Binarization of features based on frequency discretization for clustering tasks

Igor Masich[1,2*], *Guzel* Shkaberina[1,2], and *Danila* Masich[3]

[1]Laboratory "Hybrid Methods of Modeling and Optimization in Complex Systems", Siberian Federal University, 660041 Krasnoyarsk, 79 Svobodny Prospekt, Russia
[2]Institute of Informatics and Telecommunications, Reshetnev Siberian State University of Science and Technology, 660037 Krasnoyarsk, 31 Krasnoyarsky Rabochy Prospekt, Russia
[3]Physics and Mathematics School, Siberian Federal University, 660041 Krasnoyarsk, 79 Svobodny Prospekt, Russia

Abstract. This paper explores the transformation of heterogeneous features, including continuous data, into binary form using frequency discretization. This method is particularly beneficial for clustering tasks, as binary features simplify the interpretation of results using logical expressions. In unsupervised learning, where class labels are unknown, we propose a binarization approach that converts continuous features into binary values based on their frequency distribution. Our experiments show that this technique not only preserves essential information but also improves clustering quality, as measured by the Rand Index, compared to known groupings of industrial product batches. The method reduces noise, simplifies the feature space, and enhances cluster interpretability. Among various distance metrics, the best results were achieved using Cosine distance. These findings highlight the potential of frequency discretization for improving clustering outcomes.

1 Introduction

Original features that describe observed objects can be either numerical or categorical, and in many cases, features of different types are present simultaneously. In such scenarios, it is often beneficial to transform features of varying types into a single, unified format—particularly into binary form. This transformation can be useful for various reasons. Binary features are particularly advantageous when applying methods that describe clusters or subgroups of objects using logical expressions. These methods are essential in conceptual clustering [1] and when there is a need to interpret clustering results clearly and comprehensibly.

When performing cluster analysis, the initial classification of objects into specific groups or classes is unknown. Consequently, certain binarization techniques typically used in supervised learning are not applicable in this context. Instead, we propose a binarization approach based on frequency discretization of continuous (real-valued) features for such unsupervised scenarios.

* Corresponding author: i-masich@yandex.ru

Our experiments demonstrate that this method not only preserves the essential information contained in the original continuous features for clustering purposes but, in many cases, also enhances the overall quality of the clustering results. This is achieved by reducing noise, simplifying feature space, and making the clusters more interpretable, all of which are critical for practical applications in unsupervised learning tasks.

2 Frequency discretization for clustering

The k-means problem [2, 3] can be described as finding set of k cluster centers (or centroids) $X_1 \ldots X_k$ in a d-dimensional space with the minimal sum of squared distances from them to the given N points A_i (SSE, sum of squared errors):

$$\operatorname{argmin} F(X_1,\ldots,X_k) = \sum_{i=1}^{N} \min_{j\in\{1,k\}} \left\|X_j - A_i\right\|^2 . \quad (1)$$

The algorithm 1 of the same name sequentially improves a known solution, looking for a local minimum of (1). This is a simple and fast algorithm applicable to the widest class of problems. The algorithm 1 has limitation: we need to preset the number of k groups into which objects are clustered. The result is highly dependent on the initial decision, classically chosen at random.

Algorithm 1 k-means (ALA procedure – alternating location and allocation [4])

Required: data vectors $A_1 \ldots A_N$, k initial cluster centers $X_1 \ldots X_k$.
do
Step 1. For each of centers X_i, compose clusters C_i of data vectors so that each of the data vectors is assigned to the nearest center.
Step 2. Calculate new center X_i for each of clusters.
While Steps 1-2 lead to any modifications.

Feature transformation into binary form is a crucial step in data preprocessing for machine learning. This process enables the effective use of algorithms that require numerical data representation. In our work, threshold binarization is applied to transform numerical (continuous) features. Binarization of objects involves setting thresholds for numerical features to produce logical values, i.e., binarizing data by assigning feature values as either 0 or 1 based on a specified threshold.

The idea behind binarization is as follows. We sort the data points of a continuous feature in ascending order:

$$a_{i_1 j} \le a_{i_2 j} \le \cdots \le a_{i_N j}. \quad (2)$$

The range of values for each continuous feature is divided into k+1 equal-frequency intervals (bins), where k represents the number of interval boundaries (thresholds) $t_s^j, s = 1, \ldots, k, j = 1, \ldots, d$ (where d is the number of features or dimensions of the object). In our work, the threshold values are determined using frequency discretization, where each bin contains approximately the same number of data points.

Frequency discretization is a technique used to convert continuous features into discrete intervals by splitting the data based on the frequency of occurrence within the dataset. The method is based on the idea that each interval contains approximately the same number of data points, creating balanced bins that reflect the distribution of the feature values.

Unlike equal-width discretization, which divides the range of the feature into equal-sized intervals, frequency discretization focuses on grouping data points based on how frequently they occur within certain ranges.

This method is especially useful in scenarios where maintaining the underlying distribution of data is important, such as in clustering or classification tasks. By dividing features in this way, you can retain key information about the structure of the data while simplifying the feature space for analysis.

The value of each feature of an object, which is greater than or equal to the threshold value, is mapped to 1, while values less than the threshold are mapped to 0. Thus, the transformation of a continuous feature a_{ij} of an object into a binary type a_binary_{ij} through the specified threshold values t_s^j is defined as [5]:

$$a\_binary_{ij}^s = \begin{cases} 1, & a_{ij} \geq t_s^j, \\ 0, & a_{ij} < t_s^j, \end{cases} \quad (3)$$

where i is the index of the object, $i = 1, \dots, N$, and j is the index of the feature of the object, $j = 1, \dots, d$. Thus, for each continuous feature describing the recognition object, we have an ordered set of potential thresholds $t_1^j, t_2^j, \dots, t_k^j$ arranged in ascending order.

Data normalization is a mandatory step in the transformation of initial data in multidimensional statistical analysis tasks, especially if the data differ in units of measurement. There are various ways to normalize data [6], in our study the transformation is given by:

$$x\_scaled_{ij} = \frac{x_{ij} - min}{max - min}, \quad (4)$$

where x_scaled_{ij} is the normalized value, x_{ij} is real value (original value), min is the minimum value of the feature, max is the maximum value of the feature. By applying this formula (4) to each data point and feature, they will be scaled to a range with a maximum value of 1 and a minimum value of 0.

3 Computational experiments

For the experiments, we considered the sample of industrial products: Transistors 2P771A and Single-channel transistor optocouplers 3OT122A.

Transistors 2P771A is a dataset containing 182 transistors, described by 12 real features, representing the results of test electrical effects. The dataset contains a mixture of two homogeneous product groups.

Single-channel transistor optocouplers 3OT122A is a dataset containing 278 optocouplers, described by 14 real features, representing the results of test electrical effects. The dataset contains a mixture of two homogeneous product groups.

We used various types of distance measures in the experiments: Euclidean distance (EuD), Manhattan distance (ManD), Square Euclidean distance (SEuD), Cosine distance (CosD), Chebyshev distance (ChD). Each experiment was run 30 times. The best value (maximum) of the rand index is shown in Tables 1 and 2.

The algorithm was implemented in Python. The following test system was used for computational experiments: 11th Gen Intel(R) Core (TM) i7-1165G7 2.80GHz CPU, 16 GB RAM.

In this research, we compared the results of the experiment performed with k-means algorithm, which uses initial dataset with real features (initial dataset (normalized)) and binary features obtained through specified thresholds: binar_2, binar_3, binar_4, binar_5, binar_6, binar_7, binar_8. In this study the initial solution is chosen randomly.

The computational experiment demonstrated that, in most cases, the use of binary features in the k-means algorithm enhances the Rand Index [7] for the separation of homogeneous product groups (as shown in Table 1 and Table 2).

Table 1. Rand Index for transistors 2P771A

dataset	EuD	ManD	SEuD	CosD	ChD
initial dataset (normalized)	0.876	0.896	0.876	0.770	0.848
binar_2	0.915	0.905	0.915	0.967	0.926
binar_3	0.926	0.926	0.926	0.957	0.967
binar_4	**0.936**	**0.936**	**0.936**	**0.967**	**0.946**
binar_5	**0.936**	**0.936**	**0.936**	0.946	0.936
binar_6	**0.936**	**0.936**	**0.936**	0.957	0.926
binar_7	**0.936**	**0.936**	**0.936**	0.957	**0.946**
binar_8	**0.936**	**0.936**	**0.936**	0.946	0.936

For binar_4 – binar_8 datasets, the Rand Index has maximum values for Euclidean distance (EuD), Manhattan distance (ManD), Square Euclidean distance (SEuD). For binar_4 and binar_7 datasets, the Rand Index has maximum values for Chebyshev distance (ChD). Also, for binar_4 dataset, the Rand Index has maximum values for Cosine distance (CosD).

Table 2. Rand Index for Single-channel transistor optocouplers 3OT122A

dataset	EuD	ManD	SEuD	CosD	ChD
initial dataset (normalized)	0.499	0.505	0.499	0.499	0.512
binar_2	0.629	**0.709**	0.629	0.617	0.695
binar_3	0.519	0.534	0.519	0.526	0.605
binar_4	0.677	0.515	**0.677**	0.739	0.689
binar_5	0.667	0.642	**0.667**	0.709	0.698
binar_6	0.499	0.671	0.499	0.779	0.675
binar_7	**0.695**	0.689	0.695	**0.800**	**0.707**
binar_8	0.505	0.523	0.505	0.696	0.700

For the binar_2 dataset, the Rand Index reaches its highest values with the Manhattan distance (ManD). For the binar_4 and binar_5 datasets, the highest Rand Index values are observed with the Squared Euclidean distance (SEuD). For the binar_7 dataset, the Rand Index achieves maximum values with Euclidean distance (EuD), Cosine distance (CosD), and Chebyshev distance (ChD).

4 Conclusion

Binarization using frequency discretization not only allows for the conversion of original heterogeneous, especially continuous, features into binary ones, but also improves clustering quality, as measured by the Rand Index, when compared to known groupings of homogeneous classes (batches) of industrial products. This demonstrates that, in the cases studied, the information contained in the original features is not lost, and the binarization process helps to reveal the underlying cluster structure. The experiment was conducted using various distance metrics, with the best result (maximum Rand Index) obtained using Cosine distance. In future research, we plan to apply these results in conceptual clustering methods to achieve interpretable clustering outcomes.

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Grant No. 075-15-2022-1121).

References

1. R.S. Michalski, Int. J. Policy Anal. Inf. Syst. **4,** 219-244 (1980)
2. Y. Li, H. Wu, Physcs Proc., **25**,1104-1109 (2012)
3. S.P. Lloyd, IEEE T. Inform. Theory, **28**, 129-137 (1982)
4. L. Kazakovtsev, I. Rozhnov, G. Shkaberina, IJ-AI, **19**, 152-194 (2021)
5. I. Masich, N. Rezova, G. Shkaberina, S. Mironov, M. Bartosh, L. Kazakovtsev, Algorithms, **16**, 246 (2023)
6. D. Singh, B. Singh, Appl. Soft Comput., **97**, 105524 (2020)
7. W.M. Rand, J. Am. Stat. Assoc., **66**, 846-850 (1971)