A Hybrid Approach to a Classification Problem*

Piotr Sapiecha¹, Henry Selvaraj², Jarosław Stańczak³, Krzysztof Sęp³, and Tadeusz Łuba¹

¹ Department of Electronics and Information, Technology Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
² Department of Electrical and Computer Engineering, University of Nevada Las Vegas, Las Vegas, NV 89154-4026, USA
³ System Research Institute, Newelska 6, 01-447 Warsaw, Poland

Abstract. In this paper we present an applications of the argument reduction algorithm as a preprocessing method of decision tables for classifier system. The software tool for classification problem is described. This tool has a hybrid structure consists of proposed reduction strategies mixed with the classical decision rules generator based on genetic algorithms. This tool achieves significant benefits from argument reduction during analysis process of data dependencies.

1 Basic concepts

In this section we will introduce some basic concepts of information systems [8, 9]. The information system contains data-objects, characterized by certain attributes (often two classes of attributes are distinguished: condition x and decision attributes y). Such an information system is called a decision table. The decision table describes conditions that must be satisfied in order to carry out the decisions specified for them. With every decision table we can associate a decision algorithm which is a set of if... then... rules. This decision algorithm can be simplified, which results in optimal description of the data in information system. An information system is a pair $S = (U, A)$, where $U$ is a nonempty set of objects called the universe, and $A$ is a nonempty set of attributes. An argument reduction problem is an algorithmic problem of removing as many condition attributes (input variables) from a given information system (truth table) so that it still remains consistent. In this problem, two notions play an important role, namely the discernibility matrix and discernibility function [13]. A discernibility matrix of a decision table $S$ is a matrix $n \times n$ (where $n$ is the number of rows in the decision table) whose elements are denned as follows:

$$m_{ij} = \{ x \in X : x(i) \neq x(j) \} \text{ iff } \exists y \in Y : y(i) \neq y(j), \text{ otherwise } m_{ij} = \emptyset$$

* This work was supported by the State Committee for Scientific Research of Poland under grant 4 T11D 014 24

M. A. Kłopotek et al. (eds.), Intelligent Information Processing and Web Mining
© Springer-Verlag Berlin Heidelberg 2004
(where \( x(i) \) denotes “a value of the variable \( x \) in the \( i \)-th row of the truth table”). The meaning of this definition is: the element \( m_{ij} \) is an empty set if values of all output variables in rows \( i \) and \( j \) are compatible, and otherwise it is a set of all input variables that have incompatible values in both rows.

A discernibility function \( f \) is a boolean function with boolean variables \( v_i \) corresponding to attributes \( a_i \), and defines as follows: 
\[
f(v_1, v_2, \ldots, v_m) = \bigvee \bigwedge m_{ij}, \quad \text{for } 1 \leq j < i \leq n, m_{ij} \neq \emptyset,
\]
where \( \bigvee m_{ij} \) is the disjunction of all variables \( v_k \) such that \( x_k \in m_{ij} \). When writing discernibility functions, we will give the variables of \( f \) the same names as the variables from \( X \) when no confusion can arise. The introduction of discernibility matrix is very useful for the process of argument reduction. However the argument reduction problem is \( NP \)-hard.

## 2 Properties of the minimum transversal problem

In this section we will discuss some combinatorial properties of the argument reduction problem, notably transversal problem (blocking sets) \([2, 3]\). One of the most common ways of solving the argument reduction problem is to reduce it to another problem, a well-known combinatorial minimum transversal problem. Since this approach will be used in one of the algorithms proposed for the problem, it is necessary to describe the details of this construction. A **hypergraph** is a pair \( H = (V, E) \), where \( V \) is a nonempty set of **vertices** and \( E \) is a set of **edges**. Each edge is a subset of vertices, i.e. \( E \subseteq 2^V \). Note that an edge can contain any number of vertices (even one), which makes a difference between hypergraphs and graphs. In fact, hypergraph can be viewed as a direct generalization of graphs. A **transversal** \( T \subseteq V \) is a subset of vertices which has the property \( \forall e \in E : e \cap T \neq \emptyset \). In other words, a transversal is a set of vertices that covers (or blocks, hence the other name: blocking set) all the edges. A transversal is called minimum if there exists no other transversal having less elements. As we recall, the idea of argument reduction is to find a minimum implicant in the discernibility function. The discernibility function can be expressed as a hypergraph, where edges correspond to variables, and edges correspond to elements from the discernibility matrix.

## 3 Algorithms for argument reduction

In this section we will present some practical proposals for solving the argument reduction problems. One of the possible algorithms is an exact one, using a backtracking approach. But the worst-case number of steps to find a minimum transversal is \( O(2^{|V|}) \). Thus it is practically useless to solve real problems. The other one is much faster, but is not an exact, namely an approximation algorithm often called a greedy algorithm. It operates directly on a hypergraph constructed from given information system. Because of this, it essentially is the algorithm for solving the set-covering problem. Therefore,