Particle Swarm Optimization Based Feature Selection

Dr. S. Sivakumar
Teaching Assistant
Dept of Computer Science
Periyar University,Salem -636011.

Abstract- *Feature selection is the problem of selecting a subset of features without reducing the accuracy of representing the original set of features. Feature selection is used in many applications to remove irrelevant and redundant features where there are high dimensional datasets. These datasets may contain a high degree of irrelevant and redundant features that may decrease the performance of the classifiers. In this paper, continuous particle swarm optimization (PSO) is used to implement a feature selection in wrapper based method, and the k-nearest neighbor classification serve as a fitness function of PSO for the classification problem.*

Keywords- attribute reduction, PSO, k-NN, classification accuracy

I. INTRODUCTION

Classification is an important task in machine learning and data mining, which aims to classify each instance in the data into different groups. The feature space of a classification problem is a key factor influencing the performance of a classification/learning algorithm [1]. Without prior knowledge, it's hard to determine which features are useful. Therefore, a large number of features are usually introduced into the dataset, including relevant, irrelevant and redundant features. However, irrelevant and redundant features are not useful for classification. Their presence may mask or obscure the useful information provided by relevant features, and hence reduces the quality of the whole feature set [2]. Meanwhile, the large number of features causes one of the major obstacles in classification known as "the curse of dimensionality" [3]. Therefore, feature selection is proposed to increase the quality of the feature space, reduce the number of features and improve the classification performance [4-6].Feature selection aims to select a subset of relevant features that are necessary and sufficient to describe the target concept [7]. By reducing the irrelevant and redundant features, feature selection could decrease the dimensionality, reduce the amount of data needed for the learning process, shorten the running time, simplify the structure and/or improve the performance of the learnt classifiers [7].

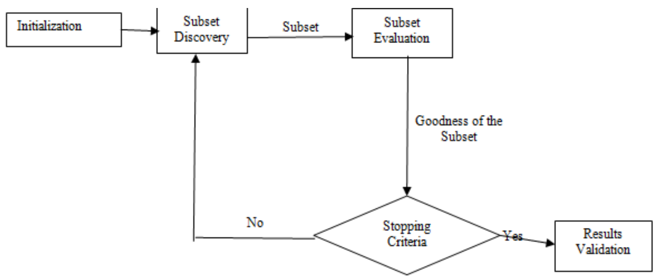

Figure 1: Feature subset selection and evaluation process

Naturally, an optimal feature subset is the smallest feature subset that can obtain the optimal performance, which makes feature selection a multi-objective problem [8]. Note that feature selection algorithms choose a subset of features from the original feature set and do not create new features. Feature selection is a difficult task. Although many approaches have been proposed, most of them still suffer from the problems of stagnation in local optima and high computational cost due mainly to the large search space. Therefore, an efficient global search technique is needed to address feature selection tasks.

Challenges of Feature Selection

Feature selection is a difficult problem [9,10], especially when the number of available features is large. The task is challenging due mainly to two reasons, which are feature interaction and the large search space. Feature interaction (also called epistasis [11]) frequently happens in classification tasks. There can be two-way, three-way or complex multiway interactions among features. On one hand, a feature, which is weakly relevant or even entirely irrelevant to the target concept by itself, can significantly improve the classification accuracy if it is complementary to other features. Therefore, the removal of such features may also miss the optimal feature subsets. On the other hand, an individually relevant feature can become redundant when working together with other features. The selection/use of such features brings redundancy, which may deteriorate the classification performance. In feature selection, the size of the search space grows exponentially with respect to the number of available features in the dataset (2^n possible subsets for n features) [12]. In most cases, it is practically impossible to search exhaustively all the candidate solutions. To better address this problem, a variety of search techniques have been applied to feature selection [12, 14]. However, existing methods still suffer from the problem of stagnation in local optima and/or high computational cost.

Feature selection is a multi-objective problem. It has two main objectives, which are to maximize the classification accuracy (minimize the classification error rate) and minimize the number of features. These two goals are usually conflicting to each other, and the optimal decision needs to be made in the presence of a trade-off between them. Treating feature selection as a multi-objective problem can obtain a set of non-dominated feature subsets to meet different requirements in real-world applications. However, there are rare studies treating feature selection as a multi-objective problem [13, 14].

Two key factors in a feature selection algorithm are the search strategy and the evaluation criterion. The search space of a feature selection problem has 2n possible points/solutions, where n is the number of available features. The algorithm explores the search space of different feature combinations to find the best feature subset. However, the size of the search space is huge, especially when the number of features is large. This is one of the main reasons making feature selection a challenging task.

II. FEATURE SELECTION APPROACHES

Existing feature selection methods can be broadly classified into two categories: filter approaches and wrapper approaches. Wrapper methods include a classification algorithm as a part of the evaluation function to determine the goodness of the selected feature subsets. Filter methods use statistical characteristics of the data for evaluation, and the feature selection search process is independent of any classification algorithm. Filter methods are computationally less expensive and more general than wrapper procedures while wrappers are better than filters in terms of the classification performance [14].

Wrapper based Feature Selection

In a wrapper model, the feature selection algorithm exists as a wrapper around a classification algorithm and the classification algorithm is used as a “black box” by the feature selection algorithm [15]. The performance of the classification algorithm is employed in the evaluation function to evaluate the goodness of feature subsets and guide the search.

Filter based Feature Selection

In filter algorithms, the search process is independent of any classification algorithm. The goodness of feature subsets are evaluated based on a particular criterion like distance measure, information measure and consistency measure [14]. Filter algorithms are argued to be computationally less expensive and more general than wrapper algorithms [15, 16], but filter algorithms totally ignore the performance of the selected feature subset on the classification algorithm, which usually leads to lower performance than wrapper algorithms on a particular classification algorithm [15]. Compared with filter algorithms, wrappers often produce better classification performance because of the interaction between the classification algorithm and the selected feature subsets during the feature selection process [17]. However, wrapper feature selection algorithms are usually computationally more expensive than filters because each evaluation of a candidate solution needs a learning/classification algorithm to be trained and tested [16].

III. PARTICLE SWARM OPTIMIZATION (PSO)

PSO is an evolutionary computation technique proposed by Kennedy and Eberhart in 1995 [18, 19]. In PSO, a population called a swarm, of candidate solutions, are encoded as particles in the search space. PSO starts with the random initialization of a population of particles. The whole swarm move in the search space to find the best solution by updating the position of each particle based on the experience of its own and its neighboring particles [18, 19]. During movement, the current position of particle I is represented by a vector *xi = (xi1, xi2,….,xiD)*, where *D* is the dimensionality of the search space. The velocity of particle i is represented as *vi = (vi1, vi2, ...,viD)*, which is limited by a predefined maximum velocity, *vmax and vtid [−vmax, vmax]*. The best previous position of a particle is recorded as the personal best *pbest* and the best position obtained by the population thus far is called *gbest*. Based on *pbest and gbest*, PSO searches for the optimal solution by updating the velocity and the position of each particle according to the following equations:

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (1)$$

$$v_{id}^{t+1} = w * v_{id}^{t} + c1 * r1i * (p_{id} - x_{id}^{t}) + c2 * r2i * (p_{gd} - x_{id}^{t}) \quad (2)$$

wheret denotes the tth iteration, *d* denotes the d^{th}dimension in the search space *D*, w is inertia weight. *c1 and c2*are acceleration constants. r_1i *and* $r_2i$are random values uniformly distributed in [0, 1]. *pid and pgd*represent the elements of *pbest and gbest* in the d^{th} dimension [18].

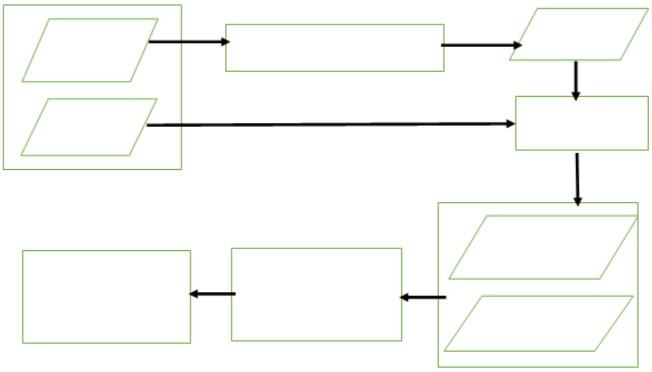

Figure 2: Structure of PSO based feature selection method

From the figure 5.2, the algorithm firstly runs on the training set of the dataset to select a subset of relevant features, which is the evolutionary training process. Then the training set and the test set are transformed to a new training set and a new test set by removing the features that are not selected. A classification algorithm is trained (learns) on the transformed training set. The learnt classifier is then applied to the transformed test set to obtain the final testing classification performance [18].

Particle Representation:

In PSO for feature selection, the representation of a particle is a n-bit string, where n is the total number of features in the dataset. The position value in the dth dimension (i.e. xid) is in [0,1], which shows the probability of the dth feature being selected. A threshold θ is used to determine whether a feature is selected or not. If xid>θ, the dth feature is selected. Otherwise, the dth feature is not selected.

Training Process:

The training process of a PSO based wrapper feature selection algorithm is shown in Figure 2. The key step is the goodness/fitness evaluation procedure. The position of a particle represents a selected feature subset. By removing the features that are not selected, the training set is transformed to a new training set.

The classification performance of the selected features is evaluated on the transformed training set. Based on the classification performance, the fitness of the particle is then calculated according to the predefined fitness function. After evaluating the fitness of all particles, the algorithm updates the pbest and gbest, and then updates the velocity and position of each particle. The algorithm stops when a predefined stopping criteria, that is the maximum number of iterations or an optimal fitness value, has been met. During the training process, Equation 5.5, which aims to minimize the classification error rate, is used as the fitness function to evaluate the goodness of particle i, where the position xi represents a feature subset [12].

$$fitness(x_i) = ErrorRate \quad (3)$$

Where the Error rate is determined by

$$ErrorRate = (FP + FN)/(FP + FN + TP + TN) \quad (4)$$

Where TP, TN, FP and FN stand for true positives, true negatives, false positives and false negatives, respectively.

The adaptive functional values were data based on the particle features representing the feature dimension; this data was classified by a k-Nearest Neighbor (k-NN) to obtain classification accuracy; the k-NN serves as an evaluator of the PSO fitness function. For example, when an 8-dimensional data set (n=8) *Sn = {F1, F2, F3, F4, F5, F6, F7, F8}* is analyzed using particle swarm optimization to select features smaller than n.

The following pseudo code shows the basic PSO Feature Selection process [18-20].

```
Basic PSO algorithm for Feature Selection
Input   : Training Data set and a Test Data set;
Output : Selected feature subset
Begin
    randomly initialize the position and velocity of each particle;
    while Maximumiterations is not reached do
        evaluate fitness of each particle ; /* according to (1) or (2) */
          for i=1 to PopulationSize do
                update the pbest of particle i;
                update the gbest of particle i;
          for i=1 to PopulationSize do
              for d=1 to Dimensionality do
                      update the velocity of particle i according to (5.3);
                      update the position of particle i according to (5.4);
    calculate the classification accuracy of the selected feature subset on the test set;
return the position of gbest (the selected feature subset);
```

Table 1: Parameter setup for PSO based Feature Selection:

Parameters	Value
Number of Iterations	200
Population Size	150,32,340
Number of particles	4,55,14
C1, C2	2,2
θ	0.6

In table 1, population size is referred to number of instances in the dataset and number of particles are referred as the number of decision attributes in the dataset.

IV. RESULTS AND DISCUSSION

Table 2: Performance analysis of the PSO with k-NN classifier

Dataset	Number of decision attributes	Selected features	Accuracy (%)	
			Before FS	After FS
Iris	4	3	82.43	96.48
Lung cancer	55	13	80.34	93.71
Leaf	14	8	90.07	97.46

V. CONCLUSION

Building an efficient classification model for classification problems with different dimensionality and different sample size is important. The main tasks are the selection of the features and the selection of the classification method. In this paper, PSO based feature selection to perform feature selection and then evaluated fitness values with a k-NN. Experimental results show that the method simplified feature selection and the total number of parameters needed effectively, thereby obtaining a higher classification accuracy.

REFERENCES

[1] S. J. Russell and P. Norvig, “Artificial Intelligence: A Modern Approach”, Second Edition, Pearson Education, 2003.
[2] H. M. Zhao, A. P. Sinha, and W. Ge, “Effects of feature construction on classification performance: An empirical study in bank failure prediction”, Expert Systems with Applications, Vol. 36, No. 2, pp. 2633–2644, 2009.
[3] I. A. Gheyas and L. S. Smith, “Feature subset selection in large dimensionality domains”, Pattern Recognition, Vol. 43, No. 1, pp. 5–13, 2010.
[4] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” Artificial Intelligence, Vol. 97, pp. 273–324, 1997.
[5] H. Liu and H. Motoda, “Feature Extraction, Construction and Selection: A Data Mining Perspective”, Norwell, MA, Kluwer Academic Publishers, USA, 1998.
[6] H. Liu and H. Motoda, “Feature Selection for Knowledge Discovery and Data Mining”, Norwell, Kluwer Academic Publishers, USA, 1998.
[7] L. Oliveira, R. Sabourin, F. Bortolozzi, and C. Suen, “Feature selection using multi-objective genetic algorithms for handwritten digit recognition,” in 16th International Conference on Pattern Recognition (ICPR’02), Vol. 1, pp. 568– 571, 2002.
[8] C. S. Yang, L. Y. Chuang, C. H. Ke, and C. H. Yang, “Boolean binary particle swarm optimization for feature selection” in IEEE Congress on Evolutionary Computation (CEC’08), pp. 2093–2098, 2008.
[9] E. Amaldi and V. Kann, “On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems” Theoretical Computer Science, Vol. 209, pp. 237–260, 1998.
[10] M. Mitchell, “An Introduction to Genetic Algorithms”, The MIT Press, 1996.
[11] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection”, The Journal of Machine Learning Research, Vol. 3, pp. 1157–1182, 2003.
[12] K. Waqas, R. Baig, and S. Ali, “Feature subset selection using multiobjective genetic algorithms” in IEEE 13th International Conference on Multitopic Conference (INMIC’09), pp. 1–6, 2009.
[13] L. Ke, Z. Feng, Z. Xu, K. Shang, and Y. Wang, “A multiobjective ACO algorithm for rough feature selection” in Second Pacific-Asia Conference on Circuits, Communications and System (PACCS), Vol. 1, pp. 207–210, 2010.
[14] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant features and the subset selection problem” in Machine Learning: Proceedings of the Eleventh International Conference (ICCCS’11), Morgan Kaufmann Publishers, pp. 121–129, 1994.
[15] C. S. Yang, L. Y. Chuang, and J. C. Li, “Chaotic maps in binary particle swarm optimization for feature selection”, in IEEE Conference on Soft Computing in Industrial Applications (SMCIA ’08), pp. 107–112, 2008.
[16] A. P. Mart´ınez, P. Larra˜naga, and I. Inza, “Information theory and classification error in probabilistic classifiers”, In Discovery Science, pages 347–351, 2006.
[17] P. Engelbrecht, “Computational intelligence: an introduction”, Second edition, Wiley, 2007.
[18] S. Yang, L. Y. Chuang, C. H. Ke, and C. H. Yang, “Boolean binary particle swarm optimization for feature selection”, in IEEE Congress on Evolutionary Computation (CEC’08), pp. 2093–2098, 2008.
[19] Unler and A. Murat, “A discrete particle swarm optimization method for feature selection in binary classification problems”, European Journal of Operational Research, Vol. 206, No. 3, pp. 528–539, 2010.
[20] Chakraborty, “Feature subset selection by particle swarm optimization with fuzzy fitness function”, in Third International Conference on Intelligent System and Knowledge Engineering (ISKE’08), Vol. 1, pp. 1038–1042, 2008.