Computers & Industrial Engineering 67 (2014) 160–167

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

An Exponential Monte-Carlo algorithm for feature selection problems ☆

Salwani Abdullah [a,*], Nasser R. Sabar [b], Mohd Zakree Ahmad Nazri [a], Masri Ayob [a]

[a] Data Mining and Optimization Research Group (DMO), Center for Artificial Intelligence Technology (CAIT), Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
[b] The University of Nottingham Malaysia Campus, Jalan Broga, 43500 Semenyih, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 25 December 2010
Received in revised form 9 October 2013
Accepted 29 October 2013
Available online 12 November 2013

Keywords:
Feature selection
Exponential Monte-Carlo
Local search

ABSTRACT

Feature selection problems (FS) can be defined as the process of eliminating redundant features while avoiding information loss. Due to that fact that FS is an NP-hard problem, heuristic and meta-heuristic approaches have been widely used by researchers. In this work, we proposed an Exponential Monte-Carlo algorithm (EMC-FS) for the feature selection problem. EMC-FS is a meta-heuristic approach which is quite similar to a simulated annealing algorithm. The difference is that no cooling schedule is required. Improved solutions are accepted and worse solutions are adaptively accepted based on the quality of the trial solution, the search time and the number of consecutive non-improving iterations. We have evaluated our approach against the latest methodologies in the literature on standard benchmark problems. The quality of the obtained subset of features has also been evaluated in terms of the number of generated rules (descriptive patterns) and classification accuracy. Our research demonstrates that our approach produces some of the best known results.

1. Introduction

Recently, there has been a great deal of attention paid to feature selection in data mining. Feature selection can be defined as the problem of finding a minimal subset of features while avoiding information loss (Pawlak, 1982, 1991). Removing redundant and misleading features can improve the performance and efficiency of a learning process (Pawlak, 1991). It is known that finding a smallest subset of features is a NP-hard problem (Pawlak, 1982). The optimal subset of features is determined by both relevancy and redundancy aspects. A feature is said to be relevant if a decision depends on it; if no decision depends on the feature, it is not relevant. However, a feature can also be considered to be redundant if it is highly correlated with other features (Pawlak, 1991). Hence, the aim is to search for features that are strongly correlated with the decision feature. Finding an optimal subset of features varies from one problem to another depending on the problem complexity.

During the last decade, there have been a number of approaches utilised to solve feature selection problems. These approaches can usually be classified as either random or heuristic based methods. In random search based method, the main idea is to randomly generate a subset of feature until optimal subset is found or reached the predefined termination criterion. The optimal subset has fewer numbers of features when compared to the original one, but the information is the same. However, despite being simple to implement, random search based methods are impractical when dealing with a huge dataset and the quality of the generated solution is unsatisfactory.

On the other hand, heuristic and meta-heuristic approaches have been successfully applied to feature selection problems. These can be classified into local search methods and population based methods. Example of population based methods are: genetic algorithms (Wroblewski, 1995; Jensen & Shen, 2003), ant colony (Jensen & Shen, 2003; Ke, Feng, & Ren, 2008), and scatter search (Jue, Hedar, Guihuan, & Shouyang, 2009). Example of local search methods are: simulated annealing (Jensen & Shen, 2004), tabu search (Hedar, Wang, & Fukushima, 2008), variable neighbourhood search (Arajy & Abdullah, 2010), iterative algorithm with composite neighbourhood structure (Jihad & Abdullah, 2010), great deluge algorithm (Abdullah & Jaddi, 2010), nonlinear great deluge (Jaddi & Abdullah, 2013a), and constructive hyper-heuristics (Abdullah, Sabar, Ahmad Nazri, Turabieh, & McCollum, 2010). Hybrid approaches have also been tested on feature selection problems such as the hybridization between fuzzy logic and record-to-record travel algorithm (Mafarja & Abdullah, 2013a), hybrid genetic algorithm with great deluge (Jaddi & Abdullah, 2013b), and memetic algorithm (Mafarja & Abdullah, 2013b). Other approaches and surveys can be found in Jensen and Shen (2004), Zhang, Qiu, and Wu (2007) and Skowron and Grzymala-Busse (1994).

In this work, we propose an Exponential Monte-Carlo algorithm for solving feature selection problems (EMC-FS). EMC-FS is similar

☆ This manuscript was processed by Area Editor Ibrahim H. Osman.

* Corresponding author. Tel.: +60 389216667.
E-mail addresses: salwani@ftsm.ukm.my (S. Abdullah), Nasser.Sabar@nottingham.edu.my (N.R. Sabar), mzan@ftsm.ukm.my (M.Z. Ahmad Nazri), masri@ftsm.ukm.my (M. Ayob).

0360-8352/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cie.2013.10.009

to simulated annealing (SA) algorithm but employs a different mechanism to escape from local optima. It belongs to the class of no-monotonic SA algorithms that were introduced in Osman (1993) and Osman and Christofides (1994) but uses a different mechanism to accept worse solutions. In this work, we select EMC to solve feature selection problems due to its ability to control the intensification/diversification problem faced by most of the local search algorithms, has less number of parameters that need to be tuned in advance and shown to be an effective method when solving hard optimization problems (Abdullah, Burke, & McCollum, 2005, 2007; Ayob & Kendall, 2003; Sabar, Ayob, & Kendall, 2009).

The proposed method has been tested on UCI datasets (Blake & Merz, 1998) and we used the rough set theory to evaluate the obtained subset of features (Pawlak, 1982, 1991). Furthermore, in contrast to available feature selection methods that they only report the numbers of generated features, we also evaluate the quality of the generated subset of features in terms of the number of generated rules (descriptive patterns) and the classification accuracy.

2. Problem description

In this section, we describe the feature selection problem, solution representation and the objective function.

2.1. Feature selection problems

Feature selection (FS) problem is a pre-processing task in data mining and has been intensively studied by researchers, due to it critical effects on the learning process. Given a set of features, the primary goal of a feature selection is to select, among the possible subset of features, the smallest subset in such a way that the information of the selected subset is the same as the original set of features and can generate a better accuracy (Jensen & Shen, 2003; Pawlak, 1982, 1991). In particular, FS can be represented by a pair of (A, γ) where A represents the original set of features (the search space of all possible solutions) and γ is the objective function which evaluates how good the selected subset is. Then the problem is to find the best subset of features $s \in A$ in such a way that the generated subset s has a smaller number of features compared to the original set A. The goal of a searching method is to search through all possible subsets of features and determine the most informative subset.

2.2. Solution representation

In this work, a solution is represented in a one-dimensional vector. The size of the vector is equal to the number of features in the original dataset. Each cell in the vector is represented by "1" or "0". The value "1" shows that the corresponding feature is selected, while "0" mean the corresponding feature is not selected.

2.3. The objective function

The objective function, γ, evaluates how good the selected subset of features compared to previous one. In this work, the generated subset of features by the search method is accepted if the objective function of the generated subset is better than the previous one or both can lead to same objective function value but the generated one has a smaller number of features. In this work, we use the dependency degree of rough set theory as the objective function to evaluate the generated subset of features (Pawlak, 1982, 1991). The dependency degree calculates data dependencies and returns a value between zero and one. The generated subset of features is called an informative if the returned value by the dependency degree is equal to one (maximization problem). That is the algorithm keeps generating a new subset of features by adding or deleting features from a given subset until the value returned by the dependency degree is equal to one. In particular, given two solutions (two subsets of features), i.e., current solution, Sol, and trial solution, Sol^*, the trial solution Sol^* is accepted if there is an enhancement in the objective function value (i.e., if $\gamma(Sol^*) > \gamma(Sol)$). If the objective function value for both solutions are the same (i.e., $\gamma(Sol^*) = \gamma(Sol)$), then the solution with the lowest number of features (denoted as #) will be accepted. In this work, the rough set theory is used to discover data dependencies and EMC-FS to search the space of all available subset of features. More details about the rough set theory for feature selection problems can be found in (Jensen & Shen, 2003; Ke et al., 2008; Pawlak, 1982, 1991).

3. Exponential Monte-Carlo algorithm for feature selection (EMC-FS)

In this work, we propose EMC-FS to deal with the feature selection problem. The EMC-FS algorithm adapted in this work aims to investigate the impact of the algorithm with fewer parameters dependent when solving the feature selection problem compared to other available approaches that have several parameters to be tuned in advance. The following subsections cover the initial solution generation method and neighbourhood operator, and the EMC-FS algorithm.

3.1. Initial solution method and the neighbourhood operator

The initial solution is constructed randomly, where each cell in the vector is assigned a value "1" or "0" at random. In this work, we use a systematic neighbourhood operator to generates a neighbourhood solution by starting from the first element of the array and use a flip strategy to change each entry of the vector and decide to accept/reject. If the value of the selected cell is "1", it will be changed to "0". This change means that one feature has been deleted from the current solution. If the value of the selected cell is "0", then it will be changed to "1", which means that one feature has been added to the current solution.

3.2. The algorithm: EMC-FS

The EMC algorithm was introduced by Ayob and Kendall (2003). EMC is similar to the acceptance criterion in a simulated annealing algorithm but no cooling schedule is required. The algorithm will always accept the better solution. A worse solution is likely to be accepted based on a certain probability that depends on the following three parameters: the quality of the solution (represented as a dependency degree), the number of iterations, and the number of consecutive no-improving iterations (we consider this third parameter as a period where the search is trapped in the local optima).

The acceptance probability is computed by $e^{-\Theta/\lambda}$ where $\Theta = \delta * t$, $\lambda = q$, where δ is the difference between the objective function of the current and trial solutions, i.e., $\delta = \gamma(Sol) - \gamma(Sol^*)$, t is an iteration counter, and q is a controller parameter that represents a consecutive no-improving counter. The probability of accepting a worse solution decreases as the number of iterations t, increases. However, if there is no improvement for a number of consecutive iterations, then the probability of accepting a worse solution will increase according to the objective function of the trial solution and the number of iterations. A worse solution is more likely to be accepted if δ is small or q is large. This is a diversification factor where the search will diversify when it is trapped