A META- HEURISTIC REGRESSION-BASED FEATURE SELECTION FOR PREDICTIVE ANALYTICS

Bharat Singh and O P Vyas*

Department of Information Technology, Indian Institute of Information Technology, Allahabad, India
Emails: bharatbbd1@gmail.com*, opvyas@iiita.ac.in

ABSTRACT

A high-dimensional feature selection having a very large number of features with an optimal feature subset is an NP-complete problem. Because conventional optimization techniques are unable to tackle large-scale feature selection problems, meta-heuristic algorithms are widely used. In this paper, we propose a particle swarm optimization technique while utilizing regression techniques for feature selection. We then use the selected features to classify the data. Classification accuracy is used as a criterion to evaluate classifier performance, and classification is accomplished through the use of k-nearest neighbour (KNN) and Bayesian techniques. Various high dimensional data sets are used to evaluate the usefulness of the proposed approach. Results show that our approach gives better results when compared with other conventional feature selection algorithms.

Keywords: Feature selection, High dimensional data, Particle swarm optimization component, Data mining, Classification

1 INTRODUCTION

The rise of advanced data gathering techniques in fields such as bioinformatics, sensor networks, and customer relationships has led to challenges in high dimensional data (Kriegel & Zimek, 2012). None of the large amounts of available data can be directly understood by analysers, researchers, or data scientists. Fortunately computational technologies, data mining, and machine learning algorithms are improving to keep up with this increase in data volume. For example, one problem found in the field of bioinformatics is high dimensional datasets, that is, data sets having a very large number of features or attributes (Kriegel, 2009; Ding, 2003). Gene microarray datasets are an example of this type of problem. For each tissue sample, a gene microarray captures gene expression levels for tens of thousands of gene probes. In practice, however, only a small handful of these genes are actually relevant to answering a specific underlying biological question. High dimensionality, i.e., a large numbers of features, is a major problem in data mining fields and consumes a large amount of computation time, affecting the quality of training datasets as well as classification models. Because of "the curse of dimensionality" (Verleysen, 2005), all significant techniques for predictive and descriptive analysis become insignificant with these data volumes (Houle, 2010).

In this paper, we address the problem of selecting an optimized number of features from a high dimensional data set. The process of feature selection can be described as a search in a state space. A number of approaches are possible. A heuristic search, for instance, considers unselected features for evaluation at each of a number of iteration steps. A random search, on the other hand, generates random subsets within the search space. Several bio-inspired and genetic algorithms use this approach. Each of these methods has its limitations. Here, we propose a new approach combining particle swarm optimization (PSO) with regression techniques to improve feature selection, which is measured by the performance of a classifier.

A problem in medical analysis illustrates the importance of feature selection. One typical medical dataset consists of patient observations, each containing *m* clinical characteristics (features). This *m*-dimensional dataset is a union of two disjoint sets. One represents a "positive" group associated with patients having a specific medical condition or disease. The other is a "negative" group that do not have that condition or disease. Medical diagnosis and prognosis have been shown to be improved by applying data classification and identifying significant features in such datasets in clinical settings (Hammer, 2006; Saastamoinen, 2006; Tsirogiannis, 2004). To improve medical diagnosis, data mining techniques can be used to identify a disease from its

symptoms and make the decisions necessary to diagnose a patient. Similarly, data mining techniques may also be used in forecasting the probable outcome of a disease. Data mining here, however, is useful only if the selected features effectively identify a disease or correctly forecast a disease outcome.

As shown in Figure 1, the purpose of the feature selection is to find relevant and important features in the original dataset that are more significant than previously recognized (original) data patterns. We perform feature selection to reduce the size of the dataset and improve the computational performance of analytical methods. Feature selection is also an exceptionally effective and valuable technique to improve classification accuracy by reducing the number of irrelevant and redundant features and identifying those that are most important. If a dataset has a large number of features, the dimensions of the working data will be large, and the dataset will contain noisy, irrelevant, and redundant data resulting in the degradation of the predictive rate of the classifiers' accuracy. Therefore, an efficient and vigorous feature selection method is sought that reduces noisy, irrelevant, and redundant data.

Conventional mathematic statistical and analytical methods are often not able to analyze the complex systems of biological medicine and other fields. For example, analyzing high dimensional data in biomedical fields can produce vagueness, ambiguity, partial truths, and approximation (Zimek, 2012). To overcome this problem, Particle Swarm Optimization (PSO)-based approaches have previously been used in the selection of an optimized number of features (Agrafiotis, 2002; Fan 2010; Elbedwehy, 2012). This meta-heuristic feature selection technique can be used to eliminate noise and irrelevant and redundant data (Agrafiotis, 2002; Song, 2004), yet this technique is both challenging and less productive for high dimensional data. Soft Computing, which uses estimation, may be an alternative means to solving these problems. Some machine learning techniques are also able to tackle these datasets.

In this paper we describe a better technique for selecting significant features from high dimensional, scientific datasets. The main focus of this work is:

1. To identify important features effectively and efficiently
2. To increase the classification accuracy of identified features
3. To deal with irrelevant and redundant features to obtain a good feature subset
4. To keep only those features that are obtained after a double filtration process
5. To evaluate the accuracy of our feature selection method by comparison with other common feature selection algorithms (Naive Bayes & K-Nearest Neighbour)

This paper proposes a Meta-Heuristic Regression Based Feature Selection approach for feature selection in high dimensional datasets. We used a regression model to establish the relationship between the number of features and classification accuracy by reducing the size of testing data and to verify whether a feature is selected. With the help of the regression model, this modified PSO approach can increase population diversity and improve global searching capability, thereby avoiding inaccurate convergence and growing population diversity in the PSO mechanism. In this paper, we have used the terms - features, dimensions, and attributes - interchangeably. We also use 'FS' as an abbreviation of feature selection.

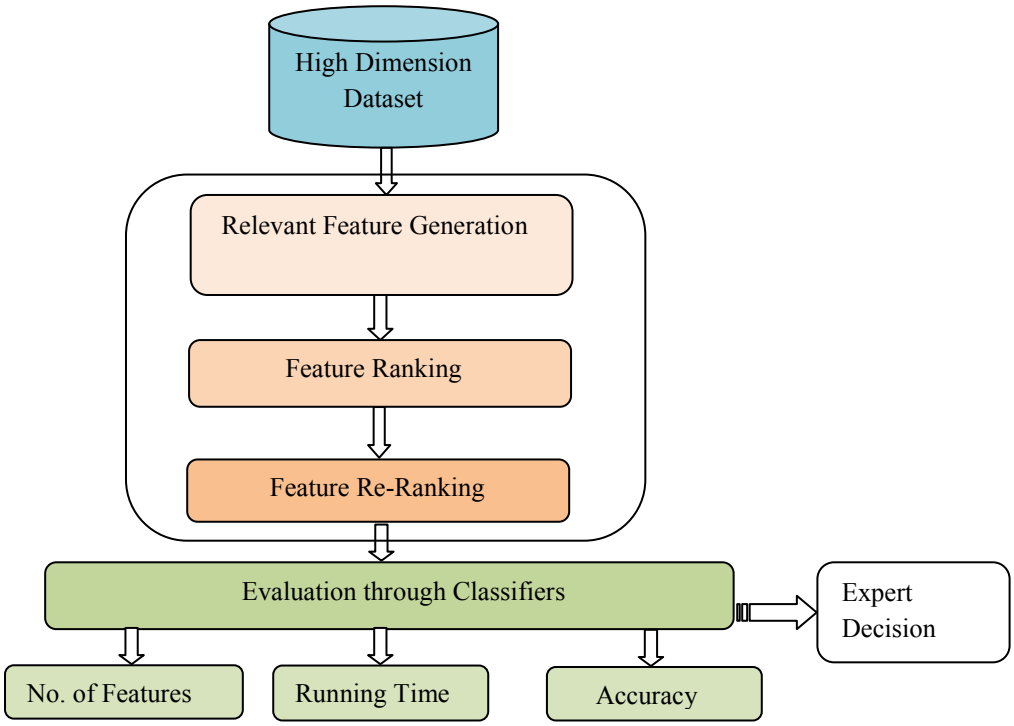

Figure 1. A framework for feature selection for high dimensional data

2 RELATED WORK

The process of feature selection is responsible for electing a subset of features that describe the important aspects of a dataset. Feature selection can be considered as a search into a state space. One can perform a full search in which all the space is traversed; however, this approach is impractical for a large number of features. A heuristic search considers the features, not yet selected at each iteration, for evaluation. A random search generates random subsets within the search space that can be evaluated for importance. Several bio-inspired and genetic algorithms use this approach (Nakamura, 2012).

Feature selection methods can be classified into two main categories: *filter* approaches (Song, 2013) and *wrapper* approaches (Song, 2013). In filter based techniques, a filtering process is performed before the classification process; therefore, the selected features are independent of the used classification algorithm (Xue, 2012). A ranking value or weight value is computed for each feature, and those features with higher ranking or weight values with respect to a user defined threshold value are selected to represent the original data set. On the other hand, wrapper approaches make use of a learning process to select a subset of features by adding and removing features that maximize learning accuracy. Wrapper methods are usually more effective than filter methods.

Kriegel (2009), Ding (2003), Agrafiotis (2002), and Elbedwehy (2012) have researched the feature selection problem. Genetic Algorithm (GA) and PSO are basic techniques that are meta-heuristic approaches (Bloomfield, 2010). Because PSO approaches converge more quickly and require less computational complexity, we have chosen to use PSO in our proposed feature selection approach.

Agrafiotis and Cedeno (2002) first applied PSO for feature selection. They devised structure-property and structure-activity correlation models for computer assisted drug design, a common technique in the pharmaceutical industry to correlate biological activity with compounds properties by identifying key features. Kennedy (2001) used the phenomenon of a neighbours' population influence as particle swarms move around a search space in which a population of individuals has settled in stochastically toward previously successful regions. This method was initially proposed for probing multidimensional continuous datasets and applied to the

feature selection by using the vector properties of the particles as probabilities. In their experimental analysis, the method compared favourably with simulated annealing techniques and identified an improved and more varied set of results, given the same amount of simulation time.

In the field of medicine, Melgani and Bazi (2008) used PSO in classifying ECG (electrocardiogram) beats and showed the advantage of the generalization capability with another classification algorithm, Support Vector Machine (SVM) approach. In this approach, a classifier was optimized by tuning its discriminate function upstream by looking for the best subset of features that feed the classifier. In particular, sensitivity has been tested using the SVM classifier by using three different base classifiers: k-Nearest Neighbour, RBF, and NN.

The Adaptive Michigan PSO (AMPSO) proposed by Cervantes (2009) used a number of different PSO versions. A single prototype in a swarm denotes each particle used in continuous classification problems. To overcome the risk of impulsive convergence, previous studies (Kennedy, 2001; Engelbrecht, 2007) have suggested changing traditional PSO operations to regroup swarms within a plausible subset of the original search space. Nearest prototype methods (Cervantes, 2009) achieved reasonable resuts with various pattern based classification approaches. In this method, a number of prototypes were found that represented the input samples accurately. In these approaches, the classifier assigns classes based on the nearest neighbour. AMPSO is different from a simple PSO because each particle in a swarm represents a single prototype in the solution. In AMPSO, each particle acts as a local classifier and thus cannot converge to a single solution. Therefore all swarm are considered for the solution. It was found by comparing the results with other classifier mehods that AMPSO gives competitive results in all the problems, particularly where the k-NN classifier does not perform effectively.

In other variations of PSO, Cervantes (2007, 2009), Fan (2010), Elbedwehy (2012), and Tasgetiren (2004) used binary particle swarm optimization methods to detect heart disease. Elbedwehy (2012) combined four techniques, binary particle swarm optimization, SVM, K-Nearest Neighbour, and a Leave-One-Out Cross-Validation approach, to produce a computer-aided diagnosis method for detecting heart valve disease. The algorithm was applied to a heart dataset consisting of 198 heart sound signals. SVM selected the features with the most weight to classify heart signals as either healthy or indicating heart valve disease. In another application, Tsanas et al. (2010) predicted average Parkinson's disease symptoms. They selected an optimally reduced subset of the measures and produced a clinically useful model in which each measure in the extracted subset is a non-overlapping physiological characteristic of speech signals. Fan and Chaovalitwongse (2010) proposed a new optimization framework for improving feature selection in medical data classification. This framework sought to identify the optimal group of features showing strong divisive power between two classes. They concluded that this method can be used as a quick decision-making tool in real clinical settings.

In the next subsections we describe the simple PSO method and the classification methods that we have used in our approach. The proposed approach is defined in Section 3.

2.1 Simple Particle Swarm Optimization (SPSO)

The PSO algorithm uses a population (called a swarm) of individual solutions (called particles) to find the best swarm solution iteratively. An initial solution is proposed for each particle (location and velocity) and then tested to see if a better overall solution (for the swarm) can be found according to some criteria. In PSO, each particle flies in the search space with a velocity adjusted by its own and its companion's history. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called p_{id} (pbest). Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called p_{gd} (gbest). Each particle has an objective function value, which is decided by a fitness function:

$$v_{id}^{t} = w \times v_{id}^{t-1} + c_1 \times r_1 \, (p_{id}^{t} - x_{id}^{t}) + c_2 \times r_2 \, (p_{gd}^{t} - x_{id}^{t}),$$

where i represents the i^{th} particle and d is the dimension of the solution space, c_1 denotes the cognitive learning factor, and c_2 indicates the social learning factor, r_1 and r_2 are the uniformly distributed random numbers in [0,1], P_{id} and P_{gd} stand for the position with the best fitness found so for the ith particle and best position in the neighbourhood, $v_{id}(t)$ and $v_{id}(t-1)$ are the velocities at time t and time t-1, respectively, and x_{id} is the position of the i^{th} particle at time t. Each particle then moves to a new potential solution depend on the following equation:

$$x_{id}^{t-1} = x_{id}^{t} + v_{id}^{t},\ d = 1,2,...,D,$$

(Kennedy, 1997) proposed a binary PSO in which a particle moves in a state space restricted to 0 and 1 in each dimension, in terms of the changes in probabilities that a particle (bit) will be in one state or the other:

$$x_{id} = \begin{cases} 1, & rand() < S(v_{i,d}) \\ 0 \end{cases},$$

$$S(v) = \frac{1}{1+e^{-v}}.$$

When applying PSO to the problem of feature selection, we use a binary digit to represent a feature. The bit values 0 and 1 represent non-selected and selected features, respectively. Each particle is coded to a binary alphabetic string. The PSO for the problem of feature selection in this study is called simple PSO (SPSO) (Wang, 2007). For example, the particle 101000 with six features means that the first and third features are selected. The function S(v) is a sigmoid limiting transformation and rand( ) is a random number selected from a uniform distribution in [0.0, 1.0].

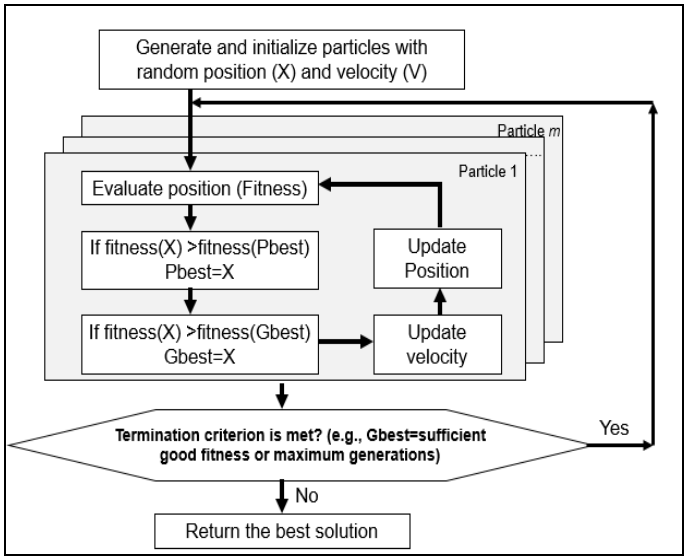

Figure 2. Flow diagram of SPSO (Simple Particle Swarm Optimization) for feature selection

2.2 k-NN Classifier

The (k-NN) technique was described by Fix and Hodges (Fix, 1951). It is a fundamental technique in data mining and machine learning and has been applied in many domains. This method classifies new cases based on similarity measures (ex., distance functions). The output is a class membership showing an inclination toward

one class over the others. A majority vote of its neighbours decides the class. K, the number of nearest neighbors that need to be considered, is a positive number that can be assigned by the user or automatically by the program.

If the class of test data matches the expected class of the pattern, we assume that it will be counted as a correctly predicted example. The fitness function is defined as the accuracy of classification, where accuracy is defined as the number of corrected predicted example divided by the total number of examples.

2.3 Naive Bayesian Classifier

A naive Bayes classifier is a simple probabilistic classifier technique based on the Bayes theorem and is especially well-matched when the input data are highly dimensional. The naive Bayes classifier method considers that the value of a particular attribute is distinct to the presence or absence of any other attributes, given the class attributes. In spite of its simple approach, the Naive Bayes approach many times outperforms more complicated classification methods (Langley, 1992). Naive Bayes classifiers make significant use of the assumption that all input features are conditionally independent, i.e., assuming that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class label. Only a small amount of training data is required to correctly classify through Naive Bayes. However, the hypothesis of conditional independence is not applicable in various real-world problems where relationships are present between the input features.

3 PROPOSED APPROACH AND ALGORITHM

A new algorithm, the Meta-Heuristic Regression Based Feature Selection (MHRFS), is proposed to investigate and improve the performance of PSO for feature selection. An overview of a PSO based feature selection algorithm has been given above. The basic PSO based algorithm (SPSO) is described as the baseline to test the performance of the newly proposed algorithm. A new fitness function, new initialisation strategies, and new pbest and gbest updating mechanisms are then proposed to improve the performance of PSO for feature selection. The terms pbest and gbest are defined in Section 2.1. The framework of the training and testing process of a PSO based feature selection technique is shown in Figure 2.

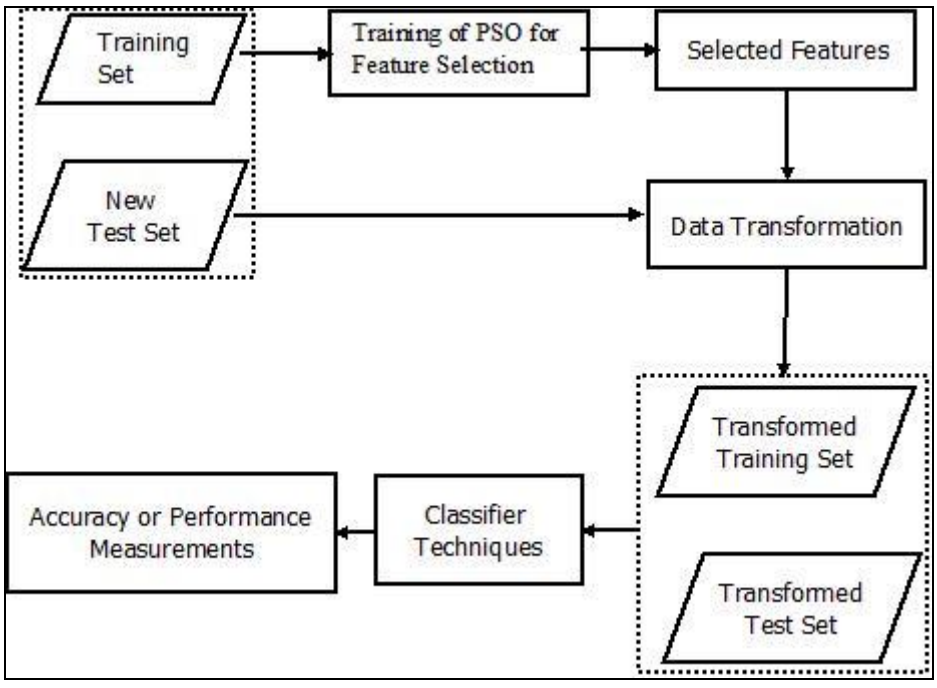

Figure 2. The Framework of PSO based feature selection methods

The algorithm begins with choosing a training set from the full dataset that is analyzed resulting in a subset of relevant features. These results are then tested using a new test set, which is also a subset of the entire dataset. Then the training data and the test data are converted to a reduced new training set and a new test set by eliminating the features that have not been selected. A classification algorithm is trained (learned) from the converted training data. The trained learning algorithm is then applied to the converted test data to obtain the final testing classification performance.

The proposed method for feature selection makes extensive use of a regression model to select a subset of features. The mathematical model of the proposed method is based on a simple concept derived from the PSO algorithm that utilizes each and every particle to search out local space and find the mutual understanding of each particle. The flow diagram of the simple particle swarm optimization (SPSO) method is depicted in Figure 2. The concept can be described as follows:

The classification accuracy y_i is used as a dependent variable while the binary variables x_{id} are treated as independent variables. Therefore, the regression model can be defined as:

$$y_i = \sigma + \lambda_1 x_{i1} + \lambda_2 x_{i2} + \ldots\ldots\ldots\ldots + \lambda_d x_{id} + \varepsilon_i, \text{ for } i=1, 2,\ldots.N$$

where σ is the intercept, and the λs are regression coefficients. The assumption is that if a feature's contributions to the accuracy are positive, then the value of λ_i should be positive. Some of the features have a positive value, $\lambda_i > 0$ and $x_{ii} = 0$, and increase the accuracy but fail to be selected by the simple PSO. Such types of features must be in the selected list of features to check if such subsets can increase the accuracy rate. On the other hand, if the PSO selects some features that have negative values, i.e., $\lambda_i < 0$ and $x_{ii} = 1$, this can reduce the accuracy. Thus these types of features should be eliminated from the selected list. The proposed approach works with the help of a regression method, which gives more accurate results. The MHRFS method is described as follows:

MHRFS Algorithm:

1. Calculate accuracy y_i for each particle, i = 1,...,N.
2. Find the coefficient λ_j of each feature by meta-heuristic (PSO) model.
3. Let $X_i^{new} = x_i$.
4. Set j=1.
5. If $\lambda_j > 0$ & $x_{ij} = 0$, then $x_{ij}^{new} = 1$;
if $\lambda_j > 0$ & $x_{ij} = 1$, then $x_{ij}^{new} = 0$.
6. If the accuracy value Y_i is less than Y_i^{new}, then $x_{ij} = x_{ij}^{new}$
and the fitness value Yi = accuracy value $Y_i^{new.}$
7. j = j + 1.
8. If j<D, go to step 2; otherwise, stop.

4 EXPERIMENTAL RESULTS

To investigate the effectiveness of the proposed approach, we used seven data sets. Classification accuracy is used as the evaluation criterion with the first nearest neighbor used to measure the accuracy. In addition, 10-fold cross validation and random sampling were utilized.

4.1 Data Sets

The seven data sets from the UCI repository (Bache, 2013) have sizes ranging from hundreds to thousands of data items and are described in Table 1. The seven data sets cover a wide vareity of measurements and have been the subject of extensive studies for high dimensional systems, serving as a test bed for many PSO-based feature selection algorithms (Azevedo, 2007; Marinakis, 2008;Yang, 2008). To allow comparison with previous PSO based approaches, a number of input features were taken from the literature (summarized in Table 1).

In the experiments, all of the data in each data set were randomly divided into two sets: 70% as the training set and 30% as the test set. During the training process, each particle (individual) represented one feature subset. The classification performance of a selected feature subset was evaluated by 10-fold cross-validation on the training set. Note that 10-fold cross-validation was performed as an inner loop in the training process to evaluate the classification performance of a single feature subset on the training set and it did not generate ten feature subsets. After the training process, the selected features were evaluated on the test set to obtain the testing classification error rate.

Table 1. Data sets and their characteristics

Datasets	**Summarization of data sets characteristics (Bache, 2013)**		
	# of instances	***# of dimensions/ features***	***# of classes***
Sonar	990	60	11
Ionosphere	351	34	2
Wine	178	13	3
Spect. Heart	267	23	2
Heart	303	75	4
Madelon	4400	500	2
Colon	62	2001	2

All of the algorithms were wrapper approaches, i.e., required a classification algorithm in the training process to evaluate the classification performance of the selected feature subset. Any classification algorithm can be used here, such as Naive Bayes, Decision Tree, and Support Vector Machine. One of the simplest and most commonly used classification algorithms, K-nearest neighbour (KNN), a Bayesian classifier, (Langley, 1992; Chuang, 2011), was used in the experiments. We defined K=5 in the classifier to simplify the evaluation process, and implemented the process in the Java machine learning library (Java-ML) (Abeel, 2009).

The proposed MHRFS based on the PSO algorithm presented for the feature selection problem was implemented in C and run on an Intel i7 2.6 GHz, 4GB Ram Machine. Evaluation of the MHRFS was assessed by a conventional genetic algorithm (GA). For this purpose, a GA was implemented in C and testing was done on randomly distributed data. The GA was a conventional one with a uniform crossover, simple inversion mutation, and a tournament selection of size 2. In the experimental analysis, we defined some parameters for the conventional GA and the proposed approach. In the proposed MHRFS, the size of the population in the swarm was taken to be the twice the number of whole features. Parameters c1 and c2, the social and cognitive parameters respectively, are kept at 2. Here c1 and c2 are learning factors. For the conventional GA and proposed MHRFS, the size of the population was kept the same. The crossover and mutation rates were 0.70 and 0.10, respectively. On average, the GA and PSO techniques were executed for 1-50 iterations. Table 2 gives the accuracy rate for different iterations of the data sets presented in Table 1.

4.2 Comments on Selected Features

Using our proposed algorithm, we achieve a high classification rate for the combination of a small number of features. However, while with any increment in the subset of features, the results show consistent classification accuracy, the time consumption increases rapidly. In some cases, as more features are included, the classification rate tends to slow down. For example, the Sonar dataset described in Table 1 produced an

accuracy rate of 59.16% in 18 iterations whereas increasing the number of iterations to 32 gave a 61.23% accuracy rate. Proposed algorithm runs iteratively for selecting feature subsets. Although each time some of the features may be common, distinct features have been selected by the proposed algorithm. It should be noted that the governing features can be estimated within the feature subset by calculating optimal number of iterations that also have high classification accuracy. For calculating optimal number of iterations we select feature subsets through our algorithm until it produces constant accuracy. As the number of iterations increases, the classification rate becomes stable while showing stable accuracy for smaller data sets from the beginning. Some of the high dimensional dataset characteristics are summarised in Table 1. On average, our proposed approach performed well in selecting representative features, as described in Table 2, for the data sets mention in Table 1.

Table 2. The number of selected features by the proposed MHRFS method, including the classification rate for the original data set (applied before feature selection). The Classification accuracy with the feature selection technique is presented in Table 3.

Data sets	**With feature selection**		**Without feature selection**	
	# original features	**# features selected**	**accuracy using k-NN (%)**	**accuracy using NB (%)**
Sonar	60	19	86.53	67.78
Ionosphere	34	10	86.32	82.62
Wine	13	5	94.94	96.62
Spect. Heart	23	8	62.03	68.98
Heart	75	23	76.23	83.49
Madelon	500	69	54.26	59.53
Colon	2001	87	77.42	53.23

4.3 Evaluation based of classification accuracy

Using feature selection and constraint optimization should increase the classification rate performance as well as decrease the response time. The proposed algorithm selects very few important features and support vectors and should reduce size and time of execution and also improve classification accuracy. Table 2 lists the prediction accuracy rate for different iterations in a 10-fold cross validation. We compared our algorithm (MHRFS) to other well known feature selection algorithms: Simple PSO (SPSO, Wang, 2007) , Regression Based PSO (RBPSO, Chen, 2013), and Backward Regression Based PSO (BRPSO, Chen, 2013). We have checked these algorithms for different iterations. Our algorithm outperforms the others some of the time, but it is a bit difficult to say that one method achieves better accuracy for all the datasets and with every iteration. Two classifiers (Bayesian and k-NN) were used to measure the accuracy of the feature selection method. We found that our algorithm gives good results with both classifiers.

Table 3. Classification accuracy comparison between different feature selection based on PSO approaches

	Bayesian Classifier					**KNN Classifier**				
Datasets	**Iteration**	**SPSO**	**MHRFS**	**RBPSO**	**BRBPSO**	**Iteration**	**SPSO**	**MHRFS**	**RBPSO**	**BRBPSO**
Sonar	1	44.26%	**45.90%**	43.96%	45.10%	1	90.00%	90.33%	**93.00%**	91.67%
	6	50.81%	54.09%	53.56%	**56.04%**	12	91.00%	91.00%	**93.67%**	92.33%
	12	50.81%	55.73%	**56.31%**	55.45%	24	94.00%	95.00%	**95.67%**	95.00%
	18	55.73%	57.16%	57.24%	**59.10%**	36	95.00%	**95.84%**	95.67%	95.00%
	24	55.73%	**60.01%**	57.93%	59.90%	48	95.00%	**95.89%**	95.67%	95.00%
	32	55.96%	**61.23%**	58.14%	59.90%	60	95.00%	**95.95%**	95.67%	95.00%
	(AVG)	52.22%	**56.02%**	54.36%	55.58%	(AVG)	94.16%	94.72%	**95.67%**	94.16%
Ionosphere	1	**60.90%**	59.95%	60.10%	60.10%	1	92.30%	93.19%	92.30%	**95.19%**
	4	61.90%	60.95%	61.90%	**61.91%**	3	94.15%	95.19%	92.30%	**95.19%**
	8	61.90%	60.94%	61.80%	**61.92%**	6	94.15%	95.19%	94.23%	**95.26%**
	12	61.90%	**61.95%**	61.80%	61.90%	9	94.15%	**95.29%**	95.01%	95.26%
	16	61.90%	61.54%	**62.10%**	61.90%	12	95.15%	**95.29%**	95.01%	**95.26%**
	24	61.90%	**62.90%**	62.10%	61.90%	48	95.15%	96.03%	**96.15%**	95.26%
	(AVG)	61.90%	**61.92%**	61.90%	61.91%	(AVG)	94.02%	**95.19%**	94.22%	94.17%
Wine	1	34.18%	**35.18%**	**35.18%**	35.00%	1	94.33%	96.22%	96.22%	**98.11%**
	4	34.18%	37.25%	35.18%	**37.28%**	4	96.22%	**98.11%**	96.22%	**98.11%**
	8	33.68%	37.08%	35.03%	**37.30%**	8	96.22%	**98.11%**	98.11%	**98.11%**
	12	33.68%	**37.37%**	34.93%	37.30%	12	96.22%	**98.11%**	98.11%	**98.11%**
	16	33.68%	**37.37%**	34.93%	37.30%	16	96.22%	**98.11%**	**98.11%**	**98.11%**
	(AVG)	33.38%	**36.73%**	35.05%	36.35%	(AVG)	96.22%	**98.02%**	97.35%	**98.11%**
Spect Heart	1	71.60%	70.83%	**71.63%**	**71.86%**	1	76.25%	**81.25%**	75.00%	76.25%
	7	72.83%	**74.07%**	72.98%	73.08%	6	78.75%	**81.25%**	76.25%	77.50%
	14	75.33%	76.54%	76.02%	**76.82%**	12	78.75%	**81.25%**	76.25%	77.50%
	21	75.33%	**77.77%**	76.65%	76.75%	18	78.75%	**81.25%**	76.25%	78.75%
	28	75.33%	**77.77%**	76.65%	76.75%	24	78.75%	**81.25%**	76.25%	80.00%
	35	76.77%	77.77%	**77.96%**	**77.96%**	30	78.75%	**82.50%**	76.25%	80.00%
	(AVG)	74.70%	**76.13%**	75.20%	75.44%	(AVG)	78.25%	**81.49%**	76.00%	78.33%
Heart	1	**40.74%**	40.24%	**40.74%**	**40.74%**	1	82.50%	82.50%	**85.00%**	**85.00%**
	8	40.74%	41.24%	**40.92%**	**40.92%**	2	82.50%	83.75%	**85.00%**	**85.00%**

	16	40.74%	**41.24%**	40.92%	40.92%	4	82.50%	**85.00%**	**85.00%**	**85.00%**
	24	40.74%	**41.24%**	40.92%	40.92%	8	82.50%	**85.80%**	85.00%	85.00%
	32	40.74%	**41.24%**	40.92%	40.92%	41	83.75%	**85.87%**	85.00%	85.00%
	(AVG)	40.74%	**41.24%**	40.74%	40.92%	(AVG)	82.75%	**84.25%**	**85.00%**	**85.00%**
Madelon	1	49.22%	49.21%	**50.47**%	**50.47**%	1	48.22%	45.21%	**50.47**%	49.14%
	8	50.32%	55.26%	56.38%	**56.61**%	6	48.32%	47.20%	**50.47**%	49.14%
	16	50.32%	57.73%	56.38%	**57.83**%	11	48.66%	49.77%	**50.47**%	49.61%
	26	50.32%	**58.14**%	56.38%	57.83%	18	48.72%	**51.26**%	50.47%	50.83%
	32	50.32%	**58.14**%	56.38%	57.83%	40	48.72%	**51.26**%	50.47%	50.83%
	(AVG)	50.10%	55.70%	55.18%	**56.11**%	(AVG)	48.53%	48.94%	**50.47**%	49.91%
Colon	1	**48.62**%	46.17%	46.22%	46.36%	1	70.10%	69.23%	**70.87**%	68.31%
	8	48.62%	48.23%	**48.97**%	47.28%	6	70.10%	**72.16**%	70.90%	70.45%
	16	48.62%	**50.74**%	49.23%	50.10%	20	70.10%	72.67%	70.90%	**73.68**%
	24	48.62%	**51.56**%	49.23%	50.10%	35	70.10%	**74.88**%	70.90%	73.68%
	32	48.62%	**51.56**%	49.23%	50.10%	43	70.10%	**74.88**%	70.90%	73.68%
	(AVG)	48.62%	**49.65**%	48.57%	48.78%	(AVG)	70.10%	**72.76**%	70.89%	71.96%

5 CONCLUSION

Despite much research on the PSO-based feature selection in the field of machine learning, there is still a shortage of high quality analytical techniques for high dimensional datasets. It is unclear how to construct a better feature selection algorithm for a specific parameter setting and classifier. In this paper, we evaluated our MHRFS technique against other well known feature selection techniques. For evaluation, we used two classifiers: k-NN and Naive Bayes. For testing purposes, we used three microarray and three non-biological but high dimensional datasets. We found that optimization of our feature selection algorithm sometimes increases the accuracy of the prediction in a comparatively reduced time span and shows good accuracy in most cases. The proposed approach could be used as a pre-processing tool to facilitate the optimization of feature selection methods as it can be used to increase classification accuracy.

6 REFERENCES

Abeel, T., de Peer, Y. V., & Saeys, Y. (2009) Java-ml: A machine learning library. *Journal Machine Learning Research*, pp 931-934.

Agrafiotis, D.K. & Cedeno, W. (2002) Feature selection for structure-activity correlation use binary particle swarms. *Journal of Medicinal Chemistry*, pp 1098-1107.

Azevedo, G., Cavalcanti, G., & Filho, E. (2007) An approach to feature selection for keystroke dynamics systems based on PSO and feature weighting. *IEEE Congress on Evolutionary Computation (CEC'07)*, pp 3577-3584.

Bache, K. & Lichman, M. (2013) UCI Machine Learnig Repository, University of California, School of Information and Computer Science, Irvine, CA. Retrieved from the World Wide Web October 27, 2014: http://archive.ics.uci.edu/ml

Bloolmfield, M.W., Herencia, J.E., & Weaver, P.M. (2010) Analysis and benchmarking of meta-heuristic techniques for lay-up optimization. *Computers and Structures*, pp 272-282.

Cervantes, A., Garia, I.M., & Isasi, P. (2009) AMPSP: a new particle swarm method for nearest neighbourhood classification. *IEEE Transactions on Systems Man and Cybernetics Part B*, pp 1082-1091.

Chen, K.H., Chen L.F., & Su, C.T. (2013) A new particle swarm feature selection method for classification. *J. Intell. Inf. Syst.*, pp 507-530.

Chuang, L. Y., Tsai, S. W. & Yang, C . H. (2011) Improved binary particle swarm optimization using catfish effect for feature selection. *Expert Syst. Appl.*, pp 12699-12707.

Ding, C. & Peng, H. (2003) Minimum redundancy feature selection from microarray gene expression data. *IEEE Computer Society Conference on Bioinformatics,* pp 523-528.

Elbedwehy, M.N. (2012) Detection of Heart Disease using Binary Particle Swarm Optimization. *Mansoura University Egypt, Computational Biology and Chemistry*, pp 29-38.

Engelbrecht, A. P. (2007) *Computational Intelligence: an Introduction* (2nd ed.). Wiley.

Fan, Y.J. & Chaovalitwongse, W.A. (2010) Optimizing feature selection to improve medical diagnosis. *Annals of Operations Research*, pp 169-183.

Fix, E. & Hodges, J.L. (1951) *Discriminatory analysis, nonparametric discrimination: Consistency properties.* Technical Report 4, Randolph Field, Texas. USAF School of Aviation Medicine.

Hammer, P.L. & Bonates T.O. (2006) Logical analysis of data—An overview: From combinatorial optimization to medical applications. *Annals of Operations Research*, pp 203-225.

Houle, M.E., Kriegel, H-P., Schubert, E., & Zimek, A. (2010) Can Shared-Neighbor Distances Defeat the Curse of Dimensionality? *21st Internation Conference on Scientific and Statistical Database Management SSDBM,* pp 482-500.

Kennedy J., & Eberhart R. (1997) A discrete binary version of the particle swarm algorithm. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Orlando, USA, pp 4104-4108.

Kennedy, J., Eberhart, R.C., & She, Y. (2001) *Swarm intelligence. Evalutionary Computation Series,* San Diego: Morgan Kaufman.

Kriegel, H-P., Krozer, P., & Zimek, A. (2009) Clustering High-Dimensional data: A survey on subspace clustering, pattern-based clustering and correlation clustering. *TKDD.*

Langley, P., Iba, W., & Thompson, K. (1992) An analysis of bayesian classifiers. In *Proceedings of the 10th National Conference on Artificial Intelligence*, MIT Press, pp 223-228.

Marinakis, Y., Marinaki, M., & Dounias G. (2008) Particle swarm optimization for pap-smear diagnosis. *Expert Systems with Applications*, pp 1645-1656.

Melgani, F. & Bazi, Y. (2008) Classification of electrocardiogram signals with support vector machine and particle swarm opt. *IEEE Transactions on Information Technology in Biomedicine*, pp 667-677.

Nakamura, R., Pereira, L., Costa, K., Rodrigues, D., & Papa, J. (2012) BBA: a binary bat algorithm for feature selection. *Conference on Graphics, Patterns and Image,* Ouro Preto, pp 22-25.