A Characterization of Region of Experience Based on Association Rules

Daniel Pop
Department of Computer Science
West University of Timişoara
Timişoara, România
e-mail: danielpop@info.uvt.ro

Abstract

Knowledge representation in classification systems is done through several forms – rules set, decision tables, decision trees – which are equivalent and which can be automatically transformed from one to another. This paper studies the construction of characterization of region of experience, i.e. the subset of the attribute space to which the real-world process and the expertise of human experts is confined. A new method to construct this characterization using association rules is proposed and the results are compared with previous works. The paper also bring some improvements of the brute-force algorithm.

1 Introduction

To represent the knowledge in classification systems, a number of forms are used, such as: rules set (production rules, association rules, rules with exceptions), decision tables, classification and regression trees, instance-based representations, and clusters. Each representation has its advantages and drawbacks.

It has been proved sometimes ago that the rules set, decision table and decision tree forms are equivalent [3]. The problem is that when an object (decision tree, decision table or rules set) is transformed to a new form, the new object can be very much larger than the original and much less compact. We'll designate this phenomenon as *inflation*.

Region of experience of a classification system is defined as the subset of the attribute space to which the real-world process and the expertise of human experts is confined. Its main scope is to limit the workspace of the system to a smaller and safe sub-region of the whole attribute space.[10]

The inflation phenomenon is a side effect of the construction of classification systems as total functions on large attribute spaces as discussed in [2]. Consequently, it seems reasonable to look for a cure by defining the systems as partial functions over the region of experience, instead of total functions on the whole attribute space. If one could get a characterization K of the region of experience associated with a process generating cases to be classified, then we could use K to prune the transformed decision objects as they are being created. A good estimate of K would not only control inflation, but would act as a filter detecting cases which are either spurious or perhaps legitimate but outside the experience of the domain experts.

In [2] the authors adapt the conversion theorems between the three knowledge representation forms so that they are consistent with K. For example, in Rule Set to Decision Table conversion, are kept only those disjuncts of the definition of an intermediate proposition which are consistent with K. In Decision Table to Decision Tree conversion, only copies of rows which are consistent with K are made (see [2] for details).

2 Region of Experience Characterization

The focus of this paper is on how to build a characterization K for the region of experience. A natural way to represent K is by a set of constraints stating that the values of certain variables are determined by the values of others. These sort of constraints are *partial functional dependencies (PFDs)*. PFD can be seen as rules of the following form:

if $antecedent$ **then** $consequent$.

We will call the set of constraints with one variable in antecedent K_1, and in general the set of constraints with n variables in antecedent K_n.

The variables are in general independent, but if the set of variables in the domain of the dependency take on a particular combination of values, then the value of the variable in the range of the dependency is fixed.

For example, in general the variables 'Vehicle Type' and 'Spare Wheel' are independent. However, if 'Vehicle Type'

takes the value bicycle, then 'Spare Wheel' takes the value 0. Actually, we can construct the following two PFDs for these attributes:

if VehicleType $\in \{bicycle, motorcycle\}$ **then** SpareWheel=0
if SpareWheel > 0 **then** VehicleType $= \{motorvehicle\}$

Another example: **if** Income level=High **then** Risk NOT Bad.

If the real-world process generates far less valid cases than the entire attribute space, it is clear that there is a set of constraints (PFDs) that limits the valid cases. This set can be built in two ways:

- manually, by human experts, as part of the process of building the classification object;
- automatically, during the "automated" process of building the classification object.

We will investigate further the automatic way. Before jumping into the details of two approaches – rough sets and association rules – let us outline the major issues that one has to focus on when designing an algorithm for estimating the set of PFDs:

- the size of the set of PFDs has to lay in acceptable boundaries and it also has to offer an appropriate superior bound for the region of experience;
- the costs (time, resources) of building the PFD set have to be kept in an acceptable range and they are related to the size of the PFDs' set;
- the set of PFDs has to be statistically reliable.

The problem is therefore not whether we can induce a set of PFDs, but whether we can induce a set of practical size which gives a sufficiently tight upper bound on the region of experience, whether the induction can be done at an acceptable cost, and whether the set induced is statistically reliable.

Remark 2.1 *A solution for the first two problems is to look for for PFDs with a small number (one or two) of elementary propositions in their antecedents. There are two reasons for this. The first one is that a single PFD with m elementary propositions in the antecedent and consequent taken together constrains the attribute space more the smaller m is. If all the variables are boolean and there are n variables, then a single constraint reduces the attribute space by a factor of 2^{m-k+1}. The second reason for wanting PFDs with few propositions in their antecedent is that the algorithms for computing PFDs are exponential in the number of propositions in the antecedent. Small PFDs are therefore both stronger and practical to compute.*

Remark 2.2 *To induce the statistically reliable PFDs, it seems reasonable (in the absence of a good theory of the statistics of PFDs) to look for PFDs which have a large number of positive examples. This prevents rare cases from contributing possibly spurious PFDs.*

Keeping these two remarks in mind, let us proceed to search for suitable algorithms for inducing an appropriate characterization of the region of experience.

2.1 Rough Sets

The first trial was proposed by Colomb in [2] and is based on *rough sets* theory [1]. They have estimated the PFDs for Garvan ES1 from the set of 9805 cases (of which 3856 are distinct), using the RSL (Rough Sets Library) [5] developed by M. Gawrys and J. Sienkiewicz at Institute of Compter Science, Warsaw University of Technology, Warsaw, Poland. After applying the algorithm to the set of 3856 distinct cases (duplicated cases did not contribute to support), with a support level of 10 examples (with no counterexamples), they obtained 332 PFDs with a single elementary proposition in their antecedent and 5858 such PFDs with 2 elementary propositions, sets called Garvan K_1, respectivelly Garvan K_2. After removing the transitive redundancies (of form: if $a \rightarrow b$ and $b \rightarrow c$ are both in K, then the redundant $a \rightarrow c$ is also) the Garvan K_1 was reduced to 296 conjuncts.

2.2 Association Rules

In this paper we propose an approach inspired from data mining field, more specifically the *association rules*. An association rule is a set of items that co-occur frequently within a data set. These rules are implicative tendencies of the form $X \rightarrow Y$ where X and Y are conjunctions of database items (boolean variables). Such a rule means that most of the records which verify X in the database verify Y too. For instance, in market basket analysis where this concept is widely used to model the customer transactions, an association rule $\{pizza, crisps\} \rightarrow beer$ means that if a customer buys a pizza and crisps then he/she most probably buys beer too. The discovery of association rules is done in two steps: 1) discover all frequent itemsets with a given support and 2) generate from these itemsets all the rules with a given confidence factor. A survey and more details about association rules discovery and most popular algorithms can be found in [11, 7].

[1] A rough set is a formal approximation of a crisp set in terms of a pair of sets which give the lower and the upper approximation of the original set. The lower and upper approximation sets themselves are crisp sets in the standard version of rough set theory (Pawlak [9]), but in other variations, the approximating sets may be fuzzy sets as well. For more details, the reader is referred to Greco et al. [6]

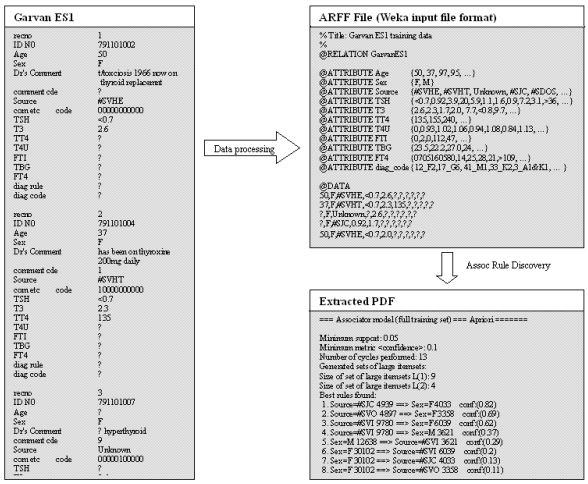

Figure 1. PFD Extraction Using Association Rules

In order to adapt the association rules discovery algorithms al terminology to automatic induction of PFDs, we have to make some considerations:

- The items are represented by the elementary propositions p_{ij}: $A_i = v_{ij}$, where A_i is an attribute and v_{ij} is a value from $dom(A_i)$ that appears in the training set.
- The transactions are represented by all the cases of the training set.
- A PFD is an association rule with a confidence factor of 100% and with a support higher enough to prevent rare cases from contributing possibly spurious PFDs.
- Recalling remark 2.1, we will search for frequent itemsets with maximum 3 items so that they will generate association rules with maximum 2 elementary propositions in their antecedent.

2.2.1 Experimental Results and Discussion

In order to compare the results with the ones of R. M. Colomb for rough sets [2] we considered the same expert system, Garvan ES1. The training set for this expert system is publicly available from University of New South Wales, Sydney, Australia [4]. The system should provide clinical interpretations for reports of thyroid hormones measurements in blood samples. The training set has 17 attributes and 43472 cases. The process of extraction of PFDs out of this data set is depicted in Figure 1.

The central step of the process is the association rules discovery algorithm. As the size of the data set is small in terms of data mining problems and we were not particularly interested in best time behavior, we selected a well-known and easy to configure algorithm: Apriori (Agrawal et al. 1994 [1]). The selected implementation is the one provided in Waikato Environment for Knowledge Analysis (WEKA)[2] [8, 11].

minsup	minconf	L_1	L_2	L_3	Rules#
0.1	0.1	5	1		2
0.05	0.1	9	4		8
0.02	0.1	40	25		32
0.01	0.1	151	93	3	111
0.00023	1.0	1201	14930	12053	775

Table 1. Apriori Results

WEKA's implementation of Apriori algorithm is expecting the input data set in the ARFF (Attribute-Relation File Format) format, the common format for all WEKA's algorithms. Therefore, a data processing step is necessary here and it is composed of two tasks: data cleaning and data preparation. During data cleaning stage, six of the original attributes of the training set were removed because they were providing only descriptive information, irrelevant for association rules discovery process. The data preparation step outputed the original Garvan ES1 cases in ARFF format. Therefore, the input data set for Apriori is having 11 attributes and 43472 examples/cases.

Figure 1 illustrates one original Garvan case, a snippet of ARFF file obtained after data processing step and the extracted PFDs (association rules).

The Apriori algorithm was ran with a set of different values for its configuration parameters: $minsup$ and $minconf$. The first parameter represents the minimum support for derived rules, i.e. the percentage of cases in which the association rules is present. The second parameter ($minconf$) represents the minimum confidence factor the rule has to pass. The results are displayed in Table 1. The number of frequent i-itemsets is displayed in the column L_i, $i = 1, 2, 3$, whereas the column Rules# holds the number of discovered association rules.

For the first run, with the support set at 0.1 (i.e. 10% of training cases) we get a single frequent 2-itemset ($\{Source = \#SVI, Sex = F\}$) that generated two association rules with confidence equals to 62%, respectively 20%. Out of the 8 rules extracted at the second run ($minsup = 5\%$), only one has its confidence factor greater than 70%. Lowering the support to 1% we get 3 frequent itemsets with three items that yield rules with two conditions in antecedent.

[2]WEKA is a portable, extensible, open-source platform developed in Java for data mining tasks support. It provides implementation for a broad spectrum of algorithms for classification and regression, association rules discovery, clustering and other machine learning tasks.

In order to compare our results with the one obtained using rough sets by Colomb [2], we lowered the support to minimum 10 cases, i.e. aprox. 0.00023, and we set the confidence factor to 100% (last row of the Table 1). The distribution of the 775 discovered association rules is: $Card(K_1) = 22$, $Card(K_2) = 591$, $Card(K_3) = 149$ and $Card(K_4) = 13$. In this step we also obtained itemsets with more than three items: $Card(L_4) = 947$ and $Card(L_5) = 13$. Comparing this characterization with the one provided using rough sets theory (where $Card(K_1) = 296$, $Card(K_2) = 5858$), we can conclude that the characterization generated using association rules is much more compact than the one built using rough sets.

2.3 Brute-force Generation of K

In this section we will present an extension of one simple algorithm to compute K_1, proposed for the first time in [2]. Let D be a training set composed of cases.

1. Construct an above the diagonal triangular array A of integers, whose rows and columns correspond to $v \in dom(A_i), i = 1 \ldots m$. The size of matrix A is $d \times d$, where $d = Card(dom(A_1)) + \ldots + Card(dom(A_m))$. Initialize this matrix to 0.

2. For each $c \in D$, increment all the cells in A which correspond to pairs of values of variables in c. In the end, each cell of A will contain the number of instances the associated variable values co-occur in the training set D.

3. Each attribute value $A_i = v$ selects a set of cells of A corresponding to all other variable values. If all but one of the cells associated with a given value v' are zero, i.e.

$$\exists! j \wedge \exists! v' \in dom(A_j) \mid A_j = v' \, and \, A_i = v,$$

then the rule $A_i = v \rightarrow A_j = v'$ is a PFD.

4. Each attribute value $A_i = v$ selects a set of cells of A corresponding to all other variable values. If all but one of the cells associated with a given value v' are non-zero, i.e.

$$\exists! j \wedge \exists! v' \in dom(A_j) \mid A_j \neq v' \, and \, A_i = v,$$

then the rule $A_i = v \rightarrow A_j \neq v'$ is a negative PFD.

By extending the matrix size to three dimensions, we can compute K_2 using the same algorithm.

#	*Debt*	*Income*	*Empl. Form*	*Risk Level*
1	High	High	Independent	High
2	High	High	Employee	High
3	High	Low	Employee	High
4	Low	Low	Employee	Low
5	Low	Low	Independent	High
6	Low	High	Independent	High
7	Low	High	Employee	Low
8	Medium	High	Independent	High
9	Medium	High	Employee	Low
10	Medium	Low	Employee	Low

Table 2. Date de antrenament 'si setul de reguli de produc'tie corespunz'ator

2.4 An Example

Table 2 depicts a dataset composed of 10 cases, with three predicting attributes (Debt Level, Income Level and Employment Form) and one predicted/target attribute (Risk Level). We'll illustrate the PFD extraction using the association rules discovery and brute-force methods presented in previous sections.

2.4.1 Extracting PFDs Using Association Rules Discovery

Each row of the Table 2 is considered a transaction of the database, each item being defined by the elementary proposition $A_i = v_{ij}$, i.e. the set of items I has nine items:

I = {Debt = High, Debt = Medium, Debt = Low,
Income = Low, Income = High,
Empl. Form = Independent, Empl. Form = Employee,
Risk = Low, Risk = High}.

In this example, we'll consider $minsup = 30\%$ and $minconf = 100\%$.

The first step of extracting the PFDs, i.e. association rules with $minsup$ and $minconf$ is to generate the frequent itemsets. All the items are frequent, therefore $F_1 = I$. In order to determine F_2, we should analyze the support of all items pairs. A significative pruning can be done knowing that two items obtained from the same attribute can not form together an itemset. All the frequent 2-itemsets (i.e. support greater of equal to $minsup$) are listed below (the figure between square brackets indicates the itemset's support):

F_2 = {(Debt = High, Risk = High [s=0.3]),
(Income = Low, Empl. Form = Employee [s=0.3]),
(Empl. Form = Independent, Risk = High [s=0.4]),
(Empl. Form = Employee, Risk = Low [s=0.4]),

		Debt			*Income*		*Empl.*		*Risk*	
		Low	Med	High	Low	High	Ind.	Emp.	Low	High
Debt	Low	-	-	-	2	2	2	2	2	2
	Med		-	-	1	2	1	2	2	1
	High			-	1	2	1	2		**3**
Income	Low				-	-	1	3	2	2
	High					-	3	3	2	4
Emp.	Ind.						-	-		**4**
	Emp.							-	**4**	2
Risk	Low								-	-
	High									-

Table 3. Example of matrix A

(Income = High, Empl. Form = Employee [s=0.3]),
(Income = High, Empl. Form = Independent [s=0.3]),
(Income = High, Risk = High [s=0.4])}

Finally, we also get one frequent 3-itemset F_3 = {(Income = High, Empl. Form = Independent, Risk = High [s=0.3])}.

The second step, after the frequent itemsets were built, is to generate the association rules with confidence factor greater or equal to $minconf$. The following association rules with confidence factor equal to 100% are generated from the F_2 and F_3 sets:

if Debt=High **then** Risk=High [s=0.3]

if Empl. Form=Independent **then** Risk=High [s=0.4]

if Risk=Low **then** Empl. Form=Employee [s=0.4]

if Income=High and Empl. Form=Independent **then** Risk=High [s=0.3]

2.4.2 Extracting PFDs Using Brute-Force Algorithm

In this section, we will apply the brute force algorithm to generated the K_1 for the training dataset illustrated in Figure 2. Table 3[3] represents the matrix A created by the brute-force algorithm. One can note that if $Debt = High$, then the attribute $Risk$ can be only $High$ and this holds for 3 cases of the dataset. Therefore, the following PFD can be stated:

if Debt=High **then** Risk=High [s=0.3][4]

Similarly, by looking at the highlighted cells in the table 3 we can construct the following PFDs:

if Empl. Form=Independent **then** Risk=High [s=0.4]

if Risk=Low **then** Empl. Form=Employee [s=0.4]

The negative PFDs are the following two:

if Debt=High **then** Risk $\neq$ Low [s=0.3]

if Empl. Form=Independent **then** Risk $\neq$ Low [s=0.4]

We shall remark that the characterization of K_1 obtained by brute-force is identical with the one obtained using association rules discovery.

3 Conclusions and Future Work

This paper addressed the problem of knowledge representation and possible translations from one representation to another. It has been proved that using the region of experience in translation process a lot of inconsistent and spurious cases are filtered, resulting in much more compact objects. Our experiments showed that the characterization of region of experience using association rules is much more compact than the one offered by rough sets. Anyway, the influence of the compactness on the transformation algorithm, from rules set representation of Garvan ES1 to decision table for example, has to be further investigated.

We also presented a brute-force algorithm for K_1 construction and illustrate both methods (association rules and brute-force algorithm) on an example.

Acknowledgment

This work was partially supported by the Romanian Government CNCSIS grant no 1385/2005 (MindSoft).

Annex

The Table 4 include the association rules (PFDs) extracted at a support of 70% from the Garvan ES1 training

[3]The cells marked with '"-'" denote cells where no values are possibles; for the sake of clarity, the bottom part of the matrix was left empty; don't forget that A is a symmetric matrix.

[4]s represents the support of the rule, i.e. the frequency of the rule in the dataset.

set at different support levels. It complements the Table 1. The rules marked with star symbol (*) are rules with two elementary propositions in their antecedent.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. of the 20th Conference in VLDB*, 1994.

[2] R. Colomb. Representation of propositional expert systems as partial functions. *Artificial Intelligence*, 109:187–209, 1999.

[3] R. Colomb and C. Chung. Strategies for building propositional expert systems. *International Journal of Intelligent Systems*, 10:295–328, 1995.

[4] P. Compton. Garvan-es1 dataset. ftp://ftp.cse.unsw.edu.au/pub/users/compton/43472.txt. [Online; accessed 3-May-2006].

[5] M. Gawrys and J. Sienkiewicz. Rough set library (rsl). http://rsds.wsiz.rzeszow.pl.

[6] S. Greco, B. Matarazzo, and R. Slowinski. Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research*, (129), 2001.

[7] M. Kantardzic. *Data Mining. Concepts, Models, Methods and Algoritms*. John Wiley and Sons Inc, 2003.

[8] U. of Waikato. Weka framework. http://www.cs.waikato.ac.nz/ml/weka/. [Online; accessed 3-May-2006].

[9] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning About Knowledge*. Kluwer Academic Publishers, 1991.

[10] D. Pop and V. Negru. Inter–translability of representation forms in classification. In *Proc. of the International Symposium SINTES*, volume 10, pages 120–123, 2000.

[11] I. H. Witten and E. Franck. *Data Mining. Practical Machine Learning Tools and Techniques. 2nd Edition*. Elsevier, 2005.

minsup=0.05 (5%)
$Source = \#SJC \rightarrow Sex = F,\ conf = 0.82$
minsup=0.02 (2%)
$Source = \#STMW \rightarrow Sex = F,\ conf = 0.93$
$Source = \#SJC \rightarrow Sex = F,\ conf = 0.82$
$Source = \#SVHT \rightarrow Sex = F,\ conf = 0.77$
$Source = \#SVHE \rightarrow Sex = F,\ conf = 0.77$
$diagcode = 32K1 \rightarrow Source = \#SVI,\ conf = 0.75$
$Source = DR \rightarrow Sex = F,\ conf = 0.73$
minsup=0.01 (1%)
$Source = \#STMW \rightarrow Sex = F,\ conf = 0.93$
$Source = \#WLG \rightarrow Sex = F,\ conf = 0.83$
$diagcode = 36L1 \rightarrow Sex = F,\ conf = 0.83$
$Source = \#SJC \rightarrow Sex = F,\ conf = 0.82$
$diagcode = 28I1 \rightarrow Sex = F,\ conf = 0.8$
$Source = \#SRP \rightarrow Sex = F,\ conf = 0.79$
$Source = \#SVHT \rightarrow Sex = F,\ conf = 0.77$
$diagcode = 16G1 \rightarrow Sex = F,\ conf = 0.77$
$Source = \#SVHE \rightarrow Sex = F,\ conf = 0.77$
$TSH =< 0.1 \rightarrow Sex = F,\ conf = 0.76$
$TSH =< 0.8 \rightarrow Sex = F,\ conf = 0.76$
$(*)\ Sex = M \wedge diagcode = 32K1 \rightarrow Source = \#SVI,\ conf = 0.76$
$diagcode = 1A1 \rightarrow Sex = F,\ conf = 0.76$
$Source = \#SJF \rightarrow Sex = F,\ conf = 0.76$
$TSH =< 0.7 \rightarrow Sex = F,\ conf = 0.75$
$diagcode = 32K1 \rightarrow Source = \#SVI,\ conf = 0.75$
$(*)\ Sex = F \wedge diagcode = 32K1 \rightarrow Source = \#SVI,\ conf = 0.74$
$Age = 70 \rightarrow Sex = F,\ conf = 0.74$
$diagcode = 11F1 \rightarrow Sex = F,\ conf = 0.74$
$diagcode = 17G6 \rightarrow Sex = F,\ conf = 0.73$
$Source = DR \rightarrow Sex = F,\ conf = 0.73$
$TSH =< 1.0 \rightarrow Sex = F,\ conf = 0.73$
$TSH =< 0.3 \rightarrow Sex = F,\ conf = 0.72$
$Age = 66 \rightarrow Sex = F,\ conf = 0.72$
$TSH =< 0.4 \rightarrow Sex = F,\ conf = 0.71$
$TSH =< 0.5 \rightarrow Sex = F,\ conf = 0.71$
$TSH =< 0.6 \rightarrow Sex = F,\ conf = 0.71$
$Age = 61 \rightarrow Sex = F,\ conf = 0.7$

Table 4. Characterization of K for Garvan ES1 training set (only rules with confidence factor greater than 70% are included)