The Borda Count as a tool for reducing the influence of the distance function on *k*means*

Noelia Rico[1], Raúl Pérez-Fernández[2,3], and Irene Díaz[1]

[1] Department of Computer Science, University of Oviedo, Spain
{noeliarico,sirene}@uniovi.es

[2] Department of Statistics and O.R. and Mathematics Didactics, University of Oviedo, Spain perezfernandez@uniovi.es

[3] KERMIT, Department of Data Analysis and Mathematical Modelling, Ghent University, Belgium raul.perezfernandez@ugent.be

Abstract. The aim of a clustering method is to create groups (clusters) of objects that are similar to each other. This similarity is usually measured by means of a distance, thus, the choice of distance function plays a crucial role in the clustering process. In this work, we propose a variant of the classical clustering method *k*means that combines the information given by different distances to group the objects. More precisely, the cluster to which an object is assigned is chosen by applying the Borda Count to the rankings of closest cluster centers induced by different distances. Experiments were carried out for 81 different datasets. For the vast majority of these datasets, the clusters obtained with the proposed method reduced the Total Distance Within Clusters (TDWC) in comparison with the clusters obtained with the classical *k*means.

Keywords: Clustering · Aggregation · Ranking · Borda Count · *k*means.

1 Introduction

The task of grouping objects of a dataset into smaller subsets of objects based on their similarity is broadly known as **clustering**. Clustering methods can be divided into different families based on their strategy for creating the clusters [8]. One of these families is called **partitional clustering** [9], being ***k*means** [5] the most prominent method of the family. This method is an iterative process that, after an initial cluster allocation, iteratively recalculates the clusters by reassigning the objects to the cluster whose center is the closest. Obviously, the way in which the distance of the objects to the centers is calculated plays a big role in the method of *k*means.

In this work, we present a variant of *k*means that reduces the influence of the distance function by modifying the step in which objects are assigned to a cluster. More precisely, the cluster to which an object is assigned is not calculated

* This research has been partially supported by Spanish MINECO project (TIN2017-87600-P) and FC-GRUPIN-IDI/2018/000176. Raúl Pérez-Fernández acknowledges the support of the Research Foundation of Flanders (FWO17/PDO/160).

based on one sole distance function but, instead, by several distance functions fixed beforehand. We bring to the attention the Borda Count social choice function [1], which has been used in the field of social choice theory for centuries, as a tool that can combine the information given by all these distances. In this way, the assignation step becomes more robust with regard to the choice of distance function. Experiments on several artificially generated datasets support the claim.

The paper is organised as follows: in Section 2, the preliminary concepts that are required for the understanding of this work are introduced (the Borda Count, the definition of some prominent distances and the $k$means method). In Section 3, the proposed method is introduced. The experiments performed to test the method and the results are discussed in Section 4. To sum up, some conclusions and future work are commented in Section 5.

2 Preliminaries

In this section, we recall some preliminary notions that will be essential to understand this work.

2.1 Borda Count

Let $\mathscr{C} = \{c_1, c_2, \ldots, c_k\}$ denote a set of candidates. A ranking on $\mathscr{C}$ is the strict part of a total order relation on $\mathscr{C}$. A list of rankings on $\mathscr{C}$ is called a profile of rankings. A **social choice function** [10] is a function that assigns a non-empty subset of $\mathscr{C}$ to any profile of rankings, being this subset understood as the object(s) ranked overall at the best positions in the profile of rankings.

The **Borda Count** [1] is among the most prominent social choice functions. For applying the Borda Count, each candidate is awarded one point every time that it is ranked at a better position than another candidate and half a point every time that it is tied at the same position as another candidate. All the points obtained by a candidate over all rankings in the profile of rankings are added up to obtain the so-called **score** of the candidate. Finally, the Borda Count outputs the candidate(s) with the highest score.

Example 1. Consider the profile of four rankings on a set of six candidates given in Table 1.

The Borda Count is applied to the profile of rankings in Table 1. Table 2 shows the points that each candidate receives in each of the rankings, and the scores obtained by each candidate after adding these points. For instance, for the ranking r_1, candidate C is ranked at the worst position, so receives no points; candidate E is only ranked at a better position than candidate C, so receives one point; candidate B is only ranked at a better position than candidates E and C, so receives two points; and so on. Notice what happens when two or more candidates are tied. For instance, for the ranking r_4, candidates A, D and F are tied with each other and are ranked at a better position than the other

Table 1. Profile of four rankings on a set of six candidates.

	Ranking
r_1	$F \succ A \succ D \succ B \succ E \succ C$
r_2	$F \succ A \sim C \sim D \succ B \succ E$
r_3	$F \succ C \succ B \succ A \sim E \succ D$
r_4	$A \sim D \sim F \succ C \succ B \sim E$

three candidates. Therefore, four points are assigned to each of A, D and F. This process is applied to all the rankings until each candidate obtains a final score. Ultimately, candidates are ranked according to their scores, resulting in the ranking $F \succ A \succ D \succ C \succ B \succ E$.

Table 2. Points for each candidate in each ranking of the profile of rankings in Table 1 applying the Borda Count.

Ranking	A	B	C	D	E	F
r_1	4.0	2.0	0.0	3.0	1.0	5.0
r_2	3.0	1.0	3.0	3.0	0.0	5.0
r_3	1.5	3.0	4.0	0.0	1.5	5.0
r_4	4.0	0.5	2.0	4.0	0.5	4.0
Total	12.5	6.5	9.0	10.0	3.0	19.0
Final ranking	$F \succ A \succ D \succ C \succ B \succ E$					

2.2 Distances

Ten prominent distances[4] in the context of clustering have been selected based on the comparative study presented in [7]. Formally, these ten distances are defined, for any two objects $\mathbf{v} = (v_1, \ldots, v_n)$ and $\mathbf{w} = (w_1, \ldots, w_n)$, as follows.

Canberra distance:

$$d_{\text{can}}(\mathbf{v}, \mathbf{w}) = \sum_{i=1}^{n} \frac{|v_i - w_i|}{|v_i| + |w_i|} \, .$$

Chebyshev distance:

$$d_{\text{che}}(\mathbf{v}, \mathbf{w}) = \max_i |v_i - w_i| \, .$$

[4] It should be noted that not all the considered distance functions fulfill all axioms normally required for a ‘metric’.

Cosine distance:

$$d_{\cos}(\mathbf{v},\mathbf{w}) = 1 - \frac{\sum_{i=1}^n v_i w_i}{\sqrt{\sum_{i=1}^n v_i^2}\sqrt{\sum_{i=1}^n w_i^2}} \,.$$

Euclidean distance:

$$d_{\text{euc}}(\mathbf{v},\mathbf{w}) = \sqrt{\sum_{i=1}^n (v_i - w_i)^2} \,.$$

Jaccard distance:

$$d_{\text{jac}}(\mathbf{v},\mathbf{w}) = \frac{\sum_{i=1}^n (v_i - w_i)^2}{\sum_{i=1}^n v_i^2 + \sum_{i=1}^n w_i^2 - \sum_{i=1}^n v_i w_i} \,.$$

Manhattan distance:

$$d_{\text{man}}(\mathbf{v},\mathbf{w}) = \sum_{i=1}^n |v_i - w_i| \,.$$

Matusita distance:

$$d_{\text{mat}}(\mathbf{v},\mathbf{w}) = \sqrt{\sum_{i=1}^n (\sqrt{v_i} - \sqrt{w_i})^2} \,.$$

Max symmetric distance:

$$d_{\text{mas}}(\mathbf{v},\mathbf{w}) = \max\left(\sum_{i=1}^n \frac{(v_i - w_i)^2}{v_i}, \sum_{i=1}^n \frac{(v_i - w_i)^2}{w_i}\right) \,.$$

Triangular discrimination distance:

$$d_{trd}(\mathbf{v},\mathbf{w}) = \sum_{i=1}^n \frac{(v_i - w_i)^2}{v_i + w_i} \,.$$

Vicissitude distance:

$$d_{\text{vsd}}(\mathbf{v},\mathbf{w}) = \sum_{i=1}^n \frac{(v_i - w_i)^2}{\max(v_i, w_i)} \,.$$

2.3 *k*means

Given m objects $\mathbf{x}_1,\ldots,\mathbf{x}_m \in \mathbb{R}^n$ (where n is the number of attributes), a clustering method aims at partitioning $\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$ into k groups (referred to as clusters) such that objects in the same cluster are the most similar as possible to each other and objects in different clusters are the most different as possible to each other.

The quality of a partition is measured in many different ways, for instance, in terms of the cohesion of the clusters that represents how similar all objects within the clusters are. This cohesion is typically described in terms of the Total Within Sum of Squares (TWSS) that sums the Within Sum of Squares (WSS_j) of all clusters $\{\mathcal{C}_j\}_{j=1}^k$. Formally, the WSS_j of a cluster $\mathcal{C}_j$ with center $\mathbf{c}_j$ is defined as

$$\text{WSS}_j = \sum_{\mathbf{x}_i \in \mathcal{C}_j} \sum_{\ell=1}^n (x_{i\ell} - c_{j\ell})^2 \,,$$

and the TWSS is defined as

$$\mathrm{TWSS} = \sum_{j=1}^{k} \mathrm{WSS}_j = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \mathcal{C}_j} \sum_{\ell=1}^{n} \left(x_{i\ell} - c_{j\ell}\right)^2 \ , \tag{1}$$

where $x_{i\ell}$ and $c_{k\ell}$ respectively denote the ℓ-th component of $\mathbf{x}_i$ and $\mathbf{c}_j$ with $\ell \in \{1, \ldots, n\}$.

As proposed by MacQueen in the original paper [5] on $k$means, the goal is to identify the partition of $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ into k clusters that minimizes the TWSS. Unfortunately, finding such partition is a very difficult problem in case m is moderately large. Therefore, an heuristic approach, referred to as $k$means, is typically considered. There are many variants of $k$means, being the three discussed in [6] the most prominent ones. We will focus on the one presented by Lloyd [4].

The method works as follows. First, it is necessary to establish the number k of clusters the dataset is going to be divided into. Next, k initial centers must be selected. This task is usually done randomly but different approaches have been presented through the years [2] making this field of research quite popular. In this work, we follow the most classical approach and select the initial centers randomly among the objects in the dataset. The method of $k$means then consists of two main parts, the assignment phase and the update of the center. After the initial centers have been selected, the assignment phase occurs and the objects are assigned to the cluster represented by the closest center. When all the objects have been assigned to a cluster, the center of each cluster is recomputed as the centroid of all the points belonging to that cluster, meaning by centroid the (componentwise) arithmetic mean of all the objects belonging to that cluster. These steps are repeated until all the points are assigned to the same cluster that in the previous iteration or a maximum number of iterations established beforehand is attained. The pseudocode of this procedure is outlined in Algorithm 1. The assignment phase is highlighted to ease the comparison with the method presented later on in Section 3.

Note that it is assured that, in each iteration, a partition with a value of TWSS smaller than that of the previous iteration is obtained. Unless the maximum number of iterations is reached, a local minimum of the TWSS is obtained by $k$means. It is common to run the method several times with different initialisations aiming at finding the closest solution to a global minimum of the TWSS.

Algorithm 1: Lloyd

```
Select $k$ random objects from the dataset as initial centers of the clusters;
for $iter$ to $max\_iter$ do
    foreach $\mathbf{x}_i \in objects$ do
        foreach $\mathbf{c}_j \in centers$ do
            if $d(\mathbf{x}_i, \mathbf{c}_j) \leq d(\mathbf{x}_i, current\_cluster)$ then
                modify $current\_cluster$ to $\mathbf{c}_j$;
            end
        end
    end
    foreach $\mathcal{C} \in clusters$ do
        update center to centroid of objects in $\mathcal{C}$;
    end
    if $none\ of\ the\ points\ change\ of\ cluster$ then
        break;
    end
```

3 The proposed method

The TWSS used for measuring the cohesion of the clusters is inherently linked to the Euclidean distance, as can be seen when rewriting Eq. (1) as

$$\mathrm{TWSS} = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \mathcal{C}_j} d_{\mathrm{euc}} \left(\mathbf{x}_i - \mathbf{c}_j \right)^2 . \tag{2}$$

Unfortunately, the Euclidean distance might not be the distance that best fits the given dataset. The goal should then be to minimize the Total Distance Within Clusters (TDWC), defined as follows

$$\mathrm{TDWC}_d = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \mathcal{C}_j} d \left(\mathbf{x}_i - \mathbf{c}_j \right)^2 , \tag{3}$$

where d does not need to be the Euclidean distance and could possibly be unknown.

In order to achieve this purpose, we focus on modifying the assignment phase of Lloyd's original algorithm [4] (highlighted in Algorithm 1). As mentioned before, in Lloyd's algorithm the object is assigned to the cluster with the closest center, thus, this assignment relies completely on the considered distance. To overcome this problem, we present an approach that combines different distances for identifying the closest centers. Because the values given by different distances typically are not specified in the same order of magnitude, it is not possible to compare the distances between objects directly. Instead, the ordinal information associated with these distances may be used for defining a ranking of closest centers.

In order to apply the method, the number of clusters k and the set $\mathscr{D}$ of distances must be previously defined. The proposed method is applied by following these steps:

1. Select k different random objects as centers;
2. For object $\mathbf{x_i}$, center $\mathbf{c}_j$ and distance d_ℓ, compute $d_\ell(\mathbf{x}_i, \mathbf{c}_j)$;
3. Obtain the ranking $\succsim_{i,\ell}$ of the centers $\mathbf{c}_j$ by increasing value of $\{d_\ell(\mathbf{x}_i, \mathbf{c}_j)\}_{j=1}^{k}$;
4. Repeat Steps 2 and 3 for the n different distances in $\mathscr{D} = \{d_1, \dots, d_n\}$;
5. Apply the Borda Count to the rankings $\{\succsim_{i,\ell}\}_{\ell=1}^{n}$ to obtain the winner $\mathbf{c}$ as the closest center;
6. The object $\mathbf{x}_i$ is assigned to the cluster whose center $\mathbf{c}$ was obtained as the closest in Step 6;
7. Update the center $\mathbf{c}_j$ of each cluster $\mathcal{C}_j$;
8. Repeat Steps 2 to 7 until none of the objects change of cluster or a max number of iterations is reached.

Algorithm 2 presents the pseudocode of this method. The highlighted box corresponds to the assignment phase, which represents the only change with respect to the original Lloyd's algorithm described in Section 2.

Algorithm 2: The proposed method

```
Select $k$ random objects from the dataset as initial centers of the
  clusters;
for $iter$ to $max\_iter$ do
    foreach $\mathbf{x}_i \in objects$ do
        initialize empty profile of rankings $\mathscr{R}_i$;
        foreach $d_\ell \in distances$ do
            foreach $\mathbf{c}_j \in centers$ do
                compute $d_\ell(\mathbf{x}_i, \mathbf{c}_j)$;
            rank centers according to increasing value of $\{d_\ell(\mathbf{x}_i, \mathbf{c}_j)\}$;
            insert ranking into $\mathscr{R}_i$;
        modify current cluster to $borda\_count(\mathscr{R}_i)$;
    foreach $\mathcal{C}_j \in clusters$ do
        update center to centroid of objects in $\mathcal{C}_j$;
    if none of the points change of cluster then
        break;
```

4 Experimental results

4.1 Evolution of the TDWC

It is known that classical $k$means guarantees that the TDWC associated with the Euclidean distance (i.e., the TWSS) decreases in each iteration. Unfortunately, this might no longer (and typically will not) be the case for the TDWC associated with a different distance.

In this section, we will analyse how the proposed method behaves with regard to the TDWC associated with all the distances presented in Section 2.2. In order to compare the proposed method and the classical $k$means, the latter has been performed 10 times, one considering each of the different distances. The TDWC_{d_ℓ} with respect to each distance d_ℓ is calculated for the proposed method and all 10 $k$means (no matter the distance that was used for the method).

Since the distances are measured in different orders of magnitude, it is necessary to exploit the relative information and deal with the rankings induced by the 10 computed TDWC. More specifically, for each distance, we calculate the associated TDWC_{d_ℓ} for the proposed method and all 10 $k$means. The 11 methods are ranked according to increasing TDWC_{d_ℓ} for each distance d_ℓ. Thus, 10 rankings of the methods are obtained. Finally, to obtain an overall ranking of the methods, the Borda Count is applied to these obtained 10 rankings.

Example 2. A little example is outlined given the toy dataset consisting of 15 objects and 2 variables (x and y) presented in Table 3.

Table 3. Points of a normalized dataset used as an example for illustrating the proposed method.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
x	0.05	0.00	0.09	0.87	1.00	0.76	0.68	0.48	0.92	0.47	0.55	0.56	0.24	0.16	0.04
y	0.34	0.41	0.27	0.20	0.09	0.00	0.55	0.44	0.61	0.88	0.86	1.00	0.64	0.61	0.68

The dataset shown in Table 3 is grouped into clusters 11 different times using different techniques: 10 times using $k$means with a different distance d_ℓ (among those presented in Subsection 2.2) and 1 additional time using the proposed method with the ten distances. The value of k has been set to 3 and all the partitions have been initialized with the same random centers.

Fig. 1 shows how the TDWC_{d_ℓ} associated with each d_ℓ evolves for each method. Note that the TDWC_{d_ℓ} for the proposed method is generally decreasing, thus showing that iterating the method leads to a better partition of the dataset into clusters. Also, the proposed method tends to lead to a smaller TDWC_{d_ℓ} than most of the other methods, even though the method with the smallest TDWC_{d_ℓ} tends to be the $k$means corresponding to the distance d_ℓ that is being used for the TDWC_{d_ℓ}. As will be later explained, all these tendencies have also been observed in all the 81 datasets that have been used for the experiments.

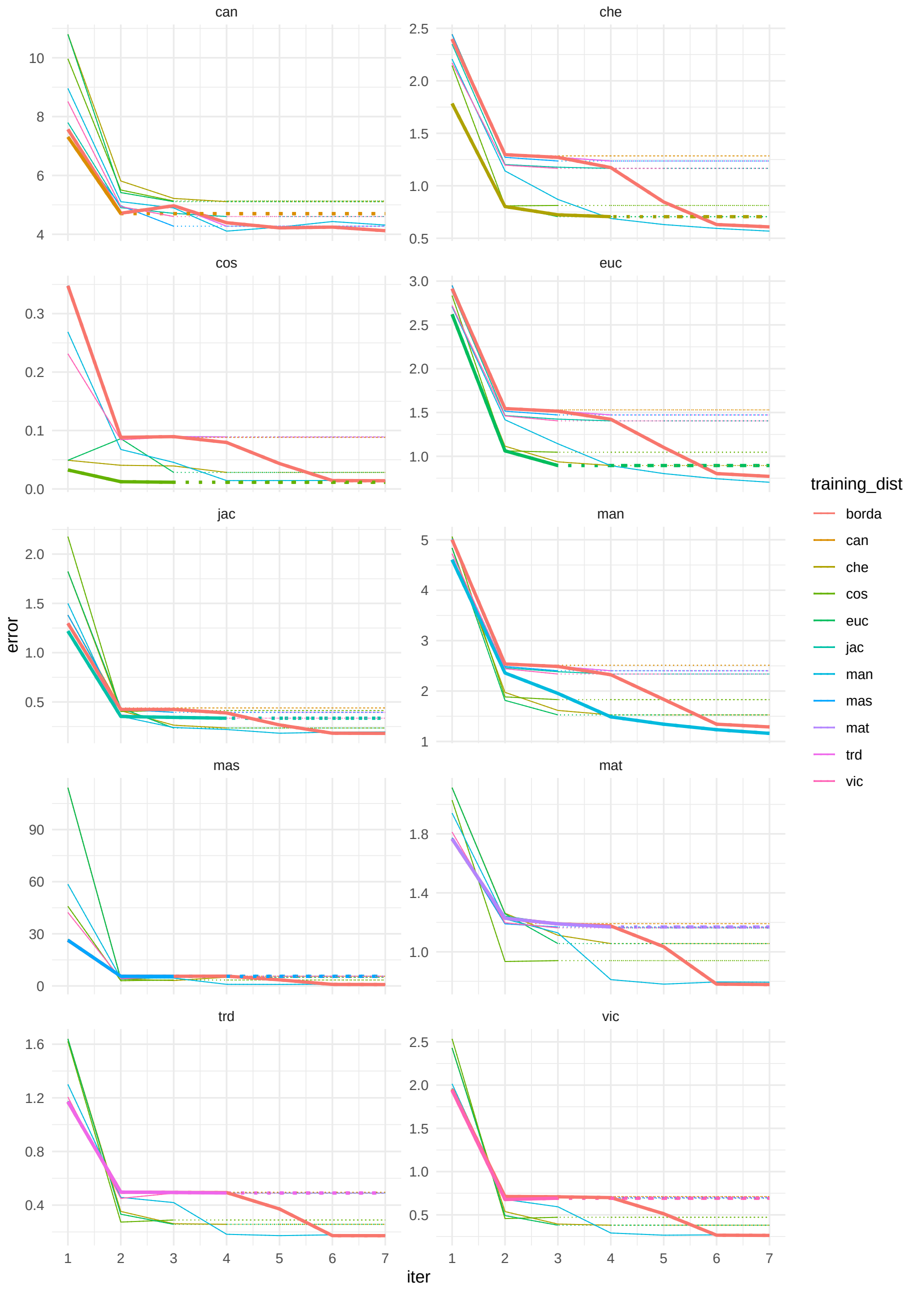

Fig. 1. Evolution of the different TDWC_{d_ℓ} according to each method through the iterations of *k*means and the proposed method. Each box represents the TDWC_{d_ℓ} associated with a certain d_ℓ and each line represents the evolution of TDWC_{d_ℓ} for one of the considered methods.

Table 4. Values of TDWC$_{d_\ell}$ for each distance d_ℓ obtained for the example dataset when applying the proposed method and all ten $k$means.

	can	che	cos	euc	jac	man	mat	trd	vic	mas	borda
TDWC$_{\text{can}}$	4.701	5.109	5.131	5.109	4.602	4.313	4.274	4.274	4.602	4.274	4.121
TDWC$_{\text{che}}$	1.285	0.706	0.813	0.706	1.166	0.568	1.236	1.236	1.166	1.236	0.608
TDWC$_{\text{cos}}$	0.088	0.028	0.011	0.028	0.089	0.014	0.089	0.089	0.089	0.089	0.014
TDWC$_{\text{euc}}$	1.530	0.894	1.046	0.894	1.404	0.704	1.472	1.472	1.404	1.472	0.770
TDWC$_{\text{jac}}$	0.439	0.236	0.413	0.236	0.335	0.196	0.395	0.395	0.335	0.395	0.181
TDWC$_{\text{man}}$	2.511	1.527	1.829	1.527	2.337	1.160	2.403	2.403	2.337	2.403	1.288
TDWC$_{\text{mat}}$	1.191	1.056	0.940	1.056	1.163	0.794	1.168	1.168	1.163	1.168	0.778
TDWC$_{\text{trd}}$	0.495	0.257	0.289	0.257	0.489	0.175	0.490	0.490	0.489	0.490	0.172
TDWC$_{\text{vic}}$	0.712	0.380	0.472	0.380	0.692	0.259	0.699	0.699	0.692	0.699	0.262
TDWC$_{\text{mas}}$	5.575	5.030	3.401	5.030	5.554	0.806	5.523	5.523	5.554	5.523	0.804

The results obtained for each TDWC$_{d_\ell}$ are shown in Table 4.

The proposed method and the 10 $k$means methods are ranked based on increasing TDWC$_{d_\ell}$ for each d_ℓ. This results in 10 different rankings (one for each TDWC$_{d_\ell}$), which are shown in Table 5.

Table 5. Rankings of the methods in terms of all ten TDWC$_{d_\ell}$.

Number of voters	Ranking
1	$borda \succ mat \sim trd \sim mas \succ man \succ jac \sim vic \succ can \succ che \sim euc \succ cos$
4	$man \succ borda \succ che \sim euc \succ cos \succ jac \sim vic \succ mat \sim trd \sim mas \succ can$
1	$cos \succ borda \succ man \succ che \sim euc \succ can \succ mat \sim trd \sim mas \succ jac \sim vic$
1	$borda \succ man \succ che \sim euc \succ jac \sim vic \succ mat \sim trd \sim mas \succ cos \succ can$
1	$borda \succ man \succ cos \succ che \sim euc \succ jac \sim vic \succ mat \sim trd \sim mas \succ can$
1	$borda \succ man \succ che \sim euc \succ cos \succ jac \sim vic \succ mat \sim trd \sim mas \succ can$
1	$borda \succ man \succ cos \succ che \sim euc \succ mat \sim trd \sim mas \succ jac \sim vic \succ can$

All the rankings shown in Table 5 are gathered into a profile of rankings, resulting in the following ranking after computing the scores associated with the Borda Count:

$$borda \succ man \succ che \sim euc \succ cos \succ jac \sim vic \succ mat \sim trd \sim mas \succ can\,.$$

We conclude that, overall, the here proposed method leads to a smaller TDWC than all ten $k$means.

4.2 Experiments and results

In order to measure the behaviour of the proposed method, 81 datasets have been tested. These datasets are built from artificially generated data and each of them combines a different number of gaussian clusters and a different number of variables. Both the number of clusters and the number of variables in the dataset range from 2 to 10. The value of k used in the methods has been fixed to the real number of clusters used in the construction of the datasets to simplify the process. Every dataset has been normalized before applying the clustering method in order to assure that there are no negative values and, therefore, that all distances can be applied. More details on the datasets can be found in the following site https://noeliarico.shinyapps.io/clustering/.

An overall comparison of all methods head-to-head is presented in Table 6. The number in each cell represents the amount of datasets for which the method corresponding to the row was ranked at a better position the method corresponding to the column in the obtained ranking. We can conclude that the proposed method (Borda) performs better in comparison with all other *k*means at least for half of the datasets. It is worth mentioning that the proposed method clearly outperforms the most standard version of *k*means in which the Euclidean distance is considered.

Table 6. Pairwise comparison of the methods taking into account all the 81 datasets. The number at the ith row and jth column represents the number of times that the method corresponding to the ith row is ranked at a better position than the method corresponding to the jth column. The last column presents the percentage of times that the method corresponding to the row performs better than the other methods over the total of 810 possible comparisons (10 methods and 81 datasets).

	borda	can	che	cos	euc	jac	man	mas	mat	trd	vic	Total
borda	0	64	71	64	56	46	65	60	49	43	41	69%
can	12	0	53	41	25	24	34	35	21	15	14	34%
che	10	26	0	26	11	10	26	25	14	9	5	20%
cos	16	39	55	0	25	18	35	41	22	19	15	35%
euc	24	54	68	54	0	37	59	53	33	28	23	53%
jac	32	55	71	61	42	0	54	54	40	35	35	59%
man	16	46	54	45	17	26	0	44	28	18	11	37%
mas	20	45	54	40	28	25	37	0	17	16	19	37%
mat	28	59	67	59	47	39	53	63	0	12	26	56%
trd	32	64	72	62	51	45	62	64	37	0	28	64%
vic	37	67	74	64	54	46	70	61	46	41	0	69%

5 Conclusions and future work

Popular clustering methods such as *k*means presume that the Euclidean distance (or some other predetermined distance) is the one that best fits the current dataset. Some research on how to learn the most suitable distance has been performed. Here, due to the difficulty that learning the most suitable distance presents in clustering, we follow a totally different approach and propose a natural variant of $k$means that jointly exploits different distances at the same time. The presented results hint that the presented method minimizes the TDWC associated with an unknown distance in comparison to the classical method of $k$means.

The absence of a ground truth for the groups of objects in clustering datasets makes the evaluation process more difficult than for other kinds of problems such as classification. Many performance measures of different nature have been proposed for evaluating clustering algorithms [3], even though most of them rely on a preliminary choice of distance. In future work, we will adapt these performance measures similarly as it has been done for the TWSS. Additionally, since the centroid is inherently linked to the Euclidean distance, future research will focus on how to reduce the importance of the choice of distance to the phase in which the centers are updated.

References

1. Borda, J.C.: Mémoire sur les Élections au Scrutin. Histoire de l'Académie Royale des Sciences, Paris (1781)
2. Celebi, M.E., Kingravi, H.A., Vela, P.A.: A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Systems with Applications **40**(1), 200–210 (2013)
3. Gan, G., Ma, C., Wu, J.: Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability). Society for Industrial and Applied Mathematics, USA (2007)
4. Lloyd, S.: Least squares quantization in pcm. IEEE Transactions on Information Theory **28**, 129–137 (1982)
5. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability. vol. 1, pp. 281–297. University of California Press (1967)
6. Morissette, L., Chartier, S.: The k-means clustering technique: General considerations and implementation in mathematica. Tutorials in Quantitative Methods for Psychology **9**, 15–24 (2013)
7. Visalakshi, N.K., Suguna, J.: K-means clustering using max-min distance measure. In: NAFIPS 2009-2009 Annual Meeting of the North American Fuzzy Information Processing Society. pp. 1–6. IEEE (2009)
8. Xu, D., Tian, Y.: A comprehensive survey of clustering algorithms. Annals of Data Science **2**, 165–193 (2015)
9. Xu, R., Wunsch, D.: Partitional Clustering. Wiley-IEEE Press (2009)
10. Young, H.P.: Social choice scoring functions. SIAM Journal on Applied Mathematics **28**(4), 824–838 (1975)