Incremental Discretization for Naïve Bayes Learning using FIFFD

Mr. Kunal Khimani 1  Mr. Kamal Sutaria 2  Ms. Kruti Khalpada 3
1 Gujarat Technological University PG School, Ahmedabad, Gujarat
2 Asst. Prof., C.E. Department, VVP Engineering College, Rajkot Gujarat
3 Institute of Technology, Nirma University, Ahmedabad, Gujarat

Abstract—Incremental Flexible Frequency Discretization (IFFD) is a recently proposed discretization approach for Naïve Bayes (NB). IFFD performs satisfactorily by setting the minimal interval frequency for discretized intervals as a fixed number. In this paper, we first argue that this setting cannot guarantee that the selecting MinBinSize is on always optimal for all the different datasets. So the performance of Naïve Bayes is not good in terms of classification error. We thus proposed a sequential search method for NB: named Flexible IFFD. Experiments were conducted on 4 datasets from UCI machine learning repository and performance was compared between NB trained on the data discretized by FIFFD, IFFD, and PKID.

Keywords: Discretization, incremental, Naïve Bayes.

I. INTRODUCTION

Naïve-Bayes classifiers are widely employed for classification tasks because of their efficiency and efficacy. Naïve Bayesian classifiers are simple, robust, and also support incremental training. Its efficiency is witnessed widespread deployment in classification task. Naïve Bayesian classifiers have long been a core technique in information retrieval. Naïve-Bayesian learning needs to estimate probabilities for each attribute-class pair. The naïve Bayesian classifier provides a very simple and yet surprisingly accurate technique for machine learning. The naïve Bayesian classifier provides a very simple and yet surprisingly accurate technique for machine learning. When classifying an instance, naïve Bayesian classifiers assume the attribute conditionally independent of each other given the class; then apply the Bayes’ theorem to estimate the probability of each class given the instance. The class with the highest probability is chosen as the class of instance.

An attribute can be either qualitative or quantitative. Discretization produces a qualitative attribute from a quantitative attribute. Naïve-Bayes classifiers can be trained on the resulting qualitative attributes instead of the original quantitative attributes then it increase the efficiency of classifier. Two terminologies bias and variance are widely used in NB discretization, which are interval frequency (the number of training instances in one interval) and interval number (the number of discretized intervals produced by a specific discretization algorithm). So we have to very careful about this two problem arising during discretization. Yang proposed the proportion K-interval discretization technique (PKID). PKID works based on the fact that there is a Tradeoff between interval number, interval frequency and the bias, variance component in the classification error decomposition. Also, “large interval frequency incurs low variance but high bias whereas large interval number produces low bias and high variance”. However, PKID does not work well with small data sets, which have at most 1200 instances. Then Ying and Webb proposed another technique called Fixed Frequency Discretization (FFD). FFD discretizes the training instances into a set of intervals which contain approximately the same number of m instances, where m is a parameter specified by user. Note that, in FFD the interval frequency is fixed for each interval without considering the number of training instances. The larger the training data size is, the larger the number of intervals is produced. However, the interval frequency will not change.

One another thing that the above both Fixed Frequency Discretization (FFD) and proportional K Interval Discretization (PKID) are not support incremental approach. Ideally, discretization should also be incremental in order to be coupled with NB. When receiving a new training instance, incremental discretization is expected to be able to adjust intervals’ boundaries and statistics, using only the current intervals and this new instance instead of re-accessing previous training data. Unfortunately, the majority of existing discretization methods are not oriented to incremental learning. To update discretized intervals with new instances, they need to add those new instances into previous training data, and then re-discretize on basis of the updated complete training data set. This is detrimental to NB’s efficiency by inevitably slowing down its learning process. Incremental Flexible Frequency Discretization (IFFD) is the first incremental discretization technique proposed for NB. IFFD sets the interval frequency ranging from MinBinSize to maxBinSize instead of single value m. The number MinBinSize and maxBinSize stand for the minimal and maximal interval frequency.

Some preliminary research has been already done to enhance incremental discretization for NB. A representative method, named PiD, proposed by Gama and Pinto is based on two layer histograms and is efficient in term of time and space complexity. The argument can be, that setting the MinBinSize as a fixed number does not guarantee that the classification performance of NB is optimum. There exists a most suitable MinBinSize for each dataset. Finally, propose a new incremental discretization method: FIFFD using a sequential search.

Computerized Numerical Control (CNC) cutting has various distinct advantages over the other cutting technologies, such as no thermal distortion, high machining versatility, high an effective technology for processing various engineering materials. The mechanism and rate of material removal during CNC machining depends both on the type of tool and on a range flexibility and small cutting
forces, and has been proven to be of cutting parameters. 
CNC can machining the hard and brittle materials like 
Steel, Non-ferrous alloys Ti alloys, Metal Matrix 
Composite, Ceramic Matrix Composite, Concrete, Stone, 
Granite, Wood, Reinforced plastics, Metal Polymer 
Laminates.

II. DISCRETIZATION FOR NAÏVE BAYES 
CLASSIFICATION

A. Naïve Bayes Classifier (NB)
Assume that an instance \( I = \{x_1, x_2, ..., x_n\} \), each value being an observation of 
an attribute \( X_i \in [1,n] \). Each instance can have a class 
label \( C_i \in \{c_1,c_2,..,c_n\} \), being a value of the class variable C. If an instance has a known class label, it is a training 
instance. If an instance has no known class label, it is a 
testing instance. The dataset of training instances is called 
the training dataset. The dataset of testing instances is called 
the testing dataset.

To classify an instance \( I = \{x_1, x_2, ..., x_n\} \), NB estimates 
the probability of each class label given I, \( P(C = c_i | I) \) 
using Formula (0.1), 0.2, 0.3, 0.4). Formula (1.2) follows 
(1.1) because P(I) is invariant across different class 
labels and can be canceled. Formula (1.4) follows (1.3) 
because of NB’s attributes independent assumption. It then 
assigns the class with the highest probability to I. NB is 
called naive because it assumes that attributes are 
conditionally independent of each other given the class 
label. Although its assumption is sometimes violated, NB 
is able to offer surprisingly good classification accuracy 
in addition to its very high learning efficiency, which makes NB 
popular with numerous real-world classification applications.

\[
P(C = c_i | I) = \frac{P(C = c_i)P(I | C = c_i)}{P(I)} \tag{0.1}
\]

\[
P(C = c_i | I) \propto P(C = c_i)P(I | C = c_i) \tag{0.2}
\]

\[
P(C = c_i)P(X = x_j | C = c_i) \tag{0.3}
\]

\[
P(C = c_i)\prod_{j=1}^{n}P(X_j = x_j | C = c_i) \tag{0.4}
\]

In naïve-Bayes classifier, the class type must be qualitative 
while the attribute type can be either qualitative or 
quantitative. When an attribute \( X_j \) is quantitative, it often 
has a large or even infinite number of values. As a result, the 
conditional probability that \( X_j \) takes a particular value \( x_j \) 
given the class label \( c_i \) covers very few instances if there is 
any at all. Hence it is not reliable to estimate 
P(\( X_j = x_j | C = c_i \)) according to the observed instances.

One common practice to solve the problem of quantitative data 
for NB is discretization.

B. Discretization
Discretization is a popular approach to transforming 
quantitative attributes into qualitative ones for NB. It groups 
sorted values of a quantitative attribute into a sequence of 
intervals, treats each interval as a qualitative value, and 
maps every quantitative value into a qualitative value 
according to which interval it belongs to. In the paper, the 
boundaries among intervals are sometimes referred to as cut 
points. The number of instances in an interval is referred to 
as interval frequency. The total number of intervals 
produced by discretization is referred to as interval number.

Incremental discretization aims at efficiently updating 
discretization intervals and associated statistics upon 
receiving each new training instance. Ideally, it does not 
require to access historical training instances to carry out 
the update. Instead it only needs the current intervals (with 
associated statistics) and the new instance.

1) Incremental Flexible Frequency Discretization
In this section, we propose a novel incremental 
discretization method, IFFD. It is motivated by the pros 
and cons of Incremental Flexible frequency discretization 
(IFFD) in the context of naïve-Bayes learning and 
incremental learning.

a) Incremental Flexible Frequency Discretization (IFFD)
IFFD sets its interval frequency to be a range [\( \text{minBinsize}, \text{maxBinsize} \)] 
instead of a single value m. The two arguments, \( \text{minBinsize} \) and \( \text{maxBinsize} \), are respectively 
the minimum and maximum frequency that IFFD allows 
intervals to assume. Whenever a new value arrives, IFFD 
first inserts it into the interval that the value falls into. IFFD 
then checks whether the updated interval’s frequency 
reaches \( \text{maxBinsize} \). If not, it accepts the change and update 
statistics accordingly. If yes, IFFD splits the overflowed 
interval into two intervals under the condition that any of the 
resulting intervals has its frequency no less than \( \text{minBinsize} \).

Otherwise, even if the interval overflows because of the 
insertion, IFFD does not split it, in order to prevent high 
classification variance. In the current implementation of 
IFFD, \( \text{minBinsize} \) is set as 30, and \( \text{maxBinsize} \) is set as 
twice of \( \text{minBinsize} \). Assume \( \text{minBinsize} = 3 \) and hence 
\( \text{maxBinsize} = 6 \). When the new attribute value “5.2” comes, 
IFFD inserts it into the second interval \{4.5, 5.1, 5.9\}. That 
interval is hence changed into \{4.5, 5.1, 5.2, 5.9\} whose 
frequency (equal to 4) is still within \[3, 6\]. So what we need 
do is only to modify NB’s conditional probably related to 
the second interval. Assume another two new attribute 
values “5.4, 5.5” have come and are again inserted into the 
second interval. This time, the interval \{4.5, 5.1, 5.2, 5.4, 
5.5, and 5.9\} has a frequency as 6, reaching \( \text{maxBinsize} \).
Hence IFFD will split it into \{4.5, 5.1, 5.2\} and \{5.4, 5.5, 5.9\} whose frequencies are both within [3, 6). Then we only need to recalculate NB’s conditional probabilities related to those two intervals. By this means, IFFD makes the update process local, affecting a minimum number of intervals and associated statistics. As a result, incremental discretization can be carried out very efficiently.

2) **Flexible IFFD**

The proposed new method FIFFD is based on the following drawback of IFFD: there exists a most suitable MinBinSize for the discretization intervals for each numeric attribute as the values of numeric attributes has some distribution. Though such a cumulative distribution does not necessarily be Gaussian distribution, if we could approximate the distribution using the optimal minimal discretization interval frequency (MinBinSize), it will in turn benefit the classification performance. It is hard to show that such an optimal interval frequency exists theoretically because our knowledge is very few about the data distribution, especially for unseen data. FIFFD works as follows: instead of setting the MinBinSize as 30 for all the data sets, we set a search space for the most suitable MinBinSize ranging from 1 up to the range specified by user. FIFFD works in rounds by testing each MinBinSize values, in each round, we do a sequential search on these range of values and set the current value as MinBinSize and discretize the data using IFFD based on the current MinBinSize value, we record the classification error for each round, if the classification error reduces once a MinBinSize is set, we will update the MinBinSize, this search process is terminated until all values ranging specified by user have been searched or the classification error no longer reduces.

The pseudo-code of FIFFD is listed in Algorithm. In FIFFD, we also set the maxBinSize as twice of MinBinSize. cut Points is the set of cut points of discretization intervals. Counter is the conditional probability table of the classifier. IFFD will update the cut Points and counter according to the new attribute value V. class Label is the class label of V. Note the FIFFD is a sequential search based supervised approach; the search efficiency for optimal MinBinSize is still efficient in the context of incremental learning. Therefore, the efficiency of FIFFD is comparable to that of IFFD.

3) **Algorithm: Flexible IFFD**

FIFFD (cut Points, counter, V, class Label, range) Generate the discretized data with most suitable minBinsize value.

**INPUT:** V: input data,
- Range: it specify the search space range.
- Counter: counter is the conditional probability table.
- Cut Points: cut Points is the set of cut points of discretization intervals.
- Class Label: class Label is the class label of V.

**OUTPUT:** discretized intervals with its most suitable binning value.

**METHOD:**
- Do a sequential search up to specified range and set the current value as minBinsize,
- While TRUE do
  - Test whether V is greater than the last cut point then
  - Insert V into the last interval;
  - Update the corresponding interval frequency;
  - Record changed interval;
- Else
  - Check for other intervals;
  - Find the cut point and insert values in to the interval;
  - Update particular interval;
- If frequency exceeded maximum size of interval
  - Get new cut points;
  - Insert new cut points in to cut points;
  - Calculate counter for each cut point;
- Note down current MinBinSize and NB classification error;
- Get new value for MinBinSize;
- End while
- Return ideal bin size;

### III. RESULT ANALYSIS

#### A. Dataset Descriptions

In this section, we will justify our claim on the existence of optimal minBinsize and evaluate our new discretization method FIFFD for NB with other alternatives, including PKID and IFFD.

We did our experiments on 4 datasets from UCI machine learning repository. Datasets information is summarized in Table 1. Size denotes the number of instances in a dataset, Qa. Is the number of numeric attributes, Cat. is the number of categorical attributes, and C means the number of different class values. We listed the empirical result for the existence of optimal minBinsize for each dataset in Figure 1.

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Dataset</th>
<th>Attributes</th>
<th>Records</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Glass</td>
<td>10</td>
<td>428</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Emotion</td>
<td>78</td>
<td>1186</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Sick</td>
<td>30</td>
<td>3772</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Pima</td>
<td>9</td>
<td>10000</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Adult</td>
<td>15</td>
<td>32560</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>Census</td>
<td>14</td>
<td>48998</td>
<td>17</td>
</tr>
</tbody>
</table>

**Table 1: Dataset information**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>FIFFD</th>
<th>IFFD_NB</th>
<th>PKID_NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass</td>
<td>95.79%</td>
<td>82.24%</td>
<td>84.11%</td>
</tr>
<tr>
<td>Emotion</td>
<td>91.39%</td>
<td>89.12%</td>
<td>89.20%</td>
</tr>
<tr>
<td>Sick</td>
<td>97.00%</td>
<td>96.95%</td>
<td>96.87%</td>
</tr>
<tr>
<td>Pima</td>
<td>96.64%</td>
<td>92.47%</td>
<td>88.02%</td>
</tr>
<tr>
<td>Adult</td>
<td>84.25%</td>
<td>82.14%</td>
<td>81.82%</td>
</tr>
<tr>
<td>Census</td>
<td>46.97%</td>
<td>46.22%</td>
<td>46.76%</td>
</tr>
</tbody>
</table>

**Table 2: Naïve Bayes Accuracy comparison**

Table 2 indicates that the classification performance of NB with FIFFD is much better than that of NB with IFFD and PKID. NB with FIFFD outperforms NB with PKID and NB with IFFD on all 6 datasets we have tested. The reason is that NB with FIFFD used a sequential search approach and tried to improve the classification performance of NB as much as possible.
B. Analysis

Figure 2 shows the performance of accuracy study which has been carried out on different size of datasets. The accuracy of the proposed system has been tested for both IFFD and PKID method. The experiment shows that the accuracy is improved in each case for the proposed system. It is provided that our method is best.

Fig. 2: Accuracy performance

Figure 3 show the classification error rate of Naïve Bayes trained on most suitable Binning and MinBinSize is ranging from 1 to 45 in below figure. It is easily concluded that a most suitable BinSize is exists for each dataset. Fig (a) shows the error rate of FIFFD is minimal when the MinBinSize is 1 for Glass, Emotion and Census datasets. If we increase the value of MinBinSize then the performance tends to be worse. Fig (b) shows the error rate of FIFFD is minimal when MinBinSize is 30 for Sick, 25 for German, 37 for Magik Gamma, 38 for Ecoli datasets. If we decrease the value, performance tends to be worse.

Fig. 3: Classification Error Rate of NB

IV. CONCLUSION

We experimentally found out that a most suitable BinSize exists for each and every datasets. The previous incremental discretization methods for Naïve Bayes learning were having the problem of fixed interval size that is not ideal for all data sets. The proposed system based on sequential search that is the incremental discretization with FIFFD can find the ideal interval size which can make the Naïve Bayes classifier more efficient by reducing the classification error rate. So NB with FIFFD is much better than that of Naïve Bayes with PKID and IFFD.

FUTURE EXTENSION

There still exists some scope for the improvement in the proposed system. One can prove it theoretically that why such kind of most suitable interval size or binning exists. The second one is if try to know something more about the data distribution and use such a domain knowledge to direct the process of discretization.

REFERENCES