Engineering Applications of Artificial Intelligence 55 (2016) 320–330

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Mining high-utility itemsets based on particle swarm optimization

Jerry Chun-Wei Lin [a,*], Lu Yang [a], Philippe Fournier-Viger [b], Jimmy Ming-Thai Wu [c], Tzung-Pei Hong [d,f], Leon Shyue-Liang Wang [e], Justin Zhan [c]

[a] School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China
[b] School of Natural Sciences and Humanities, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China
[c] Department of Computer Science, University of Nevada, Las Vegas, USA
[d] Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan, ROC
[e] Department of Information Management, National University of Kaohsiung, Kaohsiung, Taiwan, ROC
[f] Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, ROC

ARTICLE INFO

Article history:
Received 18 June 2015
Received in revised form
19 May 2016
Accepted 31 July 2016

Keywords:
Evolutionary computation
High-utility itemsets
Particle swarm optimization
Discrete
Transaction-weighted utility model

ABSTRACT

High-utility itemset mining (HUIM) is a critical issue in recent years since it can be used to reveal the profitable products by considering both the quantity and profit factors instead of frequent itemset mining (FIM) or association-rule mining (ARM). Several algorithms have been presented to mine high-utility itemsets (HUIs) and most of the designed algorithms have to handle the exponential search space for discovering HUIs when the number of distinct items and the size of database are very large. In the past, a heuristic HUPE_{umu}-GRAM algorithm was proposed to mine HUIs based on genetic algorithm (GA). For the evolutionary computation (EC) techniques of particle swarm optimization (PSO), it only requires fewer parameters compared to the GA-based approach. Since the traditional PSO mechanism is used to handle the continuous problem, in this paper, the discrete PSO is adopted to encode the particles as the binary variables. An efficient PSO-based algorithm namely $\text{HUIM-BPSO}_{\text{sig}}$ is proposed to efficiently find HUIs. It first sets the number of discovered high-transaction-weighted utilization 1-itemsets (1-HTWUIs) as the size of a particle based on transaction-weighted utility (TWU) model, which can greatly reduce the combinational problem in evolution process. The *sigmoid* function is adopted in the updating process of the particles of the designed $\text{HUIM-BPSO}_{\text{sig}}$ algorithm. Substantial experiments on real-life datasets show that the proposed algorithm has better results compared to the state-of-the-art GA-based algorithm.

1. Introduction

Knowledge discovery in database (KDD) is an emerging issue since the potential or implicit information can be found from a very large database. Most of them, frequent itemset mining (FIM) or association-rule mining (ARM) has been extensively developed to mine the set of frequent itemsets in which their occurrence frequency of an itemset is no less than minimum support threshold or its confidence is no less than minimum confidence threshold (Agrawal and Srikant, 1994; Chen et al., 1996). Since only the occurrence frequency of itemsets is discovered whether in FIM or ARM, it is insufficient to identify the high profit itemsets especially when the itemset is rarely appeared but has high profit value. To solve the limitation of FIM or ARM, high-utility itemset mining (HUIM) (Yao et al., 2004; Yao and Hamilton, 2006) was designed to discover the "useful" and "profitable" itemsets from the quantitative databases. An itemset is considered as a high-utility itemset (HUI) if its utility value is no less than the user-specified minimum utility threshold.

The traditional algorithms of HUIM have to handle the "exponential problem" of a very huge search space while the number of distinct items or the size of database is very large. Evolutionary computation is an efficient way and able to find the optimal solutions using the principles of natural evolution (Cattral et al., 2009). The genetic algorithm (GA) (Holland, 1975) is an optimization approach to solve the NP-hard and non-linear problems and used to investigate a very large search spaces to find the optimal solutions based on the designed fitness functions with various operators such as selection, crossover, and mutation. Several GA-based algorithms have been extensively studied and applied to the variants of optimization problems (Kannimuthu and Premalatha, 2014; Martnez-Ballesteros et al., 2010; Salleb-Aouissi et al., 2007).

* Corresponding author.
E-mail addresses: jerrylin@ieee.org (J.-W. Lin), luyang@ikelab.net (L. Yang), phifv@hitsz.edu.cn (P. Fournier-Viger), wmt@wmt35.idv.tw (J.-T. Wu), tphong@nuk.edu.tw (T.-P. Hong), slwang@nuk.edu.tw (L.-L. Wang), justin.zhan@unlv.edu (J. Zhan).

http://dx.doi.org/10.1016/j.engappai.2016.07.006
0952-1976/

In the past, Kannimuthu and Premalatha adopted the genetic algorithm approach and developed the high utility pattern extracting using genetic algorithm with ranked mutation using minimum utility threshold ($HUPE_{umu}$-GRAM) to mine HUIs (Kannimuthu and Premalatha, 2014). Another algorithm called $HUPE_{wumu}$-GRAM was also designed to mine HUIs without specified minimum utility threshold (Kannimuthu and Premalatha, 2014). Instead of GAs, Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) is also a bio-inspired and population-based approach for finding the optimal solutions by adopting the velocity to update the particles.

In this paper, a binary PSO-based (BPSO) (Kennedy and Eberhart, 1997) algorithm is designed for mining the HUIs. The key contributions of this paper are described below:

1. Fewer algorithms have been developed to find the HUIs based on evolutionary computation. According to authors' knowledge, this is the first paper to mine the high-utility itemsets based on binary PSO mechanism.
2. A discrete PSO-based algorithm namely HUIM-$BPSO_{sig}$ is thus designed to find the HUIs by integrating the sigmoid updating strategy and TWU model. It first sets the number of discovered 1-HTWUIs as the particle size. This approach can find the potential HUIs based on TWU model, which can be used to reduce the combinational problem in traditional HUIM especially when the number of distinct items in the database or the size of database is very large.
3. Extensive experiments were conducted on real-life datasets to evaluate the performance of the proposed approach. Results showed that the proposed approach can efficiently identify the valid HUIs from a very condense datasets and has better results compared to the state-of-the-art GA-based algorithm.

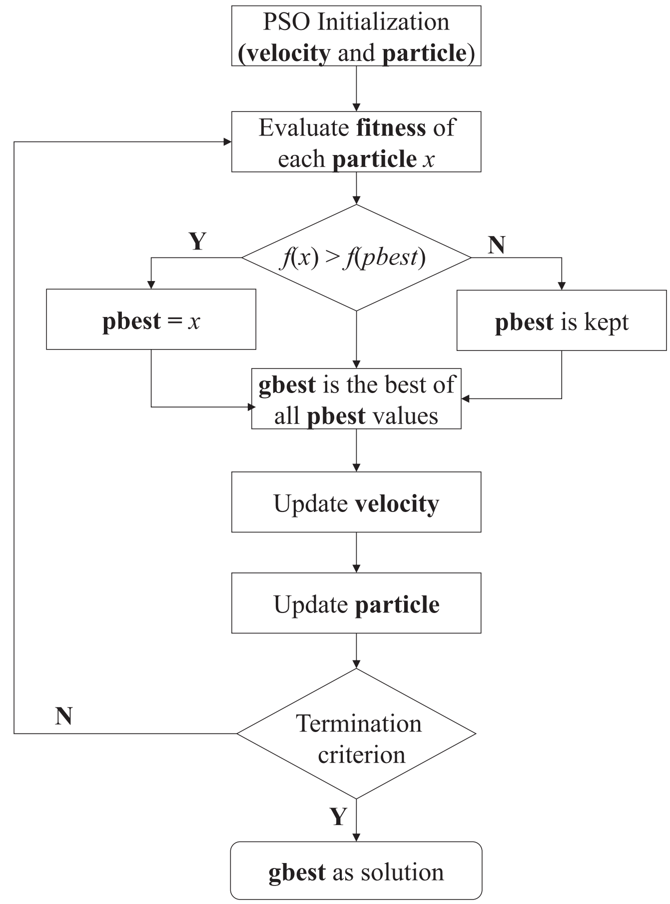

Fig. 1. Flowchart of standard PSO.

2. Related work

In this section, works related to particle swarm optimization (PSO) and high-utility itemset mining (HUIM) are briefly reviewed.

2.1. Particle swarm optimization

In the past, many heuristic algorithms have been facilitated to solve the optimization problems for discovering the necessary information in the evolutionary computation (Cattral et al., 2009; Martnez-Ballesteros et al., 2010). Kennedy and Eberhart first introduced Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) in 1995.

Many individuals (particles) are stochastically generated in the search space of PSO in the evolution process. Each particle is represented as an optimized solution by composing a set of velocities in the evolution process. Instead of GA, each particle has memories to keep its previous best particle (personal best, *pbest*) and its previous best particle by considering its neighborhoods (global best, *gbest*). The flowchart of standard PSO is shown in Fig. 1.

The PSO was originally defined to solve the continuous problems. In the updating process of PSO, the corresponding particle and velocity are described as follows:

$$v_i^j(t+1) = w_1 \times v_i^j(t) + c_1 \times rand() \times (pbest_i^j - x_i^j(t)) + c_2 \times rand() \times (gbest^j - x_i^j(t)). \quad (1)$$

$$x_i^j(t+1) = x_i^j(t) + v_i^j(t). \quad (2)$$

In the Eqs. (1) and (2), t represents current number of iterations; w_1 plays a balancing role between global search and local search; $v_i^j(t)$ is represented the velocity of j-th position in particle i; $x_i^j(t)$ is represented the item of j-th position in particle i; *rand*() is the random number in range of (0, 1); c_1 is the individual factor and c_2 is the social factor, which are usually set as 2. In the past, PSO has been adopted to various real-world applications. Kuo et al. designed an algorithm to mine the association rules (ARs) from the investor's stock purchase behavior by using the designed itemset range (IR) value instead of the specified minimum support and minimum confidence thresholds (Kuo et al., 2011). Pears and Koh presented a feasible method to mine the weighted association rule mining based on PSO (Pears and Koh, 2012). Kennedy and Eberhart also designed a discrete (binary) PSO (BPSO) (Kennedy and Eberhart, 1997) to solve the limitation of continuous PSO. Each particle in BPSO is represented as a set of binary variables. The velocity in the BPSO is updated in terms of probabilities by adopting sigmoid function. Sarath and Ravi also designed a BPSO optimization approach to discover ARs (Sarath and Ravi, 2013). Other applications by adopting PSO to mine the required information are still in progress (Menhas et al., 2011; Nouaouria et al., 2013).