ISSN: 2414-1895 DOI: 10.6919/ICJE.202107_7(7).0070

Research on Machine Learning Algorithms in Mobile Terminal Data Mining

Haoran Wang

North China Electric Power University (Baoding), Baoding, 071000, China.

Abstract

The machine learning algorithm is characterized by artificial intelligence technology, which can automatically find the parameters and modes required by the operation after training and learning a large number of sample sets. Machine learning algorithms are widely used in mobile terminal data mining. Because of this research background, the paper takes a large class of improved attribute reduction algorithms in machine learning algorithms as an example to delete redundant attributes in conditional attributes, reduce the amount of data to be processed in data mining, and improve data mining results for simplicity, the efficiency of mobile terminal data mining is improved. The data mining experiment is implemented in the MATLAB environment, and the test data of the algorithm comes from the UCI data set. By comparing the two algorithms before and after the improvement, it is confirmed that the improved machine learning algorithm has a significant improvement in the attribute reduction efficiency and the running time of the algorithm compared with the previous algorithm. The experiment proved that the terminal data mining model formed by the machine learning algorithm could be applied to various data mining systems.

Keywords

Mobile Terminal; Data Mining; Machine Learning Algorithm; Attribute Reduction Data Mining Algorithm.

1. Introduction

Mining real-time data information from big data to assist decision-making is of great significance to all walks of life. Essentially it is based on machine learning algorithms and models, such as logistic regression, support vector machines, etc., which can be solved by iterative algorithms such as gradient descent. Unlike traditional data mining applications, extensive data mining requires the correctness of the calculation results to be within a specific expected range. It has high requirements for the speed of data processing and the real-time performance of the products, so it needs to use clusters for parallel execution. The iterative algorithm improves calculation efficiency.

Literature [1] believes that the essence of cloud computing resource allocation is through a kind of dynamic feedback in time series, which specific regression algorithms can model, but the disadvantage is that it can only be suitable for small-scale data. Literature [2] proposes A data mining method based on cloud computing technology is proposed: large data sets and mining tasks are decomposed into multiple computers for parallel processing. After transforming the classic apriorism algorithm into MapReduce, a similar data mining platform based on the Hadoop open-source framework is established. Literature [3] proposes the use of data centralized bit segmentation and redundant data fragment merging methods to design decision trees and build cloud platform data feature mining the model uses KD tree for data mining index and combines fragment incorporating technology to make a data mining cloud platform model. The reduction based on rough set theory includes attribute reduction and attribute value reduction. Many scholars have done a lot of discussion

ISSN: 2414-1895 DOI: 10.6919/ICJE.202107_7(7).0070

and research on the attribute reduction of decision tables using wild set theory. In the literature, Professor Skowron and others proposed a discernibility function method based on the discernibility matrix in attribute reduction. Many subsequent discussions and researches on attribute reduction and its expansion are based on this method. This paper considers the incompatibility of the decision table. It reduces the time complexity and space complexity, and an improved machine learning algorithm for attribute value reduction based on rough set theory is proposed. By applying it to terminal data mining, the result analysis shows that the improved algorithm has higher efficiency.

2. The mobile terminal data mining process

The process of data mining is mainly composed of three parts, namely data collation, data mining, and interpretation and evaluation of results [4]. The detailed process is divided into data preparation, data selection, data preprocessing, data transformation, mining target determination, algorithm selection, data mining, model interpretation, and knowledge evaluation. The process is shown in Figure 1.

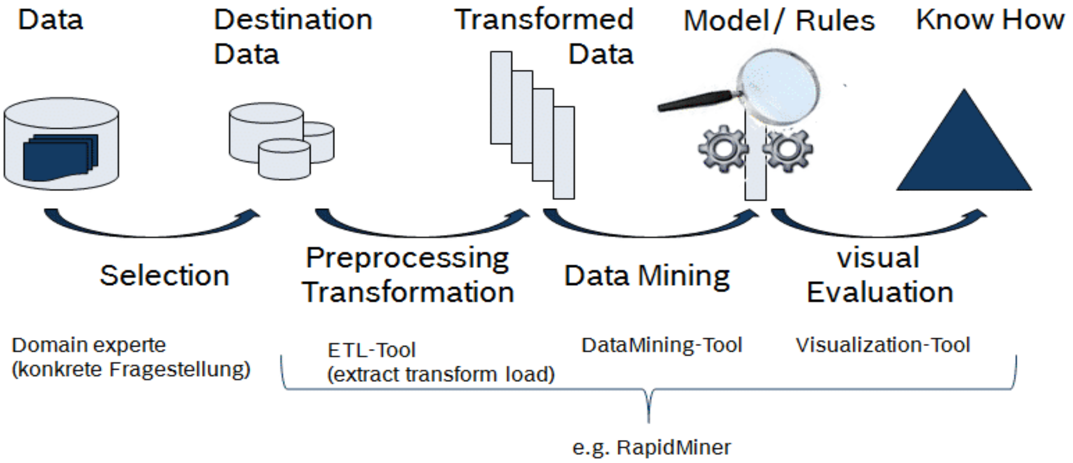

Figure 1. The process of data mining

3. Improved machine learning algorithm based on attribute value reduction of rough set theory

3.1 Heuristic algorithm based on attribute importance

A given information system is a four-tuple $S = (U, C \cup D, V, f)$, which is a sample set of as given network connection. $V = \cup V_r$ is the set of feature value ranges, where V_r E is the value range of feature r. $f: U \times R \to y$ is an information function, which specifies the value of each feature of each object in U. $Core_D(C)$ represents the set of all non-omitting relations of D in C.

Definition 1: Given an information system $S = (U, C \cup D, V, f)$, the improved discrimination matrix $M = (m_{ij})$ of S is defined as:

$$m_{ij} = \{a \in C : a(x_i) \neq a(x_j)\} \quad D(x_i) \neq D(x_j) \tag{1}$$

$$m_{ij} = \emptyset \quad D(x_i) = D(x_j) \tag{2}$$

Where $a(x)$ is the value of tuple x on attribute a, and $D(x_i)$ is the value on decision attribute D.

Definition 2: We suppose that the attribute in the information system $S = (U, C \cup D, V, f)$ has $V_i = \{V_{1i}, V_{2i}, \cdots, V_{ki}\}$ k different attribute values, then the attribute importance function of the attribute is:

$$f(a_i) = V_{1i}V_{2i} + V_{2i}V_{3i} + \cdots + V_{(k-2)i}V_{(k-1)i} + V_{(k-1)i}V_{ki} \tag{3}$$

ISSN: 2414-1895
DOI: 10.6919/ICJE.202107_7(7).0070

When the heuristic algorithm based on attribute importance uses discrimination matrix reduction, it is usually divided into the following 3 steps:

The first step is to find the discrimination matrix; the second step is to find the core, that is, to combine the elements of the discrimination matrix that contain only one attribute; the last is to find the reduction, and the non-core attributes are sorted in descending order of the importance of the attributes. The definition of importance is determined by the value of attribute importance function $f(a_i)$ given in Definition 2. The larger the value, the more important the attribute [5]. Extract the most important attribute and add it to the core, and delete all nodes that include this attribute. Continue processing in this way until the discrimination matrix is empty, and the set obtained at this time is the final demand. The specific algorithm is as follows:

Step 1: Find the discrimination matrix M according to Definition 1.

Step 2: Find the core $Core_D(C)$ of C relative to D. Suppose $Core_D(C) = \psi$, sort the attribute combinations in M according to the number of attributes from small to large. Query each element in M. If it is a single attribute combination, the set of elements of that attribute is what you want.

Step 3: Set $R = Core_D(C)$. Calculate the attribute importance function $f(a_i)$ according to definition 2, and add the attribute with the largest function value to R. And delete all elements in M that contain this attribute. Take the attribute corresponding to the largest function value in turn until M is ψ, at which time R is the final desired value [6].

3.2 Harmony Search Algorithm

The paper compares instrument $i(=1,2,\cdots,m)$ to the i variable in the optimization problem. The pitch of each instrument is equivalent to the value of each variable. The harmony of each instrument's tones is equivalent to the solution vector of the group j of the optimization problem. The evaluation of music effects is analogous to the objective function. The algorithm flow of Harmony Search Algorithm is shown in Figure 2. The calculation steps of harmony search are as follows:

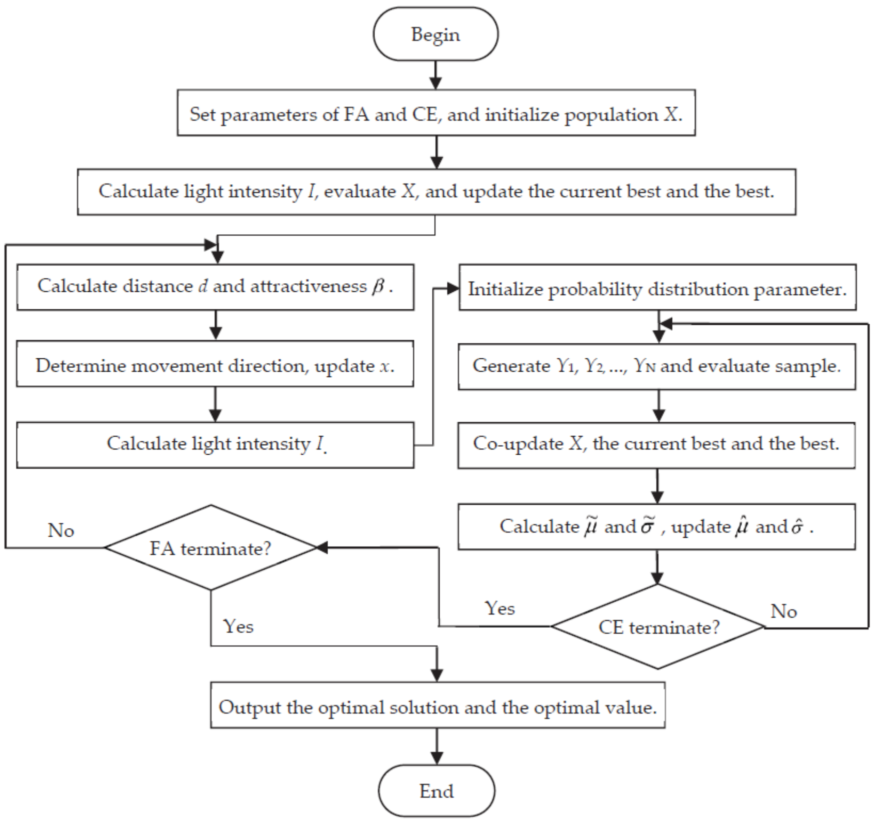

Figure 2. The algorithm flow of Harmony Search Algorithm

ISSN: 2414-1895 DOI: 10.6919/ICJE.202107_7(7).0070

Step1: Define the problem and parameter value

Assuming that the problem is minimization, the form is as follows.$min\, f\ (x)$

$$s.t.\quad x_i \in X_i, i = 1,2,\cdots,N \tag{4}$$

Step2: Initialize the harmony memory

Randomly generate HMS harmony voices $x^1, x^2, \cdots, x^{HMS}$ and put them into the harmony memory bank, where the harmony memory bank can be compared to the population in the genetic algorithm. The form of the harmony memory is as follows:

$$HM = \left[\begin{array}{c|c} x^1 & f(x^1) \\ x^2 & f(x^2) \\ \vdots & \vdots \\ x^{HMS} & f(x^{HMS}) \end{array}\right] = \left[\begin{array}{cccc|c} x_1^1 & x_2^1 & \cdots & x_N^1 & f(x^1) \\ x_1^2 & x_1^1 & \cdots & x_N^2 & f(x^2) \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_N^{HMS} & f(x^{HMS}) \end{array}\right] \tag{5}$$

Step3: Generate a new harmony

A new harmony $x_i{}' = (x_1{}', x_2{}', \cdots, x_N{}')$ is generated. Each tone $x_i{}' = (i = 1,2,\cdots,N)$ of the new harmony is generated through the following three mechanisms: learning harmony memory, fine tuning of the pitch, and random selection of the pitch.

$$x_i{}' = \begin{cases} x_i{}' \in (x_i{}^1, x_i{}^2, \cdots, x_i{}^{HMS}), & if \quad rand < HMCR \\ x_i{}' \in X_i\ , \quad otherwise\,; & \qquad i = 1,2,\cdots,N \end{cases} \tag{6}$$

Where $rand$ represents a uniformly distributed random number on $[0,1]$. Secondly, if the new harmony $x_i{}'$ comes from the harmony memory bank HM, it needs to be fine-tuned as follows:

$$x_i{}' = \begin{cases} x_i{}' + rand1 * bw\ , \quad if\ rand < HMCR \\ x_i{}'(k+m),\ m \in \{-1,1\}, if\ rand < PAR \\ x_i{}', \quad otherwise\,; \qquad i = 1,2,\cdots,N \end{cases} \tag{7}$$

Among them, bw is the pitch fine-tuning bandwidth, PAR is the pitch fine-tuning probability; $rand1$ represents a uniformly distributed random number on $[0,1]$.

3.3 Relief numerical feature extraction algorithm

The correlation between the feature and the category in the algorithm is based on the feature's close sample the ability to distinguish. This algorithm proposes a hypothesis interval, which refers to the maximum distance that the classification decision plane can move while keeping the sample classification unchanged. It can be expressed as:

$$\theta = \frac{1}{2}(||x - M(x)|| - ||\ x - H(x)||) \tag{8}$$

Among them, $H(x)$ and $M(x)$ represent the same and heterogeneous sample sets respectively. By calculating the hypothesis interval, the optimal characteristics of the classification can be estimated.

Call for a subset.

The implementation process is as follows:

(1) Establish model $Relief(S, m, N, k)$, where S is the training sample set, m is the feature dimension, N is the number of iterations, and k is the number of samples in homogeneous and heterogeneous spaces. The weight vector w represents the weight corresponding to each attribute, and the initial value is 0.

(2) Starting from $i = 1\ , i \in [1, N]$, randomly select an instance sample from it each time, and find the nearest neighbour sample instance of k that is the same as the class it belongs to, called near Hits (abbreviated as H) and the nearest neighbour sample instance of k that is different from the class it belongs to, called It is near Misses (abbreviated as M).

(3) For each cycle, starting from $j = 1\ , j \in [1, m]$, the weight of the attribute is continuously updated according to the following formula.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202107_7(7).0070

$$w[j] = w[j] - \frac{1}{k}\sum^{k} diff\left(A_j, R_i, h_p\right) + \frac{1}{k}\sum^{k} diff\left(A_j, R_i, m_p\right)$$

$$diff(A_j, R_i, h_p) = \frac{|value(A_j,R_i) - value(A_j,h_p)|}{max(A_j) - min(A_j)} \tag{9}$$

Among them, A_j represents the j attribute, R_i represents the randomly selected instance each time, and h_p and m_p represent the p element in the H and M space. $diff(A_j, R_i, h_p)$ and $diff(A_j, R_i, m_p)$ represent the difference between the sample instance R_i in the attribute A_j and H and the p instance in M, respectively [7]. If the attribute value is discrete, when the attribute values are different, $diff(A_j, R_i, h_p)$ and $diff(A_j, R_i, m_p)$ take the value 1, and when the attribute values are the same, the value is 0.

4. Terminal mobile data attribute reduction machine learning algorithm experimental design

The experimental data used in this article is the data of KDD99. The experimental results are shown in Table 1, where Red1 is the number of attribute reductions of the HORAFA algorithm; Red2 is the number of attribute reductions of the improved algorithm HORAFA-AFVDM; Time1 is the running time of the HORAFA algorithm; Time2 is the running time of the enhanced algorithm HORAFA-AFVDM; h is the hour; m is minute; s is second. It can be seen from Table 1 that except for tic-tac-toe, vehicle, and glass, the number of attribute reductions of the improved algorithm is smaller than that of the previous algorithm [8]. The reduced ability of the HORAFA-AFVDM algorithm is generally more robust than that of the HORAFA algorithm. From the perspective of the algorithm's running time, the running time of the improved algorithm has been improved to varying degrees. The HORAFA-AFVDM algorithm has been dramatically improved from the experimental results in terms of algorithm reduction ability and running time.

Table 1. Comparison of attribute reduction and running time between HORAFA algorithm and HORAFA-AFVDM algorithm

Data set	solar	tic - tac - toe	vehicle	zoo	ctr
Number of instances	333	201	150	101	21
Conditional attributes	10	9	18	16	9
Red1	9	1	2	7	7
Red2	7	1	2	5	4
Time1	6m 0.078s	3m 22.469s	3m 7.853s	41.75s	0.07s
Time2	4m 1.765s	1m 2.109s	1m 73.506s	18.703s	0.006s
Data set	flag	glass	auto_mpg	heart	house_votes
Number of instances	214	232	650	66	226
Conditional attributes	27	9	7	16	16
Red1	14	2	3	6	16
Red2	6	2	2	4	8
Time1	13m 26.56s	11m 25.678s	1h10m 57.054s	8.83s	7m 24.497s
Time2	9m 40.375s	3m 17.893s	24m 3.456s	1.894s	3m 8.656s

5. Conclusion

Based on the background of mobile terminal big data mining, the paper proposes an improved machine learning algorithm for attribute value reduction based on rough set theory. An example analysis shows that the algorithm improves the efficiency of data mining in mobile terminals and has been fully verified in the system.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202107_7(7).0070

References

[1] Riihijarvi, J., & Mahonen, P. Machine learning for performance prediction in mobile cellular networks. IEEE Computational Intelligence Magazine, 13(1) (2018) 51-60.

[2] Dash, S. K., Dash, S., Mishra, J., & Mishra, S. Opportunistic mobile data offloading using machine learning approach. Wireless Personal Communications, 110(1) (2020) 125-139.

[3] Feng, B., Fu, Q., Dong, M., Guo, D., & Li, Q. Multistage and elastic spam detection in mobile social networks through deep learning. IEEE Network, 32(4) (2018) 15-21.

[4] Anandakumar, H., & Umamaheswari, K. Supervised machine learning techniques in cognitive radio networks during cooperative spectrum handovers. Cluster Computing, 20(2) (2017) 1505-1515.

[5] Hou, R., Kong, Y., Cai, B., & Liu, H. Unstructured big data analysis algorithm and simulation of Internet of Things based on machine learning. Neural Computing and Applications, 32(10) (2020) 5399-5407.

[6] Jian, C., Yang, M., & Zhang, M. Mobile terminal trajectory recognition based on improved LSTM model. IET Image Processing, 13(11) (2019) 1914-1921.

[7] Wang, F., Jiang, D., Wen, H., & Song, H. Adaboost-based security level classification of mobile intelligent terminals. The Journal of Supercomputing, 75(11) (2019) 7460-7478.

[8] Yu, L., Wu, X., & Yang, Y. An online education data classification model based on Tr_ MAdaBoost algorithm. Chinese Journal of Electronics, 28(1) (2019) 21-28.