2024 4th International Conference on Machine Learning and Intelligent Systems Engineering (MLISE) | 979-8-3503-7507-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/MLISE62164.2024.10674630

Genetic Programming for Feature Selection and Construction to High-Dimensional Data

Jianbin Ma*
College of Information Science and Technology, Hebei Agricultural University, Baoding, China
*Corresponding author: majianbin@hebau.edu.cn

Man Zhu[a]
College of Information Science and Technology, Hebei Agricultural University, Baoding, China
[a]e-mail: 18332189735@163.com

***Abstract*—Classification on high-dimensional data is a challenging task due to a large number of redundant and irrelevant features. Feature construction using Genetic Programming (GP) is an effective feature processing approach for classification, however, in high-dimensional applications, too many redundant and irrelevant features may reduce GP's search ability and affect the classification performance of feature construction. In this paper, two feature selection and construction approaches to high-dimensional data are proposed. The first is a two-stage feature selection and construction approach named LfsFc, which first uses linear forword feature selection method (Lfs) to reduce the search space of features, and then uses a GP-based multiple feature construction approach (Fc) to construct multiple features. The second is a multi-objective GP-based feature selection and construction approach named MoFc which optimizes information gain ratio and the number of selected features, and archives elite individuals as constructed features. Experiments on ten high-dimensional datasets show that LfsFc and MoFc can improve the classification performance compared with Fc and original features in four decision tree classifiers.**

Keywords-Feature Construction; Feature Selection; High-dimensional; Classification; Genetic Programming

I. INTRODUCTION

Classification aims to find models learned from training data and predict the class labels of unknown data based on predefined features. Features are key factors that affect the classification performance. With the development of data collection technology, the collection of high-dimensional data has become increasingly easy. In text classification or biological classification applications, there are thousands or even tens of thousands of features, which have a large number of redundant and irrelevant features. Too many redundant and irrelevant features will reduce the classification performance of learned models and increase the training and testing time.

Feature selection is to select an effective feature subset from original features so that the feature space is optimally reduced and it is frequently used for feature processing. In recent years, research methods focus on improving evaluation criteria and search strategies to search for effective feature subset [1]. However, in some high-dimensional applications, original features do not have adequate discriminating ability, and there may be some correlation between features to determine class labels. In recent years, feature extraction methods using deep learning, such as auto-encoders [2], are used to extract in-dept features from multi-layer neural network mapping and perform well on image classification applications. However, these methods require a significant amount of training samples, and designing an appropriate deep neural network usually needs trial or expert knowledge of the field. Genetic Programming (GP) [3] is an effective evolutionary computation (EC) algorithm. The solutions of GP can be represented as trees, where original features form the terminal nodes and logical or arithmetic operators form the internal nodes. Due to its flexible representation, GP can be used to solve a variety of tasks. Therefore, GP can be used to construct new high-level features to build relations between original features [4]. For some applications, the features constructed by GP may achieve good classification results.

GP-based feature construction approaches focus on filter or wrapper, constructing single feature or multiple features. Most previous works concerning GP-based feature construction approaches focus on datasets with dozens or hundreds of features. However, in high-dimensional applications, due to the larger search space, solving high-dimensional applications using GP-based feature construction approaches is a challenging task. Tran et al.[4] proposed a feature construction and selection approach to high-dimensional data that simultaneously performs feature construction and feature selection. Hammami et al.[5] proposed a multi-objective filter-wrapper GP-based feature construction on high-dimensional data to solve computationally intensive of the wrapper evaluation. However, for high-dimensional applications, the impact of redundant and irrelevant features on GP-based feature construction approaches has not been verified. It is necessary to investigate whether the classification performance of GP-based feature construction approaches will be affected by high-dimensional data and whether removing redundant and irrelevant features can improve the classification performance.

Ma et al. [6] proposed to build multiple features by archiving multiple excellent individuals during a GP run. Later, Ma et al. [7] proposed to construct multiple features using the method in [6], and then select effective features from the constructed features. There has been no research on enhancing GP's search ability through feature selection for high-dimensional data to improve the effectiveness of feature construction.

In this paper, we have investigated two strategies to removing redundant and irrelevant features for feature construction approaches. The first is a two-stage feature selection and construction method named LfsFc, which first use linear forward feature selection method (Lfs) [8] to remove irrelevant and redundant original features, so as to enhance GP's search ability. Then, the multiple feature construction approach proposed in [6] is employed to construct multiple features. The second is to investigate a multi-objective GP-based feature selection and construction method to simultaneously reduce the number of selected features and improve classification performance (MoFc). Our overall goal is to investigate the combination of feature selection and feature construction to solve high-dimensional classification tasks.

II. MATERIALS AND METHODOLOGY

A. Datasets and parameter settings

Ten high-dimensional datasets are collected to verify the effectiveness of our proposed approaches. Two e-mail datasets involving text classification problems come from UCI machine learning repository [9]. Eight microarray datasets involving gene classification problems are collected from the website http://csse.szu.edu.cn/staff/zhuzx/datasets.html. The detailed information of the ten datasets is shown in TABLE 1. The dimension of these datasets is high. Datasets contains a large number of redundant and irrelevant features, which can be used to verify the effectiveness of our proposed feature selection and construction approaches that are designed for high-dimensional data.

GP is run on ECJ library [10]. TABLE 2 shows the parameter settings. The function set of GP includes +, −, *, %. When the operator % is divided by zero, it returns zero. Other parameters are consistent with those in literatures [6,7]. For fair comparison, fixed GP's parameters is setted.

Four decision tree algorithms including J48, BF tree, REP tree and Random tree (RT) are selected as classifiers to verify our proposed approaches. The Weka package is used to run the above four classifiers.

To obtain more convincing evaluation performance, we use different random seeds to generate 30 different training sets and testing sets. 70% of the samples are training sets and 30% are testing sets. To avoid the stochastic characteristics of GP, each algorithm runs independently 30 times on 30 training sets and testing sets. Training sets are used to get effective selected features or constructed features. Testing sets are used to test the experimental results using 10-fold cross validation. The experimental results are evaluated by the classification accuracy.

Table 1 Dataset's description

Dataset	Features	Instances	Classes
Colon	2000	60	2
Leukemia	7129	72	2
Lymphoma	4026	62	3
SRBCT	2308	83	4
MLL	12582	72	3
Ovarian	15154	253	2
Lung	12600	203	5
DLBCL	2648	77	2
DBWorld	4702	64	2
DBWorld_stemmed	3721	64	2

Table 2 GP's parameter setting

Parameters	Parameter value
Population size	500
Initialization	Ramped half-and-half
Maximum tree depth	17
Terminal set	Original features or selected features
Function set	+, −, ×, % (protected division)
Number of generations	50
Selection method	Tournament method
Mutation method	Random subtree creation
Cross-over probability	90%
Mutation probability	10%

B. Methodology

Training sets are used to select and construct new features. Suppose a training set is represented as D, and the original feature set of the training set D is represented as $F_0 = \{f_1, f_2, \cdots, f_n\}$, where n is the number of original features, f_j denotes the jth original feature. The purpose of this paper is to propose an effective feature processing method for high-dimensional datasets and compare with existing feature processing methods.

1) The proposed LfsFc approach

For Lfs, correlation-based evaluation method is used as feature evaluator. Linear forward selection [8] is used to search for optimal feature subsets. For Fc, we use our proposed multiple feature construction approach in [6], which stores top 20 excellent individuals during GP runs. Information gain ratio (IGR) is used as the fitness function to evaluate the constructed features. Standard GP representation methods are used to construct features. The individuals of GP are represented as tree-like structures. The internal nodes of an individual are randomly generated from a function set of {+, −, ×, %}. The terminal nodes of an individual are randomly generated from original features or selected features.

The idea of LfsFc is to first remove redundant and irrelevant features using Lfs, and then construct multiple features using Fc. The purpose of LfsFc is to verify whether redundant and irrelevant original features affect GP's search performance for feature construction on high-dimensional applications. Our proposed LfsFc is divided into two-stages. The framework of LfsFc is shown in Figure 1.

In the first stage, the original features $F_0 = \{f_1, f_2, \cdots, f_n\}$ are reduced to a smaller feature subset $F_s = \{f_1, f_2, \cdots, f_s\}$ by Lfs, where s is the number of selected features. In the second stage, top β constructed features are stored using Fc during a GP run. In this paper, β is set to 20. The impact of parameter β on experimental results is shown in [6]. Suppose the constructed features are denoted as $F_c = \{f_{c1}, f_{c2}, \cdots, f_{c20}\}$,where f_{cj} denotes the jth constructed feature. The terminal nodes of GP individuals are generated from selected features $F_s = \{f_1, f_2, \cdots, f_s\}$.

According to the constructed features Fc, the testing sets are transformed into new testing sets, and are used to evaluate the classification performance using 10-fold cross validation.

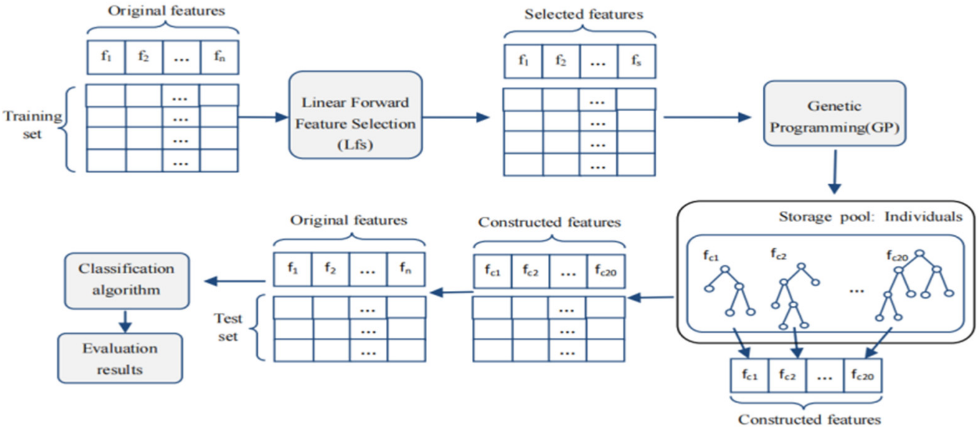

Figure 1. The framework of LfsFc

2) *The proposed MoFc approach*

The idea of MoFc is to use multi-objective Genetic Programming (MOGP) to simultaneously improve the discriminative ability of constructed features and reduce the number of selected features. We employed the evolutionary strategy of NSGA-II to develop MOGP. Our MOGP is used to maximize $F_1(x)=IGR$ and minimize $F_2(x)=f_s(x)$, subject to x is a GP individual (constructed feature), and $f_s(x)$ is the number of terminal nodes (selected features) in a GP individual. In order to maintain consistency with LfsFc, we also archives 20 elite individuals which have the highest IGR values during GP runs. The training sets are used to obtain elite individuals which are transformed to constructed features Fc. Then, the testing set are used to evaluate the classification performance of the constructed features. The Pseudo-Code of MoFc is shown in Figure 2.

With the evolution of MoFc, using MOGP to optimize IGR performance and the number of selected features will generate a large number of overlapping individuals, which makes GP converge too early and reduces GP's search space. To solve this problem, from the second generation, the genetic operators of crossover and mutation is used to generate the offspring population. Then, the overlapping individuals are removed from the offspring and the merged offspring and parent population. In order to guarantee the fixed number of population, crossover-mutation is executed again to generate more new individuals to add into new population. Then, the overlapping individuals are removed again. Generating new individuals by crossover-mutation and removing overlapping individuals are repeatedly executed until the merged offspring and parent population has no overlapping individuals and retains a fixed number of individuals. Each subsequent generation performs the same operation above, so that there are no overlapping individuals during the evolution process of MOGP. The program codes for overlapping individual removal can be seen on lines 20 to 27 in Figure 2.

III. RESULTS AND DISCUSSION

A. *Comparisons of LfsFc, MoFc and FULL, Fc*

The experimental results of our proposed LfsFc and MoFc approaches with FULL and Fc [6] are shown in TABLE 3. In the able, "F" denotes the number of features generated by different feature processing methods. The "Wtest" column denotes the Wilcoxon significance test with a P-value of 0.05 for the corresponding methods of different classifiers and the FULL. "=" denotes their results are similar, and "–" or "+" denotes the experimental result is significantly worse or better than FULL.

Algorithm 1: Pseudo-Code of MoFc algorithm

```
   Input: $\mathcal{D}$: a training set
   Output: $\mathcal{F}_C$: constructed features
1  Set Maximum generation=M, Population size=N, t=current generation;
2  Initialize the population $P_0$ by ramped-half-and-half method;
3  if $t$=1 then
4      Evaluate each individual in the $P_0$ using two fitness functions;    //
         $IGR$ and $f_s(x)$
5      Archive elite individuals;
6      Fast non-dominated sorting and crowding distance calculation for $P_0$;
7      Set the fitness value of each individual to its non-dominant ranking level;
8      Select individuals using the Tournament method based on the $l$
         (non-dominant ranking level) and $d$ (crowding distance);
9      Crossover and mutation to produce offspring ($Q_0$), $|Q_0| = N$;
10 end
11 for $t$=2 to $M$ do
12     Merge $P_{t-1}$ (parent) and $Q_{t-1}$ (offspring) into $R_t$, $|R_t| = 2N$;
13     Evaluate each individual in the $R_t$ using two fitness functions;
14     Archive elite individuals;
15     Fast non-dominated sorting and crowding distance calculation for $R_t$;
16     Choose $P_t$ from $R_t$ according to $l$ and $d$, $|P_t| = N$;
17     Pick the H (non-dominated solutions) from $P_t$;
18     Select individuals using the Tournament method based on the $l$ and $d$;
19     Let $C = P_t$, $S = \varnothing$;
20     while $|C| < 2N$ do
21         Crossover and mutation to generate $Q_t$, $|Q_t| = N$;
22         $S = S + Q_t$;
23         Remove overlapping individuals in $S$;
24         $C = C + Q_t$;
25         Remove overlapping individuals in $C$;
26     end
27     Let $Q_t$=$S$;
28 end
29 Output constructed features $\mathcal{F}_C = \{f_{c1}, f_{c2}, \cdots, f_{c20}\}$ according to elite
     individuals
```

Figure 2. The Pseudo-Code of MoFc

1) *Comparisons between LfsFc, MoFc, Fc and FULL*

The Wtest in TABLE 3 shows that Fc, LfsFc and MoFc are significantly better than FULL in almost all decision tree classifiers. As shown in TABLE 3, Fc can achieve much higher classification accuracy than FULL in almost all the decision tree classifiers. For example, for the J48 classifier, the classification accuracy of Fc are 3.89%, 17.22%, 6.45%, 17.41%, 26.11% and 6.29% higher than those of FULL respectively on Colon, Leukemia, Lymphoma, SRBCT, MLL and Lung datasets, the classification accuracy of LfsFc are 3.70%, 16.48%, 6%, 25.37%, 32.59% and 7.71% higher than those of FULL respectively on Colon, Leukemia, Lymphoma, SRBCT, MLL and Lung datasets, and the classification accuracy of MoFc are 5.55%, 19.81%, 8.23%, 19.81%, 28.79% and 2.31% higher than those of FULL respectively on Colon, Leukemia, Lymphoma, SRBCT, MLL and Lung datasets. In addition, Fc, LfsFc and MoFc can reduce the dimension of datasets from thousands or tens of thousands to twenty. The experimental results of feature construction methods, including Fc, LfsFc, and MoFc, verify that feature construction methods can discover hidden relations between features and achieve good classification performance in high-dimensional applications.

Table3 The experimental results of different feature construction methods

Dataset	Method	#F	A-J48	A-BF	A-REP	A-RT	Wtest
Colon	FULL	2000	67.78±15.67	66.67±7.17	67.22±5.04	58.33±11.72	
	Fc	20	71.67±12.94	69.63±13.36	68.70±9.24	68.89±12.22	+ + + +
	LfsFc	20	71.48±15.96	**70.37**±12.70	70.19±9.46	71.30±11.39	+ + + +
	MoFc	20	**73.33**±11.24	70.00±12.31	**72.41**±9.13	**73.52**±10.21	+ + + +
Leukemia	FULL	7129	67.78±17.59	66.30±10.14	67.41±7.13	64.81±11.77	
	Fc	20	85.00±10.06	83.89±13.25	78.70±12.68	83.33±10.24	+ + + +
	LfsFc	20	84.26±9.08	83.52±10.59	78.15±10.73	86.11±8.93	+ + + +
	LfsFc	20	**87.59**±9.26	**85.56**±13.12	**82.78**±10.58	**87.04**±7.77	+ + + +
Lymphoma	FULL	4026	84.44±8.49	77.56±6.32	78.44±3.30	86.00±8.32	
	Fc	20	90.89±4.71	**85.78**±5.09	78.44±3.30	94.22±6.38	+ + + +
	LfsFc	20	90.44±5.33	84.44±5.37	**78.67**±2.67	94.89±4.77	+ + + +
	MoFc	20	**92.67**±4.34	83.11±6.14	78.00±3.06	**96.00**±4.42	+ + + +
SRBCT	FULL	2308	52.78±13.13	55.19±13.60	30.19±9.04	44.07±9.62	
	Fc	20	70.19±8.90	65.74±10.35	57.59±8.78	72.78±8.53	+ + + +
	LfsFc	20	**78.15**±6.92	72.41±10.70	57.59±6.70	**79.26**±10.08	+ + + +
	MoFc	20	72.59±8.28	**72.72**±7.22	**77.95**±3.08	75.38±4.81	+ + + +
MLL	FULL	12582	46.67±14.53	38.70±14.66	27.22±12.03	48.33±11.03	
	Fc	20	72.78±10.58	70.37±18.11	62.41±17.90	72.41±12.63	+ + + +
	LfsFc	20	**79.26**±8.23	**81.67**±8.74	73.15±9.84	**80.74**±8.08	+ + + +
	MoFc	20	75.46±8.50	72.22±7.10	**74.84**±8.59	79.74±6.78	+ + + +
Ovarian	FULL	15154	94.89±3.23	96.13±2.93	94.18±2.89	79.11±4.91	
	Fc	20	94.98±2.52	95.73±2.78	95.51±2.49	95.20±3.76	= = = +
	LfsFc	20	96.58±2.33	97.33±2.31	96.40±2.94	97.51±2.11	+ = + =
	MoFc	20	**97.38**±1.77	**98.22**±1.70	**98.18**±1.74	**98.22**±1.55	+ + + +
Lung	FULL	12600	75.25±15.18	70.37±14.39	69.20±13.62	69.20±13.68	
	Fc	20	81.54±4.07	**82.96**±4.91	**80.43**±3.16	79.75±4.83	+ + + +
	LfsFc	20	**82.96**±5.67	82.59±5.67	79.63±3.97	**80.62**±5.23	+ + + +
	MoFc	20	77.56±4.36	79.26±4.18	78.58±3.05	75.43±6.14	+ + + +
DLBCL	FULL	2648	86.48±16.77	**94.44**±17.86	79.63±14.93	72.96±15.83	
	Fc	20	83.33±9.94	85.93±7.82	81.67±2.55	84.07±10.51	− − + +
	LfsFc	20	85.56±11.71	85.56±8.19	81.67±2.55	87.22±7.34	− − + +
	MoFc	20	**90.00**±6.48	93.15±7.95	**82.59**±1.89	**90.93**±6.49	+ = + +
DBWorld	FULL	4702	72.00±19.81	64.00±18.10	63.11±14.17	63.78±14.78	
	Fc	20	67.33±14.23	70.22±15.08	65.11±13.52	70.22±16.21	− + = +
	LfsFc	20	**76.67**±13.42	**75.11**±14.60	**71.33**±13.13	**74.22**±13.85	+ + + +
	MoFc	20	69.78±14.37	68.44±13.66	64.00±8.36	72.44±11.77	− + = +
DBWorld_S	FULL	3721	69.33±22.08	62.44±20.76	63.33±17.95	63.33±14.17	
	Fc	20	71.56±10.60	70.00±13.42	66.22±9.88	72.44±11.38	+ + + +
	LfsFc	20	**78.00**±12.78	**75.78**±14.55	**74.44**±14.10	**74.00**±14.44	+ + + +
	MoFc	20	72.00±14.54	70.22±14.88	67.33±13.48	73.78±14.39	+ + + +

2) Comparisons between LfsFc, MoFc and Fc

From TABLE 3, we can see that LfsFc can improve the classification performance of Fc on Colon, Leukemia, SRBCT, MLL, Ovarian, DLBCL, DBWorld and DBWorld_S datasets in all decision tree classifiers, and LfsFc can achieve equivalent classification performance comparing with Fc on Lymphoma and Lung datasets. For example, the classification accuracy of LfsFc are 6.48%, 11.3%, 10.74% and 8.33% higher than those of Fc respectively in the J48, BF, REP and RT decision tree classifiers on MLL dataset. The experimental results confirm that in high-dimensional data, too many redundant and irrelevant features affect GP's search performance, and removing ineffective features in advance can improve the searching ability of GP and increase the classification performance of Fc.

From the experimental results in TABLE 3, we can see that MoFc can achieve equivalent classification performance as Fc on Lymphoma, Lung, DBWorld and DBWorld_S datasets, and MoFc can achieve higher classification performance than Fc on other datasets. For example, the classification accuracy of MoFc is 2.40%, 6.98%, 20.36% and 2.60% higher than Fc in the J48, BF, REP and RT decision tree classifiers on SRBCT dataset. MoFc improves the classification performance of constructed features while reduces the number of selected features. The experimental results show that MoFc can also reduce the impact of ineffective features on classification performance and achieve higher classification performance than Fc.

IV. CONCLUSIONS

This paper proposes two feature selection and construction approaches to address the impact of high-dimensional data on feature construction methods. The first is a two-stage feature selection and construction approach (LfsFc), which first uses a linear forward feature selection method (Lfs) to remove redundant and irrelevant features from high-dimensional data, then employs a multiple feature construction approach (Fc) to construct multiple features. The second is a multi-objective GP-based feature construction (MoFc) which optimizes classification performance and the number of selected features at same time. Experiments on ten high-dimensional datasets show that LfsFc and MoFc are all effective methods to improve the classification performance comparing with using Fc.

REFERENCES

[1] K. Chen, B. Xue, M. Zhang, F. Zhou. (2022) An evolutionary multitasking-based feature selection method for high-dimensional classification, IEEE Transactions on Cybernetics 52 (7) :7172–7186.

[2] B. Du, W. Xiong, J. Wu, et al. (2017) Stacked convolutional denoising auto-encoders for feature representation, IEEE Transactions on Cybernetics 47 (4) : 1017–1027..

[3] J. R. Koza, D. Andre, F. H. Bennett, et al. (2002) Genetic programming III - Darwinian invention and problem solving, IEEE Transactions on Evolutionary Computation 7 (4) :451–453.

[4] B. Tran, B. Xue, M. Zhang. (2016) Genetic Programming for feature construction and selection in classification on high-dimensional data. Memetic Computing 8 (1) : 3–15.

[5] M. Hammami, S. Bechikh, C. C. Hung, et al. 2018. A multi-objective hybrid filter-wrapper evolutionary approach for feature construction on high-dimensional data, in: IEEE Congress On Evolutionary Computation.

[6] J. Ma, G. Teng.(2019) A hybrid multiple feature construction approach using genetic programming, Applied Soft Computing 80: 687–699.

[7] J. Ma, X. Gao. (2020) A filter-based feature construction and feature selection approach for classification using genetic programming, Knowledge-Based Systems 196:105806.

[8] M. Gutlein, E. Frank, M. Hall, et al. 2009 Large-scale attribute selection using wrappers, in: Proceedings of 2009 IEEE Symposium on Computational Intelligence and Data Mining.

[9] D. Dheeru, E. Karra Taniskidou. (2017) UCI machine learning repository. URL http://archive.ics.uci.edu/ml.

[10] S. Luke, Ecj then and now, in: Proceedings of the 2017 Genetic and Evolutionary Computation Conference Companion, 2017, pp. 1223–1230.