Induction of Fuzzy Characteristic Rules

Dan Rasmussen[1] and Ronald R. Yager[2]

[1] Computer Science Department, Roskilde University, DK-4000 Roskilde, Denmark
[2] Machine Intelligence Institute, Iona College, New Rochelle, NY 10801, USA

Abstract. In this paper we present a method for induction of fuzzy characteristic rules from a fuzzy subset. We use a two step method where in the first step we search for candidate predicates which will make up the characteristic rules. For this purpose typical values are used to isolate characteristic intervals for attributes, these intervals will make up the candidate predicates. In the second step we use the candidate predicates to induce the fuzzy characteristic rules.

1 Introduction

One of the primary goals of data mining is to find human interpretable patterns (rules, trees, time series patterns etc.) describing the data in a database. The discovered patterns can be valuable information for decision making and for people who want to understand a new data domain. Rules are frequently used as patterns in data mining and can be used to express different kinds of knowledge [1].

This paper is concerned with fuzzy characteristic rules, which is a fuzzyfication of the usual characteristic rules. For a characteristic rule $(p \rightarrow q)$, the consequence q summarize one or more properties common to all instances of the subpopulation p, where q typically is a conjunction of attribute-value conditions. Generally if for a rule we allow the subpopulation p to be a fuzzy subset and the consequence q to be fuzzy conditions we have a fuzzy rule.

For example in medicine a fuzzy characteristic rule may be used to summarize the symptoms necessary for a given disease. It could be *most people with flu have sneezing and fever* $(flu \rightarrow sneezing \wedge fever)$. Notice the fuzzy concept fever is used as a predicate in the consequent part of the rule. In [4] we show how to evaluate these fuzzy rules.

From an applications viewpoint, a good informative characteristic rule should contain a lot of conjuncted conditions in the consequence part q to describe the subpopulation p. This observation leads us to the interestingness measure, the maximal characteristic descriptions (the maximal conjunctive generalization or MCG) [5], but other measure for rule interestingness (IC++) can be found in [2].

Rules can be either categorical or hedged/quantitative, where quantitative rules are associated with statistical information about the extent of the rule $(p \rightarrow q, (80\%))$, while categorical have no associated hedging information $(p \rightarrow q)$. Fuzzy rules are examples of hedged rules, because they always are associated with a truth degree. The truth degree is a value in the unit interval [0, 1], where 1 means true and 0 false.

In this paper we will introduce a method for induction of fuzzy characteristic rules from a fuzzy subset. The method consist of two independent steps. In the first step we search for candidate predicates which will make up the characteristic rules. For this purpose we will use an algorithm which is restricted to attributes of interval domains. In the second step we induce the characteristic rules from the candidate predicates.

The rest of the paper is organized as follows, in Section 2 and 3 we present a method to isolate the candidate predicates. In Section 2 we talk about typical values for an attribute, the typical values defines characteristic intervals with high object density. In Section 3 we introduce a simple cluster algorithm to isolate the characteristic intervals, which will make up the candidate predicates. The induction algorithm for the fuzzy characteristic rules will be introduced in Section 4. In Section 5 we present some experimental results from the rule induction and at last in Section 6 the conclusion.

2 Typical Values

We will now show how typical values can be used to define characteristic intervals for the attributes. As we earlier have mentioned the characteristic intervals will be used as candidate predicates for the characteristic rules. A typical value for an attribute, can be defined as a value which is close to most of the values found in the population for that attribute. In the paper [6] Yager discusses how we can use fuzzy subsets to represent the typical values and in [7] Yager and Rasmussen introduce a relational fuzzy query language called SummarySQL in which it is possible to query for typical values.

A fuzzy subset is a object set $O = \{\mu_1/o_1 + ... + \mu_i/o_i + ... + \mu_n/o_n\}$, where each object o_i has a degree of membership $\mu_i \in [0, 1]$, this is different from an ordinary set where each object either belongs to the set or not, $\mu_i \in \{0, 1\}$. It has been shown that fuzzy subsets are useful tools to modeling vague or fuzzy concepts like *tall*, *small* and *about 10* in a very intuitive manner [8]. For example to represent the fuzzy concept *the small numbers* from $\mathbf{N}_0$ we will write $\mu_{\text{small}} = \{1/0 + .9/1 + .8/2... + .1/9 + 0/10...\}$. Often we will use a membership function to represent a fuzzy subset $\mu_O : O \rightarrow [0, 1]$, where the argument is an object o_i and the result the membership degree μ_i. For example the degree to which the number 1 belongs to the fuzzy subset *small* is $\mu_{\text{small}}(1) = 0.9$. In this paper we will used fuzzy subsets and membership functions to define typical values and fuzzy predicates.

Let O be a fuzzy subset of objects o_i and let A be an attribute from O. Further let $o_i.a$ denote the value object o_i takes for the attribute A. Then a typical value for an attribute A is a fuzzy subset of the elements $o_i.a$, where the membership degree μ_i indicates the degree of typicality or how *close* $o_i.a$ is to all the values $o_j.a$ in O. We will use the notation $\mu_{\text{typical}[A]} = \{\mu_1/o_1.a + ... + \mu_n/o_n.a\}$ for the typical value of A. To calculate the fuzzy subset $\mu_{\text{typical}[A]}$ we have to define a similarity relation $\approx$: $A \times A \rightarrow I$. For example $x \approx y$ could be $max(0, 1 - |x - y|/\delta)$, where δ is a constant. Fig. 1 shows how to construct a typical value for an attribute A in a fuzzy subset O.

```
input:  fuzzy object set: $O$, attribute: $A$
output: typical value: $\mu_{\text{typical}[A]}$

1) proc findTypicalVal
2)     forall object $o_i \in O$ do
3)         $r_i = \sum_j min(o_i.a \approx o_j.a, \mu_O(o_j)), where\ o_j \in O$
4)         $\mu_i = min(r_i : fcard(O_f), \mu_O(o_i))$
5)         extend $\mu_{\text{typical}[A]}$ with $\mu_i/o_i.a$
6)     end
7)     return $\mu_{\text{typical}[A]}$
8) end
```

Fig. 1. Find the typical value for an attribute A of the fuzzy objet set O.

The description of the algorithm is as follows: Find the object mass r_i in a local area around the value $o_i.a$ for each object $o_i \in O$ by summarizing over $min(o_i.a \approx o_j.a, \mu_O(o_j))$ for all the objects $o_j \in O$ (line 3). The similarity function ensure that the closer the values $o_j.a$ are to $o_i.a$ the more mass they will contribute and because an object o_j should not contribute with more mass than it belongs to the fuzzy subset O we use $min(o_i.a \approx o_j.a, \mu_O(o_j))$. Next find the degree of typicality μ_i as the mass r_i divided by the fuzzy cardinalitye $fcard(O)$ of the total object set O (line 4), where $fcard(O) = \sum_j \mu_O(o_j)$ [8]. We can look at μ_i as the proportion of the objects $o_j \in O$ that satisfies the relation $o_i.a \approx o_j.a$. In this case the degree of typicality μ_i cannot by greater than the membership degree of the object itself $\mu_O(o_i)$ (line 4), but it is possible to argue that the degree of typicality should be independent of $\mu_O(o_i)$. Then the degree of typicality is founded for a value $o_i.a$ then extend the fuzzy subset $\mu_{\text{typical}[A]}$ with $\mu_i/o_i.a$ (line 5). If there exist one or more values $\mu_{i1}/a_{i1}, ..., \mu_{ik}/a_{ik}$ *where* $a_i = a_{i1} = ... = a_{ik}$ then μ_i/a_i will be defined as $max(\mu_i, \mu_{i1}, ..., \mu_{ik})/a_i$.

If we plot the typical value for an attribute and compare it with a frequency plot, we will notice that the typical value is a transformation or a smoothing of the frequency plot, also known as smoothing of the empirical densities. In Fig. 2.a we show a frequency plot of the Na - concentration related to samples from headlamp glass, the samples come from UCI, 1997. In Fig. 2.b we show the typical value plot for the same attribute, in this case the similarity relation $\approx$ was defined as $max(0, 1 - |x - y|/\delta)$, where δ is 10% of the Na range.

Notice, the typical value indicates a coherent area with a high frequency density, this area will be used to define the characteristic interval or in other words, a characteristic interval for an attribute A is an interval I_c where the object density is high. This area can easily be isolated with a simple cluster algorithm

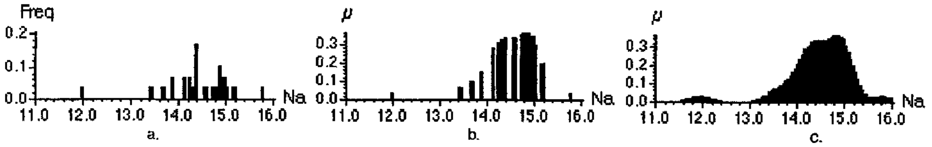

Fig. 2. a. Frequency plot, **b.** typical value plot and **c.** approximated typical value plot of the Na-attribute from the glass type headlamp.

(Fig. 4) and will constitute the candidate predicates for the characteristic rules.

Approximated Algorithm The complexity of the algorithm (Fig. 1) is $O(|O|^2)$, where $|O|$ are the size of the object set O, this is not acceptable if we are talking about several Giga-bytes of data. For this reason we will introduce an approximation for how we can find the typical value by a linear search in the size of the object set.

```
input:  fuzzy object set: $O$, attribute: $A$, interval set: $L = \{L_1, ..., L_n\}$
output: typical value: $\mu_{\text{typical}[A]}$

1)  proc findTypicalValapprox
2)    $cardArray[n] = [0, ..., 0]$
3)    forall object $o_i \in O$ do
4)      $j = findIntervalIndex(o_i.A, L)$
5)      $cardArray[j] = cardArray[j] + \mu_O(o_i)$
6)    end
7)    forall intervals $L_j \in L$ do
8)      $r_j = \sum_k (mean(L_j) \approx mean(L_k)) * cardArray[k],\ wherek \in [1, n]$
9)      $\mu_j = r_j : fcard(O)$
10)     extend $\mu_{\text{typical}[A]}$ with $\mu_j / mean(L_j)$
11)   end
12)   return $\mu_{\text{typical}[A]}$
13) end
```

Fig. 3. Find the approximated typical value for an attribute A of a fuzzy objet set O

The algorithm (Fig. 3) takes as arguments a fuzzy object set O, an attribute A from which we want to generate the typical value and $L = \{L_1, ..., L_n\}$ is a partition of the $rang(A)$, where $L_j \cap L_k = \emptyset$ if $j \neq k$ and $L_1 \cup ... \cup L_n = rang(A)$.

First define an array $cardArray$ (line 2) which represents the fuzzy cardinality of the objects belonging to the interval L_j where j is the index of the array ($cardArray[j]$). Next find the fuzzy cardinality for each interval L_j (line 3). For each interval L_j, calculate the mass r_j in a local area around L_j as the sum of the similarity between the mean values for the interval L_j and L_k times the cardinality of the interval L_k (line 8). Then find the degree of typicality μ_j as r_j divided by the cardinality $fcard(O)$ of the object set O (line 9). Then extend the fuzzy subset $\mu_{\text{typical}[A]}$ with the fuzzy value $\mu_i/mean(L_j)$ (line 10).

The function calculates the degree of typicality for each mean value $mean(L_j)$ also the intervals which does not contain any object values $o_i.a$, this is different from the first function. This approximation will define the typical value as smooth coherent curve for all values, see Fig. 2.c. The complexity of the approximation are $O(|O|+n^2)$, where $|O|$ is the size of the object set O and n the number of intervals $\{L_1, ..., L_n\}$. We can further reduce the complexity to $O(|O| + n \cdot \delta)$ if we only calculate r_j for the index k where $mean(I_k) \in [mean(L_j) - \delta, mean(L_j) + \delta]$ and δ is the distance where the similarity relation is different from 0.

```
input:  typical value: $\mu_{\text{typical}} = \{\mu_1/e_1 + ... + \mu_n/e_n\}$,
        relative minimal typicality: $\alpha$, element weights: $[\sigma_0...\sigma_m]$,
        relative maximum distance $\beta$, maximal succeed fault: $maxSucFault$

output: right index of $I_c$: $e_{\text{right}}$

1)  proc findRightOfCharacteristicInterval
2)     $e_{mode} = findMaxElement(\mu_{typical})$
3)     $right = mode$
4)     $r = mode + 1$
5)     $fault = 0$
6)     $minHeight = \alpha \cdot \mu_{\text{mode}}$
7)     $maxDelta = \beta \cdot \mu_{mode}$
8)
9)     while $(fault > maxSucFault$ or $r > n)$ do
10)       $\delta = [\sigma_0...\sigma_m] \cdot [\mu_{\text{right}}...\mu_{\text{right}} - m] - \mu_r$
11)       if $(minHeight > \mu_r$ or $\delta > maxDelta)$
12)          $fault = fault + 1$
13)       else
14)          $right = r$
15)          $fault = 0$
16)       end
17)       $r = r + 1$
18)    end
19)    return $e_{right}$
20) end
```

Fig. 4. Search for the right edge of a characteristic interval I_c.

3 Isolating the Characteristic Intervals

We will now show how to isolate the characteristic intervals from the typical values by a simple cluster method. The purpose of the cluster algorithm is to isolate the hills which define the typical intervals. Basically the algorithm will work as follows: **1.** First find the index for the maximal value of typicality (the mode). **2.** Then go right from the mode until the typicality is too different from the previous values or the typicality is too small and save the index. **3.** Repeat step 2 but this time go left. The interval defined by the right walk and left walk is then the characteristic interval I_c. More precisely let $\mu_{typical} = \{\mu_1/e_1 + ... + \mu_n/e_n\}$ be a fuzzy subset where the elements $e_1...e_n$ are in increasing order. We then search for the interval $I_c = [e_{left}, e_{right}]$.

Let α define the threshold for the minimal typicality we can accept in I_c relative to the mode, the minimal typicality is shown as vertical lines in Fig. 5.a,b. Let β be the relative accepted maximum distance between the typicality from the previous accepted elements $\{e_0...e_i\}$ and a new element $e_i + 1$. Further let $[\sigma_0...\sigma_m]$ be weights for the previous m closest elements $\{e_i...e_{i-m}\}$, where $\sum_i \sigma_i = 1$, $[\sigma_0...\sigma_m] \cdot [\mu_i...\mu_{i-m}] = \sigma_0 \cdot \mu_i + ... + \sigma_m \cdot \mu_{i-m}$ is the dot product of the vectors and $[\sigma_0...\sigma_m] \cdot [\mu_i...\mu_{i-m}] - \mu_{i+1}$ is the distance between the m previous accepted elements $e_i...e_{i-m}$ and the new element e_{i+1}. Then the following algorithm Fig. 4 will return the right element of the characteristic interval I_c.

The parameter $maxSucFault$ makes the algorithm flexible for errors or missing values in $\mu_{typical}$. For example to isolate the cluster in Fig. 2.b we have to ignore the missing value for $e_i = 14$ and the parameter $maxSucFault$ indicate the maximal succeed fault we will accept in I_c. To find the left edge we only have to change the direction of the function so it searches from e_{mode} to e_{left} and the pair e_{left} and e_{right} defines the cluster for the characteristic interval $I_c = [e_{left}, e_{right}]$. Fig. 5.a,b shows two characteristic interval founded width the cluster algorithm, where $\alpha = .2, \beta = .5, [\sigma_0\ \sigma_1] = [.8\ .2]$ and $maxSucFault$ is set to 5% of the range. If we not are interested in characteristic intervals which are too broad, we just have to raise the α parameter and by modifying β and the weights $[\sigma_0 ... \sigma_m]$ it is possible to make the cluster algorithm more sensitive to separate double hills. Further an array with lot of weights will make a high dependence between the previous accepted elements in I_c and a new element, so only new elements with similar typicality will be accepted.

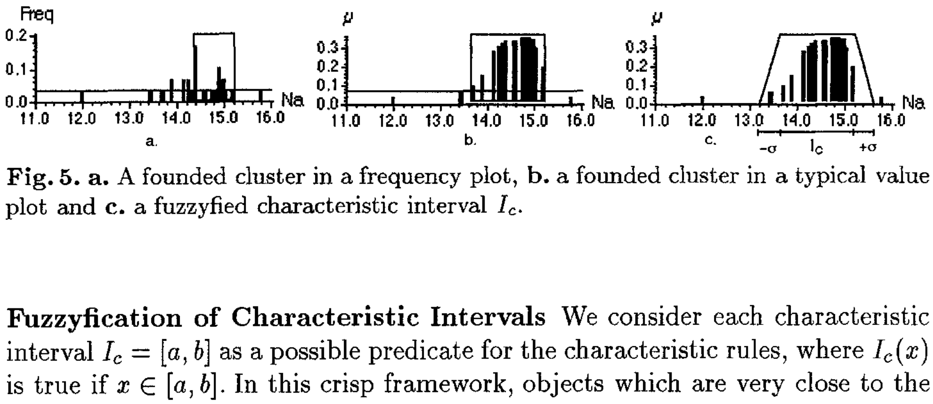

Fig. 5. a. A founded cluster in a frequency plot, **b.** a founded cluster in a typical value plot and **c.** a fuzzyfied characteristic interval I_c.

Fuzzyfication of Characteristic Intervals We consider each characteristic interval $I_c = [a, b]$ as a possible predicate for the characteristic rules, where $I_c(x)$ is true if $x \in [a, b]$. In this crisp framework, objects which are very close to the interval I_c will always make the predicate false. This approach is very logical, but as human beings we would say that the predicate was true to some degree and it could be valuable information for decisions. In the logical approach the information would be lost.

If we on the other hand use a fuzzy framework, we can express the vague information by fuzzifying the characteristic interval so objects which are close to the interval I_c are true to some degree. For this purpose we will use a member-function $I_c : x \to [0, 1]$ which represent the fuzzyfied characteristic interval. For example the characteristic interval $I_c = [a, b]$ can be fuzzyfied by the trapezoidal function I_c (Equation 1 and Fig. 5.c.):

$$I_c(x) = \begin{cases} x - a/\delta & \text{if } x \in [a - \delta, a] \\ 1 & \text{if } x \in [a, b] \\ \delta/x - b & \text{if } x \in [b, b + \delta] \\ 0 & \text{else} \end{cases} \tag{1}$$

where δ is a constant and define the fuzzy part of the fuzzyfied interval I_c.

4 Induction of Fuzzy Characteristic Rules

Once the candidate predicates (the selectors) are found, we are ready to induce the characteristic rules. The rules are induced from the selectors, in our case the fuzzyfied characteristic intervals. The reason why we find the characteristic intervals useful is because each of them characterizes the domains from which we want to induce the rules. But other selectors could also have been used, they could come from other preprocessing methods or from domain experts and are not restricted to interval domains. We could for example use the fuzzy concepts *low*, *medium* and *high* to define the selectors for the Na-concentration or for the color of a car we could use the fuzzy concepts *reddish* and *bluish*.

The characteristic rules we will search for have the form $p \rightarrow q$ where the subpopulation p is a (fuzzy) subset of a population O and the consequence q is a conjunction of the selectors. Also associated with a characteristic rule is a confidence value $\tau \in [0,1]$ and we write $p \rightarrow q(\tau)$ if the rule is satisfied to degree τ. The subpopulation p from which the characteristic rules are induced have to be defined by the user, for this purpose a fuzzy query language could be useful [4]. The consequence q is a conjunction of fuzzy predicates, the fuzzyfied characteristic intervals I_i, $q = I_1 \wedge ... \wedge I_n$. In traditional fuzzy logic, the truth of conjuncted predicates $(p_1 \wedge ... \wedge p_n)$ is defined as minimum of there truth values $min(p_1, ..., p_n)$. The same definition will be used for the aggregated consequence, so the truth of q is equal to $\mu_q = I_1 \wedge ... \wedge I_n = min(I_1, ..., I_n)$. The confidence τ of a fuzzy rule $p \rightarrow q$ in a population O is defined as the proportion of objects in p that satisfy q and should be calculated as follows:

$$\tau = \frac{fcard(p \wedge q)}{fcard(p)}. \qquad (2)$$

where $fcard(p \wedge q) = \sum_{o_i \in O} min(\mu_p(o_i), \mu_q(o_i))$ and $fcard(p) = \sum_{o_i \in O} \mu_p(o_i)$. Notice, if both the subpopulation p and the consequence q are crisp, $\mu_p(o_i) \in \{0,1\}$, then the confidence τ is equal to the original definition [3]. The complexity of finding the confidence is $O(|O|)$ or $O(|p|)$ if the subpopulation p is known, where $|O|$ is the size of the population O and $|p|$ is the size of the subpopulation p. A nice thing about a fuzzy rule is, that the objects which nearly match the rule also contribute to the confidence of the rule in some degree. Further it is possible to rank the outlier objects regarding how much they break the rule.

The Induction Algorithm The algorithm we use for induction of the characteristic rules can be viewed as a specialization of the general STAR algorithm [5]. To find the consequence q (the descriptions) of a characteristic rule $p \rightarrow q$ the STAR use a function called the *reduced star* $RG(o|\neg p, m)$, where o is an object from the subpopulation p and $\neg p$ is the negative objects from the population O. The function is flexible regarding to different search criteria on p, $\neg p$ and q. An example is a search for maximal characteristic descriptions MCG q, which cover all positive objects in p and is found by specialize selectors derived from o. The possible number of descriptions can be very large and to restrict the search a parameter m is given to ensure, that the reduced star contains no more than a fixed number, m, of descriptions at time.

We use a function $fuzzyStar(S, \gamma)$ similar to the reduced star, where the input is a set of selectors S and a confidence threshold γ. The fuzzyStar search for maximal characteristic descriptions q, where the confidence τ of the rule $p \rightarrow q$ is greater than or equal to the confidence threshold γ. Because we systematically compound the descriptions q of selectors from S, we can efficiently use the confidence threshold γ to restrict the search space. The algorithm guarantees that all maximal characteristic descriptions q with $\tau \geq \gamma$ are found. If necessary other interestingness criteria of the rules can be added as a post process.

Let I_{ij} be a selector of an attribute A_i and let the selector group $G_i = \{I_{i1}, ..., I_{im}\}$ be the set of all selectors belonging to the same attribute A_i. Further let $S = \{G_1, ..., G_k\}$ be a set of selector groups from a subpopulation p. We then search for descriptions of the form $q = I_{k_1 j_1} \wedge ... \wedge I_{k_n j_n}$ which constitute the head of the characteristic rule $p \rightarrow q$, where the indexes $k_i = k_j$ only if $i = j$. The condition of the descriptions q avoid more than one selector for each attribute.

As input the algorithm (Fig. 6) takes a set of selector groups S and a confidence threshold γ. Output is a set of descriptions $Star = \{q_1, ..., q_r\}$. To find the Star we use a beamsearch which successive add new descriptions to the Star constraint by the threshold γ. Basically the beam search uses old accepted descriptions to find new candidates. Then the *star* is found the rules will be made human comprehensible by removing descriptions which are subsumptions of other descriptions and agrees with the MCG. A subsumptions s_i of q is a subset of conjuncted selectors from q.

The search algorithm for the characteristic rules works as follows: Find the $Star$ by iteration over the interval group G_i in S (line 3). For each iteration specialize the current accepted descriptions q in the $Star$ with the selectors from G_i and add the specialized descriptions to the $Star$ if they satisfy the confidence threshold γ (line 6 & 10). In this way the $Star$ constitute the seeds for the new descriptions (line 7 & 11) and non fruitful paths will be pruned. The prune is correct because a descriptions can only be valid, $\tau \geq \gamma$, if all the subsumptions of the descriptions also are valid. The function $confidence(q)$ return the confidence of the rule $p \rightarrow q$ (line 6 & 10). To avoid two predicates for the same attribute we temporarily define $Star_{new}$ (line 4). After each iteration (line 3) $Star_{new}$ contain the new extensions of the $Star$ and will then be added to the $Star$ (line 16). Notice, the structure of the algorithm ensure we do not find permutations of already found descriptions.

Once the valid descriptions q are found the subsumptions should be removed. The test for subsumption can be done efficiently as long we keep the order of the added selectors in q and keep the order of the added descriptions q in the $Star$. Do as follows: **1.** Remove a selector q from the $Star$ in reverse order and keep it, **2.** then delete all the subsumptions of q from the $Star$ **3.** repeat from 1 until the Star is empty. The reverse extraction order ensure we never keep a subsumption and waist time by test it.

The complexity of algorithm (Fig. 6) is bounded by $O(|N| * |p| + |O|)$, where $|O|$ is the size of the population O, $|p|$ the size of the subpopulation p and $|N|$ the sum of possible descriptions q given the selectors I_{ij}. The factor $O(|N| * |p|)$ is the

input: set of selector groups: S, confidence threshold: γ
output: a descriptions set: $Star$

```
1)  procedure fuzzyStar
2)     $Star = \{\}$
3)     forall selector group $G \in S$ do
4)        $Starnew = \{\}$
5)        forall selector $I \in G$ do
6)           if $(confidence(I) \geq \gamma)$ then
7)              $Star_{new} = Star_{new} \cup \{I\}$
8)              forall descriptions $q \in Star$ do
9)                 $q_{new} = q \wedge I$
10)                if $(confidence(q_{new}) \geq \gamma)$ then
11)                   $Star_{new} = Star_{new} \cup \{q_{new}\}$
12)                end
13)             end
14)          end
15)       end
16)       $Star = Star \cup Star_{new}$
17)    end
18)    return $Star$
19) end
```

Fig. 6. Search for characteristic rules in a subpopulation p.

possible numbers of confidence tests and it takes $O(|O|)$ to find subpopulation p. The quantity $|N|$ can be exponential in the numbers of attributes A_i, where $i \in \{1, .., k\}$. Let $|G_i|$ be the numbers of selectors belonging to the selector group i then we can calculate $|N|$ as:

$$|N| = \sum_{i=1}^{k} N_i. \tag{3}$$

where $N_i = |G_i| * (1 + N_{i-1})$ and $N_0 = 0$. As we told $O(|N| * |p| + |O|)$ is a upper bound and the actual numbers of confidence test depends of the data distribution and the threshold γ.

5 Experiments and Results

A prototype of the system was developed and implemented, algorithm (Fig. 1) is used to find the typical values. The prototype was tested on a database, which can be found at UCI-1997, containing different types of glass.

The first example (Table 5.1) show different characteristics of window glass (class 2) where the confidence threshold is set to 0.9. The tuples in the table represent the descriptions q of a classification rule $p \rightarrow q$ and there confidence τ. The confidence τ of a rule can be found in the right most field, the remaining fields contain values which represent the conjuncted characteristic intervals and if no value is presented it means the total range (any value). For example the first tuple represent the rule $Class = 2 \rightarrow Al \in [0.65; 2.15] \wedge K \in [0; 0.9] \wedge Ka \in [0; 0.15], (0.95)$. We can consider the rules as different characteristic views q_i on the same subpopulation p and they can be ranked regarding to an interestingness measure [2].

Na	Al	Si	K	Ba	τ
	[0.65; 2.15]		[0; 0.9]	[0; 0.15]	0.95
[12.2; 14.3]			[0; 0.9]		0.91
	[0.65; 2.15]	[71.7; 73.6]			0.91
[12.2; 14.3]	[0.65; 2.15]				0.91
[12.2; 14.3]				[0; 0.15]	0.91

Fig. 7. Induced characteristic rules from glass type 2.

Notice, it can be difficult to find the optimal confidence threshold for a data set. If we raise the confidence threshold we also expect a lower complexity of the rules and if the complexity of the rules are too low we maybe lose interesting knowledge. At the same time if we chose the confidence threshold too low then the rules only characterize small parts of the data and the numbers of rules will raise and make it less human comprehensible.

By repeating the induction process we can find a set of characteristic rules which together cover the subpopulation p. The rules will constitute generalized view of p and we have used the following procedure to the generalization: **1.** Find the different characteristic views q_i of the subpopulation p, then we get a table like Table 7. **2.** Chose and save an appropriate view and then find the outliers, the objects which do not belong to the view. **3.** Repeat the induction process 1. but this time on the outliers and stop if the size of the outliers have reached a minimal boundary. Then the process stop, the saved characteristic rules constitute a generalized table over the objects in the original table.

Table 8 is an example of such a table where the confidence threshold was set to 0.8 for each repetition. It is also possible to adjust the confidence threshold in progress to control the quantity of each rule. Note the second tuple t_2 in Table 8 is a generalization of the first tuple t_1 and the confidence τ should be read as the part of objects which satisfy t_2 and not t_1. The total part of objects which satisfy t_2 are equal to $0.8 + 0.17 = 0.97$. The structure of the generalized table are in some ways similar to the table induced from attribute oriented induction [3] the big different are our tables usually contain empty fields (any) and the tuples can be generalizations of other tuples like t_2.

Ri	Na	Al	Si	K	Ba	Fe	τ
[1.515; 1.522]	[12.2; 14.3]	[0.65; 2.15]	[71.7; 73.6]	[0; 0.9]	[0; 0.15]		0.80
		[0.65; 2.15]		[0; 0.9]			0.17
	[12.2; 14.3]				[0; 0.15]	[0; 0.01]	0.03

Fig. 8. The generalized table of the objects glass type 2.

The last example show how a fuzzy subpopulation can be compressed to a generalized table where the characteristic intervals are fuzzyfied (table 9). In this case we defined the subpopulation p as *"the objects which belong to class 2 and have a refractive index less or similar 1.52"* ($RI <\approx 1.52$). The characteristic interval $\approx [a; b]$ should be read as about $[a; b]$, and $\approx [a; b]$ is defined as Equation 1, where δ is 10% of the range. Note the generalized table contains two main groups, the objects with low iron concentration $Fe \in\approx [0; 0.01]$ and the objects with a high iron concentration $Fe \in\approx [0.08; 0.36]$. The last tuple in Table 9 is a generalization of the two others.

Table 5.5 represent the outliers from the rules in the generalized table 10. Notice, the outliers satisfy the fuzzyfied rules to some degree of this reason we

RI	Na	Mg	Al	Si	K	Ba	Fe	τ
≉[1.515; 1.522]	≉[12.2; 14.3]	≉[2.7; 4]	≉[0.65; 2.15]	≉[71.7; 73.6]	≉[0; 0.9]	≉[0; 0.15]	≉[0; 0.01]	0.48
≉[1.515; 1.522]	≉[12.2; 14.3]	≉[2.7; 4]	≉[0.65; 2.15]	≉[71.7; 73.6]	≉[0; 0.9]	≉[0; 0.15]	≉[0.08; 0.36]	0.30
			≉[0.65; 2.15]		≉[0; 0.9]	≉[0; 0.15]		0.20

Fig. 9. The generalized table for the subpopulation $Class = 2$ and $RI <\approx 1.52$.

consider the outliers as a fuzzy subset. For example the object with the $Id = 110$ are only a week outlier ($\mu = 0.28$) meanwhile object 107 are a strong outlier ($\mu = 0.93$) cause the high Ba concentration.

Id	RI	Na	Mg	Al	Si	K	Ba	Fe	μ
107	1.53	10.7	0	2.1	69.8	0.6	3.15	0.28	0.93
129	1.52	13.6	2.1	1.67	72.2	0.5	0.27	0.17	0.38
85	1.51	14.3	3.1	2.08	72.3	1.1	0	0	0.32
110	1.52	13.7	0	0.56	74.5	0	0	0	0.28

Fig. 10. Outliers from Table 9 represented as a fuzzy subset.

6 Conclusion

In this paper we have presented a method to induce fuzzy characteristic rules from a fuzzy subset. We used a two step method, where we in the first step sought for candidate predicates and in the second step the candidate predicates were used to induce the fuzzy characteristic rules.

The candidate predicates were isolated from typical values by , simple cluster method and we showed how a confidence threshold could be us(to reduce the search space for the induction of fuzzy characteristic rules.

The discovered fuzzy characteristic rules defined different \ ews on a sub-population and it was possible to isolate outliers as a fuzzy sub t. Further we showed how to compress the information in a subpopulation anc represent it as generalized table.

References

1. Han, J.: Mining Knowledge at Multiple Concept Levels. CIKM'94, Baltimore, Maryland, USA (1995) 19–24
2. Kamber, M. and Shinghal, R.: Evaluating the Interestingness of Characteristic Rules. Second Internationale Conference on Knowledge Discovery & Data Mining, Portland, Oregon (1996) 263–266
3. Han, J. and Fu, Y.: Attribute-Oriented Induction in Data Mining. Advances in Knowledge Discovery, AAAI Press / The MIT Press (1996) 399–421
4. Rasmussen, D. and Yager, R. R.: Summary SQL - A fuzzy tool for data mining. Intelligent Data Analysis (an electronic journal), (1996)
5. Michalski, R. S.: A Theory and Methodology of Inductive Learning. Artificial Intelligence **20** (1983) 111-161
6. Yager, R. R.: A note on a fuzzy measure of typicality. International Journal of Intelligent Systems **12** (1997) 233–249
7. Rasmussen, D. and Yager, R. R.: SummarySQL-A flexible fuzzy query language. FQAS'96, Roskilde, Denmark (1996) 1–18
8. Zadeh, L. A.: A theory of approximate reasoning. Machine Intelligence, Vol. 9, Hayes, J., Michie, D., & Mikulich, L.I. (eds.), New York: Halstead Press (1979) 149–194