CrossMark

Research on time series data mining algorithm based on Bayesian node incremental decision tree

Sun Xingrong[1]

Abstract Aiming at the shortage of classic ID3 decision tree and C4.5 decision tree algorithm in ability of time series data mining, this paper increases Bayesian classification algorithm in the node of the decision tree algorithm as a preprocessing Bayesian node to form incremental learning decision tree. In incremental learning, the decision tree is more powerful in the mining of time series data, and the result is more robust and accurate. In order to verify the effectiveness of the algorithm, time series data mining simulation based on UCR time series data sets have been conducted, and the performance and efficiency of ID3 decision tree, Bayesian algorithm and incremental decision tree algorithm have been contrasted respectively in non-incremental learning and incremental learning. The experimental results show that in both the incremental and non-incremental time series data mining, the incremental decision tree algorithm based on Bayesian nodes optimization which can improve the classification accuracy. When optimizing the parameters of Bayesian nodes, the consumption of pruning time will be decreased, and the efficiency of the data mining is greatly improved.

Keywords Decision tree · Incremental learning · Bayesian · Time series data · Data mining

1 Introduction

In the era of data explosion, data mining technology can analyze and dig out the patterns and rules that are meaningful to users from a large number of data re-use patterns and rules to classify or predict new samples. In many data mining algorithms, decision tree algorithm is one of the most widely used algorithms [1–4]. Decision tree algorithm is one of the most basic classification and regression methods, given by the tree structure, usually the leaves are expressed as classification or regression results, the tree is used for example classification or regression characteristics. A tree structure can usually be seen as an if-then form of rule-determining set. Probably speaking, the decision tree model can also be seen as a conditional probability distribution defined in the feature space or class space [5–8]. Based on the actual situation, the model conducts classification and regression according to the actual features and attributes, and has good readability, small space and fast classification speed. Under normal circumstances, the decision tree optimization strategy is based on the principle of minimizing the loss function. Firstly, the original decision tree is trained by training data and optimization strategy, including decision tree generation and pruning. After the decision tree is established, each data record is classified and returned by the decision tree. Figure 1 shows the basic decision tree model

So far, for the shortcomings of traditional decision trees, many scholars have proposed a lot of optimization and improvement. Some of the scholars have optimized and improved the pruning of the decision tree algorithm [9–11], which simplifies the cumbersome pruning process. The simplified decision tree has almost the same decision result as the C4.5 algorithm, but it has not been improved in time complexity. In addition, some scholars have applied other data mining algorithms to the decision tree algorithm. For example, the association rule mining and the decision tree algorithm are fused [12–14]. By preprocessing the association rules and using it for the construction process, the consumption of the process is reduced. As the lesser branches, the pruning pro-

✉ Sun Xingrong
   caoxuefeng11@163.com

[1] Huanggang Normal University, Huanggang, Hubei Province, China

Fig. 1 Basic decision tree model
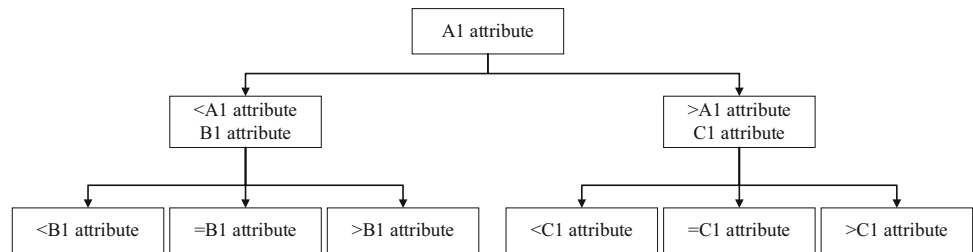
cess has also been greatly improved. Based on the fuzzy set to improve the decision tree has also achieved good results [15–17]. Fuzzy set can simplify the redundant attributes and features, so as to reduce the complexity of the data in the process of building, fundamentally reduce the data complexity of the contribution and pruning, and improve the efficiency of the decision tree.

Compared to the original decision tree algorithm, the incremental decision tree algorithm is divided into two parts, the first part is through the sample training set to generate the initial decision tree; the second part is the use of Bayesian nodes for incremental learning. In the incremental decision tree algorithm, each time the new training sample is obtained, the algorithm will match the incremental data with the attributes in the current decision tree continuously until when it reaches a leaf node of the decision tree.

If the node is not a Bayesian node, it is necessary to judge whether the division is correct. If the result is correct, keep the decision tree changed, but the decision tree classification and Bayesian classification is used to compare the incremental data The If the accuracy of the Bayesian classification method is higher than that of the decision tree classification method, the node is transformed into a Bayesian node. If the node is a Bayesian node, the Bayesian parameter needs to be updated in conjunction with the incremental data. The incremental decision tree can be constructed by continuous recursion, which can finish increment learning and solve the problem of incremental data by transforming the leaf node as the Bayesian node or updating the parameters of the node. It can be seen from the algorithm flow that the Bayesian classification is added to the node of decision tree to predict the node value, which makes the feature more clearly in the classification of decision tree, and the classification effect of decision tree is better.

In recent years, most of the scholars have designed many improved strategies for the defects of the decision tree algorithm and combined with other algorithms to improve the time complexity and spatial complexity of the traditional algorithms. However, there is no relevant algorithm to improve the decision tree algorithm in the case of the increment of time series data. With the development of the times, the data mining algorithm is applied to more and more scenes. Time series is a common data, for example, voice data, and financial data. And the time series data are often

incremented with time, the traditional decision tree algorithm has limited ability in classification, regression and prediction for this kind of data [18–24]. Therefore, this paper improves the incremental time-series data processing defects of the traditional decision tree algorithm by using the incremental learning characteristic of Bayesian algorithm, and UCR time series data sets are used to carry out simulation experiments. The experimental results show that the incremental decision tree algorithm proposed in this paper has good practical performance, and has good application effect for incremental data.

2 Decision tree algorithm

2.1 Information entropy and information gain

In the process of constructing the decision tree, the training sequence is input into the decision tree in a recursive way. If it is input into the decision tree in the recursively order, the construction efficiency is relatively slow, and it is difficult to obtain the optimal decision tree. In general, the input data preprocessing is adopted by some feature selection methods, and some attributes that do not contribute to the construction decision tree are eliminated. This makes it possible to construct the decision tree with maximum efficiency, and the decision tree tends to be optimized. Feature selection is given by information entropy and information gain. Information entropy can represent the uncertainty of random variables. By defining the probability of random variables, we can give the information entropy of random variables [25], as shown in equation (1):

$$
\begin{aligned}
&P(X = x_i) = p_i, i = 1, 2, ..., n \\
&H(X) = -\sum_{i=1}^{n} p_i \log p_i \\
&H(X|Y) = \sum_{i=1}^{n} p_i H(X|Y = y_i)
\end{aligned}
\tag{1}
$$

wherein, X is the random variable and p_i is the probability of the random variable, $H(X)$ represents the information entropy of the random variable, and $H(X|Y)$ represents the information entropy of the random variable X under a given condition Y. From the definition we can see that the greater the information entropy, the greater the uncertainty of random variables. We see the feature sequence of the decision tree to be constructed as a set of random variables. If the

entropy of a random variable is smaller, its uncertainty is smaller and the stable feature is more suitable as the practical feature of the construction decision tree. In the process of constructing the decision tree, we set the current decision tree as D and when the information entropy of the new feature A to the decision tree is the smallest, the difference between the entropy $H(D)$ of the current decision tree D and the conditional entropy $H(D|A)$ of the decision tree D under the characteristic A condition can represent the influence of a feature on the information entropy, called information gain [9].The following formula (2) gives the expression:

$$g(D, A) = H(D) - H(D|A) \tag{2}$$

wherein $g(D, A)$ the information gain and that is is the degree of uncertainty in the classification of the decision tree D due to the addition of the feature A. Logically speaking, for the decision tree D, the information gain is closely related to the new addition feature, the greater the information gain, the stronger the classification ability of the decision tree D is enhanced by the feature. The decision tree could be constructed by selecting the appropriate features through information gain, which can not only reduce the complexity of decision tree, but also reduce the time complexity.

2.2 ID3 algorithm and C4.5 algorithm

There are a variety of methods for constructing decision trees, the most commonly used is the ID3 algorithm proposed by Quinlan [26,27], the main feature of this algorithm is to select the feature with larger information gain to build the decision tree. The specific process includes starting from the root node, calculating all the information gain of all the features to be input to the current decision tree, and selecting the feature with the highest information gain as the candidate feature. The decision tree branches are constructed from all the attribute values of the feature, and the decision tree is constructed by the feature with the largest information gain currently through recursive invocation until the information gain is no longer increased or no feature is available. In fact, since the information gain value is calculated for the training feature set, there is no absolute meaning. When the information entropy of the training feature set is too large and the classification is not clear, the information gain value is too large, and there is no obvious meaning for the actual feature selection, And finally the classification ability of decision tree built by ID3 algorithm is weak. In order to solve this problem, the C4.5 algorithm uses the information gain ratio to construct the decision tree [11], and uses the information gain ratio to correct the practical problems caused by the excessive information entropy of the training feature set. The following equation (3) gives the calculation method of the information gain ratio:

$$g_R(D, A) = \frac{g(D, A)}{H(D)} \tag{3}$$

The information gain ratio is defined as the ratio of the information gain $g(D, A)$ to the information entropy of the current decision tree $H(D)$, which can normalize the information gain, thereby solving the problem that the information gain is not significant due to the excessive information entropy. C4.5 algorithm uses the information gain ratio to complete the recursive construction of the decision tree, the decision tree is more stable, and the construction of the decision tree can be completed for different types of features and attribute values [28].

2.3 Pruning of decision trees

Decision tree algorithm uses information gain or information gain to recursively construct decision tree. Normally, the characteristics of the whole training set can be classified accurately, but the characteristics of non-training set are often difficult to obtain accurate classification results due to over-fitting. The over-fitting phenomenon in the decision tree refers to the high complexity of the decision tree due to the excessive feature selection of the training set. To solve the over-fitting problem, it is necessary to prune and simplify the decision tree. The objective of the decision tree pruning is to minimize the objective function of the decision tree [29]. The smaller the optimization value of the objective function, more data the decision tree corresponding to the objective function can accurately classify.

Assuming that the number of leaf nodes in a decision tree is N and t is a leaf node, the number of samples on the leaf node is N_t. The number of samples of k is N_{tk}, $H_t(T)$ represents the information entropy on the leaf node t, we can use the information Entropy to define the actual loss function of the decision tree:

$$C_\alpha(T) = \sum_{t=1}^{N} N_t H_t(T) + \alpha N$$
$$H_t(T) = -\sum_k \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t} \tag{4}$$

Among them, $\alpha \geq 0$ is for the structural parameters, and is an important parameter in the pruning process. Conduct conversion according to the following formula (5) parameter:

$$C(T) = \sum_{t=1}^{N} N_t H_t(T) = -\sum_{t=1}^{N} \sum_{k=1}^{K} N_{tk} \log \frac{N_{tk}}{N_t}$$
$$C_\alpha(T) = C(T) + \alpha N \tag{5}$$

Among them, $C(T)$ is the prediction error of the model to training characteristics data, the fitting degree of the model to training characteristic. In order to prevent the degree of fitting of the training feature is too large, αN shows the degree of fitting of the model to other test features, N represents the complexity degree of the model, and the relationship between

the two parameters is controlled through α. Larger parameters α can reduce the complexity of the decision tree, and vice versa. The pruning process is the process of determining the parameter α, that is, selecting the subtree with the smallest loss of the actual loss function. The larger the sub-tree is, the better the fitting of the training data, the higher the complexity degree of the model; the smaller the subtree, the better the fitting of the test data, the lower the complexity degree of the model. The pruning process can greatly reduce the complexity of the model so that it can obtain a reasonable degree of fitting for all data within the maximum model complexity. The true meaning of pruning is to use the maximum likelihood estimation to conduct model selection, and select the appropriate subtree as the data mining model.

3 Incremental decision tree algorithm with the processing of Bayesian nodes

Traditional ID3 and C4.5 decision tree algorithm is suitable for offline machine learning. When there is a machine learning scene on the line, there will be increasing data. For time series data, this type of data will increase over time, the classification, regression and prediction are a huge challenge. In order to solve the shortcomings of the traditional algorithm for incremental time series data, the author preprocesses the incremental data through the Bayesian classification method, and then using the decision tree algorithm for classification or regression, it can solve the data incremental learning problems during the process of data mining classification and regression [30].

3.1 Bayesian classification method

Bayesian classification method is a fast, effective and practical statistical classification method, which uses the theory of Bayesian to build the classification process. Firstly, calculation of the prior probability and posterior probability by Bayesian method is [13]:

$$P(D|A) = \frac{P(A|D)P(D)}{P(A)} \tag{6}$$

where $P(D)$ is the prior probability, and $P(A|D)$ is the conditional probability that can be observed by A when the condition D is satisfied, $P(D|A)$ is the posterior probability that D is assumed to be true under the condition A. According to the Bayesian probability definition, the posterior probability $P(D|A)$ changes with the prior probabilities and conditional probabilities, and the variables or conditions A, D are treated as independent data or function dependent data, then use priori probability and conditional probability

to predict the posterior probability, and thus complete the classification of data.

The above Bayesian classification method is applicable to discrete random variables or discrete data feature sets. For a continuous random variable or data feature set, we can assume that the random variable obeys the Gaussian distribution [14], using the continuous function of the Gaussian distribution to complete the posterior probability calculation:

$$P(D|A) = g(D, \mu_A, \sigma_A) = \frac{1}{\sqrt{2\pi}\sigma_A} e^{\frac{(D-\mu_A)^2}{2\sigma_A^2}} \tag{7}$$

Among them, $g(D, \mu_A, \sigma_A)$ is the Gaussian function which belongs to the continuous function, and the Bayesian classification of the data feature set can be completed by mean μ_A and variance σ_A.

3.2 Incremental decision tree algorithm

The application of traditional data mining algorithms has a large limitation in the case of increasing the amount of data, especially when the time series data is obtained segmentally, then the incremental learning method is needed to solve the problem of data increment by using priori information. The traditional decision tree algorithm is a classification algorithm based on the data feature, and adopts the top-down recursive construction method. By comparing the attribution values through branch comparison, the classification results are obtained at the leaf nodes of the decision tree by contrasting downwards. In this paper, Bayesian algorithm and decision tree algorithm are combined to solve the problem of incremental learning.

According to the increasing training set sample [15], the incremental decision tree algorithm first uses the decision tree algorithm to classify the samples, and uses the recursive method to complete the segmentation of multiple small samples according to the attribute-value. After constructing the nodes of the decision tree, the actual situations build Bayesian nodes and ordinary leaf nodes.

In the process of generating leaf nodes, the algorithm is divided into two cases:

(1) When all the samples are divided into the same category attribute, the excessive matching usually leads to the fitting phenomenon. In this case, the pre-pruning method is needed: conduct metric computation and threshold segmentation to the difference value of the node before the partition. When the difference is greater than Threshold, it can be further segmented, otherwise the node generates leaf nodes. The method of calculating the difference value is shown in the following equation (8):

$$n_A = Max(n_i), n = \sum_i n_i \tag{8}$$

(a)

```
┌─────────────────────────────┐
│        Create node M        │◄──┐
└─────────────────────────────┘   │
              │                   │ N
      ╱───────────────╲           │
    ╱ All samples in    ╲─────────┘
    ╲ feature set S belong
      ╲ to category B  ╱
      ╲───────────────╱
              │ Y
┌─────────────────────────────┐
│ Return to M and marks it as │
│ a normal node and as        │
│ category B.                 │
└─────────────────────────────┘
              │
┌─────────────────────────────┐
│ Calculate the difference β  │◄──┐
│ among the current nodes     │   │
└─────────────────────────────┘   │
              │                   │ N
      ╱───────────────╲           │
    ╱ P Has no longer    ╲────────┘
    ╲ available attribute-value
      pair, or the difference β
      is less than the threshold
      ╲───────────────╱
              │ Y
┌─────────────────────────────┐
│ Learn the naive Bayesian    │
│ parameters, return to M and │
│ mark it as Bayesian nodes   │
└─────────────────────────────┘
              │
┌─────────────────────────────┐
│ For the attribute-value     │
│ pair A with minimum         │
│ distribution expectation ,  │
│ find its training subset S1 │
└─────────────────────────────┘
              │
┌─────────────────────────────┐
│ Add the return node to the  │
│ current decision tree so    │
│ that S2 = S-S1, then        │
│ continue recursion taking   │
│ S2 as the root node of the  │
│ subtree                     │
└─────────────────────────────┘
```

(b)

```
┌─────────────────────────────┐
│ When the incremental data a │
│ matches the attribute of    │
│ decision tree T0 to a leaf  │
│ node P                      │
└─────────────────────────────┘
              │
      ╱───────────────╲        Y    ┌──────────────────────────────┐
    ╱ Node P is a       ╲──────────►│ Update the parameters of the │
    ╲ Bayesian node    ╱            │ Bayesian node and return the │
      ╲───────────────╱             │ updated decision tree T      │
              │ N                   └──────────────────────────────┘
      ╱───────────────╲        Y    ┌──────────────────────────────┐
    ╱ The classification ╲─────────►│      Return subtree T0       │
    ╲ results of the feature       └──────────────────────────────┘
      a are the same as that
      of the node
      ╲───────────────╱
              │ N
┌─────────────────────────────┐
│ Compare Bias classification │
│ accuracy A1 and decision    │
│ tree classifier accuracy A2 │
└─────────────────────────────┘
              │
      ╱───────────────╲        Y    ┌──────────────────────────────┐
    ╱     A1>A2         ╲──────────►│      Return subtree T0       │
      ╲───────────────╱             └──────────────────────────────┘
              │ N
┌─────────────────────────────┐
│ Update the P node as a      │
│ Bayesian node and return    │
│ the updated decision tree T │
└─────────────────────────────┘
```
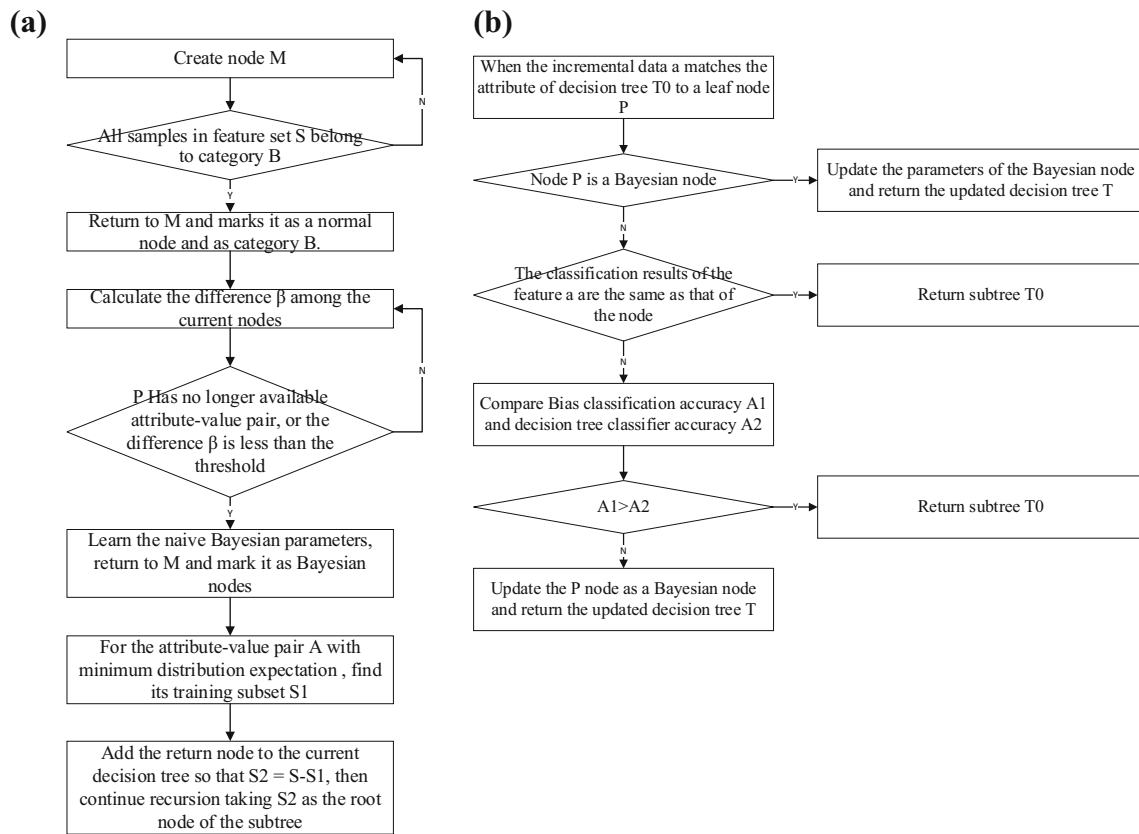
Fig. 2 Construction process of initial decision tree and incremental decision tree algorithm. **a** Initial decision tree construction process **b** Incremental decision tree construction process

Assuming that a total of n nodes data needs to be divided into the current node, and there are n_i data divided into category i, the most likely classification n_A can be calculated from the above formula, which could calculate the difference between the current node:

$$dif = \frac{1 - n_A}{n} \qquad (9)$$

(2) When there are incremental samples divided into the current node, the data can not be further divided, and the extreme case is that the spatial discrete properties of all the incremental samples are the same. In this case, the incremental decision tree algorithm generates a Bayesian node and uses the Bayesian posteriori probability algorithm to complete the most valuable attribute classification of the incremental data.

Incremental decision tree algorithm is divided into two parts, the first part is through the sample training set to generate the initial decision tree; the second part is the use of Bayesian nodes for incremental learning. In the incremental decision tree algorithm, each time the new training sample is obtained, the algorithm will match the incremental data with the attributes in the current decision tree continuously until when it reaches a leaf node of the decision tree.

If the node is not a Bayesian node, it is necessary to judge whether the division is correct. If the result is correct, keep the decision tree changed, but the decision tree classification and Bayesian classification is used to compare the incremental data The If the accuracy of the Bayesian classification method is higher than that of the decision tree classification method, the node is transformed into a Bayesian node. If the node is a Bayesian node, the Bayesian parameter needs to be updated in conjunction with the incremental data. The incremental decision tree can be constructed by continuous recursion, which can finish increment learning and solve the problem of incremental data by transforming the leaf node as the Bayesian node or updating the parameters of the node. Figure 2 shows the comparison between the traditional decision tree algorithm and the optimized incremental decision tree algorithm. It can be seen from the algorithm flow that the Bayesian classification is added to the node of decision tree to predict the node value, which makes the feature more clearly in the classification of decision tree, and the classification effect of decision tree is better.

Table 1 Description of UCR time series data sample set

Name	Number of categories	Training set size	Test set size	Time series length
Gun-Point	2	50	150	150
Lightning-2	2	60	61	637
Synthetic control	6	300	300	60
Trace	4	100	100	275
OSU leaf	6	200	242	427
50Words	50	450	455	270
Swedish leaf	15	500	625	128
Two patterns	4	1000	4000	128
Wafer	2	1000	6174	152
Yoga	2	300	3000	426

In addition, Bayesian classification improves the resolvability of features, reduces the number of decision tree layers, and reduces the number of decision tree branches. On the other hand, the decision tree pruning algorithm has a smaller space, Time complexity is lower, and the result of the classification is also better. It can be seen theoretically that the incremental decision tree constructed under the Bayesian node is suitable for dealing with the characteristic data with correlation. In the case of increasing the amount of data, the Bayesian method can well fit the existing decision tree according to the current growth data, so that the decision tree has better classification and regression ability of incremental data. In general, the time series data changes with time, and at time scale, the time series data have a strong correlation before and after, and the relationship is orderly, the dimension of the feature is higher, and it needs to complete the classification of data in the incremental way, and it is better for the use of incremental decision tree algorithm to complete the optimization.

4 Simulation experiment and result analysis

4.1 Data set and experimental environment

The UCR time series data set is the most commonly used time series signal analysis data set. The data set contains more than 200 time series sample sets, which can compare the results of various algorithms. In this paper, we select 15 of the most commonly used samples to test the improved algorithm and use non-incremental and incremental methods to train. In the process of testing, the classification accuracy and time complexity of ID3 algorithm, Bayesian algorithm and improved Bayesian node learning incremental decision tree algorithm are compared. Table 1 below shows the set of

samples in the UCR dataset selected in this paper, giving the training set size, test set size and category, time series length, and so on.

4.2 Experimental results and analysis

During the experiment, we extract 90% from each data set as the training data and take the remaining 10% of the data set as test data, then carry out 10 * 10 cross validation cycle from beginning to end. Through this process to analyze the feasibility and advantages of the algorithm proposed in this paper. In the comparison of the algorithms, take the two classical decision tree algorithms, ID3 and C4.5, and the Bayesian classification method as the baseline comparison algorithm, and compare them with the Bayesian node incremental decision tree proposed in this paper. During the process of actual comparison, compare the two aspects of performance and efficiency of non-incremental data and incremental data respectively.

Performance is one of the most important factors of the algorithm. In the 10 data sets given in Sect. 4.1, carry out the comparison experiment of ID3 decision tree algorithm, C4.5 decision tree algorithm, Bayesian classification algorithm and the proposed experiment based on Bayesian node incremental decision tree algorithm proposed in this paper respectively. Among them, carry out the cross validation for each algorithm for the 10 data sets, then give the average correct rate. Table 2 below shows the accuracy comparison of non-incremental learning classification results on the UCR sample set. Table 3 below shows the accuracy comparison of the UCR sample set in incremental learning classification results. Figure 3 shows the comparison of the correctness of non-incremental and incremental learning classification.

It can be seen from the data analysis of the simulation experiment that, the incremental decision tree algorithm

Table 2 Comparison of four algorithms in non-incremental data mining

Data set	Accuracy of classification (%)			
	ID3 decision tree	C4.5 decision tree	Bayesian classification	Proposed algorithm
Gun-point	99.0	98.2	100.0	100.0
Lightning-2	91.2	90.5	93.8	99.2
Synthetic control	94.8	94.7	94.2	98.4
Trace	92.3	95.1	90.1	97.8
OSU leaf	89.7	90.2	89.6	99.1
50Words	88.6	89.7	88.6	98.8
Swedish leaf	89.8	91.5	87.1	99.1
Two patterns	90.7	93.5	82.1	97.6
Wafer	91.2	94.4	80.7	95.7
Yoga	90.1	92.1	78.2	94.8

Table 3 Comparison of the proposed algorithm and ID3 algorithm in incremental learning

Data set	Incremental proportion (%)	Classification accuracy (%)			
		Proposed algorithm	ID3	C4.5	Bayesian classification
Gun-point	20	93.5	89.7	91.2	88.4
Lightning-2	30	94.6	91.4	92.3	89.6
Synthetic control	20	93.7	88.7	89.2	87.1
Trace	25	90.8	83.3	84.6	81.2
OSU leaf	25	93.5	90.4	89.4	87.3
50Words	30	92.6	83.7	87.8	85.8
Swedish leaf	40	94.1	80.1	82.1	83.1
Two patterns	35	93.2	81.3	86.5	85.2
Wafer	40	91.6	80.2	88.4	74.8
Yoga	50	89.7	73.2	82.4	80.1

based on Bayesian node learning proposed in this paper has a strong feasibility for classifying time series data, in contrast to the traditional ID3 decision tree algorithm and the Bayesian algorithm, in the non-incremental data mining on the 10 data samples, the proposed algorithm has an average increase of 6.31% and 9.61% than the ID3 algorithm and Bayes algorithm, in the incremental data mining, it has an average increase of 8.53% than the ID3 algorithm. To a certain extent, the proposed algorithm better performance. In the process of actual use, each node can use the Bayesian node's machine learning model to judge, this judgment is more credible, and has a more obvious enhance in data mining results, and can get more reliable, more robust data mining results.

Equivalent to performance, efficiency is also one of the most important considerations of an algorithm. Under the same experimental conditions of experiment performance, in this paper, the three baseline algorithms are compared with the algorithm proposed in this paper at the average time of 10*10 cross validation. Table 5 below shows the time complexity comparison of non-incremental learning classification on the UCR sample set. Table 5 below shows the time complexity comparison of the incremental learning classification on the UCR sample set. Figure 4 shows a comparison of the non-incremental and incremental learning classification efficiency. Besides, the storage comparison of the non-incremental and incremental learning classification is also compared in the Fig. 5.

From the data analysis of the simulation experiment, we can see that, the incremental decision tree algorithm based on Bayesian node learning proposed in this paper has the best performance for classifying time series data, and the minimum time complexity is consumed. In contrast to the traditional ID3 decision tree algorithm and the Bayesian algorithm, the algorithm proposed in this paper improves the efficiency of the 10 data samples on the non-incremental data mining by 7.21% and 8.37% respectively compared with the ID3 algorithm and the Bayes algorithm, in the incremental data mining, it has an average increase of 9.82% than
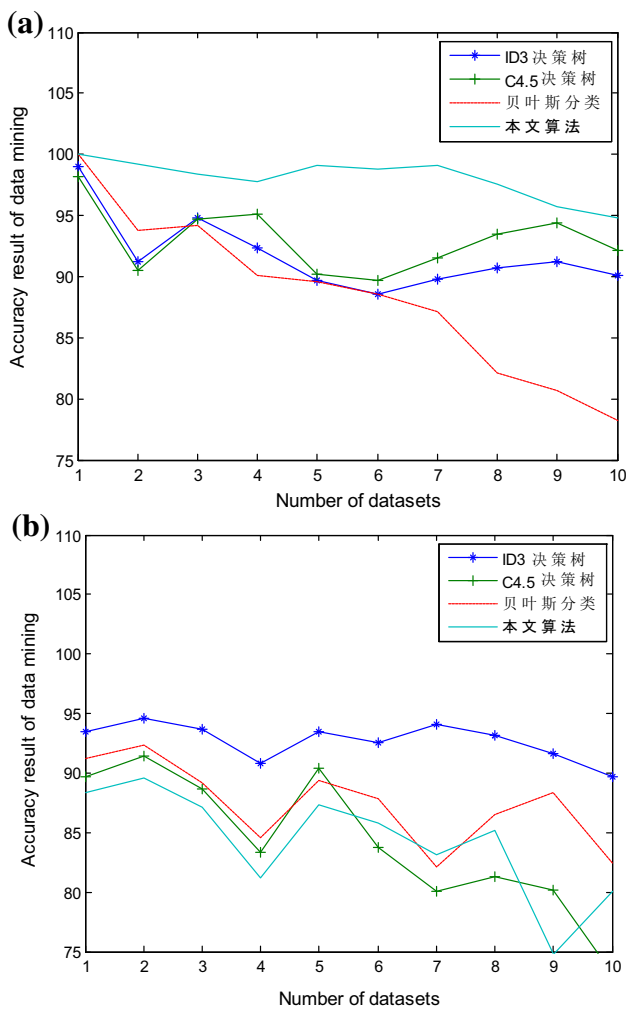
(a)

(b)

Fig. 3 Accuracy results comparison of different algorithms in non-incremental and incremental learning classification. a Accuracy comparison results of UCR sample set in non-incremental learning classification b Accuracy comparison results UCR sample set in incremental learning classification

the ID3 algorithm. The storage of algorithm is still a best index for the comparison of algorithms. From the results of the original algorithms of decision tree, it needs more space for the storage and the space usage ratio is bad for the algorithm. Conversely, the incremental algorithm with the Bayesian nodes have a better space usage ratio for the algorithm, by including the Bayesian nodes for the optimization of resource, the nodes have more data processing ability so that the storage will be less for the optimization. The storageof the 10 data samples on the non-incremental data mining by 8.46% and 9.21% respectively compared with the ID3 algorithm and the Bayes algorithm, in the incremental data mining, it has an average increase of 9.62% than the ID3 algorithm. The causes of selecting difference are not the main factors in the experiments, in fact, the selecting difference can be ignored because of the data mining methods are statistical methods.

To a certain extent, the proposed algorithm can get a better performance in the data mining. In the process of actual use, although each node needs to consume a certain amount of time to build a Bayesian node, the addition of Bayesian nodes to the decision tree greatly reduces the need for the number of branches; the time required for pruning is smaller, and the efficiency of data mining has greatly improved. For the application of data mining, the algorithm is a more useful status in the optimization of data mining. In this paper, the incremental decision tree with the Bayesian nodes will be better for the complex scene of the application of data mining. Besides, the optimal efficiency and the storage of Bayesian nodes will cope with more situations in the data mining region.

Table 4 Comparison of time complexity of four algorithms in non-incremental data mining

Data set	Time complexity (ms)			
	ID3 decision tree	C4.5 Decision tree	Bayesian classification	Proposed algorithm
5.Gun-point	102	111	78	98
Lightning-2	113	123	104	118
Synthetic control	127	138	98	105
Trace	198	201	177	189
OSU leaf	185	193	158	167
50Words	213	208	187	198
Swedish leaf	227	215	197	204
Two patterns	298	289	217	238
Wafer	679	598	542	589
Yoga	2717	2871	2389	2523

Table 5 Comparison of time complexity of four algorithms in incremental learning

Data set	Incremental proportion (%)	Time complexity (ms)			
		Proposed algorithm	ID3	C4.5	Bayesian classification
Gun-Point	20	93	87	93	91
Lightning-2	30	98	83	92	83
Synthetic control	20	108	91	98	76
Trace	25	167	134	111	99
OSU leaf	25	196	175	168	146
50Words	30	225	198	203	184
Swedish leaf	40	243	231	213	177
Two patterns	35	262	256	268	254
Wafer	40	612	598	489	368
Yoga	50	3121	2987	3192	2832
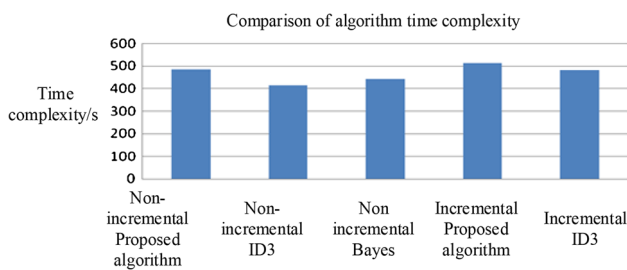
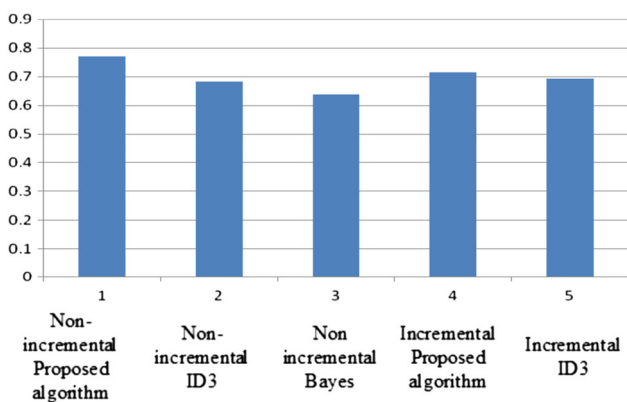**Fig. 4** Time complexity comparison results among data mining algorithms

Fig. 5 Storage compexity comparison resutls among data mining algorithms

5 Conclusion

In today's data explosion era, the analysis of time series data has a variety of purposes. Time series data are given in forms of increments with time increasing, the main applications of data mining need to be developed. In this paper, the Bayesian posteriori probability estimation method is used to judge the decision tree nodes. The incremental samples can be classified by the posterior probability, which makes the algorithm

have stronger adaptability for the incremental data. The comparison experiment on the time series data set shows that the algorithm proposed in this paper can have more classification accuracy and robustness to the incremental data experiment. In the future study, we need to consider further optimizing the time complexity of the incremental decision tree.

References

1. Ghosh, D.K., Ghosh, N.: An entropy-based adaptive genetic algorithm approach on data mining for classification problems. Int. J. Comput. Sci. Inf. Technol. (2014)
2. Cohen, S., Rokach, L., Maimon, O.: Decision-tree instance-space decomposition with grouped gain-ratio. Inf. Sci. **177**(17), 3592–3612 (2007)
3. Zang, W., Zhang, P., Zhou, C., et al.: Comparative study between incremental and ensemble learning on data streams: case study. J. Big Data **1**(1), 5 (2014)
4. Wang, T., Li, Z., Yan, Y., et al.: An Incremental Fuzzy Decision Tree Classification Method for Mining Data Streams. Machine Learning and Data Mining in Pattern Recognition, pp. 91–103. Springer, Berlin Heidelberg (2007)
5. Wang, L.: Incremental hybrid Bayesian network with noise resistance ability. Chinese J. Sci. Instrum. (2005)
6. Hu, X.G., Li, P.P., Wu, X.D., et al.: A semi-random multiple decision-tree algorithm for mining data streams. J. Comput. Sci. Technol. **22**(5), 711–724 (2007)
7. Hester, T., Stone, P.: Intrinsically motivated model learning for developing curious robots. Artif. Intell. **247**, 170–186 (2017)
8. Xing-Ming, O., Liu, Y.S.: A new algorithm for incremental induction of decision tree. Microprocessors (2008)
9. Rokach, L.: Mining manufacturing data using genetic algorithm-based feature set decomposition. Int. J. Intell. Syst. Technol. Appl. **4**(1/2), 57–78 (2008)
10. Salperwyck, C., Lemaire, V.: Incremental decision tree based on order statistics. International Joint Conference on Neural Networks, pp. 1–8 (2013)
11. Zhou, Z.H., Li, H., Yang, Q.: Advances in knowledge discovery and data mining : 11th Pacific-Asia conference, PAKDD 2007, Nanjing, China, May 22–25, 2007: Proceedings. Pacific-Asia Conference on Knowledge Discovery & Data Mining. Springer (2007)

12. Domshlak, C., Karpas, E., Markovitch, S.: Online speedup learning for optimal planning. AI Access Foundation (2012)

13. Kargupta, H.: Onboard vehicle data mining, social networking, and pattern-based advertisement (2011)

14. Li, B., Yuan, S.: Incremental hybrid Bayesian network in content-based image retrieval. Canadian Conference on Electrical and Computer Engineering, 2005. IEEE, pp. 2025–2028 (2005)

15. Jantan, H., Hamdan, A.R., Othman, Z.A.: Classification for talent management using decision tree induction techniques. Conference on Data Mining and Optimization, Dmo 2009, Universiti Kebangsaan Malaysia, 27–28 October. DBLP, pp. 15–20 (2009)

16. Qi, R., Poole, D.: A new method for influence diagram evaluation. Comput. Intell. **11**(3), 498–528 (2010)

17. Collet, T., Pietquin, O.: Bayesian Intervals, Credible, for Online and Active Learning of Classification Trees. Computational Intelligence, : IEEE Symposium. IEEE 2015, pp. 571–578 (2015)

18. Jiang, F., Sui, Y., Cao, C.: An incremental decision tree algorithm based on rough sets and its application in intrusion detection. Artif. Intell. Rev. **40**(4), 517–530 (2013)

19. Kolter, J.Z., Maloof, M.A.: Dynamic weighted majority: an ensemble method for drifting concepts. JMLR.org (2007)

20. Kargupta, H.: Onboard vehicle data mining, social networking, advertisement: US, US8478514 (2013)

21. Samet, S., Miri, A., Granger, E.: Incremental learning of privacy-preserving Bayesian networks. Appl. Soft Comput. J. **13**(8), 3657–3667 (2013)

22. Cazzolato, M.T., Ribeiro, M.X.: A statistical decision tree algorithm for medical data stream mining. IEEE, International Symposium on Computer-Based Medical Systems, pp. 389-392. IEEE (2013)

23. Kumari, S.R., Kumari, P.K.: Adaptive anomaly intrusion detection system using optimized Hoeffding tree. J. Eng. Appl. Sci. **95**(17), 22–26 (2014)

24. Zurn, J.B., Motai, Y., Vento, S.: Self-reproduction for Articulated Behaviors with Dual Humanoid Robots Using On-line Decision Tree Classification. Cambridge University Press (2012)

25. White, K.J., Sutcliffe, R.F.E.: Applying incremental tree induction to retrieval from manuals and medical texts. J. Assoc. Inf. Sci. Technol **57**(5), 588–600 (2006)

26. Wang, S.Y., Wang, Z.H.: Research on the time series classification algorithm based on incremental decision-tree. Modern Comput. (2015)

27. Bajo, J., Borrajo, M.L., Paz, J.F.D., et al.: A multi-agent system for web-based risk management in small and medium business. Expert Syst. Appl. **39**(8), 6921–6931 (2011)

28. Todorovic, S., Ahuja, N.: Unsupervised category modeling, recognition, and segmentation in images. IEEE Trans. Pattern Anal. Mach. Intell. **30**(12), 2158–74 (2008)

29. Shi, Y., Yang, Y.: An algorithm for incremental tree-augmented naive Bayesian classifier learning. International Conference on Artificial Intelligence and Computational Intelligence. IEEE, pp. 524–527 (2010)

30. Ordonez, C., Zhao, K.: Evaluating Association Rules and Decision Trees to Predict Multiple Target Attributes. IOS Press (2011)

Sun Xingrong was born in Qichun County Hubei Province. She is an associate professor of the College of Mathematics and Physics in Huanggang Normal University.