Expert Systems With Applications 42 (2015) 9612–9619

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A fast hardware software platform for computing irreducible testors

Vladimir Rodríguez-Diez [a], José Francisco Martínez-Trinidad [b], Jesús Ariel Carrasco-Ochoa [b], Manuel Lazo-Cortés [b], Claudia Feregrino-Uribe [b], René Cumplido [b,*]

[a] Universidad de Camaguey, Carretera Circunvalación Norte Km. 5, Camaguey, 74650, Cuba
[b] Instituto Nacional de Astrofísica, Óptica y Electrónica, Tonanzintla, Puebla, 72840, Mexico

ARTICLE INFO

Keywords:
Feature selection
Testor theory
Hardware architecture
FPGA

ABSTRACT

Among the systems involved with data and knowledge that give answers, solutions, or diagnoses, based on available information; those based on feature selection are very important since they allow us to solve important tasks into pattern recognition and decision making areas. Feature selection consists in finding a minimum subset of attributes that preserves the ability to discern between objects from different classes. Testor theory is a convenient way to solve this problem since a testor is defined as a subset of attributes that can discern between objects from different classes; and an irreducible testor is a minimal subset with this property. However, the computation of these minimal subsets is a problem whose space complexity grows exponentially regarding the number of attributes. Therefore, in the literature, several hardware implementations of algorithms for computing testors, which take advantage of the inherent parallelism in the evaluation of testor candidates, have been proposed. In this paper, a new fast hardware software platform for computing irreducible testors is introduced. Our proposal follows a pruning strategy that, in most cases, reduces the search space more than any other alternative reported in the literature. The experimental results show the runtime reduction achieved by the proposed platform in contrast to other state-of-the-art hardware and software implementations.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Feature selection for supervised classification consists in identifying those attributes that provide relevant information for the classification process. This procedure does not only reduce the computational cost of the classification process by eliminating superfluous information, but in some cases it could even provide better classification accuracy. Testor theory can be used for feature selection as shown in (Martínez-Trinidad & Guzmán-Arenas, 2001; Ruiz-Shulcloper, 2008). A testor is defined as a subset of attributes that can discern objects from different classes. An irreducible testor is a subset of attributes such that every attribute is indispensable for satisfying the testor condition. Finding all irreducible testors has been proven to be an NP-hard problem (Skowron & Rauszer, 1992).

Feature selection is also a key aspect in expert and intelligent systems. Martínez, León, and García (2007) used testor theory to propose a new case-based approach for developing intelligent teaching-learning systems. Irreducible testors are computed in (Medina, Martínez, García, Chávez, & García, 2007) to select problem attributes for an intelligent tutoring system which has the capacity of adapting its interaction to the user's specific needs. Recently, Torres et al. (2014) used testors for determining risk factors associated with transfusion related to acute lung injuries. Another approach to feature selection based on rough set reducts has been found to have a close relationship with irreducible testors (Lazo-Cortés, Martínez-Trinidad, Carrasco-Ochoa, & Sánchez-Díaz, 2015). Therefore, expert systems can be benefited from rough set methods, especially for feature selection by means of reducts, as shown by Yahia, Mahmod, Sulaiman, and Ahmad (2000).

Recently, there is an increasing popularity of architectures based on Field Programmable Gate-Array (FPGA) for solving complex computational problems. Several hardware software platforms based on this technology have been reported (Compton & Hauck, 2002; Pocek, Tessier, & DeHon, 2013). In these platforms, the software component handles those tasks less suited for hardware implementation, and it is also responsible of configuring the FPGA, as well as handling communication with the hardware component. The hardware component, on the other hand, performs those operations with a high parallelism degree.

* Corresponding author. Tel.: +52 222 2663100x8225; fax: +52 222 2663152.
E-mail addresses: vladimir.rdguez@gmail.com (V. Rodríguez-Diez), fmartine@ccc.inaoep.mx (J.F. Martínez-Trinidad), ariel@inaoep.mx (J.A. Carrasco-Ochoa), mlazo@inaoep.mx (M. Lazo-Cortés), cferegrino@inaoep.mx (C. Feregrino-Uribe), rcumplido@inaoep.mx (R. Cumplido).

http://dx.doi.org/10.1016/j.eswa.2015.07.037
0957-4174/© 2015 Elsevier Ltd. All rights reserved.

In spite of advances in the theoretical aspects of computing irreducible testors (Djukova, 2005; Kudryavtsev, 2006; Martínez-Trinidad & Guzmán-Arenas, 2001), there are no hardware implementations reported aside from (Cumplido, Carrasco, & Feregrino, 2006; Rojas, Cumplido, Carrasco-Ochoa, Feregrino, & Martínez-Trinidad, 2007, 2012). In the first work, an FPGA-based brute force approach for computing testors was proposed (Cumplido et al., 2006). This first approach did not take advantage of dataset characteristics to reduce the number of candidates to be tested; thus all 2^n combinations of n attributes have to be tested. Then, in (Rojas et al., 2007) a hardware architecture of the BT algorithm (Ruiz-Shulcloper, Aguila-Feros, & Bravo-Martínez, 1985) for computing irreducible testors was implemented. This algorithm uses a candidate pruning process for avoiding many unnecessary candidate evaluation, reducing the number of verifications of the irreducible testor condition. These two previous works computed a set of testors on the FPGA device whilst irreducible condition was evaluated afterwards by the software component in the hosting PC. Thus Rojas, Cumplido, Carrasco-Ochoa, Feregrino, and Martínez-Trinidad (2012) proposed a hardware software platform for computing irreducible testors that implemented the BT algorithm, as in Rojas et al. (2007), but it also included a new module that eliminates most of the non irreducible testors before transferring them to a host software application for final filtering. One disadvantage of these approaches is the huge amount of data that must be transferred to the PC. Consequently, in Rodríguez et al. (2014) we proposed a modification to this platform in order to compute irreducible testors in the hardware component.

Several hardware accelerations have been recently reported for feature selection in rough set theory. Authors of Grze, Kopczynski, and Stepaniuk (2013), Kopczynski, Grze, and Stepaniuk (2014) presented an FPGA based platform for computing a reduct. Tiwari, Kothari, and Shah (2013) presented various algorithms for attribute reduction using concepts of rough set theory and implemented the Quick Reduct algorithm in a hardware fashion. Then, Tiwari and Kothari (2014) presented a thorough survey on hardware implementation of rough set algorithms. These hardware implementations aim to compute a single reduct. Jensen, Tuson, and Shen (2014) addressed the lack of guarantees for finding an attribute subset with minimal cardinality, which is the main drawback of these approaches.

In this paper, we present an efficient hardware software platform for computing irreducible testors based on the CT-EXT algorithm proposed in (Sánchez-Díaz & Lazo-Cortés, 2007). The main contribution of this work is the design and implementation of a hardware architecture that traverses the search space in a different order than that presented in (Rodríguez et al., 2014; Rojas et al., 2007, 2012). This new strategy evaluates less candidate subsets than previous architectures, which results in shorter runtime. In comparison to the software versions of CT-EXT (Sánchez-Díaz & Lazo-Cortés, 2007; Sánchez-Díaz, Piza-Davila, Lazo-Cortés, Mora-González, & Salinas-Luna, 2010), our proposal evaluates a candidate every clock cycle, which leads to a faster execution. The runtime gain of our new hardware software platform is demonstrated throughout experiments over synthetic datasets.

The rest of this paper is structured as follows. Section 2 introduces the CT-EXT algorithm. Section 3 describes the proposed architecture. Evaluation of the proposed platform and a discussion of the experimental results are presented in Section 4. Finally, Section 5 shows our conclusions and some directions for future work.

2. CT-EXT algorithm

CT-EXT is one of the fastest algorithms for computing irreducible testors reported in the state of the art (Piza-Davila, Sánchez-Díaz, Aguirre-Salado, & Lazo-Cortes, 2015; Sánchez-Díaz & Lazo-Cortés, 2007; Sánchez-Díaz et al., 2010). In order to describe this algorithm we introduce some definitions and notations.

Let TM be a training matrix with k objects described through n attributes of any type $R = \{x_1, \ldots, x_n\}$ and grouped in r classes. Let DM be the binary pairwise comparison matrix, called dissimilarity matrix (0=similar, 1=dissimilar), obtained by means of attribute by attribute comparisons of every pair of objects from TM belonging to different classes. DM has m rows and n columns, where usually $m >> k$.

As it is known, DM commonly contains redundant rows, so algorithms for computing irreducible testors frequently work on the submatrix called basic matrix (BM); which is obtained from DM by eliminating redundant rows. For obtaining BM from DM, absorption laws are applied. Rows in BM are called basic rows.

Let T be a subset of attributes, T is a testor of BM if the attributes in T do not form a zero row in BM. It means that every row in BM has at least a 1 in those columns corresponding to attributes belonging to T. We say that a testor T is an irreducible testor if all the proper subsets of T are not testors.

We can interpret an irreducible testor as a subset of attributes being jointly sufficient and individually necessary to differentiate every pair of objects belonging to different classes.

During the search, CT-EXT follows the idea that an attribute contributes to a subset T (candidate to be an irreducible testor) if after adding this attribute to T, the attributes in T form less zero rows in BM than the amount of zero rows before adding the attribute. This idea is used for pruning the search space.

The following proposition, introduced and proved in (Sánchez-Díaz & Lazo-Cortés, 2007), constitutes the basis for the CT-EXT algorithm.

Proposition 1. *Given $T \subseteq R$ and $x_j \in R$ such that $x_j \notin T$. If x_j does not contribute to T, then $T \cup \{x_j\}$ cannot be a subset of any irreducible testor.*

Algorithm 1 shows the pseudocode of CT-EXT, a detailed explanation of this algorithm can be seen in (Sánchez-Díaz & Lazo-Cortés, 2007). The function SortBM(BM) sorts the basic matrix as follows. Randomly select one of the rows of BM with the fewest number of 1's. The selected row goes first and all columns in which it has a 1 are moved to the left.

The function Evaluate(BM, T) returns three values: *testor, irreducible* and *zero_rows*. *testor* is TRUE if the set T is a testor of BM and FALSE otherwise. *irreducible* is TRUE if the set T is an irreducible testor and FALSE otherwise. *zero_rows* is the amount of zero rows of T. The function LastOne(T) returns the position of the rightmost element in the set T.

Let us consider the basic matrix of Table 1, with $m = 3$ (rows) and $n = 5$ (attributes). After the ordering step we obtain the matrix shown in Table 2. Table 3 illustrates the application of the CT-EXT algorithm over this ordered basic matrix.

Table 1
Basic matrix for the example.

x_0	x_1	x_2	x_3	x_4
1	0	0	1	1
0	1	1	0	1
1	1	0	0	1

Table 2
Ordered basic matrix obtained from the matrix of Table 1.

x_0	x_3	x_4	x_1	x_2
1	1	1	0	0
0	0	1	1	1
1	0	1	1	0

Download English Version:

https://daneshyari.com/en/article/382261

Download Persian Version:

https://daneshyari.com/article/382261

Daneshyari.com