Journal on Big Data
DOI:10.32604/jbd.2021.010364
Article

A New Population Initialization of Particle Swarm Optimization Method Based on PCA for Feature Selection

Shichao Wang, Yu Xue* and Weiwei Jia

School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210000, China
*Corresponding Author: Yu Xue. Email: xueyu@nuist.edu.cn
Received: 15 April 2020; Accepted: 20 August 2020

Abstract: In many fields such as signal processing, machine learning, pattern recognition and data mining, it is common practice to process datasets containing huge numbers of features. In such cases, Feature Selection (FS) is often involved. Meanwhile, owing to their excellent global search ability, evolutionary computation techniques have been widely employed to the FS. So, as a powerful global search method and calculation fast than other EC algorithms, PSO can solve features selection problems well. However, when facing a large number of feature selection, the efficiency of PSO drops significantly. Therefore, plenty of works have been done to improve this situation. Besides, many studies have shown that an appropriate population initialization can effectively help to improve this problem. So, basing on PSO, this paper introduces a new feature selection method with filter-based population. The proposed algorithm uses Principal Component Analysis (PCA) to measure the importance of features first, then based on the sorted feature information, a population initialization method using the threshold selection and the mixed initialization is proposed. The experiments were performed on several datasets and compared to several other related algorithms. Experimental results show that the accuracy of PSO to solve feature selection problems is significantly improved after using proposed method.

Keywords: Feature selection; population initialization; particle swarm optimization; principal component analysis

1 Introduction

Feature selection can be regarded as optimization problem to some extent. The key is to establish an evaluation criterion to distinguish an optimal feature subset which contributes to classification and find a subset of features that are redundant, partially or completely unrelated. Different evaluation functions may lead to different results [1]. According to the relationship between the evaluation function and the classifier, feature selection methods can actually be regarded as composed of filter and wrapper. Among them, filter uses an evaluation function independent of the classifier, and the wrapper takes the error probability of the classifier as the evaluation function. The final goal of feature acquisition is to minimize the error probability of the classifier, so the most intuitive way is to use the classifier error probability as the evaluation criterion, i.e., to select features or feature combinations that minimize the error probability of the classifier. Solberg et al. classify the multi-texture features of SAR images, and Sylvie unsupervised the classification of airborne multi-spectral and multi-frequency SAR data [2], all of which are selected by comparison of classification results. However, this method is too computationally intensive and has poor practicability. Even if the class conditional distribution density is known, the error probability of calculating the classifier is very complicated. In practice, the conditional distribution density is often unknown, which is more difficult to calculate. Therefore, feature selection based on the evaluation function is more common.

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Feature selection actually includes two aspects: feature extraction and feature selection [3]. Feature extraction refer to a transformation of data from high-dimensional space to low-dimensional space to achieve dimensional reduction [4]. Feature selection is to remove redundant or unrelated features from a set of features to reduce dimensionality. Both are often used in combination. Principle components analysis (PCA) can be said to belong to feature extraction. The main idea of PCA is to reduce dimension, transforming n-dimensional features into k-dimension with few features. These reduced features are linear combinations of original features and are not related to each other, which can reflect most of the information of the original data.

One of the great benefits of PCA technology is the dimensional reduction of data, which can detect and correct for population structure [5]. After sorting the importance of the new "principal" vector, we can take the most important part in the front, and discard the unimportant dimension, so as to realize dimension reduction to simplify the model or compress the data. Meanwhile, the information of the original data is maintained to the greatest extent.

Another advantage is that it has no parameters at all. In the PCA calculation process, no artificial parameter settings are required or the calculation is intervened according to any empirical model. The final result is only associated with the dimension and independent of the researcher.

Usually, a search algorithm and a feature evaluation criterion are involved in a feature subset selection algorithm. The potential solution space is explored in the search algorithm [6], while the evaluation criterion measures the ability of feature subsets to distinguish one category from others. Exhaustive search, heuristic search and random search are three typical categories of search techniques in the FS method. The advantage of exhaustive search is that it can obtain the best solution, but for most practical applications, it is not feasible because of the complexity of its computing time. The typical heuristic search methods are Greedy Stepwise Reverse Selection (GSBS) and Linear Forward Selection (LFS), which are improved on the basis of SGS and SFS. By limiting the number of features in each step, LFS improves the operation efficiency of SFS. Although the backward selection can consider feature interaction compared with positive selection, it is still unable to cope with datasets of hundreds of features. GSBS cannot be completed in a short time because it runs on datasets with a great number of features. Besides, the front and back strategies usually face local optimal problems.

Random searches may generate subsets in a completely random manner, using the Las Vegas algorithm, such as LVW, which converges too slowly in a large search space. Unlike random generation, Evolutionary Computation (EC) is a random method that applies evolutionary principles or group intelligence to generate better subsets from the current subset [7]. Based on the above discussion, EC seems to be the best choice. Moreover, as a method of EC, PSO is a swarm intelligence technology applied to FS and shows its effectiveness. So, as a powerful global search method and calculation fast than other EC algorithms, PSO has been widely employed to solve feature selection problems. However, when facing high-dimensional disaster, the efficiency of PSO drops significantly. At the same time, the filter methods can converge quickly although its low efficiency. What's more, a great deal of useful heuristic information can be attached from them. Therefore, this paper introduces a new way for population initialization considering applying PCA into the population initialization of PSO is proposed. The main idea is first to evaluate the features using PCA and choose the most important features to initialize the population. After that, in order to make these features more flexible to use, the threshold selection and the idea of the mixed initialization [8] are employed to select features randomly. This paper is structured as follows: a brief review of the PSO, PCA and initialization techniques of PSO is presented in Section II. In the next section III, the main characteristics of the proposed method are described. Moreover, Section IV introduces the process of experimental design. Finally, Section V analyzes the results of the experiment and summarizes the main conclusions of this work.

2 Background

2.1 Particle Swarm Optimization

PSO, deriving from the research of bird predation behavior, is a kind of evolutionary algorithm. It has the advantages of fast convergence speed, high accuracy and simple implementation. PSO algorithm firstly generates the particle swarm satisfying the feasible solution, and then each particle swarm independently searches for the possible optimal solution and shares its own information with other particle swarms. Assuming that there is a group of hungry birds out looking for food. In order to find the real object as soon as possible, they need to search separately. During this period, they also need to share their location in real time so that they do not stray. In this way, they can work together to find the target (i.e., the optimal solution) efficiently and quickly. Like an interconnected Internet, they can share their location in real time as they want and judge whether someone has found the optimal solution to the goal. Finally, the whole swarm gather around the target source, representing we have found the optimal solution, i.e., the problem convergence. Constant iteration, update the position and velocity of each particle following the formulas Eq. (1) and Eq. (2) and finally get the optimal solution that satisfies the termination condition.

$$x_{id}^{t+1} = x_{id}^{t} + u_{id}^{t+1} \tag{1}$$

$$u_{id}^{t+1} = w * u_{id}^{t} + c_1 * r_1 * \left(p_{id} - x_{id}^{t} \right) + c_2 * r2 * (p_{gd} - x_{id}^{t}) \tag{2}$$

where w is the inertia weight; t is the t^{th} iteration of the evolution process; i means the current particle. d represents the d^{th} dimension. x_{id}^{t} represents the d^{th} dimension of the i^{th} particle's position and v_{id}^{t} is the velocity of the i_{th}^{d} particle. c_1 and c_2 stand for acceleration constants. r_1 and r_2 stand for random numbers in [0, 1]. p_{id}, p_{gd} represents the d^{th} value of the personal best solution and all the global best solutions respectively.

2.2 Initialization Methods of PSO

In evolutionary algorithms, population initialization is a key since it can affect the quality of the final solution and also the convergence speed. In order to solve low dimensional problems, Reference [9] utilize differential evolution algorithm with opposite optimization for population initialization to better the performance of the proposed method. For solving the high dimensional problem better, Reference [10] validates a genetic algorithm in the initial population to improve the quasi-random sequence. Some methods were also proposed, Reference [11] mentioned a population initialization method based on clustering and Cauchy deviates, which provides a good starting position for the search, increasing convergence rate noticeable. To improve the performance of PSO, several initialization methods were designed to solve feature selection problems better. For example, Reference [12] adopt the PSO with chaotic opposition-base initialization and stochastic search technique to avoid likely local optima on complex multimodal problems. Reference [13] presents a new initialization method to generate the initial population through applying a space transformation search (STS) strategy.

So far, a lot of work have been already done to increase the efficiency of the evolutionary algorithms, however, there is only a little reported research done to improve population initialization in evolutionary algorithms. Since population initialization can improve the efficiency of evolutionary algorithms significantly, we should take time to study how to better improve population initialization. What's more, PSO has a natural advantage in feature selection. However, when high-dimensional feature selection problems are encountered, the efficiency of PSO will be greatly reduced. But for filters, which can handle high dimensions quickly, they can effectively select features to provide heuristic information. However, the filters are rarely used to solve feature selection in evolutionary algorithms for now. So, this paper aims to propose a new population initialization method combined with filter to increase the efficiency of the evolutionary algorithm when facing high-dimensional disaster problems.

3 The Proposed Method

3.1 PCA

PCA method was introduced by Karl and Pearson. Its core idea is to map high-dimensional data into a low-dimensional space that can accurately represent the original data. After dimensionality reduction, the data not only retains the variation information in the high-dimensional data, but also indicates whether it is optimal according to the degree of change of the data. Mainly relying on the position information of the sample in space, PCA assumes that the sample set has the largest variance along with certain directions, and then projects the sample onto the straight line where these directions are located, so as to eliminate correlation and noise between features. The solution to the above problem can be simplified to the eigenvalue or eigenvector problem with respect to the mode correlation matrix C. And the order of the main components is determined by the value of the corresponding eigenvalue. Supposing that a dataset represents S and the number of samples represents N, the whole pseudo code of PCA is shown as Algorithm 1.

Algorithm 1: Pseudo Code of PCA algorithm

Input: dataset (*S*), the number of samples (*M*), the Dimension of features (*N*);

Output: new dimensionality reduction sample *X*

1 Pre-process the data to normalize its mean and variance (1).
2 Calculate the covariance matrix of the data (2).
3 Find the eigenvalues of the covariance matrix and the corresponding eigenvectors (3).
4 Arrange the eigenvalues from large to small, and keep the top corresponding K eigenvectors.
5 Transform the data into the new space constructed by the above K eigenvectors (4).

In PCA, the mean of the data is zeroed first and then rescale the unit variance of each coordinate to ensure different attributes are processed on the same "scale." For instance, if $x1$ was room' maximum temperature in centigrade (taking values in the high tens) and $x2$ were the size of room in square meter (taking values around 30–40). After this normalization, the attributes are more comparable. According to Algorithm 1, the following Eq. (3) are used to normalize its mean and variance.

$$u = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}. \quad \text{Replace each } x^{(i)} \text{ with } x^{(i)} - u$$

$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)}\right) \text{ . Replace each } x^{(i)} \text{ with } x_j^{(i)} / \sigma_j \tag{3}$$

where $x^{(i)}$ represents the i^{th} sample. j represents the j^{th}dimension of the i^{th}sample. u represents the sample mean. σ^j represents the standard deviation of the j^{th} dimension.

As we all know, what PCA has to do is to find a new coordinate system to maximize the variance after projecting the original data. To formalize this, given that the unit vector u and a point x, and the length of the projection of x onto u is given by $x^T u$., i.e., if $x^{(i)}$ is a point in our dataset, then its projection onto u is distance $x^T u$ from the origin. Therefore, aiming to make the variance of the projections maximized, we choose a unit-length u so as to maximize:

$$\frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)^T} u\right)^2 = \frac{1}{m}\sum_{i=1}^{m} u^T x^{(i)} x^{(i)^T} = u^T\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)} x^{(i)^T}\right)u$$

$$\Rightarrow u\lambda = \lambda u = uu^T \sum u \tag{4}$$

where λ represents $\frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)}u\right)^2$ $\cdot\sum$ represents $\frac{1}{m}\sum_{i=1}^{m} x^{(i)}x^{(i)^T}$

It is easy to realize that when $\|u\|^2 = 1$, maximizing the main eigenvector of $\sum = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}x^{(i)^T}$ is only the empirical covariance matrix of the data. So, we can simplify equation above to the following Eq. (5).

$$\sum u = \lambda u \tag{5}$$

We got it! λ and u are the eigenvalue and the eigenvector of Σ. The best projection line is the feature vector corresponding to the maximum value of the feature value λ, followed by the second largest corresponding feature vector of λ, and so on.

After eigenvalue decomposition of covariance matrix, we extract the eigenvectors corresponding to the first k eigenvalues as the best new eigenvalues and the new features are orthogonal. After getting the first k u, the sample $x^{(i)}$ can get a new sample by the following Eq. (6).

$$y^{(i)} = \begin{bmatrix} u_1^T x^{(i)} \\ u_2^T x^{(i)} \\ \vdots \\ u_k^T x^{(i)} \end{bmatrix} \tag{6}$$

3.2 Initialization Method Combining the Filter and Threshold Selection

Filter can be used to screen for appropriate features, since it can effectively and quickly select valid features and apply heuristic information. Also, PCA can project the data into a new space and then select the main components in front, which retain most of the original data information. So, we can use PCA to extract the most important features to initialize the PSO. However, the key is to choose an appropriate number of features to better maximize the classification effect. To solve this problem, we use mixed initialization here. After sorting the "remolding" of features by the PCA and prioritizing them, we select a few of the most important features to initialize most of the particles. Meanwhile, a relatively great number of important features are used to initialize the remaining particles. In detail, we first employ PCA to make features sorted by importance from top to bottom and then select the top 80%. To further reduce the number of features, the threshold mechanism is added to further extract features from the selected 80% features. In PSO, the location update and velocity of a particle are both continuous values limited to [0, 1]. To convert this continuous value into a discrete value, the threshold mechanism is needed. More narrowly, when the continuous value is greater than or equal to θ, it will be set to 1, which means that this feature is appropriate. Conversely, if the continuous value is less than θ, then it will be set to 0, which implies that it has been discarded. According to the above discussion, we can get our initialization method–the filter and threshold selection based initialization method (FTSI). We make appropriate modifications to the threshold mechanism combined with the characteristics of the mixed initialization. First, after using PCA to measure the features, the top 80% of them are stored in VS. Then in terms of the features selected in VS, for most particles, if the value of the j^{th} dimension of i^{th} particle in $\mathrm{PF}_{ps\times D}$ is more than the threshold θ, the corresponding position of the matrix $\mathrm{PF}_{ps\times D}$ will be set to 1, meaning the corresponding feature is appropriate. If not, it will be set to 0 meaning that corresponding feature is abandoned. By contrast, for the remaining small amount of particles, the corresponding position of the matrix $\mathrm{PF}_{ps\times D}$ is set to 1 when the value of the j^{th} dimension of the i^{th} particle in $\mathrm{PF}_{ps\times D}$ is no more than the threshold θ, meaning that the corresponding feature is appropriate. Otherwise, it will be set to 0. Algorithm 2 shows the detailed steps of FTSI.

4 Experiments Design

4.1 Datasets

The three algorithms are tested and compared on 4 UCI datasets. Tab. 1 shows all datasets, where all data is ordered and represented by "DSth", "NoE", "NoF", "NoC" respectively represent the number of instances, the number of features, and the number of categories. And each dataset consists of 70% training set and 30% test set.

Table 1: Information of datasets

ID	Datasets	NoE	NoF	NoC
DS1	ConnectionistBenchData	208	60	2
DS2	hill	606	100	2
DS3	musk1	476	166	2
DS4	marti	150	1024	2

4.2 Parameter Settings

In this paper, mixed initialization, random initialization and our proposed method are applied to compare experimental results in the following experiments. The involved Algorithm 2 and parameter settings are shown below:

- The standard PSO (random initialization);
- PSO of mixed-initialization (mixed initialization);
- Population initialization with PCA (PCA-PSO) (our approach);

In this experiment, we use k-Nearset Neighbour as classifier and the parameter k is set to 3. The population size ps is set to 100 and the maximum health assessment is set to 300,000. Set to 0.6, the threshold θ can be used to determine appropriate features. We also use a continuous value limited to [0,1] to represent a solution for each particle. The average number of evaluations is utilized to measure the performance of convergence. To facilitate the comparison, the same parameter settings were kept same as in involved algorithms to ensure a fair comparison. All the experiments are run over 30 times on each dataset, and the average results throughout the optimization runs are recorded.

5 Results and Analysis

The comparison result among PSO, PSO of mixed-initialization and PCA-PSO are shown in Tabs. 2–4, where "Mean" stands for the average best function value found in the last generation and "Std" indicates the standard deviation throughout 30 runs. Tab. 2 shows the solution size of PSO on training sets with three initialization methods. Tab. 3 shows the classification accuracy of three methods on training sets. And the corresponding classification accuracy on test sets is shown in Tab. 4. Among them, "P-M" means the mixed initialization method, "P-R" means the random initialization method, and "P-F" means the method FTSI we proposed. "CA" represents the classification accuracy. "SZ" represents the solution size. And the bold represents the best average for each dataset.

Table 2: Solution size of different methods on training

Datasets	P-M		P-R		P-F	
	mean	std	mean	std	mean	std
DS1	26.00	2.83	**20.00**	1.83	27.75	2.06
DS2	42.75	6.08	42.00	5.72	**42.00**	4.97
DS3	77.25	19.02	68.00	7.62	**66.00**	2.16
DS4	480.50	135.25	411.50	9.75	**382.00**	66.86

Algorithm 2: Detailed steps of FTSI method

Input: population size (ps), the number of features (D), the threshold (θ);

Output: $\mathrm{PF}_{pS\times D}$

```
Create matrix P and all zero matrix PF. Then PCA is used to evaluate and analyze the features. After sorting in VS from large to small, the features with first 80% of weights are selected.
for i ≤ ps do
    for 80% of particles do
        for j ≤ D do
            if (p_{i,j} ≥ θ) and (j ⊆ VS) then
                PF_{i,j} = 1;
            end
            else
                PF_{i,j} = 0;
            end
        end
    end
    for 20% of particles do
        for j ≤ D do
            if (p_{i,j} < θ) and (j ⊆ VS) then
                PF_{i,j} = 1;
            end
            else
                PF_{i,j} = 0;
            end
        end
    end
end
```

5.1 Results of Solution Sizes

Tab. 2 shows the solution size after the data are processed by three method. We can see that all three methods can effectively reduce the feature size by 54% to 67%. Moreover, as the number of features increase, the better effect of the proposed method is played on feature dimension reduction, which indicates that our proposed method may have certain advantages when dealing with large-scale features. Perhaps it is because PCA has an inherent advantage for high-dimensional data.

5.2 Results of Classification Accuracy on Training Sets and Test Sets

As Tab. 3 and Tab. 4 show, the proposed method has excellent performance on all the test sets and the training sets we used. Although in DS1, FTSI performs slightly worse on the DS1 test set, it is almost as efficient as the other two methods and can be ignored. This may be because the method we proposed is not very good at data with lower dimensions. In particular, the method we proposed is outstanding in DS2 training set, and the effect is more than 35% of the other two methods. Perhaps because the method we proposed is more suitable for DS2 training set. It can be obtained easily from the above analysis that our proposed method

is effectively helpful to reduce the feature dimension thus to improve the classification accuracy on dataset. This proves that PCA really helps the population initialization of PSO and enhances its effect.

Table 3: Classification accuracy of different methods on training sets (%)

Datasets	P-M		P-R		P-F	
	mean	std	mean	std	mean	std
DS1	0.9202	0.0183	0.9310	0.0158	**0.9359**	0.0186
DS2	0.6415	0.0051	0.6467	0.0118	**0.9853**	0.0022
DS3	0.9189	0.0104	0.9234	0.0063	**0.9324**	0.0058
DS4	0.8829	0.0023	0.8821	0.0027	**0.8929**	0.0017

Table 4: Classification accuracy of different methods on test sets (%)

Datasets	P-M		P-R		P-F	
	mean	std	mean	std	mean	std
DS1	**0.7544**	0.0366	0.7542	0.0450	0.7510	0.0637
DS2	0.4698	0.0104	0.4886	0.0209	**0.4908**	0.0034
DS3	0.7907	0.0170	0.7711	0.0105	**0.7953**	0.0455
DS4	0.8769	0.0039	0.8785	0.0066	**0.8934**	0.0055

5.3 Convergence Performance of Different Methods on the Training Sets

Fig. 1 shows the classification accuracy of three different population initialization methods in PSO algorithm on four training sets as they converge over time. Compared to the other two methods, it can be obviously obtained easily from the figure that our proposed method can quickly obtain a higher accuracy in a shorter time. And in the later convergence process, its convergence speed is slower, that is to say, it still has higher momentum to reach higher accuracy, which is especially obvious in high-dimensional data. This also proves that our proposed method does have a more efficient effect on high-dimensional data.

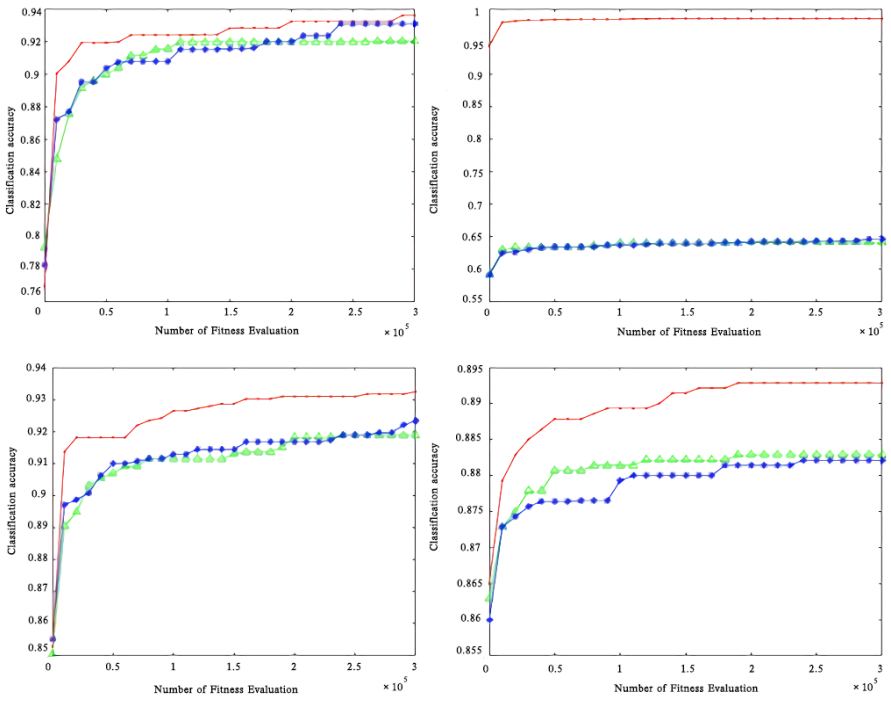

Figure 1: Convergence curves of different methods on DS1-DS4

6 Conclusions and Future Work

This paper proposes a new initialization approach of PSO for initialization of population. The proposed population initialization method considering PCA reduces features by more than half but still retains most of the information. The experimental results also indicate that the proposed method can increase the classification accuracy effectively while reducing the feature dimension, which is especially obvious in high-dimensional data.

Of course, our experiments only verified a small number of data sets, and whether the proposed method is suitable for most data sets is still debatable. And there are many filter methods, each filter method has its scope of application. Whether there is a better filter approach to improve the initialization of population in PSO is still worth exploring. These are all directions we will continue to study in the future.

Funding Statement: This work is supported by National Natural Science Foundation of China (Grant Nos. 61876089, 61403206), by Science and Technology Program of Ministry of Housing and Urban-Rural Development (2019-K-141), by Entrepreneurial Team of Sponge City (2017R02002), and by Innovation and entrepreneurship training program for College Students.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] B. Nguyen, B. Xue, I. Liu and M. Zhang, "Filter based backward elimination in wrapper based PSO for feature selection in classification," *IEEE Congress on Evolutionary Computation*, pp. 3111–3118, 2014.

[2] A. Solberg and A. K. Jain, "Texture fusion and feature selection applied to SAR imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, pp. 475–479, 1997.

[3] D. L. David, "Feature selection and feature extraction for text categorization," *Association for Computational Linguistics*, pp. 212–217, 1992.

[4] T. Due, A. K. Jain and T. Taxt, "Feature extraction methods for character recognition–A survey," *Pattern Recognition*, vol. 29, no. 4, pp. 641–662, 1996.

[5] X. Zheng, D. Levine, J. Shen, S. M. Gogarten, C. Laurie *et al.,* "A high-performance computing toolset for relatedness and principal component analysis of SNP data," *Bioinformatics*, vol. 28, no. 24, pp. 3326–3328, 2012.

[6] K. Z. Mao, "Orthogonal forward selection and backward elimination algorithms for feature subset selection," *IEEE Transactions on Cybernetics, Part B*, vol. 34, no. 1, pp. 629–634, 2004.

[7] D. L. L. Beatriz, "Evolutionary computation for feature selection in classification problems," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 3, no. 6, pp. 381–407, 2013.

[8] B. Xue, M. Zhang and M. N. Browne, "Particle swarm optimization for feature selection in classification: Novel initialisation and updating mechanisms," *Applied Soft Computing*, vol. 18, pp. 261–276, 2014.

[9] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.

[10] H. Maaranen, K. Miettinen and M. Mäkelä, "Quasi-random initial population for genetic algorithms," *Computers & Mathematics with Applications*, vol. 47, no. 12, pp. 1885–1895, 2004.

[11] D. Bajer, G. Martinović and J. Brest, "A population initialization method for evolutionary algorithms based on clustering and Cauchy deviates," *Expert Systems with Applications*, vol. 60, pp. 294–310, 2016.

[12] W. F. Gao, S. Y. Liu and L. L. Huang, "Particle swarm optimization with chaotic opposition-based population and stochastic search technique," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 11, pp. 4316–4327, 2012.

[13] H. Wang, Z. J. Wu, J. Wang, X. J. Dong, S. Yu *et al.,* "A new population initialization method based on space transformation search," in *2009 Fifth Int. Conf. on Natural Computation*, pp. 332–336, 2009.