Angle-Based Data Binarization Framework

Maciej Zakrzewicz
Poznan University of Technology
Poznan, Poland *maciej.zakrzewicz@put.poznan.pl*

Tadeusz Morzy†
Poznan University of Technology
Poznan, Poland

Abstract

Data binarization involves converting a continuous data attribute into a finite set of binary attributes while minimizing information loss. It plays a crucial role in feature engineering in the data mining analysis. Data binarization simplifies data, improves model training quality, enhances model performance and interpretability of results, helping in understanding complex patterns. In this paper we present an original data binarization framework, called *angle-based data binarization*, that converts continuous attributes into discrete binary attributes. The proposed framework allows not only to simplify machine learning models, but can also lead to the improvement of the accuracy of a number of well-known traditional machine learning methods. We present results of an extensive series of experiments which evaluate the efficiency of the proposed method in the area of data classification. Using popular classification algorithms, we compared classification quality achieved on source datasets with classification quality achieved on their binarized versions. We also discuss binary attribute pruning, based on elimination of attributes with poor discriminative power.

Keywords: data mining, discretization framework, continuous attributes, classification algorithms

1. Introduction

Data which are subject to data mining analysis usually come from different data sources in a variety of formats and data types. They are characterized by varying degrees of noise. Hence, a very important step in the data mining process is *data preprocessing* that begins after the collection of data and before the analytical steps of data mining. The data preprocessing phase comprises a number of different techniques that can be used in data mining applications, such as: data integration, data cleaning, data reduction, data transformation, and data conversion. The last technique involves a conversion of a data set with a particular set of attributes into a data set with another set of attributes of a different type [1].

The conversion method that plays an absolutely crucial role in feature engineering in data mining analyses is *data discretization*. By grouping similar data objects into bins, random variations or noise in the data are minimized. It also improves classification model performance since many machine learning algorithms perform better with discrete data, as they handle discrete attributes more effectively. Finally, data discretization improves the result interpretability since discretized data are easier to understand and interpret, particularly when making data-driven decisions. When data discretization results in binary attributes, it is referred as *data binarization*.

There are numerous discretization methods available in the literature. These methods can be categorized across several dimensions: static vs. dynamic, supervised vs. unsupervised, or

† Dedicated to our coauthor, Professor Tadeusz Morzy, who passed away on 21 January 2025, shortly after this paper was completed.

local vs. global, etc. Another popular classification of discretization methods distinguishes: discretization by binning, discretization by histogram analysis, entropy-based discretization, discretization by clustering. As follows from a comprehensive survey of data discretization methods [14], choosing a suitable discretization method is generally a complex problem and largely depends on user's need and available information concerning attribute values distribution and class labels. If no information concerning class labels is available, only simple unsupervised methods can be applied (e.g. binning). When the information is available, supervised methods can be applied (e.g. entropy-based discretization or error-based). The main findings of the survey as well as of some other works presenting and discussing discretization methods are rather consistent and point toward entropy-based discretization method (MDLP – minimum description length principle) being identified as the first choice.

In this paper we present an original data binarization framework, called *angle-based data binarization*, that converts numeric attribute values into binary attribute values. The framework uses angles instead of plain distance measures while converting data. The proposed binarization framework, on one side, allows to simplify a classification model due to binary attributes being used, on the other side, can lead to significant improvement of the accuracy of a number of well-known traditional classification methods (e.g. kNN algorithm) which carried out in multidimensional feature space are characterized by bias and degradation of performance.

The idea of the framework refers to the theatrical stage lighting metaphor. A theatrical stage is given, on which there are objects of various types (people, elements of scenery, etc.), and a set of spotlights illuminating these objects. The task of the lighting director is to determine the number and location of spotlights in such a way as to be able to illuminate each object (or type of an object) separately. Similarly, our task is to determine the number and location of a set of spotlights that will be able to illuminate objects of different classes separately. The spotlights will then be used to generate binary attributes that will replace the original, numeric attributes of the data objects.

To simplify the problem of determining the number and position of spotlights in a multidimensional space, we decompose it into problems of searching for the "optimal" number and position of spotlights in 2-dimensional projections of the input datasets. So, for the input data set D of dimension m we generate $\binom{m}{2}$ projections of D. For each projection of D and for each class we search for the "optimal"number and position of spotlights. A set of spotlights for a given projection and a given class is called a *spotlight layout*.

In our framework, we consider each spotlight layout as a binary variable. For each data object, if the object is illuminated by the set of spotlights belonging to a given spotlight layout, then the value of the variable associated with the spotlight layout for the object is 1, otherwise 0. The proposed binarization framework is based on three basic concepts: on the use of a set of spotlights, on the use of angles, instead of plain distances, and on the use of "one versus all" (OVA) approach. The framework consists of a few steps. For each 2-dimensional projection of the input dataset, we find the optimal spotlight layouts to let us separate objects of a given class (one class at a time - OVA). To find the optimal spotlight layout, we iteratively add spotlights to the scene and we set them in such a way as to maximize object separation accuracy. Spotlight settings involve: the position, the light beam direction and the light beam angle. The number of spotlights in a spotlight layout is dynamic since we keep adding spotlights until no further improvement is observed. After all the spotlight layouts have been generated for all 2-dimensional projections and all classes, we start the conversion of the dataset objects. For each dataset object, binary attributes are generated by observing whether the projected object is illuminated by the spotlight layout or not.

Example (**Leading example**). Let us consider the sample dataset D consisting of four data objects in 3-dimensional space belonging to two classes $c = c_1$ and $c = c_2$ (see Fig. 1 and Tab. 1). We generate $\binom{3}{2}$ 2-dimensional projections of the data set D (see Fig. 2). For each

Table 1. Original input data set D

data object	a_1	a_2	a_3	class
1	2	1	1	c_1
2	4	1	1	c_2
3	2	4	1	c_1
4	1	1	4	c_2

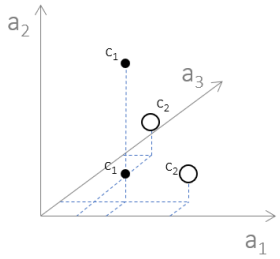

Fig. 1. Input data set D

2-dimensional projection and for each class c_i we are looking for a set of spotlights that will "best separate" objects of a given class from objects of other classes (OVA approach). By "best separates" objects of a given class from objects of other classes, we mean that the light streams of the generated set of spotlights illuminate objects of a given class and do not illuminate objects of other classes. Sample spotlights layouts for classes c_1 and c_2, for dataset projections (a_1, a_2), (a_1, a_3) and (a_2, a_3), are depicted in Fig. 3. It is easy to notice that the spotlight layout for the a_1, a_2 projection, for the class $c = c_1$, perfectly separates objects of the class c_1 from objects of c_2, while spotlight layouts for a_1, a_3 and a_2, a_3 projections illuminate objects of both classes. As we mentioned above, in our framework, we consider each spotlight layout as a binary variable. For each data object, if the object is illuminated by the spotlight layout, then the value of the variable associated with this spotlight layout for the object is 1, otherwise 0. The result of the conversion of the sample dataset D (see Tab. 1) is presented in Tab. 2. The notation $c_i a_k a_l$ stands for spotlight layout for the projection $(a_k a_l)$ for the class c_i.

2. Basic Definitions

Assume a dataset $D = \{i_1, i2, \ldots, i_N\}$, containing tuples (records) i_i in the form $(a_1, a_2, \ldots, a_n, c)$, where a_j is an attribute and c is a class label. Dataset **binarization** function $b : D_a \rightarrow D_b$ maps each tuple $(a_1, a_2, \ldots, a_n, c) \in D_a$, where a_j is a normalized continuous attribute $(a_j \in \langle -1; 1 \rangle)$, into a tuple $(b_1, b_2, \ldots, b_m, c) \in D_b$, where b_j is a bi-

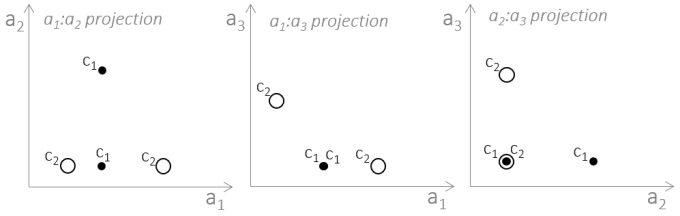

Fig. 2. 2-attribute projections of the dataset D

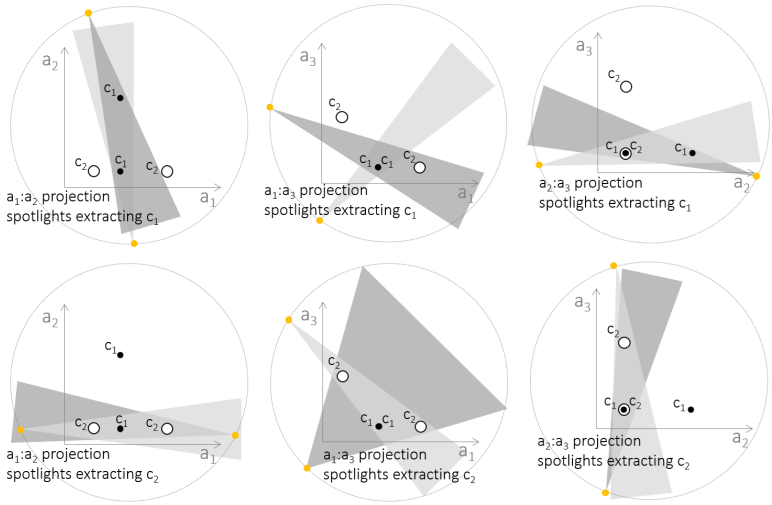

Fig. 3. Spotlights layouts for projections

Table 2. Converted data set.

data object	$c_1a_1a_2$	$c_1a_1a_3$	$c_1a_2a_3$	$c_2a_1a_2$	$c_2a_1a_3$	$c_2a_2a_3$	class
1	1	1	1	1	1	1	c_1
2	0	0	1	1	1	1	c_2
3	1	1	1	0	1	0	c_1
4	0	0	0	1	1	1	c_2

nary attribute ($b_j \in \{0, 1\}$). Dataset **projection** function $\pi_{ak,al} : D_a \rightarrow D_p$ maps each tuple $(a_1, a_2, \ldots, a_n, c) \in D_a$ into a tuple $(a_k, a_l, c) \in D_p$. A **spotlight** s_i is represented as a triple $(center_i, beam\_left\_edge_i, beam\_right\_edge_i)$, where $center_i$ is the angle of the spotlight on the circle around the stage, $beam\_left\_edge_i$ and $beam\_right\_edge_i$ represent angles of the light beam edges on the circle around the stage (see Fig. 4). An object o is illuminated by the spotlight s_i, denoted as $o \in s_i$, if the object is covered by the spotlight shape (angle). A **spotlight layout** l is a set of spotlights $\{s_1, s_2, \ldots, s_k\}$, placed around the stage in order to extract objects of a specific class. In order to assess extraction efficiency of a given spotlight layout, we introduce **spotlight layout discriminativeness** measure $d\left(l, \pi_{ak,al}\left(D\right), c\right)$, where l is the spotlight layout, D is the dataset, a_k and a_l are projection attributes, and c is a class label. The measure is based on the concept of *statistical accuracy* and *imbalanced classes weighting*, and is defined as follows:

$$d\left(l, \pi_{ak,al}\left(D\right), c\right) = \frac{|i \in \pi_{ak,al}(D) : class(i) = c \text{ and } \forall_{s_k \in l}\ i \in s_k| \cdot w_{TP}(c)}{|D|} +$$

$$+ \frac{|i \in \pi_{ak,al}(D) : class(i) \neq c \text{ and } \forall_{s_k \notin l}\ i \notin s_k| \cdot w_{FP}(c)}{|D|}$$

where:

$$w_{TP}(c) = \frac{|D|}{2 \cdot |\{i : class(i) = c\}|}$$

$$w_{FP}(c) = \frac{|D|}{2 \cdot |\{i : class(i) \neq c\}|}$$

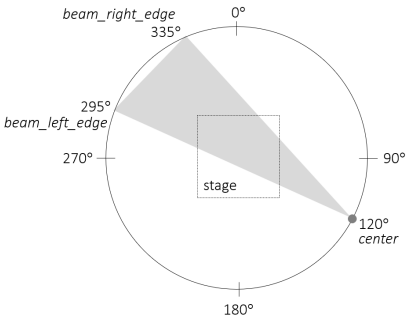

Fig. 4. Sample spotlight $s_i = (120, 295, 335)$

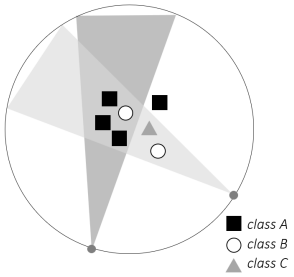

Fig. 5. Sample spotlight

Example. Consider the dataset projection $\pi(D)$ and 2-spotlight layout l in Fig. 5. In order to evaluate the spotlight layout discriminative measure for *classA*, we calculate $w_{TP}(classA) = 7/(2\cdot 4) = 7/8$, $w_{FP}(classA) = 7/(2\cdot 3) = 7/6$, and finally $d(l, \pi(D), classA) = (3\cdot(7/8) + 3\cdot(7/6))/7 = 0.875$ (or 87.5%).

□

A **binarization model** $m(D)$ is a set of spotlight layouts generated for each class $c \in C$ and for each projection $\pi_{ak,al}(D)$, i.e. $m(D) = \{l(c_1, a_1, a_2), l(c_1, a_1, a_3), \ldots, l(c_k, a_{n-1}, a_n)\}$, where $l(c_m, a_n, a_p)$ is a spotlight layout for class c_m and dataset projection $\pi_{an,ap}(D)$. Notice that $|m(D)| =$ number of classes $\cdot$ number of attributes $\cdot$ (number of attributes $- 1)/2$.

The dataset binarization function $b(D)$ based on the binarization model $m(D)$ maps each tuple $(a_1, a_2, \ldots, a_n, c)$ into $(b_{1,1,2}, b_{1,1,3}, \ldots, b_{k,n-1,n}, c)$ in the following way:

$b_{1,1,2} = 1$ if (a_1, a_2) is illuminated by all spotlights in $l(c_1, a_1, a_2)$, 0 otherwise,
$b_{1,1,3} = 1$ if (a_1, a_3) is illuminated by all spotlights in $l(c_1, a_1, a_3)$, 0 otherwise,
...
$b_{k,n-1,n} = 1$ if (a_{n-1}, a_n) is illuminated by all spotlights in $l(c_k, a_{n-1}, a_n)$, 0 otherwise.

3. The Data Binarization Method

The proposed data binarization method is a two-phase approach: first build a binarization model for the dataset, then use the binarization model to map dataset tuples into binary tuples. In order to build the binarization model, we determine optimal spotlight layouts for each class and for each 2-attribute dataset projection. The optimal spotlight layouts are generated by iteratively adding new spotlights and finding positions and beam angles that maximize spotlight layout discriminativeness. The spotlights are added to the spotlight layout as long as they are improving the discriminativeness, therefore different spotlight layouts may have different numbers of spotlights.

Phase I: Generate the Binarization Model m(D)

```
foreach class c, foreach pair (a_k, a_l) of attributes of D:
   start with empty spotlight layout l = ∅
   iteratively add new spotlights to the l,
      positioning them around the stage to maximize d(l, π_{ak,al}(D), c)
   stop adding spotlights when d(l, π_{ak,al}(D), c) does not get improved any more
```

In the second phase, we process tuples from the source dataset and generate new tuples by executing the binarization function based on the generated binarization model.

Phase II: Preprocess dataset D using the Binarization Model m(D)

```
foreach tuple i ∈ D:
   foreach spotlight layout l:
      if i is illuminated by all spotlights from l
         then output 1;
         else output 0;
```

Finding optimal spotlight layouts. The most important task in our method is to find the "optimal" spotlight layouts for all classes. The construction of a spotlight layout for a given class is performed as follows. We are moving the first spotlight around the stage to find a spotlight center angle that provides the best spotlight layout discriminativeness. Initially, the spotlight beam edges are configured to exactly embrace all the class' objects. Once the first spotlight has been set, we add the second spotlight and follow the same method to find its optimal location. We keep adding spotlights until they no more improve the spotlight layout discriminativeness. After the spotlights' center angles have been determined, we revisit each spotlight and try to narrow the beam edges in hope to further improve the spotlight layout discriminativeness. The process ends after all the beams have been optimized. Then, we perform the same procedure for the next class. Constructing spotlight layouts for all classes completes phase I of the binarization method. All constructed spotlight layouts constitute the binarization model $m(D)$.

Spotlight distance. The spotlights' distances from the stage may influence the observed discriminativeness. Our current implementation of the binarization method performs best if all spotlights are placed on a circle with radius equal to double or triple the stage width. We believe that using nonuniform spotlight distances within a spotlight layout would also be beneficial to maximize discriminativeness.

Alternative spotlight layout discriminativeness measures. Our discriminativeness measure is based on the concept of statistical accuracy and imbalanced classes weighting. Alternative measures can also be deployed. We have experimentally verified measures inspired by F-score, prevalence, true positive rate, positive likelihood ratio, and informedness, however, the accuracy-based indicator resulted in the best binarization quality.

Alternative binarization functions. When mapping source dataset tuples into binary tuples, a projected object must be illuminated by all spotlights in a spotlight layout in order to be mapped to 1. However, other approaches can also be considered. For example, we may require that objects are illuminated only by majority of spotlights to result in 1s. Spotlight order can also be taken into account as sort of a weight, as spotlights added first seem to have stronger influence on the discriminativeness than those added later.

Dimensionality reduction in the binarized dataset. The binarized dataset will have more attributes than the source dataset (number of classes x number of source attributes x (number of source attributes - 1) / 2). To reduce the number of resulting binary attributes, we can add a

Table 3. Summary of Data Sets.

Data set	Records	Features	Classes	Binary Features
Rice	3810	8	2	42
Abalone	4177	8	3	84
Yeast	1484	8	10	280
Wine Quality White	4873	12	5	385
Breast Cancer Wisconsin	699	9	2	72

pruning step to the spotlight layout generation phase. Spotlight layouts that provide discriminativeness below a predefined threshold will be eliminated from the binarization model. We observed that eliminating 10-20% of the weakest binary attributes would even improve the quality of the binarization.

Datasets containing non-numerical attributes. Our binarization method assumes that all source dataset attributes are continuous and normalized because we need to project dataset tuples as 2-dimensional objects in Cartesian coordinate system. To use our method for datasets that contain categorical attributes, we recommend that binarization covers the continuous attributes only, leaving the categorical ones intact. The resulting dataset will be binarized "partially", but we will still be able to benefit from its characteristics. Another way to deal with categorical attributes is to map them to n additional binary attributes, using one-hot encoding.

4. Experimental Evaluation

Although the main goal of the paper is to propose a method for converting numerical data into binary data, we did perform an extensive series of experiments in order to evaluate the efficiency of the method in the area of data classification. Using popular classification algorithms provided by Weka framework [6], we compared classification quality achieved on source datasets with classification quality achieved on their binarized versions. No dimensionality reduction was applied.

We used 5 popular data sets obtained from the UCI repository (see Tab. 3). The data sets had various sizes and varying numbers of features, they were not subjected to any pre-processing techniques except for the deletion of records containing incomplete data or data in an erroneous format. In addition, the Table shows the numbers of binary attributes generated by the binarization (Binary Features). The experiments were performed using The Waikato Environment for Knowledge Analysis (WEKA) version 3.8.6, using various classification algorithms, including: Random Forest, Random Tree, J48 (C4.5), RepTree, IBk (kNN), Simple Logistic (linear logistic regression), AdaBoost M1, and Naive Bayes, all run with default parameter settings [4], [8], [7]. The classification model performance was evaluated by 10-fold cross-validation.

To better understand the quality of our proposed angle-based binarization method, we also ran a series of experiments which evaluated the usefulness and efficiency of the proposed method in comparison to known and widely used discretization methods in the area of data classification: equi-frequency binning and entropy-based discretization (MDLP), a state-of-the-art discretization algorithm [12]. Finally, we also present and discuss our results on binary attribute pruning, based on elimination of attributes with poor discriminative power.

Experimental Results – Original Dataset vs Binarized Dataset Representation. The results are shown in Tab. 4, using bold formatting to emphasize best results. The column *Original Dataset* shows the quality of data classification achieved with the original dataset. The column *Angle-based Binarization* shows the quality of data classification achieved with the binarized

dataset. It is worth noting that in most cases the binarized dataset representation allows us to reach higher accuracy values. The best classifiers for binarized datasets were not necessarily the same as the best classifiers discovered for original datasets as not every classification algorithm benefits from binary dataset format. The observed number of spotlights in optimal spotlight layouts varied between 2 and 13, depending on datasets and attributes used for projection.

Experimental Results – Comparison of Discretization Methods. The column *One-Hot Equi Freq* of Tab. 4 presents the quality of data classification observed on the dataset discretized using equi-frequency binning and one-hot encoding. Each continuous attribute was discretized into one of ten equal frequency bins, and then it was converted to ten binary attributes using one-hot encoding. The number of bins (ten) was selected in order to get a similar number of binary columns for the solutions being compared. We were able to observe much worse results compared to those from our angle-based binarization. Finally, the column *Entropy Discr* shows the quality of data classification achieved on a dataset discretized using MDLP version entropy-based discretization algorithm. In 32 test cases out of 45, our angle-based binarization method outperformed the entropy-based discretization.

Experimental Results – Attribute Pruning. We also analyzed effects of binary attribute pruning, based on elimination of spotlight layouts with poor discriminativeness (below specified threshold). Tab. 5 shows classification quality (accuracy) achieved using binarized datasets with different pruning settings. We have not observed significant degradation of classification quality even for heavily pruned binarized datasets, eg. for Breast Cancer Wisconsin dataset, pruning of 84% of binary attributes resulted in classification quality drop from 97.90% to only 97.52% (Naive Bayes).

Analysis of results. Summing up the presented results of the experiment, the following can be stated. First, the proposed framework improves the classification quality for most of analyzed data sets and classification methods, outperforming the commonly used discretization methods. Second, the computational complexity of data binarization is not expensive as the number of required spotlights per a spotlight layout was relatively low. Third, prunning of binary attributes helped significantly reduce the dimensionality without significant degradation of classification quality.

5. Related Work

There exists a very extensive literature on data preprocessing, particularly, related to data integration, data cleaning, feature selection or data transformation. A broad discussion of the basic methods of data preprocessing can be found in [1]. A detailed survey of various data cleaning techniques is provided in [10]. Data integration techniques, such as record linkage and data fusion, are discussed in [2].

Many discretization methods were described in the literature. A comprehensive survey of discretization methods, their comparison, effect on classification is given in Liu et al. [14]. The simplest method to discretize a continuous-valued attribute is binning. This top-down unsupervised splitting technique involves dividing attribute continuous values into a specified number of bins. Equal-width or equal-frequency binning can be applied, followed by replacing each bin value with the bin mean or median. It is possible to improve the quality of binning by using class labels information to adjust boundaries of neighboring bins [3], [9]. Discretization by histogram analysis, like binning, is an unsupervised technique that partitions attribute continuous values into disjoint ranges (buckets or bins). Histograms are effective for approximating sparse, dense, skewed, and uniform data. Various partitioning rules can be used to define histograms. This kind of discretization is commonly used by query optimizers in databases [17].

Entropy-based discretization method uses the entropy measure to determine the "best" partitions of the attribute values for discrete intervals. Different variants of the entropy-based dis-

Table 4. Classification quality on binarized data (% of correctly classified instances)

Dataset	Classifier	Original Dataset	One-Hot Equi-Freq	Entropy Discr	Angle-based Binarization
Rice	Random Forest	91.88	90.79	91.18	92.23
	Random Tree	88.80	88.94	91.46	92.26
	J48	92.93	91.67	92.96	**92.93**
	RepTree	92.30	91.74	92.79	92.37
	kNN	88.38	90.06	91.63	92.61
	Simple Logistic	92.61	91.74	93.00	92.47
	AdaBoost M1	**92.96**	84.74	92.61	92.33
	Naive Bayes	91.46	91.11	91.49	**92.93**
	Bagging	92.51	91.81	92.79	92.54
Abalone	Random Forest	54.31	51.66	52.71	54.92
	Random Tree	48.66	50.06	52.49	54.02
	J48	52.14	53.32	53.64	54.79
	RepTree	54.05	52.33	54.25	54.95
	kNN	50.57	50.96	53.45	54.12
	Simple Logistic	**55.59**	52.49	52.84	**56.58**
	AdaBoost M1	53.70	44.19	48.02	55.04
	Naive Bayes	51.79	51.69	52.30	51.66
	Bagging	54.76	53.10	53.54	56.42
Breast Cancer	Random Forest	**97.71**	97.14	96.76	97.33
	Random Tree	95.42	92.94	94.66	97.52
	J48	94.66	93.70	96.00	96.38
	RepTree	95.23	93.89	95.42	96.95
	kNN	96.37	96.76	96.76	**97.90**
	Simple Logistic	97.14	96.18	96.00	96.76
	AdaBoost M1	94.66	95.61	95.80	97.14
	Naive Bayes	96.37	96.95	97.90	**97.90**
	Bagging	97.33	94.66	96.18	97.14
Wine Quality	Random Forest	**65.86**	65.48	57.24	**68.03**
	Random Tree	57.68	56.36	53.87	58.28
	J48	56.69	53.39	55.41	57.84
	RepTree	55.27	52.68	54.34	52.09
	kNN	59.79	61.72	55.52	61.24
	Simple Logistic	53.71	53.34	54.94	58.23
	AdaBoost M1	45.06	45.82	44.59	51.96
	Naive Bayes	45.17	48.57	49.74	39.72
	Bagging	60.25	57.96	55.84	61.40
Yeast	Random Forest	59.75	54.81	54.72	**61.28**
	Random Tree	47.62	43.49	54.27	49.60
	J48	53.19	50.13	57.77	53.73
	RepTree	56.78	53.55	57.05	54.36
	kNN	51.93	47.44	54.45	51.84
	Simple Logistic	57.23	56.06	56.33	61.19
	AdaBoost M1	40.25	37.29	40.25	39.53
	Naive Bayes	56.24	55.89	55.62	57.86
	Bagging	**60.56**	55.08	56.87	60.65

Table 5. Effects of binary attribute pruning

Dataset	Classifier	Pruning threshold	Accuracy
Yeast	Random Forest	no pruning	61.28
		threshold = 0.6 (22% pruned)	61.10
		threshold = 0.65 (40% pruned)	59.57
		threshold = 0.75 (56% pruned)	55.26
	Naive Bayes	no pruning	57.86
		threshold = 0.6 (22% pruned)	58.22
		threshold = 0.65 (40% pruned)	57.50
		threshold = 0.75 (56% pruned)	54.36
Breast Cancer Wisconsin	Random Forest	no pruning	97.33
		threshold = 0.6 (22% pruned)	97.90
		threshold = 0.65 (40% pruned)	97.90
		threshold = 0.75 (56% pruned)	96.38
	Naive Bayes	no pruning	97.90
		threshold = 0.6 (22% pruned)	97.52
		threshold = 0.65 (40% pruned)	97.90
		threshold = 0.75 (56% pruned)	97.52

cretization method can be found in literature. These variants differ either in the choice of the cut points or in the stopping criterion of the discretization procedure. Entropy-based discretization method with $C4.5$ algorithm is described in [18]. In [5] the minimum description length principle (MDLP) is employed to determine a stopping criterion for the discretization procedure.

In [16] an error-based discretization algorithm is presented. The algorithm discretizes attribute continuous values by producing a set of intervals that result in the minimum error on the training dataset. The discretization methods presented above do not exhaust the list of possible approaches. ChiMerge and Chi2, presented in [11], [15], are supervised discretization algorithms that both apply the χ^2 measure to conduct a significance test on the relationship between the attribute values and the class labels. The χ^2 statistic determines the similarity of adjacent intervals of an attribute based on some significance level. The discretization by cluster analysis is briefly presented in [7].

One previous research effort is also partially related to our method. In [13] a new approach to outliers detection in high-dimensional data space was presented based on the use of variance of angles between objects in a dataset. The use of angle measure eliminates the phenomenon of the curse of dimensionality, which is a problem with using distance measures in high-dimensional data. Our framework refers to the presented idea of using angles instead of plain distances in multidimensional space.

However, none of the prior studies propose the use of angles to discretize values of continuous attributes.

6. Conclusions and Summary

In this paper we have introduced a new data conversion framework to handle one of the data preprocessing steps in the data mining process. The concept was inspired by the theatrical stage lighting metaphor, where spotlights were used to extract scene objects. We perform projections of multidimensional tuples into 2-dimensional points and generate optimal spotlight layouts in order to best extract points belonging to a specific class. Then, for each dataset tuple, the

spotlight layouts are used to produce binary output attributes with the value of "1" (if a projected point belongs to fully illuminated area of the stage) or "0" (otherwise). In result, we obtain a binary-only representation of the original dataset.

We have showed that the binarized datasets can not only simplify the classification process due to binary attributes being used, but can also improve the accuracy of well-known classification methods when compared to the original dataset scenario. Our extensive experiments demonstrated that the binarized datasets can easily replace the original datasets in the data mining process and that the quality of achieved data mining results is far better than provided by e.g. straightforward binning-based one-hot conversion of numerical attributes to binary attributes. The key reasons why the proposed angle-based binarization method leads to better classification quality include: operating on 2-dimensional tuple projections instead of on the original multidimensional representations (or on single attributes only), using 2-dimensional angles instead of multidimensional distances, and one-versus-all approach to extract single class' objects from objects belonging to all other classes.

We are also researching other application areas of the theatrical stage lighting metaphor, which are out of the scope of this paper - e.g. using spotlight layouts as a standalone classification model.

References

1. Aggarwal, C.C.: Data Mining - The Textbook. Springer (2015)
2. Dong, X., Srivastava, D.: Big data integration. In: Proceedings of the VLDB Endowment. vol. 6, pp. 1245–1248 (2013)
3. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: 12th Int. Conf. on Machine Learning. pp. 194–202 (1995)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification (2nd Edition). Wiley-Interscience, 2 edn. (2000)
5. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: 13th Int. Joint Conference On Artificial Intelligence. pp. 1022–1029 (1993)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The weka data mining software: An update. SIGKDD Explorations 11, pp. 10–18 (11 2008)
7. Han, J., Kamber, M., Pei, J.: Data mining concepts and techniques, third edition (2012)
8. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference and prediction. Springer, 2 edn. (2009)
9. Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. Machine Learning 11, pp. 63–90 (1993)
10. Ilyas, F., Chu, X.: Data Cleaning. ACM Books (2019)
11. Kerber, R.: Chimerge: discretization of numeric attributes. In: Proceedings of the Tenth National Conference on Artificial Intelligence. p. 123–128. AAAI'92, AAAI Press (1992)
12. Kohavi, R., Sahami, M.: Error-based and entropy-based discretization of continuous features. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. p. 114–119. KDD'96, AAAI Press (1996)

13. Kriegel, H.P., Schubert, M., Zimek, A.: Angle-based outlier detection in high-dimensional data. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 444–452. KDD '08, Association for Computing Machinery, New York, NY, USA (2008)

14. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: An enabling technique. Data Mining and Knowledge Discovery 6(4), pp. 393–423 (2002)

15. Liu, H., Setiono, R.: Chi2: Feature selection and discretization of numeric attributes. In: Proceedings of the International Conference on Tools with Artificial Intelligence. pp. 388 – 391 (1995)

16. Maass, W.: Efficient agnostic pac-learning with simple hypothesis. In: Proceedings of the Seventh Annual Conference on Computational Learning Theory. p. 67–75. COLT '94, Association for Computing Machinery, New York, NY, USA (1994)

17. Oracle: Oracle database 19c sql tuning guide (2024), `https://docs.oracle.com/`

18. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)