A Genetic Algorithm-Optimized Big Data Feature Selection Method (Postprint)

Authors: Zhang Wenjie, Jiang Liehui

Date: 2018-11-29T00:00:00+00:00

Abstract

Feature selection is an important preprocessing method for large datasets that can make subsequent data analysis and processing more efficient and accurate. A large-scale data feature selection algorithm based on genetic algorithms is proposed. The algorithm first evaluates features across all dimensions, adjusting their weights according to the degree of difference each feature exhibits on same-class nearest neighbors and different-class nearest neighbors. Based on these feature weights, it guides the search process of the genetic algorithm to improve the algorithm' s search capability and the accuracy of feature acquisition. Then, it calculates feature fitness by combining the feature weights, uses fitness as the evaluation metric, initiates the genetic algorithm to obtain the optimal feature subset, and finally achieves efficient and accurate large-scale data feature selection. Experimental analysis shows that the algorithm can effectively reduce the number of classification features and improve feature classification accuracy.

Full Text

Preamble

A Genetic Algorithm-Optimized Feature Selection Method for Big Data

Zhang Wenjie[1],[2], **Jiang Liehui**[1],[2] [1]. School of Cyberspace Security, PLA Information Engineering University, Zhengzhou 450001, China; [2]. State Key Laboratory of Mathematical Engineering & Advanced Computing, Zhengzhou 450001, China

Abstract: Feature selection is a crucial preprocessing method for large datasets that enhances the efficiency and accuracy of subsequent data analysis and processing. This paper proposes a novel feature selection algorithm for big data based on genetic algorithms. The algorithm first evaluates features across all dimensions, adjusting each feature' s weight according to its discriminative power

between similar nearest neighbors (intra-class) and dissimilar nearest neighbors (inter-class). These feature weights guide the genetic algorithm' s search process to improve both search capability and feature acquisition accuracy. Subsequently, the algorithm computes feature fitness by incorporating the feature weights and uses this fitness metric to drive the genetic algorithm in obtaining an optimal feature subset, ultimately achieving efficient and accurate big data feature selection. Experimental analysis demonstrates that the proposed algorithm effectively reduces the number of classification features while improving classification accuracy.

Keywords: big data; feature selection; genetic algorithm; feature subset

0 Introduction

With the rapid development of internet communication, data storage, and information processing technologies, both data volume and dimensionality continue to grow exponentially. Large datasets often contain thousands of feature dimensions, with massive amounts of data harboring substantial redundant and invalid features that severely impact and limit the performance of big data analysis and mining [1,2]. To address these challenges, feature selection eliminates redundant information from big datasets to extract representative feature subsets, thereby reducing data scale and dimensionality while enhancing analysis efficiency. In recent years, as big data analytics and processing technologies have advanced, feature selection methods have attracted widespread research attention and been extensively applied to big data clustering, text classification, multimedia analysis, and numerous other domains [3,4].

Feature selection serves three primary functions: (a) by selecting partial feature data from large collections, it significantly reduces the scale of data requiring analysis and processing, thereby decreasing computational complexity for subsequent big data analytics; (b) it removes large amounts of irrelevant or redundant information, making big data easier to understand and interpret while facilitating later-stage processing; and (c) it effectively reduces dataset dimensionality, overcoming the limitations imposed by massive dimensions on big data mining and consequently improving the accuracy and effectiveness of machine learning methods. Beyond reducing storage requirements and computational overhead, feature selection reveals hidden structural patterns and regularities within big datasets, providing important impetus for subsequent mining and analysis.

Current feature selection methods primarily fall into three categories: wrapper methods, embedded methods, and filter methods [5]. Embedded methods integrate filter and wrapper approaches, substantially reducing computation time, but they concentrate search within limited local spaces, resulting in restricted coverage. Reference [6] proposed a feature selection method based on an improved multi-objective artificial bee colony algorithm, transforming the big data feature selection problem into a multi-objective optimization problem to enhance efficiency. Reference [7] introduced a new criterion for measuring

feature subset discernibility that, unlike previous approaches considering only individual feature impacts, simultaneously incorporates all features to compute their collective effect on discernibility metrics, using support vector machines as the classification tool to guide the selection process. Reference [8] presented an improved feature selection optimization algorithm based on artificial bee colony that reduces feature count and computational load while improving efficiency and accuracy. Reference [9] proposed a multi-criteria fusion-based feature selection algorithm that, diverging from traditional single-criterion quantization, introduces multiple criteria simultaneously to enhance feature subset diversity and algorithmic search capability. However, most existing feature selection methods focus predominantly on individual feature importance, often oversimplifying importance assessment and neglecting correlations between different features and how these correlations affect feature significance, thereby degrading overall big data feature selection performance.

To achieve efficient feature selection, this paper proposes a heuristic feature selection algorithm based on genetic algorithms. The algorithm first evaluates each dimensional feature, adjusting its weight according to differences observed on similar nearest neighbors (intra-class) and dissimilar nearest neighbors (inter-class). It then combines these feature weights to calculate feature fitness, using fitness as the evaluation metric to drive the genetic algorithm in obtaining an optimal feature subset, ultimately achieving efficient and accurate big data feature selection.

1 Research Background Overview

Feature selection involves selecting an m-dimensional feature subset S from a complete big dataset D based on appropriate strategies, then applying this subset to subsequent data analysis and processing. During big data feature selection, two attribute types are generally considered unnecessary: (a) attributes unrelated to target data, and (b) redundant attributes relative to target data. Minimizing these unnecessary attributes requires dataset reduction through feature selection—a process of selecting attribute subsets that identifies important attributes, removes irrelevant or redundant ones, and obtains refined data. Feature selection finds extensive and deep applications in data mining, machine learning, and other fields, representing a crucial preprocessing method in big data analysis and processing.

Recent studies [10,11] have found that genetic algorithms (GA) are particularly suitable for big data feature selection problems. GA is a stochastic search method that operates directly on processing objects without restrictions such as complex differentiability, differentiability, or continuity. The algorithm is not constrained by fixed rules during iteration and can autonomously adjust search direction based on selection probability, demonstrating broad adaptability and powerful global search capabilities. In big data environments where data space dimensionality is high and internal characteristics remain unknown, genetic algorithms can obtain optimized feature extraction results through heuristic self-

learning during each iteration.

Most existing feature selection algorithms employ single evaluation criteria, failing to adequately consider weight differences between similar and dissimilar features, which prevents effective guidance of the genetic operator search process. This limitation causes blind variation in genetic operators for big data feature selection, restricting overall algorithmic performance. To address this, we propose a heuristic feature selection algorithm based on genetic algorithms that first computes the discriminative difference of each feature on similar and dissimilar nearest neighbors to comprehensively adjust its weight. It then combines feature weights to calculate feature fitness, using this fitness to guide genetic algorithm mutation and search, thereby improving search performance and ultimately achieving efficient and accurate big data feature selection.

2 Genetic Algorithm-Based Big Data Feature Selection Algorithm

2.1 Algorithm Architecture

Feature selection is a critical preprocessing step for big data that effectively eliminates redundant attributes, improves efficiency of subsequent processing, and enhances big data analysis performance. The essence of feature selection involves searching iteratively to obtain the most representative feature subset from big data, evaluating its importance according to assessment criteria before performing iterative selection until an optimal subset is obtained.

As shown in [Figure 1: see original paper], the iterative process of feature selection primarily comprises three important steps: feature evaluation, subset generation, and stopping criteria. Given big data' s characteristics of large volume and high dimensionality, this paper adopts a heuristic feature selection method based on genetic algorithms. The approach first evaluates feature weights by comprehensively considering each feature' s similar nearest neighbors (intra-class) and dissimilar nearest neighbors (inter-class), then computes feature fitness using these weights to guide genetic algorithm search, thereby improving feature selection accuracy in big data environments.

2.2 Feature Weight Evaluation

Current big data research [10,12] has revealed that in large datasets, data items belonging to the same class and located close to each other share similar data characteristics, while nearby data items from different classes exhibit significant feature differences. Based on this observation, the feature weight evaluation algorithm proceeds as follows: randomly select a data item x_i from big dataset D, search for its k similar nearest neighbors $x_r(x_i)$ and m dissimilar nearest neighbors $x_h(x_i)$, compute the difference values between each dimensional feature and the similar nearest neighbors, and between each dimensional feature and the dissimilar nearest neighbors, then adjust feature weights accordingly.

Through repeated iteration, the top M features with highest weight values are selected to form a new feature subset.

Let $diff(x_i, x_j, f)$ denote the difference between data items x_i and x_j on feature f. For data items x_i and x_j, their difference on feature dimension f is defined as:

$$diff(x_i, x_j, f) = \frac{|x_i^f - x_j^f|}{\max(f) - \min(f)}$$

During each iteration, the weight ω_j for feature f_j is adjusted based on the differences between data item x_i and its k similar nearest neighbors $x_r(x_i)$ and m dissimilar nearest neighbors $x_h(x_i)$:

$$\omega_j \leftarrow \omega_j - \frac{\sum_{h=1}^{M} diff(x_i, x_h(x_i), f_j)}{M} + \frac{\sum_{r=1}^{M} diff(x_i, x_r(x_i), f_j)}{M}$$

The specific description of the feature weight evaluation algorithm is as follows:

Input: Big dataset D, iteration count t, subset dimension M.

Output: Feature weights ω_j (where $j = 1, \ldots, M$).

1. Initialize feature weights $\omega_j = 0$ in big dataset D.
2. For $i = 1$ to t:
    3. Randomly obtain a data item x_i.
    4. Search for k similar nearest neighbors $x_r(x_i)$ and m dissimilar nearest neighbors $x_h(x_i)$ of x_i.
    5. For $j = 1$ to M:
    6. Update feature weight ω_j using the formula above.
    7. End for.
    8. End for.

The feature weight evaluation algorithm compares each sample data item with its k similar nearest neighbors and m dissimilar nearest neighbors, adjusting weights according to differences in relevant feature dimensions. Smaller intra-class differences and larger inter-class differences indicate more representative feature dimensions, leading to weight increases; conversely, larger intra-class differences and smaller inter-class differences indicate less representative dimensions, resulting in weight decreases. Compared with general feature selection algorithms, this approach comprehensively considers correlations between features and their similar/dissimilar nearest neighbors across dimensions, enabling feature weights to more objectively reflect dimension representativeness and improving both the performance and robustness of subsequent genetic algorithm-based feature search and selection.

2.3 Genetic Algorithm-Based Feature Selection Method

This paper employs a heuristic feature selection method based on genetic algorithms. The approach first evaluates feature weights by comprehensively considering each feature' s similar and dissimilar nearest neighbors, then computes feature fitness using these weights to guide the genetic algorithm' s feature search, thereby improving feature selection accuracy in big data environments. The specific procedure is as follows:

a) Randomly generate initial population P_0 with population size N. Encode and initialize the solution space.

b) Calculate the fitness of all individuals in generation t according to the predefined fitness function.

c) Comprehensively compare inter-class and intra-class distances of feature subsets, using the ratio of inter-class distance to intra-class distance as the fitness function:

$$f(x_i) = \frac{\frac{1}{c}\sum_{i=1}^{c} ||\bar{x} - \bar{x}_i||}{\frac{1}{c}\sum_{i=1}^{c} \frac{1}{n_i}\sum_{j=1}^{n_i} ||x_{ij} - \bar{x}_i||}$$

where $\bar{x}$ represents the mean vector of the feature subset in the big dataset, $\bar{x}_i$ represents the mean vector of the feature subset in class i, x_{ij} represents the j-th sample vector of class i, n_i represents the number of samples in class i, and c represents the number of classes. Larger inter-class distance and smaller intra-class distance indicate higher fitness of the feature subset; conversely, smaller inter-class distance and larger intra-class distance indicate lower fitness.

d) Estimate feature selection probability by comprehensively considering feature weights obtained from the weight evaluation algorithm and their fitness values. The probability of selecting individual x_i from the previous generation population P_t for the next iteration is:

$$p(x_i) = \frac{f(x_i) \cdot \omega(x_i)}{\sum_{i=1}^{N} f(x_i) \cdot \omega(x_i)}$$

e) Select two individuals from the updated population P_{t+1} with equal probability, perform chromosome crossover and recombination with probability P_c. Simultaneously, mutate certain gene loci of individuals with probability P_m to obtain a new generation population P_{t+2}. Identify the individual x^*_{t+2} with the highest fitness value in P_{t+2}.

f) Compare the fitness value of x^*_{t+2}. If it exceeds the relevant fitness threshold or the maximum iteration count has been reached, terminate the process and output x^*_{t+2}; otherwise, set $t = t + 1$ and return to step b).

g) Select the top M features ranked by fitness from big dataset D to form the feature subset.

3 Experimental Results and Analysis

The experimental data primarily utilizes standard UCI databases, selecting 10 representative datasets as test data. Detailed dataset information is provided in . In our UCI dataset [13] experiments, the k value is set to 10. For each test, one dataset is selected as the independent test set while the remaining nine serve as training data. These 10 datasets contain multi-class classification problems with sample sizes ranging from 150 to 569 and feature dimensions from 4 to 255. The diverse data types and characteristics provide broad representativeness for comprehensively and effectively measuring and comparing the performance metrics of various feature selection algorithms. The experimental environment consists of a Lenovo M9620T desktop computer with an Intel(R) Core(TM) i3-3240 3.39 GHz CPU, 4.0 GB RAM, Windows 7 64-bit operating system, and MATLAB R2010b software.

To comprehensively compare our algorithm' s performance with similar feature selection methods, experiments compare it against the GA_{SVM} algorithm [14] (genetic algorithm-based) and the ReliefF algorithm [15] (traditional feature selection). GA_{SVM} and ReliefF represent well-established algorithms in their respective domains. presents the classification accuracy comparison between our algorithm and the genetic algorithm-based method. In the experiments, the iteration count is generally determined empirically or through multiple trials. Ten repeated calculations estimate the classification accuracy of both methods on UCI datasets, with results expressed as mean percentage $\pm$ standard deviation.

As shown in , among the 10 datasets, the GA_{SVM} algorithm and our algorithm achieve identical classification accuracy on the Iris dataset due to its small number of features and classes. On the remaining nine datasets, our algorithm outperforms GA_{SVM}, with classification accuracy improving to varying degrees. Additionally, only on the Dermatology dataset does our algorithm exhibit a slightly higher standard deviation than GA_{SVM}; on all other nine datasets, our algorithm' s standard deviation is lower, demonstrating superior classification accuracy and stability compared to GA_{SVM}.

compares classification accuracy between our algorithm and the traditional ReliefF algorithm. Similarly, ten repeated calculations estimate both methods' classification accuracy on UCI datasets, with results expressed as mean percentage $\pm$ standard deviation for both accuracy and selected feature count. As shown in , our algorithm demonstrates more significant performance improvements over ReliefF in terms of classification accuracy and selected feature count. Across all 10 datasets, our algorithm achieves accuracy improvements of varying magnitudes while maintaining a smaller mean number of selected features. Although our algorithm' s classification accuracy standard deviation is slightly higher than ReliefF' s on the Iris dataset, it is lower on all remaining datasets. Similarly, while the standard deviation of selected features is slightly higher on the Dermatology dataset, it is lower on all other datasets. These results confirm

that our algorithm surpasses ReliefF in classification accuracy, stability, and achieves the highest accuracy with the fewest selected features.

Overall, our feature selection algorithm can efficiently and accurately obtain big data feature subsets while significantly reducing computational complexity for subsequent big data analysis and processing.

4 Conclusion

To address current challenges of dimensionality explosion and high computational complexity in big data, this paper proposes a genetic algorithm-based feature selection algorithm that eliminates redundant features in big datasets and improves feature subset selection accuracy. Experiments on 10 UCI datasets demonstrate that compared with other feature selection algorithms, our approach effectively enhances big data classification accuracy while reducing the number of features in selected subsets. The algorithm successfully decreases storage requirements and computational overhead, and also reveals potential structural patterns hidden within big datasets, providing important support for subsequent big data mining and analysis.

References

[1] Zhou Qi. Research on feature selection and feature learning algorithm [D]. Hefei: University of Science and Technology of China, 2017.

[2] Zhang Junbo. Research on efficient feature selection and learning algorithms for big data [D]. Chengdu: Southwest Jiaotong University, 2015.

[3] Li Jun. Research on feature selection and classification based on intelligent optimization algorithms [D]. Wuhan: Wuhan University, 2014.

[4] Lyu Hui. Research and design of clustering method based on big data and High-dimensional data [D]. Kunming: Yunnan University, 2015.

[5] Wang Xiang, Hu Xuegang. Overview on feature selection in high-dimensional and small-sample-size classification [J]. Journal of Computer Applications, 2017, 37 (9): 2433-2438.

[6] Chao Xiuqin, Li Wei. A feature selection method optimizedby artificial bee colony algorithm [J]. Journal of Frontiers of Computer Science and Technology, 2018, 16 (1): 71-78.

[7] Xie Juanying, Xie Weixin. Several feature selection algorithms based on the discernibility of a feature subset and SupportVector machines [J]. Chinese Journal of Computers, 2014, 37 (8): 1704-1718.

[8] Hancer E, Xue Bing, Zhang Mengjie, et al. Pareto front feature selection based on artificial bee colony optimization [J]. Information Sciences, 2018, 422 (1): 462-479.

[9] Guan Xiaoyin, Chen Guo, Lin Tong. Feature selectionmethod based on differential evolution and genetic algorithm with multi-criteria evaluation and its applications [J]. Acta Aeronautica et Astronauticas Sinica, 2016, 37 (11): 3455-3465.

[10] Zhao Rongzhen; Li Kunjie. Fault feature selection method based on within-class and among-class criterion and genetic algorithm [J]. Journal of Lanzhou University of Technology, 2017, 43 (2): 35-39.

[11] Wang Na. Study on hybrid feature selection method based on genetic algorithm [D]. Xi' an: Shaanxi Normal University, 2015.

[12] Chen Lei. Text representation model and feature selection algorithm [D]. Hefei: University of Science and Technology of China, 2017.

[13] Murphy P M, Aha D W. UCI repository of machine learning database [DB/OL]. (2006-05-12). http://www.ics.uci.edu/mlearn/MLRepository.html.

[14] Huang C, Wang C. A GA-based feature selection and parameters optimization for support vector machines [J]. Expert Systems with Applications, 2016, 31 (2): 231-240.

[15] Wu Jiehua. Complex network link classification based on RReliefF feature selection algorithm [J]. Computer Engineering, 2017, 43 (8): 208-214.

Note: Figure translations are in progress. See original paper for figures.