An Evolutionary Algorithm to Mine High-Utility Itemsets

Jerry Chun-Wei LIN[1], *Lu YANG*[1], *Philippe FOURNIER-VIGER*[2], *Jaroslav FRNDA*[3],
Lukas SEVCIK[3], *Miroslav VOZNAK*[3]

[1]School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School,
Xiaoyuan Rd, Shenzhen, Guangdong, China
[2]Department of Computer Science, University of Moncton, 18 Antonine-Maillet Ave, Moncton,
NB E1A 3E9, Canada
[3]Department of Telecommunications, Faculty of Electrical Engineering and Computer Science,
VSB–Technical University of Ostrava, 17. listopadu 15, 708 00 Ostrava-Poruba, Czech Republic

jerrylin@ieee.org, luyang@ikelab.net, philippe.fournier-viger@umoncton.ca, jaroslav.frnda@vsb.cz,
lukas.sevcik.st1@vsb.cz, miroslav.voznak@vsb.cz

DOI: 10.15598/aeee.v13i4.1474

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) of association rules (ARs). In this paper, an evolutionary algorithm is presented to efficiently mine high-utility itemsets (HUIs) based on the binary particle swarm optimization. A maximal pattern (MP)-tree strcutrue is further designed to solve the combinational problem in the evolution process. Substantial experiments on real-life datasets show that the proposed binary PSO-based algorithm has better results compared to the state-of-the-art GA-based algorithm.*

Keywords

Binary PSO, evolutionary computation, particle swarm optimization, utility-itemset mining, tree.

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 the occurrence frequency of items is no less than minimum support threshold or the confidence of items is no less than minimum confidence threshold [2], [5]. Since only the occurrence frequencies of itemsets are discovered whether in FIM or ARM, it is insufficient to identify the high profit item/sets especially when the itemsets are rarely appeared but have high profit values. To solve the limitation of FIM or ARM, high-utility itemset mining (HUIM) [20], [21], [22] was designed to discover the "useful" and "profitable" itemsets from the quantitative databases. Chan et al. [4] first mentioned utility mining problem instead of FIM. Yao et al. [20] concerned the quantity of items as the internal utility and the unit profit of items as the external utility to discover the HUIs. Liu et al. designed the two-phase (TWU) model and developed the transaction-weighted downward closure (TWDC) property for mining HUIs [16]. Lin et al. developed the condensed high-utility pattern (HUP)-tree and related algorithm for discovering HUIs [14]. Lan et al. designed the mining algorithm based on index-projection mechanism and developed the pruning strategy to efficiently mine the HUIs [10]. An efficient list-based algorithm (HUI-Miner) was also proposed to mine the HUIs without candidate generation [15]. Fournier-Viger et al. adopted the HUI-Miner algorithm and presented a FHM algorithm for mining HUIs based on the constructed structure of 2-itemsets [7], which is the state-of-the-art algorithm in traditional HUIM.

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 [3]. The genetic algorithm (GA) [11] 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. In the past, Kannimuthu and Premalatha adopted the genetic algorithm and developed the high utility pattern extracting approach using genetic algorithm with ranked mutation using minimum utility threshold (HUPE$_{\mathrm{umu}}$-GRAM) to mine HUIs [9]. Instead of GAs, Particle Swarm Optimization (PSO) [12] 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) [13] algorithm with an improved maximal pattern (MP)-tree structure is designed for mining the HUIs. The major contributions of this paper are described below:

- We have presented a discrete PSO-based algorithm to mine high-utility itemsets (HUIs) based on the sigmoid updating strategy. The TWU model is also used in the designed algorithm to reduce the size of each particle, thus speeding up the the combinational phase for revealing the HUIs in the evolution process.
- An efficient maximal pattern (MP)-tree structure is developed to reduce the multiple database scans by early pruning the invalid combinations of the particles.
- Extensive experiments showed that the proposed approach has better results compared to the state-of-the-art GA-based algorithm [9] whether in runtime or the number of discovered HUIs.

2. Related Work

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 [3]. The simple genetic algorithm (sGA) [11] is a fundamental search technique to find the feasible and optimal solution in a limit amount of time, which was inspired by the Darwinian principles. Many variants of GAs have been extensively studied and applied to a wide range of optimization problems [9], [19].

Kennedy and Eberhart first introduced Swarm Optimization (PSO) [12] in 1995, which was inspired by the flocking behavior of birds to solve the optimization problems. 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 PSO was originally defined to solve the continues valued spaces. In real-world situations, many problems are set as the discrete variable spaces such as scheduling and routing problems. Kennedy and Eberhart then also designed a discrete (binary) PSO (BPSO) [13] to solve the limitation of continuous PSO. Sarath and Ravi also designed a BPSO optimization approach to discover ARs [18]. Other applications by adopting PSO to mine the required information are still processed in progress [8], [17].

2.2. High-Utility Itemset Mining

High-utility itemset mining (HUIM) is an emerging topic in recent decades. Chan et al. first developed a mining framework for discovering the top-k closed utility patterns [4]. Yao et al. then designed an approach to discover the profitable itemsets by considering both the purchase quantity (also considered as internal utility) and profit (also considered as external utility) of items to reveal HUIs [20], [21]. Liu et al. then developed a two-phase (TWU) model and designed the transaction-weighted downward closure (TWDC) property to early prune the unpromising HUIs for discovering HUIs [16]. Erwin et al. presented a parallel projection scheme and used the disk storage for handling the large-scale databases and the designed approach performs well whether in dense or sparse databases [6]. Lan presented a projected-based approach to level-wise reduce the size of the processed databases for mining HUIs [10]. Fournier-Viger et al. [7] proposed a FHM algorithm to keep the relationships of 2-itemsets, thus reducing the computations for mining the HUIs based on HUI-Miner algorithm [15], which is state-of-the-art algorithm in traditional HUIM.

Instead of traditional HUIM, Kannimuthua and Premalatha first designed the GA-based algorithm to mine HUIs with the ranked mutation [9], which is the state-of-the-art algorithm for mining HUIS in the evolution process. In their approach, it is not easier to find the initial 1-HTWUIs as the chromosome to find HUIs. A very huge computation is necessary to initially set the appropriate chromosomes. Moreover, some specific parameters are required in the evolution process of GAs. In this paper, a binary PSO-based algorithm with an efficient maximal pattern (MP)-tree structure are designed to avoid the invalid combinations, thus improving the efficiency for discovering HUIs.

3. Proposed Algorithm

In the designed binary PSO (BPSO)-based model [13], it consists of pre-processing, particle encoding, fitness evaluation and the updating processes to mine HUIs. In the first pre-processing process, the high-transaction-weighted utilization 1-itemsets (1-HTWUIs) are first discovered based on TWU model [16]. The valid combinations of the itesmets in the database is also compressed as a maximal pattern (MP)-tree structure, which can be used to avoid the invalid generations of the particles in the updating process. It uses the OR and NOR operators to determine whether the combined itemsets can generate the valid combinations in the database. A simple MP-tree structure is shown in Fig. 1.

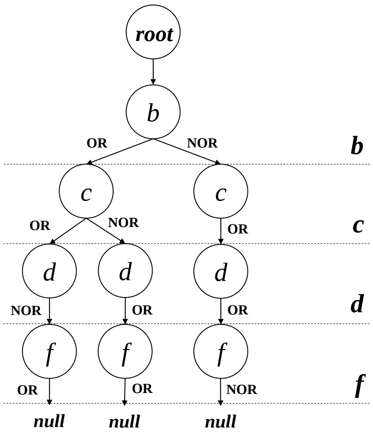


Fig. 1: The developed MP-tree structure.

In the second particle encoding process, the items of 1-HTWUIs are sorted in their *alphabetic-ascending* order corresponding to the j-th position of a particle. The particle is thus encoded as the set of binary variables corresponding to the sorted order of 1-HTWUIs. In the fitness evaluation, the particles are then evaluated to find the *pbest* and *gbest* particles in the evolution process. For the last updating process, the particles are correspondingly updated by velocities, *pbest*, *gbest*, and the sigmoid function. The MP-tree structure is used here to generate only the valid combinations of the particles. If the fitness value of the particle is no less than the minimum utility count (the minimum utility threshold multiplied by the total utility), it is concerned as a HUI and put into the set of HUIs. In the designed algorithm, the fitness function is the same as the traditional HUIM as:

$$fitness(p_i) = u(X), \tag{1}$$

in which X is the union of j-th positions in the particle p_i. This iteration is repeated until the termination criteria is achieved. After that, the set of HUIs is discovered. Details of the proposed algorithm is shown below.

Algorithm 1: Proposed algorithm

```
Input: $D$, a quantitative database; $ptable$, a profit
       table, $\delta$, the minimum utility threshold;
       $M$, the number of particles of each
       iteration.
Output: $HUIs$, a set of high-utility itemsets.
1  find 1-HTWUIs;
2  set $m$ = |1-HTWUIs|;
3  initialize $p(t)$ = either 1 or 0;
4  initialize $v(t) = rand()$;
5  initialize $pbest(t)$ and $gbest(t)$;
6  while termination criteria is not reached do
7      update the $v(t+1)$ of $M$ particles;
8      update the $p(t+1)$ of $M$ particles;
9      for $i \leftarrow 1, M$ particles do
10         if $fitness(p_i(t+1)) \geq \delta$ then
11             $HUIs \leftarrow GItems(p_i(t+1)) \cup HUIs$;
12         find $pbest(t+1)$ of each $M$ particle;
13         find $gbest(t+1)$ among $M$;
14     set $t \leftarrow t+1$;
15 return HUIs;
```

In the designed algorithm, the set of 1-HTWUIs is thus discovered based on two-phase model [16] (Line 1). The number of 1-HTWUIs is thus used as the particle size, which can reduce the combinational problem in traditional HUIM [20], [21] (Line 2). The particles are initialized by either 0 and 1 based on the binary PSO approach [13] (Line 3). The velocities of particles are randomized in the range of (0, 1) (Line 4). After that, the *pbest* and *gbest* are also initialized in the same way (Line 5). The velocities of particles are thus updated according to the traditional PSO approach [12]. The particles are then, updated based on the sigmoid function used in [13] and the designed MP-tree structure (Line 8). In this procedure, the invalid particles (not existing in the original database) cannot be generated, which can greatly reduce the computations of multiple database scans. The fitness values of the updated particles are then determined to find the actual HUIs (Lines 10 to 11). After that, the pbest and gbest of the current particles are then updated for next iteration (Lines 12 to 13). This procedure is repeatedly progressed until the termination criteria is achieved (Lines 6 to 14). After that, the set of the discovered HUIs is outputted (Line 15).

4. Experimental Results

Substantial experiments were conducted to verify the effectiveness and efficiency of the proposed algorithm compared to the state-of-the-art evolutionary HUPE$_{\text{umu}}$-GRAM algorithm [9]. The algorithms in the experiments were implemented in C++ language, performing on a PC with an Intel Core2 i3-4160 CPU and 4 GB of RAM, running the 64-bit Microsoft Windows 7 operating system. Two real-world datasets called chess [1] and mushroom [1] are used in the experiments. A simulation model [16] was developed to generate the quantities and profit values of items in transactions for all datasets. A log-normal distribution was used to randomly assign quantities in the [1,5] interval, and item profit values in the [1,1000] interval. In the conducted experiments for mining HUIs, the performed algorithms are all performed for 10,000 iterations and the population size is set as 20. The experiments are executed five times and the results are the average values of them. The algorithms are then compared in terms of execution time and number of HUIs. Note that the runtime of the designed algorithm includes the construction time of the designed MP-tree structure. The parameters of the designed algorithm are respectively set as: $c_1 = c_2$ $(= 2)$ and $w(= 0.9)$.

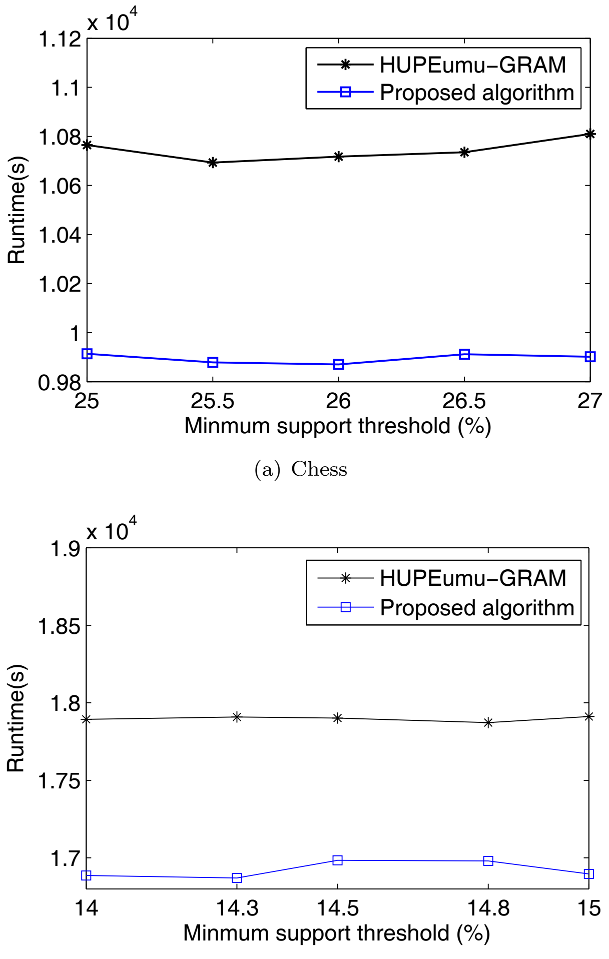
4.1. Runtime

In the conducted experiments of runtime in two datasets, the state-of-the-art evolutionary HUPE$_{\text{umu}}$-GRAM algorithm of HUIM is compared to the designed algorithm. The TWU model is an algorithm in traditional HUIM for mining HUIs, which is insufficient and inappropriate to compare with designed algorithm in term of runtime. The results are shown in Fig. 2.

From Fig. 2, it can be seen that the proposed algorithm has better performance in terms of execution time compared to the HUPE$_{\text{umu}}$-GRAM algorithm w.r.t. different minimum thresholds. The reason is that the proposed algorithm only generates the valid combinations of itemsets existing in the database, which can greatly avoid the combinational problem in the evolution process. The HUPE$_{\text{umu}}$-GRAM generates, however, many unpromising itemsets. The multiple database scans are required to determine the invalid itemsets. This process takes very huge computations and non-HUIs can thus be generated from the database. When the minimum utility threshold is set lower, more high transaction-weighted utilization 1-itemsets (1-HTWUIs) are discovered; it takes more combinations to find the promising HUIs in the evolution process. Overall, the proposed algorithm can solve the combinational problem in the evolution process compared to the state-of-the-art HUPE$_{\text{umu}}$-GRAM algorithm.

4.2. Number of HUIs

In this section, the number of HUIs is evaluated to show the performance of the compared algorithms. The state-of-the-art two-phase [16] algorithm is used to discover the actual and complete HUIs from the quantitative databases. Experiments are conducted and shown in Fig. 3.

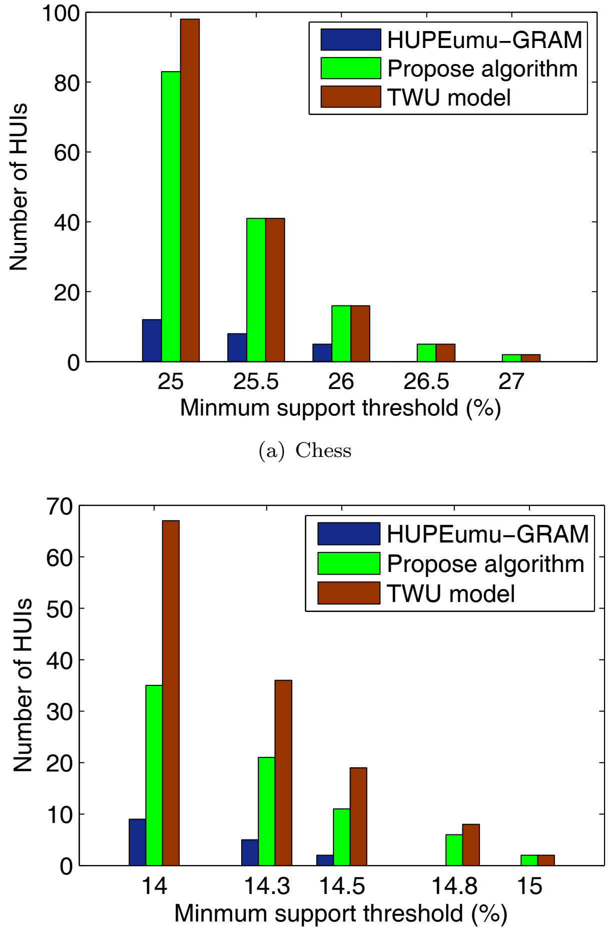


(a) Chess

(b) Mushroom

Fig. 2: Runtime of compared algorithms.

From Fig. 3, it can be seen that the proposed algorithm can generate nearly the same number of HUIs compared to the state-of-the-art two-phase algorithm especially when the minimum utility threshold is set higher. The reason is that the size of a particle is associated with the number of 1-HTWUIs, less computations are required when the minimum utility threshold is set higher. For more condense dataset such as chess or mushroom datasets, the number of 1-HTWUIs is close to the number of discovered HUIs under higher minimum utility threshold; the number of generated HUIs of the designed algorithm is close to the traditional way for mining the complete HUIs. The designed algorithm produce less number of the HUIs compared to the TWU model under lower utility threshold. This is reasonable since the proposed algorithm is an evolutionary way to learn and discover the HUIs; when the minimum utility threshold



(a) Chess

(b) Mushroom

Fig. 3: Number of HUIs of the compared algorithms.

is set lower, more high transaction-weighted utilization 1-itemsets (1-HTWUIs) are discovered but it is not easy to produce the same number of HUIs in the evolution process. However, the proposed algorithm outperforms the state-of-the-art HUPE$_{\text{umu}}$-GRAM algorithm, which can be found in Fig. 3(a) and Fig. 3(b). This is reasonable since the GA-based algorithm is barely to find the promising HUIs but the designed algorithm can efficiently find the valid HUIs by concerning both the local and global results.

5. Conclusions

In this paper, we first propose a binary particle swarm optimization (BPSO)-based algorithm to efficiently mine high-utility itemsets (HUIs). The contributions of this paper are as follows. First, we adopted the discrete mechanism and set the size of each particle as the number of discovered high-transaction-weighted utilization 1-itemsets (1-HTWUIs) based on TWU model. This approach can greatly reduce the combinational problem in the evolution process. Second, the sigmoid function is adopted and a maximal pattern (MP)-tree structure is further designed to produce the valid combinations of itemsets, which can greatly reduce the computations of multiple database scans. Third, from the conducted experiments, the proposed approach has better results compared to the state-of-the-art GA-based algorithm for mining HUIs in based on the evolution process in terms of execution time and the ability to discovered the complete of HUIs.

Acknowledgment

This publication has been created within the project Support of VSB–Technical University of Ostrava activities with China with financial support from the Moravian-Silesian Region and and partially was supported by the grant SGS reg. no. SP2015/82 conducted at VSB–Technical University of Ostrava, Czech Republic, and was partially supported by the Shenzhen Peacock Project, China, under grant KQC201109020055A, by the Tencent Project under grant CCF-TencentRAGR20140114, by the Natural Scientific Research Innovation Foundation in Harbin Institute of Technology under grant HIT.NSRIF.2014100, and by the Shenzhen Strategic Emerging Industries Program under grant ZDSY20120613125016389.

References

[1] Frequent itemset mining dataset repository. *FIMI* [online]. 2012. Available at: `http://fimi.ua.ac.be/data/`.

[2] AGRAWAL, R. and R. SRIKANT. Fast algorithms for mining association rules in large databases. In: *The International Conference on Very Large Data Bases*. San Francisco: Morgan Kaufmann Publishers Inc., 1994, pp. 487–499. ISBN 1-55860-153-8.

[3] CATTRAL, R., F. OPPACHER and K. J. L. GRAHAM. Techniques for evolutionary rule discovery in data mining. In: *IEEE Congress on Evolutionary Computation*. New Jersey: IEEE, 2009, pp. 1737–1744. ISBN 978-1-4244-2958-5. DOI: 10.1109/cec.2009.4983151.

[4] CHAN, R., Q. YANG and Y. D. SHEN. Minging high utility itemsets. In: *The Third IEEE International Conference on Data Mining*. Washington: IEEE, 2003, pp. 19–26. ISBN 0-7695-1978-4. DOI: 10.1109/ICDM.2003.1250893.

[5] CHEN, M. S., J. HAN and P. S. YU. Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*. 1996, vol. 8, iss. 6, pp. 866–883. ISSN 1041-4347. DOI: 10.1109/69.553155.

[6] ERWIN, A., R. P. GOPALAN and N. R. ACHUTHAN. Efficient Mining of High Utility Itemsets from Large Datasets. In: *12th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2008)*. Berlin: Springer, 2008, pp. 554–561. ISBN 978-3-540-68125-0. DOI: 10.1007/978-3-540-68125-0_50.

[7] FOURNIER-VIGER, P., C. W. WU, S. ZIDA and V. S. TSENG. FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning. In: *21st International Symposium (ISMIS 2014)*. Roskilde: Springer, 2014, pp. 83–92. ISBN 978-3-319-08325-4. DOI: 10.1007/978-3-319-08326-1_9.

[8] MENHAS, M. I., M. FEI, L. WANG and X. FU. A Novel Hybrid Binary PSO Algorithm. In: *Second International Conference (ICSI 2011)*. Berlin: Springer, 2011, pp. 93–100. ISBN 978-3-642-21514-8. DOI: 10.1007/978-3-642-21515-5_12.

[9] KANNIMUTHU, S. and K. PREMALATHA. Discovery of high utility itemsets using genetic algorithm with ranked mutation. *Applied Artificial Intelligence*. 2014, vol. 28, iss. 4, pp. 337–359. ISSN 0883-9514. DOI: 10.1080/08839514.2014.891839.

[10] LAN, G. C., T. P. HONG and V. S. TSENG. An efficient projection-based indexing approach for mining high utility itemsets. *Knowledge and Information Systems*. 2013, vol. 38, iss. 1, pp. 85–107. ISSN 0219-1377. DOI: 10.1007/s10115-012-0492-y.

[11] HOLLAND, J. *Adaptation in Natural and Artificial Systems*. 1st ed. Cambridge: MIT Press, 1992. ISBN 978-0262082136.

[12] KENNEDY, J. and R. EBERHART. Particle swarm optimization. In: *IEEE International Conference on Neural Networks*. Perth: IEEE, 1995, pp. 1942–1948. ISBN 0-7803-2768-3. DOI: 10.1109/ICNN.1995.488968.

[13] KENNEDY, J. and R. EBERHART. A discrete binary version of particle swarm algorithm. In: *IEEE International Conference on Systems, Man, and Cybernetics*. Orlando: IEEE, 1997, pp. 4104–4108. ISBN 0-7803-4053-1. DOI: 10.1109/ICSMC.1997.637339.

[14] LIN, C. W., T. P. HONG and W. H. LU. An effective tree structure for mining high utility itemsets. *Expert Systems with Applications*. 2011, vol. 38, iss. 6, pp. 7419–7424. ISSN 0957-4174. DOI: 10.1016/j.eswa.2010.12.082.

[15] LIU, M. and J. QU. Mining high utility itemsets without candidate generation. In: *ACM International Conference on Information and Knowledge Management*. New York: ACM, 2012, pp. 55–64. ISBN 978-1-4503-1156-4. DOI: 10.1145/2396761.2396773.

[16] LIU, Y., W. K. LIAO and A. CHOUDHARY A two-phase algorithm for fast discovery of high utility itemsets. In: *9th Pacific-Asia Conference (PAKDD 2005)*. Berlin: Springer, 2005, pp. 689–695. ISBN 978-3-540-31935-1. DOI: 10.1007/11430919_79.

[17] NOUAOURIA, N., M. BOUKADOUMA and R. PROULX. Particle swarm classification: A survey and positioning. *Pattern Recognition*. 2013, vol. 46, iss. 7, pp. 2028–2044. ISSN 0031-3203. DOI: 10.1016/j.patcog.2012.12.011.

[18] SARATH, K. N. V. D. and V. RAVI. Association rule mining using binary particle swarm optimization. *Engineering Applications of Artificial Intelligence*. 2013, vol. 26, iss. 8, pp. 1832–1840. ISSN 0952-1976. DOI: 10.1016/10.1016/j.engappai.2013.06.003.

[19] SALLEB-AOUISSI, A., C. VRAIN and C. NORTET. QuantMiner: a genetic algorithm for mining quantitative association rules. *Journal of Machine Learning Research*. 2013, vol. 14, iss. 1, pp. 1035–1040. ISSN 1532-4435.

[20] YAO, H., H. J. HAMILTON and C. J. BUTZ. A foundational approach to mining itemset utilities from databases. In: *The 2004 SIAM International Conference on Data Mining*. Lake Buena Vista: SIAM, 2004, pp. 211–225. ISBN 978-0-89871-568-2 DOI: 10.1137/1.9781611972740.51.

[21] YAO, H. and H. J. HAMILTON. Mining itemset utilities from transaction databases. *Data & Knowledge Engineering*. 2006, vol. 59, iss. 3, pp. 603–626. ISSN 0169-023X. DOI: 10.1016/j.datak.2005.10.004.

[22] YEN, S. J. and Y. S. LEE. Mining high utility quantitative association rules. In: *Data Warehousing and Knowledge Discovery*. Berlin: Springer, 2007, pp. 283–292. ISSN 0302-9743. DOI: 10.1007/978-3-540-74553-2_26.

About Authors

Jerry Chun-Wei LIN received Ph.D. degree in Dept. of Computer Science and Information Engineering in 2010 from National Cheng Kung University, Tainan, Taiwan. He is currently working as an assistant professor at School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, China. He has published around 100 research papers in referred journals and international conferences. His interests include data mining, soft computing, privacy preserving data mining and security, social network and cloud computing.

Lu YANG is a master student studying in School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, China. Her research interests include data mining and evolutionary computation.

Philippe FOURNIER-VIGER is an assistant-professor at University of Moncton, Canada. He received a Ph.D. in Cognitive Computer Science at the University of Quebec in Montreal (2010). He has published more than 75 research papers in refereed international conferences and journals. He is the founder of the popular SPMF open-source data mining library.

Jaroslav FRNDA was born in 1989 in Martin, Slovakia. He received his M.Sc. from the VSB–Technical University of Ostrava, Department of Telecommunications, in 2013, and now he is continuing his Ph.D. study at the same place. His research interests include Quality of Triple play services and its objective evaluation.

Lukas SEVCIK was born in 1989 in Cadca, Slovakia. He received his M.Sc in Informatics from the Faculty of Management Science and Informatics, University of Zilina, in 2013. Now, he is continuing his Ph.D. study at the Department of Telecommunications, VSB–Technical University of Ostrava. His research interests include Quality of Triple play services and IP networks.

Miroslav VOZNAK is an associate professor at the Department of Telecommunications, VSB–Technical University of Ostrava. He received his Ph.D. in telecommunications, and his dissertation thesis was entitled “Voice traffic optimization with regard to speech quality in networks with VoIP technology” in 2002. Topics of his research interests are Next Generation Networks, IP telephony, speech quality and network security.