Differential Evolution based Feature Subset Selection

Rami N. Khushaba, Ahmed Al-Ani, and Adel Al-Jumaily
Mechatronics and Intelligent Systems – Faculty of Engineering
University of Technology, Sydney
P. O. Box: 123, Broadway 2007, Sydney - Australia
E-mail:[Rkhushab, Ahmed, Adel@eng.uts.edu.au]

Abstract

In this paper, a novel feature selection algorithm based on the use of Differential Evolution (DE) optimization technique is presented. The new algorithm, called DEFS, is the first attempt in which a real problems optimizer like DE is modified to suit the problem of feature selection. The proposed DEFS highly reduces the computational costs while at the same time proving to present powerful performance. The DEFS technique is applied in a brain-computer-interface (BCI) application and compared with other dimensionality reduction techniques. The practical results indicate the significance of the proposed algorithm in terms of both solutions optimality and memory requirement.

1. Introduction

Feature selection (FS) is an indispensable dimensionality reduction technique commonly used with high dimensional data. The FS techniques study how to select a subset of attributes or variables that are used to construct models describing data. The reason behind using FS techniques include reducing dimensionality, removing irrelevant and redundant features, reducing the amount of data needed for learning, improving algorithms' predictive accuracy, and increasing the constructed models' comprehensibility [1].

As a part of any feature subset selection algorithm, there are several factors that need to be considered, the most important are [2]: the evaluation measure and the search procedure. The existing evaluation measures utilized in feature selection techniques are divided into two categories according to their dependency on the classification algorithms, those are: filters and wrappers. Filter based feature selection methods are in general faster than wrapper based methods. This is due to the fact that the filter based methods depend on some type of estimation of the importance of individual features or subset of features. Comparing with the filter methods, wrapper based methods are more accurate as the importance of feature subsets is measured using a classification algorithm.

On the other hand, a search strategy is needed to explore the feature space. Various search algorithms that differ in their optimality and computational cost have been developed to search the solution space. These methods include for example: Tabu Search (TS) [3], Simulated Annealing (SA) [4], Genetic algorithms (GA) [5] and other methods. Another trend of search procedures is based on swarm intelligence including both the Ant Colony Optimization (ACO) [6], and Particle Swarm Optimization (PSO) [7].

This paper, proposes a novel approach in which a real number optimizer is modified to serve the purpose of feature selection. Differential Evolution (DE) optimization technique is employed as a search procedure due to its fast convergence properties and solutions optimality as was proved by Price et al [8].

This paper is structured as follows: Section 2 introduces the reader to the DE optimization technique. Section 3 describes the proposed DE-based feature selection algorithm. Practical results are presented in section 4. Finally, a conclusion is given in section 5.

2. Differential Evolution

Differential Evolution (DE) is an optimization method, capable of handling non-differentiable, nonlinear and multimodal objective functions. It is a

simple, parallel, direct search, and easy to use method having good convergence and fast implementation properties [8]. The crucial idea behind DE is a new scheme for generating trial parameter vectors by adding the weighted difference vector between two population members (r_1, and r_2) to a third member (r_3). The following equation shows how to combine three different, randomly chosen vectors to create a mutant vector, $V_{i,g}$ from the current generation g:

$$V_{i,g} = X_{r0,g} + F \,.\, (X_{r1,g} - X_{r2,g}) \qquad (1)$$

where $F \in (0, 1)$ is a scale factor that controls the rate at which the population evolves.

Extracting both distance and direction information from the population to generate random deviations result in an adaptive scheme that has an excellent convergence property. DE also employs uniform crossover, also known as discrete recombination, in order to build trial vectors out of parameter values that have been copied from two different vectors. In particular DE crosses each vector with a mutant vector:

$$U_{j,i} = \begin{cases} V_{j,i} & if\,(rand(0,1) \le Cr \;or\; j = j_{rand}) \\ X_{j,i} & otherwise \end{cases} \qquad (2)$$

where $U_{j,i}$ is the j'th trial vector along i'th dimension. The crossover probability, $Cr \in [0,1]$, is a user defined value that controls the fraction of parameter values that are copied from the mutant. If the newly generated vector results in a lower objective function value (higher fitness) than the predetermined population member, then the resulting vector will replace the vector with which it was compared [9].

To this end, DE optimization may not be used directly in feature selection problems. The next section identifies a simple way to utilize a modified DE in feature selection problems.

3. The Proposed Feature Selection Technique

The proposed DE-based FS technique is shown schematically in Fig. 1. The first step in the algorithm is to generate new population vectors from the original population. For each position in the population matrix, a new mutant vector is formed by first selecting two random vectors; then performing a weighted difference, adding the result to a third random (*base*) vector. The mutant vector is then crossed with the original vector that occupies that position in the original matrix. The result of this operation is called a *trail* vector. The corresponding position in the new population will contain either the trail vector (or its corrected version) or the original target vector depending on which one of them achieved a higher fitness (classification accuracy).

Due to the fact that a real number optimizer is being used, nothing will prevent two dimensions from settling at the same feature coordinates. As an example, if the resultant vector is [244.3024 30.1646 48.1263 43.4240 243.8665], then the rounded value of the resulting vector would be [244 30 48 43 244]. This result is completely unacceptable within feature selection problems, as a certain feature (feature index = 244) is used twice. In order to overcome such a problem, we propose to employ feature distribution factors to replace duplicated features. A roulette wheel weighting scheme is utilized. In this scheme a cost weighting is implemented in which the probabilities of individual features are calculated from the distribution factors associated with each feature. The distribution factor of feature f_i is given by the following equation:

$$FD_i = a_1 \times \left(\frac{PD_i}{PD_i + ND_i} \right) + a_2 \times \left(1 - \frac{(PD_i + ND_i)}{\max(PD_i + ND_i)} \right) \qquad (3)$$

where a_1, a_2 are constants. PD_i is the positive distribution factor that is computed from the subsets that achieved an accuracy that is higher than the average accuracy of the whole subsets. ND_i is the negative distribution factor that is computed from the subsets that achieved an accuracy that is lower than the average accuracy of the whole subsets. This is shown in Fig.2 schematically with the light gray region being the region of elements achieving less error than the average error values and the dark gray being the region with elements achieving higher error rates than the average. The rationale behind Eq. (3) is to replace the replicated parts of the trail vectors according to two factors. The $PD_i/(PD_i+ND_i)$ factor indicates the degree to which f_i contributes in forming good subsets. On the other hand the second term in Eq. (3) aims at favoring exploration, where this term will be close to 1 if the overall usage of a specific feature is very low.

In order to better understand the algorithm, consider the same example with redundancies above. The aim here is to correct the current trail vector [**244** 30 48 43 **244**] and replace duplicated feature with another one that is most relevant to the problem. Let's presume that the features ranked by the roulette wheel according to the highest distribution factors are [55, 244, 30, 210, 68, 74]. After excluding features that appear in the trial vector, the rest can be used to replace the duplicated features of the trail vector.

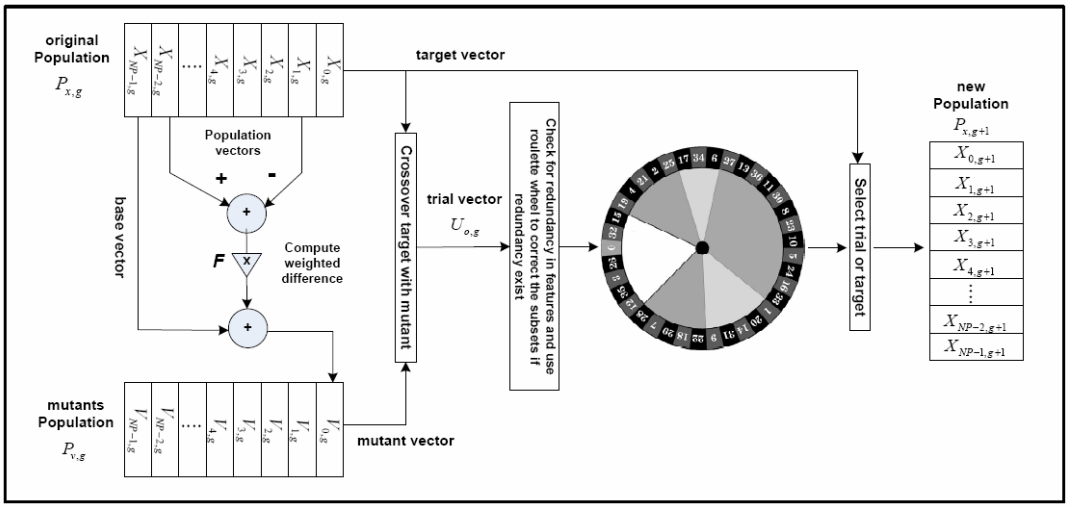

Fig.1. the proposed DEFS algorithm

Thus for our example, the trial vector would be represented by [244 30 48 43 55].

In order to compare the memory requirements with other evolutionary techniques like GA and PSO, let's consider the same example again. When selecting only 5 features and setting the population size to 50, then the population size of the proposed DE-based FS technique will be (50 x 5). On the other hand, both GA and PSO employ binary strings to represent the feature subsets. In such techniques, every bit represents an attribute. The value of '1' means that the attribute is selected while '0' means not selected. This will increase the memory requirements for large problems. If the original dataset consists of 300 features then for both GA and PSO the resulting population size would be (50 x 300), which is indeed much larger than the one needed by DE. It is important to mention that other population based FS methods also need to apply certain restrictions to the search process. For instance, if the crossover and mutation operators of GA are applied without any restrictions, then the feature subsets may consist of higher or lower number of features than what is desired.

4. Experiments and Results

In order to prove the effectiveness of the proposed DE-based FS algorithm, the method was tested in a brain in a brain-computer-interface (BCI) problem. The application involves the classification of the Electroencephalogram (EEG) signal from human brain. The data used here was obtained from the Department of Medical Informatics, University of Technology, Austria.

EEG signals were recorded form three right-handed females with 56 Ag/AgCl Electrodes using monopolar montage, with reference electrode on the right ear. The subjects were placed in an armchair and asked to imagine right or left finger movements according to stimuli on screen. A total of 8 seconds of data were recorded at 128 Hz sampling rate, 2 seconds before the stimuli and 6 after it. A total of 406 trials were used, 208 for left movement and 198 for right. The wavelet packet transform was used in this paper to extract features from this dataset. The total number of features extracted was 168 features (56 channels x 3 features/channel). For more information about the feature extraction process the reader can refer to [10].

The proposed DEFS algorithm was tested against other parallel search algorithms like GA and PSO. The desired number of selected features was varied from 3 to 99 features. Each of the mentioned algorithms was executed for ten times for each of the given number of desired features. For example when selecting 9 features, each method was used ten times and the average result is reported here. It is also worth to mention that the same initial population was used within all the methods. The results of the comparison are shown in Fig. 3.

In order to analyze the results, one can start by first looking at the performance of the methods when selecting a small feature subset. It is clearly shown that the proposed DEFS algorithm achieved higher classification accuracies than both PSO and GA in almost all cases. In addition the performance of PSO is initially shown to be better than that of GA when selecting small number of features. When the number of selected feature increases, the PSO performance starts degrading and GA enhancing. This mainly

caused by the fact that the performance of PSO algorithm is sensitive to the dimensionality of the problem. In general, the DEFS exhibits the best performance followed by GA and then PSO. The highest classification accuracies achieved by all FS methods used in this paper are stated as: 91.60% for DEFS, 90.75% for GA and 87.64% for PSO. The rationale behind the good performance of the DEFS is the due to fact that differential mutation can leads to better solutions with faster convergence properties as pointed out by price et al [8]. Another factor that contributes to the enhanced performance of the DEFS is the incorporation of feature distribution factors. This leads the population towards the vicinity which contains subsets with features that are most relevant to the problem.

5. Conclusion

A new feature selection method was presented in this paper based on the Differential Evolution optimization technique. The performance of the proposed algorithm was compared with other population based feature selection techniques like GA and PSO. It was shown that the proposed DEFS required smaller memory than other methods which yields a reduction in the computational cost. Also, when testing on a BCI problem, the proposed algorithm managed to outperform both GA and PSO in terms of classification performance yielding an accuracy of 91.6%. All of the results presented proved the effectiveness of the proposed DEFS algorithm.

Acknowledgment

The authors would like to thank the Department of Medical Informatics, University of Technology, Graz, Austria for providing the EEG data.

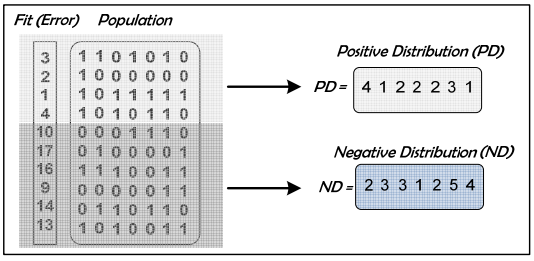

Fig.2 The feature distribution factors

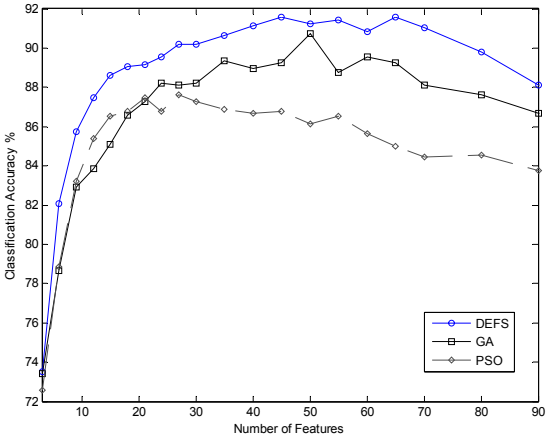

Fig.3. the performance of various methods with different number of features

References

[1] H. Liu, E. R. Dougherty, J. G. Dy, K. A. Torkkola, E. Tuv, H. A. Peng, C. A. Ding, F. A. Long, M. A. Berens, L. A. Parsons, Z. A. Zhao, L. A. Yu, and G. A. Forman, "Evolving feature selection," *Intelligent Systems, IEEE,* vol. 20, pp. 64-76, 2005.

[2] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence,* vol. 97, pp. 245-271, 1997.

[3] M. A. Tahir, A. Bouridane, F. Kurugollu, and A. Amira, "Feature selection using tabu search for improving the classification rate prostate needle biopsies," in *Proceedings of the 17th International Conference on Pattern Recognition*, Washington, DC, USA, 2004.

[4] M. Filippone, F. Masulli, and S. Rovetta, "Supervised classification and gene selection using simulated annealing," in *International Joint Conference on Neural Networks, IJCNN '06.*, pp.3566-3571, 2006.

[5] O. C. H. Frohlich, and B. Scholkopf, "Feature selection for support vector machines by means of genetic algorithms," in *Department of Mathematics and Computer Science*. vol. Diploma Thesis: Philipps-University Marburg, 2002.

[6] A. Al-Ani, "Feature subset selection using ant colony optimization," *International Journal of Computational Intelligence,* vol. 2, pp. 53 – 58, 2005.

[7] H. A. Firpi and E. Goodman, "Swarmed feature selection," in *Proceedingsof 33rd Applied Imagery Pattern Recognition Workshop*, pp.112-118, 2004.

[8] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A practical approach to global optimization*: Springer, 2005.

[9] A. K. Palit and D. Popovic, *Computational intelligence in time series forecasting: theory and engineering applications*: Springer, 2005.

[10] A. Al-Ani and A. Al-Sukker, "Effect of feature and channel selection on EEG classification," *Proceedings of the 28th IEEE EMBS Annual International Conference,* pp. 2171-2174, Aug 30-Sept 3, 2006.