Advances in Engineering Research, volume 208
Proceedings of the International Conference on Innovation in Science and Technology (ICIST 2020)

High-Utility Association Rules Mining Based-on Binary Particle Swarm Optimization

R. Gunawan*
Department of Informatics, Faculty of Science and Technology
Sanata Dharma University
Yogyakarta, Indonesia
*rido@usd.ac.id

E. Winarko, R. Pulungan
Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada
Yogyakarta, Indonesia

***Abstract*—Traditional association rule mining algorithm only generates a set of rules from frequent itemset, the rules obtained cannot generate rules from high-utility itemset. This is because the framework that's being used to obtain rules from traditional association rule is support-confidence while getting high-utility itemset association rules uses the utility-confidence framework. The model for high-utility association rule mining proposed is using particle swarm optimization. The fitness function to get high-utility association rules does not use support-confidence but uses the utility-confidence framework. The association rule mining model of high-utility itemset does not look for high-utility itemset first but together with the high-utility itemset mining process. The high-utility association rule mining using the particle swarm optimization approach has better rule set quality than using the traditional approach, Apriori. Testing with five datasets: chess, connect, mushroom, accident, and foodmart, shows the average utility-confidence obtained using our proposed method is above 88%.**

Keywords—high-utility association rules mining, binary particle swarm optimization, and BPSO approach

I. Introduction

Data mining is a technique used to get hidden knowledge, useful knowledge, and interesting knowledge from large data set. One of the techniques in data mining is association rule mining. The purpose of association rule mining is to obtain transaction patterns, correlation, or association between items. The association rule mining begins with finding frequent itemset that often appears together. The itemset that often appear only pay attention to their existence, not paying attention to quantity of itemset and the weight of the itemset. All items are considered to have the same utility value.

The association rules from high utility itemset cannot be obtained either using classical algorithms or incorporating the computational intelligence paradigm. The existing algorithms only get the collection of high-utility itemset, but not yet in the form of the association rule X → Y. The search for association rules while obtaining frequent itemset can be done using the computational intelligence approach. However, the search for association rules together with high-utility itemset has not been carried out. The purpose of the simultaneously carried out search process is to optimize the association rules obtained. The association rule is important for users to get itemset combinations that match the expected criteria, have high utility value and still get high interestingness (utility-confidence) value.

The main contribution of the paper is an algorithm based on binary particle swarm optimization to obtain objectives high-utility association rules. In terms of objectivity, the minimum value of utility and utility-confidence is no longer determined experimentally according to user preferences. The approach taken is to combine the classical search for association rules with a computational intelligence approach. Another contribution is formulating a fitness function that involves utility and utility-confidence factors to obtain meaningful association rules.

The rest of this paper is organized as follows: related work is briefly reviewed in Section 2. Section 3 provides the definitions. The proposed algorithm for high-utility association rule mining is described in Section 4. Experimental design, result, and analysis are presented in Section 5. The paper is concluded in Section 6.

II. Related Work

In this section, work related to methods for finding association rules from frequent itemset and high-utility itemset using computational intelligence and high-utility association rule mining using deterministic algorithm is briefly reviewed. There are three computational intelligence algorithms used, namely genetic algorithm (GA), particle swarm optimization (PSO), and ant colony optimization (ACO).

GA is used to get association rules without first determining the value of support and confidence, the computation time is faster than searching using classic algorithms and only interesting rules are obtained [1]. The fitness function used is focused on support and confidence, however Vishnoi and Badhe used profit pattern as a fitness function [2].

The search for association rules from frequent itemset using the PSO approach was carried out by Kuo et al. [3], Gupta et al. [4], Sarath and Ravi [5], and Sehrawat and Rohil [6]. There is an increase in computational efficiency and quality of rules compared to using the classic Apriori algorithm and FP-Growth. Agrawal et al. [7] used the PSO binary to get positive and negative association rules, while Kabir et al. [8] used both frequent and infrequent itemset.

Ankita et al. [9] divided the two types of PSO application research for association rule mining. First, association rules are obtained by combining PSO algorithm concepts with classical association algorithms, such as Apriori or FP-Growth [3,10,11]. Second, optimizing the rules that have been obtained from classical algorithms using PSO [12-14].

Kuo and Shih [15] and Kuo et al. [16] proposed the use of ant colony system (ACS) which is a development of ACO to get frequent itemset. Optimization of the association rule mining using ant colony has been carried out, among others [17-20].

In addition to obtaining association rules from frequent itemset, computational intelligence is also used to obtain high-utility itemset. The GA approach is carried out by Kannimuthu and Premalatha [21], the use of BPSO by Lin et al. [22] and Lin et al. [23], ACO by Wu et al. [24] and a bio-inspired algorithm approach conducted by Song and Huang [25]. However, it has not been able to obtain association rules.

Research to obtain association rules preceded by the search for high-utility itemset was carried out by Sahoo et al. [26] which used the mining high utility closed itemset (HUCI) approach and Mai et al. [27] using the lattice approach. However, both of them still need utility and utility-confidence threshold, have to get high-utility itemset first, and not using computational intelligence approach.

III. Definition

The basic concept to get high-utility itemset in detail can be seen in Lin et al. [23], Fournier-Viger et al. 28], and Zida et al. [29]. The definitions given in this section are those related to the definition of utility-confidence. Given a finite set of items $I = \{i_1, i_2, \dots, i_m\}$, an itemset X is a set of distinct k items and X ⊆ I. A transaction database $D = \{T_1, T_2, \dots, T_n\}$ is a set of transactions T_c.

A. *Definition 1*

The following definitions are required to search utility-confidence:

- The utility of itemset X in database D is defined $u(X) = \sum_{T_c \in g(X)} u(X, T_c)$ where g(X) is the set of all transactions containing itemset X.
- The local utility of an item i in an itemset X is defined by $luv(i, X) = \sum_{x \subseteq T_c \wedge T_c \in D} u(i, X)$.
- The utility of an itemset X in another itemset Y such that X ⊆ Y is defined by $luv(X, Y) = \sum_{i \in X \subseteq Y} luv(i, Y)$.
- The utility array of an itemset $X = i_1, i_2, i_3, \dots, i_k$ is defined by $U(X) = u_1, u_2, u_3, \dots, u_k$ where each u_l is $luv(u_l, X); 1 \le l \le k$.
- The utility-confidence of rule R is defined by $onf\, f(R) = \frac{luv(X, X \cup Y)}{u(X)}$, where R: X → Y is an association rule.

IV. Proposed Method

Our proposed method combines the basic BPSO and BPSO-based to obtain high-utility itemset [23]. Our proposed method consists of four processes, namely: pre-processing, particle encoding, fitness evaluation, and updating phases. Our proposed method uses the analogy adopted by Kuo et al. [3] and Sarath and Ravi [5], which does not specify a minimum threshold for utility and utility-confidence. The proposed method is called Association Rules direct with High-Utility Itemset based-on BPSO (ARHU-dBPSO). The pseudo-code of this method is shown in Figure 1. Algorithm (1).

Algorithm 1 Algorithm High-Utility Association Rule Mining based-on BPSO

```
function ARHU-dBPSO(D, pop_size, iterations): ARHUs
    for each transaction T ∈ D do                              ▷ Phase 1: pre-processing
        Compute utility of each item i ∈ T: u(i,T)
        Compute total utility of T: TU(T)
    end for
    Population P ← ∅                                           ▷ Phase 2: initializing and encoding
    for j ← 1 to pop_size do
        T ← Dequeue(D)
        Encode T into particle p⃗
        Generate antecedent pa⃗
        Generate consequent pc⃗
        R ← (pa⃗ → pc⃗)
        Compute utility confidence uconf(R)                    ▷ Definition (1)
        Compute fitness of p⃗: fit(p⃗)     ▷ Phase 3: fitness evaluation     ▷ Equation (1)
        Generate velocity v⃗
        P ← P ∪ { (p⃗, pa⃗, pc⃗, v⃗, uconf(R), fit(p⃗)) }
    end for
    ARHUIs ← Copy(P)
    P(b) ← Copy(P)
    p⃗(g) ← FindBestParticle(P(b))
    for i ← 1 to iterations do                                 ▷ Phase 4: updating
        for each (p⃗, v⃗, fit(p⃗)) ∈ P and corr. (p⃗(b), v⃗(b), fit(p⃗(b))) ∈ P(b) do
            Generate antecedent pa⃗
            Generate consequent pc⃗
            R ← (pa⃗ → pc⃗)
            Compute utility confidence uconf(R)                ▷ Definition (1)
            Update velocity v⃗
            Update particle p⃗
            Compute fitness of p⃗: fit(p⃗)                      ▷ Equation (1)
            ARHUs ← ARHUs ∪ { p⃗ }
            if fit(p⃗) > fit(p⃗(b)) then
                p⃗(b) ← Copy(p⃗)
                p⃗(g) ← FindBestParticle(P(b))
            end if
        end for
    end for
    return ARHUs
end function
```

Fig. 1. Algorithm (1).

The fitness function used is as follows:

$$fit(\vec{p}) = uconf(R) \times u(R) = \frac{luv(X, X \cup Y)}{u(X)} \times u(R) \quad (1)$$

where:

X is antecedent itemset, Y is consequent itemset, $X \neq Y$, X ∪ Y = R.

V. Experiment, Result and Analysis

A. Experiment

The datasets used to test the algorithm are chess, connect, mushroom, accident, and foodmart, as was done by Lin et al. [23]. All datasets that already have a utility value can be downloaded at https://bit.ly/30otNXu with detailed characteristics of each dataset as obtained by Gunawan et al. [30].

The process of association rule mining together with searching for high-utility itemset is analyzed based on twenty rule sets. The population size is twenty and the number of iterations is 10,000. The algorithm performance observed is execution time as well as memory usage. Besides showing the performance of the algorithm, it also shows the results of the rules obtained along with the utility, utility-confidence and fitness value.

Association rule mining by first obtaining the high-utility itemset is called ARHU-BPSO. To test the ARHU-dBPSO proposed algorithm, the results were compared with the ARHU-BPSO. The purpose of testing is to find which algorithm can produce better interestingness values.

B. Result and Analysis

The rule results have been obtained from the five data sets. Table 1 is an example of the results from the Mushroom data set which has the best average *uconf* value among other datasets.

Table 2 shows the measurement results for ARHU-dBPSO. The average utility and utility-confidence are obtained from twenty rules with the highest utility. The rules obtained have a high-utility confidence value, overall above 80% which indicates a high level of interestingness in the results of the rules obtained. Table 3 is a recapitulation of experiments from the five datasets using the ARHU-BPSO approach. ARHU-BPSO gets the entire high-utility itemset first, then looks for the association rules of the high-utility itemset that have been obtained.

Judging from the quality of the resulting association rules, namely from the number of rules obtained, the average utility value and the average utility-confidence, ARHU-dBPSO generally give better results. The average value of utility confidence is quite acceptable because it has a value above 0.8 which means that more than 80% of the user's confidence level in the association rules is obtained. However, ARHU-dBPSO has a slower execution time because there is an additional search process for association rules every time an itemset is obtained and there is a process for calculating utility confidence. Memory usage for both methods has little effect.

TABLE I. Result Set from Dataset Mushroom.

No	Rule set	Utility	uconf	Fitness
1	85 86 90 → 34	525889	1.00	525889
2	85 → 34 36 86 90	503368	0.78	392627
3	34 39 85 90 → 86	422250	1.00	422250
4	86 90 → 34	408489	1.00	408489
5	36 86 → 34 90	401842	0.95	381750
6	39 85 → 34 86	391617	0.96	375952
7	34 36 39 86 → 85 90	387664	0.92	356651
8	34 53 90 → 85 86	368422	1.00	368422
9	34 59 85 86 → 90	364960	0.91	332114
10	34 36 90 → 85	352723	1.00	352723
11	34 90 → 85	351510	1.00	351510
12	34 36 39 → 85 86	350784	1.00	350784
13	36 85 86 → 34 59 90	347682	0.60	208609
14	63 85 86 → 34 90	345766	0.91	314647
15	85 90 → 39 86	345272	0.64	220974
16	34 39 → 86 90	344652	0.88	303294
17	24 34 86 90 → 85	336026	1.00	336026
18	36 53 90 →34 85 86	330297	1.00	330297
19	63 85 → 34 36 86 90	326891	0.75	245168
20	1 34 36 85 90 → 86	326484	1.00	326484

TABLE II. Experiment Results from ARHU-dBPSO.

Dataset	Execution Time (s)	Memory (MB)	Number of rules	Avg Utility	Avg uconf
Chess	1123	474	19544	415247	0.90
Connect	88206	1416	21831	10667984	0.89
Mushroom	3204	641	5010	376629	0.91
Accident	43110	2654	17114	2093085	0.89
Foodmart	7664	1053	40	3164	0.82

TABLE III. Experiment Results from ARHU-BPSO.

Dataset	Execution Time (s)	Memory (MB)	Number of rules	Avg Utility	Avg uconf
Chess	213	466	468	211362	0.65
Connect	187	1241	122	3895500	0.85
Mushroom	28	282	173	89214	0.60
Accident	103	553	389	623716	0.43
Foodmart	31	287	70	2141	0.68

VI. Conclusion and Future Work

High-utility association rules can be obtained using the BPSO approach without the determination of utility and utility confidence. The utility confidence value is quite high, approximately above 80\%. The quality of high-utility association rules also depends on the density of the dataset, too sparse dataset cannot produce good high-utility association rules. If we want the quality of the association rules, the ARHU-dBPSO is more appropriate to use, but if we want better speed, the ARHU-BPSO is more precise.

For future work, we can use other swarm intelligence such as ant colony optimization or bee colony. Further research can also be carried out to get a better fitness function formula, which can be combined with other interestingness formulas besides utility-confidence.

REFERENCES

[1] X. Yan, C. Zhang, and S. Zhang, "Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support," Expert Systems with Applications, vol. 36, no. (2), pp. 3066-3076, 2009.

[2] S.K. Vishnoi and V. Badhe, "Association Rule Mining for Profit Patterns Using Genetic Algorithm," International Journal of Emerging Technology and Advanced Engineering, vol. 4, pp. 855–858, 2014.

[3] R.J. Kuo, C.M. Chao, and Y.T. Chiu, "Application of particle swarm optimization to association rule mining," Applied soft computing, vol. 11, no. (1), pp. 326-336, 2011.

[4] A. Gupta, K.K. Swarnkar, and K. Wadhwani, "Combined economic emission dispatch problem using particle swarm optimization," International Journal of Computer Applications, vol. 49, no. (6), pp. 7628-0695, 2012.

[5] K.N.V.D. Sarath, and V. Ravi, "Association rule mining using binary particle swarm optimization," Engineering Applications of Artificial Intelligence, vol. 26, no. (8), pp. 1832-1840, 2013.

[6] P. Sehrawat and H. Rohil, "Association rule mining using Firefly algorithm," Int. J. Latest Trends Eng. Technol, vol. 3, no. (2), pp. 263-270, 2013.

[7] J. Agrawal, S. Agrawal, A. Singhai, and S. Sharma, "SET-PSO-based approach for mining positive and negative association rules," Knowledge and information systems, vol. 45, no. (2), pp. 453-471, 2015.

[8] M.M.J. Kabir, S. Xu, B.H. Kang, and Z. Zhao, "Association rule mining for both frequent and infrequent items using particle swarm optimization algorithm," International Journal on Computer Science and Engineering, vol. 6, no. (7), p. 221, 2014.

[9] S. Ankita, A. Shikha, A. Jitendra, and S. Sanjeev, "A review on application of particle swarm optimization in association rule mining," Adv Intell Syst Comput., vol. 247, pp. 405–414, 2013.

[10] M. Ykhlef, "A quantum swarm evolutionary algorithm for mining association rules in large databases," Journal of King Saud University-Computer and Information Sciences, vol. 23, no. (1), pp. 1-6, 2011.

[11] M. Nandhini, M. Janani, and S.N. Sivanandham, "Association rule mining using swarm intelligence and domain ontology," In 2012 International Conference on Recent Trends in Information Technology, pp. 537-541, 2012. IEEE.

[12] S. Mishra, D. Mishra, and S.K. Satapathy, "Particle swarm optimization based fuzzy frequent pattern mining from gene expression data," In 2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011), pp. 15-20, 2011. IEEE.

[13] S. Mishra, D. Mishra, and S.K. Satapathy, "Fuzzy frequent pattern mining from gene expression data using dynamic multi-swarm particle swarm optimization," Procedia Technology, vol. 4, pp. 797-801, 2012.

[14] M.J. Abdi and D. Giveki, "Automatic detection of erythemato-squamous diseases using PSO–SVM based on association rules," Engineering Applications of Artificial Intelligence, vol. 26, no. (1), pp. 603-608, 2013.

[15] R.J. Kuo and C.W. Shih, "Association rule mining through the ant colony system for National Health Insurance Research Database in Taiwan," Comput Math with Appl., vol. 54, pp. 1303–1318, 2006.

[16] R.J. Kuo, S.Y. Lin, and C.W. Shih, "Mining association rules through integration of clustering analysis and ant colony system for health insurance database in Taiwan," Expert Systems with Applications, vol. 33, no. (3), pp. 794-808, 2007.

[17] V. Mangat, "Swarm intelligence based technique for rule mining in the medical domain," International Journal of Computer Applications, vol. 4, no. (1), pp. 19-24, 2010.

[18] B. Patel, V.K. Chaudhari, R.K. Karan, and Y.K. Rana, "Optimization of association rule mining apriori algorithm using ACO," International Journal of Soft Computing and Engineering, vol. 1, no. (1), pp. 24-26, 2011.

[19] A.S. Sadh and N. Shukla, "Apriori and ant colony optimization of association rules," International Journal of Advanced Computer Research, vol. 3, no. (2), p. 35, 2013.

[20] S. Aggarwal and B. Rani, "Optimization of association rule mining process using Apriori and Ant Colony Optimization algorithm," International Journal of Current Engineering and Technology, vol. 3, no. (2), p. 623, 2013.

[21] S. Kannimuthu and K. Premalatha, "Discovery of High Utility Itemsets Using Genetic Algorithm," Int J Eng Technol., vol. 5, pp. 4866–4880, 2013.

[22] J.C.W. Lin, L. Yang, P. Fournier-Viger, J.M.T. Wu, T.P. Hong, L.S.L. Wang, and J. Zhan, "Mining high-utility itemsets based on particle swarm optimization," Engineering Applications of Artificial Intelligence, vol. 55, pp. 320-330, 2016.

[23] J.C.W. Lin, L. Yang, P. Fournier-Viger, T.P. Hong, and M. Voznak, "A binary PSO approach to mine high-utility itemsets," Soft Computing, vol. 21, no. (17), pp. 5103-5121, 2017.

[24] J.M.T. Wu, J. Zhan, and J.C.W. Lin, "An ACO-based approach to mine high-utility itemsets," Knowledge-Based Systems, vol. 116, pp. 102-113, 2017.

[25] W. Song and C. Huang, "Mining high utility itemsets using bio-inspired algorithms: A diverse optimal value framework," IEEE Access, vol. 6, pp. 19568-19582, 2018.

[26] J. Sahoo, A.K. Das, and A. Goswami, "An efficient approach for mining association rules from high utility itemsets," Expert systems with Applications, vol. 42, no. (13), pp. 5754-5778, 2015.

[27] T. Mai, B. Vo, and L.T. Nguyen, "A lattice-based approach for mining high utility association rules," Information Sciences, vol. 399, pp. 81-97, 2017.

[28] P. Fournier-Viger, C.W. Wu, S. Zida, and V.S. Tseng, "FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning," In International symposium on methodologies for intelligent systems, pp. 83-92, 2014. Springer, Cham.

[29] S. Zida, P. Fournier-Viger, J.C.W. Lin, C.W. Wu, and V.S. Tseng, "EFIM: a highly efficient algorithm for high-utility itemset mining," In Mexican international conference on artificial intelligence, pp. 530-546, 2015. Springer, Cham.

[30] R. Gunawan, E. Winarko, and R. Pulungan, "A BPSO-based method for high-utility itemset mining without minimum utility threshold," Knowledge-Based Systems, vol. 190, p. 105164, 2020.