Class Dependent Feature Construction as a Bi-level Optimization Problem

Marwa Hammami[a], Slim Bechikh[a], Mohamed Makhlouf[b], Chih-Cheng Hung[c], Lamjed Ben Said[a]
[a]SMART lab, University of Tunis, ISG, Tunis, Tunisia
[b]Kedge Business School, Talence, France
[c]Kennesaw State University, USA & Anyang Normal, University, China
slim.bechikh@fsegn.rnu.tn

***Abstract*—Feature selection and construction are important pre-processing techniques in data mining. They allow not only dimensionality reduction but also classification accuracy and efficiency improvement. While feature selection consists in selecting a subset of relevant features from the original feature set, feature construction corresponds to the generation of new high-level features, called constructed features, where each one of them is a combination of a subset of original features. However, different features can have different abilities to distinguish different classes. Therefore, it may be more difficult to construct a better discriminating feature when combining features that are relevant to different classes. Based on these definitions, feature construction could be seen as a BLOP (Bi-Level Optimization Problem) where the feature subset should be defined in the upper level and the feature construction is applied in the lower level by performing mutliple followers, each of which generates a set class dependent constructed features. In this paper, we propose a new bi-level evolutionary approach for feature construction called BCDFC that constructs multiple features which focuses on distinguishing one class from other classes using Genetic Programming (GP). A detailed experimental study has been conducted on six high-dimensional datasets. The statistical analysis of the obtained results shows the competitiveness and the outperformance of our bi-level feature construction approach with respect to many state-of-art algorithms.**

***Index Terms*—Class dependent features, features construction, bi-level optimization, evolutionary algorithms.**

I. Introduction

Classification is one of the important tasks in machine learning and data mining, which aims to classify each instance in the dataset into different classes based on its features. It is difficult to determine which features are useful without a prior knowledge. However, not all features in a feature vector are essential since many of them are irrelevant and redundant, which may negatively affect the classification accuracy and reduce the quality of the whole feature set due to the large search space known as "the curse of dimensionality" [1].

Feature selection and feature construction can be performed through the wrapper, filter or embedded approach [2], which differ in their evaluations. While the wrapper methods use learning techniques to evaluate which features are useful, filter methods use the intrinsic characteristics of the training data to evaluate features. Wrapper methods usually require a high demand of computation time, but the features selected or constructed by the wrapper methods usually achieve higher classification accuracy than those generated by the filter methods. Embedded methods simultaneously combine the feature selection/construction step and learning a classifier [4]. This process is typically faster than that of wrapper methods.

Feature selection and feature construction are popular methods used to enhance the quality of feature space [3]. Feature selection aims at selecting relevant features from the original feature set. Feature construction selects informative features and combines them to constructing new high-level features that may provide better discrimination for the problem [28]. However, different features can have different abilities to distinguish different classes [5]. For example, a feature may be good at distinguishing samples of class A from those of class B, C and D, but may not be good at differentiating samples of class B from those of C and D. Therefore, it may be more diffcult to construct a better discriminating feature when combining features that are relevant to different classes. Feature construction is still a very challenging task. This could be explained by the large search space of feature combinations, whose size is a function of the number of features. Therefore, finding the optimal combination for each class is expected to achieve a good performance.

Bi-level optimization is an important research area of mathematical programming [6]. It has emerged as an important field for progress in handling many real life problems in different domains such as classification and machine learning [2, 6]. The BLOP is a hierarchy of two optimization tasks (upper level or leader, and lower level or follower problems). The lower level task appears as a constraint such that only an optimal solution to the lower level problem is a possible feasible candidate to the upper level one. In this context, feature selection and construction can be treated as a bi-level optimization problem by performing feature selection in the upper level and feature construction in the lower level. Each upper level solution (feature subset) is associated with a set of optimal class-dependent feature subset combinations.

To the best of our knowledge, there is no work in the literature that considers class dependent feature construction as a bi-level optimization problem. The main idea of our paper is to evolve an upper level population for the task of feature selection, while optimizing the feature construction at the lower level by evolving multiple followers population. It is worth noting that for each upper level individual (feature subset), multiple lower level populations are optimized to

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

find the corresponding (near) optimal feature combination for each class (class dependent constructed feature). In this way, BCDFC would be able to output a set of optimized constructed features for each class. The principal contributions of our paper are the following:

1) Proposing a new bi-level evolutionary approach for feature construction, called BCDFC.
2) Adapting an existing algorithm named CODBA to our problem to obtain optimal class dependent contructed features.
3) Reporting experimental results with respect to the state-of-the-art algorithms.

The rest of this paper is organized as follows. Section 2 presents the related work in this research area. Section 3 describes the bi-level evolutionary approach for class dependent feature construction. Section 4 gives the experimental results in this study. Finally, section 5 concludes the paper and provides future perspectives.

II. BACKGROUND AND RELATED WORK

A. Feature selection and feature construction

Evolutionary Computation (EC) techniques have been used to address feature selection tasks [7]. In this context, Zhu et al. [9] propose a feature selection method using a memetic algorithm called WFFSA. In their algorithm, individual features were ranked first according to a filter ranking method. Genetic Algorithm (GA) is used to add or delete a feature based on the ranked individual features. In [10], Hammami et al. designed a multi-objective hybrid evolutionary approach for feature selection under a memetic framework by developing a new IBMOLS in order to reduce the number of wrapper evaluations.

Feature selection does not create new features. However, if the original features are not informative enough to achieve promising performance, feature construction may complement for the problem. In this context, an automated feature construction and selection framework for biological sequences classification was proposed [11]. This approach uses a two-stage process to construct a set of candidate sequence-based features first and then select a most effective subset for the classification task at hand. Tran et al. [12] proposed a GP-based method that simultaneously performs multiple feature construction and feature selection to automatically transform high-dimensional datasets into much smaller set. The constructed features are evaluated by a hybrid weighted-sum objective function. Recently, Hammami et al. [13] proposed a hybrid filter-wrapper multi-objective evolutionary approach for feature construction. Only non-dominated (best) feature subsets are improved using an indicator-based local search that optimizes three objective functions.

The evaluation measure is one of the key factors in EC for feature selection and construction. A majority of the computational time is spent on the wrapper evaluation procedure and many filter approaches. In the literature, there are some existing fast evaluation measures such as mutual information [14]. In this paper, as a specific measure we use the mutual information, which quantifies the amount of dependence between two random variables [27]. The entropy is a measure of the uncertainty of a discrete random variable X. It is defined as:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{1}$$

where $p(x) = Pr(X = x)$ is the probability density function of X. Note that entropy does not depend on actual values, just the probability distribution of the random variable. When a certain variable is known and others are unknown, the remaining uncertainty is measured by the conditional entropy. Assume that variable Y is given, the conditional entropy $H(X|Y)$ of X with respect to Y is:

$$H(X|Y) = -\sum_{x \in X, y \in Y} p(x,y) \log_2 p(x|y) \tag{2}$$

If X completely depends on Y, then $H(X|Y)$ is zero, which means that no more other information is required to describe X when Y is known. On the other hand, $H(X|Y) = H(X)$ denotes that knowing Y will do nothing to observe X, i.e. they are fully independent or unrelated.

Mutual information, $I(X;Y)$, defines the information shared between two random variables. Given variable X, how much information one can gain about variable Y, which is defined as:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= -\textstyle\sum_{x \in X, y \in Y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \end{aligned} \tag{3}$$

The mutual information $I(X;Y)$ will be large if two variables X and Y are closely related. Otherwise, $I(X;Y)$ will be zero if X and Y are totally unrelated.

B. Bi-level optimization basic definitions

In this section, we provide a general formulation for bi-level optimization problem. A BLOP could be seen as a combination of two optimization problems where the lower level one appears as a constraint of the upper level one [6]. There are two classes of variables for a BLOP, namely, the upper level (leader) variables x_u and the lower level (follower) variables x_l (cf. Fig. 1). It is important to note that for the lower level problem, the optimization task is performed with respect to the variables x_l and the variables x_u acting as parameters. Consequently, for each x_u corresponds a different lower level problem whose optimal solution needs to be

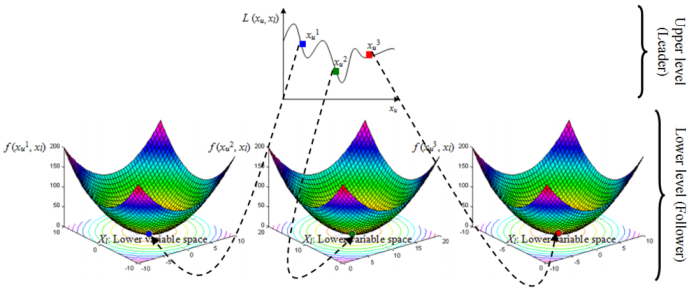

Fig. 1. Illustration of a bi-level optimization problem [16].

determined. All variables (x_u and x_l) are considered in the upper level problem, and the optimization is expected to be performed with respect to both sets of variables. The BLOP can be formally stated as follows:

$$\begin{aligned} &\min_{x_u \in X_U, x_l \in X_L} L(x_u, x_l) \\ &subject\ to \\ &x_l \in ArgMin\{f(x_u, x_l), g_j(x_u, x_l) \leq 0, j = 1, \ldots, J\} \\ &G_k(x_u, x_l) \leq 0, k = 1, \ldots, K \end{aligned} \tag{4}$$

where g_j represents the constraint set of the lower level problem, and G_k denotes the constraint set of the upper level one. The difficulty in bi-level optimization arises from the fact that only the optimal solutions of the lower level optimization task may be acceptable as possible feasible candidates to the upper level one. For example, a member $x^1 = ({x_u}^1, {x_l}^1)$can be considered feasible at the upper level only if x^1 satisfies the upper level constraints, and ${x_l}^1$ is an optimal solution to the lower level problem corresponding to ${x_u}^1$.

To sum up, a variety of feature selection and construction approaches has been proposed, but the use of bi-level model for solving class dependent feature construction and selection problem has not yet been investigated. Therefore, the development of a bi-level evolutionary approach for feature construction is still an open issue. This paper presents the first study of bi-level class dependent method for feature construction on high-dimensional data using GP.

III. The proposed Bi-level Class Dependent Feature Construction (BCDFC)

A. Main idea and motivations

Feature selection selects relevant features from the original feature set, which could be not informative enough to achieve good performance. Therefore, feature construction may work well as it creates new features. Constructing informative features is a challenging task. Since different features can have different abilities to distinguish different classes, it may be more difficult to construct a better discriminating feature when combing features that are relevant to different classes. In this study, we propose an original bi-level hybrid filter-wrapper evolutionary approach for multiple feature construction on high-dimensional data that takes into account class dependency in the feature construction process. The main contribution of our paper is to show that a bi-level model can be used efficiently to solve feature selection and construction problem. A schematic of BCDFC showing the interplay between the upper and the lower level, is shown in Fig. 2. The BLOP is applied to perform feature selection and consequently produce optimal feature subset combinations for each class. The proposed BCDFC consists of two stages and each stage employs an EA. In the upper level, the Evolutionary Feature Selection (EFS) algorithm is used to select a subset of features deemed most informative without sacrificing performance. These subset of features are ranked using the relavance measure (t-Test) in Equation 5. These terminal sets are then input to the lower level, where a second Evolutionary Class Dependent Feature Construction (CDFC) algorithm produces multiple combinations of terminal sets. Each upper level solution is associated with multiple followers, each of which corresponds to one class. The number of followers is the number of classes. Each follower produces a sequence of terminal set combinations. Only the optimal combination for each class in terms of mutual information is hereafter submitted to the upper level to perform the evaluation process. Finally, a set of optimal class dependent constructed features are retreived. The upper level seeks to optimize a combined filter-wrapper objective including the number of features and the classification accuracy, while the lower level optimizes one filter objective, namely the mutual information. Due to the high computational cost of bi-level optimization algorithms, we have used CODBA as a recently effective and efficient evolutionary bi-level algorithm for combinatorial optimization [17]; which uses decomposition, multi-threading, and co-evolution to reduce the lower level computational cost as possible.

B. UGAFS: Upper level GA for Feature Selection

1) Solution encoding: In the upper level, a representation for candidate feature subset is encoded as a chromosome in such a way that each bit encodes a single feature: $S = F_1\ F_2\ F_3 ... F_i ... F_n$. In fact, each individual in the population, i.e. a subset of features, is represented by a vector of n bits where each bit can takes the value of 1 or 0. Where n is the number of features. In Fig. 3, "1" represents that the corresponding feature is selected and "0" otherwise.

2) Class dependent feature subset relevance: Because the goal of a constructed feature is to differentiating instances of a class from the other classes, it should be constructed based on class relevant features. That is, different sets of features should be selected for the feature construction process of different classes. Thus, lower level trees will have different terminal sets including features that are relevant to the targeted class only. A feature f is relevant to class c if its values appeared in class c are considerably different from its values in other classes. In BCDFC, we will use the t-Test to measure the relevance of a feature in relation to a class as proposed in [20]. Firstly, values of a feature f will be splitted into two subsets, one including values belonging to class c and one from other classes. Thereafter, the relevance measure $Rel_{f,c}$ is computed based on Equation 5. $Rel_{f,c}$ is equal to 0 if the two groups are not significantly different (i.e. p-value≥ 0.05). Otherwise, it is equal to the absolute of t-value divided by p-value. The larger the value of $Rel_{f,c}$, the more relevant the feature f to class c [20].

$$Rel_{f,c} = \begin{cases} 0, & if p-value \geq 0.05 \\ \frac{|t-value(f_{class=c}, f_{class \neq c}|}{p-value}, & otherwise \end{cases} \tag{5}$$

In the upper level, features are ranked by its $Rel_{f,c}$ values for each class c. Then the top-ranked features will be used to form the terminal set of class c. By doing so, we not only eliminate irrelevant features but also narrow the search space so that the searching process will be more efficient.

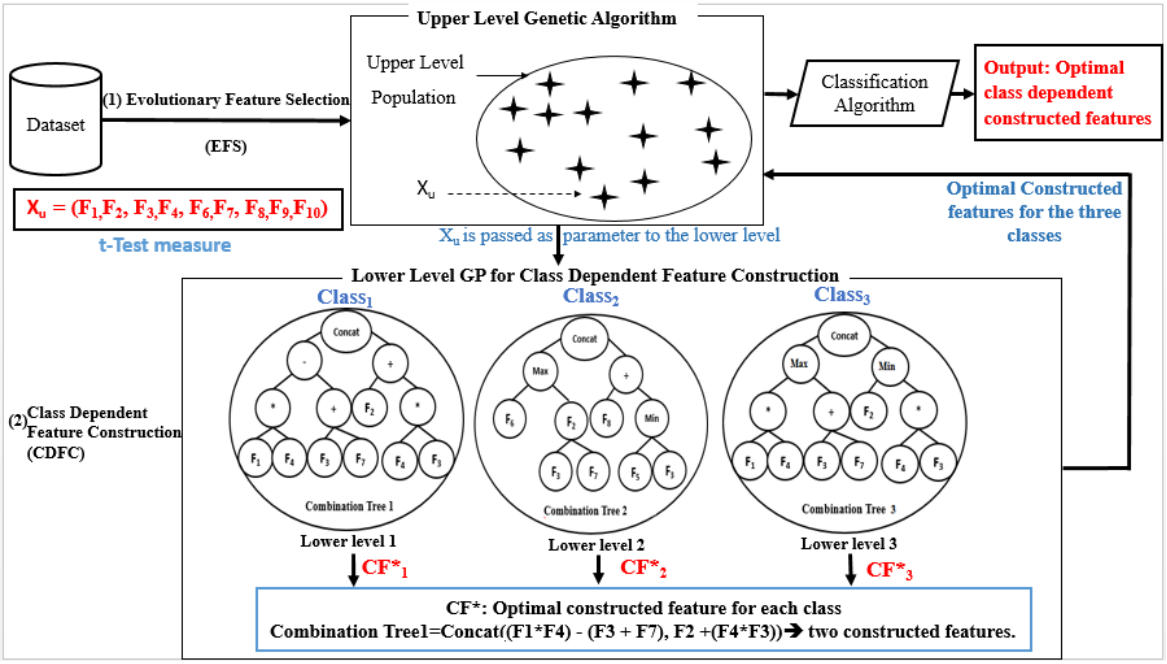

Fig. 2. General algorithmic scheme of BCDFC.

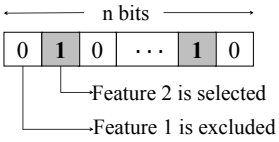

Fig. 3. Upper level solution representation.

3) Fitness function: In the upper level, two well-known measures, the number of features and the classification accuracy, are used to form the new fitness function. The upper level combined fitness function is represented in Equation 6. Equation 6 seeks to minimize the number of features and maximize the classification accuracy, where $|X|$ represents the number of selected/constructed features and Acc is the classification accuracy, which depends on the number of samples correctly classified (true positives plus true negatives).

$$F_u(X) = \frac{1}{|X|} + Acc \tag{6}$$

The classification accuracy Acc is computed according to Equation 7, where t is the number of samples correctly classified, and N is the total number of samples.

$$Acc = \frac{t}{N} \times 100 \tag{7}$$

According to a previous study [12], we use the classification accuracy estimated from K-fold ($K = 3$) cross validation on the training set. This K-fold CV is repeated L times ($L = 3$) with different data splitting similar to [18] in order to avoid overfitting. In total, $K \times L$ models were constructed to evaluate the set of new constructed features. We use the balanced accuracy [18] since there is some unbalance in many high-dimensional datasets. This balanced accuracy is shown in Equation 8 in which c is the number of classes, TP_i and T_i are the number of correctly identified instances and the number of total instances of class i, respectively.

$$BalancedAcc = \frac{1}{c}\sum_{i=1}^{c}\frac{TP_i}{|T_i|} \tag{8}$$

Each lower and upper level objective function is normalized in the range of [0,1] [19]. The normalization function is defined according to Equation 9.

$$f_i^{norm} = \frac{f_i(x) - f_i^{min}}{f_i^{max} - f_i^{min}} \tag{9}$$

where f_i^{min} is the minimum objective function value and f_i^{max} is the maximum value of the objective function.

C. LGPCDFC: Lower level GP for Class Dependent Feature Construction

1) Solution encoding: In this study, we aim to construct multiple features. Each constructed feature seeks to discriminate one class from the other classes. BCDFC enables to construct multiple features for one class based on a user given ratio r controlling the number of trees for each class per individual. The number of constructed features cf is equal to r multiplied by the number of classes. In the bi-level context, each feature vector (i.e. chromosome) in the upper level has cf trees. Each tree is a sequence of feature subset combinations (concatenation of combinations). Fig. 4 shows an example of 6 constructed features with $r = 2$ for a three-class problem. A detailed discussion of why and how the parameter cf is applied is given in [13].

- Constructed feature set: $CF1_1, CF1_2, CF2_1, CF2_2, CF3_1, CF3_2$.

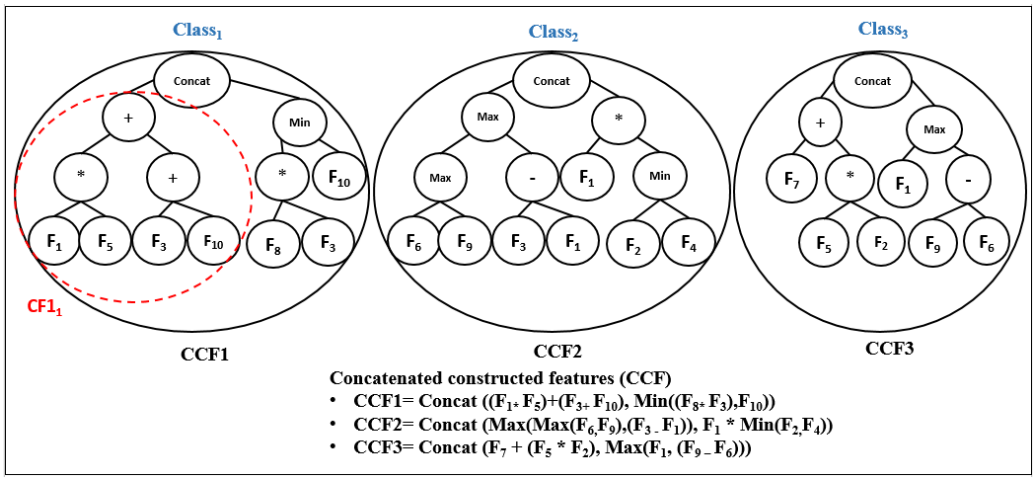

Fig. 4. Lower level solution representation.

- Selected feature set: $F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9$, and F_{10}.

2) Fitness function: The goal of using filter approaches is to speed up the fitness evaluation procedures. So a computationally cheap measure, mutual information, is employed here to form the filter evaluation. The filter fitness function aims to maximise the relevance of the selected features to the class labels. The filter objective function is described in Equation 10, where x is an individual feature in X, and c is the class label. The details of the mutual information I are provided in section II.

$$F_l(X) = \sum_{x \in X} I(x; c) \tag{10}$$

IV. Experimental Study

This section gives detailed experimental study on the datasets used to evaluate the performance of BCDFC and compare with those of the competitor algorithms including HybridGPFC, MCIFC, and CDFC.

A. Datasets

Six high-dimensional gene datasets [1] with thousands to tens of thousands of features are used in the experiments. The dataset description is given in Table I. The last column shows the class distribution of the data which is the percentage of instances in each class. These datasets are unbalanced data due to the significant difference between the percentages of the class-distribution.

We used discretization for all datasets, since they are biological data. Each feature is discretized into three category values (-1, 0 and 1) using mean (μ) and standard deviation (σ) of the feature values. Values that fall in the interval $[(\mu - \sigma)/2 ... (\mu + \sigma)/2]$ are transformed to state 0. x will be set to -1 if $x < (\mu - \sigma)/2$ and set to 1 if $x > (\mu + \sigma)/2$. For more details about discretization methods for biological data, please refer to [21]. .

[1]These datasets are available at http://www.gems-system.org, and http://csse.szu.edu.cn/staff/zhuzx/Datasets.html

TABLE I
Datasets

Dataset	#Features	#Classes	#Instances	Class-Distribution
Colon	2000	2	62	35% - 65%
DLBCL	5469	2	77	25% - 75%
CNS	7129	2	60	35% - 65%
Prostate	10509	2	102	50% - 50%
Leukemia1	5327	3	72	13%-35%-53%
SRBCT	2308	4	83	13%-22%-30%-34%

B. Competitor algorithms and parameter setting

To show the effectiveness of the proposed method, we compare the classification accuracy of the constructed features with those produced by a multiple feature construction algorithm named HybridGPFC [13], a class independent feature construction algorithm called MCIFC [22], a class dependent feature construction algorithm called CDFC [20], as well as the original feature set using the average test accuracy of KNN, NB and DT. According to a previous study [24], we use $K = 5$. Since the number of instances in each datasets is very small, we split these datasets into 10 folds and perform 10-fold cross validation (10-CV) [23] to generate training and test set for evaluating BCDFC performance. During the feature construction process, 3-fold CV within the training set is used to evaluate the constructed features (see Section 3 (B)). As GP is a stochastic algorithm, 30 independent GP runs with 30 different random seeds were conducted for each training set to remove statistical variations.

The BCDFC parameters setting are described in Table II which has the same parameters as used in [15]. The function set comprises of 3 arithmetic operators ($+, -, \times$). The function $min(x_1, x_2)$ returns the minimum of two inputs and the function $max(x_1, x_2)$ returns the maximum of two inputs. The function if function takes three values and returns the second if the first value is greater than zero, otherwise it returns the third value. $Concat(CFT1_1, CFT1_2)$ returns the concatenation of two constructed features trees. cf is the number of maximum constructed features in a single tree. cf is the CF_Ratio multiplied by the number of classes. CF_Ratio is a ratio controlling the number of trees for each class per lower level individual. To ensure fairness of comparison, we

use the same number of function evaluations, which is set to 6250000, for all the algorithms. Experiments were run on a PC with Intel Core i7-4770 CPU @ 3.4 GHz, programming in Java 1.8 with a total memory of 8GB.

TABLE II
GP SETTINGS USED IN BCDFC

Function set	$+, -, /, \times, min\ max, if, concat$
Terminal set	Features values
Upper level population	50
Lower level population	50
Upper level generation	50
Lower level generation	50
Stopping criterion	6250000 evaluations
Maximum Tree Depth	8
cf	CF_Ratio×Nbr Classes
CF_Ratio	2
Selection Method	Tournament Method
Tournament Size	7
Crossover Rate	0.8
Mutation Rate	0.2
Elitism Size	1

C. Performance metrics and statistical testing approach

In this paper, we aim to compare the results using the following two indicators to ensure that both selected and constructed features are improved using the proposed algorithm:

- The number of features,
- The classification accuracy.

The classification accuracy of the selected/constructed features will be calculated on the test set according to Equation (11). It is the ratio of the number of correct predictions to the total number of input samples.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{11}$$

Wilcoxon test [25] is used in our case with the significance level of 0.05 to compare the classification accuracy achieved by using all the features for classification, and those of the features constructed by BCDFC and the competitor algorithms.

V. RESULTS AND DISCUSSIONS

Table III shows the experimental result of the BCDFC constructed features compared with Full (i.e. using the original feature set), HybridGPFC, MCIFC, and CDFC, where column "#F" shows the average size of each feature set. Columns of "B-K-NN", "B-NB", and "B-DT" represent the best results of K-Nearest Neighbor (K-NN), Naïve Bayes (NB), and Decision Trees (DT), respectively. All columns of "A$\pm$Std-K-NN", "A$\pm$Std-NB", and "A$\pm$Std-DT" represent the average and the standard deviation of the accuracy. Each column of "A$\pm$Std" was achieved through 30 independent runs obtained by K-NN, NB and DT using the full feature set "Full", the constructed feature by HybridGPFC, MCIFC, and CDFC. The Wilcoxon significance test results for KNN, NB and DT are displayed in column S1, S2, and S3, respectively. Symbol "+" or "−" means that the result is significantly better or worse than the proposed algorithm and symbol "=" means they are similar. In other words, the more "−", the better the proposed method. The numbers under the dataset name is the number of instances in the dataset.

A. Performance of the constructed features

Table III shows the results of constructed features obtainted by the BCDFC compared to Full, HybridGPFC, MCIFC, and CDFC. It can be seen that the number of features constructed by the BCDFC is negligible compared with the original feature size. The " $-$ " marks appeared in column S_1 of Table III show that the constructed features help KNN achieve significantly higher accuracy than using full feature sets on all datasets. The highest improvement is on CNS dataset with 50% on average and 47% in the best case. For NB, the constructed features by BCDFC obtain better performance than Full on almost all datasets. The largest improvement that NB achieves is on Prostate dataset with 61% on average and 63% in the best case. DT using features constructed by BCDFC also has significantly better performance on four datasets compared to the Full. DT has a considerable increase on Prostate with 22% on average. BCDFC obtains about 3% and 5% lower average accuracy than Full on DLBCL and SRBCT.

Compared to the constructed features by the HybridGPFC, the constructed features by BCDFC help K-NN, NB and DT obtain a significantly better results on almost all datasets. The highest improvement of 21% on average is found on Prostate using KNN. Similarly, the constructed features by the BCDFC obtain higher performance than that of MCIFC on all datasets except for Colon dataset with DT but its best accuracy always higher than the best accuracy obtained by MCIFC. BCDFC achieves better accuracy on almost all datasets than using those constructed by CDFC except for DLBCL dataset with DT. The best accuracy of BCDFC is 32% increase on Prostate using DT compared with CDFC. This improvement could be due to the ability of BCDFC to find the optimal combination of the constructed features formed by terminal sets with class-relevant features.

To sum up, the BCDFC outperforms the competitor algorithms for the three classification algorithms on almost all datasets. The results indicate that the use of the bi-level model enables GP to produce optimal class dependent combinations for each feature subset and consequently construct a small number of features with high discrimination ability and generalised well to the three learning algorithms in almost all datasets. The outperformance of our algorithm over the three peer algorithms could be explained by the main distinction of our BCDFC that consists in optimizing the feature construction for each class at the lower level; which is not the case for the three other algorithms that use a single level of optimization.

B. Evaluation of BCDFC features splitting ability

Hierarchical clustering is an agglomerative clustering algorithm that yields a dendrogram, which can be cut at a chosen height to produce the desired number of clusters [26]. Fig. 5 shows four dendrogram plots of the hierarchical clustering of twenty observations on mtcars dataset. The distance of split or merge (called height) is shown on the y-axis of the dendrogram. The similarity between the clusters is often calculated from the dissimilarity measures like the Euclidean distance between two clusters. Thus, the larger the distance

TABLE III
BEST, AVERAGE AND STD OF THE ACCURACY OF THE SELECTED FEATURES. BEST-VALUES ON EACH DATASET ARE MARKED IN BOLD.

Dataset	Subset	#F	B-K-NN	A±Std-K-NN	S_1	B-NB	A±Std-NB	S_2	B-DT	A±Std-DT	S_3
Colon	Full	2000	73.27	72.20±1.00	(−)	72.80	71.80±2.00	(−)	74.42	74.42±0.00	(−)
(62)	HybridGPFC	30	83.42	75.30±4.02	(−)	84.03	74.28±3.18	(−)	85.95	84.96±4.66	(=)
	MCIFC	21	83.68	74.45±3.30	(−)	78.45	71.36±0.23	(−)	87.25	87.01±4.00	(+)
	CDFC	16	82.20	73.05±4.02	(−)	80.95	80.62±2.65	(−)	80.22	70.89±4.14	(−)
	BCDFC	18	**95.43**	**80.92±1.30**		**90.77**	**84.00±2.76**		**92.89**	**85.49±2.41**	
DLBCL	Full	5469	84.36	81.35±2.00	(−)	81.23	81.23±0.00	(−)	89.12	88.22±0.00	(+)
(77)	HybridGPFC	34	87.54	80.31±0.10	(−)	96.25	83.23±3.42	(−)	95.75	83.94±5.16	(−)
	MCIFC	33	95.07	81.77±2.54	(−)	90.58	85.95±2.13	(−)	**97.55**	86.00±3.47	(−)
	CDFC	31	90.60	82.82±5.48	(−)	87.45	85.15±3.53	(−)	95.57	**93.22±3.41**	(+)
	BCDFC	36	**98.92**	**97.99±2.85**		**98.04**	**96.61±2.00**		90.81	85.03±3.12	
CNS	Full	7129	59.12	58.12±2.00	(−)	57.93	59.93±1.00	(−)	72.03	71.03±0.00	(−)
(60)	HybridGPFC	41	70.56	70.16±3.28	(−)	70.53	60.87±3.34	(−)	72.76	60.34±5.27	(−)
	MCIFC	52	74.23	56.37±2.42	(−)	64.98	60.83±2.37	(−)	70.72	59.22±4.60	(−)
	CDFC	55	80.31	73.39±1.40	(−)	84.70	82.14±2.76	(−)	80.34	79.02±4.44	(−)
	BCDFC	49	**87.77**	**87.53±0.18**		**89.34**	**88.47±3.10**		**88.23**	**84.91±2.23**	
Prostate	Full	10509	81.15	76.15±1.73	(−)	60.05	60.55±1.00	(−)	86.08	80.49±0.00	(−)
(102)	HybridGPFC	54	90.00	80.15±4.27	(−)	89.54	86.06±2.76	(−)	90.08	81.22±3.19	(−)
	MCIFC	46	88.38	80.12±0.54	(−)	85.75	80.22±2.82	(−)	91.10	85.11±2.63	(−)
	CDFC	46	90.99	84.62±2.86	(−)	86.88	76.31±2.25	(−)	80.78	74.32±3.29	(−)
	BCDFC	38	**97.86**	**97.28±1.49**		**97.98**	**97.93±1.15**		**99.69**	**98.99±4.00**	
Leukemia1	Full	7129	88.03	76.03±1.00	(−)	90.96	90.96±0.00	(−)	91.21	90.21±5.00	(−)
(72)	HybridGPFC	69	92.00	79.31±3.27	(−)	95.25	85.42±3.42	(−)	91.15	81.04±5.87	(−)
	MCIFC	85	95.57	83.24±2.16	(−)	90.57	84.95±1.56	(−)	90.17	82.15±3.87	(−)
	CDFC	50	79.06	85.77±2.14	(−)	90.00	85.99±3.34	(−)	83.17	88.77±2.55	(−)
	BCDFC	32	**98.99**	**98.40±4.47**		**98.70**	**96.47±1.00**		**99.25**	**96.11±1.47**	
SRBCT	Full	15154	89.07	88.07±1.73	(−)	85.05	82.05±2.00	(−)	**98.77**	**97.99±2.00**	(+)
(83)	HybridGPFC	66	98.40	88.75±0.16	(−)	96.22	93.35±0.23	(−)	98.12	91.29±1.63	(−)
	MCIFC	71	90.84	86.20±1.79	(−)	94.62	79.61±0.22	(−)	90.20	80.16±0.11	(−)
	CDFC	45	92.30	90.10±4.75	(−)	89.89	89.19±2.47	(−)	86.98	85.09±3.25	(−)
	BCDFC	42	**100.00**	**97.79±2.28**		**99.82**	**99.00±3.95**		95.00	92.26±2.57	

B/A/Std: Best/Average/standard deviation of the accuracy; +/−/=: means that the result is significantly better/worse/similar than using all features.

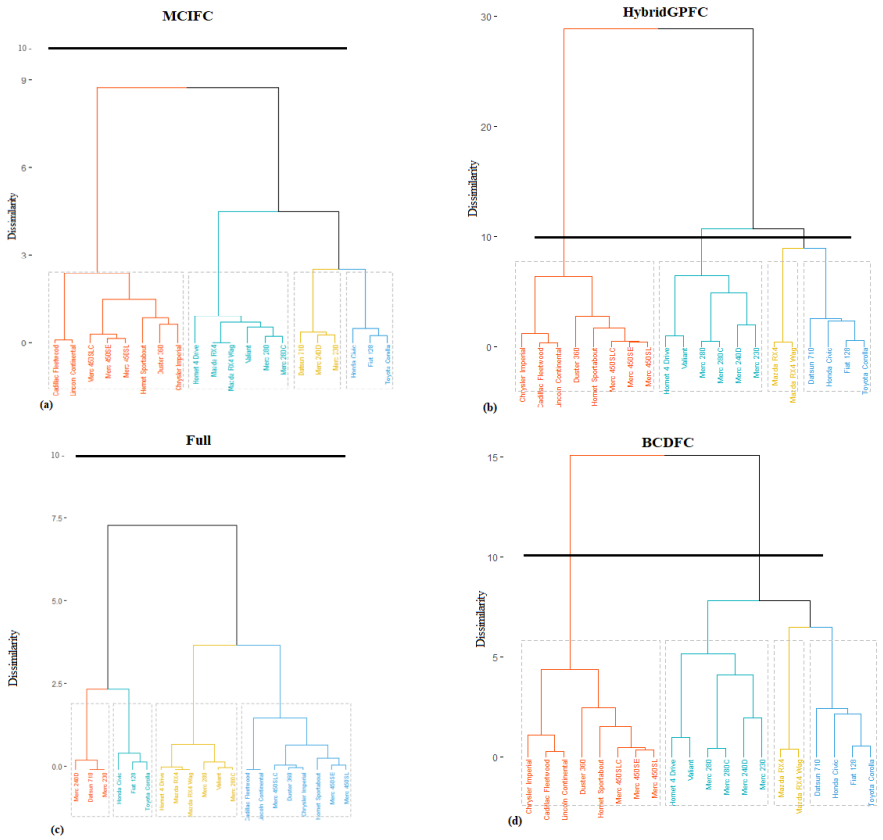

Fig. 5. Dendogram plots of the constructed features-based hierarchical clustering on mtcars dataset: (a) MCIFC, (b) HybridGPFC, (c) Full, and (d) BCDFC.

between two clusters is, the better the splitting is. We have used the Euclidean distance to calculate the dissimilarity measure (height) and the Ward's minimum variance agglomeration method to cluster the dataset. Using dendrograms, we compare our BCDFC constructed features to: (1)MCIFC constructed features (b)HybridGPFC constructed features , and (3)the full set of the original features. Fig. 5 shows that instances grouped together in dendrogram (d) are closer to each other than they are in the other dendrograms. Moreover, the dendrogram of the BCDFC constructed features (d) visualizes a proper separation of the two main clusters when the cut is at the height level of 10. Besides, the dendrogram (b) demonstrates that the use of HybridGPFC constructed features fails in finding the optimal separation for the same cut; which is also the case for the use of MCIFC and the full set of original features as illustrated by Fig. 5(a) and Fig. 5(c).

VI. Conclusion and future work

In this research work, we have proposed BCDFC as a new evolutionary method for class dependent feature construction. The latter is first framed as a bi-level optimization problem and then solved using an adapted version of CODBA as a search engine. The core novelty of our method is the use of multiple follower algorithms at the lower level each sampling a whole search space to find optimized constructed features for its predefined class. A set of comparisons are performed with regard to four relevant state-of-art methods using three classifiers that are DT, NB, and K-NN. The statistical analysis of the obtained results showed the outperformance and the merits of our proposal. This work could be extended in a number of ways. First, it would be interesting to investigate the performance of a modified version of BCDFC where multiple leaders exist in the upper level. In this way, the upper level searches for class dependent feature subsets and then each subset is sent to its corresponding follower algorithm to find optimized constructed features for the corresponding class. Second, as the wrapper evaluation could incur a considerable computational cost, we believe that using surrogate models could be a good choice to reduce this cost. Finally, inspired from the group decision making field [29], dataset instances could have different labels from one decision maker to another due the subjective nature of their opinions, knowledge, and experiences. Hence, investigating the class dependent feature selection and construction problems in a multi-decision maker environment could be a very challenging direction, as it requires efficacious preference modeling and aggregation procedures.

References

[1] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," Pattern Recognition, vol. 43, pp. 5-13, 2010.

[2] M. Cerrada, R. V. Sanchez, F. Pacheco, D. Cabrera, G. Zurita and C. Li, "Hierarchical feature selection based on relative dependency for gear fault diagnosis," Applied Intelligence, vol. 44, pp. 687-703, 2016.

[3] H. Liu and H. Motoda, "Feature Extraction, Construction and Selection: A Data Mining Perspective,"Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[4] B. Tran, M. Zhang, and B. Xue,"A PSO based hybrid feature selection algorithm for high-dimensional classification," in IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 3801–3808.

[5] L. Wang, N. Zhou, and F. Chu, " A general wrapper approach to selection of class-dependent features," IEEE Transactions on Neural Networks, vol. 19, pp. 1267-1278, 2008.

[6] B. Colson,P. Marcotte and G. Savard, "An overview of bilevel optimization," Annals of Operations Research, vol. 153, pp. 235-256, 2007.

[7] B. Xue, M. Zhang, W. N. Browne and X. Yao," A Survey on evolutionary computation approaches to feature selection,"IEEE Transactions on Evolutionary Computation,vol. 20, pp. 606-626, 2016.

[8] Z. X. Zhu , Y. S. Ong and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," IEEE Transactions Systems, Man, and Cybernetics B, vol. 37, pp. 70-76, 2007.

[9] Z. X. Zhu , Y. S. Ong and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," IEEE Transactions Systems, Man, and Cybernetics B, vol. 37, pp. 70-76, 2007.

[10] M. Hammami, S. Bechikh, C. C. Hung and L. Ben Said, "A Multi-objective Hybrid Filter-Wrapper Evolutionary Approach for Feature Selection," Memetic Computing, vol. 11, pp. 193-208, 2018.

[11] U. Kamath, K. De Jong and A. Shehu, "Effective Automated Feature Construction and Selection for Classification of Biological Sequences," PLoS ONE, vol. 9, doi: 10.1371/journal.pone.0099982, 2014.

[12] B. Tran, M. Zhang and B. Xue, "Multiple feature construction in classification on high-dimensional data using GP," in IEEE Symposium Series on Computational Intelligence," 2016, pp. 1-8.

[13] M. Hammami, S. Bechikh S, C. C. Hung and L. Ben Said, "A multi-objective hybrid filter-wrapper evolutionary approach for feature construction on high-dimensional data," in IEEE Congress on Evolutionary Computation (CEC), 2018, pp. 1-8.

[14] L. Cervante, B. Xue, M. Zhang, and L. Shang, "Binary particle swarm optimisation for feature selection: A filter based approach," in IEEE Congress on Evolutionary Computation (CEC), 2012, pp. 881–888.

[15] M. Hammami, S. Bechikh, C. C. Hung and L. Ben Said, "Weighted-Features Construction as a Bi-level Problem," in IEEE Congress on Evolutionary Computation (CEC), 2019, pp. 1604-1611.

[16] D. Sahin, M. Kessentini, S. Bechikh and K. Deb, "Code-Smell Detection as a Bilevel Problem," ACM Transactions on Software Engineering and Methodology, vol. 24, pp. 1-44, 2014.

[17] A. Chaabani, S. Bechikh and L. Ben Said, "A co-evolutionary decomposition-based algorithm for bi-level combinatorial optimization," in IEEE Congress on Evolutionary Computation (CEC), 2015, pp. 1659-1666.

[18] G. Patterson and M. Zhang, "Fitness functions in genetic programming for classification with unbalanced data," in Advances in Artificial Intelligence, 2007, pp. 769-775.

[19] JS. Arora, "Introduction to Optimum Design," Academic Press, ISBN: 9780128009185, 2017.

[20] B. Tran, B. Xue, and M. Zhang, "Class Dependent Multiple Feature Construction Using Genetic Programming for High-Dimensional Data," in Advances in Artificial Intelligence, vol. 10400, W. Peng, D. Alahakoon, and X. Li, Eds. Lecture Notes in Computer Science, Springer, 2017, pp. 182-194.

[21] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," Journal of bioinformatics and computational biology, vol. 3, pp. 185-205, 2005.

[22] B. Tran, B. Xue, and M. Zhang, "Genetic programming for multiple-feature construction on high-dimensional classification," Pattern Recognition, vol. 93, pp. 404-417, 2019.

[23] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin D and S. Levy," A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis," Bioinformatics, vol. 21, pp. 631-643, 2005.

[24] B. Xue, M. Zhang and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multiobjective approach," IEEE Transactions on Cybernetics, vol. 43, pp. 1656-1671, 2013.

[25] J. Derrac, S. García, D. Molina and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," Swarm and Evolutionary Computation, vol. 1, pp. 3-18, 2011.

[26] L. Kaufman and P. J. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis," Wiley, New York, ISBN:9780470316801, 1990.

[27] C. Shannon and W. Weaver, "The mathematical theory of communication," Urbana, The University of Illinois Press, 1948.

[28] B. Tran, B. Xue, and M. Zhang, "Genetic programming for feature construction and selection in classification on high-dimensional data," Memetic Computing, vol. 8, pp. 3-15, 2015.

[29] S. Bechikh, L. Ben Said, and K. Ghédira,"Negotiating decision makers' reference points for group preference-based Evolutionary Multi-objective Optimization," in 11th International Conference on Hybrid Intelligent Systems (HIS), 2011, pp. 377-382.