Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition Letters xxx (2008) xxx–xxx

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Selecting and constructing features using grammatical evolution

Dimitris Gavrilis [b], Ioannis G. Tsoulos [a,*], Evangelos Dermatas [b]

[a] Department of Computer Science, University of Ioannina, Greece
[b] Department of Electrical and Computer Engineering, University of Patras, Greece

Received 5 November 2007; received in revised form 5 February 2008

Communicated by T. Vasilakos

Abstract

A novel method for feature selection and construction is introduced. The method improves the classification accuracy, utilizing the well-established technique of grammatical evolution by creating non-linear mappings of the original features to artificial ones in order to improve the effectiveness of artificial intelligence tools such as multi-layer perceptron (MLP), Radial-basis-function (RBF) neural networks and nearest neighbor (KNN) classifier. The proposed method has been applied on a series of classification and regression problems and an experimental comparison is carried out against the accuracy obtained on the original features as well as on features created by the PCA method.

Keywords: Artificial neural networks; Feature selection; Feature construction; Genetic programming; Grammatical evolution

1. Introduction

In classification theory, a set of features composes the pattern, typically classified in two categories; the primitive features, and a set of non-linear projections of the primitive features. The number and type of features are critical to the classification accuracy and computational complexity. As the number of features increases, additional examples are required to complete a reliable training process, allowing for more robust generalization capabilities without over-fitting effects. In the case of limited number of available data, the "curse of dimensionality" introduces two major problems: the selection of a subset from the original set of features that preserves the classification scheme and the construction of a robust set of new features from non-linear-transformations of the original set.

According to the Covers' theorem (Cover, 1965), at least one non-linear extension of the features vector exists which defines a features space where the classes of an arbitrary set of examples are linearly separated. In this direction, many methods have already been proposed for automatic definition of a non-linear extension of the features vector such as, multi-layer perceptron (MLP), polynomial, Radial-basis-function (RBF) neural networks etc. The proposed method utilizes the grammatical evolution technique to select and construct a subset of the original features set that satisfies or approaches a priory defined classification accuracy. In contrast to the traditional features selection approach, where experts or semi-automatic methods derive or transform the original set of features, the proposed method (Fig. 1) is fully automatic. The proposed method is quite different from others because:

(1) It utilizes a BNF grammar to represent the search space. This allows the easy manipulation of the search space through the grammar.
(2) The method can be easily scaled to use complex functions (like statistics, neural networks, etc.). In order to achieve that, one can define a new function and then simply use it in the grammar.
(3) The results produced by the proposed method are in a human – readable form. In some problems this can

[*] Corresponding author. Tel.: +30 2651098871; fax: +30 2651098890.
E-mail address: itsoulos@cs.uoi.gr (I.G. Tsoulos).

0167-8655/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2008.02.007

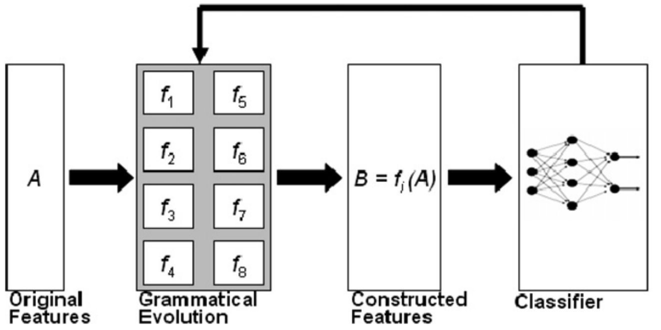

Fig. 1. Features selection and construction using grammatical evolution and neural networks.

lead scientists to a better evaluation and better understanding of the resulting features.

In Georgoulas et al. (2007) the proposed method was compared against three known classifiers (Ldc, Qdc, 1-nn) which reduced the original number of features using PCA. The proposed method achieved and overall accuracy of 88.13% which was the best against the other three (Ldc: 68.75%, Qdc: 72.50%, 1-nn: 80.63%). In Gavrilis et al. (2006), a variation of the proposed method was used to compress a large set of original features (15,000) into 20 features that were used for classification. After the compression, an average accuracy of 96–97% was achieved.

2. Related work

Representation transformations in classification and regression problems can be grouped into three different categories: feature selection, construction, and reduction through linear transformation. In the first case, a subset of the original features is defined, eliminating the irrelevant and less significant features. In the second case, a number of efficient solutions have already been proposed using dimensionality reduction techniques including genetic programming (GP) (Raymer et al., 2000), singular value decomposition, and principal component analysis. In a more sophisticated approach, the linear weights are estimated by a fully automatic algorithm (Jarmulak, 1999). The feature construction method is more complex, since it usually involves features selection, followed by linear and non-linear-transformations of the primitive set. The evolutionary approach is among the most recent methods of constructing new features from the original set of primitive features (Krawiec, 2002).

Related work in the area of feature selection and construction, mostly involves feature selection rather than feature construction. In Jarmulak (1999), a genetic algorithm is used to reduce the initial set of 26 features, eliminating irrelevant features and improve the classification accuracy using a C4.5 decision tree and the nearest neighbor rule. The authors proposed an evolutionary approach to select the relevant features, and real valued genes by assigning weights to the original features. The experimental results show that the selection method improves the classification rate.

Raymer et al. (2000) a genetic algorithm (GA) in combination with a K-nearest neighbor classifier to select a subset of the original feature set in the bio-chemistry domain. A binary representation in the chromosome is used to select or discard a feature. The number of centers of the KNN classifier is also encoded in the chromosome and weights are assigned to each feature. Experimental results in a bio-chemistry domain dataset show that the GA/KNN performed well along eight the following classifiers:

1. The linear discriminant
2. The quadratic discriminant
3. Nearest neighbor
4. Bayes (independent)
5. Bayes (2nd order)
6. Neural network (back prop)
7. Predictive value max.
8. CART tree

The genetic programming framework and a tree-like representation have already been presented in discovering new features for classification problems by Krawiec (2002).

In this paper, a fully automatic features selection and construction (FSC) method for robust pattern recognition is presented, introducing a novel feature construction method with certain advantages. Taking into account the close relation between the features vector and the classification system, three problems are solved simultaneously in a single optimization process: the features selection, construction, and the training process of the classification system. The proposed FSC method is based on the original genetic programming method proposed by Krawiec (2002), enriched by several innovations and extensions: The use of a Backus Naur form (BNF) description is less restrictive, as the features construction form is concerned. Thus, the resulting function can become very complicated, if necessary, using the appropriate number of the original features. New basic functions can be easily introduced, and complex restrictions can be encoded in the BNF description. The genetic search prefers simpler expressions and the new features are normalized automatically. The proposed data-driven FSC method detects an effective set of features dedicated to the pattern classification module involved in the GP optimization function including also the training process of the pattern classification system.

The proposed FSC method is evaluated using twenty artificial and real well-known datasets and the experimental results are compared with a genetic feature selection method (Raymer et al., 2000), and the popular features reduction method known as principal component analysis. In all experiments, the multi-fold cross-validation methodology is used in order to make the datasets less prone to both training and classification error estimation due to the limited number of examples.

Please cite this article in press as: Gavrilis, D. et al., Selecting and constructing features using grammatical evolution, Pattern Recognition Lett. (2008), doi:10.1016/j.patrec.2008.02.007

The structure of the paper is as follows. In Section 3, the original grammatical evolution FSC method is presented. In Section 4, a detailed description of the features extraction and construction method is given. A short presentation of the evaluation process, the compared methods and the datasets as well as the experimental results are described in Section 5. In the last section, the advantages and the drawbacks of the proposed method along with some conclusions are presented.

3. Grammatical evolution

Grammatical evolution (O'Neill and Ryan, 2001) is a class of methods that use an evolutionary algorithm and a context-free grammar in BNF notation to create a sequence of terminal symbols in an arbitrary language. A context-free grammar (G) is a formal grammar in which all production rules are in the form $V \rightarrow w$ where V is a non-terminal symbol and w is a sequence of terminal and non-terminal symbols. A context-free grammar can be represented by the quad-tuple:

$$G = (V_{\mathrm{T}}, V_{\mathrm{N}}, P, S)$$

where V_{T} is a finite set of terminal symbols, V_{N} is a finite set of non-terminal symbols, P is a set of production rules, and S is a non-terminal symbol known as the *Start* symbol.

Typically, in grammatical evolution methods genes are represented as integer numbers and each gene denotes a production rule from set P. The algorithm gradually replaces all non-terminal symbols with the right-hand of the selected production rule starting from the start symbol (S). The replacement procedure is performed using the rule number:

$$\mathrm{RULE} = B \mathrm{mod} R_{\mathrm{N}}$$

where B is a gene and R_{N} the number of rules for the specific non-terminal symbol. This symbol-replacement process is repeated until the end of the chromosome is reached. If at the end of the chromosome no valid expression has been produced, the algorithm starts again from the beginning of the chromosome (wrapping effect) or the mapping procedure is terminated by assigning a very small fitness value to the corresponding chromosome. The grammatical evolution procedure has been used with success in many fields such as symbolic regression (O'Neill and Ryan, 2003), discovery of trigonometric identities (Ryan et al., 1998), robot control (Collins and Ryan, 2000), caching algorithms (O'Neill and Ryan, 1999), financial prediction (Brabazon and O'Neill, 2003) etc.

4. Features selection and construction

The proposed FSC method is based on grammatical evolution by constructing new features from existing ones in order to improve the classification rate of an arbitrary classifier (Fig. 2). The FSC processing steps are the following:

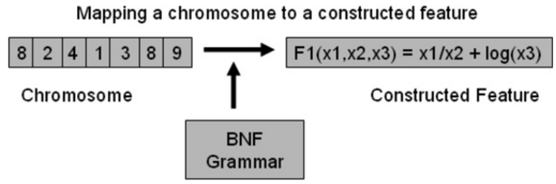
Mapping a chromosome to a constructed feature
8 2 4 1 3 8 9
F1(x1,x2,x3) = x1/x2 + log(x3)
Chromosome
Constructed Feature
BNF Grammar

Fig. 2. The feature $f_1(..)$ is constructed from the original features (x_1, x_2, x_3) through BNF grammar and a gene sequence.

(1) *The data set is split in two independent sets, train and test set*. The train set is used for the features construction and the test set is used for features evaluation in the adopted classification method.

(2) *The genetic algorithm parameters are defined*. The parameter N_f determines the number of features constructed or selected from the original set. The parameter N_g denotes the total number of chromosomes in the genetic population and L_g denote the size of each chromosome. The proposed algorithm uses fixed – length chromosomes instead of variable – length. This restriction limits the creation of very large expressions decreasing also the search space. The parameter R_s determines the fraction of the number of chromosomes that will go through unchanged to the next generation, and the R_{m} controls the mutation rate i.e. the average number of random changes inside a chromosome.

(3) *The grammar G is defined*. A context-free grammar, describing all the possible algebraic expressions of the original set of features is created. A typical example is shown of Fig. 6, where the constructed features are created using valid ordinary arithmetic operations and the original set of features as shown in the subset of rules presented in Fig. 7. The numbers in parentheses denote the sequence number of the corresponding production rule to be used in the selection procedure. N is the total number of original features. The symbol S denotes the start symbol of the grammar.

(4) *Chromosome initialization*. Every part of each chromosome in the genetic pool is initialized randomly in the range $[0, 255]$.

(5) *Fitness evaluation*. Each chromosome g is evaluated as follows:

String	Chromosome	Operation
<expr>	9,8,6,4,16,10,17,23,8,14	9 mod 3=0
(<expr><op><expr>)	8,6,4,16,10,17,23,8,14	8 mod 3=2
(<terminal><op><expr>)	6,4,16,10,17,23,8,14	6 mod 2=0
(<xlist><op><expr>)	4,16,10,17,23,8,14	4 mod 3=1
(x2<op><expr>)	16,10,17,23,8,14	16 mod 4=0
(x2+<expr>)	10,17,23,8,14	10 mod 3=1
(x2+<func>(<expr>))	17,23,8,14	17 mod 4=1
(x2+cos(<expr>))	23,8,14	23 mod 3=2
(x2+cos(<terminal>))	8,14	8 mod 2=0
(x2+cos(<xlist>))	14	14 mod 3=2
(x2+cos(x3))		

Fig. 3. Example of the feature construction process.

Please cite this article in press as: Gavrilis, D. et al., Selecting and constructing features using grammatical evolution, Pattern Recognition Lett. (2008), doi:10.1016/j.patrec.2008.02.007

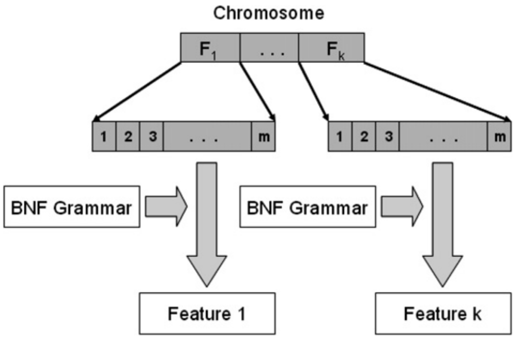

Fig. 4. Features construction using a k-genes chromosome and a BNF grammar to generate in parallel valid algebraic expressions.

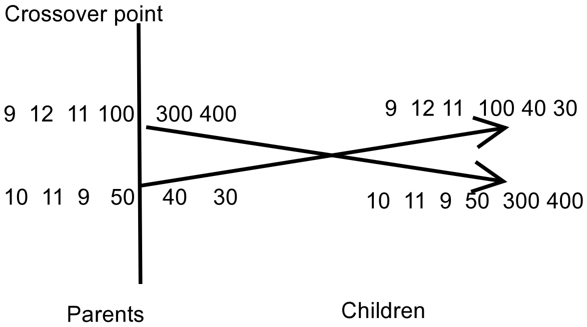

Fig. 5. One point crossover.

(a) The chromosome is split into N_f equal parts. Each part $g_i,\ i=1,\ldots,N_f$, is used to construct a feature.

(b) Denote by $f_i,\ i=1,\ldots,N_f$ the features that are constructed from the each part g_i using the mapping process. This process can be executed in parallel and it is shown in Fig. 4. An example of the feature construction process is given in Fig. 3, where the valid function $f(x)=x_2+\cos(x_3)$ is produced from the chromosome $x=(9,8,6,4,16,10,17,23,8,14)$. Starting from the non-terminal symbol $\langle\text{expr}\rangle$, the next integer is 9 and $\langle\text{expr}\rangle$ has 3 rules. The selected rule number is 3 ($9 \bmod 3=0$), thus the corresponding rule $\langle\text{expr}\rangle ::= (\langle\text{expr}\rangle\langle\text{op}\rangle\langle\text{expr}\rangle)$ is applied. The process is repeated until all the integers in g_i are used.

(c) The original data sets both train and test are transformed, according to the constructed features transformation f_i functions, to the new features data sets. The new train set is used to train the classification system and the fitness of the chromosome g_i is set equal to the classification accuracy. In regression problems, the fitness value is estimated by the negative value of the mean square error between the actual and the predicted values.

(6) *Chromosome transformation using the genetic operators*. The genetic operators of crossover and mutation are applied to the genetic population forming the next generation of chromosomes. In the crossover procedure, $n=(1-R_s)N_g$ new chromosomes are created. The new population replace the chromosomes with the lowest fitness value in the current generation. Pair of chromosomes, randomly selected parents from the current pool, are cut at a randomly chosen point and the right-hand sub-chromosomes are exchanged as shown in Fig. 5. The parents are selected through tournament selection i.e. first a group of $K>2$ randomly chosen chromosomes is formed and the individual with the best fitness in the group is selected, the others are discarded. In the mutation procedure, for every element in a chromosome a random number in the range $[0,1]$ is chosen. If this number is less than or equal to the mutation rate R_m the corresponding element is changed randomly, otherwise it is remained intact.

(7) *Termination check*. If the maximum number of generations is reached or the best chromosome has fitness value (classification accuracy) that exceeds a predefined threshold, then the features construction process terminates, otherwise a features re-estimation process is starting again from step 6.

5. Experiments and datasets

5.1. Setup of the experiments

The features construction method has been evaluated in several experiments including both artificial and real pattern classification and regression problems. Half of these

```
S::=<expr> (0)
<expr> ::= ( <expr> <op> <expr> ) (0)
           | <func> ( <expr> ) (1)
           |<terminal> (2)
 <op> ::= + (0)
           | - (1)
           | * (2)
           | / (3)
<func> ::= sin (0)
           |cos (1)
           |exp (2)
           |log (3)
<terminal>::=<xlist> (0)
           |<digitlist>.<digitlist> (1)
<xlist>::=x1 (0)
           | x2 (1)
           .....
           |xN (N-1)
<digitlist>::=<digit> (0)
           | <digit><digitlist> (1)
<digit> ::= 0 (0) | 1 (1) | 2 (2) .... | 9 (9)
```

Fig. 6. Grammar of the proposed method for the feature construction case.

```
S::=<xlist> (0)
<xlist>::=x1 (0)
             | x2 (1)
             .....
             |xN (N-1)
```

Fig. 7. Grammar of the proposed method for the feature selection case.

datasets refer to classification problems and the other half to regression problems. The number of the original parameters in these problems is varied between 2 and 34. The number of requested features are less or equal to the number of the original features, but this restriction can be omitted in the case where the original set of features are non-linear extended in a large features set. In some of the datasets, the initial vector was expanded to include noise parameters. The parameters of the genetic algorithm in all experiments remain the same as shown in Table 1. The size of each chromosome was $40K$ where K is the requested number of features. All the experiments ran on a Beowulf cluster of 48 nodes (Athlon 1800 + running Debian Linux). The evaluation of the produced features was made with the FunctionParser programming library (Nieminen and Yliluoma, 2005). The number of constructed features was limited to 3 for all datasets except for the Spiral, Spiral2 and the Circular dataset, where it was limited to 2.

The features produced from the best chromosome of the genetic population, are applied to the train set as well as in the test set. The resulting train set is used for the training of different models such as, multi-layer perceptron, RBF-NN, K-nearest neighbor (for the case of classification). The classifiers used for the evaluation of the produced features are a KNN classifier, an RBF-NN (Haykin, 1999, Bishop, 1995) and an MLP-NN trained using a Powell's variant of the BFGS algorithm (Powell, 1989), known also as the Tolmin local optimization procedure. In the RBF-NN, the hidden-layer weights are estimated using the K-means algorithm and the pseudo-inverse method is used to derive the output-layer weights, using five hidden nodes.

Both RBF-NN and MLP-NN have one hidden-layer that varies from 1 to 10 neurons, and two output neurons. In the case of the KNN classifier, K varies from 1 to 10. In the direction to minimize the influence of the sub-optimal training process applied to NNs, each experiment was repeated 30 times and the mean classification error (MCE) is used to evaluate the features quality.

Table 1
Genetic algorithm parameters

Parameter	Value
Maximum number of generations	200
Population size	500
Chromosome length	$40K$
Mutation rate	0.05
Tournament size	10
Selection rate	0.05

5.2. Classification datasets

The classification datasets were found in the Machine Learning Repository in the following URL: http://archive.ics.uci.edu/ml/ as well as some artificial datasets. The description of the classification datasets has as follows:

(1) Circular artificial data: The circular artificial dataset (CIRCULAR) contains 1000 examples that belong to two categories (500 examples each). The two-dimensional data in the first class belong to a circle and the data of the second class belong to a circular disc outside the first circle. Each feature vector is expanded by adding 3 more features using a random number generator which follows the same normal distribution for both classes.
(2) Spiral artificial data: The spiral artificial dataset (SPIRAL) contains 1000 two-dimensional examples that belong to two classes (500 examples each). The number of the features is 2. The data in the first class are created using the following formula: $x_1 = 0.5t\cos(0.08t)$, $x_2 = 0.5t\cos(0.08t + \frac{\pi}{2})$ and the second class data using: $x_1 = 0.5t\cos(0.08t + \pi)$, $x_2 = 0.5t\cos(0.8t + \frac{3\pi}{2})$.
(3) Spiral artificial data-2: The second spiral dataset (SPIRAL2) is created as the first dataset. Its difference is that its primitive set is expanded by adding 3 more noisy features using the same normal distribution for both classes.
(4) Ionosphere dataset: The Ionosphere dataset (IONOSPHERE) contains data from the Johns Hopkins Ionosphere database. The two-class dataset contains 351 examples of 34 features each.
(5) Pima Indians diabetes: The PIMA datasets contains 768 examples of 8 features each that are classified into two categories: healthy and diabetic.
(6) Wisconsin diagnostic breast cancer: The Wisconsin diagnostic breast cancer dataset (WDBC) contains data for breast tumors. It contains 569 training examples of 30 features each that are classified into two categories.
(7) Wine: The Wine recognition dataset (WINE) contains data from Wine chemical analysis. It contains 178 examples of 13 features each that are classified into three classes.
(8) Glass: This dataset (GLASS) contains glass component analysis for glass pieces that belong to 6 classes. The dataset contains 214 examples with 10 features each.
(9) Liverdisorder: This dataset contains blood analysis data from people with liver disorders. It consists of 345 examples of 6 features each.

5.3. Regression datasets

The regression datasets are available from the Statlib URL: http://lib.stat.cmu.edu/datasets/. The description of these datasets has as follows:

Please cite this article in press as: Gavrilis, D. et al., Selecting and constructing features using grammatical evolution, Pattern Recognition Lett. (2008), doi:10.1016/j.patrec.2008.02.007

(1) BK: This dataset comes from Smoothing methods in statistics (Simonoff, 1996) and it used to estimate the points scored per minute in a basketball game. The dataset has 96 patterns of 4 features each.

(2) BL: This dataset can be downloaded from StatLib. It contains data from an experiment on the affects of machine adjustments on the time to count bolts. It contains 40 patters of 7 features each.

(3) FA: The FA dataset contains percentage of body fat, age, weight, height, and ten body circumference measurements. The goal is to fit body fat to the other measurements. The number of the features is 18 and the total number of patterns is 252.

(4) FY: This dataset measures the longevity of fruit flies depending on increased reproduction. This dataset has 125 patterns of 4 features each.

(5) LW: This dataset is produced from a study that was to identify risk factors associated with giving birth to a low birth weight baby. Data were collected on 189 women, 59 of which had low birth weight babies and 130 of which had normal birth weight babies. The number of features in this dataset is 9.

(6) MB: This dataset is available from Smoothing methods in statistics (Simonoff, 1996) and it has 61 patterns and the number of features is 2.

(7) NT: This dataset contains data from (Mackowiak et al., 1992) that examined whether the true mean body temperature is 98.6 F. The number of patterns is 131 and the number of features 2.

(8) PO: This dataset has its source in McDonald and Schwing (1973) and contains pollution data. The dataset contains 60 patterns of 15 features each.

(9) PW: This dataset contains numeric prediction data using instance-based learning with encoding length selection. The source of this dataset is in Kilpatrick and Cameron-Jones (1998). It contains 200 patterns of 10 features each.

(10) SN: This dataset contains data on a viticultural experiment that was conducted to investigate different methods of trellising and pruning. It is available from STATLIB. It contains 576 patterns of 11 features each.

5.4. Experimental results

In the following tables, the results from the application of the proposed method in the test problems are listed. In all tables, the cell holding the best value is identified by the bold font and the number in the cells represent classification error for the case of classification datasets and MSE for the case of regression datasets. In all cases 3 features were selected or constructed except for the cases of Circular, Spiral, Spiral2, MB and NT where 2 features were selected or constructed.

In Table 2 the results from the application of an MLP with 10 hidden nodes in each dataset are listed. The column MLP denotes the test error of the MLP for the original dataset, the column MLP-PCA denotes the test error of the MLP for the dataset that was produced by the original with the use of the PCA method. The column MLP(FS) denotes the test error of the MLP for the dataset that was modified by selecting with grammatical evolution the 3 best features. Finally, the column MLP(FC) denotes the test error from the application of the MLP to the 3 best features created by the proposed method.

Table 2
Results from the application of MLP

DATASET	MLP	MLP-PCA	MLP(FS)	MLP(FC)
Wine	48.86%	9.74%	17.04%	**7.81%**
Glass	53.93%	60.28%	52.83%	**43.18%**
Liverdisorder	38.86%	40.08%	**32.43%**	33.43%
Ionosphere	17.08%	21.59%	12.39%	**9.43%**
Wdbc	20.91%	6.06%	8.44%	**4.56%**
Pima	35.65%	26.35%	27.61%	**26.07%**
Circular	7.95%	40.77%	6.11%	**4.14%**
Spiral	45.31%	47.32%	45.31%	**36.38%**
Spiral2	48.47%	50.20%	48.47%	**41.21%**
BK	4.44	0.38	0.23	**0.17**
BL	0.33	0.53	0.13	**0.00**
FA	0.44	0.17	**0.13**	0.20
FY	218.09	1.11	28.17	**0.37**
LW	0.91	0.30	**0.13**	0.19
MB	0.68	0.42	**0.27**	0.42
NT	3.14	1.09	1.20	**0.12**
PO	0.25	0.32	0.21	**0.21**
PW	0.21	0.26	0.15	**0.08**
SN	2.31	0.47	**0.44**	0.48

In Table 3 the results from the application of an RBF with 10 hidden units are presented. The column RBF denotes the test error for the case where the original features are used. The column RBF(PCA) denotes the case where the three best features derived from the PCA procedure were used. The column RBF(FS) denotes the case

Table 3
Results from the application of RBF

DATASET	RBF	RBF(PCA)	RBF(FS)	RBF(FC)
Wine	25.44%	8.00%	11.30%	**6.70%**
Glass	**45.61%**	60.28%	54.70%	48.75%
Liverdisorder	31.48%	43.80%	30.31%	**29.58%**
Ionosphere	16.06%	21.40%	13.71%	**9.89%**
Wdbc	7.85%	6.62%	7.24%	**3.51%**
Pima	28.12%	27.03%	25.93%	**25.10%**
Circular	5.94%	38.93%	3.59%	**3.49%**
Spiral	45.74%	46.13%	45.74%	**28.02%**
Spiral2	44.70%	49.63%	44.70%	**34.07%**
BK	0.11	0.16	**0.10**	0.18
BL	0.29	0.84	0.37	**0.00**
FA	0.14	0.13	0.12	**0.11**
FY	0.42	0.35	**0.31**	0.35
LW	0.41	0.27	**0.12**	0.13
MB	1.57	**0.30**	1.57	0.40
NT	0.43	0.27	0.37	**0.10**
PO	0.37	0.38	**0.13**	0.26
PW	0.10	0.68	0.13	**0.08**
SN	0.47	0.52	0.42	**0.36**

Table 4
Results from the application of KNN

Dataset	KNN (%)	KNN-PCA (%)	KNN(FS) (%)	KNN(FC) (%)
Wine	24.44	**5.56**	8.89	**5.56**
Glass	32.71	30.84	32.71	**28.97**
Liverdisorder	35.26	38.15	33.52	**30.06**
Ionosphere	12.50	13.64	12.50	**9.66**
Wdbc	7.37	5.96	4.56	**3.86**
Pima	29.43	27.60	24.48	**23.18**
Circular	5.40	37.40	2.80	**2.80**
Spiral	43.00	37.40	43.00	**25.10**
Spiral2	45.50	43.80	45.50	**36.40**

where the three best features selected by the proposed method were used. Finally, the column RBF(FC) the results from the application of the RBF to the 3 best features constructed by the proposed method are listed.

In Table 4 the results from the application of a KNN with $K = 10$ to the classification datasets are listed. The column KNN denotes the test error for the case where the original features are used. The column KNN(PCA) denotes the case where the three best features derived from the PCA procedure were used. The column KNN(FS) denotes the case where the three best features selected by the proposed method were used. In column KNN(FC) the results from the application of the KNN to the 3 best features constructed by the proposed method are listed. Furthermore, we display in Figs. 8 and 9 the experimental results for the above methods with K varying from 1 to 10.

In almost all experiments, the proposed method constructs a more efficient set of three artificial features and in some experiments the selection technique guided by the grammatical evolution procedure can outperform the other methods. Especially in the case of the SPIRAL data, one of the most complex pattern classification problems using artificial data, the proposed method defines a set of two more efficient artificial features, and in all experiments the irrelevant (noisy) features are not used to construct the artificial features (case SPIRAL2). In case of a large set of primitive features (IONOSPHERE, WDBC) the proposed method decreases the minimum classification error using only the three artificial features against a number of more than 30 primitive features. The proposed method is outperformed in only two cases by the PCA method. The fact that PCA does not give the best results in most experiments shows that using the variance or information for linear dimensionality reduction does not necessary improve classification performance.

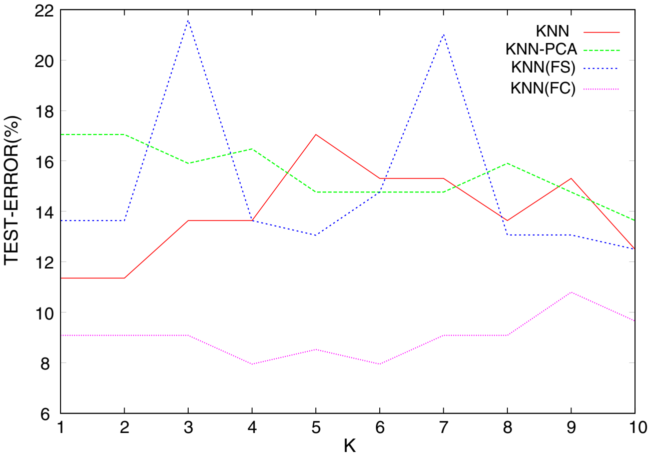

Fig. 8. KNN results for the Wine dataset.

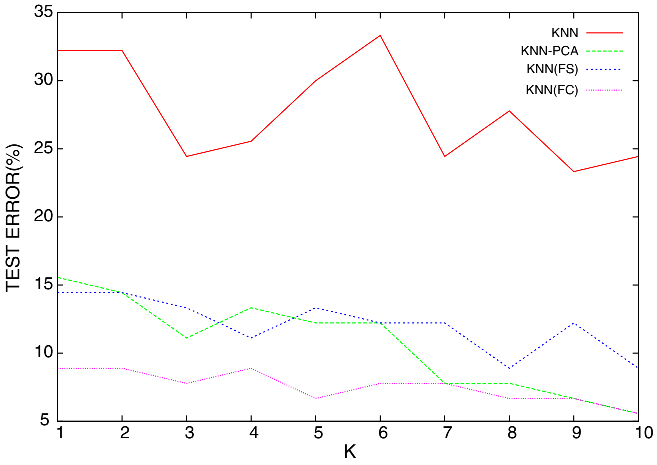

Fig. 9. KNN results for the Ionosphere dataset.

6. Conclusions

The proposed FSC method, which is based on a combination of grammatical evolution and artificial neural networks, has been used to improve the classification and regression accuracy in many classification and regression datasets. Three different classifiers have been used (RBF-NN, MLP-NN and KNN) showing that the proposed method can be used as a wrapper for a variety of classification methods. The proposed method has also been compared to feature selection methods using genetic algorithms and dimensionality reduction using principal component analysis.

In the proposed method the objective function used for the feature construction process includes the MCE for the classification problems and the MSE in the case of regression problems, in contrast to the maximization of variance or information criteria used by other methods. This important advantage leads to the construction of robust features set and hence giving better classification results as shown in the corresponding tables.

The proposed algorithm can be used to improve the efficiency of classification and regression problems by constructing new features from existing ones. Future work will include more classifiers along with a method to propose an optimal grammar depending on each problem.

References

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford University Press.

Brabazon, A., O'Neill, M., 2003. A grammar model for foreign-exchange trading. In: Arabnia, H.R., et al. (Ed.), Proc. Internat. Conf. on Artificial Intelligence, vol. II. CSREA Press, pp. 492–498.

Collins, J.J., Ryan, C., 2000. Automatic generation of robot behaviors using grammatical evolution. In: Proc. AROB 2000, 5th Internat. Symposium on Artificial Life and Robotics.

Cover, T.M., 1965. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Trans. Electron. Comput. EC-14, 326–334.

Gavrilis, D., Tsoulos, I.G., Dermatas, E., 2006. Neural recognition and genetic features selection for robust detection of E-mail spams. In: SETN 2006, vol. 1, pp. 498–501.

Georgoulas, G., Gavrilis, D., Tsoulos, I.G., Stylios, C., Bernades, J., Groumpos, P.P., 2007. Novel approach for fetal heart rate classification introducing grammatical evolution. Biomed. Signal Process. Control 2, 69–79.

Haykin, S., 1999. Neural Networks A Comprehensive Foundation. Prentice Hall.

Jarmulak, J., 1999. Genetic algorithms for feature selection and weighting. In: Proc. IJCAI.

Kilpatrick, D., Cameron-Jones, M., 1998. Numeric prediction using instance-based learning with encoding length selection. In: Progress in Connectionist-Based Information Systems. Springer-Verlag, Singapore.

Krawiec, K., 2002. Genetic programming-based construction of features for machine learning and knowledge discovery tasks. Genetic Program. Evol. Machine. 3, 329–343.

Mackowiak, P.A., Wasserman, S.S., Levine, M.M., 1992. A critical appraisal of 98.6 degrees f, the upper limit of the normal body temperature, and other legacies of Carl Reinhold August Wunderlich. J. Amer. Med. Assoc. 268, 1578–1580.

McDonald, G.C., Schwing, R.C., 1973. Instabilities of regression estimates relating air pollution to mortality. Technometrics 15, 463–482.

Nieminen, J., Yliluoma, J., 2005. Function Parser for C++, v2.8. <http://warp.povusers.org/FunctionParser/>.

O'Neill, M., Ryan, C., 1999. Automatic generation of caching algorithms. In: Miettinen, Kaisa, Mkel, Marko M., Neittaanmki, Pekka, Periaux, Jacques (Eds.), Evolutionary Algorithms in Engineering and Computer Science. John Wiley & Sons, pp. 127–134.

O'Neill, M., Ryan, C., 2001. Grammatical evolution. IEEE Trans. Evolut. Comput. 5, 349–358.

O'Neill, M., Ryan, C., 2003. In: Grammatical Evolution: Evolutionary Automatic Programming in a Arbitrary Language, of Genetic Programming, vol. 4. Kluwer Academic Publishers.

Powell, M.J.D., 1989. A tolerant algorithm for linearly constrained optimization calculations. Math. Program. 45, 547–566 (547).

Raymer, M.L., Punch, W.F., Goodman, E.D., Kuhn, L.A., Jain, A., 2000. Dimensionality reduction using genetic algorithms. IEEE Trans. Evol. Comput. 4, 164–171.

Ryan, C., O'Neill, M., Collins, J.J., 1998. Grammatical evolution: Solving trigonometric identities. In: Proc. Mendel 1998, 4th Internat. Mendel Conf. on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks, Rough Sets, Technical University of Brno, Faculty of Mechanical Engineering, pp. 111–119.

Simonoff, J.S., 1996. Smoothing Methods in Statistics. Springer-Verlag.

Please cite this article in press as: Gavrilis, D. et al., Selecting and constructing features using grammatical evolution, Pattern Recognition Lett. (2008), doi:10.1016/j.patrec.2008.02.007