^{k+1}})$, where c^k (i.e. an attribute value generalized to the k^{th} -abstraction level), and c^{k+1} (one of its direct generalizers) are endpoints of ℓ , and $\mathbf{m}_{c^k, c^{k+1}}$, being the weight of the link ℓ , represents the strength of conviction that the concept c^k (a source of ℓ) should be qualified as concept c^{k+1} (a target of ℓ) when the data generalization process moves to the next abstraction level. During attribute-oriented fuzzy induction (AOFI) the value of $\mathbf{m}_{c^k, c^{k+1}}$ dictates actually what fraction of a vote, representing original attribute values generalized already to the concept c^k , is to be propagated to its higher-level representation, c^{k+1} .

An example of a fuzzy concept hierarchy (FCH) presented in the Fig. 2. comes from [13]. According to the authors, utilization of the four popular text editors could be generalized as follows (we denote fuzzy generalization of concept a to its direct abstract b with membership degree c as $a \stackrel{c}{\rightarrow} b$):

1st level of abstraction: $\{emacs \stackrel{1.0}{\rightarrow} editor/ 1.0; emacs \stackrel{0.1}{\rightarrow} documentation/ 0.1;$
 $vi \stackrel{1.0}{\rightarrow} editor/ 1.0; vi \stackrel{0.3}{\rightarrow} documentation/ 0.3; word \stackrel{1.0}{\rightarrow} documentation / 1.0;$
 $word \stackrel{0.1}{\rightarrow} spreadsheet/ 0.1; wright \stackrel{1.0}{\rightarrow} spreadsheet/ 1.0\}$

2nd level of hierarchy: $\{editor \stackrel{1.0}{\rightarrow} engineering/ 1.0; documentation \stackrel{0.8}{\rightarrow} engineering / 1.0; documentation \stackrel{1.0}{\rightarrow} business / 1.0;$
 $spreadsheet \stackrel{0.8}{\rightarrow} business / 1.0; spreadsheet \stackrel{1.0}{\rightarrow} engineering / 0.8; spreadsheet \stackrel{1.0}{\rightarrow} business/ 1.0\}$

3rd level of hierarchy: $\{engineering \stackrel{1.0}{\rightarrow} w / 1.0; business \stackrel{1.0}{\rightarrow} w / 1.0\}$

Fuzzy hierarchies of concepts allow a more flexible representation of real life dependencies. A significant weakness of the above-mentioned approaches is a lack of automatic preservation of *exact vote propagation* at each stage of attribute-oriented fuzzy induction (AOFI). To guarantee exactness of the data summarization via AOFI we need to assure that each record from the original database relation will be counted exactly once at each of the levels of the fuzzy hierarchy.

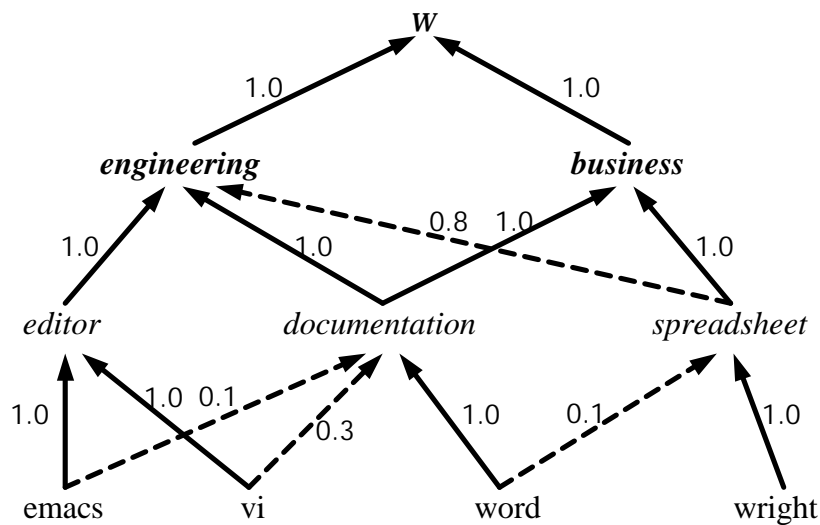


Fig. 2. A fuzzy ISA hierarchy on computer programs [13].

This issue was successfully resolved by utilization of consistent fuzzy concept hierarchies [18-19], which preserve *exact vote propagation* due to their completeness and consistency. Speaking formally, to preserve com-

pleteness and consistency of the AOFI process for all adjacent levels, C^k and C^{k+1} , in fuzzy concept hierarchy the following relationship must be satisfied:

$$\sum_{j=1}^{\|C^{k+1}\|} m_{c^k c_j^{k+1}} = 1.0, c^k \in C^k, c_j^{k+1} \in C^{k+1}$$

In other words, the sum of weights assigned to the links leaving any single node in a fuzzy concept hierarchy needs to be 1.0. Preservation of the above property prevents any attribute value (or its abstract) from being counted more or less during the process of AOFI (i.e. consistency of the fuzzy induction model). It also guarantees that a set of abstracts concepts at each level of hierarchy will cover all of the attribute values that occurred in the original dataset (i.e. completeness of the model). In effect, we are guaranteed to not lose count of the original tuples when performing fuzzy induction on their attribute values.

Formally, each non-complete fuzzy concept hierarchy can be transformed to the complete generalization model via simple normalization of membership values in all outgoing links of the hierarchical induction model.

Summarizing, for the purpose of formal classification, we can distinguish three basic types of generalization hierarchies that have been used to date:

1. *Crisp Concept Hierarchy* [1, 4-12], where each attribute variable (concept) at each level of hierarchy can have only one direct abstract (its direct generalization) to which it fully belongs (there is no consideration of the degree of relationship).
2. *Fuzzy Concept Hierarchy* [13-17], which allows the reflection of degree with which one concept belongs to its direct abstract and more than one direct abstract of a single concept is allowed. Because of the lack of guarantee of exact vote propagation, such a hierarchy seems to be more appropriate for approximate summarizations of data, or to the cases where subjective results are to be emphasized (we purposely want to modify the role or influence of certain records).
3. *Consistent Fuzzy Concept Hierarchy* [18-19], where each degree of membership is normalized to preserve an exact vote propagation of each tuple when being generalized.

2.2. Similarity- and Proximity-based Fuzzy Databases

The similarity-based fuzzy model of a relational database [20-23] is actually a formal generalization of the ordinary relational database [24]. The fuzzy model, based on the max-min composition of a fuzzy similarity relation, which replaces the classical equivalence relation coming from the theory of crisp sets, was further extended by Sheno and Melton [25-28] with the concept of the proximity relation.

Important aspects of fuzzy relational databases are: (1) allowing non-atomic domain values, when characterizing attributes of a single entity and (2) generation of equivalence classes based on the specific fuzzy relations applied in the place of traditional identity relation.

As mentioned above, each attribute value of the fuzzy database record is allowed to be a subset of the whole base set of attribute values describing a particular domain. Formally, if we denote a set of acceptable attribute values as D_j , and we let d_{ij} to symbolize a particular (j^{th}) attribute value, characterizing the i^{th} entity. Instead of $d_{ij} \hat{\mathbf{I}} D_j$ the more general case $d_{ij} \subseteq D_j$ is allowed, i.e. any member of the power set of accepted domain values can be used as an attribute value except the null set. A fuzzy database relation is a subset of the cross product of all power sets of its constituent attributes $2^{D_1} \times 2^{D_2} \times \dots \times 2^{D_m}$. This allows representation of inexactness arising from the original source of information. When a particular entity's attribute cannot be clearly characterized by a single descriptor, this aspect of uncertainty can be reflected by multiple attribute values.

Another feature characterizing proximity fuzzy databases is substitution of the ordinary equivalence relation, defining the notion of redundancy in the ordinary database, with an explicitly declared proximity relation of which both the identity and similarity relations are actually special cases. Since the original definition of fuzzy proximity relations (also called tolerance relations) was only reflexive and symmetric, which is not sufficient to effectively replace the classical equivalence relation, the transitivity of proximity relation was added [27]. This was achieved by extending the original definition of a fuzzy proximity relation to allow transitivity via similarity paths (sequences of similarities), using Tamura chains [29]. So the α -proximity relation used in proximity databases has the following structure:

If P is a proximity relation on D_j , then given an $\mathbf{a} \hat{\mathbf{I}} [0, 1]$, two elements $x, z \hat{\mathbf{I}} D_j$ are \mathbf{a} -similar (denoted by $x P_{\mathbf{a}} z$) if and only if $P(x, z) \geq \mathbf{a}$, and are said to be \mathbf{a} -proximate (denoted by $x P_{\mathbf{a}}^+ z$) if and only if they are (1) either \mathbf{a} -similar or (2) there exists a sequence $y_1, y_2, \dots, y_m \hat{\mathbf{I}} D_j$, such that $x P_{\mathbf{a}} y_1 P_{\mathbf{a}} y_2 P_{\mathbf{a}} \dots P_{\mathbf{a}} y_m P_{\mathbf{a}} z$.

Table 1. Proximity table for a domain COUNTRY.

	Canada	USA	Mexico	Colombia	Venezuela	Australia	N. Zealand
Canada	1.0	0.8	0.5	0.1	0.1	0.0	0.0
USA	0.8	1.0	0.8	0.3	0.2	0.0	0.0
Mexico	0.5	0.8	1.0	0.4	0.2	0.0	0.0
Colombia	0.1	0.3	0.4	1.0	0.8	0.0	0.0
Venezuela	0.1	0.2	0.2	0.8	1.0	0.0	0.0
Australia	0.0	0.0	0.0	0.0	0.0	1.0	0.8
N. Zealand	0.0	0.0	0.0	0.0	0.0	0.8	1.0

Each of the attributes in the fuzzy database has its own *proximity table*, which includes the *degrees of proximity (a-similarity)* between all values occurring for the particular attribute. A proximity table for the domain of COUNTRIES, which we will use as an example for our further analysis, is presented in the Table 1.

The proximity table can be transformed by Tamura chains to represent such an *a-proximity relation*. Results of this transformation are seen in Table 2.

Table 2. α -proximity table for a domain COUNTRY.

	Canada	USA	Mexico	Colombia	Venezuela	Australia	N. Zealand
Canada	1.0	0.8	0.8	0.4	0.4	0.0	0.0
USA	0.8	1.0	0.8	0.4	0.4	0.0	0.0
Mexico	0.8	0.8	1.0	0.4	0.4	0.0	0.0
Colombia	0.4	0.4	0.4	1.0	0.8	0.0	0.0
Venezuela	0.4	0.4	0.4	0.8	1.0	0.0	0.0
Australia	0.0	0.0	0.0	0.0	0.0	1.0	0.8
N. Zealand	0.0	0.0	0.0	0.0	0.0	0.8	1.0

Now the disjoint classes of attribute values, considered to be equivalent at a specific *a-level*, can be extracted from the table. They are marked by shadings in Table 2.

Such separation of the equivalence classes arises mainly due to the sequential similarity proposed by Tamura. For instance, despite the fact that the proximity degree, presented in Table 1, between the concepts *Canada* and *Venezuela* is *0.1*, the *a-proximity* is *0.4*. Using the sequence of the original proximity degrees: $CanadaP_aMexico = 0.7 \hat{U} MexicoP_aColombia = 0.4 \hat{U} ColombiaP_aVenezuela = 0.8$, we obtain $CanadaP_a^+Venezuela = 0.4$, as presented in Table 2.

The transformation based on the sequences of proximities converts the original proximity table back to a similarity relation [30] as in a similarity database model. The practical advantage of the proximity approach comes from the lack of necessity to preserve the *max-min transitivity* when defining the proximity degrees. This makes a proximity table much easier for a user to define. The \mathbf{a} -proximity table, although based on the proximity table, is generated dynamically only for the attribute values which were actually used in the fuzzy database.

3. Attribute-Oriented Induction in Fuzzy Databases

In view of the fact that a fuzzy database model is an extension of the ordinary relational database model, the generalization of fuzzy tuples through the concept hierarchies can always be performed by the same procedure as presented by Han [6, 8] for ordinary relational databases. In this work however we focus on the utilization of the unique features of fuzzy databases in the induction process. First of all, due to the nature of fuzzy databases we have an ability to re-organize the original data to the required level of detail (by the merge of records, considered to be identical at a certain *a-cut level*, according to the given similarity relation), so then we can start attribute-oriented generalization from the desired level of detail. This approach, while valuable in removing unnecessary detail, must be applied with caution. When merging the tuples in fuzzy databases according to the equivalence at the given similarity level (e.g. by using SELECT queries with a high threshold level), we are not able to keep the track of the number of original data records to be merged to a single tuple. Lack of such information may result in significant change of balance among the tuples in the database and lead to the erroneous (not reflecting reality) information presented later in the form of support and confidence of the extracted knowledge. This problem, which we call a *count dilemma* is derived from the principle of *vote propagation* and can be easily avoided by performing extraction of initial data table at the very detailed level (i.e. $\mathbf{a}=1.0$), where only identical values are merged (e.g. values *England* and *Great Britain* will be unified). So then no considerable number of records would be lost as the result of such redundancy removal, which one way or another is a self-contained part of the data pre-processing phase in many of data mining activities.

3.1. Extraction of multi-level concept hierarchies

The generation of an α -proximity relation for a particular domain D_j , in addition to providing a fuzzy alternative to an equivalence relation, also allows extraction of a partition tree, which can then be successfully employed by non-experts to perform attribute-oriented induction.

From the placement of shadings in the Table 2, we can easily observe that the equivalence classes marked in the table have a nested character. As in [30], each α -cut (where $\alpha \in (0, 1)$) of the fuzzy binary relation (Table 2) creates disjoint equivalence classes in the domain D_j . If we let P_α denote a single equivalence class partition induced on the domain D_j by a single α -level-set, through the increase of the value of α to α' we are able to extract the subclass of P_α , denoted $P_{\alpha'}$ (a refinement of the previous equivalence class partition). A nested sequence of partitions $P_{\alpha^1}, P_{\alpha^2}, \dots, P_{\alpha^k}$, where $\alpha^1 < \alpha^2 < \dots < \alpha^k$, may be represented in the form of a partition tree, as in Figure 3.

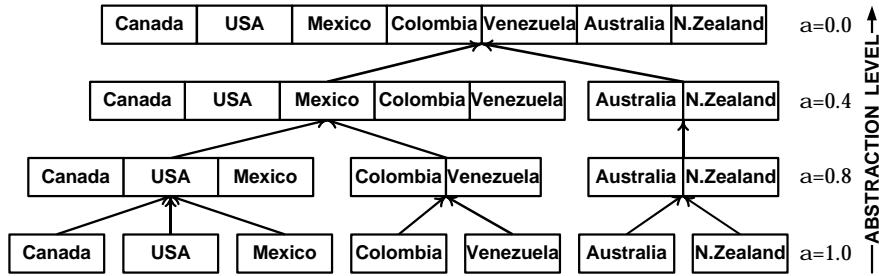


Fig. 3. Partition tree of domain COUNTRY, built on the basis of Table 2.

This nested sequence of partitions in the form of a tree has a structure identical with that of a crisp concept hierarchy applicable for AOI. Since the AOI approach is based on the reasonable assumption that an increase of degree of abstraction makes particular attribute values appear identical, it seems to be appropriate to allow the increase of conceptual abstraction in the concept hierarchy to be reflected by the decrease of α values (representing degrees of similarity between original attribute values), as long as the context of similarity (i.e. dimensions it is measured in) is in full agreement with the context (directions) of performed generalization (e.g. Figure 4). The lack of abstraction (0-abstraction level) at the bottom of generalization hierarchy complies with the 1-cut of the α -proximity relation ($\alpha=1.0$) from the fuzzy model and can be denoted as $P_{1.0}^+$. In other words, it is the

level where only those attribute values that have identical meaning (i.e. synonyms) are aggregated.

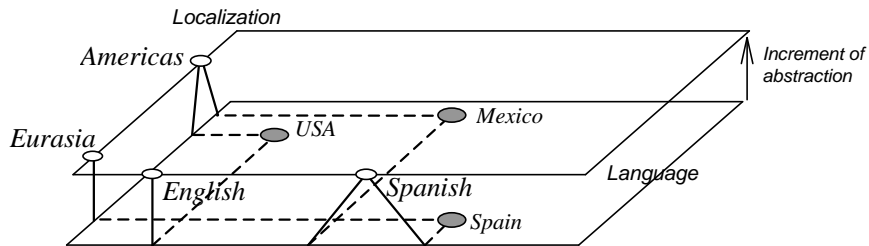


Fig. 4. Example of two different contexts of generalization.

The only thing differentiating the hierarchy in the Figure 3 from the crisp concept hierarchies applicable for AOI is the lack of abstract concepts, which are used as the labels characterizing the sets of generalized (grouped) concepts. To create a complete set of the abstract labels it is sufficient to choose only one member (the original value of the attribute) of every equivalence class P at each level of hierarchy (reflected by α), and assign a unique abstract descriptor to it. Sets of such definitions (original value of attribute and value of α linked with the new abstract name) can be stored as a database relation (Table 3), where the first two attributes create a natural key for this relation.

Table 3. Table of abstract descriptors (for Figure 3).

ATTRIBUTE VALUE	ABSTRACTION LEVEL (α)	ABSTRACT DESCRIPTOR
<i>Canada</i>	<i>0.8</i>	<i>N. America</i>
<i>Colombia</i>	<i>0.8</i>	<i>S. America</i>
<i>Australia</i>	<i>0.8</i>	<i>Oceania</i>
<i>Canada</i>	<i>0.4</i>	<i>Americas</i>
<i>Australia</i>	<i>0.4</i>	<i>Oceania</i>
<i>Canada</i>	<i>0.0</i>	<i>Any</i>

The combination of partition tree in Figure 3 and the relation of abstract descriptors (Table 3) allows us to create the classical generalization hierarchy in the form of Figure 5.

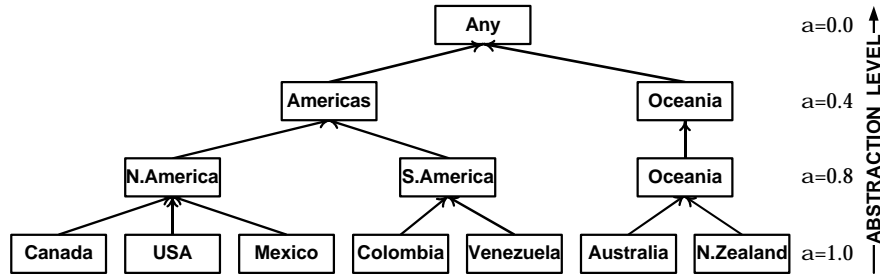


Fig. 5. Crisp generalization hierarchy formed using Tables 4 and 7.

The disjoint character of equivalence classes generated from the α -proximity (i.e. similarity) table does not allow any concept in the hierarchy to have more than one direct abstract at every level of generalization hierarchy. Therefore this approach can be utilized only to form a crisp generalization hierarchy. Such a hierarchy, however, can be then successfully applied as a foundation to the development of a fuzzy concept hierarchy – by extending it with additional edges to represent partial membership of the lower level concepts in their direct abstract descriptors. Depending on the values of assigned memberships, data-analysts can generate consistent or inconsistent fuzzy concept hierarchies.

A partition tree generated directly from the originally defined proximity table (without transforming it first to the α -proximity table as presented in section 2.2) has a structure much less useful for AOI than the solution discussed above. Such a graph (since it does not preserve a tree-like structure), although extractable, usually has a rather complicated structure. Due to the lack of the max-min transitivity requirement in fuzzy proximity relation, partitions of proximity generated by α -cuts (when we increment α) do not always have a nested character. Generated equivalence classes may overlap at the same α -level (e.g. Table 4). In consequence, the extracted graph of partitions often contains the same attribute value assigned multiple times to different equivalence classes at the same abstraction level (α). Furthermore, links between nodes (i.e. proximity partitions, i.e. equivalence classes) may cross the neighboring levels of the hierarchy, which is in contradiction with requirement of minimal tree ascension introduced by Han. When utilizing a partition hierarchy extracted directly from proximity table, we are taking a risk that some tuples would not be counted at the particular stage of induction, which in result may cancel correctness of the obtained results.

Table 4. Overlapping partitions of proximity generated by 0.4-cut.

	<i>Canada</i>	<i>USA</i>	<i>Mexico</i>	<i>Colombia</i>	<i>Venezuela</i>	<i>Australia</i>	<i>N. Zealand</i>
<i>Canada</i>	1.0	0.8	0.5	0.1	0.1	0.0	0.0
<i>USA</i>	0.8	1.0	0.8	0.3	0.2	0.0	0.0
<i>Mexico</i>	0.5	0.8	1.0	0.4	0.2	0.0	0.0
<i>Colombia</i>	0.1	0.3	0.4	1.0	0.8	0.0	0.0
<i>Venezuela</i>	0.1	0.2	0.2	0.8	1.0	0.0	0.0
<i>Australia</i>	0.0	0.0	0.0	0.0	0.0	1.0	0.8
<i>N. Zealand</i>	0.0	0.0	0.0	0.0	0.0	0.8	1.0

An original proximity table can be however successfully utilized when applying other approaches to AOI, as presented in the remaining part of section 3.

3.2. Generation of single-level concept hierarchies

Both similarity and proximity relations can be interpreted in terms of fuzzy classes $P(x)$ [31], where memberships of other elements (in our case other attribute values) in each of the fuzzy classes $P(x)$ are derived from the rows in the proximity (or similarity) relations. In other words the grade of membership of attribute value y in the fuzzy class $P(x)$ (e.g. a fuzzy representative of attribute value x , or of the set of such values), denoted by $\mu_{P(x)}(y)$, is $x P_a y$ (or $x P_a^+ y$ for the similarity relation). This alternative view on fuzzy binary relations provides motivation for development of two other approaches. These may be more convenient for a user, who does not have the expertise to initiate the whole induction process from the basics and would like to use some support derived from existing proximity tables. Both of the techniques to be described require some knowledge about generalized domains, but not at an expert's level.

3.2.1. Induction based on typical elements

In practice, when generalizing attribute values, there are usually distinguished subsets of lower-level concepts that can be easily assigned to the particular abstract concepts even by users who have only ordinary knowledge about the generalized domain. At the same time the assignment of the other lower-level values is problematic even among experts. Using the terminology presented in [32] we will say that each attribute has a domain

(allowed values), a range (actually occurring values) and a typical range (most common values). In this section we apply this approach to the generalization process. When we have an abstract concept we can often identify its *typical direct specializers*, which are the elements clearly belonging to it (e.g. we all would probably agree that the country *Holland* can be generalized to the concept *Europe* with 100% surety). This can be represented as a core of the fuzzy set (abstract concept) characterized by a membership of 1.0 . However, there are usually also lower-level concepts, which cannot be definitely assigned to only one of their direct abstracts (e.g. assigning the *Russian Federation* fully to the abstract concept *Asia* would probably raise some doubts, as almost one fifth of this country lies west of the Urals). We call such cases *possible direct specializers* and define them as the concepts occurring in the group of lower level concepts characterized by the given abstract descriptor (fuzzy set) with the membership $0 < m \leq 1$. Such elements create the support of a fuzzy set and are to be interpreted as the range of the abstract concept.

In short, the idea behind this approach is to define initially the abstract concepts via choosing their basic representative attribute values (i.e. typical representative specializers) and then to use the proximity table to extract a more precise definition of the abstract class. For such extraction we assume a certain level of similarity (\mathbf{a}), which should be interpreted as a level of precision reflected in our abstract concept definition.

Using this approach we initially characterize each new abstract concept as a set of its typical original attribute values with the level of doubt about its other possible specializers reflected by the value of \mathbf{a} . Then we select the fuzzy similarity class created from the α -cut of similarity relation for these pre-defined typical specializers and assess if this class fits well our expectations. Obviously some background knowledge about a generalized domain is now necessary so that the data analyst could point out typical direct specializers. However, there is still a significant support provided by the proximity table allowing smooth generalization of the most problematical cases. When we characterize an abstract concept by more than one typical element, we must also choose an intersection operator which best fits our preferences. This is necessary so that we would be able to extract a unified definition in the case when the typical elements (i.e. fuzzy classes) are overlapping.

For instance, if we predefine the abstract concept *North America* by its two typical countries *Canada* and *USA* with the level of similarity $\mathbf{a}=0.6$, and with the operator *MAX* to reflect our optimistic approach, we can derive from Table 1 that:

$North\ America = MAX(Canada_{0.6}, USA_{0.6}) = \{MAX(Canada/1.0, USA/0.8, Mexico/0.5; Canada/0.8; USA/1.0; Mexico/0.8) = \{Canada/1.0; USA/1.0; Mexico/0.8\}$.

Defining now *Middle America* as the cut of fuzzy class *Colombia* on the 0.4-level we can extract from the proximity table:

$$Middle\ America = Colombia_{0.4} = \{Mexico/0.4; Colombia/1.0; Venezuela/0.8\}.$$

As a result we can derive the fuzzy concept hierarchy and even modify the generalization model to become consistent through the normalization of derived memberships:

$$North\ America = \{Canada/1.0; USA/1.0; Mexico/\frac{0.8}{0.8+0.4}\} = \{Canada/1.0; USA/1.0; Mexico/0.67\}$$

$$Middle\ America = \{Mexico/\frac{0.4}{0.8+0.4}; Colombia/1.0; Venezuela/0.8\} = \{Mexico/0.33; Colombia/1.0; Venezuela/0.8\}.$$

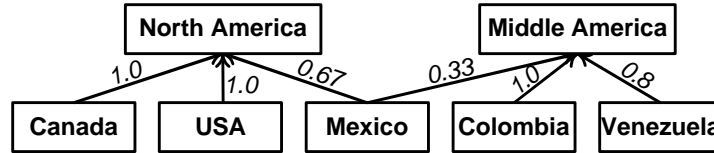


Fig. 6. Consistent fuzzy generalization hierarchy built by extraction of partial knowledge stored in the proximity relation for the attribute COUNTRY.

As distinct from the previous approach, here users are given much more freedom in defining abstracts. It is totally their decision as to which lower-level concepts will be aggregated and to what abstraction level. They can modify two elements to control results of the generalization process: (1) sets of typical elements, and (2) level of acceptable proximity between these values (α). Depending on the choice the extracted definitions of abstract concepts may have different characteristics (Table 5).

Table 5. Characteristics of abstract definitions extracted from the proximity table.

Quantity of typical elements \ Similarity level (α)	LOW (close to 0)	HIGH (close to 1)
SMALL (close to 1)	Spread widely	Precise (Short)
LARGE (close to the total number of attribute values)	Confusing (Error suspected)	Precise (Confirmed)

When extracting overlapping definitions of abstract concepts from a proximity table, a fuzzy concept hierarchy results and one must be extremely careful to keep the definitions semantically meaningful. For instance it makes no sense to pre-define two or more general concepts at a level of abstraction so high, that they are interpreted almost as identical. Some basic guidelines are necessary when utilizing this approach:

1. We need to assure that the intuitively assumed value of α extracts the cut (subset) of attribute values that corresponds closely to the definition of the abstract descriptor we desired. The strategy for choosing the most appropriate level of α -cut when extracting the abstract concept definitions comes from the principle of minimal generalization (minimal concept tree ascension strategy in [6]), which translates to minimal proximity level decrease in our approach. Accepting this strategy we would recommend always choosing the definition extracted at the highest possible level of proximity (biggest α), where all pre-defined typical components of desired abstract descriptor are already considered (i.e. where they occur first time).
2. The problem of selecting appropriate representative elements without external knowledge about a particular attribute still remains, however it can now be supported by the analysis of the values stored in the proximity table. Choosing typical values and then extracting detailed definition from the similarity (or proximity) table will make AOFI more accessible for non-experts. By using common knowledge they may be able to point out a few typical elements of a generalized concept even while lacking expert knowledge necessary to characterize a particular abstract in detail.
3. Moreover, we should be aware that if the low-level concepts, pre-defined as typical components of the particular abstract descriptor, do not occur in the common proximity class, then the contexts of the generalized descriptor and the proximity relation may be not in agreement

and revision of the proximity table (or the abstract concepts) is necessary.

This approach allows us to place at the same level of the concept hierarchy, abstract concepts that were extracted with different levels of proximity (different α -values). As a result this achieves more effective (i.e. compressed) induction. However, allowing such a situation especially when using a similarity (i.e. α -proximity) table, we must always remember that the abstract concepts derived from the similarity relation have a nested character. Placement of one abstract concept at the same level of abstraction with another, which is its actual refinement, may lead to the partial overlapping of partitions. This is in contradiction with the character of a similarity relation. This restriction may not always appear in the case of a proximity relation, as it has an intransitive character.

3.2.2. Induction with context oriented on the single attribute values

As in the previous approach, this technique also leads to the generation of one-level hierarchies. It is derived from a slightly different interpretation of the extracted fuzzy similarity (or proximity) classes. The biggest advantage of this method is its ability to perform generalization of all original attribute values in the context reflected in the proximity table, but from the point of view of the relevance of all of these attribute values with respect to the distinguished one (i.e. the one being used as a basis of the extracted fuzzy class).

Let us use a simple example to illustrate this. From the proximity relation (Table 1) we can extract a single row, representing a fuzzy proximity class for the attribute value Canada (as presented in Table 6).

Table 6. Fuzzy class extracted from Table1; it represents proximity of different countries to Canada.

	Canada	USA	Mexico	Colombia	Venezuela	Australia	N. Zealand
Canada	1.0	0.8	0.5	0.1	0.1	0.0	0.0

Now we can generate subsets of this fuzzy class's domain (which is actually the whole set of acceptable attribute values), defining disjoint ranges of acceptable membership values:

$$\text{Canada-similar} = \{\text{Canada}_{[0.6, 1.0]}\} = \{\text{Canada, USA}\}$$

$$\text{Canada -semi-similar} = \{\text{Canada}_{[0.1, 0.6]}\} = \{\text{Mexico, Colombia, Venezuela}\}$$

$$\text{Canada -un-similar} = \{\text{Canada}_{[0.0, 0.1]}\} = \{\text{Australia, N. Zealand}\}.$$

The size of the assumed ranges is dependent on the preferences of data analyst. Smaller ranges generate larger number of abstract classes, but may reduce the number of lower-level concepts in each of the extracted abstracts. The classes presented above provide us sufficient information to perform generalization of all values occurring in the attribute COUNTRY from the point of view of the similarity of these values to the country *Canada* (Figure 7). Obviously values *Canada* and *USA* will be generalized as *Canada-similar* concepts, countries more distant as *semi-similar* and *un-similar*. Since degrees of proximity were already utilized to extract these three subsets, we have not inserted them in the concept hierarchy.

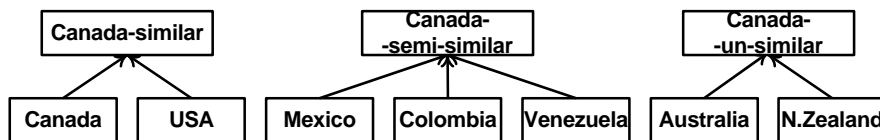


Fig. 7. Value-similarity based generalization.

Technically speaking, we simply sliced the single fuzzy class extracted from the proximity relation into layers reflecting levels of the relevance of the attribute values to the original attribute value, which then became a center of our attribute-oriented induction process.

These two approaches we have discussed above allow us to form only one-level generalization hierarchies or to derive the generalized concepts at the first level of abstraction in the concept hierarchy. Each of abstract concepts defined with this method is a generalization of original attribute values, and therefore cannot be placed at the higher level of the concept hierarchy.

The inability to derive multi-level hierarchical structures does not stop this approach from being appropriate and actually very convenient for rapid data summarization or something we call – selective attribute-oriented generalization. To quickly summarize the given data set we may actually prefer to not perform gradual (hierarchical) generalization, but to replace it with a one-level hierarchy covering whole domain of attribute values. Such “flat hierarchies” still represent dependencies between the attribute values which were originally characterized with the proximity relation; although these dependencies are now restricted to a certain extent, as reflected by the user’s preferences. These flat hierarchies can also be successfully utilized as a foundation for further hierarchical induction, as discussed in the next section.

3.3. Multi-context AOI

The previously discussed approaches, despite being very convenient (and almost automatic) to use, have one common limitation. As described both similarity and proximity relations can be successfully utilized when building concept hierarchies. However the nature of the generalization based on these particular hierarchies can only reflect the context represented by the original similarity or proximity tables. Each of the attribute values can be considered basically as a point in a multi-dimensional space, reflecting multiple contexts of possible inductions (e.g. countries can be aggregated by province, and then by continent, or by the official native language, and then by the linguistic groups). Utilization of the pre-defined proximity tables restricts the context of an AOFI to the dimensionality originally reflected by these tables. This might be limiting for any users who want to perform the AOFI in a different context than represented in proximity tables, or need to reduce the number of similarity dimensions (i.e. generalization contexts) at a certain stage of hierarchical induction. The techniques presented above do not provide flexibility in allowing such modifications. Obviously, the data-mining analysts can be allowed to modify the values in proximity table in their own user-views of the fuzzy database to represent the similarity between the concepts (attribute values) in the context of their interests. The advantage of such modifications is that now these new proximity tables can be used to merge the records that are similar in the context of their interests.

The last two methods presented in the previous section allowed solely one-level generalization. To extend AOI to a higher abstraction level we need to simply define new similarity or proximity tables reflecting similarity between the generalized concepts. There is nothing preventing users from defining such tables in the new context, as long as they have sufficient knowledge about generalized domain to be able to correctly describe such dependencies. We can even halt the induction process performed with the multilevel hierarchy, presented in the Figure 5, at any level and build new similarity (or proximity) tables reflecting the distance between abstract concepts in a totally different context than those primarily represented by the original hierarchy. In Table 7 we present such a similarity relation, reflecting similarities at the higher level of abstraction and in a context different than the one originally represented by Table 1.

Table 7. New similarity-table reflecting relationship between concepts from 1^{st} -abstraction level, according to the placement on Northern and Southern hemisphere of the Earth.

	N. America	S. America	Oceania
N. America	1.0	0.5	0.5
S. America	0.5	1.0	0.7
Oceania	0.5	0.7	1.0

By cutting the hierarchy from Figure 5 after the first level of generalization and introducing new abstract names, which better fit the relation presented in Table 7, we can generate a new concept hierarchy allowing gradual induction from the 1^{st} -abstraction level to the 3^{rd} -level in the new context. Then it will be merged with layers cut from Figure 5 to perform AOI based on the modification of generalization context at the 1^{st} level of the abstraction. The hierarchy constructed in this manner is seen in Figure 8.

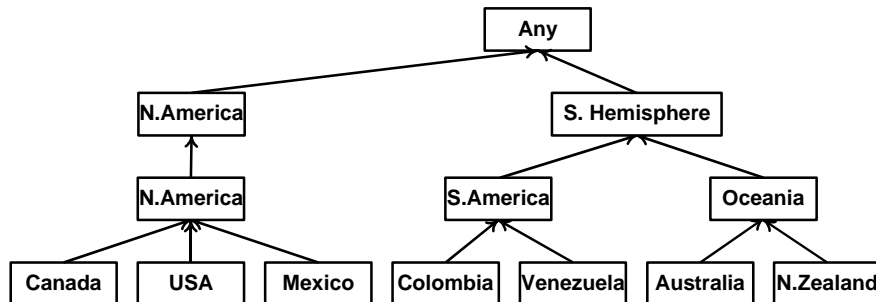


Fig. 8. Multi-contextual generalization hierarchy built from merge of two similarity tables starting at different abstraction levels.

4. AOI over imprecise data

There are two actual carriers of imprecision's representation in the fuzzy database schema. First, as already mentioned in background section, is the occurrence of multiple attribute values. Obviously, the more descriptors we insert to characterize a particular record in the database, the more imprecise is its depiction. The uncertainty about the description is also im-

explicitly reflected in the similarity of values characterizing any particular entity. When the reports received by Department of Homeland Security state that a particular individual was seen recently in the following countries $\{Iran, Germany, Australia\}$, they cause more doubt about the person's current location than in the case where he/she was seen in $\{Iran, Iraq, Afghanistan\}$, since this information would be rather immediately interpreted as "*Middle East*". There are exactly the same number of attribute values in each case and they all are at the same level of abstraction, however the higher similarity of reports provided in the second set results in the higher informativeness carried by the second example. So the imprecision of the original information is actually reflected both in the number of inserted descriptors for particular attributes and in the similarity of these values.

A simplified characterization of data imprecision can be provided by an analysis of the boundary values. The range of imprecision degree ranges between 0 (i.e. the lack of uncertainty about stored information) and infinity (maximum imprecision). Following the common opinion that even flawed information is better than the lack of information, we then say that imprecision is at a maximum when there are no data values at all. Since our fuzzy database model does not permit empty attributes we will not consider this case further. A minimum (zero value) is achieved with a single attribute value. If there are no other descriptors, we have to assume the value to be an exact characterization of the particular entity's feature. The same minimum can be also accomplished with multiple values, in the case where they all have the same semantics (synonyms). Despite the fact that multiple, identical descriptors further confirm the initially inserted value, they do not lead to further reduction of imprecision, since it already has a minimal value. Therefore the descriptors, which are so similar that they are considered to be identical, can be reduced to a single descriptor. Obviously, some attribute values, initially considered as different, may be treated as identical at the higher abstraction level. Therefore we can conclude that the practically achievable minimum of imprecision depends on the abstraction level of employed descriptors, and can reach its original (absolute) 0 value only at the lowest level of abstraction (for $\alpha = 1.0$ in our fuzzy database model).

During attribute-oriented induction we usually employ concept hierarchies that have single attribute values in their leaves (at the 0-abstraction level). Therefore the generalization of tuples with single descriptors is straightforward. The problem arises however when there are multiple attribute values describing a single entity. Where should we expect to find a drug dealer who, as not-confirmed reports say, was seen recently in $\{Canada, Colombia, Venezuela\}$? Our solution is based on partial vote propaga-

tion, where a single vote, corresponding to one database tuple, is partitioned before being assigned to the concepts placed in the leaves of the concept hierarchy. Now the fractions of vote are assigned to separate 0-level concepts to represent each of the originally inserted attribute values. During AOI all fractions of this vote propagate gradually through multiple levels of generalization hierarchy, the same way as the regular (precise) records do. The only difference is that the tuple with uncertainty has multiple entries to the generalization paths (different leaves of concept hierarchy for different vote's fractions), whereas each of the precise tuples has only one beginning of its generalization path.

The most trivial solution would be to split the vote equally among all inserted descriptors: $\{Canada/0.(3), Colombia/0.(3), Venezuela/0.(3)\}$. This approach however does not take into consideration real life dependencies, which are reflected not only in the number of inserted descriptors, but also in their similarity. We propose replacement of the even distribution of vote with a nonlinear spread, dependent on the similarity and the number of inserted values. Using the partition tree built from the similarity table, we can extract from the set of the originally inserted values those concepts which are more similar to each other than to the remaining descriptors. We call these subsets of resemblances (e.g. $\{Colombia, Venezuela\}$ from the above-mentioned example). Then we use these as a basis for calculating a distribution of vote's fractions. An important aspect of this approach is extraction of these subsets of similarities at the lowest possible level of their occurrence, since the nested character of α -proximity relation guarantees that above this α -level they are going to co-occur every time. Repetitive extraction of such subsets could unbalance the original dependencies among inserted values.

The algorithm to achieve it is straightforward. Given (1) a set of attribute values inserted as a description of particular entity, and (2) a hierarchical structure reflecting Zadeh's partition tree for the particular attribute; we want to extract a table, which includes (a) the list of all subsets of resemblances from the given set of descriptors, with (b) the highest level of α -proximity of their common occurrence. The algorithm uses preorder recursive traversal for searching the partition tree. The partition tree is searched starting from its root and if any subset of the given set of descriptors occurs at the particular node of the concept hierarchy we store the values that were recognized as similar, and the adequate value of α . An example of such a search for subsets of resemblances in a tuple with values $\{Canada, Colombia, Venezuela\}$ is depicted in Figure 9. Numbers on the links in the tree represent the order in which the particular subsets of similarities were extracted.

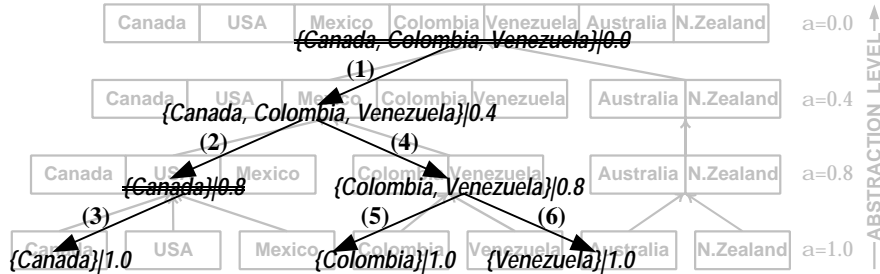


Fig. 9. Subsets of similar values extracted from the original set of descriptors.

An output with subsets of resemblance generated for this example is presented in Table 8.

Table 8. Subsets of resemblances and their similarity levels for the analyzed example.

OUTPUT	Comments
$\{Canada, Colombia, Venezuela\} 0.0$	STORED
$\{Canada, Colombia, Venezuela\} 0.4$	UPDATED
$\{Canada\} 0.8$	STORED
$\{Canada\} 1.0$	UPDATED
$\{Colombia, Venezuela\} 0.8$	STORED
$\{Colombia\} 1.0$	STORED
$\{Venezuela\} 1.0$	STORED

After extracting the subsets of similarities (i.e. *subsets of resemblances*), we apply a summarization of α values as a measure reflecting both the frequency of occurrence of the particular attribute values in the subsets of similarities, as well as the abstraction level of these occurrences. Since the country *Canada* was reported only twice, we assigned it a grade 1.4 ($1.0+0.4$). The remaining attribute values were graded as follows:

$$Colombia / (1.0 + 0.8 + 0.4) = Colombia / 2.2$$

$$Venezuela / (1.0 + 0.8 + 0.4) = Venezuela / 2.2$$

In the next step, we added all generated grades ($1.4+2.2 + 2.2 = 5.8$) to normalize grades finally assigned to each of the participating attribute values:

$$Canada / (1.4/5.8) = Canada / 0.24$$

$$Colombia / (2.2/5.8) = Colombia / 0.38$$

$$Venezuela / (2.2/5.8) = Venezuela / 0.38$$

This leads to the new distribution of the vote's fractions, which more accurately reflects real life dependencies than a linear weighting approach. The results obtained are presented in the Figure 10.

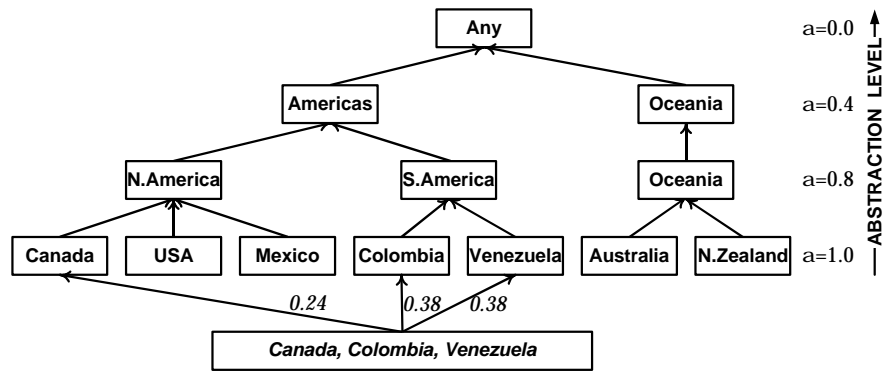


Fig. 10. Partial Vote Propagation for records with uncertainty.

Normalization of the initial grades has a crucial meaning for preservation of the generalization model's completeness. It guarantees that each of the records is represented as a unity, despite being variously distributed at each of the generalization levels.

During an AOI process all fractions of the vote may gradually merge to finally become unity at a level of abstraction high enough to overcome the originally occurring imprecision. In such a case, we observe that there is a removal of imprecision from the data due to its generalization. Such connection between the precision and certainty seems to be natural and has already been noted by researchers [33-34]. In general, very abstract statements have a greater likelihood of being valid than more detailed ones.

4. Conclusions

In this paper we discussed three possible ways that similarity and proximity relations, implemented as the essential parts of fuzzy databases, can be successfully applied to knowledge discovery via attribute-oriented generalization. We proved that both of these relations, due to their basic properties, could be successfully utilized when building bottom-up oriented concept hierarchies, assuming that both the context of an intended generalization and the perspective represented in the similarity or proximity table are in the agreement. Moreover, we also considered how both of these fuzzy relations could be employed to building one-level concept hierar-

chies, involving the generalization of original attribute values into a representative group of concepts via single layer fuzzy hierarchies. Finally, we presented an algorithm allowing generalization of imprecise data. More advanced applications of our approach in the areas where fuzzy databases are the most applicable (i.e. spatial databases) remain for further work to illustrate practical use of these approaches.

References

1. Han J, Fu Y (1996) Exploration of the Power of Attribute-Oriented Induction in Data Mining. In: Fayyad UM, Piatetsky-Shapiro G, Smyth P, Uthurusamy R (eds.) *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, Menlo Park, CA, pp. 399-421.
2. Goebel M, Gruenwald L (1999) A Survey of Data Mining & Knowledge Discovery Software Tools. *ACM SIGKDD Explorations Newsletter* 1(1), pp. 20-33.
3. Feelders A, Daniels H, Holsheimer M (2000) Methodological and practical aspects of data mining, *Information & Management* 37, pp. 271-281.
4. Cai Y, Cercone N, Han J (1989) Attribute-Oriented Induction in Relational Databases, In: *Proc. IJCAI-89 Workshop on Knowledge Discovery in Databases*, Detroit, MI, pp. 26-36.
5. Cai Y, Cercone N, Han J (1991) Attribute-Oriented Induction in Relational Databases. In: Piatetsky-Shapiro G, Frawley WJ (eds.) *Knowledge Discovery in Databases*, AAAI/MIT Press, Menlo Park, CA, pp. 213-228.
6. Han J, Cai Y, Cercone N (1992) Knowledge discovery in databases: An attribute-oriented approach. In: *Proc. 18th Int. Conf. Very Large Data Bases*, Vancouver, Canada, 1992, pp. 547-559.
7. Chen MS, Han J, and Yu PS (1996) Data Mining: An Overview from a Database Perspective. *IEEE Transactions on Knowledge and Data Engineering* 8(6), pp. 866-883.
8. Han J, Kamber M (2000) *Data Mining: Concepts and Techniques*. Morgan Kaufmann, New York, NY.
9. Hamilton HJ, Hilderman RJ, Cercone N (1996) Attribute-oriented induction using domain generalization graphs. In: *Proc. 8th IEEE Int'l Conf. on Tools with Artificial Intelligence*, Toulouse, France, pp. 246-253.

10. Carter CL, Hamilton HJ (1998) Efficient Attribute-Oriented Generalization for Knowledge Discovery from Large Databases. *IEEE Transactions on Knowledge and Data Engineering* 10(2), pp. 193-208.
11. Hilderman RJ, Hamilton HJ, Cercone N (1999) Data mining in large databases using domain generalization graphs. *Journal of Intelligent Information Systems* 13(3), pp. 195-234.
12. Hwang HY, Fu WC (1995) Efficient Algorithms for Attribute-Oriented Induction. In: *Proc. 1st International Conference on Knowledge Discovery and Data Mining*, Montreal, Canada, pp. 168-173.
13. Lee DH & Kim MH (1997) Database summarization using fuzzy ISA hierarchies. *IEEE Transactions on Systems, Man, and Cybernetics - part B* 27(1), pp. 68-78.
14. Lee KM (2001) Mining generalized fuzzy quantitative association rules with fuzzy generalization hierarchies. In: *Proc. Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, Vancouver, Canada, pp. 2977-2982.
15. Cubero JC, Medina JM, Pons O, Vila MA (1999) Data Summarization in Relational Databases Through Fuzzy Dependencies. *Information Sciences* 121(3-4), pp. 233-270.
16. Raschia G, Ughetto L, Mouaddib N (2001) Data summarization using extended concept hierarchies. In: *Proc. Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, Vancouver, Canada, pp. 2289 -2294.
17. Raschia G, Mouaddib N (2002) SAINTETIQ: a fuzzy set-based approach to database summarization. *Fuzzy Sets and Systems* 129(2), pp. 137-162.
18. Angryk RA, Petry FE (2003) Consistent fuzzy concept hierarchies for attribute generalization. In: *Proc. IASTED International Conference on Information and Knowledge Sharing (IKS '03)*, Scottsdale, AZ, pp. 158-163.
19. Angryk RA, Petry FE (2003) Data Mining Fuzzy Databases Using Attribute-Oriented Generalization. In: *Proc. 3rd IEEE International Conference on Data Mining (ICDM '03)*, Workshop on Foundations and New Direction in Data Mining, Melbourne, FL, pp. 8-15.
20. Buckles BP, Petry FE (1982) A fuzzy representation of data for relational databases. *Fuzzy Sets and Systems* 7(3), pp. 213-226.
21. Buckles BP, Petry FE (1983) Information-theoretic characterization of fuzzy relational databases. *IEEE Transactions on Systems, Man, and Cybernetics* 13(1), pp. 74-77.
22. Buckles BP, Petry FE (1984) Extending the fuzzy database with fuzzy numbers. *Information Sciences* 34(2), pp. 145-155.

23. Petry FE (1996) *Fuzzy Databases: Principles and Applications*. Kluwer Academic Publishers, Boston, MA.
24. Codd FE (1970) A relational model of data for large share data banks. *Communications of the ACM*, 13(6), pp. 377-387.
25. Sheno S, Melton A (1989) Proximity Relations in the Fuzzy Relational Database Model. *International Journal of Fuzzy Sets and Systems*, 31(3), pp. 285-296.
26. Sheno S, Melton A (1990) An Extended Version of the Fuzzy Relational Database Model. *Information Sciences* 52 (1), pp. 35-52.
27. Sheno S, Melton A (1991) Fuzzy Relations and Fuzzy Relational Databases. *International Journal of Computers and Mathematics with Applications* 21 (11/12), pp. 129-138.
28. Sheno S, Melton A, Fan LT (1992) Functional Dependencies and Normal Forms in the Fuzzy Relational Database Model. *Information Sciences* 60, pp. 1-28.
29. Tamura S, Higuchi S, Tanaka K (1971) Pattern Classification Based on Fuzzy Relations. *IEEE Transactions on Systems, Man, and Cybernetics* 1(1), pp. 61-66.
30. Zadeh LA (1970) Similarity relations and fuzzy orderings. *Information Sciences*, 3(2), pp. 177-200.
31. Dubois D, Prade H (1980) *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, NY.
32. Dubois D, Prade H, Rossazza JP (1991) Vagueness, typicality and uncertainty in class hierarchies. *International Journal of Intelligent Systems* 6, pp. 167-183.
33. Bosc P, Prade H (1993) An introduction to fuzzy set and possibility theory based approaches to the treatment of uncertainty and imprecision in database management systems. In: *Proc. 2nd Workshop on Uncertainty Management in Information Systems (UMIS '94): From Needs to Solutions*, Catalina, CA.
34. Parsons S (1996) Current approaches to handling imperfect information in data and knowledge bases. *IEEE Transactions on Knowledge and Data Engineering* 8(3), pp. 353-372.