Cluster Comput
DOI 10.1007/s10586-017-0894-4

EPACO: a novel ant colony optimization for emerging patterns based classification

Zulfiqar Ali[1] · **Waseem Shahzad**[1]

Received: 27 October 2016 / Revised: 17 April 2017 / Accepted: 28 April 2017

Abstract In this paper, a novel approach for discovering emerging patterns has been proposed. Majority of the existing algorithms for the discovery of emerging patterns are tree-based which involve growth and shrinking of trees for this purpose. These algorithms follow greedy search approach for discovery of emerging patterns. The proposed approach utilizes the diversity of ant colony optimization and avoids complexity and greedy search of tree-based algorithms for discovery of emerging patterns. The experiments show that the proposed approach provides higher accuracy than existing state of the art classifiers as well as emerging pattern-based classifiers.

Keywords Emerging patterns · Patterns discovery · Data mining · Classification · Ant colony optimization

1 Introduction

Classification is one of the mostly used techniques in data mining and knowledge discovery [1]. Classification is a type of supervised learning based approach. In literature, supervised learning based classification are proposed in abundance i.e. artificial neural network (ANN) [2,3], support vector machine (SVM) [4–6] KNN [7], decision trees [8,9] and genetic algorithm (GA) [10–13]. The hybrid approaches with the combination of ANN, SVM and GA are exploited for the data discovery and classification tasks [14–16]. Emerging pattern (EP) based classification, proposed by Dong and Li [17], is relatively new technique use in different classification problems. Emerging patterns is defined as instance whose membership to a particular class label is abruptly changed. Emerging Patterns are used in situation where multi attribute contrasts between two classes of data is being to be handle. Furthermore, magnitude of change for a particular instance from one class to another class is measured, called growth rate. Emerging Pattern is more promising to deal with noise and outlier in data as compared to other classification techniques [18]. Emerging Patterns is further elaborated with the help of following definitions.

Suppose, $Obj = \{a_1, a_2, a_3, \ldots\}$ be a data object following the schema $\{A_1, A_2, A_3, \ldots A_n\}$. Where A_1, A_2, A_3 $\ldots An$, are called attributes of object and a_1, a_2, $a_3 \ldots a_n$ are values associated to these attributes. Moreover, corresponding pair $(A_{1,a1})$ is known as item. If collection of allowed items in a dataset D is denoted by $\boldsymbol{I}$ then item sets are subsets of $\boldsymbol{I}$. Here we can say an instance Y contains an item set $X \subseteq Y$.

Definition 1 (*Support*) Support S_D for an item set p from given dataset D is defined as by Eq. (1).

$$S_D(p) = \frac{count\,D(p)}{|D|} \tag{1}$$

where countD(p) represents the number of instances in dataset D containing p and $|D|$ is total number of instances in dataset D.

Definition 2 (*Growth Rate*) For given two datasets D_1 and D_2 belonging to different classes. Let support of the itemset X in the dataset Di is denoted by $Si(p)$. The growth ratio of

✉ Zulfiqar Ali
zulfiqar.ali@nu.edu.pk

Waseem Shahzad
waseem.shahzad@nu.edu.pk

1 National University of Computer and Emerging Sciences, Islamabad, Pakistan

Published online: 18 May 2017

an itemset p from $D1$ *to* $D2$, $grD1 \rightarrow D2(p)$, is defined as in the Eq. (2).

$$\mathbf{gr}_{\mathrm{D1}\rightarrow\mathrm{D2}}(\boldsymbol{p}) = \begin{cases} \mathbf{0} & \textbf{if } \mathbf{S1}(\boldsymbol{p}) = \mathbf{0} \textbf{ and } \mathbf{S2}(\boldsymbol{p}) = \mathbf{0} \\ \infty & \textbf{if } \mathbf{S1}(\boldsymbol{p}) = \mathbf{0} \textbf{ and } \mathbf{S2}(\boldsymbol{p}) \neq \mathbf{0} \\ \frac{\mathbf{S2}(\boldsymbol{p})}{\mathbf{S1}(\boldsymbol{p})} & \textbf{otherwise} \end{cases} \tag{2}$$

The value of growth ratio will be equal to zero, if $S1(p) = 0$ and $S2(p) = 0$. If value of s1 (p) is equal to zero and s2 (p) is nonzero then the value of growth ratio is infinity. In the third case the growth rate will be equal to the ratio of supports in both datasets. Here we use interchangeable growth rate and growth ratio for the same purpose.

Definition 3 (*Emerging Pattern*) An itemset p is said to be ρ-emerging pattern or simply Emerging Pattern, for a given growth ratio threshold $\rho > 1$, from $D1$ to $D2$ if $\mathrm{gr}_{\mathrm{D1}} \rightarrow \mathrm{D2}(p) \geq \rho$.

In Ref. [19] Li et al. proposed a concept of jumping emerging patterns which is defined with definition 4.

Definition 4 (*Jumping Emerging Pattern*) An itemset X is a *Jumping Emerging Pattern (JEP)* from D_1 to D_2 if $S_{D1}(p) = 0$ and $S_{D2}(p) > 0$.

Definition 5 (*Strong Jumping Emerging Pattern*) An item set X is a *Strong Jumping Emerging Pattern (SJEP)* from D_1 to D_2 with given threshold ($\varepsilon > 0$) as a minimum support and followed by these conditions:

1. $\mathrm{S}_{\mathrm{D1}}(\mathrm{X}) = 0$ and $\mathrm{S}_{\mathrm{D2}}(\mathrm{X}) \geq \varepsilon$ and
2. Any proper subset of X does not satisfy condition 1.

Suppose here a set of class labels are $\mathrm{C} = \{\mathrm{c}_1, \mathrm{c}_2 \ldots \mathrm{c}_\mathrm{k}\}$. For each object *obj*, there exists a class label $Cobj \in C$ associated with it, in a training dataset. A function from attributes $\{\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \ldots \mathrm{A}_\mathrm{n}\}$, to class label $\{\mathrm{c}_1, \mathrm{c}_2 \ldots \mathrm{c}_\mathrm{k}\}$, that assigns class labels to unseen examples, is known as classifier.

The Contrast Pattern Tree (CP-tree) based algorithms [20,21] are deterministic and provide less diversity because they follow greedy approach. In tree-based pattern mining algorithms first, the tree is constructed and expended then it is contracted to discover emerging patterns. The tree based emerging pattern discovering approaches are more complex due to the nature of greedy search approach. Consequently, the discovery of emerging patterns is computationally expensive. Furthermore, all these factors contribute to lower classification accuracy for tree based classifiers.

In this paper bio-inspired ant colony optimization (ACO) [22] technique is exploited for the discovery of emerging patterns. ACO is employed to discover high quality emerging patterns in comparatively complex classification problems. Nevertheless, the inherent characteristics of ACO are utilized to built a classifier that performs better as compare to other classifiers i.e. tree based, SVM and statistical. Moreover, the proposed approach, EPACO is provided diversity to avoid exhaustive and greedy search to achieve the same or even better accuracy. The comparative analysis of EPACO and the other state of the art classifiers show that the proposed approach is more promising to discover emerging patterns.

Following are the main contributions of this article in the field of knowledge discovery and classification.

1. *Ant colony optimization (ACO) is first time exploited for the discovery of emerging patterns according to our best of knowledge.*
2. *The proposed EPACO algorithm is indented to mitigate the short comings of tree-based approach, to provide computationally better classifier.*
3. *Provide a comparative analysis of proposed approach with existing classifiers in terms of classification accuracy.*

The rest of the paper is organized as follows; Sect. 2 provides the related work. Section 3 introduces Ant Colony Optimization for classification and application of ACO for associative classification. Section 4 describes the new proposed EPACO approach. Section 5 presents the experimental results and discussion. Conclusions and future directions are elaborated in the last section.

2 Related work

In contemporary literature, various techniques have been proposed for the discovery of emerging patterns. Mostly, these techniques exploit tree based approach with variation to solve emerging patterns classification problems. This section presents literature related to the discovery of emerging patterns and classification. A tree based technique, Constraint-based Emerging Pattern Miner (ConsEPMiner), proposed by Xiuzhen Zhang et al. in Ref. [23]. The inspiration for this approach is Dense-Miner proposed for mining of association rules. It works on the set-enumeration tree search framework and the breadth-first search strategy. Their work is basically utilized for constraint-based data mining that effectively prunes the search space. Later on, Li et al. [24] also proposed tree structure based approach to discover jumping emerging pattern in the database. The proposed algorithm, instance-based lazy discovery, and classification system named as DEEPs Classifier utilized contrast pattern tee (CP-tee) structure. This approach discovers jumping emerging patterns for the classification task. Maximal Emerging Patterns (MEP) mining problem is addressed by ZhouWang et al. [25] using

CP-tree based algorithm, CMaxEP. In [26] Roman Podraza and Krzysztof Tomaszewski proposed KTDA framework based on CP-tree. The KTDA is utilized for the purpose of discovering emerging patterns. Yet another tree-based approach is presented by Alhammady in [27] to discover emerging patterns in data streams. The proposed technique Matching Emerging Patterns (MEP) mined a block of data from the stream and discovers patterns. The same process is repeated for other blocks of data steam. Michelangelo Ceci et al. exploited CP-tree for the discovery of multi-relational emerging patterns [28,29]. Authors proposed two methods multi-relational-classification by aggregating emerging patterns (Mr-CAEP) and multi-relational probabilistic emerging patterns (Mr-PEPC) for the discovery of emerging patterns.

Frequent Emerging Graph Pattern Extraction (FEGPE) technique proposed in [30] by Guillaume Poezevara et al. The presented technique ensure to mine the entire search space to explore the frequent emerging graph patterns by utilizing the concept of Apriori-based and pattern growth based algorithms. epSICAR [31], an algorithm presented by Tao Gu et al. for the discovery of emerging patterns from the sequential activities. The esSICAR approach exploits FP-Tree data structure for the discovery of emerging patterns. Xiangtao Chen and Lijuan Lu [32] utilize another variation of tree-based approach by using suffix sub-tree for the discovery of strong jumping emerging patterns. In [33] authors used prefix tree based structure for the discovery of emerging melody patterns. Maybin K. Muyeba proposed an algorithm for the discovery of emerging patterns from the breast cancer database that is named as Attribute-Oriented Induction- High-level Emerging (AOI-HEP) in [34]. This approach exploits tree structure for the discovery of emerging patterns. A CP-tree variation based approach is proposed by Quanzhong Liu et al. to discover strong emerging patterns in [35]. Authors describe how to exploit Dynamically Grown Contrast Pattern Tree (DGCP-tree) for the discovery of patterns with comparatively less computational time. Recently Harsha Parmar proposed a classification approach exploiting CP-tree structure for mining emerging patterns in [36].

The above approaches are exploited CP-tee structure for the discovery of emerging patterns. There are some other variants of tree-based approaches proposed for the discovery of emerging patterns. The Frequent Pattern-Tree (FP-Tree) is also exploited for the discovery of emerging patterns. Tomasz Gambin and Krzysztof Walczak in [37] introduced another FP-tree based approach for jumping emerging patterns provided with pre-specified data set requirements. Modified RAAT is a technique proposed by Zalak V Vyas et al. [38] use FP-tree to discover emerging patterns in real databases.

Moreover, some authors adopt decision tree based approach for emerging pattern problem. García-Borroto et al. proposed an algorithm for mining Crisp Emerging Pattern Miner (CEPMC) in [39]. The algorithm exploits decision tree data structure for the discovery of crisp emerging patterns. Milton Garc et al. enhanced CEPMC to discovery emerging patterns using cascading approach in [40]. Their approach applies local discretization method for better performance. Liang Wang et al. introduced a framework based on Random Forest for the discovery of emerging patterns in [41]. Other authors i.e. Garc-a-Borroto et al. introduced a fuzzy emerging pattern discovery approach named as Fuzzy Emerging Pattern Miner (FEPM) in [42]. The FEPM approach uses graph-based strategy (fuzzy decision trees) for the discovery of fuzzy emerging patterns. Nevertheless, other authors present different techniques not based on tree approach. Hsieh-Hui Yu et al. exploits bitmap approach based on the SAX and PIPs methods for the discovery of time series emerging patterns in [43]. Bayesian Network is utilized by Kui Yu et al. for the discovery of emerging pattern in their proposed work [44].

The tree based algorithms are deterministic and provide comparatively less diversity than proposed ACO based EPACO. The tree based algorithms follow greedy approach, while ACO is based on a relatively better probabilistic approach for search space. The complexity of any tree based algorithm depends on the number steps required to grow and shrink the tree. The proposed EPACO algorithm is indented to provide a competitive classifier to tree based emerging pattern classifiers with complexity and diversity of ACO. The proposed approach EPACO avoids an exhaustive and greedy search of tree-based algorithms to achieve the same or even better accuracy by discovering strong jumping emerging patterns with a diversity of ACO.

3 Ant colony optimization

The ACO is a bio-inspired paradigm of Swarm Intelligence (SI) for designing the meta-heuristic approaches for optimization problems. Swarm Intelligence is an emerging branch of Computational Intelligence. Swarm Intelligence is a collection of algorithmic approaches inspired by the collective intelligence behavior of groups of simple agents [45]. Colorani, Dorigo, and Maniezzo exploited the concept of Ant Colony Optimization in an optimization algorithm namely known as Ant System in early 1990 [22]. The basic working procedure of ACO based approaches is inspired by the food searching behavior of real ants. The insect's members of the swarm, such as ants and bees, can perform simple tasks individually while their cooperative behavior emerged solution concepts for complex and hard problems. The simulation of foraging behavior of real ants provides the solution of the hard and complex real world problems. The pheromone value provides mechanism for the mutual sharing of the information in the real ants. This cooperative behavior results in solution of the hard problems of engineering, industry and business problems. An artificial ant can be considered as a simple computational agent. The phenomena of pheromone evaporation of the real ant are simulated by using the mathematical formulae in the artificial ant systems. Generally the pheromone evaporation rate is directly proportional to the length of path. The ACO based meta-heuristic approaches are more promising for the problems where optimization is desired.

Ant colony optimization is prominently used for the discovery of classification rules and association rules which results in efficient, robust and more accurate classifiers. The ACO was first applied for the discovery of classification rules by Parpinelli et al. which is known as AntMiner [46]. Liu et al. proposed extension in the basic AntMiner algorithm in AntMiener2 [47] and AntMiner3 in [48]. Martens et al. [49] proposed an AntMiner+ algorithm based on Max-Min Ant System that differs from the previously proposed AntMiners in several aspects. Abdul Rauf Baig et al. proposed improvements in the cAntMiner algorithm in [50] that provided promising classification rule discovery in medical data sets. Waseem Shahzad and Abdul Rauf Baig proposed a new bio inspired hybrid classification approach, named ACO-AC in [51]. ACO-AC algorithm exploits hybrid approach by combining the idea of association rules mining and supervised classification. The literature study shows the application of ACO for the discovery of rules for the classification task using supervised training data. Fernando E. B. Otero et al. proposed a classification rule mining ACO based algorithm which introduced improvements in Ant-Miner for coping with continuous attributes, named cAnt-Mine [52]. The ACO approach is already has been utilized in rule mining and associative classification to achieve better performance. In this article ACO is exploited for the discovery of emerging patterns and emerging patterns based classification.

4 Proposed approach

This section elaborates the proposed approach, a novel Ant Colony Optimization for Emerging Patterns Mining (EPACO).

4.1 A novel ant colony optimization for emerging patterns mining (EPACO)

The proposed approach ant colony optimization for emerging patterns mining (EPACO) is based on the Ant-Miner [46] and ACO-A [51]. In the EPACO, the class is selected based on support and growth ratio of the discovered pattern while in ACO-AC the class is selected first and then associative classification rules are discovered. The EPACO use new heuristic value calculation mechanism given in the Eq. (7) . The Algorithm 1 explains our main proposed approach. While algorithms given 1.1, 1.2 and 1.3 provide basic working of sub algorithms. The EPACO finds the jumping emerging pattern and strong jumping emerging patterns by using Definition 4 and Definition 5. As presented in algorithm 1 the process starts with initialization of various terms in the data set. This is followed by iterative structure to find 1-term strong jumping emerging pattern in data set using all initialized terms. Each term is used individually to find if it is alone enough as a SJEP. The algorithm then calculates support and growth ratio for each already discovered pattern by using Eqs. (1) and (2) respectively. If support is greater than or equal to some given minimum support threshold value (line 5), the pattern is added to the pattern-list (line 6) and the list is sorted by growth ratio in descending order (line 7). In algorithm line 8 calculates the coverage of the entire pattern list. If coverage reaches the minimum coverage threshold (line 9–10), the search for 1-term patterns is stopped (consequently, the search for any pattern is stopped).

Algorithm 1 EPACO

```
1.  Initialize terms
2.  For all s ϵ Terms do
3.    Build 1-term pattern featuring s
4.    Calculate pattern support and pattern growth ratio
5.    If pattern support >= min. support
6.        Add pattern to discovered pattern list
7.        Sort pattern list by pattern growth ratio
8.    Set Coverage ← Find Coverage()
9.    If Coverage >= min.Coverage
10.       Break
11. End For
12. Initialize Pheromone()
13.   Calculate term Heuristics ( )
14. Set g ← 2
15. Set i ← 0
16. While (i<num_of_iterations)
17. While g<|Attributes| ^  Coverage < min. Coverage
18.       For all t ϵ ants do
19. Calculate Selection Probability( )
20.       Build g-terms pattern by ant t
21.       If pattern support >= min. support
22.           Add pattern to pattern list
23.           Update pheromone w.r.t.  pattern growth ratio
24.   End For
25.   Sort pattern list by pattern growth ratio
26.   Set Coverage ← Find Coverage()
27.   If Coverage >= min. Coverage
28.       Break
29.   Set g ← g + 1
30. End While
31. End While     [step 16 loop]
32. Prune pattern list
33. Sort pattern list w.r.t. pattern growth ratio
```

Lines 13–27 discover patterns with multiple terms per pattern till the coverage of the discovered pattern list is less than minimum coverage and the attribute counter g is less than the number of non-class attributes. Note that g is initialized with the value of 2 (line 14) because 1-term patterns already discovered in lines 2–11. Line 12 initializes the pheromone value. Line 13 calculates the heuristic values for each term. Lines 16–31 are repeated for each ant. An ant discovers a strong jumping emerging pattern (SJEP) containing g terms (line 20). If the support of the pattern discovered by the ant *t* has greater than or equal to the minimum threshold, the pattern is added to the discovered pattern list and the pheromone values are updated according to the growth ratio (lines 21–23). The pattern list is sorted with respect to growth ratio in descending order (line 25). Lines 22–25 are same as lines 8 to 10. The value of *g* is incremented in line 29.

The construction of 1-term Pattern is given in Algorithm 1.1, calculation of support ratio and coverage finding is presented in Algorithm 1.2 and Algorithm 1.3 respectively.

Algorithm 1.1 Construct_ 1_term Pattern(t)

1. *Create an empty pattern p where each attribute value is set to unknown (e.g.?)*
2. *Put the value of the term t in its attribute value.*
3. *Return the pattern*

Algorithm 1.2 Calculate_Support_Ratio(p)

1. *Calculate supports for the pattern p in each class of the training set.*
2. *If support for p in every class is less than the minimum threshold then set support and growth ratio to 0, set pattern class to null and return.*
3. *If support of p in all classes is 0 then set the pattern support to the support of the class c in which p has non-zero support, class of the pattern to c, set growth ratio to infinity and return.*
4. *Return the support of the class c in which p has the highest support, set growth ratio to the support of p in c divided by the sum of supports of p in all classes other than c, and set c as the pattern class.*
5. *Return*

Algorithm 1.3 Find_Coverage(P)

```
Set count=0
For each pattern p in P
  For each instance in D
    Set Match=false
    For each attribute in Attributes
      If p.attribute<>? And p.attribute=instance.attribute Then
        Set match=true
        Break the inner most loop
    End For
    If match Then
      Set count=count+1
      Break the instance loop
  End For
End For
Return count
```

The working procedure and flow of the information of the EPACOare elaborated with the help of flowchart given in Fig. 1. In the first step, training dataset is loading and system parameters are initialized. In the next step stopping criteria of the training process is evaluated if the training criteria are not met then pheromone and heuristic values are initialized for the Ant System for the discovery of emerging patterns (EPACO). After the initialization, each time interval convergence criteria is evaluated. If a convergence criterion is not met then this loop will iterate until the convergence criteria are met. In this loop emerging pattern is discovered by each ant for each time interval, support and growth rate of each discovered emerging pattern is calculated. If the discovered pattern meets the threshold of support and growth rate, the pattern is assigned majority/convergence class. The fitness of the pattern is evaluated and pheromone value is updated. Again convergence criteria are evaluated. If pattern meets the convergence criteria then the pattern is selected and pattern list is pruned for the best patterns. After this phase again stopping criteria is evaluated if criteria are met training process is stopped and testing data is provided for the evaluation of the proposed classification approach. The last phase of the proposed classification approach reports the predictive accuracy of the classifier on the given testing dataset.

4.2 Pheromone initialization

The pheromone values on all edges are initialized before the start of while loop for each new class. The equal amount of pheromone is initialized on the edges between all items of a pattern set. The initial value of pheromone (τ) of each term is initialized with the Eq. (3).

$$\boldsymbol{\tau}_{i,j}^{(t=1)} = \frac{\mathbf{1}}{\sum_{i=1}^{\mathbf{a}} \boldsymbol{b}_i} \tag{3}$$

where '*an*' is the number of attributes and b_i is the number of domain values of the ith attribute.

4.3 Heuristic function

In our proposed method heuristic value of each term is based on the information gain of each term. Information gain is calculated on the basis of the entropy measure. Entropy (E) of the entire data set (S) is calculated by using Eq. (4).

$$\mathbf{E(S)} = -\sum_{\mathbf{c}} \boldsymbol{P}_c \boldsymbol{lg} \, \boldsymbol{P}_c \tag{4}$$

where c represents the class label, lg is the log with base 2 of the probability p of class c.

Information gain g of term t is calculated with the Eq. (5).

$$\boldsymbol{g}(\boldsymbol{C}|\boldsymbol{A}_i = \boldsymbol{t}_{ij}) = -\sum_{\boldsymbol{c}\in\boldsymbol{C}} \boldsymbol{p}(\boldsymbol{c}|\boldsymbol{A}_i = \boldsymbol{t}_{ij}).\boldsymbol{lgp}(\boldsymbol{c}|\boldsymbol{A}_i = \boldsymbol{t}_{ij}) \tag{5}$$

where A_i represents the ith attribute of the data set, while t_{ij} is the jth value of the ith attribute. Now the finally Heuristic function for the term t is computed with the help of Eq. (6).

$$\eta_{ij} = \frac{\mathbf{lg}(\boldsymbol{C}) - \boldsymbol{g}(c|\boldsymbol{A}_i = \boldsymbol{t}_{ij})}{\sum_{i=1}^{\mathbf{a}} x_i \,.\, \sum_{j=1}^{b_i} (\mathbf{lg}(\boldsymbol{C}) - \boldsymbol{g}(\boldsymbol{c}|\boldsymbol{A}_i = \boldsymbol{t}_{ij}))} \tag{6}$$

where C represents the set of classes in the dataset, a represents the number of attributes of the data set and b represents the number of domain values of the attribute i.

The probability of the selection of the term is calculated with Eq. (7).

$$\boldsymbol{P}_{ij} = \frac{\boldsymbol{\tau}_{ij}^{\alpha}.\boldsymbol{\eta}_{ij}^{\beta}}{\sum_{i=1}^{a} \mathbf{x_i}.\sum_{j=1}^{b_j} (\boldsymbol{\tau}_{ij}.\boldsymbol{\eta}_{ij})} \tag{7}$$

where τ_{ij} is the amount of pheromone between $item_i$ and $item_j$ in the current iteration. The term η_{ij} in Eq. (7) is the value of the heuristic function on the link between $item_i$ and $item_j$ for the currently selected class. Where α and β are the constant exponents to control the impact of pheromone and heuristic of the $term_{(i,j)}$ in the probability of the selection of the particular item. Their default value is 1.

4.4 Pattern discovery stoppage

An ant continues to add discovered pattern to the pattern list in each generation. The pattern construction process can stop if the value of generation counter is equal to the total number of attributes present in the data set except class attribute that means all attributes have been tested and second if the coverage of the instances of the data set reaches the minimum coverage threshold.

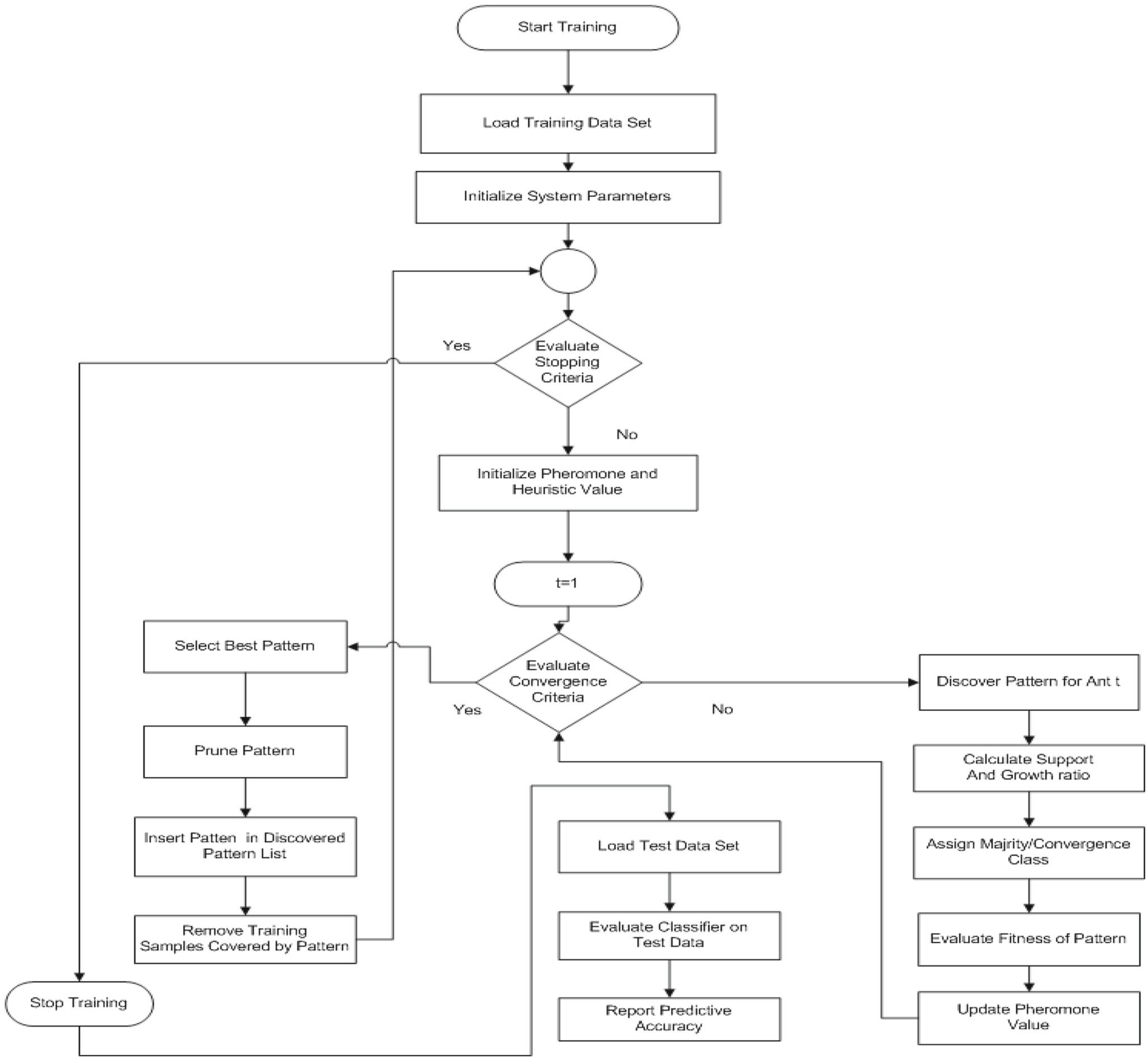

Fig. 1 Flowchart of proposed approach (EPACO)

4.5 Quality of pattern and pheromone update

The quality of a pattern is computed on the basis of growth ratio of the pattern which is calculated in Eq. (8).

$$Q = \frac{TP}{Covered} \tag{8}$$

where *Covered* is the number of training samples that match with the rule antecedent part constructing by an ant and *TP* is the number of training samples whose antecedent is same as the antecedent of the pattern of ant and whose consequent is also same as consequent of the pattern of ant.

In our proposed ACO based emerging pattern mining algorithm uses Eq. (9) for the updating of pheromone values after each generation so that in next generation ants use this information for the discovery of emerging patterns. The amount of pheromone is updated on the links between items occurring in those patterns which satisfy minimum support threshold and growth ratio.

$$\tau_{ij}^{(g+1)} = \tau_{ij}^{(g)}(1-\rho) + \left(1 - \frac{1}{1+Q}\right) \cdot \tau_{ij}^{(g)} \tag{9}$$

where $\tau_{ij}^{(g)}$ the pheromone value between $item_i$ and $item_j$ is in current generation, ρ indicates the pheromone evaporation rate and Q is the quality of the pattern discovered by an ant. The pheromone value of these patterns are increased so that in next generation search space exploration by the ant can be optimized instead of searching around those patterns which already inserted in the discovered pattern set. It is the promising strategy in ACO which increases the diversity of the search by focusing and enabling exploration of the unexplored search area previously. Each pheromone value is normalized by dividing it by the summation of all pheromone values of its competing pattern items.

4.6 Pattern selection process

The selection of the term in a pattern is deterministic for the 1st term where each term is used to discover SJEPs. All other terms are selected based on probabilities of selection. The probability of selection of a term is based on both the pheromone value of the term and its heuristic value.

4.7 Discovered patterns set and pruning of set

When the coverage of discovered patterns set reaches a coverage threshold, then pattern discovery process is stopped. This process is repeated for all classes and finally mined pattern list contains discovered patterns of all classes. The discovered pattern set may contain a large number of patterns and may be with the redundant pattern. The redundant patterns are those patterns which do not fire for any single training sample. These redundant patterns are removed from the final pattern set. The discovered pattern set of patterns is first sorted on the basis of growth ratio. Then, it is applied to classify the samples of the training set. For each sample, the patterns in the discovered pattern set are tested one by one in order of their sorting. If a pattern fires for the test sample then the patterns below it are not tested. The pattern pruning process flags all those patterns which are fired for at least one training sample. This process exposes those patterns which are never used. All such patterns are discarded from the pattern set. The rest of the pattern set becomes the final classifier and used to predict unseen test cases. The pruning of the patterns set improves the comprehensibility of classifier as a smaller number of patterns promising for the understandability of the domain expert as well as it causes classification process fast because for classifying a test case we check each pattern one by one.

5 Experimental results and discussion

The proposed algorithm EPACO is implemented in language C#. We have conducted experiments on a machine that has 1.57GHZ dual processor with 1GB RAM. We compare the results of proposed technique EPACO with other state-of-the-art, well-known classification algorithms which are C4.5Rules-C, PART-C, Ripper-C, C45-C, Kernal-C, ID3-C, KNN-C, NB-C, NU-SVM-C, C-SVM-C, Logistic-C, CPSO-C and Ant-Miner-C by using implementation in data mining and machine learning tool, Knowledge Extraction based on Evolutionary Learning (KEEL) [53]. Default values are used as parameters for all the classification approaches corresponding in KEEL Tool. The proposed approach is also compared with the emerging pattern discovery based on CP-tree classification approaches SJEP, NEP and GNEP [21] which are being reproduced in C#. Table 3 shows the detailed performance analysis of the proposed EPACO classification approach based on emerging patterns by using KEEL Data Mining and Machine Learning tool for the corresponding algorithms. In our experiments, 10-fold cross-validation methodology is applied and mean of results of ten-fold performance is reported. The same data set partition and exactly the same training and testing data are used for all the classifiers involved. Table 3 shows comparative accuracy performance in percentage, of the emerging pattern-based classifier EPACO with state-of-the-art statistical and evolutionary classification algorithms.

5.1 Data set description

The Table 1 shows the description of data sets which are used for the performance evaluation of the bio-inspired proposed approach with state-of-the-art statistical and evolutionary approaches. Table 1 gives information about the number of instances, the number of attributes and the number of classes of the various public data sets that are downloaded from the UCI website [54]. In the present study 28 data sets are being used with a different number of instances, a number of attributes and number of classes for the performance analysis with a variety of number of instances, attributes, and classes. The data sets also have diversity in terms of their number of attributes, the number of transactions, the number of classes and types of the attributes. As our proposed approach works only for the categorical attribute, therefore, we discretize the continuous attributes in a preprocessing step by using the supervised discretization filter. The data Sets are discretized after replacing missing values that are used for the comparative performance analysis with state-of-the-art approaches that are described in Sect. 4.

5.2 Parameters used in experiments

The parameters used for the EPACO during the experiments are given in Table 2. The data sets are created by using ten folds methodology. The number of Ants used is 15. The value of minimum support is 0.3, minimum coverage is 1.0 and

Table 1 Data sets description

Dataset	#Instances	#Attributes	#Classes	Dataset	#Instances	#Attributes	#Classes
Agaricus	8124	22	2	Ionosphere	351	34	2
Australian	690	14	2	Iris	150	4	3
Backup	307	35	2	Labor	40	16	2
Breast	699	10	2	Lenses	24	6	3
Bupa	345	6	2	Nursery	12960	9	3
Cleveland	303	13	2	Pima	768	8	2
Crx	690	15	2	Shuttle	15	7	6
Diabetes	768	8	2	Sick-Euthyroid	3163	25	2
German	1000	20	2	Soybean	307	35	4
Glass	214	9	7	Tic-Tac-Toe	958	9	2
House-Votes-84	435	16	2	Waveform	5000	21	3
Hepatitis	155	19	2	Wine	178	13	3
Horse	300	28	2	Xaa	94	18	4
Hypothyroid	3163	26	2	Xab	94	18	4

Table 2 Parameters used in experiments

Parameter name	Value
No. of ants	15
Minimum support (SJEPs only)	0.3
Minimum coverage	1.0
No. of folds	10
Pheromone evaporation rate	0.09
No of iterations	100
Alpha	1
Beta	1

pheromone evaporation rate is .09. There are 100 numbers of iterations for the termination of the algorithm. The values of Alpha and Beta in ACO algorithmic formula is adjusted as 1.

5.3 Performance evaluation

Table 3 shows the detailed performance analysis of the proposed EPACO classification approach based on emerging patterns by using KEEL Data Mining and Machine Learning tool for the corresponding algorithms. Table 3 also shows comparative performance in terms of accuracy in percentage with the state-of-the-art classification algorithms. The performance of the proposed algorithm EPACO is very promising for some data sets (e.g. Australian, Breast, Hepatitis, German and Pima) in the context of accuracy with respect to the state -of-the- art algorithms. The algorithm ID3 performed well on Glass and Nursery data sets while the performance of NEP was promising for Cleveland and Shuttle data sets by accuracy. With the critical observation of the accuracy results given in Table 3 shows that EPACO is wining 16 times, 2 times draw from the other state of the art algorithms in the context of accuracy. The accuracy results of the proposed approach are very competitive and comparative with other classification approaches. On average the accuracy EPACO for the majority data sets is higher than other approaches as shown in Table 3 in bold faces. The naming convention for classification approaches is same exploited that is used in KEEL implementation for the corresponding classifiers.

5.3.1 Comparison with decision tree based classifiers

Table 3 shows the performance comparison with decision-based classifiers Iterative Dicotomizer 3 (ID3-C) [55] and C4.5(C4.5-C) [56] with the proposed classification approach in terms of accuracy. The performance of EPACO is significantly promising for data sets Bupa, German, Hepatitis, Pima and waveform. The EPACO is winner 25 times out of 28 data sets in terms of accuracy from the decision tree based classifiers. The performance of EPACO lowers for Agaricus, Glass and Nursery data sets.

5.3.2 Comparison with support vector machines

Table 3 depicts the performance comparative analysis of proposed classification approach EPACO based on ACO with NU-SVM (NU_SVM-C) [57] and C-SVM (C_SVM-C) [58] by using the KEEL implementation of corresponding support vector machines. With observation of results the proposed approach is winner 26 times out of 28 data sets. The performance of EPACO is more promising on the data sets (i.e

Table 3 Comparative performance analysis

Data sets	Decision tree-based classifiers		Support vector machines		Crisp rule learning classifiers			Evolutionary crisp rule learning classifiers		Proposed classifier
	C45-C Accuracy %	ID3-C Accuracy %	NU_SVM-C Accuracy%	C_SVM-C Accuracy %	C45Rules-C Accuracy %	PART-C Accuracy %	Ripper-C Accuracy %	CPSO-C Accuracy %	Ant_Miner-C Accuracy %	EPACO
Agaricus	62.210841	*82.422306*	35.895243	62.998473	61.853638	49.421743	63.047901	47.673171	57.52045	63.44
Australian	85.652174	80.289855	75.072464	85.507246	84.927536	59.565217	81.014493	84.347826	84.202899	**99.86**
Backup	97.72043	98.053763	97.72043	96.741935	93.182796	84.709677	98.376344	86.086022	96.419355	**99.08**
Breast	94.279503	93.128364	93.850932	94.846791	92.556936	65.950311	94.559006	91.691511	90.552795	**98.61**
Bupa	59.159664	65.823529	58.588235	57.983193	59.386555	57.983193	59.084034	51.840336	60.02521	**99.15**
Crx	85.072464	79.42029	75.652174	85.507246	85.652174	61.304348	82.463768	64.492754	83.913044	**99.85**
Diabetes	73.691046	80.989405	50.550239	68.491114	70.704033	64.974368	67.828093	68.617567	72.648667	**97.27**
German	71.5	68.6	67.1	71.9	69	70	66.1	68.2	69.9	**99.8**
Glass	61.796537	*83.181818*	58.896104	61.709957	63.181818	35.541126	43.073593	55.606061	50.930736	26.59
House-Votes	96.79704	94.936575	95.877378	95.877378	96.337209	88.261099	94.249472	95.174419	92.891121	**99.77**
Hepatitis	63.791667	57.25	60.5	70.333333	60	54.166667	63.25	59.375	65.291667	**98.66**
Horse	85	80.333333	79	83.666667	77.666667	70.333333	78.666667	64	70.666667	**99.34**
Hypothyroid	63.791667	57.25	60.5	70.333333	60	54.166667	63.25	59.375	65.291667	**98.67**
Ionosphere	86.325397	86.595238	85.174603	78.936508	76.047619	64.103175	87.15873	64.388889	78.055556	**98.29**
Iris	96	92	96.666667	97.333333	90	42	90	69.333333	92	**98.67**
Labor	67.5	80	82.5	70	90	67.5	75	77.5	75	**97.5**
Lenses	85	78.333333	78.333333	76.666667	78.333333	61.666667	66.666667	55	78.333333	**96.67**
Nursery	97.114198	*99.74537*	58.356481	90.99537	96.574074	82.908951	97.453704	79.521605	76.705247	65.21
Pima	72.908407	68.993507	59.509569	65.105947	67.4419	65.235817	71.221805	63.422761	70.440875	**99.87**
Cleveland	53.419355	54.150538	50.193548	55.107527	52.505376	54.11828	50.193548	49.16129	57.11828	**57.8**
Shuttle	50	60	60	70	*75*	60	60	55	55	**75**
Sick-Euthyroid	95.985106	95.858424	95.763886	95.289602	93.993831	90.705187	94.688636	90.736932	91.273609	**100**
Soybean	97.72043	98.053763	97.72043	97.075269	93.817204	88.612903	98.376344	93.182796	94.462366	**98.71**
Tic-Tac-Toe	85.502193	95.302632	91.855263	71.299342	83.300439	65.344298	97.497807	72.751096	69.828947	**100**
Waveform	74.04	69.88	66.46	59.72	63.44	34.18	71.98	49.24	65.28	**99.66**
Wine	83.202614	91.633987	96.045752	92.679739	83.137255	47.156863	81.372549	64.575163	87.679739	**96.61**
Xaa	52.888889	51.222222	45.333333	32.555556	50.222222	29.666667	39.777778	31.555556	42.444444	**66**
Xab	44.444444	53.222222	41.444444	28.666667	34.777778	26.444444	37.888889	27.444444	41.111111	**70.33**
Average	76.518359	78.452517	71.94859	74.54743	75.108585	60.572178	74.079993	65.689055	72.678135	**89.3003**

Table 3 continued

	Statistical classifiers			CP-tree based classifiers			Proposed classifier
Data sets	NB-C Accuracy %	Logistic-C Accuracy %	Kernal-C Accuracy %	SJEP Accuracy %	NEP Accuracy %	GNEP Accuracy %	EPACO Accuracy %
Agaricus	*64.340474*	61.521489	38.749387	54.55	58.35	56.24	63.44
Australian	84.492754	85.507246	55.507246	94.65	98.41	95.65	**99.86**
Backup	97.72043	95.419355	84.709677	96.74	*99.35*	99.35	99.08
Breast	97.283644	93.991718	79.254658	72.38	91.26	87.06	98.61
Bupa	61.722689	57.420168	57.983193	42	46.45	90.72	**99.15**
Crx	84.637681	85.942029	55.652174	88.84	97.1	95.65	**99.85**
Diabetes	73.036569	68.619275	65.105947	66.93	94.53	79.3	**97.27**
German	74.6	73.4	70	72.8	98.6	98.5	**99.8**
Glass	*63.181818*	62.705628	35.541126	32.71	61.68	61.68	26.59
House-Votes	90.359408	95.179704	61.379493	96.78	*99.54*	97.7	**99.77**
Hepatitis	70.333333	69.083333	54.791667	59.33	89.68	89.68	**98.66**
Horse	80.333333	82	67	62.33	*100*	*100*	99.34
Hypothyroid	70.333333	69.083333	54.791667	59.35	86.45	83.87	**98.67**
Ionosphere	90.309524	78.952381	70.380952	93.45	95.44	95.44	**98.29**
Iris	94	96	93.333333	91.31	97.35	95.35	**98.67**
Labor	92.5	80	65	95	*97.5*	*97.5*	**97.5**
Lenses	71.666667	76.666667	61.666667	16.67	95.83	*100*	96.67
Nursery	90.29321	90.740741	65.061728	97.25	*98.53*	96.12	65.21
Pima	75.111073	67.062543	65.105947	69.01	84.51	84.51	**99.87**
Cleveland	57.451613	55.451613	54.11828	78.23	84.29	*89.11*	57.8
Shuttle	70	75	65	40	*100*	*100*	75
Sick-Euthyroid	93.676876	95.44793	90.736833	96.4	97.85	95.83	**100**
Soybean	97.72043	96.731183	84.709677	96.64	96.42	98.05	**98.71**
Tic-Tac-Toe	70.144737	70.982456	65.344298	*100*	*100*	*100*	**100**
Waveform	80.66	59.54	52.76	35.1	33.14	33.14	**99.66**
Wine	96.013072	89.281046	92.647059	92.5	*97.75*	*97.75*	96.61
Xaa	59.777778	35.111111	33.777778	29.9	*87.23*	*87.23*	66
Xab	58.111111	32.777778	43.555556	27	*77.66*	74.47	70.33
Average	78.921841	74.986383	63.702298	69.92321	88.03214	88.56785	**89.3003**

Bold values show the better performance of the corresponding classifier (algorithm)
Italic values indicate that the corresponding approach is winning on the specific dataset

Table 4 Performance of EPACO on the variation of growth ratio

Data Sets	Accuracy % at (0.1)	Accuracy % at (0.2)	Accuracy % at (0.3)	Accuracy % at (0.4)	Accuracy % at (0.5)	Accuracy % at (0.6)	Accuracy % at (0.7)	Accuracy % at (0.8)	Accuracy % at (0.9)	Average
Agaricus	60.16788	45.21189	42.66375	42.6637	29.4928	29.44356	20.1996	19.59619	16.44505	33.9872
Australian	100	99.85507	99.85507	99.8551	99.56522	99.56522	99.5652	99.56522	98.55072	99.5974
Backup	99.03226	99.03226	99.03226	99.0323	99.03226	99.03226	99.0323	99.03226	97.09677	98.8172
Breast	99.56936	100	100	100	100	100	100	100	100	**99.9522**
Bupa	100	98.55462	96.21849	95.3529	94.48739	89.57143	89.5714	57.98319	57.98319	76.6835
Crx	100	99.85507	99.85507	99.8551	99.56522	99.56522	99.5652	99.56522	98.55072	99.5974
Diabetes	99.47881	99.21907	98.95762	97.2659	97.26589	92.58202	92.582	67.31374	65.10595	89.9746
German	99.9	100	99.9	100	100	100	100	100	99.7	99.9444
Glass	24.26407	20.95238	22.27273	19.5455	19.06926	13.54978	9.35065	7.922078	7.922078	16.0943
House-Votes-84	99.76744	99.76744	99.76744	99.7674	99.76744	99.76744	99.7674	99.76744	99.76744	99.7674
Hepatitis	100	100	99.33333	99.3333	98.66667	98.66667	98.6667	100	100	99.4074
Horse	99.33333	99.66667	99.66667	99.6667	99.33333	99.33333	100	100	100	99.6667
Hypothyroid	100	100	99.33333	99.3333	98.66667	98.66667	98.6667	100	100	99.4074
Ionosphere	98.57143	99.42857	99.71429	100	100	100	100	100	100	99.746
Iris	98.66667	98.66667	98.66667	98.6667	98.66667	98.66667	98.6667	98.66667	98.66667	98.6667
Labor	97.5	97.5	97.5	95	97.5	95	97.5	97.5	97.5	96.9444
Lenses	96.66667	96.66667	93.33333	100	100	100	100	100	100	98.5185
Nursery	64.6142	64.74537	64.55247	64.5525	64.55247	64.54475	64.5525	64.55247	64.55247	64.5799
Pima	100	98.55462	96.21849	95.3529	94.48739	89.57143	89.5714	57.98319	57.98319	86.6359
Cleveland	55.43011	55.43011	55.43011	54.1183	54.11828	54.11828	54.1183	54.11828	54.11828	54.5556
Shuttle	75	75	90	90	90	90	90	90	100	87.7778
Sick -Euthyroid	100	100	100	100	100	100	100	100	100	100
Soybean	99.03226	99.03226	99.03226	99.0323	99.03226	99.03226	99.0323	99.03226	94.76344	98.5579
Sonar	98.52381	98.09524	99.04762	100	91.35714	86.57143	78.2857	53.38095	53.38095	84.2937
Tic-Tac-Toe	100	100	100	100	85.38925	65.3443	65.3443	65.3443	65.3443	82.974
Vehicle	100	100	100	100	85.38925	65.3443	65.3443	65.3443	65.3443	82.974
Waveform	99.12	97.46	84.56	49.8	33.92	33.92	33.92	33.92	33.92	55.6156
Wine	97.18954	97.7451	97.7451	76.2745	59.47712	57.81046	40.5229	39.93464	39.93464	62.9666
Xaa	65.88889	54.33333	37.22222	31.8889	29.77778	29.77778	29.7778	29.77778	29.77778	37.5802
Xab	63.77778	59.33333	41.11111	26.4444	26.44444	26.44444	26.4444	26.44444	26.44444	35.8765
Average	**89.716484**	88.470191	87.032981	84.426723	81.500807	79.196323	78.001592	74.224821	74.095079	

Bupa, Pima and Hepatitis) with respect to other support vector machines.

5.3.3 Comparison with crisp rule learning classifiers

Table 3 represents the performance comparative analysis of proposed classification approach EPACO based on ACO with C4.5Rules (C45Rules-C) [56], PART (PART-C) [59] and Repeated Incremental Pruning to Produce Error Reduction (Ripper-C) [60] by using the KEEL implementation of corresponding crisp rule leaning classifiers. With observation of results, the proposed approach is winner 26 times out of 28 data sets. The performance of is more promising on the data sets(i.e. Bupa, Pima, and Hepatitis) with respect to other support vector machines.

5.3.4 Comparison with evolutionary crisp rule leaning classifiers

In this section, the performance of proposed approach is compared with the implementation of Swarm Intelligence based classifiers Constricted particle swarm optimization (CPSO-C) [61] and Ant-Miner (Ant-Miner-C) [46] in KEEL tool.

The Table 3 shows that the EPACO is winner 26 times out of 28 data sets, 1 time loses (for Glass data set) and 1 time drawn (for Cleveland data set with Ant-Miner-C).

5.3.5 Comparison with statistical classifiers

Table 3 depicts accuracy comparative analysis of proposed approach EPACO with the implementation of Statistical Classifiers Kernel Classifier (Kernal-C) [62], Multinomial logistic regression model with a ridge estimator (Logistic-C) [63] and Naïve-Bayes (NB-C) [64] in KEEL tool by using public data sets. The EPACO is winner 22 times out of 28 data sets, 2 times loses (for Glass and Nursery data sets) and 4 times drawn (for Cleveland, Wine, Shuttle and Agaricus data sets).

5.3.6 Comparison with CP-tree based classifiers

In this section, the performance of proposed approached EPACO is compared with the implementation of contrast pattern tree (CP-tree) based classifiers (SJEP, NEP, and GNEP) [21]. The mostly approaches for emerging patterns discovery exploit constant pattern tree. The CP-Tree based classification approaches (SJEP, NEP, and GNEP) are implemented in C# programming language. The EPACO is winner 16 times out of 28 data sets, 6 times loses (i.e. Xab, Xaa, Cleveland, Nursery and Glass data sets) and 6 times drawn (i.e. Backup, Horse, Labor, Tic-Tac-Toe and Wine data sets).

5.3.7 Performance of EPACO by variating growth ratio

Table 4 represents the performance behavior of EPACO with the variation of growth ratio. We have critically analyzed the impact of variation of growth ratio from 0.1 to 0.9; on the accuracy of the proposed classification approach. Table 4 reveals the performance that EPACO is better on the lower growth ratio and degrades with the increase of growth ratio. The impact of the growth ratio is negligible and even improves the performance for the data sets having smaller instances (i.e. Labor and Shuttle) and for Backup, Breast, German and Sick-Euthyroid EPACO performance remained constant with the variation of growth ratio. Table 4 depicts the average performance of EPACO vertical as well as horizontal. The average performance of the algorithm is promising on the lower (0.1) as shown in Table 4.

6 Conclusion and future work

Bio-inspired meta-heuristic ant colony optimization is first time exploited for the discovery of emerging patterns. A novel algorithm, EPACO has been proposed to discovery emerging pattern with relatively higher accuracy and less classifier complexity. Furthermore, comprehensive experimental analysis has been presented to show the better performance of the proposed approach. The proposed approach is tested for 28 standard benchmark datasets. In most cases, the proposed approach discovers high quality emerging patterns within reasonable time and high accuracy.

Nevertheless, the experimental study also reveals that the proposed approach is sensitive to the number of instances or size of the dataset. Moreover, comparative analysis with state-of-the-art classifiers shows that EPACO outperforms other classifiers with better accuracy. Yet, sensitivity analysis shows that classifier accuracy is inversely related to the growth ratio. The results show that proposed classification approach is more accurate, efficient and robust than other statistical and evolutionary classification approaches.

In future, the proposed approach can be improved and analyzed the performance for the data sets having a larger size and more classes for the discovery of emerging patterns. Further, the approach can be applied in the field of bioinformatics for the analysis and classification of the voluminous data.

References

1. Kwasnik, B.H.: The role of classification in knowledge representation and discovery. Libr. Trends **48**(1), 22 (1999)
2. Zhang, G.P.: Neural networks for classification: a survey. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **30**, 451–462 (2000)

3. Yoon, H.-S., Lee, S.-H., Kim, J.H.: Application of emerging patterns for multi-source bio-data classification and analysis. Advances in Natural Computation, pp. 965–974. Springer, Berlin (2005)
4. Fan, H., Ramamohanarao, K.: A weighting scheme based on emerging patterns for weighted support vector machines. In: Granular Computing, 2005 IEEE International Conference, IEEE (2005)
5. Wu, G., et al.: The classification prognosis models of hepatitis b virus reactivation based on Bayes and support vector machine after feature extraction of the genetic algorithm. In: Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016 12th International Conference, IEEE (2016)
6. de Boves Harrington, P.: Support vector machine classification trees based on fuzzy entropy of classification. Anal. Chim. Acta **954**, 14–21 (2017)
7. Yong, Z., Youwen, L., Shixiong, X.: An improved KNN text classification algorithm based on clustering. J. Comput. **4**(3), 230–237 (2009)
8. Gokgoz, E., Subasi, A.: Comparison of decision tree algorithms for EMG signal classification using DWT. Biomed. Signal Process. Control **18**, 138–144 (2015)
9. Samantaray, S., ACHLERKAR, P., Manikandan, M.S.: Variational mode decomposition and decision tree based detection and classification of powerquality disturbances in grid-connected distributed generation system (2016)
10. Guan, S.-U., Zhu, F.: An incremental approach to genetic-algorithms-based classification. IEEE Trans. Syst. Man Cybern Part B **35**(2), 227–239 (2005)
11. Enee, G., Escazut C.: Classifier systems evolving multi-agent system with distributed elitism. In: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress, IEEE (1999)
12. Keshavarz, H., Abadeh, M.S.: SubLex: Generating subjectivity lexicons using genetic algorithm for subjectivity classification of big social data. In: Swarm Intelligence and Evolutionary Computation (CSIEC), 2016 1st Conference, IEEE (2016)
13. Adeniyi, D., Wei, Z., Yongquan, Y.: Automated web usage data mining and recommendation system using K-nearest neighbor (KNN) classification method. Appl. Comput. Inform. **12**(1), 90–108 (2016)
14. Khashei, M., Hejazi, S.R., Bijari, M.: A new hybrid artificial neural networks and fuzzy regression model for time series forecasting. Fuzzy Sets Systems **159**(7), 769–786 (2008)
15. Basha, S.H., Abdalla, A.S. Hassanien, A.E.: GNRCS: hybrid classification system based on neutrosophic logic and genetic algorithm. In: Computer Engineering Conference (ICENCO), 2016 12th International, IEEE (2016)
16. MohammadZadeh, J.: Social networks classification using DBN neural network based on genetic algorithm. Social Networks (2016)
17. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM (1999)
18. Fan, H., Ramamohanarao, K.: Noise tolerant classification by chi emerging patterns. Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin (2004)
19. Li, J., Ramamohanarao, K., Dong, G.: The space of jumping emerging patterns and its incremental maintenance algorithms. In: ICML (2000)
20. Ramamohanarao, K., Bailey, J., Fan, H.: Efficient mining of contrast patterns and their applications to classification. In: Intelligent Sensing and Information Processing, 2005. ICISIP 2005. Third International Conference, IEEE (2005)
21. Fan, H., Ramamohanarao, K.: Fast discovery and the generalization of strong jumping emerging patterns for building compact and accurate classifiers. IEEE Trans. Knowl. Data Eng. **18**(6), 721–737 (2006)
22. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed optimization by ant colonies. In: Proceedings of the First European Conference on Artificial Life, Paris (1991)
23. Zhang, X., Dong, G., Kotagiri, R.: Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM (2000)
24. Li, J., et al.: Deeps: a new instance-based lazy discovery and classification system. Mach. Learn. **54**(2), 99–124 (2004)
25. Wang, Z., Fan, H., Ramamohanarao, K.: Exploiting maximal emerging patterns for classification. Australasian Joint Conference on Artificial Intelligence. Springer, Berlin (2004)
26. Podraża, R., Tomaszewski, K.: KTDA: emerging patterns based data analysis system. Ann. UMCS Sect. AI Inform. **4**(1), 279–290 (2006)
27. Alhammady, H.: A novel approach for mining emerging patterns in data streams. In: Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium, IEEE (2007)
28. Ceci, M., Appice, A., Malerba, D.: Discovering emerging patterns in spatial databases: a multi-relational approach. Knowledge Discovery in Databases: PKDD 2007, pp. 390–397. Springer, Berlin (2007)
29. Ceci, M., Appice, A., Malerba, D.: Emerging pattern based classification in relational data mining. Database and Expert Systems Applications. Springer, Berlin (2008)
30. Poezevara, G., Cuissart, B., Crémilleux, B.: Discovering emerging graph patterns from chemicals. Foundations of Intelligent Systems, pp. 45–55. Springer, Berlin (2009)
31. Gu, T., et al.: epsicar: an emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In: Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference, IEEE (2009)
32. Chen, X., Lu, L.: An improved algorithm of mining Strong Jumping Emerging Patterns based on sorted SJEP-Tree. In: Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference, IEEE (2010)
33. Li, H.-F., Chen, H.-S.: Discovering emerging melody patterns from customer query data streams of music service. In: Multimedia and Expo (ICME), 2011 IEEE International Conference, IEEE (2011)
34. Muyeba, M.K., et al.: A framework to mine high-level emerging patterns by attribute-oriented induction. International Conference on Intelligent Data Engineering and Automated Learning. Springer, Berlin (2011)
35. Liu, Q., et al.: A novel approach of mining strong jumping emerging patterns based on BSC-tree. Int. J. Syst. Sci. **45**(3), 598–615 (2014)
36. Parmar, H., Chand, C.: Improved high growth-rate emerging pattern based classification. Int. J. Comput. Sci. Mob. Comput. **4**, 479–490 (2015)
37. Gambin, T., Walczak, K.: Classification based on the highest impact jumping emerging patterns. In: Computer Science and Information Technology, 2009. IMCSIT'09. International Multiconference, IEEE (2009)
38. Vyas, Z.V., et al.: Modified RAAT (reduced Apriori Algorithm using tag) for efficiency improvement with EP (emerging patterns) and JEP (Jumping EP). In: Advances in Computer Engineering (ACE), 2010 International Conference, IEEE (2010)
39. García-Borroto, M., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: A New emerging pattern mining algorithm and its application in supervised classification. Advances in Knowledge Discovery and Data Mining, pp. 150–157. Springer, Berlin (2010)
40. García-Borroto, M., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: Cascading an emerging pattern based classifier. Advances in Pattern Recognition, pp. 240–249. Springer, Berlin (2010)

41. Wang, L., Wang, Y., Zhao, D.: Building emerging pattern (EP) random forest for recognition. In: Image Processing (ICIP), 2010 17th IEEE International Conference, IEEE (2010)
42. García-Borroto, M., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: Fuzzy emerging patterns for classifying hard domains. Knowl. Inf. Syst. **28**(2), 473–489 (2011)
43. Yu, H.-H., Chen, C.-H., Tseng, V.S.: Mining emerging patterns from time series data with time gap constraint. Int. J. Innov. Comput. Inf. Control **7**(9), 5515–5528 (2011)
44. Yu, K., et al.: Mining emerging patterns by streaming feature selection. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM (2012)
45. Dorigo, M.: Optimization, learning and natural algorithms. Ph. D. Thesis, Politecnico di Milano, Italy (1992)
46. Parpinelli, R.S., Lopes, H.S., Freitas, A.A.: Data mining with an ant colony optimization algorithm. IEEE Trans. Evol. Comput. **6**(4), 321–332 (2002)
47. Liu, B., Abbass, H.A., McKay, B.: Density-based heuristic for rule discovery with ant-miner. In: The 6th Australia-Japan Joint Workshop on Intelligent and Evolutionary System (2002)
48. Liu, B., Abbass, H.A., McKay, B.: Classification rule discovery with ant colony optimization. In: IAT (2003)
49. Martens, D., et al.: Classification with ant colony optimization. IEEE Trans. Evol. Comput. **11**(5), 651–665 (2007)
50. Baig, A.R., Shahzad, W.: A correlation-based ant miner for classification rule discovery. Neural Comput. Appl. **21**(2), 219–235 (2012)
51. Shahzad, W., Baig, A.: Hybrid associative classification algorithm using ant colony optimization. Int. J. Innov. Comput. Inf. Control **7**(12), 6815–6826 (2011)
52. Otero, F.E., Freitas, E.E., Johnson, C.G.: cAnt-Miner: an ant colony classification algorithm to cope with continuous attributes. International Conference on Ant Colony Optimization and Swarm Intelligence. Springer, Berlin (2008)
53. Alcala-Fdez, J., et al.: KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput. **13**(3), 307–318 (2009)
54. Bay, S.D., et al.: The UCI KDD archive of large data sets for data mining research and experimentation. ACM SIGKDD Explor. Newsl. **2**(2), 81–85 (2000)
55. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
56. Salzberg, S.L.: C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. Mach. Learn. **16**(3), 235–240 (1994)
57. Schölkopf, B., et al.: New support vector algorithms. Neural comput. **12**(5), 1207–1245 (2000)
58. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
59. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. (1998)
60. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the Twelfth International Conference on Machine Learning (1995)
61. Sousa, T., Silva, A., Neves, A.: Particle swarm based data mining algorithms for classification tasks. Parallel Comput. **30**(5), 767–783 (2004)
62. McLachlan, G.: Discriminant Analysis and Statistical Pattern Recognition, vol. 544. Wiley, Hoboken (2004)
63. Le Cessie, S., Van Houwelingen, J.C.: Ridge estimators in logistic regression. Applied Statistics **41**, 191–201 (1992)
64. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Mach. Learn. **29**(2–3), 103–130 (1997)

Zulfiqar Ali is PhD (Computer Science) candidate in National University of Computer & Emerging Science (NU-FAST), Islamabad, Pakistan. He has done MS (Computer Science) from same (NU-FAST) and BS (Computer Science) from Allama Iqbal Open University (AIOU), Islamabad, Pakistan. Now he is working as a faculty member (Lecturer) in the department of computer science, in The University of Lahore, Pakistan.His research area is Data Mining, Computational Intelligence, Swarm Intelligence and Machine Learning.

Dr. Waseem Shahzad is currently working as Associate Professor at the department of computer science of FAST-National University of Computer and Emerging Sciences, Islamabad. He is interested in conducting research in areas related to Machine Learning, Data mining and Knowledge Discovery, Data Warehousing and Evolutionary Computing. Currentlyhe is working on classification, associative classification and feature subset selection using Swarm Intelligence techniques. He has several research publications in these areas.