A GA-like Dynamic Probability Method With Mutual Information for Feature Selection

Gaoshuai WANG · Fabrice Lauri · Amir HAJJAM EL HASSANI

Received: date / Accepted: date

Abstract Feature selection plays a vital role in promoting the classifier’s performance. However, current methods ineffectively distinguish the complex interaction in the selected features. To further remove these hidden negative interactions, we propose a GA-like dynamic probability (GADP) method with mutual information which has a two-layer structure. The first layer applies the mutual information method to obtain a primary feature subset. The GA-like dynamic probability algorithm, as the second layer, mines more supportive features based on the former candidate features. Essentially, the GA-like method is one of the population-based algorithms so its work mechanism is similar to the GA. Different from the popular works which frequently focus on improving GA’s operators for enhancing the search ability and lowering the converge time, we boldly abandon GA’s operators and employ the dynamic probability that relies on the performance of each chromosome to determine feature selection in the new generation. The dynamic probability mechanism significantly reduces the parameter number in GA that making it easy to use. As each gene’s probability is independent, the chromosome variety in GADP is more notable than in traditional GA, which ensures GADP has a wider search space and selects relevant features more effectively and accurately. To verify our method’s superiority, we evaluate our method under multiple conditions on 15 datasets. The results demonstrate the outperformance of the proposed method. Generally, it has the best accuracy. Further, we also compare the proposed model to the popular heuristic methods like POS, FPA, and WOA. Our model still owns advantages over them.

Keywords Genetic Algorithm · Dynamic Probability · Feature Selection · Mutual Information

Gaoshuai WANG, Amir HAJJAM EL HASSANI
Nanomédecine, Imagerie, Thérapeutique - EA4662, Université Bourgogne Franche-Comté, UTBM, F-90010 Belfort, France
E-mail: gaoshuai.wang@utbm.fr, amir.hajjam-el-hassani@utbm.fr

Fabrice Lauri
Connaissance et Intelligence Artificielle Distribuées (CIAD) - UMR 7533, Université Bourgogne Franche-Comté, UTBM, F-90010 Belfort, France
E-mail: fabrice.lauri@utbm.fr
1 Introduction

Feature selection is a non-negligible pre-processing step when dealing with high-volume or high-dimensional datasets [1]. It has been extensively used in various domains, such as image process [2] [3], data clustering [4] [5], data classification [7] [8]. By eliminating the irrelevant features and reducing the dataset’s feature scale, it can promote the classifier’s performance and convergence speed [9]. In total, we categorize feature selection methods into three parts, typically, they are filter methods, wrapper methods, and embedded methods [10]. Filter methods choose a certain number of features based on criteria like correlation, similarity, information gain, etc; Wrapper methods attempt to evaluate all feature combinations with the classifier which needs lots of computing resources; The embedded methods integrate feature selection and classifier learning into a single process. Filter methods usually are not as computationally expensive as wrapper methods. While filters lack the evaluation of the selected features’ performance on a classifier, wrappers contain a classifier as a part of it, which leads to a better result by wrappers than filters [11].

The appearance of Shannon entropy theory [12] provides a series of powerful methodologies. Methods like the decision tree and mutual information are designed based on Shannon’s theory. As noted above, the mutual information method should belong to the filter method, which assesses a feature’s importance by information gain. If the existence of one feature can tremendously alleviate the uncertainty of the class label, it represents the feature is in favor of the classifier to discriminate the classification. In recent years, it has gained much attention, and lots of works have demonstrated mutual information’s effectiveness [13]. Initially, researchers directly rank the candidate features and adopt the features with top-ranked mutual information as the input of the classifier. However, this method ignored the redundancy existing in features, which will harm the former mutual information selection. Fig. 1 depicts the accuracy growth tendency to the increment of subset features. The accuracy curve’s fluctuation means some features are redundant. Hence, MRRMR (minimum redundancy maximum relevance) is proposed by Ding et al. [14] which considers both relevancy and redundancy and achieved huge success. It also inspired researchers to develop more similar methods.

Nevertheless, after summarizing mutual information methods, we found that they only locally release some interactions, for example, these methods often employ the sum of mutual information between the candidate feature and individual feature to replace the mutual information between the candidate feature and the whole of the selected subset. Namely, it’s not a perfect replacement and remains some redundancies.

Except for the mutual information methods, researchers also attempt to introduce the heuristic methods such as genetic algorithm (GA) [15], particle swarm optimization (PSO) [16] to select optimal features [11] [17]. Heuristic methods are a series of algorithms inspired by creatures and often used in NP-hard problems. Commonly, the heuristic method owns a strong global search ability which assures they can perform well [18]. GA is one of the high frequently applied heuristic methods, which is from the biological evolution process [19]. A classical GA comprises several elements, such as chromosome, gene, fitness function, crossover, and mutation. The representative flow generally goes through: initialize the chromosome; select the chromosome by a certain fitness function, generate the new chromosome,
The whole process will end until the result converges or reaches the threshold. In most cases, GA method is a feature selector with some classifiers for searching in a global space. However, directly employing GA to choose features still has some defects: it costs too much computation time and hardly makes a balance between local search and global search [20]. Besides, there are plenty of parameters lying in the traditional GA methods which adversely increase the usage difficulty [21].

Considering the characteristics of filter and wrapper methods, the hybrid algorithm has increasingly gained attraction from researchers and various combined methods are constantly emerging. Admiring the merits of the hybrid algorithm, we propose a novel GA-like dynamic probability algorithm with mutual information for feature selection. In our method, the mutual information method is employed to rank all features and then selects top-\(N\) features. It can be treated as a global feature selector. Then, the GA-like method exploits further search locally and offers the selected features to a classifier. In terms of the GA-like method, it abandons the crossover and mutation operators and generates the new chromosome depending on each gene’s probability. It’s easier for setting parameters and strengthen our proposed method’s local search ability.

In this research, our contributions are:

1) We propose a two-layer feature selection model which both considers the global search and local search.
2) To accelerate the feature selection, we employ the MRMR method as the global selector.
3) We propose a new GA-like dynamic probability method that is simpler compared to GA for searching features locally.
4) Dynamic probability update mechanism replaces the complex operators in GA.
5) Lots of experiments are exploited to demonstrate the characteristics and performance of the proposed method.

The rest of this paper is arranged as follows. In Section 2, we present some related works on mutual information and heuristic algorithm in feature selection. Section 3 presents the main concept of the GA method and other methods. Section 4 details the proposal GA-like method and the whole process of feature selection. In Section 5, the datasets and processing method are described. After, we compare our method to the classical GA method with various conditions. Finally, Section 6 concludes this paper and refers to the possible further work.
2 Related works

Numerous works have been done since the effectiveness of genetic algorithms and mutual information algorithms. This section will present researchers’ efforts at improving and applying them in various domains.

As for the genetic algorithm, researchers often improve its crossover and mutation operators which have a profound effect on its performance. Xue et al. \[22\] proposed an adaptive crossover operator based on a genetic algorithm for feature selection. In their method, they adopted five crossover operators with proper probability and each of them will be selected in different process stages because they consider the single crossover operator is problem-dependent. Their method was compared with five well-known evolutionary multi-objective algorithms on ten datasets. The results demonstrated that the proposed algorithm can remove redundant features and keep low classification error and is superior to other methods. Manzoni et al. \[23\] used three different balanced crossover operators in cryptography and design theory for the sake of their balancedness constraint. They found that the balanced crossover operators perform better than the one-point crossover. Sun and Liu \[24\] considered that premature convergence often happened on classical genetic algorithms due to the rapid multiplication of outstanding individuals. They put up a method to qualify the diversity and similarity between adjacent generations and adjust the crossover and mutation probability dynamically. Their method is proven that greatly improved the convergence speed and global search ability. In \[25\], GA was used to select features directly for the second elastic net which was flexible in adjusting the penalty terms in the regularization process and time efficiency. Moreover, researchers also explored the initialization \[26, 27\], encoding/decoding schemes \[28, 29\] and other aspects.

About mutual information, reducing irrelevant features, and keeping supportive features are the core of researchers’ work. Maes et al. \[30\] applied the MIM method for estimating the parameters of the hidden Markov model in 1986. It promoted the probability largely when parameters were estimated by maximum likelihood estimation. However, the feature relevancy ignored by MIM makes mutual information still has a huge improvement space. Battiti \[31\] proposed the mutual information feature selection (MIFS) which considers both the relevancy and redundancy. $\beta$ represents the parameter of the feature redundancy. But, it’s difficult to choose a proper value of $\beta$. The same problem also happens on its variants, MIFS-ND and MIFS-U \[32, 33\]. To balance the relationship between feature relevancy and feature redundancy, Pent et al. \[34\] set the $\beta$ equal to the inverse of the selected feature scale in MRMR. After that, other methods such as conditional information feature extraction (CIFE) \[35\], double input symmetrical relevance (DISR) \[36\], and Joint mutual information (JMI) \[37\] are brought up which emphasised different aspects.

Except for the single genetic algorithm and the single mutual information method, the combined methods of genetic algorithm and mutual information attracted many researchers’ attention. Jadhav et al. \[38\] carried out the information gain and genetic algorithm in credit rating. The gained information availed reducing the computing complexity and the genetic algorithm further promoted the classifiers’ performance. Salesi et al. \[39\] used a two-layer framework that combined the MRMR method and tabu asexual genetic algorithm to enhance the generalization power. The results show that their algorithm was superior to other conven-
tional and state-of-art algorithms. Huang et al. [40] made the genetic algorithm as the first layer and mutual information as the second layer. Their experimental results show that the hybrid method achieved excellent classification accuracy.

3 Preliminaries

Darwin’s theory of evolution by natural selection inspires the appearance of GA that mimics nearly all the stages of gene heredity. Generally, gene heredity is a complex process in biological view, which mainly takes account of several vital parts: such as chromosome representation, crossover, mutation, and fitness selection. Chromosome representation transfers the specific problem into chromosome format; Crossover indicates the parent chromosomes change their part gene in the same position and produce new chromosomes; Mutation represents some positions of the chromosomes are randomly flipped on a certain probability; Fitness selection will discard or change gene’s existence probability in the next off-springs, which assures the gene’s environmental adaptability.

Fig. 2 depicts several relative elements and processes in GA. The color bar means a chromosome where several genes exist.

1) **Chromosome representation**: In the beginning, we need to transform our specific problem into a chromosome which is an essential step because it determines the successful application of GA. Different from the creature which usually contains dozens of chromosomes, in GA, one chromosome represents the whole resolution space, numerous chromosomes will form a population. In terms of feature selection, we can regard each feature as a chromosome gene located in a fixed position. A feature only has two states: be chosen or not, in Fig. 2, color block means the gene is selected. There are lots of chromosomes in each process which assures the global search ability of GA.

2) **Crossover**: When the new generation chromosomes are produced, the crossover will happen among the chromosomes, for example, in Fig. 2 Step1 to Step2, two parents’ chromosomes generate two children. G1, G4, and G6 are exchanged between the parents. Crossover increasingly enhances solution space’s variety and avoids the local optimum. But, it also adds more parameters in the setting process, the crossover proportion, and position.

3) **Mutation**: Mutation means a novel gene appearing or some changes beyond the crossover process. In Fig. 2 the mutation process exists in Step2 to Step3. G7 disappears in one chromosome and a new gene appears in the G8 position. It must be emphasized that there is no new gene’s appearance in the feature selection problem.

4) **Fitness selection**: After the previous processes, fitness selection evaluates each individual’s performance. In the natural environment, an excellent gene will strengthen the individual’s survival ability, then, its heredity probability increases, correspondingly, and the population gradually owns the adaptability. In GA, when dealing with the fitness result, people usually take two kinds of actions: One is discarding the bad performance individuals directly; Another is reducing their chances of reproduction.

Loop these four steps repeatedly until trigger the terminal conditions.
PSO was brought up by Kennedy and Eberhart. The basic elements named particles are placed in the search space and the fitness function evaluates its current location. Then, each particle decides its velocity and direction in the space relying on its current location and the best location with some members of the swarm. All the particles that have been moved are a complete iteration. Eventually, all the particles like a flock of birds seek food until they find the fitness function’s optimum.

Flower Pollination Algorithm (FPA) \[41\] was inspired by the natural pollination process. FPA incorporates biotic cross-pollination and abiotic pollination as the global search and local search processes. Pollen-carrying pollinators such as insects or the wind move in a way that obeys Levy flights. Besides, the switching of local pollination and global pollination is controlled by the probability \( p \in [0, 1] \). The whole process intimates the plants’ pollination.

The whale optimization algorithm (WOA) \[42\] mimics the social behavior of humpback whales. The way that whales hunt their prey is named the bubble-net feeding method. In WOA, each solution is regarded as a whale. The whale moves its place in the search space considered as a reference to the best element of the group. Two search mechanisms are used by the whales to hunt. In the first one, the preys are encircled and the second is the bubble net.

4 Proposed method

Although GA has a puissant ability for global search, it’s exhaustive when the scale of the dataset’s dimensionality increases to several thousand. It’s not easy for GA to obtain a pretty good result. Considering the high effectiveness of filter methods such as mutual information, we combine these two methods and form a two-layer hybrid method. In mutual information methods proposed with different considerations, we found that MRMR often is one of the baselines and its performance is well. Thus, in the first layer, we adopt the MRMR method for ranking all the candidate features and select top-\( N \) features as the second layer’s input. Next, the proposed GA-like method with a specific classifier to search for optimal combinations until it triggers some termination conditions.
Let \( X = \{x_1, x_2, \ldots, x_n\} \) be a feature vector and \( p(x_i) \) is the probability of \( x_i \), let \( Y = \{y_1, y_2, \ldots, y_n\} \) as the class vector. The mutual information between \( X \) and \( Y \) is calculated in Eq. 1:

\[
I(X, Y) = H(X) - H(X|Y)
\] (1)

### Algorithm 1: MRMR [13]

**input**: The training dataset \( D \) with original feature set \( F = \{f_1, f_2, \ldots, f_n\} \), class label \( Y \) and the required feature number \( T = 70 \)

**output**: The selected subset feature \( S_T \)

1. \( S_T \leftarrow \emptyset; \)
2. \( \text{foreach } f_i \text{ in } F \text{ do } M_{i} = I(f_i; y); \)
3. \( f \leftarrow \max(MI); \)
4. \( S_T \cup f; \)
5. \( F = F - f; \)
6. \( \text{for } i \leftarrow 2 \text{ to } T \text{ do } \)
7. \( \text{for each } f_i \text{ in } F \text{ do } \)
8. \( J(f_i) = M_{i} - \frac{1}{|S_T|} \sum_{f_j \in S_T} I(f_j; f_i); \)
9. \( \text{select } f_s \text{ from } J(f_i) \text{ with the largest value}; \)
10. \( S_T = S_T \cup f_s; \)
11. \( F = F - f_s; \)
12. \( \text{return } S_T \)

### 4.1 GA-like dynamic probability method

GA has already been applied in many areas because it can obtain pretty results in a short time even with a complex task. In GA, the typical processes are crossover, mutation, selection, and reproduction. The crossover and mutation make the offsprings inherit merits from their parent chromosomes and surpass them in variety. Meanwhile, there is an inevitable issue that we should have a proper change rate and position during the crossover and mutation processes which easily lead to results unpredictable. Hence, to discover better operators, researchers designed lots of rules in crossover and mutation processes, such as the single point crossover, uniform crossover, cycle crossover; Inversion mutation, the bit flipping mutation, etc. What’s more, the selection method is also mandatory when the algorithm produces a new generation. As mentioned above, the GA parameters setting is so complex that researchers need to spend much attention on this part.

Furthermore, we think that GA could not balance the trade-off well locally and globally as there is no feature pre-screening, which will hinder its application. For simplifying the traditional GA, we propose a GA-like method based on a dynamic probability mechanism. In the whole process, the birth of each new generation is according to their probability.

1. **Encode and Decode.** The feature selection problem is simple to encode, we just employ the binary encode to process the feature subset. Its merit is that each feature is independent which assures chromosome variety. In the binary
encode, ‘1’ represents the feature participating to classify the label; ‘0’ indicates the feature is not selected. Decode is transforming the binary chromosome to the feature subset. We define the $i$th chromosome in population as $\theta_i$, the $g$th gene or feature in chromosome $\theta_i$ as $\theta_{ig}$. $P_t(\theta_{ig})$ is the probability of $i$th chromosome’s $g$th gene at $t$th generation.

2. **Initialization.** Initialization demonstrates whether a feature should be selected or not in the beginning. In most cases, GA adopts the random initialization method. If the scale of the population is large enough, chromosome variety enhances its global search ability. In our proposed method, whether a feature appears in the chromosome depends on its probability. All the features have the same probability. The too-large probability will degrade the selection function and the too-small probability will decrease the variety of chromosomes.

$$P_0(\theta_{ig}) = 0 \quad 0 \leq i < n, 0 \leq g < m$$  \hspace{1cm} (2)

where $n$ is the number of population, $m$ is the amount of gene or feature in one chromosome. $P_0$ is the probability at 0th generation.

3. **Fitness selection.** In our method, we introduce the Gaussian bayesian (GB), Support vector machine (SVM), and K-nearest neighbors (KNN) as the evaluation function. The higher accuracy means the chromosome has a better performance. In each Fitness selection, $1/3$ population will be discarded. The rest of the population will take part in the probability update. Among these populations, The best performance subset will be kept. If a chromosome in the next generation is better than the current chromosome, it will be replaced.

4. **Probability update.** After the discarding process, relying on remained chromosomes, each gene’s probability will be updated. If the existence of one feature promotes accuracy, which indicates the feature is positive in classification, its probability will increase in the next generation. The probability $P_t(\theta_{ig})$ is calculated as:

$$E_{t-1}(\theta_{ig}) = \begin{cases} 1 & \text{gene selected} \\ 0 & \text{gene not selected} \end{cases}$$  \hspace{1cm} (3)

$$P_t(\theta_{ig}) = \left(\sum_{i=1}^{2n} E_{t-1}(\theta_{ig})\right)/\frac{2n}{3}$$  \hspace{1cm} (4)

5. **Chromosome replication.** Different from the popular chromosome replication method where the new chromosome is produced from a pair of parents. In our proposed method, there is no parent chromosome, all the genes or features only rely on each probability.

To have a better understanding of the variety, we apply the information entropy to calculate the information volume of each gene. At generation $t$, the information entropy of gene $g$ and a chromosome are computed by Eq. 5 and Eq. 6. In Fig. 3, we present one gene’s information entropy variation with its probability.

$$H_t(\theta_{ig}) = -P_t(\theta_{ig}) \ln(P_t(\theta_{ig})) - (1 - P_t(\theta_{ig})) \ln(1 - P_t(\theta_{ig}))$$  \hspace{1cm} (5)

$$H_t(\theta) = \sum_{g} H_t(\theta_{ig})$$  \hspace{1cm} (6)
Fig. 3: The information entropy of gene $g$. Point 1 is the initial information entropy. With the growth of generations, each gene’s probability tends to be stable, these supportive genes’ information entropy will be close to Point 2 and the redundant genes’ information entropy will reach Point 3.

In contrast to the classical GA, seemingly, their selection method, such as the elite selection or probabilistic selection is similar to our proposed method. Information entropy also can estimate GA’s information volume. However, each gene’s probability is not independent on account of the crossover mechanism that decreases off-springs uncertainty and the mutation only increases the uncertainty limitedly. What’s more, the proposed method needn’t keep the aforementioned elite generation chromosomes for obtaining good results, all training process influences are in the probability. Thus, we deem that our proposed method has a more wide search area. We display the Pseudo-code of GADP in Algorithm 2.

Finally, we present the flowchart of our proposed method and classical GA with mutual information process in Fig. 4. It could be brutally divided into two search parts: global search and local search. MRMR works for the global search part, which reduces the scale of search space largely; GADP is responsible for further removing redundant features. Compared to the classical GA, its structure is more compact.

Fig. 4: The flowchart of proposed method and classical GA
Algorithm 2: Pseudo-code for GADP

\textbf{input :} Feature set $F = \{f_1, f_2, \ldots, f_n\}$, class label $Y$, population set $N$, initial probability set $P$, iteration number $I$, threshold $t$, discard rate $l$

\textbf{output:} The selected subset feature $S_F$

1. $N.size \leftarrow 50, P.size \leftarrow \text{len}(F), I \leftarrow 50, t \leftarrow 10$
2. Push $N.size$ empty chromosomes into population set $N$
3. $P \leftarrow \{0.65, 0.65, \ldots, 0.65\}$
4. $i \leftarrow 0, \text{Bestset} \leftarrow \varnothing$
5. While $i < I$ do
6. \hspace{1em} \textbf{foreach} chromosome $\theta$ in population set $N$ do
7. \hspace{2em} $g \leftarrow 1$ at the probability of $P(g)$ // Produce new chromosome
8. \hspace{1em} \textbf{foreach} chromosome $\theta$ in population set $N$ do
9. \hspace{2em} $F_{\text{position}} \leftarrow \text{Decode}(\theta)$
10. \hspace{2em} $x_{\text{train}} \leftarrow F[F_{\text{position}}]$
11. \hspace{2em} $E(\theta) \leftarrow \text{EvulationFunction}(x_{\text{train}}, Y)$
12. \hspace{1em} \text{Discard} $N/3$ chromosomes based on $E(\theta)$
13. \hspace{1em} \text{Bestset} \leftarrow \text{best performance chromosome}$
14. \hspace{1em} \text{Break the condition if Bestset doesn’t update more than $t$ times continuously;}
15. \hspace{1em} \textbf{foreach} gene $g$ in one chromosome do
16. \hspace{2em} $P(g) \leftarrow \text{Eq.}\ 4$ // Probability update
17. \hspace{1em} Remove all chromosomes and push $N.size$ empty chromosomes into population set $N$
18. \hspace{1em} \textbf{return} Bestset

5. Experimental results and analysis

5.1 Metrics

We use the accuracy, $F1$ and AUC as our evaluation indicators, their calculation formulas are as follows:

\begin{equation}
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)
\end{equation}

\begin{equation}
\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \quad (8)
\end{equation}

\begin{equation}
F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)
\end{equation}

\begin{equation}
\text{AUC} = \int_0^1 t_{pr}(f_{pr})df_{pr} = P(X1 > X0) \quad (10)
\end{equation}

where $TP, TN, FP$ and $FN$ are the true positive, true negative, false positive and false negative, respectively. The $t_{pr}$ is the true positive rate, $f_{pr}$ is the false positive rate, and $P(X1 > X0)$ are the probability of the confidence scores $X1 > X0$ for a negative and positive instance. The AUC value describes the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one.
5.2 Dataset and pre-processing

To verify the generalization of our proposed method, we employ 15 datasets to assess its performance. We summarize their characteristics of instance, feature, and classes in Table 1. Three datasets are the binary classification task, and the rest of them are the multi-classification task. Some datasets are continuous types, which will hinder the accuracy of mutual information selection because continuous types could paralyze mutual information calculation. Therefore, we need to redivide all feature attributes into 10 intervals. The first step is normalizing the attribute as Eq. 11. Afterward, multiply $x^*$ by 10, so all the attributes will fall into the 10 regions. It’s a necessary process when using mutual information in a continuous dataset. K-fold cross-validation is used in this research, $K = 5$.

Table 1: The summary of benchmark datasets

| Number | Name              | Instances | Features | Classes |
|--------|-------------------|-----------|----------|---------|
| 1      | PC-MAC            | 1943      | 3289     | 2       |
| 2      | ISOLET           | 1560      | 617      | 26      |
| 3      | Gloma            | 1002      | 5966     | 2       |
| 4      | RELATHÉ          | 1427      | 4322     | 2       |
| 5      | TEX-ATI          | 171       | 574     | 4       |
| 6      | USPS             | 171       | 574     | 4       |
| 7      | CNAE-9           | 1079      | 855      | 9       |
| 8      | Lung discrete    | 73        | 325      | 7       |
| 9      | warpPIE108       | 210       | 2420     | 10      |
| 10     | Lung             | 205       | 3312     | 5       |
| 11     | COIL20           | 1440      | 1024     | 20      |
| 12     | Yale             | 145       | 1024     | 15      |
| 13     | Brain_Tumor      | 90        | 5360     | 5       |

\[ x^* = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \]  \hspace{1cm} (11)

5.3 Experimental setting

Some of our datasets are unbalanced, thus, we use the macro averaging metrics to evaluate the algorithm’s performance. In the mutual information process, we only select the first 70 features provided to GADP and other compared algorithms. Essentially, GA belongs to the wrapper method, the classifier type has a certain influence on the algorithm’s performance, we adopt GB, KNN (k=3), and SVM algorithms as classifiers. What’s more, we compare GADP with GA and other heuristic methods, such as PSO, FPA, and WOA. The hyperparameters and initial settings are described in Table 2. We apply the average value and standard deviation to implement a paired two-tailed t-test with other methods or settings. “+”, “=”, and “−” mean that the objective algorithm performs "better than ", "equal to", and "worse than" the corresponding method When the P-Value is less than 5%. W/T/L indicates the win/tie/loss statistics of one method relative to another method.
Table 2: The parameter setting of heuristic algorithm

| Name   | Parameter Setting                                      |
|--------|--------------------------------------------------------|
| GADP   | max_feature = 50, initial probability = 0.65, population = 50, max_iteration = 50 |
| GA     | max_feature = 50, population = 50, max_iteration = 50 |
| PSO    | particles = 50, max_iteration = 50                    |
| FPA    | search_agents = 50, max_iteration = 50                 |
| WOA    | search_agents = 50, max_iteration = 50                 |

5.4 The effect of MRMR selection

To evaluate the effect of MRMR, we compare the prediction results of MRMR and the full features based on GADP. After primary selection, only the top 70 features remain. In Table 3, we can see that the accuracies of MRMR are higher than the full features on three classifiers except for a few cases. It denotes that eliminating redundant features is effective and MRMR can select more supportive features to promote the algorithm’s performance. Besides, generally, we regard that a smaller feature scale should have a smaller time cost. In most cases, MRMR indeed spends less time because the search space of the full features is wider which makes the algorithm converge slower. As for the full features cases with higher accuracies, we can notice that their time cost usually is higher. This is because searching for better results needs more time. The phenomenon also appears on the MRMR feature subset. For example, COIL20 in GB and KNN.

Table 3: The influence of MRMR selection

| Name     | GADP | Tumor       | Parameter Setting | Tumor       | Parameter Setting | Tumor       | Parameter Setting |
|----------|------|-------------|-------------------|-------------|-------------------|-------------|-------------------|
| 70       | 65.93 | 77.12       | 1.26 (±0.03)     | 65.93       | 77.12             | 1.26 (±0.03) |
| 70       | 65.93 | 77.12       | 1.26 (±0.03)     | 65.93       | 77.12             | 1.26 (±0.03) |
| 70       | 65.93 | 77.12       | 1.26 (±0.03)     | 65.93       | 77.12             | 1.26 (±0.03) |

5.5 The influence of probability setting

Gene probability is the core of GADP, which determines the production of each new chromosome. A proper probability setting is vital for GADP to search the feature space and 3 pairs of probability are studied based on GADP and MRMR in this part. We depict the tendency of partial chromosomes’ probability in Fig. 5. There are three pairs of probability thresholds, 0-1, 0.3-0.8, and 0.4-0.7. We can note that, in the first subfigure, each gene’s probability reaches its final value (0-1) after approximately 20 generations. When the selection process is terminated, the results demonstrate that 2 of 7 genes are selected. The earlier convergence of gene probability will decrease the search space, in some way, which could lead to the local optimum.
thus, further selection tries to eliminate the irrelevant features. In this section, we note that features selected by the MRMR method have a certain redundancy, which makes converge slower and has a more broad search space, thus, it will cost lots of time. In Fig. 5c nearly all the features are under an unstable situation, it seems that the fitness function loses its effect.

Moreover, we also make a comparison of different probability thresholds in Table 4. The accuracies without the probability limitation are better than those with limitations. They achieve 15, 14 and 10 of 15 with the highest accuracies on GB, KNN, and SVM. It seems that the advantages of 0-1 pair on SVM are not as apparent as on GB and KNN. We think that the SVM has a strong information mining capability which partly decimates the probability threshold’s influence. About the time cost, 0-1 pair spends less average time on GB and SVM. However, on KNN, the situation is different and this should be caused by the influence of KNN’s characteristics. On the whole, we consider that the 0-1 pair’s performance is better and adopt it in the rest of the experiments.

5.6 Comparison of GADP with GA and MRMR

We note that features selected by the MRMR method have a certain redundancy, thus, further selection tries to eliminate the irrelevant features. In this section, we
compare the results of GADP algorithm with GA and no process based on MRMR
classified by GB, KNN, and SVM. We depict the average accuracies and standard
deviations in Tables 57. From accuracy and F1 aspects on three classifiers, the
GADP outperforms GA. Both GADP and GA can further promote classifier’s
performance. Therefore, further feature selection based on MRMR is necessary.
About the AUC, MRMR performs better than the results on accuracy and F1. By
comparison, we conclude that GADP is superior to the traditional genetic
algorithm for feature selection.

Table 5: Comparison of GADP with MRMR-GA and MRMR on SVM

| Name          | Accuracy | F1 Score | AUC     |
|---------------|----------|----------|---------|
| MRMR-GA       | 90.97±1.85 | 88.95±2.02 | 86.93±1.14 |
| MRMR          | 76.61±0.48 | 70.96±1.17 | 66.71±0.83 |
| GADP          | 98.54±0.83 | 97.06±1.96 | 96.65±2.42 |

Table 6: Comparison of GADP with MRMR-GA and MRMR on KNN

| Name          | Accuracy | F1 Score | AUC     |
|---------------|----------|----------|---------|
| MRMR-GA       | 90.97±1.85 | 88.95±2.02 | 86.93±1.14 |
| MRMR          | 76.61±0.48 | 70.96±1.17 | 66.71±0.83 |
| GADP          | 98.54±0.83 | 97.06±1.96 | 96.65±2.42 |

Table 7: Comparison of GADP with MRMR-GA and MRMR on SVM

| Name          | Accuracy | F1 Score | AUC     |
|---------------|----------|----------|---------|
| MRMR-GA       | 90.97±1.85 | 88.95±2.02 | 86.93±1.14 |
| MRMR          | 76.61±0.48 | 70.96±1.17 | 66.71±0.83 |
| GADP          | 98.54±0.83 | 97.06±1.96 | 96.65±2.42 |
5.7 Comparison of GADP, PSO, FPA and WOA

In Section 5.4, we explored the effect of MRMR, the results directly applying GADP on full features are not good. Thus, we compare our method to other heuristic algorithms, such as PSO, FPA, and WOA based on MRMR in Table 8. On GB and KNN, our proposed method obtains the highest accuracies nearly on all the datasets. On SVM, GADP achieves 11 of 15 the highest accuracies. Otherwise, it seems that the PSO is a little better than FPA and WOA. What’s more, compared with MRMR’s results in Table 5-7, PSO, FPA and WOA also promote their performances as GADP and GA. Finally, for the feature selection task, we believe that GADP is a considerable method.

Table 8: Comparison of GADP, PSO, FPA and WOA

| Name    | SVM  | GB   | KNN  | SVM  | GB   | KNN  | SVM  | GB   | KNN  | SVM  | GB   | KNN  |
|---------|------|------|------|------|------|------|------|------|------|------|------|------|
| PCMAC   | 91.99| 79.97| 90.97| 82.06| 94.19| ±    | 8.00 | ±    | ±    | ±    | ±    | ±    |
| Name    |      |      |      |      |      |      |      |      |      |      |      |      |
| GADP    | 91.60| 81.22| 91.99| 82.06| 94.19| ±    | 8.00 | ±    | ±    | ±    | ±    | ±    |
| PSO     | 91.50| 81.22| 91.99| 82.06| 94.19| ±    | 8.00 | ±    | ±    | ±    | ±    | ±    |
| FPA     | 91.50| 81.22| 91.99| 82.06| 94.19| ±    | 8.00 | ±    | ±    | ±    | ±    | ±    |
| WOA     | 91.50| 81.22| 91.99| 82.06| 94.19| ±    | 8.00 | ±    | ±    | ±    | ±    | ±    |
| MRMR    | 91.50| 81.22| 91.99| 82.06| 94.19| ±    | 8.00 | ±    | ±    | ±    | ±    | ±    |
| Name    |      |      |      |      |      |      |      |      |      |      |      |      |
| GADP    | 91.60| 81.22| 91.99| 82.06| 94.19| ±    | 8.00 | ±    | ±    | ±    | ±    | ±    |
| PSO     | 91.50| 81.22| 91.99| 82.06| 94.19| ±    | 8.00 | ±    | ±    | ±    | ±    | ±    |
| FPA     | 91.50| 81.22| 91.99| 82.06| 94.19| ±    | 8.00 | ±    | ±    | ±    | ±    | ±    |
| WOA     | 91.50| 81.22| 91.99| 82.06| 94.19| ±    | 8.00 | ±    | ±    | ±    | ±    | ±    |

6 Conclusion

Selecting the optimal subset of features is a vital job for promoting the classifier’s performance. Combining the advantages of different methods is a reasonable approach. In this article, we propose a novel population-based algorithm inspired by the genetic algorithm. Different from the traditional genetic algorithm, the crossover and mutation operators are replaced by the dynamic probability. The value of probability assures the existence of heredity and mutation. For evaluating the performance of our proposed method, we explore the influence of probability limitation. Results show that no limitation makes the convergence faster and accuracy higher. After, we test the effect of MRMR by comparing GADP’s results on the selected features and full features. The results of MRMR are better than the full features’ because there exist lots of irrelevant features which let classifiers trap into local optimum and increase their time consumption. What’s more, we also compare the GADP with GA and no process. Further feature selection can promote the classifier’s results. GADP surpasses GA which indicates our proposed method is accessible. Most of cases, the time cost of MRMR is lower than full features’, at the same time, with higher accuracy. Sometimes, the results of MRMR have a higher time cost and lower accuracy. This may concern the search space. Larger search space and less supportive features make the classifier’s performance bad and cost more time. MRMR filter may lose some key features. Finally, other famous heuristic algorithms like PSO, FPA, and WOA are compared with our method. Still, our algorithm shows its advantages in feature selection. In these heuristic algorithms, PSO performs better than FPA and WOA. In the future, we plan to extend our method to other domains.
References

1. Carlos Oscar Sánchez Sorzano, Javier Vargas, and A Pascual Montano. A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877*, 2014.

2. Jingwei Too and Seyedali Mirjalili. A hyper learning binary dragonfly algorithm for feature selection: A covid-19 case study. *Knowledge-Based Systems*, 212:106553, 2021.

3. Liang Sun, Zhanhao Mo, Fuhua Yan, Liming Xia, Fei Shan, Zhongxiang Ding, Bin Song, Wanchun Gao, Wei Shao, Feng Shi, et al. Adaptive feature selection guided deep forest for covid-19 classification with chest ct. *IEEE Journal of Biomedical and Health Informatics*, 24(10):2798–2805, 2020.

4. R Purushothaman, SP Rajagopalan, and Gopinath Dhandapani. Hybirdizing gray wolf optimization (gwo) with grasshopper optimization algorithm (gos) for text feature selection and clustering. *Applied Soft Computing*, 96:106651, 2020.

5. Ammar Kamal Abasi, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, Syibrah Naim, Sharif Naser Makhadmeh, and Zaid Abdi Alkareem Alyasser. An improved text feature selection for clustering using binary grey wolf optimizer. In *Proceedings of the 11th national technical seminar on unmanned system technology 2019*, pages 503–516. Springer, 2021.

6. Guo Zhong and Chi-Man Pun. Subspace clustering by simultaneously feature selection and similarity learning. *Knowledge-Based Systems*, 193:105512, 2020.

7. Yonghong Peng, Zhiqing Wu, and Jianmin Jiang. A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics*, 43(1):15–23, 2010.

8. Jiliang Tang, Salem Aledyani, and Huan Liu. Feature selection for classification: A review. *Data classification: Algorithms and applications*, page 37, 2014.

9. Kuan-Cheng Lin, Yi-Hung Huang, Jason C Hung, and Yung-Tso Lin. Feature selection and parameter optimization of support vector machines based on modified cat swarm optimization. *International Journal of Distributed Sensor Networks*, 11(7):365869, 2015.

10. Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6):1–45, 2017.

11. Bing Xue, Mengjie Zhang, Will N Browne, and Xin Yao. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626, 2015.

12. Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

13. Jorge R Vergara and Pablo A Estévez. A review of feature selection methods based on mutual information. *Neural computing and applications*, 24(1):175–186, 2014.

14. Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

15. John H Holland. Genetic algorithms and the optimal allocation of trials. *SIAM journal on computing*, 2(2):88–105, 1973.
16. Russell Eberhart and James Kennedy. Particle swarm optimization. In Proceedings of the IEEE international conference on neural networks, volume 4, pages 1942–1948. Citeseer, 1995.

17. Arpan Kumar Kar. Bio inspired computing—a review of algorithms and scope of applications. Expert Systems with Applications, 59:20–32, 2016.

18. Chu Kwong Chak and Gang Feng. Accelerated genetic algorithms: combined with local search techniques for fast and accurate global search. In Proceedings of the 1995 IEEE International Conference on Evolutionary Computation. Part 1 (of 2), pages 378–383, 1995.

19. Zbigniew Michalewicz and Marc Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. Evolutionary computation, 4(1):1–32, 1996.

20. Tarek A El-Mihoub, Adrian A Hopgood, Lars Nolle, and Alan Battersby. Hybrid genetic algorithms: A review. Eng. Lett., 13(2):124–137, 2006.

21. Nikolaus Hansen, Dirk V Arnold, and Anne Auger. Evolution strategies. In Springer handbook of computational intelligence, pages 871–898. Springer, 2015.

22. Yu Xue, Haokai Zhu, Jiayu Liang, and Adam Slowik. Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification. Knowledge-Based Systems, 227:107218, 2021.

23. Luca Manzoni, Luca Mariot, and Eva Tuba. Balanced crossover operators in genetic algorithms. Swarm and Evolutionary Computation, 54:100646, 2020.

24. Na Sun and Yong Lu. A self-adaptive genetic algorithm with improved mutation mode based on measurement of population diversity. Neural Computing and Applications, 31(5):1435–1443, 2019.

25. Fatemeh Amini and Guiping Hu. A two-layer feature selection method using genetic algorithm and elastic net. Expert Systems with Applications, 166:114072, 2021.

26. Debjani Mustafi and Gadadhar Sahoo. A hybrid approach using genetic algorithm and the differential evolution heuristic for enhanced initialization of the k-means algorithm with applications in text clustering. Soft Computing, 23(15):6361–6378, 2019.

27. Deepti Gupta and Shabina Ghafir. An overview of methods maintaining diversity in genetic algorithms. International journal of emerging technology and advanced engineering, 2(5):56–60, 2012.

28. Anit Kumar. Encoding schemes in genetic algorithm. International Journal of Advanced Research in IT and Engineering, 2(3):1–7, 2013.

29. Antonio Costa, Giovanni Celano, Sergio Fichera, and Enrico Trovato. A new efficient encoding/decoding procedure for the design of a supply chain network with genetic algorithms. Computers & Industrial Engineering, 59(4):986–999, 2010.

30. Lalit Bahl, Peter Brown, Peter De Souza, and Robert Mercer. Maximum mutual information estimation of hidden markov model parameters for speech recognition. In ICASSP’86. IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 11, pages 49–52. IEEE, 1986.

31. Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on neural networks, 5(4):537–550, 1994.
32. Nazrul Hoque, Dhruba K Bhattacharyya, and Jugal K Kalita. Mifs-nd: A mutual information-based feature selection method. *Expert Systems with Applications*, 41(14):6371–6385, 2014.
33. Nojun Kwak and Chong-Ho Choi. Input feature selection for classification problems. *IEEE transactions on neural networks*, 13(1):143–159, 2002.
34. Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.
35. Dahua Lin and Xiaoou Tang. Conditional infomax learning: An integrated framework for feature extraction and fusion. In *European conference on computer vision*, pages 68–82. Springer, 2006.
36. Patrick Emmanuel Meyer, Colas Schretter, and Gianluca Bontempi. Information-theoretic feature selection in microarray data using variable complementarity. *IEEE Journal of Selected Topics in Signal Processing*, 2(3):261–274, 2008.
37. Howard Yang and John Moody. Data visualization and feature selection: New algorithms for nongaussian data. *Advances in neural information processing systems*, 12, 1999.
38. Swati Jadhav, Hongmei He, and Karl Jenkins. Information gain directed genetic algorithm wrapper feature selection for credit rating. *Applied Soft Computing*, 69:541–553, 2018.
39. Saeidja Salesi, Georgina Cosma, and Michalis Mavrovouniotis. Taga: Tabu asexual genetic algorithm embedded in a filter/filter feature selection approach for high-dimensional data. *Information Sciences*, 565:105–127, 2021.
40. Jinjie Huang, Yunze Cai, and Xiaoming Xu. A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern recognition letters*, 28(13):1825–1844, 2007.
41. Xin-She Yang. Flower pollination algorithm for global optimization. In *International conference on unconventional computing and natural computation*, pages 240–249. Springer, 2012.
42. Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in engineering software*, 95:51–67, 2016.