Intelligent Information Systems 2008
ISBN 978-83-60434-44-4, pages 405–414

Knowledge representation for composited knowledge bases

Agnieszka Nowak[1], Roman Simiński[1], and Alicja Wakulicz-Deja[1]

Institute of Computer Science, Silesian University, Katowice, Poland

Abstract

In this paper the problem of effective and proper knowledge representation methods is considered. The more data in given knowledge base the more difficult to store and analyse them. The most popular method of representation knowledge in decision support system is the rule's based method. We think that such techniques as *decision units* and *cluster analysis* could let for modularity the given knowledge base and help with managing such sources. In result of our research we saw that both method are very similar and they help to optimize the inference processes in such systems.

Keywords: composited knowledge bases, knowledge representation, clusters of rules, decision units

1 Influence of Knowledge Representation On Decision Support System's efficiency

Large knowledge bases are an important problem in decision systems. It is well known that the main problem of forward chaining is that it fires a lot of rules, that are unnecessary to fire, because they aren't the inference goal. A lot of fired rules forming a lot of new facts that are difficult to interpret them properly. That is why the optimization of the inference processes in rule based systems is very important in artificial intelligence area. Fortunately there are some methods to solve such problem. For example, we may reorganize the knowledge base from list of not related rules, to groups of similar rules (thanks to *cluster analysis* method) or *decision units*. Thanks to this it is possible to make the inference process (even for really large and composited knowledge bases) very efficient. Simplify, when we clustering rules, then in inference processes we search only small subset of rules (cluster), that the most similar to given facts or hipothesis. In case of using *decision units* concept, thanks to built such units, in *backward chaining* technique we make inference process only on proper decision unit (that with the given conclusion attribute). That is why we propose to change the structure of knowledge base to *cluster* or *decision unit* structure.

2 VSM – Vector Space Model as a method of representation of composited knowledge bases

In the context of information retrieval in such source of data as rule-based knowledge bases, a general definition of clustering is the following: given a large set of rules, automatically discover diverse subsets of rules that share a similar topic (attributes and values of such attributes). Input rules (objects) are first transformed into a mathematical model where each rule is described by certain features (attributes: conditional and decision). The most popular representation for text in text mining area is the *Vector Space Model*. In that model, objects (documents) are expressed as rows in a matrix, where columns represent unique terms (features) and the intersection of a column and a row indicates the importance of a given word to the document. If we use such method to represent rules we will say that in such a model, rules are similarly expressed as rows in a matrix, where columns represent unique attributes and the intersection of a column and a row is a value of given attribute. It is needed to construct for discovered groups of rules- a label, a symbolic description for each cluster, something to represent the information that makes rules inside a cluster similar to each other and that would convey this information to the user. Cluster labeling problems are often present in modern text and web mining applications with document browsing interfaces. We propose to use this representation technique in rule-based application, especially in *inference engine* process with rule browsing interfaces.

3 Why clustering rules in knowledge bases?

We think that cluster analysis brings useful technique to reorganize the knowledge base to hierarchical structure of rules. It will optimizes the inference processes cause we will search only the one chosen cluster, that with the highest similarity value (similarity to the given set of facts). The created structure we called *hierarchical* cause applied algorithm of agglomerative hierarchical clustering builts the tree (called *dendrogram*) which shows the hierarchy of rules. Because the created tree has all features of the binary tree, we can simplify notes that the time efficiency of searching such trees is $O(\log_2 n)$. Cluster analysis groups data objects based only on information found in the data that describes the objects and their relationships. The goal is that the objects within a group be similar (or related) to one another and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group and the greater the difference between groups, the better or more distinct the clustering. The reasons of clustering rules in large (compose) knowledge bases are following: for improving recall in search applications, for speeding up vector space retrieval, navigation and presentation of search results.

3.1 What is a cluster?

A *cluster* is a collection of data objects that *are similar to one another within the same cluster* and *are dissimilar to the objects in other clusters*. The goal of clustering is to find groups containing objects most homogeneous within these groups. There are two ways to represent clusters in a measurenment space: as an existing object in the cluster or as as a hypothetical point which is not an object in the cluster, called *centroid* or *cluster representative.* Sometimes the *medoid* is a representative vector of given cluster. Clusters are represented in the measurenment space in the same form as the objects they contain. Some common definitions are collected from the clustering literature and given below (Dubes and Jain, 1998; Aldenderfen and Blashfield, 1984):

- *A cluster is a set of entities which are alike, and entities from different clusters are not alike.*
- *A cluster is an aggregation of points in the space such that the distance between two points in the cluster is less than the distance between any point in the cluster and any point not in it.*
- *Clusters may be described as connected regions of a multidimensional space containing a relatively high density of points, separated from other such regions by a region containing a relatively low density of points.*

Good clustering is such a clustering that produces high quality clusters, which means that intra-cluster similarity is high and inter-cluster class similarity is low. There are some so called *quality factors*: similarity measure and its implementation, definition and representation of cluster chosen and clustering algorithm, which let us make our clustering the best possible.

3.2 VSM – based method of representation of rules

In the *centroid* based classifiaction algorithm, the rules are represented using the VSM (ang. *Vector Space Model*) method. In this model, each rule r is considered to be a vector in the attribute-space. In its simplest form, each rule is represented by the value of attribute vector $\overrightarrow{r}_V = (v_1^j, v_2^j, \ldots, v_n^j)$, where v_i^j is the j-th value of the i-th attribute in the rule r. Then we can represent the j-th value of i-th attribute (v_i^j) in rule r_k as:

$$v_i^j = \begin{cases} a^i & \text{if } a_i \in r_k; \\ 0 & \text{if } a_i \notin r_k. \end{cases} \tag{1}$$

where a^i represents the value of attribute a_i in rule r_k. Combining all rule vectors creates an attribute-rule matrix. Given a set R of rules and their corresponding vector representations, we define the centroid vector $\overrightarrow{c}$ to be:

$$\overrightarrow{c} = \frac{1}{|R|} \sum_{r \in R} \overrightarrow{r} \tag{2}$$

, which is nothing more than the vector obtained by averaging the values of the various attributes present in the rules of R (Kaufman and Rousseeuw, 1990). An example of such a matrix is shown in $Table$ 1. There are two different methods to represent a cluster in the given system:

- as a *centroid* (i.e. center of gravity, i.e. the mean of each feature, for the samples in the cluster),

Table 1: An Attribute-Rule matrix

a_i	a_1	a_2	a_3	$a_4 \dots a_n$
r_1	$v^j_{a_1}$	$v^j_{a_2}$	$v^j_{a_3}$	$\dots$
r_2	$v^j_{a_1}$	$v^j_{a_2}$	$v^j_{a_3}$	$\dots$
r_3	$v^j_{a_1}$	$v^j_{a_2}$	$v^j_{a_3}$	$\dots$
r_4	$v^j_{a_1}$	$v^j_{a_2}$	$v^j_{a_3}$	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
r_m	$\dots$	$\dots$	$\dots$	$\dots$

- graphically by using nodes in a clustering tree.

A *centroid* (central value) is a set of attributes and values of attributes that are statistically important to a cluster of rules. Such centroids could be used both to classify relevant rules and to identify salient sentences in a cluster. A *medoid* is some representative point from all of points clustered in given set. The main problem to store knowledge of composited knowlede base is the form of its representation. It seems that it is much easier to present that form graphically. To represent the given part of knowledge we can use the *descriptors*, often called *concept*. It is a vector with descriptors which are characteristic for a given cluster of rules. The less number of descriptor in given concept the more characteristic the cluster is, the better the groups are. Many descriptors resulting difficulty in seeing differences between clusters. It is better if we have a different concepts for our clusters, cause it makes the process of searching the clusters faster.

4 The Hierarchical Structure Of Knowledge Base

The hierarchy is a very simple and natural form of presentation the real structure and relationships between data in large data sets. Instead of one long list of all rules in knowledge base, we prefer to build composited knowledge bases as a set of groups of similar rules.

4.1 The knowledge base structure

The formal definition should consider system as some sixth elements object: $S_{HC} =< X, A, V, dec, F_{sim}, Tree >$, where:

$X = \{x_1, .., x_n\}$ – set of rules with Horn's forms,

$A = \{a_1, .., a_m\}$ – where $A = C \cup D$ (condition and decision attributes),

$V_i = \cup_{a_i \in A} v_i$ – the set of values of a_i attribute,

$x_i \in V_i$, for $1 \leq i \leq n$,

$X = V_1 \times V_2 \times \ldots \times V_n$,

$dec : X \to V_{dec}$, where $V_{dec} = \{d_1, .., d_m\}$,

$F_{sim} : X \times X \to R|[0..1]$,

$Tree = \{w_1, .., w_{2n-1}\} = \bigcup_{i=1}^{2n-1} w_i$ (or $Tree = \{w_1, .., w_k\} = \bigcup_{i=1}^{k} w_i$ where $k \leq 2n - 1$ if we apply the $mAHC$ algorithm),

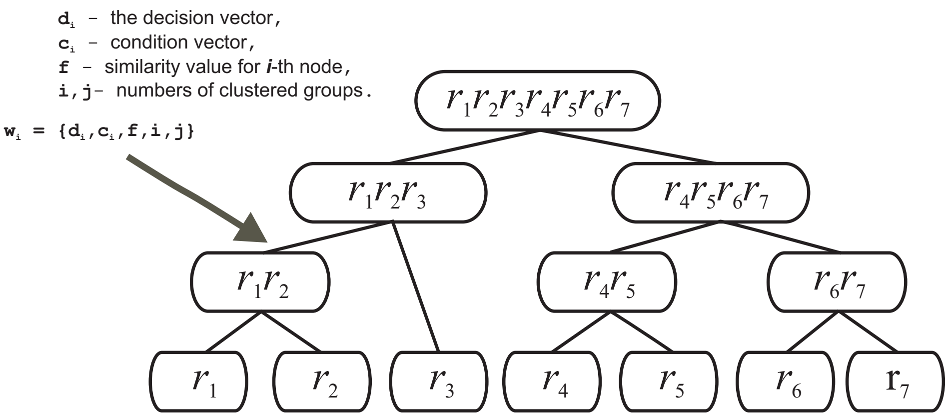

FIGURE 1: Clusters building tree

$w_i = \{d_i, c_i, f, i, j\}$, where $f = F_{sim}(x_i, x_j) \to [0..1]$,
$i, j \in (1, 2, .., 2n-1), d_i \in V_{dec}, c_i = X$.

In that system, besides the decision values *dec* given for each rule, there is also the similarity function value F_{sim}, which shows how high is the similarity of clustered rules (group of rules). $Tree$ is the set of all nodes in created tree structure. In this tree, each node w_i is defined by five elements: d_i – decision vector, c_i – condition vector (both represented with VSM model form), f_i – similarity function value for created i-th node, and i and j are numbers of clustered groups (Nowak and Wakulicz-Deja, 2005, 2006). The vectors d_i and c_i are vectors represented using VSM model presented in section 3.2.

5 Decision units as tool for decomposition

Decision units (Simiński, Wakulicz-Deja, 2000) oryginally came into existence as a tool facilitating the realisation of global and local rule knowledge base verification. This approach is devoted to knowledge bases, in which there are rules that probably create deep inference path. Decision units conception assumes, that we use backward chaining inference. In this way, decision units allow us to divide rules into subgroups in different way than — described above — rules clusters.

We shall introduce conception of decision units determined on a rule base containing the *Horn* clause rules, where literals are coded using attribute-value pairs. We assume a backward inference. A decision unit U is defined as a triple:

$$U = (I, O, R)$$

where:

Set of input entries: $I = \{(attr_i, val_{ij}) : \exists r \in R(attr_i, val_{ij}) \in antec(r)\}$,
Set of output entries: $O = \{(attr_i, val_{ij}) \forall r \in R : attr_i = conclAttr(r)\}$,

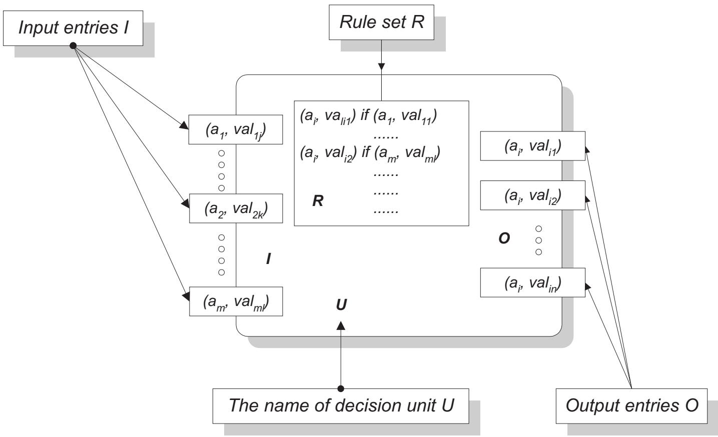

FIGURE 2: The structure of decision unit

Set of rules : $R = \{r : \forall i \neq j, r_i, r_j \in R : conclAttr(r_i) = conclAttr(r_j)\}$.

These sets are defined as follows:

$I = \{(attr_i, val_{ij}) : \exists r \in R(attr_i, val_{ij}) \in antec(r)\}$,
$O = \{(attr_i, val_{ij}) \forall r \in R : attr_i = conclAttr(r)\}$,
$R = \{r : \forall i \neq j, r_i, r_j \in R : conclAttr(r_i) = conclAttr(r_j)\}$.

Two functions are defined on a rule r : $conclAttr(r)$ returns attribute from conclusion of rule r, $antec(r)$ is a set of conditions of rule r. As it can be seen, decision unit U contains the set of rules R, each rule $r \in R$ contains the same attribute in the literal appearing in the conclusion part. All rules grouped within a decision unit take part in an inference process confirming the aim described by attribute, which appears in the conditional part of each rule. The process given above is often considered to be a part of decision system, thus it is called – a *decision unit*. All pairs *(attribute, value)* appearing in the conditional part of each rule are called decision unit input entries, while all pairs (attribute, value) appearing in the conclusion part of each set rule R are called decision unit *output entries*.

Decision unit net can be considered as a global model of dependences occurring in knowledge base (Simiński, Wakulicz-Deja, 2003). It allows not only to carry the verification actions but to retrieve the model, hidden in, potentially numerous, set of rules. Decision units' conception has been used in kbBuilder system — the application that assists building and verification of rule knowledge bases.

The approach provided by decision units are insufficient for the specifics of modeling processes occurring when drawing a conclusion during inference (Simiński, Wakulicz-Deja, 2004). This weakness of decision units forced us to look for methods of extending the features of decision units to the possibility of modeling the dynamics of inference processes. It was somehow natural, that our concern was

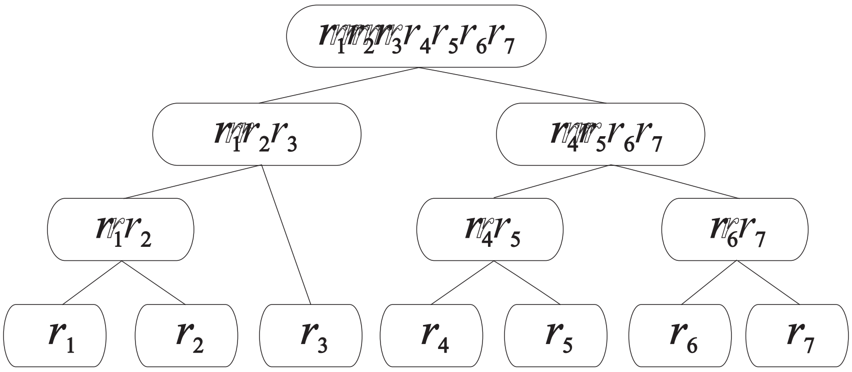

FIGURE 3: Result clusters of the rules for an example knowledge base

addressed to Petri nets — well known tool for modelling the dynamical processes. Petri nets allows modelling the processes of inference, which takes place by propagation of tokens. Launching of transitions corresponds to launching the rules. There exists the similarity between knowledge base in the form of Petri net and in the form of decision units net (Simiński, Wakulicz-Deja, 2006). In particular cases, these are the same representations which differ in general conception and notation. In more general case the decision units net is the generalized representation of rule base, thus it is some kind of Petri net's abstraction. Connecting both approaches shall allow to extend the idea of modelling the rule knowledge base in the form of decision units with clear method of representation of inference process dynamics.

6 The examples of the presented representation methods

For a given set of rules:

$r_1 : a_1{=}1 \rightarrow d_1 = 1$
$r_2 : a_4 = 1 \rightarrow d_1 = 1$
$r_3 : a_2 = 1 \wedge a_4 = 1 \rightarrow d_1 = 2$
$r_4 : a_1 = 2 \wedge a_3 = 1 \rightarrow d_2 = 1$
$r_5 : a_1 = 2 \wedge a_3 = 1 \wedge a_4 = 1 \rightarrow d_2 = 2$
$r_6 : a_1 = 2 \wedge a_3 = 2 \wedge a_4 = 4 \rightarrow d_3 = 1$
$r_7 : a_1 = 2 \wedge a_3 = 2 \wedge a_4 = 5 \rightarrow d_3 = 2$

, their clustering process is presented as a dendrogram in *Figure 2*. This same knowledge base in the form of decision units net we present at *Figure 3*. In our example we have three decision units because knowledge base contains three decision attributes: d_1, d_2, d_3. When we use decision units, the number of decision attributes, imply the number of decision units. The number of rules cluster depends on stop condition of clustering algorithm. That algorithm compares only conditional parts of the rules. In our example clusters of rules and rules in decision units are very similar. This is a special case of course. In the real word knowledge

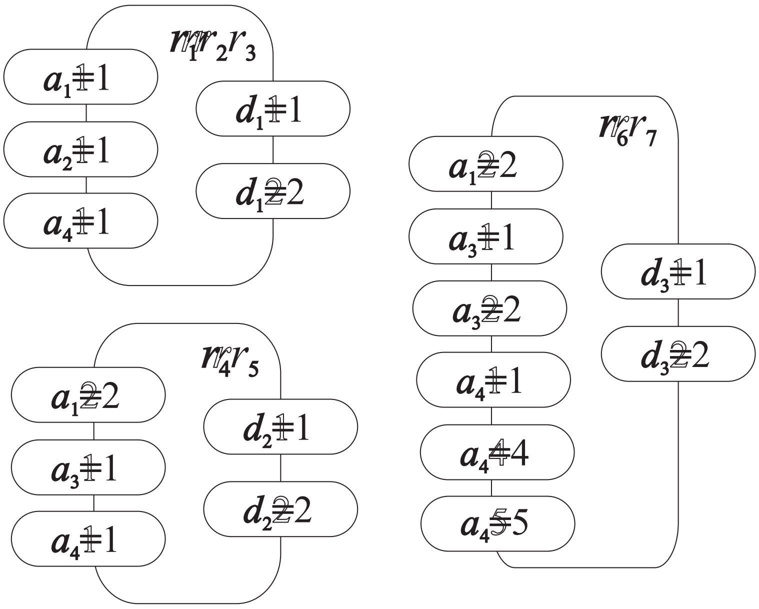

FIGURE 4: Decision units net for an example knowledge base

base each cluster may contain rules with different decision attributes. Thus we can obtain local decision units within each cluster and those decision units allow us to divide rules in cluster into hierarchical organized subgroups. In this way we can join two different approaches: clustering based on conditional parts of rules and decision units based on decision discrimination. Such dual representation allow us to perform two way optimization of inference processes. When we need to do forward reasoning, we use rules clusters to find proper set of rules, which conditionals parts match to fact base. When we use backward reasoning, we use decision units to obtain short search space. If we use mixed inference regime we can switch inference algorithm from backward to forward version and two previously described modular representations can be used.

7 Experiments. Tests given on real data.

In the experiments taken on different set of data we were analyzing quite a composited knowledge bases from two knowledge sources: "media.kb" and "credit.kb". Using rules clustering before inference process in given knowledge bases, instead of all set of rules we searching really small percent of the data set: in large sets it was 2.2% (data: *271* rules, 12 searched nodes), 4.53% (data: *177* rules, 16 searched nodes), for small sets the profits weren't much smaller: 40% (data:*13* rules, 10 searched nodes) and 32.2% (data:*16* rules, 10 searched nodes). It lets us implicited the conclusion that in large knowledge bases (with real data) with the multidimensional vectors of clusters of rules we obtained structures ensuring the high efficiency of the inference process. Optimalization of backward chaining inference using decision units, concerns on choosing proper decision unit, with output entries match to the goal of inference. We don't need to search all rules, only subset of rules inside decision unit must be checked for confirmation. This proces repeats recursively if we need to confirm subgoals, and again, we can consider only subset of rules in particular decision unit.

8 Summary

In our opinion modularization of the rule knowledge base allow us to optimize the efficiency of inference process. Using modular representation we can limit the number of rules to process during the inference. Thanks to properties of the cluster and the decision units we can perform different inference algorithm optimizations, depending on user requirements. On this stage of our work we can only present the general conception of modular rule base organization. We can't formally proof that our conception really will cause growth of efficiency. But in our opinion hierarchical organization of rule knowledge base allow us to decrease the number of rules necessary to process during inference, thus we hope that global inference efficiency will grow. On this stage of our reaserch, decision units (with Petri nets extensions) and rules clusters are paraller tools for rule base decomposition rather than a one coherent approach. Therefore we have two methods of rule base decomposition — into the rules clusters if we want to perform forward chaning inference and into the decision units, if we want to do backward chanining inference. The main goal of our future work is to create coherent conception of modularization of large rule bases. This conception shall join to main subgoals: optimalization of forward and backward chaining inference process and practical approach for rule base modelling and verification. In our opinion, two methods of rule base decomposition described in this work, allow as to obtain our goals. It is very important, that exists software tools, dedicated for rules clustering and decision units approach. Practical tests allow us to say, that we need specialized software tools when we work with large, composited rule bases. We expect, that our mixed approach is base for creating such software tools.

9 Acknowledgements

The research has been partially supported by the project "Decision support — new generation systems" of Innovative Economy Operational Programme 2007-2013 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland.

References

L. Kaufman and P.J. Rousseeuw (1990), Finding Groups in Data: An Introduction to Cluster Analysis, in *John Wiley Sons, New York*.

A. Nowak and A. Wakulicz-Deja (2005), The concept of the hierarchical clustering algorithms for rules based systems, in *Intelligent Information Systems 2005 – New Trends in Intelligent Information Processing and Web Mining*, Gdańsk, Poland.

A. Nowak and A. Wakulicz-Deja (2006), The inference processes on clustered rules, in *Springer-Verlag Berlin Heidelberg – Advances in Soft Computing 5*, pp 403-411.

A. Nowak and A. Wakulicz-Deja and R. Simiński (2006), Towards modular representation of knowledge base, in *Springer-Verlag Berlin Heidelberg – Advances in Soft Computing 5*, pp 421-428.

R.C. Dubes and A.K. Jain (1998), *Algorithms for clustering data*, Prentice Hall.

M.S. Aldenderfen and R.K. Blashfield (1984), *Cluster analysis*, Sage Publications London, England.

R. Simiński and A. Wakulicz-Deja (2003), *Decision units as a tool for rule base modeling and verification*, in *Advances in Soft Computing*, Physica–Verlag, Springer Verlag Company.

R. Simiński and A. Wakulicz-Deja (2004), *Application of Decision Units in Knowledge Engineering*, in *Lecture Notes in Artificial Intelligence*, Springer-Verlag.

R. Simiński and A. Wakulicz-Deja (2000), *Verification of Rule Knowledge Bases Using Decision Units*, in *Advances in Soft Computing*, Intelligent Information Systems, Physica Verlag, Springer Verlag Company.

R. Simiński and A. Wakulicz-Deja (2006), *Extending decision units conception using Petri nets*, in *Advances in Soft Computing*, Intelligent Information Systems, Physica Verlag, Springer Verlag Company.