Enhancing Forecasting Performance of Naïve-Bayes Classifiers with Discretization Techniques

Ruxandra PETRE
University of Economic Studies, Bucharest, Romania
ruxandra_stefania.petre@yahoo.com

During recent years, the amounts of data, collected and stored by organizations on a daily basis, have been growing constantly. These large volumes of data need to be analyzed, so organizations need innovative new solutions for extracting the significant information from these data. Such solutions are provided by data mining techniques, which apply advanced data analysis methods for discovering meaningful patterns within the raw data. In order to apply these techniques, such as Naïve-Bayes classifier, data needs to be preprocessed and transformed, to increase the accuracy and efficiency of the algorithms and obtain the best results.

This paper focuses on performing a comparative analysis of the forecasting performance obtained with the Naïve-Bayes classifier on a dataset, by applying different data discretization methods opposed to running the algorithms on the initial dataset.

Keywords: *Discretization, Naïve-Bayes classifier, Data mining, Performance*

1 Introduction

Nowadays, organizations collect large amounts of data every day. These data need to be analyzed in order to find the meaningful information contained by it and reach the best conclusions, to support decision making.

Data mining is an innovative new solution, which provides the required tools for processing the data in order to extract significant patterns and trends. Before running data mining algorithms against it, the raw data needs to be cleaned and transformed. This is accomplished through the preliminary steps of the Knowledge Discovery in Databases process – data preprocessing and data transformation. One of the key methods used during data transformation is data discretization.

Discretization methods transform the continuous values of a dataset attribute to discrete ones. It can help improve significantly the forecasting performance of classification algorithms, like Naïve Bayes, that are sensitive to the dimensionality of the data.

Naïve-Bayes is an intuitive data mining algorithm that predicts class membership, using the probabilities of each attribute value to belong to each class.

Discretization methods needs to be applied on datasets before analyzing them, in order to transform the continuous variables to discrete variables and, thus, to improve the accuracy and efficiency of the classification algorithm.

2. Naïve-Bayes classifiers: overview

Classification is a fundamental issue in machine learning and statistics. It is a supervised data mining technique, with the goal of accurately predicting the class label for each item in a given dataset. A classification model built to predict class labels, from the attributes of the dataset, is known as a classifier.

In data mining, Bayesian classifiers are a family of probabilistic classifiers, based on applying Bayes' theorem. The theorem, named after Reverend Thomas Bayes (1701–1761), who has greatly contributed to the field probability and statistics, is a mathematical formula used for calculating conditional probabilities. It relates current probability to prior probability.

Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class. Studies comparing classification algorithms have found a simple Bayesian classifier known as the naïve Bayesian classifier to be comparable in performance with decision tree and selected neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases. [1]

The Naïve-Bayes classifier is an intuitive data mining method that uses the probabilities of each attribute value belonging to each class to predict class membership.

A Bayesian classifier is stated mathematically as the following equation: [1]

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \qquad (1)$$

where,

- $P(C_i|X)$ is the probability of dataset item X belonging to class C_i;
- $P(X|C_i)$ is the probability of generating dataset item X given class C_i;
- $P(C_i)$ is the probability of occurrence of class C_i;
- $P(X)$ is the probability of occurrence of dataset item X.

Naïve-Bayes classifiers simplify the computation of probabilities by assuming that the probability of each attribute value to belong to a given class label is independent of all the other attribute values.

This method goes by the name of Naïve Bayes because it's based on Bayes' rule and "naïvely" assumes independence—it is only valid to multiply probabilities when the events are independent. The assumption that attributes are independent (given the class) in real life certainly is a simplistic one. But despite the disparaging name, Naïve Bayes works very effectively when tested on actual datasets, particularly when combined with some of the attribute selection procedures. [2]

3 Discretization techniques: theoretical framework

The Knowledge Discovery in Databases (KDD) process is an iterative process for identifying valid, new and significant patterns in large and complex datasets. The core step of the KDD process is data mining, which involves developing the model for discovering patterns and trends in the data.

Data preprocessing and data transformation are crucial steps of the KDD process. After performing them better data should be generated, in a form suitable for the data mining algorithms.

Data transformation methods include dimension reduction (such as feature selection and extraction, and record sampling), and attribute transformation (such as discretization of numerical attributes and functional transformation). [3]

Most experimental datasets have attributes with continuous values. However, data mining techniques often need that the attributes describing the data are discrete, so the discretization of the continuous attributes before applying the algorithms is important for producing better data mining models.

The goal of discretization is to reduce the number of values a continuous attribute assumes by grouping them into a number, n, of intervals (bins). [4]

Mainly there are two tasks of discretization. The first task is to find the number of discrete intervals. Only a few discretization algorithms perform this; often, the user must specify the number of intervals or provide a heuristic rule. The second task is to find the width, or the boundaries, of the intervals given the range of values of a continuous attribute. [5]

Data discretization comprises a large variety of methods. They can be classified based on how the discretization is performed into: supervised vs. unsupervised, global vs. local, static vs. dynamic, parametric vs. non-parametric, hierarchical vs. non-hierarchical etc.

There are several methods that can be used for data discretization. Supervised discretization methods can be divided into error-based, entropy-based or statistics based. Among the unsupervised discretization methods there are the ones like equal-width and equal-frequency.

- *Equal-width discretization*

This method consists of sorting the values of the dataset and dividing them into intervals (bins) of equal range. The user specifies k, the number of intervals to be calculated, then the algorithm determined the minimum and maximum values and divides the dataset into k intervals.

Equal-width interval discretization is a simplest discretization method that divides the range of observed values for a feature into k equal sized bins, where k is a parameter provided by the user. The process involves sorting the observed values of a continuous feature and finding the minimum, V_{min} and maximum, V_{max}, values. [5]

The method divides the dataset into k intervals of equal size. The width of each interval is calculated using the following formula:

$$Width = \frac{V_{\max} - V_{\min}}{k} \qquad (2)$$

The boundaries of the intervals are calculated as: V_{min}, V_{min}+Width, V_{min}+2Width, ... , V_{min}+(k-1)Width, V_{max}.

The limitations of this method are given by the uneven distribution of the data points: some intervals may contain much more data points than other. [5]

- *Equal-frequency discretization*

This method is based on dividing the dataset into intervals containing the same number of items. Partitioning of data is based on allocating the same number of instances to each bin. The user supplies k, the number of intervals to be calculated, then the algorithm divides n, the total number of items belonging to the dataset, by k.

Equal-Frequency Discretization predefines k, the number of intervals. It then divides the sorted values into k intervals so that each interval contains approximately the same number of training instances. Suppose there are n training instances, each interval then contains n/k training instances with adjacent (possibly identical) values. [3]

The method divides the dataset into k intervals with equal number of instances. The intervals can be computed using the following formula:

$$Interval = \frac{n}{k} \qquad (3)$$

Equal-frequency binning can yield excellent results, at least in conjunction with the Naïve Bayes learning scheme, when the number of bins is chosen in a data-dependent fashion by setting it to the square root of the number of instances. [2]

- *Entropy-based discretization*

One of the supervised discretization methods, introduced by Fayyad and Irani, is called the entropy-based discretization. An entropy-based method will use the class information entropy of candidate partitions to select boundaries for discretization. [5]

The method calculates the entropy based on the class labels and finds the best split-points, so that most of the values in an interval fit the same class label – the split-points with the maximal information gain.

The entropy function for a given set S is calculated using the formula: [5]

$$Info(S) = -\sum p_i \log_2 (p_i) \qquad (4)$$

Based on this entropy measure, the discretization algorithm can find potential split-points within the existing range of continuous values. The split-point with the lowest entropy is chosen to split the range into two intervals, and the binary split is continued with each part until a stopping criterion is satisfied. [5]

Many other discretization methods may be applied on raw data, both supervised and unsupervised. Among the supervised methods we can mention Chi-Square based discretization, while a sophisticated unsupervised method is k-means discretization, based on clustering analysis.

Discretization techniques are generally considered to improve the forecasting performance of data mining techniques, particularly classification algorithms like Naïve-Bayes classifier, and, at the same time, it is thought that, choosing one discretization algorithm over another, influences the significance of the forecasting improvement.

4. Case study: Evaluating the performance of Naïve-Bayes classifiers on discretized datasets

This case study focuses on presenting the experimental results obtained by forecasting the class label for the Credit Approval dataset, using the Naïve-Bayes classifier.

The algorithm was applied to the original data, as well as to each transformed dataset, obtained by using each of the discretization methods described in this paper.

The dataset used for the experimental study concerns credit card applications and it was obtained from UCI Machine Learning Repository [6].

The Credit Approval dataset comprises 690 instances, characterized by 15 attributes and a class attribute. The values of the class attribute in the dataset can be "+" (positive) or "-" (negative) and they indicate the credit card application status for each submitted application.

This experimental study was performed using RapidMiner Software [7]. RapidMiner is a software platform, developed by the company of the same name, which provides support for all steps of the data mining process.

RapidMiner Software supports data discretization through its discretization operators. Five discretization methods are provided by RapidMiner: Discretize by Binning, Discretize by Frequency, Discretize by Size, Discretize by Entropy and Discretize by User Specification.

Among these, three methods were used during the case study, corresponding to the ones described in the paper: [7]

- *Discretize by Binning* – this operator discretizes the selected numerical attributes into user-specified number of bins. Bins of equal range are automatically generated, the number of the values in different bins may vary;
- *Discretize by Frequency* – this operator converts the selected numerical attributes into nominal attributes by discretizing the numerical attribute into a user-specified number of bins. Bins of equal frequency are automatically generated, the range of different bins may vary;
- *Discretize by Entropy* – this operator converts the selected numerical attributes into nominal attributes. The boundaries of the bins are chosen so that the entropy is minimized in the induced partitions.

During the experiment I defined a data mining process in RapidMiner. This process applied Naïve-Bayes classifier on the Credit Approval dataset, first on the original data and afterwards on the datasets obtained by applying each discretization method. I obtained performance indicators for each of these cases and I performed a comparative analysis on the results achieved without discretization and the results achieved with each discretization method.

The process flow defined in RapidMiner, for applying the Naïve-Bayes classifier, is presented in figure 1:

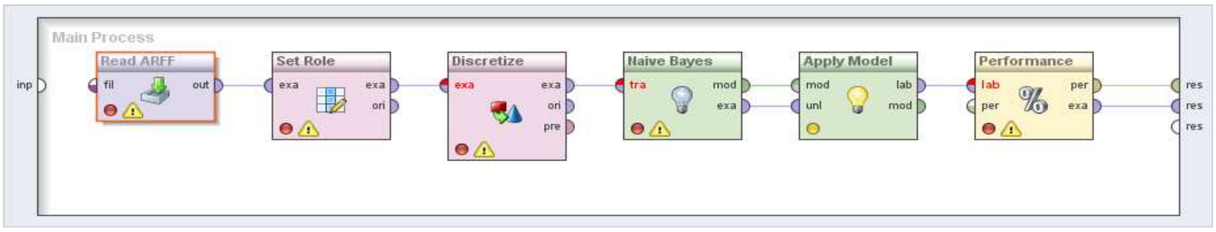

Fig. 1. Naïve-Bayes process flow in RapidMiner

The process defined for evaluating the performance of Naïve-Bayes classifier includes the following RapidMiner operators:

- *Read ARFF* – this operator is used for reading an ARFF file, in our case the file credit-a.arff;
- *Set Role* – this operator is used to change the role of one or more attributes;
- *Discretize* – this operator discretizes the selected numerical attributes to nominal attributes. RapidMiner supports applying all three discretization methods described in this paper: for equal-width discretization we can use Discretize by Binning, for equal-frequency discretization we can use Discretize by Frequency and for entropy-based discretization we can use Discretize by Entropy.
- *Naïve-Bayes* – this operator generates a Naive Bayes classification model. A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "'independent feature model". In simple terms, a Naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class (i.e. attribute) is unrelated to the presence (or absence) of any other feature [7];
- *Apply Model* – this operator applies an already learnt or trained model, in this case Naïve-Bayes, on a dataset, for prediction purposes;
- *Performance* – this operator is used for performance evaluation, through a number of statistical indicators. For my case study I chose to use accuracy, kappa and root mean squared error (RMSE).

The process described above was executed first without the Discretize operator, on the original Credit Approval dataset, and afterwards including the Discretize operator, for each discretization method. By executing the results, in each case, the process generated results for evaluating the performance of the Naïve-Bayes classifier on the dataset. Discretization methods should increase the efficiency of Naïve-Bayes classifiers. For performance evaluation I compared the results obtained in terms of accuracy, kappa and root mean square error. In order to establish which discretization technique was better for transforming the dataset, the accuracy of the classification should be as higher, as well as kappa, while the root mean square error should be as low as possible. A comparative analysis of the performance achieved, for the Naïve-Bayes classifier, with each discretization method, is shown in table 1:

Table 1. Comparative analysis of the Naïve-Bayes classifier performance with discretization methods

Discretization method	**Performance indicator**		
	Accuracy	*Kappa*	*Root mean squared error*
None	78.26%	0.546	0.432
Equal-width discretization	87.83%	0.754	0.314
Equal-frequency discretization	84.06%	0.677	0.346
Entropy-based discretization	86.96%	0.733	0.325

Based on the results in the table above, it is obvious that applying discretization methods to the dataset, before running the Naïve-Bayes classifier, has significantly improved the performance of the classification algorithm and increased the accuracy of the results obtained.

Among the discretization methods applied as part of the experimental study, equal-width discretization produces the lowest root mean squared error, 0.314, compared to the other two methods. Equal-frequency discretization generates the highest root mean squared error, of 0.346.

The prediction accuracy of the Naïve-Bayes classifier has the highest value when applying equal-width discretization - 87.83%, while equal-frequency discretization has an accuracy of 84.06%.

Performance indicator kappa compares the observed accuracy with the expected accuracy of the classifier. Thus, the best discretization method is the one generating the highest kappa. The discretization method producing the highest kappa, of 0.754, is equal-width discretization. Applying equal-frequency discretization generates the lowest kappa, 0.677.

My experiment compares the quality of the classification achieved through the Naïve-Bayes classifier, on the Credit Approval dataset, without performing discretization on the raw data and after applying discretization methods on the data. Through this experiment I am also comparing the performance obtained by applying each of the discretization methods, against each other.

Based on the previous statements, all three discretization methods improve the quality of the classification, compared to running the classification on the initial dataset. Among the methods, the best quality classification is obtained by applying equal-width discretization, opposed to equal-frequency discretization, which generates the lowest quality classification.

5. Conclusions

Discretization is a very important transformation step for data mining algorithms that can only handle discrete data. The results of my tests confirm that the performance of Naïve-Bayes classifier is improved when discretization methods are applied on the dataset used in the analysis. In this paper, I have studied the effect that applying different discretization methods, can have on the results obtained by performing a classification analysis, with the Naïve-Bayes classifier. The conclusion, based on the results obtained, is that applying the discretization methods prior to running the classification algorithm is beneficial for the analysis, since better performance indicators have been obtained on the discrete data.

Based on the experimental results, I can assert that Naïve-Bayes classifier generates better results on discrete datasets and, also, that for the particular dataset we used, the most efficient discretization method was equal-width discretization, while equal-frequency discretization was the least efficient for improving the classification efficiency and accuracy of the Naïve-Bayes classifier.

References

[1]Jiawei Han, Micheline Kamber and Jian Pei – “Data Mining: Concepts and Techniques. Third Edition”, Morgan Kaufmann Publishers, USA, 2011, ISBN 978-0-12-381479-1.

[2]Ian H. Witten, Eibe Frank and Mark A. Hall - “Data Mining: Practical Machine Learning Tools and Techniques. Third Edition”, Morgan Kaufmann Publishers, USA, 2011, ISBN 978-0-12-374856-0.

[3]Oded Maimon and Lior Rokach – “Data Mining and Knowledge Discovery Handbook. Second Edition”, Springer Publisher, London, 2010, ISBN 978-0-387-09822-4.

[4]Krzysztof J. Cios, Witold Pedrycz, Roman W. Swiniarski and Lukasz A. Kurgan – “Data Mining. A Knowledge Discovery Approach”, Springer

Publisher, USA, 2007, ISBN 978-0-387-33333-5. [5]Rajashree Dash, Rajib Lochan Paramguru and Rasmita Dash – "Comparative Analysis of Supervised and Unsupervised Discretization Techniques", International Journal of Advances in Science and Technology, Vol. 2, No. 3, 2011, ISSN 2229-5216.

[6]UCI Machine Learning Repository – Credit Approval Data Set, Available online (May 16th, 2015): https://archive.ics.uci.edu/ml/datasets/Credit+Approval.

[7]RapidMiner Software, Available online (May 14th, 2015): https://rapidminer.com/.

Ruxandra PETRE graduated from the Faculty of Cybernetics, Statistics and Economic Informatics of the Bucharest University of Economic Studies in 2010. In 2012 she graduated the Business Support Databases Master program. Currently, she is a PhD candidate, coordinated by Professor Ion LUNGU in the field of Economic Informatics at the Bucharest University of Economic Studies. Her scientific fields of interest include: Databases, Data Warehouses, Business Intelligence, Decision Support Systems and Data Mining.