Application of adaptive genetic algorithm and machine learning in English text analysis teaching system

Xuan Wang
Huaxin College of Hebei GEO University

Yuanyuan Chen (✉ chouyuaner1986@163.com)
Hebei University of Science and Technology

Research Article

Keywords: Adaptive genetic algorithm, English text, background elimination, recognition model, machine learning

Posted Date: May 3rd, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2767248/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
Abstract

Fuzzy English text recognition will be affected by the complex background, resulting in low recognition accuracy. In order to improve the effect of English text recognition, it is necessary to remove the background. Based on the machine vision algorithm, this paper improves the traditional genetic algorithm to obtain an adaptive genetic algorithm, and builds an English text background elimination model based on the adaptive genetic algorithm. Moreover, this paper introduces the hill-climbing method with better local optimization effect to perform local optimization in the iterative process of genetic algorithm, and constructs an adaptive genetic algorithm based on the hill-climbing method to make up for the defects of local optimization in the evolution process of genetic algorithm. In addition, this paper constructs a projection pursuit clustering model based on hill-climbing adaptive genetic algorithm, and constructs the functional module of the English text background elimination model based on actual needs. Finally, this paper designs a control experiment to verify the performance of the model. The research results show that the effect of the model constructed in this paper is good.

1. Introduction

Visual perception is the most advanced human ability to obtain information from the outside world. Obviously, images play a very important role in human perception [1]. Moreover, it is widely used in air defense early warning, weather cloud map forecasting, biological tissue movement analysis, traffic monitoring management, security video monitoring, high-definition TV band compression, content-based retrieval, etc. [2]. From the perspective of target characteristics, target detection techniques can be divided into two categories: moving target detection and single image target detection [3].

2. Related work

After solving the problem of "gradient disappearance", neural network optimization has become a popular optimization method. For those systems that are complex, uncertain, and have little information, we can establish the relationship between input and output through neural networks, and combine certain data training to use neural network algorithms for various classification problems [4]. When looking for the best path for planning, we often learn from the principle of ant foraging [5], and convert the feasible solution of the problem to be optimized into the path of ants. By increasing the amount of pheromone released by ants with shorter paths, the ant colony algorithm can complete the work of path optimization [6]. As an algorithm that simulates the search for the optimal solution in the natural environment, the genetic algorithm draws on the idea of natural selection to complete the task of finding a satisfactory solution in the variable space at a lower cost [7]. The literature [8] regarded evolutionary computation as an optimization method, and successfully applied Darwin's theory of evolution to the candidate solution population. Moreover, it proposes to simulate the process of reproduction and mutation in biological evolution through operators such as crossover and mutation in the algorithm, and through directional selection pressure, the algorithm can evolve a solution that meets the requirements after many iterations. The literature [9] proposed "evolution strategy". Other computer scientists also independently proposed
similar theories at that time, and the literature [10] proposed related concepts of genetic algorithms. The literature [11] showed how to abstract Darwin's theory of evolution into a mathematical model and use it to optimize strategies, imitate chromosome sequence coding, and combine mutation, selection, and crossover operators to enable the population in the algorithm to evolve in a certain direction. Although this blurs the boundaries between many methodologies, it provides us with a wealth of tools to make it more convenient for us to solve practical problems, and it also makes more articles about the improvement of genetic algorithms appear. Aiming at the problems of poor convergence and easy to fall into local optima in traditional genetic algorithms, the literature [12] proposed an adaptive genetic algorithm. With the continuous development of many fields such as actual industrial production, commercial development planning, scientific research, etc., the number of multi-objective optimization problems encountered is also increasing. It can be said that in almost every important practical decision-making problem, there are situations where a single goal cannot be optimized due to the constraints of many conditions. For example, the contradiction between the strict requirements for safety in the aerospace field and the consequent high cost has caused many problems in design optimization. The literature [13] proposed a method to find non-inferior solutions, which improved the original selection algorithm and calculation process, so that the efficiency of population evolution to the Pareto optimal solution was improved. Later, this idea was further developed, and a variety of multi-objective genetic algorithms that can obtain the Pareto optimal solution were produced. Among them, the most famous is the NSGA algorithm, which was proposed by literature [14]. The literature [15] further improved the NSGA algorithm in terms of computing speed and stability, resulting in the algorithm NSGA-I, and subsequent improvements to the algorithm are also endless. The literature [16] combined RBF neural network on the basis of multi-objective optimization algorithm (NSGA-II), so that the algorithm has better performance in convergence speed and convergence accuracy.

The literature [17] proposed a method of fitness sharing, and the literature [18] reduced the degree of aggregation of individuals with the same fitness, thereby maintaining population diversity. The literature [19] proposed dynamic fitness sharing, which classifies populations based on an improved genetic algorithm based on dynamic fitness sharing. Individuals of the same class are at the same peak, and fitness sharing can only be performed among individuals of the same class. This algorithm can obtain all local optimal solutions of some functions. However, it is still very difficult to effectively select the niche radius. The dual-population method proposed in the literature [20] is also one of the effective methods to maintain population diversity. This method introduces a population to maintain population diversity.

3. Genetic algorithm

It uses the roulette method to select chromosomes into the next generation population. It needs to calculate the chromosome selection probability $P_{s_i}$, that is, the ratio of the fitness value of the chromosome to the sum of the fitness values of all chromosomes. The higher the fitness value, the greater the probability of entering the next generation of chromosomes. If the population size is set to $N$ and the fitness value of the chromosome is $F_i$, the calculating formula for selection probability $P_{s_i}$ of i-th chromosome is [21]:
Moreover, the sum of the selection probabilities of all individuals is 1. The sum of the selection probabilities of the first i chromosomes is called the cumulative probability. Then, the cumulative formula for the probability $p_i$ of the i-th chromosome is:

$$P_i = \sum_{j=1}^{i} P_{s_i}, i, j = 1, 2, L, N$$

N interval segments $[0, p_1], [p_1, p_2], L, [p_{N-1}, p_N]$ are obtained by cumulative probability. From the above formula, we can find:

$$p_N = 1$$

Then, N random numbers of $[0, 1]$ are generated:

$$\{u_k | k = 1 \sim N \}$$

Each random number will correspond to one of the N intervals, if

$$u_k \in [p_{i-1}, p_i]$$

The flow chart of genetic algorithm is shown in Fig. 1 [22].

Improved genetic algorithm

Genetic algorithm is a kind of bionic algorithm that simulates biological evolution. It has global convergence and has obvious advantages compared with traditional optimization algorithms.

First, genetic algorithms are self-adaptive, self-organizing and self-learning. The genetic algorithm can automatically obtain environmental information during the evolution process, thereby adjusting the search strategy, so that individuals with higher fitness have a greater probability of surviving. The initialization is shown in Fig. 2.
Second, the genetic algorithm has low requirements on the objective function, does not need to have special requirements on the nature of the function, and does not need other information and only needs the function to be computable. Moreover, it judges the pros and cons of individuals through the value of fitness function. Moreover, the scale can be further changed, which makes the application range of genetic algorithm wider.

Third, the genetic algorithm uses the rule of probability transition to guide its search direction, and it has strong robustness.

Fourth, genetic algorithms are inherently parallel. When the genetic algorithm performs global optimization, it can search multiple places in a larger search range at the same time, which can increase the speed of the algorithm and make the search more thorough.

Fifth, genetic algorithm has a selection operator, which is a directional random search algorithm.

The calculation formula for crossover and mutation probability is:

\[
p_c = \begin{cases} 
  \frac{k_1 (F_{\text{max}} - F')}{{F_{\text{max}}} - F} & F' > \bar{F} \\
  k_2 & F' \leq \bar{F} 
\end{cases}
\]

\[
p_m = \begin{cases} 
  \frac{k_3 (F_{\text{max}} - F)}{{F_{\text{max}}} - F} & F > \bar{F} \\
  k_4 & F \leq \bar{F} 
\end{cases}
\]

In the formula, \(p_c\) represents the crossover probability, \(F'\) represents the larger fitness value of the two crossover individuals, \(\bar{F}\) represents the average fitness value of the population, \(F_{\text{max}}\) is the maximum fitness function value in the current population, \(p_m\) represents the probability of mutation, \(F\) represents the fitness value of the current individual, and \(k_1, k_2, k_3, k_4\) is constant parameters.

In 2006, Ren Ziwu and others proposed a new adaptive probability, and the calculation formula for crossover and mutation probability is:

\[
p_c = \begin{cases} 
  p_{c2} - \frac{(p_{c2} - p_{c1}) (F' - \bar{F})}{{F_{\text{max}}} - F} & F' > \bar{F} \\
  p_{c2} & F' \leq \bar{F} 
\end{cases}
\]
In the formula, \( p_c \) represents the crossover probability, \( p_m \) represents the mutation probability, \( p_{c_1} \) and \( p_{c_2} \) represent the minimum and maximum values of the crossover probability, respectively, and \( \bar{F}' \) represents the larger fitness value of the two chromosomes where the crossover occurs, \( \bar{F} \) represents the average fitness value of the population, \( F'_{\text{max}} \) is the maximum fitness function value in the current population, \( F \) represents the fitness value of the current individual, and \( p_{m_2} \) and \( p_{m_1} \) represent the minimum and maximum values of mutation probability, respectively.

### 4. Improved adaptive probability

The probabilities of the two adaptive genetic algorithms proposed in literature [23] and literature [24] have different deficiencies. The adaptive probability proposed in [25] when the individual fitness value is close to or equal to the maximum fitness value in the contemporary population, the crossover probability and mutation probability are close to zero. If this happens in the early stage of evolution and the individual is not the optimal solution, the algorithm is very easy to fall into the local optimal solution, which makes the results obtained by the genetic algorithm uncertain. The adaptive probability formula proposed by Ren Ziwu makes the probability of crossover and mutation of chromosome and its fitness value have a linear proportional relationship. If the fitness value of the chromosome in the early stage of evolution is close to the maximum fitness value in the current population, the probability value will become very small, and it is easy to end the optimization process prematurely and fall into a local optimal solution. The above-mentioned main problem is that individual excellent individuals may appear in the early stage of evolution. The above-mentioned probability selection method escorts their reproduction, leading to too fast reproduction and leading the algorithm to a local optimal solution. To solve the above problems, a new adaptive probability value method is proposed. When chromosomes are close to the highest fitness value in the contemporary population in the early stage of evolution, individuals with higher fitness values can be generated through crossover and mutation. In the later stage of evolution, since the individuals in the population will converge to the optimal individual, the chromosomes of the offspring will also converge to the optimal chromosome, which will not destroy the existing excellent gene model, can solve the problem of rapid reproduction of excellent individuals in the early stage of evolution, and ensure that the genetic algorithm obtains the global optimal solution. The new adaptive crossover and mutation probability value formula is:

\[
\begin{align*}
    p_c = \begin{cases} 
    \frac{p_{c_2} + p_{c_1}}{2} & |F'_{\text{max}} - F'| \leq \eta \\
    p_{c_2} - \frac{(p_{c_2} - p_{c_1})(F' - \bar{F})}{F'_{\text{max}} - \bar{F}} & F' > \bar{F} + \eta \\
    p_{c_2} & F' \leq \bar{F} - \eta
    \end{cases}
\end{align*}
\]
The crossover and mutation probability calculated by the new adaptive probability formula is obtained by the chromosome according to its own fitness value and the current population fitness value, and it is dynamically adjusted with the evolution process. The probability value when the fitness value of the chromosome is close to or equal to the maximum fitness value is the middle value of the probability interval. It can not only avoid the overreproduction and premature phenomenon of the better individuals in the early stage of evolution, but also have a greater probability of jumping out of the local optimal solution and continuing to search for the global optimal solution. Moreover, it can not only maintain the population diversity to a large extent, but also confirm whether the global optimal solution is obtained instead of the local optimal solution by judging whether to jump out of the obtained solution.

The hill climbing method is a local optimization algorithm. It does not have the ability to jump out of the local optimum. It is easy to fall into the local optimum solution when dealing with complex problems, and it is difficult to get the global optimum solution, so it cannot handle complex problems such as multi-peak. However, the local optimization ability of the hill climbing method is very strong, the optimization speed in the local area is fast, and the accuracy is high.

Then, the algorithm returns the optimal solution found. The calculation formula of the new search interval \([\lambda, \theta]\) is:

\[
\begin{align*}
\lambda &= \max (\lambda_1, t_1 - \omega) \\
\theta &= \min (\theta_1, t_1 + \omega)
\end{align*}
\]

In the formula, \([\lambda_1, \theta_1]\) is the original value interval of the first variable \(T\), \(t\) represents the actual value of the index, and \(\omega\) represents the neighborhood range.
In the formula, \( \lambda \) is the original value interval of the first indicator variable, \( t_1 \) is the actual value of the indicator, \( P \) is the number of indicators, \( n \) is the population size, and \( \omega \) is the neighborhood range.

After determining the new search interval for the first decision index, the algorithm performs the golden section method on the first decision index of the solution to obtain a better solution and replace the original solution with the new solution. If no better solution is found, no replacement is made. After that, the algorithm continues to perform the above operations on the second decision index of the solution until all the decision indexes are subjected to the golden section method. Finally, a better solution than the original solution is obtained to complete the local optimization task. The flowchart of adaptive genetic algorithm is shown in Fig. 3.

5. Adaptive genetic algorithm based on hill climbing method

Genetic algorithm is a classic global optimization algorithm. The global optimization ability is relatively strong, but the local search ability is weak. Although genetic algorithms are more directional than completely random search algorithms, they are less effective than their larger search ranges and large-scale samples. The hill-climbing method with better local optimization effect is used to locally optimize the individuals obtained after each evolutionary iteration of the genetic algorithm, which can make the genetic algorithm obtain better individuals during the evolution process, and improve the optimization speed and accuracy of the genetic algorithm.

By adaptively adjusting the crossover and mutation probability of genetic algorithm and optimizing through the hill-climbing method, the evolution direction of the genetic algorithm can be adjusted locally, so that the individual obtained by the genetic algorithm is better, and the local adjustment of the genetic algorithm is faster. Moreover, genetic algorithms can converge to the optimal solution faster and more accurately.

The following describes the adaptive genetic algorithm process based on the hill climbing method.

In order not to lose generality, we set the objective function as:

\[
\lambda = \max (\lambda_1, t_1 - \omega)
\]

\[
\theta = \min (\theta_1, t_1 + \omega)
\]

\[
\omega = \frac{(\theta_1 - \lambda_1) p}{2n}
\]
\[
\min f(v_1, v_2, L, v_p), \partial_j \leq v_j \leq \beta_j, j = 1, 2, L, p
\]

In the formula, \(v_j\) is the value of the \(j\)-th optimized variable, \([\partial_j, \beta_j]\) is the variation interval of \(v_j\) and \(P\) is the number of optimized variables.

**Step 1** The algorithm performs real number coding on the feasible solution of the function problem, and converts the real solution of the problem into a coding form that the algorithm can handle. The coding method is:

\[
v_{ij} = \partial_j + y_{ij} (\beta_j - \partial_j), i = 1, L, N, j = 1, L, p
\]

In the formula, \(v_{ij}\) represents the value of the \(j\)th index of the \(i\)-th chromosome, \(y_{ij}\) represents the random number on \([0, 1]\), \(N\) represents the population size, and \(P\) represents the number of indicators.

**Step 2** The algorithm randomly generates \(N\) chromosomes as the initial population \(C_0\).

Moreover, the algorithm generates a \(N \times P\) dimension random matrix \(J\) on \([0, 1]\), and the component of each row vector is the component value of the corresponding solution vector. The initial population of \(N\) chromosomes can be obtained by substituting the components of the row vector into the above formula.

**Step 3** The algorithm establishes a suitable fitness function, calculates the fitness value of the chromosomes in the population, and evaluates the chromosomes in the population. The fitness function for evaluating each chromosome is:

\[
F_i = \frac{1}{f_i^2 + 0.01}
\]

6. **Model building**

The background difference method (BS) is the most widely used algorithm for moving target detection under fixed camera shooting scenes. The background is understood as an empty scene that does not contain moving targets. The background difference method establishes a background model for each pixel, and judges whether the pixel belongs to a moving target or a background area by comparing the similarity between the pixel of the current frame to be detected and the background model of the pixel. The main process of the background difference method is shown in Fig. 4.

In a complex background situation, compared to the background model in the training codebook stage, the background in the detection stage may change at any time. If the trained background model is always
used, once the background changes, the changed area will continue to be detected as a moving target. Obviously, this detection result is obviously wrong. In order to better detect moving targets, the background needs to be updated at any time. To this end, we have implemented hierarchical modeling and detection. The main purpose is to update the background and better detect moving targets from the new complex background. Figure 5 is a flowchart of the establishment of a hierarchical codebook model.

The characteristics of the background and shadow pixels in the YCbCr color space are used to calculate the difference between the foreground image containing the shadow and the original unobstructed background image. After that, according to the statistical characteristics of the shadow difference, a corresponding method is designed to remove the shadow area of the candidate foreground area, so as to realize the shadow removal and retain the real moving foreground. The structural block diagram is shown in Fig. 6.

The pixels in the red box in Fig. 7 are the R-neighborhood pixel set at p(x,y), and the pixels marked in black are randomly selected pixels for building the background model.

### 7. Conclusion

The background difference method mostly uses a fixed-size background sample number, but the fixed-size background sample number is not suitable for the specific conditions of different pixels in the scene. If the number of background samples is too small, it will affect the detection effect of moving targets, and if the number of background samples is too large, it will increase the time and memory usage of the algorithm. Therefore, this paper proposes an adaptive number of background samples based on pixel complexity, and adjusts the number of background samples adaptively by calculating the background complexity of each pixel. The experimental results prove that when the dynamic background area obtains a large number of background samples, the static background area has a smaller number of background samples, which takes into account the time and memory consumption of the algorithm and the detection effect.

### Declarations

**Conflict of interest**

The authors declare that they have no conflict of interests.

**Ethical approval**

This article does not contain any studies with human participants performed by any of the authors.

**Data Availability**

Data will be made available on request.
References


Figures
Figure 1

Flow chart of genetic algorithm
Figure 2

Initialization
Figure 3

Flow chart of adaptive genetic algorithm
Figure 4

Flow chart of background difference method
Figure 5

Block diagram of building a hierarchical codebook model
Figure 6
Block diagram of the shadow elimination method based on chroma
Figure 7

R-Neighborhood pixels