A Discretization Algorithm for Meteorological Data and its Parallelization Based on Hadoop

Chao Liu\textsuperscript{a}, Wen Jin, Yuting Yu, Taorong Qiu, Xiaoming Bai and Shuilong Zou\textsuperscript{b}

School of Electrical and Information Engineering, Nanchang Institute of Science & Technology, Nanchang, China
Email: \textsuperscript{a}mekisstherain@163.com, \textsuperscript{b}392748648@163.com

Abstract. In view of the large amount of meteorological observation data, the property is more and the attribute values are continuous values, the correlation between the elements is the need for the application of meteorological data, this paper is devoted to solving the problem of how to better discretize large meteorological data to more effectively dig out the hidden knowledge in meteorological data and research on the improvement of discretization algorithm for large scale data, in order to achieve data in the large meteorological data discretization for the follow-up to better provide knowledge to provide protection, a discretization algorithm based on information entropy and inconsistency of meteorological attributes is proposed and the algorithm is parallelized under Hadoop platform. Finally, the comparison test validates the effectiveness of the proposed algorithm for discretization in the area of meteorological large data.

1. Introduction
Data discretization as a means of data preprocessing is also increasingly showing its irreplaceable role in the field of data mining. In losts of areas such as rule extraction, rough set and granular calculation, feature classification that continuous attribute data must be discretized before it can be used. In the meteorological field, the state collects large amounts of data in real time every day. Those data are not only large in volume, but also have many attributes and with continuous values and the data are not complete. Obviously, the discretization of data for such large data needs to face more challenges[1]. Some of the classic data mining methods, such as rough set[2] and other methods to extract the features of the extraction or association rules need to discretization first, and some discretization algorithms based on machine learning, such as clustering[3]. Therefore, it is very important to choose an efficient discretization algorithm. At present, different fields of research scholars have proposed a variety of different discretization algorithm, according to the classification of different standards, data discretization algorithm can be divided into: supervision[4] discrete algorithm and unsupervised[5] discrete algorithm such as equal frequency[6], equal width[7] discrete algorithm. Today's more commonly used discretization algorithms such as chi-square discretization algorithms Chimerge[8]. It is a self-low discrete algorithm. This algorithm introduces the statistical theory into the discretization algorithm. Due to the need to set the threshold, the subjective impact threshold is too high will lead to excessive discretization, too low will lead to insufficient discretization. Later, Liu et al improved the formation of the CHi2[9] algorithm. The algorithm uses a gradual decline in the level of significance, and ultimately can make more intervals merges but Chi2 algorithm processing speed slow in the face of large data, also not good as top-down algorithm. Since then, the algorithm can improve the ability to deal with large samples by changing the way of the breakpoint measurement, but the processing time also increased when the sample is discretized in the face of the large scale data.
volume. And this algorithm still requires the maximum threshold of the set value inconsistency and the case where the threshold is too high leads to the excessive discretization or too low leads to insufficient discretization.

This paper is based on the discrete algorithm proposed in paper[10]. On the one hand, the setting of the attribute inconsistency threshold is based on the manual method. In this paper, we improve the subjective influence of setting the attribute inconsistency threshold by improving the adjacency interval condition. On the other hand, consider the actual data set, especially in the field of meteorological data sets, the importance of each attribute in the decision-making class is not the same, this paper combines the importance of the attribute to determine whether the need to merge adjacent intervals. The improved method is designed to be able to effectively reduce the number of discrete points to improve the efficiency of the algorithm. At the same time, considering the actual environment, the need for discrete data sets are large data, so this article uses Hadoop platform to improve the method of parallel implementation.

2. Related Theory

2.1 Collection Information Entropy

There is a decision information system\([11]\) \( S = (U, AT, V, F) \), \( V = C \cup \{d\} \), \( C \) is a set of conditional attributes, \( \{d\} \) is decision attribute set. The number of objects with the decision property \( j(j = 1, 2, \ldots, r(d)) \) is \( k \) in set \( x \), then the information entropy \( H(X) \) of the set \( x \) is defined as:

\[
H(X) = - \sum_{j=1}^{r(d)} p_j \log_2 p_j, p_j = \frac{k}{|X|}
\]  

Among them, \( 0 \leq H(X) \leq 1 \). Information entroy reflects the set \( x \) confusion, the smaller the information entropy is, the more object of decision attribute value in \( x \) and the smaller the confusion of the set, and \( H(X) = 0 \) when there is only one decision attribute value in \( x \).

2.2 Breakpoint Importance Definition

According to the formula (1) can be defined as follows: in decision information system \( S = (U, AT, V, f) \), \( P \) is a set of breakpoint on attribute \( a \), \( L = \{Y_1, Y_2, \ldots, Y_m\} \) are the number of \( m \) subset of \( U \) that is divided by \( P \), and the domain \( U \) is defined according to the information entropy of the set of breakpoint \( P \):

\[
H'(U) = \frac{|Y_1|}{|U|}H(Y_1) + \ldots + \frac{|Y_m|}{|U|}H(Y_m)
\]  

After add a new breakpoint \( c \notin P \), the information entropy of breakpoint set \( P + \{c\} \) can be defined as:

\[
H'(U) = \frac{|Y_1|}{|U|}H(Y_1) + \ldots + \frac{|Y_i|}{|U|}H(Y_i) + \ldots + \frac{|Y_{i'}|}{|U|}H(Y_{i'}) + \ldots + \frac{|Y_{m'}|}{|U|}H(Y_{m'})
\]

\( Y_i, Y_i' \) are subsets which are devided by breakpoint \( c \). Add a breakpoint \( c \notin P \) to \( P \), then the information entropy of the breakpoint \( c \) can be defined as:

\[
H(c) = H'(U) - H'_{i-1}(U)
\]

The value of the breakpoint information entropy reflects the importance of the breakpoint. The smaller the value of \( H(c) \), the smaller the decision attribute value of the equivalence class subset divided by
the new breakpoint set, the decrease in the degree of confusion, the greater the $H(c)$ the more important breakpoint $c$ is.

2.3 Inconsistency Introduction

In decision information system $S = (U, A^T, V, f)$, $P$ is a set of breakpoint on attribute $a$, $L = \{X_1, X_2, ..., X_m\}$ are the number of $m$ subset of $U$ that is divided by $P$, and $x \cup x_1 \cup ... \cup x_n = U$, the number of objects whose decision attribute value is $j (j = 1, 2, ..., r(d))$ is $k$, $M_x$ is the class label of the largest decision class in the set $X$, the $x$ inconsistency $\xi_x$ is defined as:

$$\xi_x = \frac{|X| - |M_x|}{|X|}$$

(4)

$|M_x| = \max(k)$ reflects the degree of inconsistency of the decision category in the $x$ set, $0 \leq \xi_x \leq 1$, $\xi_x$ is smaller, the set $x$ is the more consistent.

2.4 MapReduce Programming Model

MapReduce[12] is a distributed computing framework model was originally introduced by Google engineers Jeffery Dean and Sanjay Ghemawat in 2004. It's divide complex problems running on large clusters into two function, map and reduce function. The map function can be understood as the size of the decomposition problem, which aims to divide a complex large-scale data set into several blocks that can be processed by each individual node in the cluster and analyze the subset of the data into a set of <$K_1, V_2$> pairs, then it's processed by framework will be merged, sorted and generated the intermediate results <$K_2, V_2$> pairs, and all the same $K_2$ value of the $V_2$ set together as a reduce function input, and after computed by reduced function would got the final result set <$K_3, V_3$> pairs, the process shown in Fig 1.

![MapReduce diagram](image)

**Figure 1** MapReduce calculation mode.

3. Related Theory Algorithm and Improvement Based on Information Entropy and Inconsistency

3.1 An Algorithm Based on Information Entropy and Inconsistency

The algorithm is mainly divided into three steps: the first step is to select candidate discrete points; the second step is to select discrete points according to the information entropy; the third step is to merge the interval according to the inconsistency rate and the importance of the attributes.

3.2 Improved Discretization Algorithm Based on Information Entropy and Inconsistency

The set of breakpoints generated by step2 of the above algorithm divides the continuous condition attribute values into finite value intervals, and merges the adjacent intervals satisfying the given...
combining condition by step3 to reduce the number of discrete points. However, there are two shortcomings in the interval merging method used in step3 above. One is that it is necessary to set the interval inconsistency threshold manually. This setting is subjective and difficult to determine. By testing in practical meteorological data set, we found that the meteorological data after noise pretreatment is not so by setting the $\xi_{\infty}$ to control the artificial noise seem a bit superfluous and the interval combination of discrete points decrease was not obvious; the second is the actual field data from different attributes influence on decision attribute or importance is not entirely consistent, but the algorithm does not consider the merger in the interval. In view of the above shortcomings, this paper improves the interval merging condition by deleting the inconsistency rate threshold and introducing the importance of attributes.

Based on attribute importance to merge interval is under this point: when a condition attribute in the decision table of decision attribute dependence is stronger, the important degree is higher, the number of discretization points or intervals on the properties change little or almost no change; and when a condition attribute in decision table the decision attribute dependence becomes weak, that is the degree of importance becomes smaller, the number of discretization points or the number of intervals on the attribute changes and more.

In the above algorithm, we have the condition that the discretization interval on the conditional property $a \in C$ can be merged: $\xi_{\infty} \leq \xi_{\infty}$, and $\xi_{\infty} \leq \xi_{\infty}$, $\xi_{\infty} \cup \xi_{\infty} \leq \xi_{\infty}$ and $\xi_{\infty} \cup \xi_{\infty} \leq \xi_{\infty}$, change it to: $\xi_{\infty} \cup \xi_{\infty} \leq (\xi_{\infty} + \xi_{\infty})*(1-a)$, where $a$ is the importance of the conditional attribute. Since the study object of this paper is the element in the meteorological field, the importance of the meteorological element is set by the paper[13] when the specific meteorological element is discretized. When the importance of a property is very small, the formula $\xi_{\infty} \cup \xi_{\infty} \leq (\xi_{\infty} + \xi_{\infty})*(1-a)$ increase the probability of establishment, then the possibility of merging the two adjacent areas also increased. And when it comes to the importance of a property, $\xi_{\infty} \cup \xi_{\infty} \leq (\xi_{\infty} + \xi_{\infty})*(1-a)$, the probability of establishment is reduced, so the possibility of merging between adjacent two sections is also reduced. And the type of conditions for the same attribute in different locations adjacent to the interval can be combined with the above-mentioned algorithm is also dynamic changes. Therefore, the improved interval combination judgment not only takes into account the different importance of attribute, but also meets the different practical requirements. At the same time, the modified merge condition solves the problem of setting the inconsistency threshold $\xi_{\infty}$, so that the improved algorithm has less discrete effect caused by the parameter influence. The following is a detailed description of the third step to improve the algorithm:

**Step3.** Interval merge based on attribute importance and inconsistency

**Input:** based on information entropy breakpoint set $s'$

**Output:** based on the inconsistency of the combined breakpoint set $p'$

**Algorithm description:**

1. The equivalence class $r'$ is divided according to the breakpoint, and the reference[13] is used to determine the importance of the condition attribute $a$.

2. For $p, q \in r'$, $p, q$ adjacent
   (2.1) according to the formula (4), respectively calculate $p, q$ class inconsistency icRateP, icRateQ.
   (2.2) merge $p, q$ and assignment set to $z$
   (2.3) according to the formula (4) calculate the inconsistency of $z$ (icRateZ)
   (2.4) if $cRateZ<=(icRateP+icRateQ)*(1-a)$, then perform the step (2.5), otherwise continue
   (2.5) delete the breakpoint between $p, q$ classes from $s'$
   (2.6) merge $p, q$ and assignment set top, and get next equivalence class as $q$

End for
4. Parallel Design Of Discrete Algorithm

Hadoop parallelization of the discrete process is divided into the following two steps correspond to the two jobs; Job1 is mainly on the data set of preprocessing, generate a formatted sample. Job2 is the format of the sample input, using the above 3.1 in step one, step two and improved step three of the algorithm for discretization. Details as follows:

**Step1:** Input: Complete decision table \( S = (U, AT = C \cup D, V, f) \)
Output: Candidate formatted sample collection \( B^c \)
Parallel description:
(1) Map read decision table system \( S \), attribute name \( \text{attId} \), attribute value \( \text{attValue} \), decision attribute value \( \text{conValue} \)
(2)Map formatted output: key=\( \text{attId}+\text{attValue}+\text{conValue} \),value=1
(3) Reduce read form (2) the Map output
(4) Reduce calculates the number of the same key and assigns it to sumcount
(5)Reduce output: key=\( \text{attId}+\text{attValue}+\text{conValue} \),value=sumcount

**Step2:** Input: the Step1’s Reduce output
Output: The collection of breakpoints for each attribute \( B^c \)
Parallel description:
(1) Map read from the output of step1 (each line is formatted as Value=\( \text{attId}+\text{attValue}+\text{conValue}+\text{sumcount} \) )
output after formatted: key=\( \text{attId} \) (attribute name),
value=\( \text{attValue}+\text{conValue}+\text{sumcount} \)
(2) Map output: key=\( \text{attId} \),value=\( \text{attValue}+\text{conValue}+\text{sumcount} \)
(3) Reduce input: Reduce read (2) the Map output
(4) Reduce calculation: according to the steps of algorithm 3.1, to obtain a candidate breakpoint collection \( B^c \), information entropy breakpoint collection \( s^c \), attribute importance degree inconsistency merge breakpoint collection \( p^c \).
(5) Reduce output: key=\( \text{attId} \), Value= \( p^c \)

5. Algorithmic Test and Result Analysis

5.1. Test Data and Environment Introduce
The experimental data are derived from the daily value data of chinese climatological ground data provided by the China Meteorological Data Sharing Network, and the data of a ground site in Jinxi Province from January 1960 to December 2016 are selected. At the same time, according to the paper[13] to provide the importance of meteorological factors on the choice of rainfall and rainfall correlation coefficient of 11 elements as a conditional attribute. And use the rainfall at 20~20 as a decision attribute. Because of the large amount of data, some incomplete data is deleted directly, and finally 1800000 data sets of 1.4G size are formed. The training set of different training sample sizes is extracted from the data set to form the four data sets as shown in Table 1.

| Table 1 Experimental use of the data set |
|-----------------------------------------|
| dataset | object numbers | size of dataset |
| D_Set1  | 225000         | 179MB          |
| D_Set2  | 450000         | 358MB          |
| D_Set3  | 900000         | 716MB          |
| D_Set4  | 1800000        | 1.4GB          |

This test environment is Sugon server I620-G20, with 2 Xeon E5-2630CPU, each CPU is 8-core 16 threads, the machine's memory is 128G, hard disk size of 300G, using KVM technology to manage the virtual 8 virtual machines, assigned to each virtual machine resources are 2-core 4 threads CPU, 12G...
memory, 20G hard disk, and the operating system adopts Ubuntu16.04 server version, each virtual machine are equipped with Hadoop2.7.2, as a Hadoop nodes, a total of 8 Hadoop nodes, one of which has a separate namenode node responsible for storing the meta information of the files in the distributed system is not the actual computing node, the other seven nodes are the datanode nodes, in the distributed file system datanode node is the actual storage node and also the actual calculate node of algorithm program.

5.2. Test and Result Analysis
The classification algorithm used in the data mining is a classic classification algorithm (KNN classification algorithm), using this algorithm test the data set D_Set1 which processed by discretization. Then we divide D_Set1 into four group data set, each of which is a 1/4-size test set, a 2/4-size test set, a 3/4-size test set, and a test set itself. Each data set randomly selects 20% data as the test set, and selects 80% data as the training set. Using KNN algorithm, in k=5 case, cross validation is used to calculate the prediction accuracy, and the test is taken ten times as the final prediction accuracy data. The test results are shown in table 2.

Table 2 Predictive classification results

| indicators dataset | KNN algorithm DIEIC | Improved DIEIC |
|--------------------|---------------------|----------------|
| D_Set1_1           | 66.48               | 67.34          |
| D_Set1_1           | 66.33               | 67.42          |
| D_Set1_1           | 66.35               | 67.30          |
| D_Set1             | 66.44               | 67.21          |

Compared with the DIEIC algorithm, the improved average prediction accuracy is improved by 0.92%. Although the improved algorithm improve accuracy is not obvious, the main reason is that in the second step algorithm based on information entropy for discrete breakpoint is completed preliminary breakpoint extraction, thus proves the validity of access breakpoint based on information entropy, and the improved DIEIC algorithm is also improved a little discrete effect of instructions on adding attribute importance factor is compared with the discrete demand, it also can effectively reduce the number of discrete points

6. Concluding remarks
The discretization algorithm has a direct impact on the quality of data mining or knowledge discovery. In general, the importance of dependency of different conditional attributes and decision values is different in the decision table. It is one of the keys to improve the discretization effect when considering the importance of the attribute in designing the discretization algorithm. Based on the characteristics of the data set in the meteorological field involved in the research, this paper improves the algorithm based on the information discretization algorithm on information entropy and inconsistency. The improvement of the algorithm is mainly to improve the existing problems of discretization interval merging. The specific improvement includes eliminating the parameters of the original algorithm that need to manually set the inconsistency threshold, and consider different meteorological elements with different decision value dependencies and in the discretization interval combine the dynamic inequality when merging. In addition, for the needs of large meteorological data applications, the improved data discretization algorithm based on attribute importance and inconsistency is processed and implemented in parallel. Improved discretization algorithm and algorithm parallelization in the real meteorological data set under the comparative test and the results of the analysis to verify the effectiveness of the improved algorithm and the feasibility of parallelization. This has a reference value of and knowledge discovery applications for the meteorological large data mining and a certain guiding significance.
7. References

[1] Ramírez - Gallego S, García S, Mourio - Talín H, et al. Data discretization: taxonomy and big data challenge[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2016, 6(1): 5-21.

[2] Jiang F, Sui Y. A novel approach for discretization of continuous attributes in rough set theory[J]. Knowledge-Based Systems, 2015, 73: 324-334.

[3] Mincheng C, Jingling Y, Xiaoyan W. A Discrete Random Forest Parallel Classification Algorithm Based on Weakly Correlated Feature Subspace Selection[J]. Computer Science, 2016, 43(6) : 55 – 58. (In Chinese)

[4] Huang W, Pan Y, Wu J. Supervised discretization for optimal prediction[J]. Procedia Computer Science, 2014, 30: 75-80.

[5] Nguyen H V, Müller E, Vreeken J, et al. Unsupervised interaction-preserving discretization of multivariate data[J]. Data Mining and Knowledge Discovery, 2014, 28(5-6): 1366-1397.

[6] Noor Suhana S, Rohani A B. Rough Set Discretization: Equal Frequency Binning, Entropy/MDL and Semi Naïves Algorithms of Intrusion Detection System[J]. arXiv, 2016, 282: 77-87.

[7] Jiawei Han, Micheline kamber. data mining concepts and techniques[M] :Beijing: Machinery Industry Press, 2004. (In Chinese)

[8] Kerber R. Chimere: Discretization of numeric attributes[C]//Proceedings of the tenth national conference on Artificial intelligence. Aaai Press, 1992: 123-128.

[9] Liu H, Setiono R. Feature selection via discretization[J]. IEEE Transactions on knowledge and Data Engineering, 1997, 9(4): 642-645.

[10] Hailiang Y, Deqin Y. A Discretization Algorithm for ontinuous Attributes of Decision Tables Based on Information Theory[J]. Computer Science, 2010, 37(4) : 231 – 233. (In Chinese)

[11] Duolian M, Daoguo L.rough set theory algorithm and application[M].Tsinghua University Press, 2008. (In Chinese)

[12] Li F, Ooi B C, Özsu M T, et al. Distributed data management using MapReduce[J]. ACM Computing Surveys (CSUR), 2014, 46(3): 31. (In Chinese)

[13] Chenyang Z, Zhiqiang M, Limin L. Researchon Meteorological Data Mining Based on Rough Set and Bayesian under Hadoop[J]. computer application and software, 2015, 32(4) : 72 – 76. (In Chinese)