Novel Incremental ID3 Algorithm for Classification

R. G. Mehta, N. J. Mistry, and M. Raghuvanshi

Abstract

Discretization transforms continuous attribute values into a finite number of intervals and associates each interval with a numerical or discrete value. For mixed-mode (continuous and discrete) data, discretization is usually performed prior to learning and plays an important role in KDD process. CAIM is very efficient, supervised discretization algorithm. Recent data mining technology is found to be slow to handle data of very large scale. In addition, data mining needs to be an online process, rather than an occasional one-shot approach which has created a need for incremental approaches for effective model preparation and updating. Incremental classification is proposed in literature needs online discretization, has created a need for fast and efficient discretization algorithm. Modified CAIM (MoCAIM) algorithm is proposed and used as online discretization algorithm. Improved NID3 (INID3) is proposed to improve the classification accuracy by considering the unclassified instances of the test and prediction phases of the classification process. The outperforming results of MoCAIM and INID3 algorithms in term of classification accuracy and execution time motivate to explore the process further for the streamed data classification.

© 2013 Elixir All rights reserved

Introduction

In present era of Information Technology, data and information present in the internet and database of organization or corporation changes rapidly. The data collected from online data sources are streamed in nature possibly with time stamp. Streamed data have produced many issues like very huge size and large number of dimensions. Data Mining (DM) algorithms dig the hidden pattern form the well of the data. The DM processes like tree based classification (e.g. ID3, C4.5) derive tree based model to store the conditions to derive the category of the instance, which help in predicting the category of the unknown instances. But they are suitable for the categorical data only.

The continuous data are transformed into corresponding categorical values using discretization process. Many discretization algorithms exist among which CAIM shows outperforming results in term of execution time [1] [2] [3]. CAIM discretization algorithm maximizes the class attribute interdependency and subsequently, possibly makes smallest number of Interval, without any user intervention and subsequently improves the result of classification algorithm. In our previous research paper we have shown the outperforming results of the tree based classification with CAIM as online discretization criteria [4].

The streamed data has created a need to implement an incremental approach in the traditional DM algorithms [5]. Incremental algorithms build and refine the model as new data arrive at different points in time, in contrast to the traditional tree induction algorithms where they perform model building in batch manner [6]. Typically the new data arrived will be merged with the previous data and the model will be reconstructed. But, in this approach all the previously discovered knowledge will be lost and the merging is not always a viable option for situations like unavailability and inaccessibility of previous datasets. Incremental classifier is the solution to such scenarios, which can be defined as the process of extracting new patterns without losing prior knowledge from an additional dataset that later becomes available [6]. In this paper the Novel incremental classification is proposed where the model is prepared using existing dataset. The model can then be used to make predictions about new data items whose category is unknown. During the prediction phase, the unpredicted instance can be applied to the model for the further refinement in the model. During the prediction phase of the classification process, whenever the new data instance has been arrived, having some different characteristic than the old training data, we need to rebuild the model with all data (old and new). But it is more time consuming and more complex as it builds the tree from the scratch. Here novel incremental classification technique is proposed, which modify the existing model using the unclassified instances, identified during the test phase.

In this paper Incremental Novel ID3 (INID3) algorithm is proposed along with Modified CAIM algorithm (MoCAIM). This algorithm uses NID3, proposed by M. Lad et al. in 2012, where they have used CAIR as attribute selection criteria instead of traditional Information Gain and CAIM for online discretization of quantitative features of the active dataset during the classification model preparation phase. The process of CAIM is updated to reduce time and space complexity of CAIM and proposed a MoCAIM algorithm. INID3 uses CAIR as an
attribute selection criteria and the proposed MoCAIM is used as online discretization to improve the classification accuracy and execution speed.

The rest of the paper is organized as follows. Section II covers the review of related works. Section III presents Novel ID3 algorithm for classification, proposed MoCAIM discretization algorithm along with INID3 classification is described in section IV, followed by performance analysis in section V. The classification accuracy of incremental ID3 is tested using both the discretization approaches. Conclusion and future work is discussed in section VI.

Relative Work

Extensive work is done for classification algorithm and ID3 algorithm is found to be a most popular tree based classification algorithm [7]. Many updates are proposed for ID3 by many researchers [8] [9] [10]. The novel ID3 algorithm is proposed with CAIR as attribute selection criteria and CAIM as online discretization criteria have shown the outperforming results compared to other traditional attribute selection criteria [5].

Discretization algorithms are mainly categorized as supervised and unsupervised algorithms. Class Attribute Interdependence Maximization (CAIM) is supervised top-down algorithm considers the highest interdependence between class and attribute and improves classification accuracy. Unlike other discretization algorithm CAIM automatically generate the intervals and interval boundaries for the given data without any user input. But out of many advantages of CAIM discretization algorithm, the algorithm suffers from larger size quanta matrix that increases the time and space complexity of the algorithm.

CAIM and Novel ID3 classification algorithms are discussed in the next section followed by proposed MoCAIM algorithm and proposed Incremental classification algorithm with MoCAIM as online discretization.

Novel ID3 : A tree based classification

CAIM criteria:

The Class Attribute Interdependency Maximization (CAIM) criterion is a heuristic measure that is used to quantify the interdependence between classes and the discretized attribute. It measures the dependency between the class variable C and the discretization variable D for attribute F, for a frequency distribution (quanta matrix) as shown in Figure 1, is defined as:

$$CAIM(C, D | F) = \frac{\sum_{r} \max_{i} M_{ir}}{n}$$

Where n is the number of intervals, r iterates through all intervals, i.e., r = 1, 2, . . ., n, max, is the maximum value among all d_{ir} values (maximum value within the r^{th} column of the quanta matrix), i = 1, 2,...,S, M_{ir} is the total number of continuous values of attribute F that are within the interval [d_{ir}, d_{ir+1}].

![Fig.1 Quanta Matrix](image)

The CAIM algorithm consists of these two steps:

1. Initialization of the candidate interval boundaries and the initial discretization scheme and
2. Consecutive additions of a new boundary that results in the locally highest value of the CAIM criterion.

Algorithm works in greedy manner. The discretization schemes generate high class-attribute interdependency and small number of discretization intervals.

Novel ID3 (NID3) : A Novel ID3 classification

The Novel Tree based classification algorithm is presented [4], which follows the model creation process as ID3 but use CAIR [2] as attribute selection and CAIM [3] as online discretization criteria. The Algorithm is described as below.

Algorithm 1: NID3 [4]

1. Select an attribute except that attribute whose value has to be predicted.
2. If attribute is continuous, discretize it using CAIM.
3. Calculate CAIR value for that attribute using Equation.
4. Repeat steps 1 and 2 for each attribute.
5. Then select an attribute for which CAIR is maximum
6. Make node containing that attribute.
7. Then on the basis of that attribute, divide the given training set in to subsets.
8. Then recursively apply the algorithm on each subset until the set contains instances of the same class. If the set contains instances of the same class, then return that class.

In this proposed algorithm, if attribute is continuous than it is discretized using CAIM. Then CAIR for each attribute is calculated. The attribute with highest CAIR will be selected as the current node of the tree. Then using the different values of that attribute, divide the given training set into subsets. The process is recursively applied on each subset. If the set contains instances of the same class, then return the class. The model is applied to test data where data will be generated as Unclassified, correctly classified and non- correctly classified. The efficiency is calculated based on percentage of the correctly classified records. To improve the classification accuracy incremental approach is proposed which is discussed in next section.

Proposed MoCAIM discretization based Incremental ID3 (INID3) classification

Modified CAIM Discretization

The main goal of the MoCAIM algorithm is to reduce the space and time complexity of the CAIM algorithm. In CAIM discretization process, quanta matrix created for each continuous attributes are very huge, it allocates lots of space in the main memory that affect the speed and increases load on the system. The proposed algorithm reduces the space complexity, speed and execution time of the original CAIM discretization algorithm.

Algorithm 2: MoCAIM

1.1 Find the minimum(d_{0}) and maximum(d_{n}) values of F_{i}
1.2 Call Tree-set in java, which form a set in such a way that it takes all the unique value in ascending order. Assign first value as minimum(d_{0}) and last value as maximum(d_{n}).
1.3 Merge all the quanta matrix intervals having the highest value at the same level (row).
1.4 Set the initial discretization scheme D: (d_{0},...,d_{n}) and set GlobalCAIM=0;
2.1 Initialize k=1;
2.2 Tentatively add the next higher value in the Tree-set and check the CAIM values for each attributes.
2.3 After checking all tentative values accept the one which have highest CAIM value.
2.4 If(CAIM>GlobalCAIM or K=S) then update D with the accepted value in step 2.3 and set GlobalCAIM = CAIM, else Terminate.
2.5 Set k = k +1 and goto 2.2

Output: Discretization D.

Proposed INID3 Classification framework

In NID3, the classification model is built on training data set as described in Algorithm1. The test dataset is applied to the model to check the accuracy. The test process will generate three categories of the instances: Correctly classified, Incorrectly classified and Unclassified. To improve the model efficiency, the unclassified records are applied to the classification model. Based on the tree creation statistic of Algorithm 1, the model is updated and nodes and branches will be added/updated. Here the MoCAIM is used in step 2 of Algorithm 1 of the classification process. The process is shown in the block diagram of Fig. 2.

Fig 2 Proposed INID3 classification model

The small dataset for Buying a computer is depicted in Figure 3. The INID3 process is used using this dataset.

Table 1: A small Data Set of Buying a computer

<table>
<thead>
<tr>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit rating</th>
<th>Buy computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>31…40</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>31…40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>Medium</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;30</td>
<td>Medium</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>31…40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
</tbody>
</table>

1. The first three records are considered as training set.
2. The classification model is prepared as depicted in Figure 3.
3. This model is evaluated using all the records of test data sets.
   Its result is given below.
   - Correctly Classified records: 7
   - Incorrectly Classified records: 2
   - Not Classified records: 5
   - Classification Accuracy: 50.0%

Fig. 3: Tree generated after classification

Fig. 4: The Updated classification tree of INID3

In the result section we have shown the various results of MoCAIM discretization algorithm with and without incremental classification. To analyze our proposed algorithm, different datasets from the famous UCI repository are selected. The Datasets are described in table 2.

Table 2: Data Set Description for INID3 analysis

<table>
<thead>
<tr>
<th>Data Set</th>
<th>No. of Classes</th>
<th>No. of Attributes</th>
<th>No. of Continuous Attributes</th>
<th>No. of Training Records</th>
<th>No. of Test Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ozone</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>1000</td>
<td>2534</td>
</tr>
<tr>
<td>Adult</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>5000</td>
<td>52561</td>
</tr>
<tr>
<td>Car</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>1000</td>
<td>1728</td>
</tr>
<tr>
<td>CRX</td>
<td>16</td>
<td>6</td>
<td>0</td>
<td>200</td>
<td>990</td>
</tr>
</tbody>
</table>

Table 3: Efficiency analysis for NID3 and INID

<table>
<thead>
<tr>
<th>Data set</th>
<th>Classification Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NID3</td>
<td>1</td>
</tr>
<tr>
<td>INID3</td>
<td>2</td>
</tr>
<tr>
<td>Ozone</td>
<td>94.08</td>
</tr>
<tr>
<td>Adult</td>
<td>80.90</td>
</tr>
<tr>
<td>Car</td>
<td>85.35</td>
</tr>
<tr>
<td>CRX</td>
<td>62.02</td>
</tr>
</tbody>
</table>

Table 4: Execution time analysis for INID3 with CIAIM vs INID3 with MoCAIM

<table>
<thead>
<tr>
<th>Data set</th>
<th>INID3 with CIAIM (ms)</th>
<th>INID3 with MoCAIM (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ozone</td>
<td>185297</td>
<td>14040</td>
</tr>
<tr>
<td>Adult</td>
<td>21809</td>
<td>6724</td>
</tr>
<tr>
<td>Car</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CRX</td>
<td>47</td>
<td>32</td>
</tr>
</tbody>
</table>

The highlighted values of Table 3 depict the improved classification accuracy of the INID3. As the test dataset of “Ozone” dataset contains no unseen data, so the incremental classification shows same result but for all the remaining datasets of Table 2, the higher classification accuracy in Column 2 of table 3 are motivating for further analysis. As discretization intervals of the MoCAIM will remain same as that of CAIM,
classification accuracy will not be affected; only improved execution time for applying test data to the classification model is reflected in table 4.

Summary and Conclusion

Discretization played a major role in the tree based classification and Incremental classification algorithm for the knowledge discovery process. From the existing research, it is proved that CAIM discretization gives very efficient discretization for tree based classification scheme. The proposed MoCAIM has improved the classification accuracy of NID3 and Proposed INDID3 classification algorithm. are inspiring to develop a framework to implement incremental NID3 for stream data, with efficient storage structure.

References

10. C. Guan and X. Zeng, “An Improved ID3 Based on Weighted Modified Information Gain,” 7th international Conference on Computational Intelligence and Security.
13. UCI Repository, www.uci-repository.edu