Knowledge Mining Approach for Optimization of Inference Processes in Rule Knowledge Bases

Agnieszka Nowak-Brzezińska and Roman Simiński

University of Silesia, Institute of Computer Science
ul. Będzińska 39, 41-200 Sosnowiec, Poland
{agnieszka.nowak,roman.siminski}@us.edu.pl

Abstract. The main aim of the article is to present modifications of inference algorithms based on information extracted from large sets of rules. The conception of cluster analysis and decision units will be used for discovering knowledge from such data.

Keywords: knowledge bases, inference, decision units, cluster analysis.

1 Introduction

Methods of inference dedicated for rule bases have a long history and are fairly well known, but no significant changes in the inference processes since RETE algorithm were made. The purpose of this study is to present a new approach to inference optimization in rule-based systems and evaluate the effectiveness of the proposed approach. The thesis of this work is that using information discovered in a rule base we transform that rule base into a net of decision units or clusters of rules, which in result improves the efficiency of inference algorithms.

2 Towards a New Modular Structure of a Knowledge Base

The decision units are simple and intuitive models describing relations in a knowledge base (KB), thus, the decision units can be considered as a simple tool for modelling rule bases. The concept of a decision unit is described in the [4,5]. The idea of decision units allows the division of a rule set into subsets called *elementary decision units*, *ordinal decision units* and *general decision units*. An example of using the decision units' properties in the inference optimization task is presented in [1]. Similar issues concering rule base modelling are shown in [5]. By organizing data into clusters, additional information about the analyzed data is acquired. The groups of rules can be further analyzed in order to observe the additional patterns which can be used to simplify the structure of a KB either by simplifying the rules or by reducing their number. Having the set of rules from a KB, in each step the two most similar rules or clusters of rules are found and linked into a cluster. The created structure is similar to a binary tree and the time efficiency of searching such trees is $O(\log_2 n)$. What is more

P. Herrero et al. (Eds.): OTM 2012 Workshops, LNCS 7567, pp. 534–537, 2012.

the clustering algorithm has to be executed only once [3] and though the time complexity of a typical hierarchical algorithm varies, it is often a member of the $O(n^2)$ class.

3 Modifications of Classical Versions of Inference Algorithms

The proposed approach assumes that we reorganize any attributive rule KB from a set of not related rules to groups of similar rules using cluster analysis [1,2] or decision units [5,4]. Thus, we can decompose a rule set into a hierarchical structure. This structure will be a source for extracting interesting information for used in the inference process.

The modification of the backward inference algorithm consists of initializing only promising recursive calls of the classical backward algorithm. The decision units' net provides information which allow a preliminary assessment of whether the call could potentially confirm the nominated subgoal of inference. The classic version of the algorithm doesn't known whether the call is promising — ie, it is unknown whether there is a rule that fits the new goal of inference. Indeed, in real cases very often there is no such rule and the recursive call is unnecessary. In figure 1 we present the backward inference algorithm, which uses the information from the decision unit's net described earlier. Input data for the backward inference algorith consists of: D — set of decition units, F — set of facts, g — goal of inference. The output data of the algorithm is: a set of facts with new facts obtained through inference, and a boolean value - true if the goal g is in the set of facts ($g \in F$), or false otherwise. The modification of the forward inference algorithm includes two steps: finding the relevant rule first, and then confirmating all premises of this selected rule. The first step means that we search the created structure ($Tree$, (T)) by comparring at each level of the tree the similarity between the set of facts (F) and the representatives of the left and right branch of a given node (($node.c$)). Having this we choose a higher value of such similarity. Finally, we get the level of rules (instead of rule clusters) and we start to confirm all premises of a given rule. If all premises are true a conclusion of a given rule is added to the set of facts ($F = F \cup \{node.d\}$). Input data for the forward inference algorithm consists of: T — set of rule clusters, F — set of facts. The output data of the algorithm is: a set of facts with new facts obtained through inference, and a boolean value - true if the re is at least one rule in a given KB, which premises are included in the set of facts, or false otherwise. In figure 2 we present the pseudocode of the forward inference algorithm.

4 Experiments

The goal of the experiments is to analyze the level of optimization brought by the proposed methods. For each KB the results of using a a classical version of inference algorithms and those based on rule clusters or decision units are

```
function bckInfDU( D, g, var F ) : boolean
begin
  if g ∈ F then return true
  else
    A ← ∅
    select d ∈ D where g ∈ O(d)
    truePremise ← false
    while ¬truePremise ∧ {R(d) − A} ≠ ∅ do
      select r ∈ {R(d) − A}
      forall w ∈ cond(r) do
        truePremise ← ( w ∈ F )
        if ¬truePremise ∧ w ∈ IC(d) then
          truePremise ← bckInfDU( D, w, F )
          if ¬truePremise then
            truePremise ← environmentConfirmsFact( w )
            if ¬truePremise then
              break
      endfor
      if ¬truePremise then
        A = A ∪ {r}
    endwhile
  endif
  if truePremise then
    F = F ∪ {g}
  return truePremise
end
```

Fig. 1. Backward Inference Algorithm

```
function forwardInferenceCA( T, F ) : boolean
begin
    node ← T[2n − 1]
    FactsConfirmation ← false
    while ¬ FactsConfirmation ∧ node.level > 0 do
              leftBranch ← T[node.i]
              rightBranch ← T[node.j]
          if similarity(leftBranch.c,F) > similarity(rightBranch.c,F) then
              node ← leftBranch
          else
              node ← rightBranch
              endif
    endhile
    check ← 1
    forall cond ∈ node.c do
      if truePremise(cond) then
          check ← check ∗ 1
      else
          check ← check ∗ 0
      endif
    endfor
    if check == 1 then
      FactsConfirmation ← true
      F = F ∪ {node.d}
    else
      FactsConfirmation ← false
    endif
  return FactsConfirmation
end
```

Fig. 2. Forward Inference Algorithm

Knowledge Base	rules number	Data Driven (AHC)	Goal Driven (AHC)	Goal Driven (DU)
a	32	44%	31%	87%
b	32	44%	31%	87%
c	32	25%	19%	6%
d	26	38%	46%	30%
e	26	31%	38%	8%
f	26	46%	38%	23%
g	49	24%	20%	12%
h	49	8%	20%	2%
i	23	70%	35%	9%
j	23	26%	35%	9%

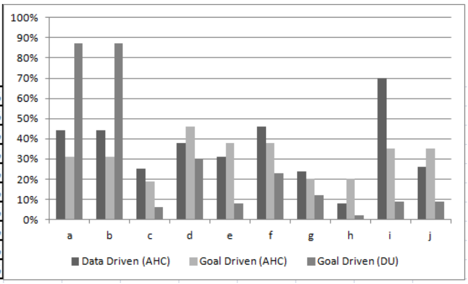

Fig. 3. Results of the experiments with inference optimization

presented. In figure 3 the results of experiments performed on four different KBs are presented. Instead of searching through the whole KB, we need to analyze only a small procentage of all rules to find the proper rule and finish the inference process successfully. There were also cases when during the searching process algorithms based on clustering rules did not find a suitable rule to activate, despite of the fact that it exists in a given KB. That is because, the procedure of creating good descriptions for rule clusters does not always give expected results.

5 Conclusions

In the presented work we propose modifications of inference algorithms based on information discovered in a rule base. We introduce a technique which uses the ,,knowledge maning aproach" — the main goal of this approach is to discover useful, potentially implicit and directly unreadable information from large rule sets. In this paper we combined two approaches — the hierarchical decomposition of large rule bases using cluster analysis and the decision units' conception. The proposed models of a KB allow us to analyze only a small procentage of all rules during the inference process. Only promising decision units are selected for further processing and only a selected subset of the whole rule set is processed in each iteration. Eventually only promising recursive calls are made. The modification of the forward inference algorithm is based on the rule clusters' tree, which is used to find the most relevant rule with a $O(\log 2n - 1)$ time complexity. We also plain to extend proposed algorithm for nondeterministic rules [6].

References

1. Nowak, A., Simiński, R., Wakulicz-Deja, A.: Towards modular representation of knowledge base. In: Intelligent Information Processing and Web Mining. AISC, pp. 421–428. Physica-Verlag, Springer Verlag Company (2006)
2. Nowak, A., Wakulicz-Deja, A.: The Way of Rules Representation in Composited Knowledge Bases. In: Cyran, K.A., Kozielski, S., Peters, J.F., Stańczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. AISC, vol. 59, pp. 175–182. Springer, Heidelberg (2009)
3. Nowak-Brzezińska, A., Simiński, R., Jach, T., Xięski, T.: Towards a Practical Approach to Discover Internal Dependencies in Rule-Based Knowledge Bases. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 232–237. Springer, Heidelberg (2011)
4. Simiński, R., Wakulicz-Deja, A.: Verification of Rule Knowledge Bases Using Decision Units. In: Intelligent Information Systems. AISC, pp. 185–192. Physica Verlag, Springer Verlag Company (2000)
5. Simiński, R.: Decision units approach in knowledge base modeling. In: Recent Advances in Intelligent Information Systems, pp. 597–606. Academic Publishing House EXIT (2009)
6. Delimata, P., Marszał-Paszek, B., Moshkov, M., Paszek, P., Skowron, A., Suraj, Z.: Comparison of Some Classification Algorithms Based on Deterministic and Nondeterministic Decision Rules. In: Peters, J.F., Skowron, A., Słowiński, R., Lingras, P., Miao, D., Tsumoto, S. (eds.) Transactions on Rough Sets XII. LNCS, vol. 6190, pp. 90–105. Springer, Heidelberg (2010)