A New Nearest Neighbor Median Shift Clustering for Binary Data

Gael Beck[1,2], Mustapha Lebbah[1(✉)], Hanene Azzag[1], and Tarn Duong[1]

[1] Computer Science Laboratory of Paris North (LIPN, CNRS UMR 7030), Sorbonne Paris Nord University, 93430 Villetaneuse, France
{beck,mustpha.lebbah,hanane.azzag,duong}@lipn.univ.paris13.fr
[2] HephIA SAS, 30, rue de Gramont, 75002 Paris, France
Gael@hephia.com

Abstract. We describe in this paper the theory and practice behind a new modal clustering method for binary data. Our approach (BinNNMS) is based on the nearest neighbor median shift. The median shift is an extension of the well-known mean shift, which was designed for continuous data, to handle binary data. We demonstrate that BinNNMS can discover accurately the location of clusters in binary data with theoretical and experimental analyses.

Keywords: Density gradient ascent · Hamming distance · Mean shift

1 Introduction

The goal of clustering (unsupervised learning) is to assign cluster membership to unlabeled candidate points where the number and location of these clusters are unknown. Clustering is an important step in the exploratory phase of data analysis, and it becomes more difficult when applied to binary or mixed data. Binary data occupy a special place in many application fields: behavioral and social research, survey analysis, document clustering, and inference on binary images.

Clusters are formed usually from a process that minimizes the dissimilarities inside the clusters and to maximizes the dissimilarities between clusters. A popular clustering algorithm for binary data is the k-modes [8], and it is similar to the k-means clustering [14] wherein the modes are used instead of the means for the prototypes of the clusters. Other clustering algorithms have been developed using a matching dissimilarity measure for categorical points instead of Euclidean distance [12], and a frequency-based method to update modes in the clustering process [10].

In this paper, we focus on the mean shift clustering [5,6], which is another generalization of the k-means clustering. Mean shift clustering belongs to the class of modal clustering methods where the arbitrarily shaped clusters are defined in terms of the basins of attraction to the local modes of the data density, created by the density gradient ascent paths. In the traditional characterization of the mean

I. Farkaš et al. (Eds.): ICANN 2021, LNCS 12895, pp. 101–112, 2021.
https://doi.org/10.1007/978-3-030-86383-8_8

shift, these gradient ascent paths are computed from successive iterations of the mean of the nearest neighbors of the current prototype. Due to its reliance on mean computations, it is not suited to be directly applied to binary data. Our contribution is the presentation of a modified mean shift clustering which is adapted to binary data. It is titled Nearest Neighbor Median Shift clustering for binary data (BinNNMS). The main novelty is the that the cluster prototypes are updated via iterations on the majority vote of their nearest neighbors. We demonstrate that this majority vote corresponds to the median of the nearest neighbors with respect to the Hamming distance [7].

The rest of the paper is organized as follows: Sect. 2 introduces the traditional mean-shift algorithm for continuous data, Sect. 3 presents our new median shift clustering procedure for binary data BinNNMS, and Sect. 4 describes the results of the BinNNMS compared to the k-modes clustering.

2 Nearest Neighbor Mean Shift Clustering for Continuous Data

The mean shift clustering proceeds in an indirect manner based on local gradients of the data density, and without imposing an ellipsoidal shape to clusters or that the number of clusters be known, as is the case for k-means clustering. For a candidate point $\boldsymbol{x}$, the mean shift method generates a sequence of points $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \dots\}, \boldsymbol{x}_j \in \mathbb{R}^d, j = 1, 2, \dots$, which follows the gradient density ascent. The theoretical mean shift recurrence relation is

$$\boldsymbol{x}_{j+1} = \boldsymbol{x}_j + \frac{\mathbf{A}\mathsf{D}f(\boldsymbol{x}_j)}{f(\boldsymbol{x}_j)} \tag{1}$$

for a given positive-definite matrix $\mathbf{A}$, for $j \geq 1$ and $\boldsymbol{x}_0 = \boldsymbol{x}$. The output from Eq. (1) is the sequence $\{\boldsymbol{x}_j\}_{j \geq 0}$ which follows the density gradient ascent $\mathsf{D}f$ to a local mode of the density function f.

To derive the formula for the nearest neighbor mean shift for a random sample $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ drawn from a common density f, we replace the density f and density gradient $\mathsf{D}f$ by their nearest neighbor estimates

$$\begin{aligned} \hat{f}_{\mathrm{NN}}(\boldsymbol{x}; k) &= n^{-1} \delta_{(k)}(\boldsymbol{x})^{-d} \sum_{i=1}^{n} \frac{K((\boldsymbol{x} - \boldsymbol{X}_i)}{\delta_{(k)}(\boldsymbol{x}))} \\ \mathsf{D}\hat{f}_{\mathrm{NN}}(\boldsymbol{x}; k) &= n^{-1} \delta_{(k)}(\boldsymbol{x})^{-d-1} \sum_{i=1}^{n} \frac{\mathsf{D}K((\boldsymbol{x} - \boldsymbol{X}_i)}{\delta_{(k)}(\boldsymbol{x}))} \end{aligned} \tag{2}$$

where K is a kernel function and $\delta_{(k)}(\boldsymbol{x})$ as the k-th nearest neighbor distance to $\boldsymbol{x}$, i.e. $\delta_{(k)}(\boldsymbol{x})$ is the k-th order statistic of the Euclidean distances $\|\boldsymbol{x} - \boldsymbol{X}_1\|, \dots, \|\boldsymbol{x} - \boldsymbol{X}_n\|$. These nearest neighbor estimators were introduced by [13] and elaborated by [5,6] for the mean shift.

These authors established that the beta family kernels are computationally efficient for estimating f and $\mathsf{D}f$ for continuous data. The uniform kernel is the

most widely known member of this beta family, and it is defined as $K(\boldsymbol{x}) = v_0^{-1}\mathbf{1}\{\boldsymbol{x} \in B_d(\mathbf{0}, 1)\}$ where $B_d(\boldsymbol{x}, r)$ is the d-dimensional hyper-ball centered at $\boldsymbol{x}$ with radius r and v_0 is the hyper-volume of the unit d-dimensional hyper-ball $B_d(\mathbf{0}, 1)$. With this family of kernels, and the choice $\mathbf{A} = (d+2)^{-1}\delta_{(k)}(\boldsymbol{x})\mathbf{I}_d$, the nearest neighbor mean shift becomes

$$\boldsymbol{x}_{j+1} = k^{-1} \sum_{\boldsymbol{X}_i \in \mathrm{NN}_k(\boldsymbol{x}_j)} \boldsymbol{X}_i \tag{3}$$

where $\mathrm{NN}_k(\boldsymbol{x})$ is the set of the k nearest neighbors of $\boldsymbol{x}$. For the derivation of Eq. (3), see [4,6]. This nearest neighbor mean shift has a simple interpretation since in the mean shift recurrence relation, the next iterate $\boldsymbol{x}_{j+1}$ is the sample mean of the k nearest neighbors of the current iterate $\boldsymbol{x}_j$. On the other hand, as these iterations calculate the sample mean, the mean shift is not directly applicable to binary data.

3 Nearest Neighbor Median Shift Clustering for Binary Data

A categorical feature, which has a finite (usually small) number of possible values, can be represented by a binary vector, i.e. a vector which is composed solely of zeroes and ones. These categorical features can either ordinal (which have an implicit order) or can be nominal (which no order exists). Table 1 presents the two main types of the coding for a categorical feature into a binary vector, additive and disjunctive, for an example of 3-class categorical feature.

Table 1. Additive and disjunctive coding for a 3-class categorical feature.

Class	Additive coding	Disjunctive coding
1	1 0 0	1 0 0
2	1 1 0	0 1 0
3	1 1 1	0 0 1

The usual Euclidean distance is not adapted to measuring the dissimilarities between binary vectors. A popular alternative is the Hamming distance $\mathcal{H}$ [11]. The Hamming distance between two binary vectors $\boldsymbol{x}_1 = (x_{11}, \ldots, x_{1d})$ and $\boldsymbol{x}_2 = (x_{21}, \ldots, x_{2d})$, $\boldsymbol{x}_j \in \{0,1\}^d, j \in 1, 2$, is defined as:

$$\begin{aligned} \mathcal{H}(\boldsymbol{x}_1, \boldsymbol{x}_2) &= \sum_{j=1}^{d} |x_{1j} - x_{2j}| \\ &= d - (\boldsymbol{x}_1 - \boldsymbol{x}_2)^\top (\boldsymbol{x}_1 - \boldsymbol{x}_2). \end{aligned} \tag{4}$$

Equation (4) measures the number of mismatches between the two vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$: as the inner product $(\boldsymbol{x}_1 - \boldsymbol{x}_2)^\top (\boldsymbol{x}_1 - \boldsymbol{x}_2)$ counts the number of elements

which agree in both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, then $d-(\boldsymbol{x}_1-\boldsymbol{x}_2)^\top(\boldsymbol{x}_1-\boldsymbol{x}_2)$ counts the number of disagreements.

The Hamming distance is the basis from which we define the median center of a set of observations $\mathcal{X}=\{\boldsymbol{X}_1,\ldots,\boldsymbol{X}_n\}$, $\boldsymbol{X}_i\in\{0,1\}^d, i=1,\ldots,n$. Importantly the median center of the set of binary vectors, as a measure of the centrality of the values, remains a binary vector, unlike the mean vector which can take on intermediate values. The median center of $\mathcal{X}$ is a point $\boldsymbol{w}=(w_1,\ldots,w_d)$ which minimizes the inertia of $\mathcal{X}$, i.e.

$$\boldsymbol{w}=\underset{\boldsymbol{x}\in\{0,1\}^d}{\operatorname{argmin}}\ \mathcal{I}(\boldsymbol{x}) \tag{5}$$

where

$$\mathcal{I}(\boldsymbol{x})=\sum_{i=1}^{n}\pi_i\mathcal{H}(\boldsymbol{X}_i,\boldsymbol{x})=\sum_{i=1}^{n}\sum_{j=1}^{d}\pi_i\mathcal{I}(x_j),$$

and π_i are the weights and $\mathcal{I}(x_j)=|X_{ij}-x_j|$.

Each component w_j of $\boldsymbol{w}$ minimizes $\mathcal{I}(x_j)$. In the case where all the weights are set to 1, $\pi_i=1, i=1,\ldots,n$, the w_j can be easily computed since it is the most common value in the observations of the j-th feature. This is denoted as $\text{maj}(\mathcal{X})$, the component-wise majority vote winner among the data points. Hence the median center is the majority vote, $\boldsymbol{w}=\text{maj}(\mathcal{X})$.

If we minimize the cost function in Eq. (5) using the dynamic clusters [3] then this leads to the k-modes clustering. Like the k-means algorithm, the k-modes operates in two steps: (a) an assignment step which assigns each candidate point $\boldsymbol{x}$ to the nearest cluster with respect to the Hamming distance, and (b) an optimization step which computes the median center as the majority vote. These two steps are executed iteratively until the value of $\mathcal{I}(\boldsymbol{x})$ converges.

Now we show how the median center can be utilized to define a new modal clustering for binary data based on the mean shift paradigm. In Sect. 2, the beta family kernels were used in the mean shift for continuous data. The most commonly used smoothing kernel, introduced by [1], for binary data is the Aitchison and Aitken kernel:

$$K_\lambda(\boldsymbol{x})=\lambda^{d-\boldsymbol{x}^\top\boldsymbol{x}}(1-\lambda)^{\boldsymbol{x}^\top\boldsymbol{x}},\ \boldsymbol{x}\in\{0,1\}^d.$$

Observe that the exponent for λ is the Hamming distance of $\boldsymbol{x}$. The tuning parameter $\frac{1}{2}\le\lambda\le 1$ controls the spread of the probability mass around the origin $\mathbf{0}$:

- For $\lambda=1/2$, then $K_{1/2}(\boldsymbol{x})=(1/2)^d$, which assigns a constant probability to all points $\boldsymbol{x}$, regardless of its distance from $\mathbf{0}$.
- For $\lambda=1$, $K_1(\boldsymbol{x})=\mathbf{1}\{\boldsymbol{x}=\mathbf{0}\}$, which assigns all the probability mass to $\mathbf{0}$.
- For intermediate values of λ, we have intermediate assignment of between point and uniform probability mass.

Using K_λ, the corresponding kernel density estimate is

$$\tilde{f}(\boldsymbol{x};\lambda) = n^{-1}\sum_{i=1}^{n}\lambda^{[d-(\boldsymbol{x}-\boldsymbol{X}_i)^\top(\boldsymbol{x}-\boldsymbol{X}_i)]}(1-\lambda)^{[(\boldsymbol{x}-\boldsymbol{X}_i)^\top(\boldsymbol{x}-\boldsymbol{X}_i)]}. \tag{6}$$

Since the gradient of the kernel K_λ is $\mathsf{D}K_\lambda(\boldsymbol{x}) = 2\boldsymbol{x}\log((1-\lambda)/\lambda)K_\lambda(\boldsymbol{x})$, the density gradient estimate is

$$\mathsf{D}\tilde{f}(\boldsymbol{x};\lambda) = 2\log(\lambda/(1-\lambda))n^{-1}\left[\sum_{i=1}^{n}\boldsymbol{X}_i K_\lambda(\boldsymbol{x}-\boldsymbol{X}_i) - \boldsymbol{x}\sum_{i=1}^{n}K_\lambda(\boldsymbol{x}-\boldsymbol{X}_i)\right]. \tag{7}$$

To progress in our development of a nearest neighbor median shift for binary data, we focus on the point mass kernel $K_1(\boldsymbol{x}) = \mathbf{1}\{\boldsymbol{x}=\mathbf{0}\}$. In order that ensure that it is amenable for the median shift, we modify K_1 with two main changes:

1. K_1 is multiplied by the indicator function $\mathbf{1}\{\boldsymbol{x}\in B_d(\mathbf{0},1)\}$
2. the indicator function $\mathbf{1}\{\boldsymbol{x}=\mathbf{0}\}$, which places the point mass at the center $\mathbf{0}$, is replaced an indicator that places it on $\mathrm{maj}(B_d(\mathbf{0},1))$, where $\mathrm{maj}(B_d(\mathbf{0},1))$ is the majority vote winner/median center of the data points $\boldsymbol{X}_1,\ldots,\boldsymbol{X}_n$ inside of $B_d(\mathbf{0},1)$.

This second modification results in an asymmetric kernel as the point mass is no longer always placed in the centre of the unit ball. This modified, asymmetric kernel L is

$$L(\boldsymbol{x}) = \mathbf{1}\{\boldsymbol{x}=\mathrm{maj}(B_d(\mathbf{0},1))\}\mathbf{1}\{\boldsymbol{x}\in B_d(\mathbf{0},1)\}.$$

Since L is not directly differentiable, we define its derivative indirectly via $\mathsf{D}K_1$ and the convention that $\log(\lambda/(1-\lambda)) = 1$ for $\lambda = 1$. As $\mathsf{D}K_\lambda(\boldsymbol{x})\big|_{\lambda=1} = 2\boldsymbol{x}K_1(\boldsymbol{x})$ then analogously we define $\mathsf{D}L(\boldsymbol{x}) = 2\boldsymbol{x}L(\boldsymbol{x})$. To obtain the corresponding estimators, we substitute $L, \mathsf{D}L$ for $K, \mathsf{D}K$ in $\tilde{f}, \mathsf{D}\tilde{f}$ in Eqs. (6)–(7) to obtain $\hat{f}, \mathsf{D}\hat{f}$:

$$\begin{aligned}\hat{f}(\boldsymbol{x};k) &= n^{-1}\delta_{(k)}(\boldsymbol{x})^{-d}\sum_{i=1}^{n}L((\boldsymbol{x}-\boldsymbol{X}_i)/\delta_{(k)}(\boldsymbol{x}))\\ \mathsf{D}\hat{f}(\boldsymbol{x};k) &= 2\delta_{(k)}(\boldsymbol{x})^{-d-1}n^{-1}\left[\sum_{i=1}^{n}\boldsymbol{X}_i L((\boldsymbol{x}-\boldsymbol{X}_i)/\delta_{(k)}(\boldsymbol{x})) - \boldsymbol{x}\sum_{i=1}^{n}L((\boldsymbol{x}-\boldsymbol{X}_i)/\delta_{(k)}(\boldsymbol{x}))\right].\end{aligned} \tag{8}$$

To obtain a nearest neighbor mean shift recurrence relation for binary data, we substitute $\hat{f}, \mathsf{D}\hat{f}$ for $f, \mathsf{D}f$ is Eq. (1). For these estimators, the appropriate choice of $\mathbf{A} = \frac{1}{2}\delta_{(k)}(\boldsymbol{x})\mathbf{I}_d$. Then we have

$$\begin{aligned}\boldsymbol{x}_{j+1} &= \boldsymbol{x}_j + \frac{\delta_{(k)}(\boldsymbol{x})}{2}\frac{\mathsf{D}\hat{f}(\boldsymbol{x}_j;k)}{\hat{f}(\boldsymbol{x}_j;k)}\\ &= \frac{\sum_{i=1}^{n}\boldsymbol{X}_i L((\boldsymbol{x}_j-\boldsymbol{X}_i)/\delta_{(k)}(\boldsymbol{x}_j))}{\sum_{i=1}^{n}L((\boldsymbol{x}_j-\boldsymbol{X}_i)/\delta_{(k)}(\boldsymbol{x}_j))}.\end{aligned}$$

We can simplify this ratio if we observe that the scaled kernel is

$$L((\boldsymbol{x}-\boldsymbol{X}_i)/\delta_{(k)}(\boldsymbol{x})) = \mathbf{1}\{\boldsymbol{X}_i \mathrm{maj}(B_d(\boldsymbol{x},\delta_{(k)}(\boldsymbol{x})))\} \cdot \mathbf{1}\{\boldsymbol{X}_i \in B_d(\boldsymbol{x},\delta_{(k)}(\boldsymbol{x}))\};$$

and that $B_d(\boldsymbol{x},\delta_{(k)}(\boldsymbol{x}))$ comprises the k nearest neighbors of $\boldsymbol{x}$, then $\mathbf{1}\{\boldsymbol{X}_i \in B_d(\boldsymbol{x},\delta_{(k)}(\boldsymbol{x}))\} = \mathbf{1}\{\boldsymbol{X}_i \in \mathrm{NN}_k(\boldsymbol{x})\}$. If m is the number of nearest neighbors of $\boldsymbol{x}_j$ which coincide with the majority vote, then

$$\begin{aligned}\boldsymbol{x}_{j+1} &= \frac{\sum_{\boldsymbol{X}_i \in \mathrm{NN}_k(\boldsymbol{x}_j)} \boldsymbol{X}_i \mathbf{1}\{\boldsymbol{X}_i = \mathrm{maj}(\mathrm{NN}_k(\boldsymbol{x}_j))\}}{\sum_{\boldsymbol{X}_i \in \mathrm{NN}_k(\boldsymbol{x}_j)} \mathbf{1}\{\boldsymbol{X}_i = \mathrm{maj}(\mathrm{NN}_k(\boldsymbol{x}_j))\}} \\ &= \frac{m \cdot \mathrm{maj}(\mathrm{NN}_k(\boldsymbol{x}_j))}{m} \\ &= \mathrm{maj}(\mathrm{NN}_k(\boldsymbol{x}_j)). \end{aligned} \tag{9}$$

Therefore in the median shift recurrence relation in Eq. (9), the next iterate $\boldsymbol{x}_{j+1}$ is the median center of the k nearest neighbors of the current iterate $\boldsymbol{x}_j$. Thus, once the binary gradient ascent has terminated, the converged point can be decoded using Table 1), allowing for its unambiguous symbolic interpretation. The gradient ascent paths towards the local modes produced by Eq. (9) form the basis of Algorithm 1, our nearest neighbor median shift clustering for binary data method (BinNNMS).

The inputs to BinNNMS are the data sample $\boldsymbol{X}_1,\ldots,\boldsymbol{X}_n$ and the candidate points $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m$ which we wish to cluster (these can be the same as $\boldsymbol{X}_1,\ldots,\boldsymbol{X}_n$, but this is not required); and the tuning parameters: the number of nearest neighbors k_1 used in BGA task, the maximum number of iterations $j_{\max}$, and the tolerance under which two cluster centres are considered form a single cluster ε. The output are the cluster labels of the candidate points $\{\mathrm{c}(\boldsymbol{x}_1),\ldots,\mathrm{c}(\boldsymbol{x}_m)\}$.

The aim of the ε-proximity cluster labeling step is to gather all points which are under a threshold ε. In order to apply this method we have to build the Hamming similarity matrix which has a $O(n^2)$ time complexity. We initialize the process by taking first point and cluster with it all point whose distance is less than ε. Thus we apply this iterative exploration process by adding the nearest neighbors. Once the first cluster is generated, we take another point from the reduced similarity matrix and repeat the process, until all points are assigned a cluster label. A notable problem still remains with the choice of main tuning parameter ε: we set it to be the average of distance from each point to their k_2 nearest neighbors.

Algorithm 1. BinNNMS – Nearest neighbor median shift clustering for binary data

Input: $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}, k_1, k_2, j_{\max}$
Output: $\{\mathrm{c}(\boldsymbol{x}_1), \ldots, \mathrm{c}(\boldsymbol{x}_m)\}$
/* **BGA task**: compute binary gradient ascent paths */
for $\ell := 1$ to m **do**
  $j := 0$; $\boldsymbol{x}_{\ell,0} := \boldsymbol{x}_\ell$;
  $\boldsymbol{x}_{\ell,1} := \mathrm{maj}(\mathrm{NN}_{k_1}(\boldsymbol{x}_{\ell,0}))$;
  while $j < j_{\max}$ **do**
    $j := j + 1$;
    $\boldsymbol{x}_{\ell,j+1} := \mathrm{maj}(\mathrm{NN}_{k_1}(\boldsymbol{x}_{\ell,j}))$;
  $\boldsymbol{x}^*_\ell := \boldsymbol{x}_{\ell,j}$;
/* ε**-proximity cluster labeling task**: create clusters by merging near final iterates*/
for $\ell_1, \ell_2 := 1$ to m **do**
  if $\mathcal{H}(\boldsymbol{x}^*_{\ell_1}, \boldsymbol{x}^*_{\ell_2}) \leq \varepsilon(k_2)$ **then** $\mathrm{c}(\boldsymbol{x}^*_{\ell_1}) := \mathrm{c}(\boldsymbol{x}^*_{\ell_2})$;

4 Numerical Experiments

In this section, we present an experimental comparison of the BinNNMS to the k-modes clustering (as outlined in Sect. 3). Table 2 lists the details of the dataset obtained from the UCI Machine learning repository [2].

- The Zoo data set contains $n = 101$ animals described with 16 categorical features: 15 of the variables are binary and one is numeric with 6 possible values. Each animal is labelled 1 to 7 according to its class. Using disjunctive coding for the categorical variable with 6 possible values, the data set consists of a 101×21 binary data matrix.
- The Digits data concerns a dataset consisting of the handwritten numerals ("0"–"9") extracted from a collection of Dutch utility maps. There are 200 samples of each digit so there is a total of $n = 2000$ samples. As each sample is a 15×16 binary pixel image, the dataset consisted of a 2000×240 binary data matrix.
- The Spect dataset describes the cardiac diagnoses from Single Proton Emission Computed Tomography (SPECT) images. Each patient is classified into two categories: normal and abnormal; there are $n = 267$ samples which are described by 22 binary features.
- The Soybean data is about 19 classes, but only the first 15 have been justified as it appears that the last four classes are not well-defined. There are 35 categorical attributes, with both nominal and ordinal features.
- The Car dataset contains examples with the structural information of the vehicle is removed. Each instance is classified into 4 classes. This database is highly unbalanced since the distribution of the classes is $(70.02\%, 22.22\%, 3.99\%, 3.76\%)$.

Table 2. Overview of experimental datasets.

Dataset	size (n)	#features (d)	#classes (M)
Zoo	101	26	7
Digits	2000	240	10
Spect	267	22	2
Soybean	307	97	18
Car	1728	15	4

4.1 Comparison of the k-Modes and the BinNNMS Clustering

To evaluate the clustering quality, we compare the known cluster labels in Table 2 to the estimated cluster labels from BinNNMS and k-modes. For comparability, the k-modes clustering is also based on the binary median center from Eq. (5). Values of the Adjusted Rand Index (ARAND) [9] and the normalized mutual information (NMI) [15] close to one indicate highly matched cluster labels, and values close to zero for the NMI/less than zero for the ARAND) indicate mismatched cluster labels. Our scala codes to reproduce all results are available at https://github.com/Clustering4Ever/Clustering4Ever.

Table 3 reports the results in terms of the NMI and ARAND after 10 runs of the BinNNMS and k-modes. Unlike BinNNMS, the k-modes clustering requires an a priori number of clusters k, then we set k to be whichever value between the target number of classes from Table 2, or to be the number of clusters obtained from the BinNNMS clustering gives the highest clustering accuracy. The BinNNMS, apart from the Car dataset, outperforms the k-modes algorithm on Zoo, Digits, Spect, and Soybean datasets. Upon further investigation for the Car dataset, recall that the distribution of the cluster labels is highly unbalanced which leads the BinNNMS giving a single class (i.e. no clustering). These unbalanced clusters also translate into low values of the NMI and ARAND for the k-modes clustering.

4.2 Comparison of the Quantization Errors for the BinNNMS

An important and widely used measure of resolution, the quantization error, is computed based on Hamming distances between the data points and the cluster prototypes:

$$Error = \frac{1}{n} \sum_{m=1}^{M} \sum_{\boldsymbol{x}_j \in \mathcal{C}_m} \mathcal{H}(\boldsymbol{x}_j, \boldsymbol{w}_m) \tag{10}$$

where $\{\mathcal{C}_1, \ldots, \mathcal{C}_M\}$ is the set of M clusters, $\boldsymbol{x}$ is a point assigned to cluster $\mathcal{C}_m$, and $\boldsymbol{w}_m$ is the prototype.median center of cluster $\mathcal{C}_m$.

The right hand column in Fig. 1 shows the evolution of the quantization errors for the BinNNMS with different values of k_1 with respect to the target cluster prototypes. As the quantization errors decrease this implies that the data points

Table 3. Comparison of clustering quality indices (NMI and ARAND) for k-modes and BinNNMS. The bold value indicates the most accurate clustering for the dataset.

	NMI		
Dataset	k-modes	k	BinNNMS
Digits	0.360 ± 0.011	40	$\mathbf{0.880 \pm 0.000}$
Zoo	0.789 ± 0.023	8	$\mathbf{0.945 \pm 0.000}$
Soybean	0.556 ± 0.000	40	$\mathbf{0.743 \pm 0.000}$
Spect	0.135 ± 0.000	47	$\mathbf{0.145 \pm 0.000}$
Car	$\mathbf{0.039 \pm 0.019}$	**4**	Single class
	ARAND		
Dataset	k-modes	k	BinNNMS
Digits	0.166 ± 0.021	40	$\mathbf{0.876 \pm 0.000}$
Zoo	0.675 ± 0.032	8	$\mathbf{0.904 \pm 0.000}$
Soybean	0.178 ± 0.000	40	$\mathbf{0.331 \pm 0.000}$
Spect	$\mathbf{-0.009 \pm 0.055}$	**2**	-0.019 ± 0.000
Car	0.016 ± 0.039	**4**	Single class

converge toward their cluster prototypes, and that the decreasing intra-cluster distance further facilitates the clustering process. Thus at the end of the training phase, the data points converge towards to their local mode. In comparison with the ARAND scores in Table 3, the magnitude of the decrease in the quantization errors is inversely proportional to the cluster quality indices. That is, the largest decrease for the Digits dataset implies that BinNNMS clustering achieves here the highest ARAND score.

If we run the labeling phase during the BGA phase for a fixed k_1 then we compute the intermediate prototypes $\boldsymbol{w}_m$ of the clusters $\mathcal{C}_m$ during the binary gradient ascent BGA task. Since BinNNMS provides clusters as the basins of attraction to the local median created by the binary gradient ascent paths, the left column of Fig. 1 shows the quantization error with respect to the intermediate median centers/prototypes. In this case we compute at each iteration 7 modes for Zoo dataset, 10 modes for the Digits, 18 modes for Soybean and 2 modes for Spect datasets using ground truth. These quantization errors decrease to an asymptote for all datasets as the iteration number increases.

Visual Comparison of k-modes and BinNNMS on the Digit Dataset:

to obtain a visual representations of binary digit dataset we have the possibility to transform the binary vector into a binary image where each pixel represent one dimension. We present here prototypes of ground truth and clustering results. Figure 2 show the cluster prototypes provided by k-modes and BinNNMS, displayed as 15×16 binary pixel images. For the k-modes image, the cluster prototype for the "4" digit has been incorrectly associated with the "9" cluster. On the other hand, the BinNNMS image correctly identifies all ten digits from "0" to "9".

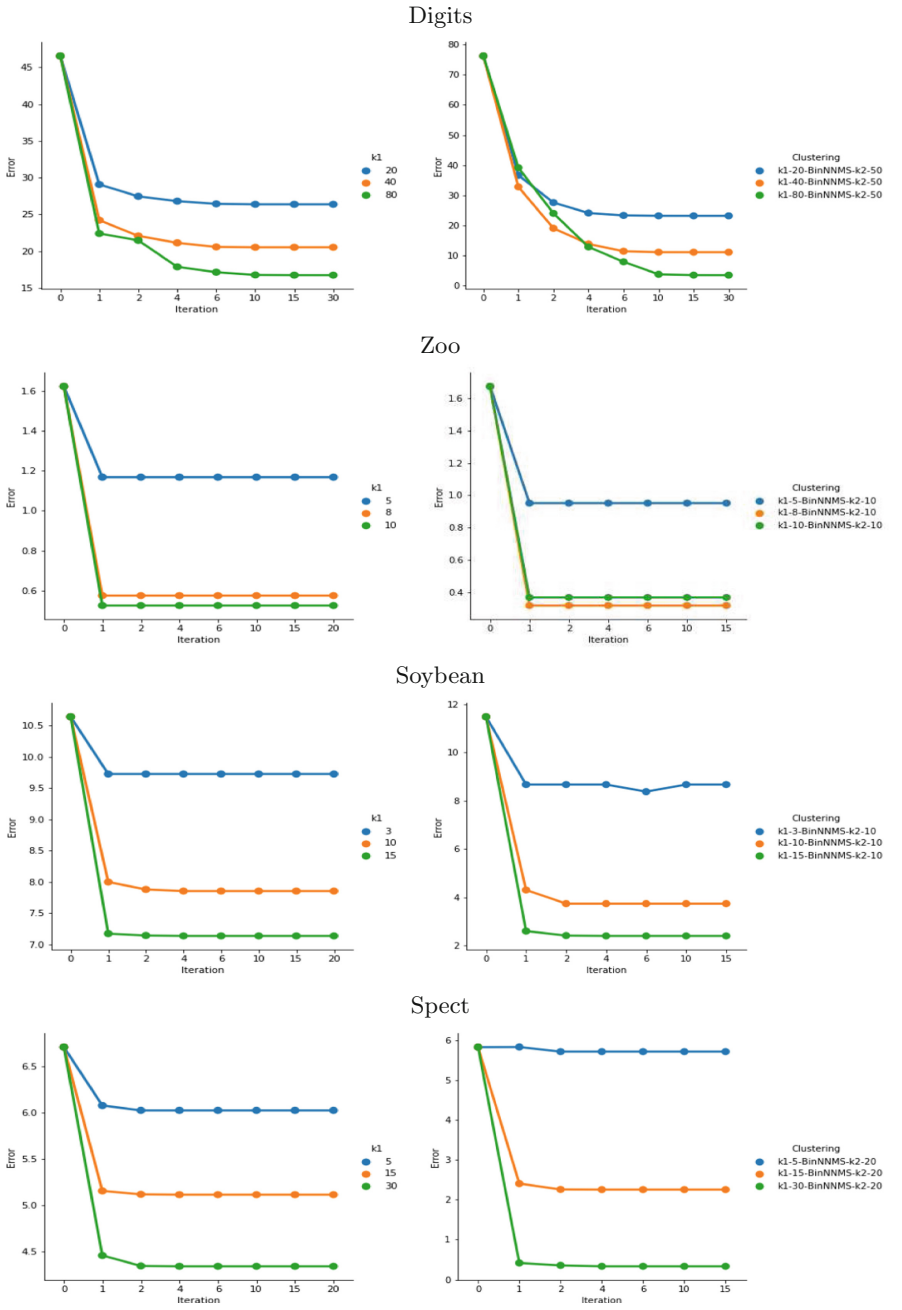

Fig. 1. Evolution of quantization errors as a function of the k_1 and k_2 tuning parameters in BinNNMS for the Digits, Zoo, Soybean and Spect datasets. Left. Quantization errors between the data points and the target prototypes. Right. Quantization errors between the data points and the intermediate median centers in the BGA task and the cluster prototypes.

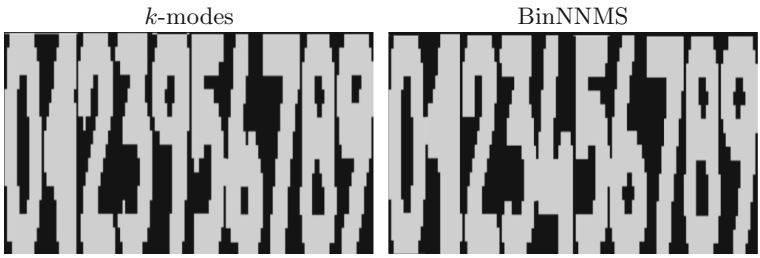

Fig. 2. Comparison of the k-modes and BinNNMS clustered images for the Digits dataset.

5 Conclusion

In this paper, we have proposed a new and efficient modal clustering method for binary data. We introduced a mathematical analysis of the nearest neighbor estimators for binary data. This was then combined with the Aitchison and Aitken kernel in order to generalize the traditional mean shift clustering to the median shift clustering for binary data (BinNNMS). Experimental evaluation for a number of experimental datasets demonstrated that the BinNNMS outperformed the k-modes clustering in terms of visual criteria, as well as quantitative clustering quality criteria such as the adjusted Rand index, the normalized mutual information and the quantization error. In the future we envisage to make our algorithm as automatic as possible by optimizing the choice of the tuning parameters, and to implement a scalable version for Big Data by using approximate nearest neighbor searches.

References

1. Aitchison, J., Aitken, C.G.G.: Multivariate binary discrimination by the kernel method. Biometrika **63**, 413–420 (1976)
2. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
3. Diday, E., Simon, J.C.: Clustering Analysis, pp. 47–94. Springer, Berlin (1976). https://doi.org/10.1007/978-3-642-96303-2_3
4. Duong, T., Beck, G., Azzag, H., Lebbah, M.: Nearest neighbour estimators of density derivatives, with application to mean shift clustering. Pattern Recogn. Lett. **80**, 224–230 (2016). https://doi.org/10.1016/j.patrec.2016.06.021
5. Fukunaga, K., Hostetler, L.: Optimization of k-nearest-neighbor density estimates. IEEE Trans. Inform. Theory **19**, 320–326 (1973)
6. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE T. Inform. Theory **21**, 32–40 (1975). https://doi.org/10.1109/TIT.1975.1055330
7. Hamming, R.W.: Error detecting and error correcting codes. Bell Syst. Tech. J. **29**, 147–160 (1950). https://doi.org/10.1002/j.1538-7305.1950.tb00463.x

8. Huang, Z.: Clustering large data sets with mixed numeric and categorical values. In: The First Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 21–34 (1997)
9. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)
10. Lebbah, M., Badran, F., Thiria, S.: Topological map for binary data. In: ESANN 2000, 8th European Symposium on Artificial Neural Networks, Bruges, Belgium, 26–28 April 2000, Proceedings, pp. 267–272 (2000)
11. Leisch, F., Weingessel, A., Dimitriadou, E.: Competitive learning for binary valued data. In: Niklasson, L., Bodén, M., Ziemke, T. (eds.) ICANN 1998. PNC, pp. 779–784. Springer, London (1998). https://doi.org/10.1007/978-1-4471-1599-1_120
12. Li, T.: A unified view on clustering binary data. Mach. Learn. **62**, 199–215 (2006). https://doi.org/10.1007/s10994-005-5316-9
13. Loftsgaarden, D.O., Quesenberry, C.P.: A nonparametric estimate of a multivariate density function. Ann. Math. Statist. **36**, 1049–1051 (1965). https://doi.org/10.1214/aoms/1177700079
14. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, pp. 281–297. University of California Press, Berkeley, USA (1967). https://projecteuclid.org/euclid.bsmsp/1200512992
15. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)