

The inference processes on composited knowledge bases

Agnieszka Nowak and Alicja Wakulicz-Deja

Institute of Computer Science, Silesian University, Katowice, Poland

Abstract

In this paper the problem of long and not quite efficient inference process is considered. There is some problem with large set of data, e.g. set of rules, which causes long time of inference process. The paper presents the idea of hierarchical structure of knowledge base (using *agglomerative hierarchical clustering-AHC* algorithm and *modified AHC – mAHC*). In composited knowledge bases, groups of similar rules are created on each level of hierarchy. Then, the rule interpreter searches the tree and finds the most similar cluster. The inference process is done on a given cluster. It lets to make the all process of decision making more effective (not only because of the time, but also because of firing the minimal number of rules).

Keywords: composited knowledge bases, inference processes, cluster analysis

1 The Optimization of the Decision Support System

Decision support systems using the inference processes to get new facts from rules and initial facts. In this paper we consider only one method of inference process: *forward chaining*, but in our research *backward chaining* procedure is also included. 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 goal of inference. 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. We propose such an optimization by changing the structure of knowledge base. It consists of applying *cluster analysis* method to build the clusters of similar rules. Then in inference process, 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 builds the tree (called *dendrogram*) which shows the hierarchy of rules. Because the created tree is a kind of binary tree, we can simplify notes that the time efficiency of searching such trees is $O(\log_2 n)$. It means that the rule interpreter in given decision support system doesn't have to search the whole knowledge base in the time efficiency of $O(n)$ (like it is for classical knowledge bases). It makes the inference processes faster and more effective not only because of the time but also because of exploring the exact knowledge. Simplify, we think that in such systems, when we find the

most similar cluster and then we make inference process only on such small part of knowledge base, only the most relevant rules are fired and only the most important new facts are created and given to the user.

2 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. Such groups we called *clusters*. The similarity is checked by compare conditional part of rules. Those rules, which have the same or very similar knowledge inside, are clustered. The metrics used in clustering process is very important in the process of searching such structure. We analyzed different metrics of similarity: *cosine*, *ovlap* and *Gower's* and the metrics of *euclidean distance*. The results showed that only the *Gower's* measure is effective for such different types of data that we have in composited knowledge bases.

Assume that objects (rules) are represent as specific structure with two vectors (one for condition part and the second one for decision) in n -dimensional space \mathbb{R}^n for each rule: $\vec{x} = [x_1, x_2, \dots, x_n]$ where x_i is the value of i -th attribute for given rule x (Everitt, 1993).

2.1 The knowledge base structure

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

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

$A = \{a_1, \dots, a_m\}$ – 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 $1 \leq i \leq n$,

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

$dec : X \rightarrow V_{dec}$, where $V_{dec} = \{d_1, \dots, d_m\}$,

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

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

$w_i = \{d_i, c_i, f, i, j\}$, where $f = F_{sim}(x_i, x_j) \rightarrow [0..1]$,

$i, j \in (1, 2, \dots, 2n - 1)$, $d_i \in V_{dec}$, $c_i = X$.

In such a system, besides the decision values dec given for each rule, there is also the similarity function value F_{sim} , which show how high was the similarity of clustered rules (group of rules). $Tree$ is the set of all nodes in created tree structure. In this tree, each node is defined by five elements object: d_i – decision vector, c_i -condition vector, 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).

2.2 Source of Data

Each variable of each rule's feature vector is normalised and standardized to the vector of quantitative values and presented in n -dimensional space. For each step of the algorithm, a new group of rules perceived as similar (according to Gower's metrics) is created. For each newly created aggregate, a representative of the group (a centroid) is created, which is a vector of mean values for each group.

3 Hierarchical Clustering of Rules

Agglomerative algorithm starts with each object being a separate itself, and successively merge groups according to a distance measure. The clustering may stop when all objects are in a single group (classical *AHC*) or at any other point the user wants (modified *AHC*, so called *mAHC*) (Nowak and Wakulicz-Deja, 2006; Koronacki and Ćwik, 2005; Everitt, 1993).

The results of hierarchical methods are usually summarized in an agglomeration schedule. It can be visualized by a so called *dendrogram*. Finally all objects are combined to one cluster or if it was specified the clustering has been finished when some pre-defined threshold is reached. In this case we use some kind of modified hierarchical clustering algorithm (called *modified Agglomerative Hierarchical Algorithm*). The short idea of it we present in subsection 3.1. The stopping criterion (used in a given algorithm) was discussed in more details in Theodoridis and Koutroumbas 1999.

3.1 Modified Agglomerative Hierarchical Algorithm

<p>INPUT:A set O of n objects and a matrix of similarities between the objects.</p> <p>OUTPUT: Clusterings C_0, C_1, \dots, C_{k-1} of the input set $O; k < n$ $C_0 =$ the trivial clustering of n objects in the set input set O;</p> <p>while ($sim(c_i, c_j) > STOP^*$) do</p> <ul style="list-style-type: none"> find $c_i, c_j \in C_{k-1}$ where similarity $s(c_i, c_j)$ is maximal; $C_k = (C_{k-1} \setminus \{c_i, c_j\}) \cup (c_i \cup c_j)$; calculate similarity $s(c_i, c_j) \forall c_i, c_j \in C_k$; <p>end</p>

* *STOP* is the stopping condition, $STOP = T$, where T is the value given by user or it is threshold criterion for continue the clustering [sec. 3.2].

3.2 What is the Best Moment to Stop Clustering with *mAHC* ?

The only problem at this stage seems to be the similarity threshold T that the groups should exceed so that their further merger would make sense. T is a value from the range $[0..1]$ (if we use similarity measure in clustering process). In Nowak and Siminski and Wakulicz-Deja 2006; Nowak and Wakulicz-Deja 2005 we proposed some forms for this threshold, but we didn't get satisfied results, that is why we started to looking forward. Now, we want to fit it to the most individual cases, to make sure that even then it will take a properly values. That is why

we use the *Theodoridis & Koutroumbas* measure (Theodoridis and Koutroumbas, 1999; Stapor, 2005). With this measure (equation (1)), the best moment to cut the tree is the moment when the condition presented below is true.

$$\forall_{G_i, G_j} D_{min}(G_i, G_j) > \max\{h(G_i, G_j)\} \quad (1)$$

where: $h(G_i)$ is the self similarity measure of group. It means, it is the similarity measure between vectors in given group. It can be the maximal distance between vectors in one group:

$$h(G) = \max\{d(x, y) | x, y \in G\} \quad (2)$$

Equation (2) represents the condition, which let us make sure that the clustering process should be stopped when the distance between some given pair of groups (clusters) is higher than the similarity in all of groups (Stapor, 2005). *Figure 1* shows two big clusters of rules (*Figure 2* shows three clusters) instead one finite cluster with all rules. The criterion function shows that the best moment to stop clustering is when we achieved those two groups: $\{r_1, r_2, r_3\}$ and $\{r_4, r_5, r_6, r_7\}$. With this partitions the rules in clusters are similar together in the same cluster, and dissimilar to rules from other group.

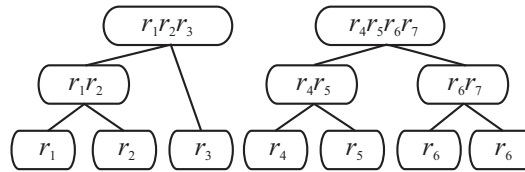


FIGURE 1: The dendrogram with stop criterion – 2 clusters.

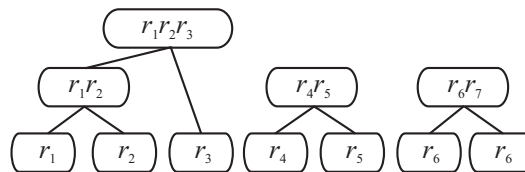


FIGURE 2: The dendrogram with stop criterion – 3 clusters.

4 Inference process on hierarchical knowledge base – forward chaining

4.1 Decision support systems with classical KB vs. hierarchical KB

In classical knowledge bases, the rule interpreter has to check each rule, one by one, and firing these which exactly match to given observations. It causes that the efficiency time of this algorithm takes in time $O(n)$, where n is the number of

clusters. In hierarchical knowledge bases, rule interpreter doesn't have to check each rule, one by one. Simplify, because now it is necessary to compute some similarity or distance values between given facts and created hierarchical structure and at each level choose this node with highest value. It causes that the efficiency time of this algorithm is minimalized to $O(\log n)$, where n is the number of clusters.

4.2 The Rules Retrieval Algorithm

There are various types of methods which are searching trees. For binary trees there is very effective method, which we called "*the best node in all tree*". It is very fast, often gives higher accuracy and always choose one element, that with higher similarity value. Other method we considered was "*the minimal value of coefficient*", which needs to specify the value of the minimal similarity between given node in the tree and searched data. It is often difficult to choose proper value. If it is too small it could increase the time of all processes, and if it is too high it is possible that we will not find the answer from the system (Nowak and Wakulicz-Deja, 2005, 2006; Nowak and Siminski and Wakulicz-Deja, 2006).

5 Experiments. Tests given on real data.

In the experiments taken on different set of data we were analyzing three cases:

- full clustering (*AHC*) and inference processes taken on clustered rules (**case a**),
- clustering with the *mAHC* algorithm with specified *stop criterion*, which is the case, when the similarity between groups is higher than similarity between objects inside groups (**case b**),
- clustering with the *mAHC* algorithm with given similarity *threshold* T_{min} and inference processes are taken on such structure (**case c**).

We present the results for two different sets of data: "media.kb" and "credit.kb". They are presented in the *Table 1*. *Figure 3* presents the plot of the efficiency of the rule interpreter in the case of number of rules in knowledge base. The results given in the table showing full clustering as the most effective if we look at the time of the clustering, but created tree often doesn't have high quality of the groups on higher levels of dendrogram. Instead of all knowledge base we have to search really small percent of the set: in large sets it was 2.2% (data: *case 6a*), 4.53% (data: *case 11a*), for small sets the profits were much smaller: 40% (data: *case 1a*) and 32.2% (data: *case 7a*). We noticed that *the best node* method is rather nonresistant for bad estimation of some similarities (cosine and overlap metrics). Those measures got quite big values for data, that weren't such similar at all. The algorithm always chooses the first maximal value in the list, and if the given node (rule) wasn't relevant the inference process taken on it finished with failure. That is the reason why we choose the clustering with the algorithm *mAHC*, which doesn't cluster rules till we get one finite cluster, but it stops clustering earlier, in the most proper time. It is based on one of two criteria: given *threshold* T or so called *stop criterion value* which checks whether in a given iteration all groups are similar

TABLE 1: Similarity measure: *Gower's* measure, *Single Linkage* method of centroid.

data	nA	nR	nS	nE	wzely	Treshold	rel	<i>D</i>	<i>Sens</i>	<i>%bw</i>
case 1a	12	13	12	25	10		1	1.0	1.0	40%
case 1b	12	13	9	17	13	0.97	1	1.0	1.0	76.4%
case 1c	12	13	4	22	10	0.95	1	1.0	1.0	45.4%
case 2a	12	28	27	55	6		1	1.0	1.0	10.9%
case 2b	12	28	21	35	23	0.97	1	1.0	1.0	65.7%
case 2c	12	28	4	52	6	0.9	1	1.0	1.0	11.5%
case 3a	12	55	54	109	12		1	1.0	1.0	11%
case 3b	12	55	38	72	40	0.97	1	1.0	1.0	55.5%
case 3c	12	55	4	106	14	0.9	1	1.0	1.0	13.2%
case 4a	12	73	72	145	12		1	1.0	1.0	8.27%
case 4b	12	73	52	94	54	0.98	1	1.0	1.0	57.4%
case 4c	12	73	4	142	14	0.9	1	1.0	1.0	9.8%
case 5a	12	135	134	269	14		1	1.0	1.0	5.2%
case 5b	12	135	110	160	114	0.98	1	1.0	1.0	71.25%
case 5c	12	135	5	265	17	0.9	1	1.0	1.0	6.4%
case 6a	12	271	270	541	12		1	1.0	1.0	2.2%
case 6b	12	271	235	307	237	0.99	1	1.0	1.0	77.1%
case 6c	12	271	5	537	15	0.9	1	1.0	1.0	2.8%
case 7a	46	16	15	31	10		1	1.0	1.0	32.2%
case 7b	46	16	9	23	13	0.98	1	1.0	1.0	56.5%
case 7c	46	16	15	31	10	0.93	1	1.0	1.0	32.2%
case 8a	46	25	24	49	12		1	1.0	1.0	24.4%
case 8b	46	25	18	32	18	0.99	1	1.0	1.0	56.25%
case 8c	46	25	24	49	12	0.9	1	1.0	1.0	24.4%
case 9a	46	67	66	133	18		1	1.0	1.0	13.5%
case 9b	46	67	45	89	45	0.99	1	1.0	1.0	50.5%
case 9c	46	67	8	126	16	0.96	1	1.0	1.0	12.6%
case 10a	46	108	107	215	22		1	1.0	1.0	10.2%
case 10b	46	108	75	141	75	0.99	1	1.0	1.0	53.1%
case 10c	46	108	6	210	20	0.95	1	1.0	1.0	9.52%
case 11a	46	177	176	353	16		1	1.0	1.0	4.53%
case 11b	46	177	158	196	158	0.99	1	1.0	1.0	80.6%
case 11c	46	177	4	350	14	0.95	1	1.0	1.0	4%

enough. The results are presented in the *Table1*. While we analysed the *case b* (with given stop criterion) we noticed that it was quite high value (0.96..0.99) and the result were a lot of small groups (small means: with small number of rules clustered in it). It tied in longer time of searching those rules, but the process was much precised and it almost always found the relevant rules. Thanks to this, the all process succeeded. By using the method with given *threshold T* we also needed the high value cause only in this case the algorithm stopped clustering before formed one finite cluster with all rules. When the value *T* was between 0.9 and 0.95 we obtained the structure with few grups, which we could search fast and effecively. As we can observe, the disadvantage of the *case b* is that it makes strange. But the good thing is that it always provides relevant rules, and never searches all tree structure (50% – 80%). *Table 1* let us implicated following conclusions:

- Very important is the measure we choose to clustering rules: the *Gower* measure turned out the most effective (resistant to the mistakes),
- In large knowledge bases (with real data) with the multidimensional vectors

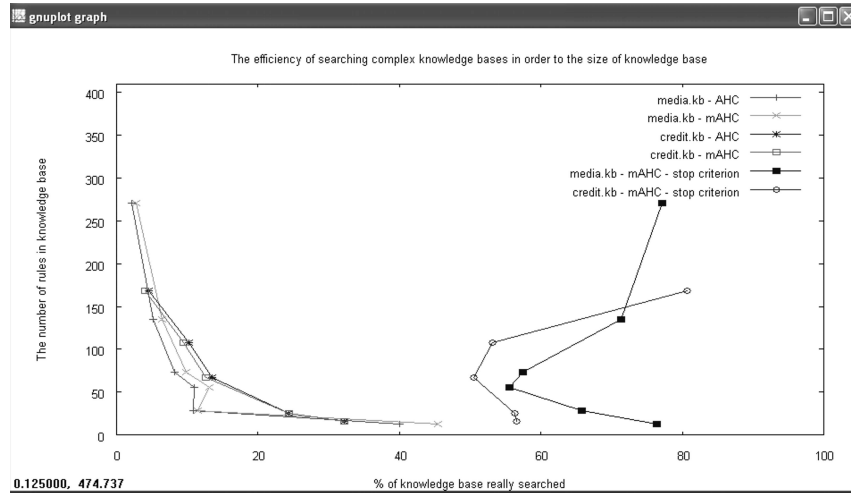


FIGURE 3: The influence the number of rules to the efficiency of the inference process (forward chaining).

of clusters of rules we obtained structures ensuring the high efficiency of the inference process,

- The optimization of the inference process even for classical knowledge bases didn't want to get full recall of the system, but it wanted to get the highest possible precision,
- In large knowledge bases with real data the quality of the created structure is stable enough to obtain the highest values of the parameters such *precision*, *sensitivity* of efficiency of the system.

While we carried out the experiments we noticed that the optimal results we could get by using the backward chaining for all structure of cluster firstly. It leads to get small set (f.e. one rule or few rules) and on that small group (cluster) we can make the forward chaining inference. In our opinion it is new concept of the well known so called *mixed inference*, which till now wasn't well-rounded. We can surely say that, as much bigger knowledge bases we will clustering, as higher efficiency parameters we will achieve.

6 Summary

Large knowledge bases are an important problem in decision systems. Fortunately cluster analysis bring quite useful techniques for smart organisation rules. We propose to change the structure of knowledge base known up to now to hierarchical structure. In this special structure, in the same cluster there are similar rules. After this process we may sure that those rules which are similar together are placed in one group. It let us assume, that in each inference process we can find the most similar groups and get the forward chaining procedure only on this smaller group. It decreases the time of all process and also explores only necessary

new facts, not all facts which we can get from given set of rules and facts. In our opinion, the idea of clustering rules in inference process of decision support systems could be very helpful to improve the efficiency of those systems.

7 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

- M.R. ANDENBURG (1973), *Cluster analysis for applications*, New York, Academic Press.
- R.C. DUBES and A.K. JAIN (1998), *Algorithms for clustering data*, Prentice Hall.
- B.S. EVERITT (1993), *Cluster Analysis (3rd edition)*, Edward Arnold / Halsted Press, London.
- L. KAUFMAN and P.J. ROUSSEEUW (1990), Finding Groups in Data: An Introduction to Cluster Analysis, in *John Wiley Sons, New York*.
- J. KORONACKI and J. ĆWIK (2005), Statystyczne systemy uczące się, in *WNT, Warszawa*.
- 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.
- S. THEODORIDIS and K. KOUTROUMBAS (1999), Pattern Recognition, *Academic Press*.
- K. STĄPOR (2005), Automatyczna klasyfikacja obiektów, *Akademicka Oficyna Wydawnicza EXIT*, Warszawa.