Applied Soft Computing 38 (2016) 453–468

Contents lists available at ScienceDirect

Applied Soft Computing

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

Grammar induction using bit masking oriented genetic algorithm and comparative analysis

Hari Mohan Pandey[a,*], Ankit Chaudhary[b], Deepti Mehrotra[c]

[a] Department of Computer Science & Engineering, Amity University, Uttar Pradesh, India
[b] Department of Computer Science, Truman State University, USA
[c] Amity School of Engineering & Technology, Amity University, Uttar Pradesh, India

ARTICLE INFO

Article history:
Received 2 May 2014
Received in revised form
24 September 2015
Accepted 24 September 2015
Available online 17 October 2015

Keywords:
Bit-masking oriented data structure
Context free grammar
Genetic algorithm
Grammar induction mask-fill operator
Premature convergence

ABSTRACT

This paper presents bit masking oriented genetic algorithm (BMOGA) for context free grammar induction. It takes the advantages of crossover and mutation mask-fill operators together with a Boolean based procedure in two phases to guide the search process from *i*th generation to $(i+1)$th generation. Crossover and mutation mask-fill operations are performed to generate the proportionate amount of population in each generation. A parser has been implemented checks the validity of the grammar rules based on the acceptance or rejection of training data on the positive and negative strings of the language. Experiments are conducted on collection of context free and regular languages. Minimum description length principle has been used to generate a corpus of positive and negative samples as appropriate for the experiment. It was observed that the BMOGA produces successive generations of individuals, computes their fitness at each step and chooses the best when reached to threshold (termination) condition. As presented approach was found effective in handling premature convergence therefore results are compared with the approaches used to alleviate premature convergence. The analysis showed that the BMOGA performs better as compared to other algorithms such as: random offspring generation approach, dynamic allocation of reproduction operators, elite mating pool approach and the simple genetic algorithm. The term success ratio is used as a quality measure and its value shows the effectiveness of the BMOGA. Statistical tests indicate superiority of the BMOGA over other existing approaches implemented.

1. Introduction

Grammar induction or grammar learning deals with idealized learning procedures for acquiring grammars on the basis of the evidence about the languages [45,62,63]. It was extensively studied [6,46–50,63] due to its wide fields of application to solve practical problems in a variety of different fields, which includes a compilation and translation, human machine interaction, graphic languages, design of programming language, data mining, computational biology, natural language processing, software engineering and machine learning, etc.

The first learning model was proposed by Gold in 1967 [26]. Gold addressed the question "*Is the information sufficient to determine which of the possible languages is the unknown language*?" [26]. It was shown that an inference algorithm can identify an unknown language in the limit from complete information in a finite number of steps. The key issue with the Gold's approach is that there is no information present with inference algorithm about identification of correct grammar because it is always possible that next sample may invalidate the previous hypothesis. Angluin [58] proposed "*tell tales*" (a unique string makes the difference between languages) to avoid the drawback. Although Gold [26] laid the foundation of grammar inference, Bunke and Sanfeliu [38] presented the first usable grammatical inference algorithm in syntactic pattern recognition community with the aim of classification and analysis of patterns, classification of biological sequence, character recognition, etc. The main drawback of this algorithm was it only dealt with positive data, was unable to deal with noisy data, did not fit exactly into a finite state machine and therefore good formal language theories were lost. Stevenson and Cordy [39,40] explain that theorists and empiricists are the two main groups contributing in the field of grammar inference. Language classes and learning models were considered by theorists group to set up the boundaries of what is learnable and how efficiently it can be learned. On the other hand empiricists group dealt with a practical problem by solving it; finally they have made a significant contribution

* Corresponding author. Tel.: +91 9810625304.
E-mail address: profharimohanpandey@gmail.com (H.M. Pandey).

http://dx.doi.org/10.1016/j.asoc.2015.09.044
1568-4946/

in grammatical inference. Teacher and query is another learning model, where a teacher also referred to as an Oracle knows the target languages and is capable to answer particular types of questions/queries from the inference algorithm. Six types of queries were described by Angluin [59], two of which are membership and equivalence queries that have significant impact on learning. In case of membership queries, the inference algorithm presents either "*yes*" or "*no*" as answer to the oracle, whereas oracle receives "*yes*" if the hypothesis is true and "*no*" otherwise by inference algorithm. Valiant [60] presented *Probably Approximately Correct* (PAC) Learning Model, which takes the advantages of both identification of the limit and *teachers and queries* learning models. The PAC learning model is different from other two former learning models because of two reasons: first it does not guarantee exact identification with certainty, second compromises between accuracy and certainty. The problem with the PAC model is that the inference algorithm must learn in polynomial time under all distributions, but it is believed to be too strict in reality. These problems occur because many apparently simple classes are either known to be NP-hard or at least not known to be polynomial learnable for all distributions [39]. To mitigate this issue, Li et al. [61] proposed inference algorithm to consider the simple distribution only. Apart from the above popular learning models, many researchers explain the suitability of the neural network for grammar inference problem. Neural network shown the ability to maintain a temporal internal state like a short term memory [40]. In case of the neural network, a set of inputs, and their corresponding outputs (Yes: string is in the target language, No: otherwise) and a defined function need to learn, which describes those input-output pairs [40]. Alex et al. [54] conducted experiments for handwriting recognition using neural network and it was explained that this network has the capability to predict subsequent elements from an input sequence of elements. Cleeremans et al. [53] implemented a special case of a recurrent network presented by Elman [55] known a simple recurrent network to approximate a DFA (deterministic finite automata). Delgado and Pegalajar [56] presented a multi-objective genetic algorithm to analyze the optimal size of a recurrent neural network to learn positive and negative examples. Authors of [56] utilized the merits of self organizing map to determine the automation, once the training is completed. Although neural network is widely used for grammar inference since it was found good at simulating an unknown function, but it was observed that there is no way to reconstruct the function from the connections in a trained network [40]. A detailed survey of various grammar inference algorithms is presented in [6,39,40,51,52,57]. Inductive inference is the process of making generalization from the input (string). Wyard [3] presented the impact of different grammar representation and experimental result show that the evolutionary algorithm (EA) using standard context free grammar (CFG) (Backus Naur Form (BNF)) outperformed others. Thanaruk and Okumaru [27] classified grammar induction methods into three categories, namely; supervised, semi-supervised and unsupervised depending on the type of required data. Javed et al. [28] presented genetic programming (GP) based approach to learn CFG. The work presented in [28] was the extension of the work done in [3] by utilizing grammar specific heuristic operator. In addition, better construction of the initial population was suggested. Choubey and Kharat [29] presented a sequential structuring approach to perform coding and decoding of binary coded chromosomes into terminal and non-terminals and vice versa. A CFG induction library was presented using the GA contains various Java classes to perform grammar inference process [30,9]. The proposed approach for grammar induction is discussed in Section 2.

Section 3 discusses an important issue in GA known as premature convergence. In evolutionary search, the diversity (difference among individual at the genotype or phenotype levels) decreases as the population converges to a local optimum. It is the situation when an extraordinary individual takes over a significant proportion of finite population and leads towards an undesirable convergence. Diverse population is a prerequisite for exploration in order to avoid premature convergence to local optima [44,64]. On the other hand, promoting diversity in an evolutionary process is good where exploitation is needed. Hence, to reach to the global solution and explore the search space adequately, maintaining population diversity is very important. Also, it had been explained clearly that degree of population diversity is one of the main cause of premature convergence [42,43]. It is necessary to select the best solution of the current generation to direct the GA to reach to the global optimum. The tendency to select the best member of the current generation is known as selective pressure. It is also a key factor that plays an important role in maintaining genetic diversity. A proper balancing is required between genetic diversity and selection pressure to direct the GA search to converge in a time effective manner and to achieve global optima. High selective pressure reduces the genetic diversity; hence in this situation premature convergence can occur while the little selective pressure prohibits the GA to converge to an optimum in reasonable time. An approach to increase the population size is not sufficient to alleviate premature convergence because any increase in population size will add twofold cost both extra computing time and number of generations to converge on global optima. Applying mutation alone converts GA search into a random search whereas crossover alone generates a sub-optimal solution. Also, applying selection, crossover and mutation together may result the GA search to noise tolerant hill climbing approach. Several approaches to handle the premature convergence are suggested [41]. Although these approaches address the premature convergence in their ways by maintaining population diversity, these methods suffer with its own strengths and weaknesses. Pandey et al. [41] presented a detailed and comprehensive comparison of various premature convergence handling approaches on the basis of their merits, demerits and other important factors.

The focus of this paper is towards development of a grammar induction tool and addressing premature convergence in GA. An effort is made towards utilizing the applicability of the bit masking oriented data structures to apply mask-fill reproduction operators and the Boolean based procedure. The diversity of the population is created through the Boolean based procedure during offspring generation, which helps in avoiding premature convergence in a grammar inference problem using the GA. A learning system uses a finite set of examples and to train the system sufficient training set is needed, which can be achieved employing generalization and specialization. Authors [65–67] discussed the importance of generalization and specialization for the learning systems. It was discussed that overgeneralization is a major issue in the identification of grammars for formal languages from the positive data [40] [65]. Therefore, the minimum description length principle is used to address this issue (discussed in Section 2.4). This paper makes five main contributions, as enumerated below:

(a). The four steps of grammar induction are presented. An example is considered to illustrate the step by step process of grammar induction.
(b). A sequential mapping of chromosome is presented. In addition, the applicability of the parser is shown helps in removing insufficiencies and decide acceptance and rejection of training data.
(c). An algorithm "*BMOGA*" is presented for CFG induction from the positive and negative corpus. Furthermore, we show the crossover and mutation mask-fill reproduction operators and detailed procedure of the new offspring generation.