ISSN: 2321-7782 (Online)
Volume 3, Issue 10, October 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Improving Classification Precision of Feature Selection using Genetic Programming

Rupali Koushal[1]
Information Technology department, shri vaishnav institute of Technology and Science
SVITS
Indore, India

Manoj Dhawan[2]
Information Technology department, shri vaishnav institute of Technology and Science
SVITS
Indore, India

Abstract: ***After all study in the field feature selection and classifier designing it is found that Feature Selection is very important task in classification and machine learning.In this paper we present an methodology to select the optimum number of features from the dataset using genetic programming. Form best classifier we select optimum number of feature for this Genetic programming life cycle is run for the fifty percent of generation. After this process we get set of optimum features. Other feature present in classifier is replace with features exist in optimum set and this process is run till last generation. This experiment is performing on different datasets. Different dataset has different number of feature. The dimensionality of dataset is varying from low to high. The result of the experiment is quite better than previous result.***

Keywords: Genetic Programming, Feature selection, Crossover, Mutation, fitness function, filter approach, wrapper approach, ramped half-and-half method

I. INTRODUCTION

Feature selection is an approach to obtain more accurate solution to a given problem. The goal of feature selection is to obtain subset of feature which is more efficient. Feature selection is useful because all existing feature is not necessary because of the irrelevant feature computing cost of the problem is increased. Feature selection can extensively improve the comprehensibility of the resulting classifier. Further, it is often the case that finding the proper subset of predictive features is an vital problem. For example, physician may make a decision based on the selected features whether a dangerous surgery is necessary for treatment or not [1]. Classification is one important task in machine learning and data mining whose focus is to classify all instances into data class of different groups, those are based on that information this is describe by those features. If we don't have any knowledge from past, then it gets tough to determine useful feature correctly. As a result large number of feature in the dataset. In which relevant features, irrelevant features are included, but in classification redundant and irrelevant features are of no significance. These redundant and irrelevant features degrade the performance of classification. Question is how they do it? Because of large no of features large search space becomes. From large search space it is difficult to fine the solution these degrade the performance. Feature selection focus on the solution of problems there large no of feature exist [2]. Function of feature selection is to focus on those features which are useful for classification and reduce irrelevant features. If there is small set of features than it reduces training time and classifycation performance is improved. Features selection is tough task because complex interaction among features may arise. It may be a case that relevant feature may not give appropriate result if they are not used properly. Therefore an optimal feature subset should be a group of complementary features that span over the diverse properties of the classes to properly discriminate them. The advantage of classification is it provides a low dimensional representation decrease the problem of over fitting. Classification and feature selection is applicable in different area like bioinformatics, web-intelligence, and medical science [4]. Genetic programming is an evolutionary problem solving methods which has been extensively used to evolve programs or sequences of operations. GP is introduced by Koza. Genetic programming automatically generating computer programs to perform some specific task. Using GP we design

the classifier along with feature selection. Purpose of using GP is that simultaneously among features we want to select useful features and construct the classifier. Application area of GP is very large. In a single GP lifecycle classifier designing as well as feature selection both are performed [5].

II. Related Work

Durga Prasad muni et al intend a new process for designing classifiers for a c-class ($c \geq 2$) difficulty using genetic programming (GP). The projected approach takes an combine view of the entire classes once the GP evolves. A multitree depiction of chromosomes is used**.** Genetic algorithm comprises a set of individual's elements and a set of biologically inspired operators defining the population itself. It is an integrated evolutionary approach where classifier trees for all classes are evolved simultaneously. In computing, GA maps problems onto a set of strings each string representing a potential solution For genetic operation, tree is selected on the basis of their unfitness. On the basis of their unfitness proposed a new crossover procedure and a modified mutation process named nondestructive directed point mutation. To optimize the classifier an OR-ing operation is introduced and a weight based scheme for conflict resolution. Then tested classifier with several data set and obtained satisfactory result [6]. Peter I. Rockett et al. In this paper an optimized feature extraction framework is presented. That uses multiobjective genetic programming (MOGP). Since in previous method set of inputs place into one dimensional decision space. While in MOGP method data set place into multidimensional decision space to obtain excellent classification performance. In this paper pareto dominance set is used for vector minimization By which maximal class separability is obtain [7]. Sebastián Maldonado et al proposed embedded method for classification in feature selection by penalizing each feature utilize in the double formulation of SVM. This scheme is known as KP-SVM optimize shape of an anisotropic RBF kernel reducing feature that have irrelevance for classification. it hamper elimination of those feature which degrade the performance of classifier. Author examines this method with real world object and compares result with well known feature selection technique [8]. Isac Sandin et al propose an aggressive, however very useful, selection of features using Genetic Programming (GP). This strategy is able to deal with large dimensionality and perform feature selection. It reduces the dimensionality of features. It uses some common FS matrices namely Correlation Coefficient, $\chi 2$, Information Gain, Odds Ratio. At the end combine the result of all the matrices into new subsets of features. Result is compare with every individual FS matrices apply a k8 cancer-rescue mutants data set [9]. In the classification problem lots of feature is present in dataset but hole are not useful for classification. Unwanted and duplicate feature may minimize the performance. The objective of feature selection is to choose subset of necessary feature to achieve greater performance rather than using every feature. This method has two main objectives first enlarge the classification performance and second one is reduce the number of feature. However in the previous feature selection algorithm they have only task. Very first study on particle swarm optimization (PSO) for feature selection is presented by this paper. For selecting feature two PSO based multiobjective feature selection algorithm is used in this paper. The primary algorithms initiate the scheme of nondominated organization into PSO to deal with feature selection troubles. The next algorithm concern the thoughts of crowding, mutation, dominance to Particle Swarm Optimization to find Pareto front solutions which gives better results than the other methods mentioned previously [10]. Nikhil R. Pal et al in this paper briefly review proposed GP methodology for selecting a good subset of feature and using these feature constructing a classifier. It provides solution for c-class problem. Solution proposed by author introduce two new crossover operations to outfit the feature selection procedure as a by-product algorithm produce feature ranking scheme. The two crossover operation use by author is homogenous crossover and heterogenous crossover [11]. Fawaz A. Alsulaiman et al uses genetic programming for gens expression programming that is very effective for generating analytic model. The analytic model has two important function first one is behave as classifier in high dimensional haptic feature space. Second one is act as general dimensionality reducers [12]. Anuradha Purohit et al randomly select the feature from the available set of feature and two type of crossover is use here first on is homogeneous and heterogeneous crossover. For the fitness calculation a formula is use [13].

III. PROPOSED WORK

This paper represents proposed methodology for Feature selection and classifier design using Genetic programming. This method only single run is required to obtain the optimal features subsets this required following steps:

A. *Initialization*

The first step of an evolutionary algorithm is the initialization of the population. In the case of tree-based Genetic Programming this means we have to construct syntactically valid trees. Each of the trees for each individual is initialized randomly using the function set F which consists of arithmetic functions and the terminal set T containing feature variables and constants. The function set F = (+, - ,* ,/, ^, sin, cos) and terminal set T = (feature variable, R) where R contains randomly generated constants in [0.0, 1.0]. There are different methods for initialization of tree they are full method, grow method and Ramed half-and-half method. In this paper ramed half-and-half method is use for initialization.

B. *Fitness Evaluation*

Following fitness algorithm is use to compute the fitness of the each chromosomes of the population

for i = 1, 2…N

count=0;

if((T_i Є classifier A) and ($f(T_i) \geq 0$))

count=count+1;

end if

if((T_i !Є classifier A) and ($f(T_i) < 0$))

count = count +1;

End if

End for.

C. *Feature selection*

For feature selection following algorithm is used:

1. Arrange all the classifiers in decreasing order on the basis of their fitness value.

C_1= 98.6 C_2= 97.2 C_3=95.1……..C_N=76.4

2. Select ten best classifiers and calculate their average fitness value. This average fitness value is known as cumulative fitness.

$$C_{CUM}=(C_1+C_2+C_3+C_4+C_5+C_6+C_7+C_8+C_9+C_{10})/10$$

3. Those classifier whose fitness value is greater than the cumulative fitness value is being selected.

If ($C_i > C_{CUM}$) than C_i is selected

4. Weight is provided to the feature according to their occurrence in classifier.

f_0=14, f_1=3, f_2=7,f_3=5, f_4=18…….f_n=9

5. According to weight provided to feature calculate average weight of entire feature.

f_{avg} = (f_0+f_1+ f_2+f_3+f_4+…….+f_n)/ Total no of feature

6. Those feature whose weight is greater than the average weight of all the feature is useful for classification and get the set of relevant and irrelevant feature. This process is continued until reach up to 50% of population.

if($f_i > f_{avg}$) than feature is selected.

feature subset = {f_i f_j …}

7. The irrelevant feature is replaced by relevant feature in the classifier.

8. After feature replacement calculate fitness of classifier if fitness is greater than the cumulative fitness then this classifier is being forwarded to next generation.

D. *Genetic operator*

After the initial population has been initialized and evaluated by a fitness function the actual evolutionary process starts. The first step of each generation is the selection of parents from the current population. These parents are employed to produce offspring using one or more genetic operators.

1) Reproduction: In reproduction selected individual copies itself into the next generation. In this paper we allowed 10% of the population to reproduce. If the fitness test does not change, reproduction can have a major effect on the total time necessary for GP because a reproduced individual will have an equal fitness score to that of its parent.

2) Crossover: The crossover or recombination operator works by exchanging "genetic material" between two or more parent individuals and may result in several offspring individuals. Crossover requires two individuals and produces two different individuals for the new population. In this technique genetic material from two individuals is mixed to form offspring.

3) Mutation: The mutation operator is applied to a single individual at a time and makes (small) changes in the genetic code of an individual.

E. *Termination criteria*

The GP is terminated when all N training samples are classified correctly by a classifier or A predefine number M of generation are completed.

IV. Experimental Result

Automatically classifying data with high dimensionality is a difficult task. When we have skewed datasets this becomes more difficult. Different method are used for reducing dimensionality by feature selection are usually biased towards the largest class. Our goal is to maximize classification accuracy in the smallest class.

A. *Datasets*

1) Vehicle: The purpose is to classify a given silhouette as one of four types of vehicle, using a set of features extracted from the silhouette. The vehicle may be viewed from one of many different angles. This data set consists of 946 samples in 18-dimension distributed in four classes [14]. shown in table I.

2) Wine: These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. Alcohol, Malicacid, Ash, Alcalinity of ash, Magnesium, Total phenols Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline [15] shown in table I.

3) Sonar: **:** This data set contains 208 patterns obtained by bouncing sonar signals off a metal cylinder and rocks at various angles and under various conditions. Each pattern is represented by 60 attributes. Each attribute represents the energy within a particular frequency band [16]. shown in table I.

4) Wisconsin Breast Cancer (WBC): This data set consists of 699 samples in 10-dimension distributed in two classes (malignant and benign) [17]. shown in table I.

5) IRIS: This is the well-known Anderson's Iris data set. It is a set of measurements in four dimension taken on 150 Iris flowers, 50 each from three different species or classes. The four features are sepal length, sepal width, petal length and petal width [18]. shown in table I.

6) WDBC: WDBC: This Wisconsin Diagnostic Breast Cancer (WDBC) data set contains observations on 569 patients with either Malignant or Benign breast tumor. Each data point consists of 30 features [19]. shown in table I.

TABLE I DATASET

Name of dataset	Number Of Classes	Number of feature	Size of dataset
Wine	3	13	178(59+71+48)
Vehicle	4	18	946(240+240+240+226)
Sonar	2	60	208(97+111)
WBC	2	10	699(444+239)
IRIS	3	4	150(50+50+50)
WDBC	2	32	569(357+212)

B. *GP parameters*

The GP parameters we have used and are general for all the data sets are given in Table II. These parameters are essential for GP based classifier design. We have considered larger populations for higher dimensional data since use of a large population helps GP to grow to a good solution without using lots of generations.

TABLE III GP parameters

Parameters	**Values**
Probability of crossover operation, p_c	0.5
Probability of reproduction operation, p_r	0.25
Probability of mutation operation, p_m	0.25
Probability of selecting a function node during crossover operation, q_{fc}	0.7
Probability of selecting a terminal node during crossover operation, q_{tc}	0.3
Probability of selecting a function node during mutation operation, q_{fm}	0.8
Probability of selecting a terminal node during mutation operation, q_{tm}	0.2
Tournament size, τ	8
Total number of generations the GP evolved, M	30
Maximum height of a tree	6
Minimum height of a tree	2
Maximum number of nodes allowed in a tree	275

C. *Result*

We divide the datasets into testing set and training set and then performed our experiments. We have compared the average performance of the classifiers obtained by the proposed method using a set of selected features with the classifiers designed using all the features. The average classification (test) accuracy with all features and with the selected features and the average number of selected features over fifteen GP runs for four data sets are given in Table III. Results show that our method of designing classifiers with selected features outperforms.

We have compared our result with previous method for feature selection and classification using genetic programming. We found our result better than all the available previous result which is shown in table IV [11] [13].

TABLE IIIII Experimental Result

Dataset	Average performance of classifier with all feature	Average performance of classifier with selected feature
Vehicle	87.00	87.35
Wine	96.80	97.77
WBC	99.61	99.45
Sonar	95.75	99.55
IRIS	98.00	99.50
WDBC	96.35	98.00

TABLE IVV Result Comparison

Dataset	Proposed method	Online feature selection	Modified crossover operator	GPmtfs	Relief	NNFS1	IFN
Vehicle	87.35	78.45	-	-	-	-	-
Wine	97.77	94.82	91.26	94.82	95.0	-	91.7
WBC	99.45	96.84	97.13	96.84	93.6	94.15	94.0
Sonar	99.55	86.26	86.96	86.26	-	93.81	-
IRIS	99.50	98.69	96.41	98.69	96.0	-	94.0
WDBC	98.00	-	-	-	-	-	-

V. Conclusion

Automatically classifying data with high dimensionality is a difficult task. When we have skewed datasets this becomes more difficult. Different method are used for reducing dimensionality by feature selection are usually biased towards the largest class. Our goal is to maximize classification accuracy in the smallest class.

We proposed a method for FS and classifier design using GP. In our method only single run is required to obtain the optimal features subsets. We generate the initial population in such a manner that the classifier use different features subsets. The initialization generates the classifiers using smaller features subsets with higher probability. The classifier which classifies more sample using fewer features having higher fitness values to the fitness function. The multiobjective fitness function help to accomplish both FS and classifier design simultaneously our proposed method have several advantages first on is a low dimensional representation reduce the risk of our overfitting second one is using fewer features decreases the model complexity which improve its generalization ability and last one is a low dimensional representation require less computational efforts. Our method has two goals that is enlarge the classification performance and reduce the redundant or irrelevant. We tested the classifier with different datasets and gain quite satisfactory result. In future we will focus on reducing the size of the trees. We also want to investigate the utility of the logical and othe function for designing classifier.

Acknowledgement

The authors express their thanks to Assistant Prof. Manoj Dhawan, SVITS, Indore, who has helped in different ways to complete this work. Thanks are also due to the anonymous reviewers and our colleagues for their suggestions to improve the manuscript.

References

1. YongSeog Kim, W. Nick Street, and Filippo Menczer “Feature Selection in Data Mining” University of Iowa, USA.
2. M. Dash and H. Liu, “Feature selection for classification,” Intell. Data Anal., vol. 1, no. 1–4, pp. 131–156, 1997.
3. K. Neshatian and M. Zhang, “Genetic programming for feature subset ranking in binary classification problems” in Proc. Eur. Conf. Genetic Program., 2009, pp. 121–132.
4. J. R. Koza, “Genetic Programming: On the Programming of computers by Means of Natural Selection.” Cambridge, MA: MIT Press, 1992
5. S.N. Sivanandam, S.N. Deepa “principle of soft computing” Wiley publisher second edition, 18 January 2011.

6. Durga Prasad Muni, Nikhil R. Pal, Senior Member, IEEE, and Jyotirmoy Das "A Novel Approach to Design Classifiers Using Genetic Programming" IEEE transactions on evolutionary computation, VOL. 8, NO. 2, APRIL 2004.

7. Yang Zhang, Peter I. Rockett. "A generic optimising feature extraction method using multiobjective genetic programming" Applied Soft Computing 11 (2011) 1087–1097.

8. Sebastián Maldonado, Richard Weber, Jayanta Basak "Simultaneous feature selection and classification using kernel-penalized support vector machines" Information Sciences 181 (2011) 115–128 www.elsevier.com/locate/ins.

9. Isac Sandin, Guilherme Andrade, Felipe Viegas, Daniel Madeira and Leonardo Rocha "Aggressive and Effective Feature Selection using Genetic Programming", WCCI 2012 IEEE World Congress on Computational Intelligence June, 10-15, 2012 - Brisbane, Australia.

10. Bing Xue, Mengjie Zhang and Will N. Browne "Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach" IEEE transactions on cybernetics, vol. 43, NO. 6, december 2013.

11. Durga Prasad Muni, Nikhil R. Pal, Jyotirmoy Das "Genetic programming for simultaneous feature selection and classifier design" IEEE transactions on systems, man, and cybernetics—part b: cybernetics, vol. 36, no. 1, february 2006

12. Muhammad Waqar Aslama, Zhechen Zhu, Asoke Kumar Nandi "Feature generation using genetic programming with comparative partner selection for diabetes classification" journal homepage: www.elsevier.com/locate/eswa Expert Systems with Applications 40 (2013) 5402–5412.

13. Durga Prasad Muni, Nikhil R. Pal, Jyotirmoy Das "Genetic programming for simultaneous feature selection and classifier design" IEEE transactions on systems, man, and cybernetics—part b: cybernetics, vol. 36, no. 1, february 2006.

14. https://archive.ics.uci.edu/ml/datasets/vehicle access on date 2/10/15

15. http://archive.ics.uci.edu/ml/datasets/Wine access on date 2/10/15

16. https://archive.ics.uci.edu/ml/datasets/sonar access on date 2/10/15

17. https://archive.ics.uci.edu/ml/datasets/wbc access on date 2/10/15

18. https://archive.ics.uci.edu/ml/datasets/iris access on date 2/10/15.

19. https://archive.ics.uci.edu/ml/datasets/wdbc access on date 2/10/15

