Pattern Recognition 67 (2017) 62–72

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

The Minkowski central partition as a pointer to a suitable distance exponent and consensus partitioning

Renato Cordeiro de Amorim[a,*], Andrei Shestakov[b], Boris Mirkin[b,c], Vladimir Makarenkov[d]

[a] School of Computer Science, University of Hertfordshire, College Lane AL10 9AB, UK
[b] Department of Data Analysis and Machine Intelligence, National Research University Higher School of Economics, Moscow, Russian Federation
[c] Department of Computer Science and Information Systems, Birkbeck University of London, Malet Street, London WC1E 7HX, UK
[d] Département d'Informatique, Université du Québec à Montréal, C.P. 8888 succ. Centre-Ville, Montreal (QC) H3C 3P8 Canada

ARTICLE INFO

Article history:
Received 20 December 2015
Revised 19 October 2016
Accepted 1 February 2017
Available online 1 February 2017

Keywords:
Clustering
Central clustering
Feature weighting
Minkowski metric
Minkowski ensemble

ABSTRACT

The Minkowski weighted K-means (MWK-means) is a recently developed clustering algorithm capable of computing feature weights. The cluster-specific weights in MWK-means follow the intuitive idea that a feature with low variance should have a greater weight than a feature with high variance. The final clustering found by this algorithm depends on the selection of the Minkowski distance exponent. This paper explores the possibility of using the central Minkowski partition in the ensemble of all Minkowski partitions for selecting an optimal value of the Minkowski exponent. The central Minkowski partition appears to be also a good consensus partition. Furthermore, we discovered some striking correlation results between the Minkowski profile, defined as a mapping of the Minkowski exponent values into the average similarity values of the optimal Minkowski partitions, and the Adjusted Rand Index vectors resulting from the comparison of the obtained partitions to the ground truth. Our findings were confirmed by a series of computational experiments involving synthetic Gaussian clusters and real-world data.

1. Introduction

Clustering algorithms aim at revealing the class structure of a dataset. Many of them do it by partitioning a given dataset Y into K clusters, $S = \{S_1, S_2, \ldots, S_K\}$, so that each cluster $S_k \in S$ contains similar entities. Clustering algorithms have been used in many practical applications, including those in the fields of banking, bioinformatics, computer vision, marketing, security, and general data mining [1–3].

The K-means algorithm [1,3,4] is arguably the most popular clustering method nowadays. To test this claim, we used three most popular search engines, i.e., Google, Bing and Yahoo, to assess the numbers of web pages they return with respect to queries of six popular clustering methods or approaches, including K-means [4], Hierarchical clustering [5], Neighbor-joining [6], Spectral clustering [7], Single linkage [8], and Agglomerative clustering [5]. The results reported in Table 1 do show the prevalence of K-means over other clustering techniques. Implementations of K-means can be easily found in various software packages frequently used in data analysis, such as MATLAB [9], R [10], SPSS [11], and SciPy [12]. Given a dataset Y composed of N entities (i.e., objects) y_i, each described over the same V features (i.e., variables), K-means generates a pre-specified number K of disjoint clusters, so that $S_k \cap S_l = \emptyset$ for $k, l = 1, 2, \ldots, K$ and $k \neq l$, covering the entire dataset. The traditional K-means algorithm runs update-centers/update-clusters iterations as described below.

K-means algorithm

1. Assign the values of K entities of Y, selected at random, to the initial centers $c_1, c_2, \ldots, c_K$. Set $S_k \leftarrow \emptyset$.
2. Assign each entity $y_i \in Y$ to the cluster S_k whose center, c_k, is the nearest to y_i. If there are no changes in S, stop and output clusters S and their centers C.
3. Update each center c_k with respect to the vector of component-wise means of its cluster S_k. Go to step 2.

This method is known to alternatingly minimize the following least-squares criterion:

$$W(S, C) = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{v=1}^{V} (y_{iv} - c_{kv})^2, \quad (1)$$

* Corresponding author.
E-mail addresses: r.amorim@herts.ac.uk (R. Cordeiro de Amorim), avshestakov@hse.ru (A. Shestakov), bmirkin@hse.ru (B. Mirkin), makarenkov.vladimir@uqam.ca (V. Makarenkov).

http://dx.doi.org/10.1016/j.patcog.2017.02.001
0031-3203/

Table 1
Counts of relevant web pages returned by the most popular search engines with respect to queries of the named methods obtained on November 15, 2015 at Birkbeck University of London.

Search engine	Google	Bing	Yahoo
K-means	2,070,000	481,000	537,000
Hierarchical clustering	677,000	251,000	268,000
Neighbor-joining	591,000	146,000	148,000
Spectral clustering	202,000	71,500	78,100
Single linkage	140,000	30,900	32,800
Agglomerative clustering	130,000	33,100	33,000

where $c_k \in C$ is the center of cluster $S_k \in S$, with respect to two groups of variables, clusters $S = \{S_1, S_2, \ldots, S_K\}$ and centroids $C = \{c_1, c_2, \ldots, c_K\}$.

Despite its popularity K-means has several important weaknesses, among them:

1. The number of clusters, K, must be known beforehand;
2. This is a local search algorithm that usually gets trapped in a local minimum;
3. The resulting clustering, S, heavily depends on the initial centers;
4. All features equally contribute to the solution, regardless of their individual degree of relevance.

In this paper, we are mostly concerned with the last item of this list. Until recently, the issue of taking into account the extent of relevance of any specific feature was difficult to address because the traditional K-means algorithm and its objective function (Eq. (1)) lack an explicit feature weighting step. This step has been introduced in several works, thus transforming the two-step iterations of K-means into three-step iterations [13–16]. The additional third step assigns weights to features in such a way that feature's weight gets greater when the feature better accords to the current partition. As we have recently shown, the weights are most naturally fit into the Minkowski distance framework: they are associated with feature scale factors in this perspective [16]. Our algorithm, Minkowski weighted K-means (MWK-means) [16], automatically calculates cluster specific weights for each feature and applies the Minkowski distance to ensure these weights can be seen as feature rescaling factors (more details are given in Section 2). However, the quality of cluster recovery of MWK-means is subject to the selection of a suitable Minkowski distance exponent p. This selection depends on the data structure of Y, making it impossible to have a single value of p that provides optimal clustering in all cases. The issue of finding a proper value of p can be addressed in the framework of semi-supervised clustering [16], yet it is of interest to try tackling it in the unsupervised clustering perspective.

Here we propose an approach associated with the structure of the Minkowski partition ensemble, that is, the set of partitions S_p found at different Minkowski exponent values $p \geq 1$. This ensemble resembles partition ensembles used in consensus clustering, a research direction which became popular in the past decade. It involves a representative set of partitions found by various algorithms or various combinations of parameters (partition ensemble) and a rule for finding an "average" partition according to the ensemble (see, for example, [17–19]). The average partition is supposed to be close to the ground truth partition behind the dataset from which partitions in the ensemble are obtained. Yet, there are properties of the Minkowski partition ensemble that distinguish it from the others considered so far:

1. **Completeness.** Usually, the elements of a partition ensemble are obtained as results of different runs of the K-means clustering algorithm at different initializations, sometimes with additional randomization steps [18,20]. In such a process, one is never able to know how well such a random sample reflects the landscape of possible partitions. In this regard, the Minkowski partition set is complete by the virtue of taking into account MWK-partitions at all possible p. One even may speculate on the nature of Minkowski partitions, as they correspond to the full spectrum of Minkowski distances, from the city-block distance that sums all the component-wise differences between entities at $p = 1$ to the Tchebychev distance that takes into account only the maximum of the differences (at p tending to infinity).
2. **Refinement.** Unlike in the conventional approaches, each of MWK-means partitions results from multiple runs of K-Means rather than from a single run. In practice, the optimal S_p partition is the best out of partitions found at a hundred runs of MWK-means. Moreover, one should not forget that the result is found at features weighted according to their relevance to the partition. That means that the Minkowski partition ensemble is a much more refined set of partitions.
3. **Natural diversity.** There is a claim that a partition ensemble to be successful in recovering the ground truth partition should have a significant level of diversity [18]. This claim generated a series of publications which established that the claim is not quite sound, yet the extent of diversity can be put under control [19]. In our view, the extent of diversity of a partition ensemble should not be considered separately from the structure of the dataset under consideration. For example, if a dataset consists of a set of well-separated compact clusters, then any run of K-means, with an appropriate K, will result in the same partition so that the resulting partition set will consist of many copies of the same partition – the minimum diversity, yet perfectly reflecting the structure of the dataset. Therefore, the extent of diversity of an admissible partition ensemble should depend on the cluster structure of the dataset: the more confusing is the structure, the greater the diversity of the partition ensemble. The Minkowski partition ensemble fully accords with the principle.

These properties of the Minkowski partition ensemble lead us to hypothesize that there exists a "central" partition such that it accords most with both the appropriate Minkowski exponent and the ground truth partition. If true, this hypothesis would also mean that the central partition may well serve as a consensus partition without further elaborations. The goal of this paper is to test this hypothesis in different practical situations. We provide computational evidence that our hypothesis is correct for a large variety of datasets, both synthetic and real. Moreover, we find an empirical signal indicating whether the hypothesis is correct for a given dataset. Also, we show that similar constructions for other partition ensembles cannot warrant that their central partitions have anything to do with the ground truth.

To implement our framework computationally, we define the Minkowski partition ensemble by using a discrete series of values of p, from $p = 1$ to $p = 5$ with a step of 0.1, so that the ensemble consists of the selected MWK-means partitions S_p corresponding to $p = 1.0, 1.1, 1.2, \ldots, 5.0$. The upper boundary value, $p = 5$, according to our experience is quite large, so that larger values of p bring no different partitions. As a measure of similarity between partitions we use the popular Adjusted Rand Index (ARI) [21]. This index is usually chosen, over other indices such as Normalized Mutual Information (NMI), by many authors because, first, its intuitive clarity and, second, its propensity for "picking up" right choices in computations, as mentioned for example in [18]. We use ARI to define what is referred to as Minkowski profile further on.

The Minkowski profile is defined as a mapping of the Minkowski exponent values $p = 1.0, 1.1, \ldots, 5.0$ into the average

similarity values of the corresponding MWK-means partition, S_p, to the other Minkowski partitions. Thus defined, the Minkowski profile can be considered as a concept detailing the notion of diversity of a partition ensemble used in [18–20] in two different formulations. The diversity-one with respect to the ensemble is defined as the average value of all the pairwise partition-to-partition dissimilarity values; the dissimilarity being defined as unity minus the average ARI index value [18]. The diversity-two is defined with respect to any "central" partition, S, as the average dissimilarity with S. Thus, the values constituting the Minkowski profile are the diversity-two characteristics of each specific partition S_p taken as S. On the other hand, the average value of the entire Minkowski profile subtracted from 1 is the diversity-one characteristic of the Minkowski partition ensemble.

Our experiments with synthetic datasets entailing Gaussian clusters of simple structure do show that the central Minkowski partition indeed can be used as a statistical tool for finding both an appropriate Minkowski exponent and a meaningful consensus clustering for a given dataset.

The remainder of the paper is organized as follows. Next section describes all the details regarding the MWK-means algorithm as it is implemented and applied in this study. Section 3 introduces the concepts of the Minkowski partition ensemble and Minkowski profile. The following section describes our experimental findings. Our experiments on testing collinearity between the Minkowski profile and the quality of cluster recovery are described there too. Section 4 recalls the concept of consensus partition, defines the central Minkowski partition, tests experimentally how well this partition works as a consensus partition and points out to an optimal value of the Minkowski exponent. The Conclusion section reviews our findings and describes possible extensions of this work.

2. Minkowski weighted K-means

The Minkowski weighted K-means (MWK-means) algorithm involves both the Minkowski distance and cluster-based feature weights [16]. These feature weights follow the intuitive idea that a given feature v may have different degrees of relevance at different clusters $S_k \in S$ ($k = 1, 2, \ldots, K$). The more a feature is dispersed within a cluster, the lower its weight at this cluster is. The Minkowski distance between an entity y_i and a centroid c_k is defined by $d_p(y_i, c_k) = (\sum_{v=1}^{V} |y_{iv} - c_{kv}|^p)^{1/p}$, where p is the Minkowski exponent.

Any distance measure in the framework of the K-means general scheme introduces some bias to the shapes of clusters to be found. Assuming a two-dimensional space for an easier visualization, the Euclidean distance used in (1) makes K-means biased towards circular clusters. At values of p equal to one, two and tending to ∞, the Minkowski distance is referred to as the Manhattan, Euclidean and Tchebychev distances, respectively. For instance, a value of p located between one and two leads to a bias towards a shape between a rhombus and a circle. In general, we can set the shape bias of the Minkowski distance towards any interpolation between a rhombus (at $p = 1$) and a square (at $p \to \infty$). In fact, the Minkowski distance introduces a bias towards a shape similar to that of a Lamé curve (also known as Superellipse), whose precise shape depends on the selected value of p (see Fig. 1). In the MWK-means algorithm, the Minkowski distance depends on the feature scales. Assuming that the objective is to minimize the sum of distances between entities and their respective centroids, as typical for K-means (1), one can introduce a rescaling factor w_{kv} for each feature v at each cluster $S_k \in S$. This rescaling factor within the Minkowski K-means framework can be interpreted as the feature weight, and the weighted Minkowski distance can be defined as

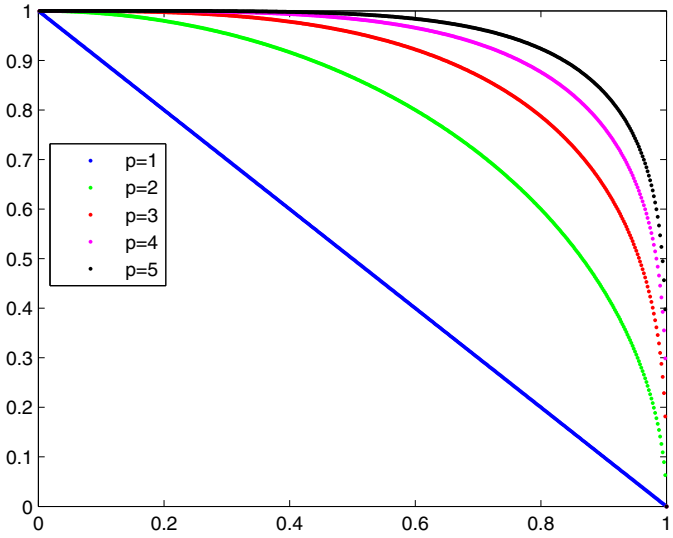

Fig. 1. Fragments of Minkowski plane circles at $p = 1.0, \ldots, 5.0$. The blue line represents the case $p = 1$, green curve - $p = 2$, red curve - $p = 3$, purple curve - $p = 4$, and black curve - $p = 5$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

follows:

$$d_p(y_i, c_k) = \left(\sum_{v=1}^{V} w_{kv}^p |y_{iv} - c_{kv}|^p \right)^{1/p}. \quad (2)$$

Provided that cluster S_k and its center c_k have been pre-specified, the optimal weight w_{kv} of feature v within cluster S_k is inversely proportional to the dispersion D_{kv} of v at S_k. The dispersion D_{kv} is defined by equation $D_{kv} = \sum_{i \in S_k} |y_{iv} - c_{kv}|^p$. Then, the optimal weight w_{kv} is given by:

$$w_{kv} = \left(\sum_{u \in V} [D_{kv}/D_{ku}]^{1/(p-1)} \right)^{-1}. \quad (3)$$

The MWK-means algorithm carries out a series of iterations, each involving three steps specifying how each of the three items, the centroids, the clusters, and the weights, are updated, provided that two of them are given (i.e., optimized at the previous steps).

MWK-means

1. *Parameter setting*. Choose the number of clusters, K, and the Minkowski exponent, p. Set $S \leftarrow \emptyset$, and $w_{kv} = 1/V$ for $k = 1, 2, \ldots, K$ and $v = 1, 2, \ldots, V$.
2. *Setting the centers*. Assign the values of K entities from Y, selected at random, to be the initial cluster centers $c_1, c_2, \ldots, c_K$.
3. *Cluster update*. Assign each entity $y_i \in Y$ to the cluster S_k represented by the nearest c_k as per (2), generating the clustering $S' = \{S'_1, S'_2, \ldots, S'_K\}$. If $S' = S$, then go to Step 6 to end the computation.
4. *Center update*. Update each center $c_k \in C$ to the component-wise Minkowski center of $y_i \in S_k$.
5. *Weight update*. Update each weight w_{kv} using Eq. (3). Set $S \leftarrow S'$, then go to Step 3.
6. *Output*. Output the clustering $S = \{S_1, S_2, \ldots, S_K\}$, centers $C = \{c_1, c_2, \ldots, c_K\}$, and feature weights w.

The central value c_k in Step 4 is given by the component-wise median, mean and mid-range of $y_i \in S_k$, at $p = 1, 2$ and ∞, respectively. At other values of p, subject to $p \geq 1$, $\gamma_v(\mu) = \sum_{i \in S_k} |y_{iv} - \mu|^p$ is a U-shape curve with a minimum located in the interval

$[min_i(y_{iv}), max_i(y_{iv})]$ [16,22]. The center in this case is a minimizer of $\gamma_v(\mu)$. In our previous work [16], a gradient method for finding this minimum has been applied. Here we use a much simpler and faster procedure involving no derivatives. We begin by setting $\mu_{kv} = |S_k|^{-1} \sum_{i \in S_k} y_{iv}$, i.e., the mean value, and then iteratively change it using a pre-specified step size, say 0.001, i.e., adding or subtracting it depending on the side on which the value of γ_v is minimized.

The MWK-means algorithm alternatingly minimizes the following objective function:

$$W_p(S, C, w) = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{v=1}^{V} w_{kv}^p |y_{iv} - c_{kv}|^p, \tag{4}$$

subject to $\sum_{v=1}^{V} w_{kv} = 1$ and $w_{kv} \geq 0$ for $k = 1, 2, \ldots, K$ and $v = 1, 2, \ldots, V$, in the framework of a crisp clustering in which each entity y_i is assigned to a single cluster S_k.

Note that the objective function (4) involves pth power of Minkowski distance rather than the distance itself. This choice is analogous to the use of the squared Euclidean distance, rather than Euclidean distance, in K-means. This objective function also supports cluster-specific feature weights. It shows indeed that the interpretation of weights as the rescaling factors is meaningful because the same exponent p is applied to both the distance and the weights. We have recently shown that using these factors to rescale datasets does improve the likelihood that cluster validity indices return the correct number of clusters [22]. The interpretation of feature weights as feature re-scaling factors is not valid in other feature weighting algorithms such as Weighted K-Means [23], Attribute Weighting K-Means [24], or Improved K-Prototypes [25].

Clearly, the final clustering given by MWK-means depends on the initial centroids chosen in Step 2. When using K-means, this issue is often addressed by running the algorithm a hundred or more times [26] and by selecting the clustering S that provides the best value of the objective function (1). This strategy can still be followed within MWK-means for a given value of p. However, it cannot be used for finding an optimal value of p within MWK-means because the values of the objective function (4) are not comparable at different values of p. However, any cluster validity index that does not depend on p can be used in this case to select the best partitioning.

3. Minkowski partition ensemble, Minkowski profile, and their use

Consider the set of partitions S_p minimizing the objective function (4) at any given value of the exponent $p \geq 1$. It is clear that there can be only a finite number of different partitions S_p because the number of objects is finite. We refer to this set of partitions, $S_M = \{S_p\}$ at various $p \geq 1$, as the set of optimal Minkowski partitions. Of course, finding the set of optimal Minkowski partitions in its entirety is almost unfeasible because the task of minimization of criterion (4) is computationally hard. In practice, there are different options one might wish to explore. Here, we experimented with three of these. For a considered value of p, we carried out MWK-means 100 times, each with a random start. We then took as S_p the partition providing either (i) the minimum value of the objective function W_p (Eq. (4)), or (ii) the maximum value of the Silhouette width (SW) [27], or (iii) the maximum value of the Calinski–Harabasz index (CH) [28].

No sole cluster validity index is clearly superior to all the others in all cases. However, the Silhouette width (SW) and the Calinski–Harabasz index (CH) tend to be among the top performers according to several comprehensive simulation studies [29–31]. There are other potentially valuable alternatives, based for example on the stability-based approach, and we direct interested readers to [29,32,33] and references therein.

The Silhouette width for a clustering is defined as follows:

$$SW = \frac{1}{N} \sum_{i=1}^{N} \frac{b(y_i) - a(y_i)}{max\{a(y_i), b(y_i)\}}, \tag{5}$$

where $a(y_i)$ is the average distance between $y_i \in S_k$ and $\{y_j: y_j \in S_k\}$, and $b(y_i)$ is the lowest average distance between y_i and $\{y_j: y_j \in S_l\}$, where $l \neq k$. The Calinski–Harabasz index is defined as follows:

$$CH = \frac{B}{W} \times \frac{(N-K)}{(K-1)} \tag{6}$$

where W is the overall within-cluster variance, B is the overall between-cluster variance, N is the number of entities, and K is the number of clusters.

We think that there is no need in using values of p outside of interval [1, 5] in our simulations, since the best partitions have never appeared at p greater than 5 in our previous computations [16,22,34]. In fact, the higher the value of p, the more uniform the weights are, thus voiding any advantage provided by the use of feature weights. Therefore, we consider a set $S_M = \{S_p\}$ of 41 Minkowski partitions S_p found at $p = 1.0, 1.1, \ldots, 5.0$, each of them optimising one of the three above-discussed indices (SW, CH, and W_p) over a series of 100 random starts. This set represents an empirical estimate of the set of all optimal Minkowski partitions and constitutes a version of the Minkowski partition ensemble.

Let us now define the concept of Minkowski profile for a given Minkowski partition ensemble. As explained above, we use the Adjusted Rand Index (ARI) [21] to capture the extent of similarity between two partitions. This index is based on the proportion of entity pairs that are consistent between the two partitions, i.e., belong or not to the same cluster in both compared partitions. The ARI index is computed from the confusion table between two cluster partitions, $S_p = \{S_{p1}, S_{p2}, \ldots, S_{pm_p}\}$ and $S_q = \{S_{q1}, S_{q2}, \ldots, S_{qm_q}\}$, where m_p and m_q are the numbers of clusters in S_p and S_q, respectively. The confusion table has rows corresponding to classes of S_p and columns to classes of S_q; its entry (k, l) is the number of objects in the intersection of S_{pk} in S_p and S_{ql} of S_q, $N_{kl} = |S_{pk} \bigcap S_{ql}|$. The confusion table is referred to as the contingency table in statistics. Let N be the total number of entities, N_k - the number of entities in kth cluster of S_p, and N_l - the number of entities in lth cluster of S_q. Then, ARI can be defined as follows:

$$\phi(S_p, S_q) = \frac{\sum_{k,l} \binom{N_{kl}}{2} - C_p C_q \Big/ \binom{N}{2}}{\frac{1}{2}(C_p + C_q) - C_p C_q \Big/ \binom{N}{2}}, \tag{7}$$

where $C_p = \sum_k \binom{N_k^p}{2}$ and $C_q = \sum_l \binom{N_l^q}{2}$. The values of ARI vary between -1 and 1, and ARI $= 1$ if and only if the two compared partitions coincide, i.e., $S_p = S_q$.

For each partition $S_p \in S_M$, we can define a characteristic of its similarity to all the partitions in S_M, i.e., the average similarity:

$$\phi(S_p) = \sum_q \phi(S_p, S_q)/|S_M|. \tag{8}$$

Then, the Minkowski profile $\phi(S_M)$ is defined as a mapping $p \rightarrow \phi(S_p)$ of the set of all considered values of p, into the set of the corresponding average similarity values $\phi(S_p)$, $p = 1.0, 1.1, \ldots, 5.0$.

We can now define the central Minkowski partition as the partition $S_p \in S_M$ corresponding to that p at which the maximum of the Minkowski profile is reached. This means that S_p maximizes the average similarity to S_M over all considered values of p.

Given a partition ensemble, the problem of finding its consensus partition has attracted considerable attention (see, for example, [2,17,18,35] and [36] for the latest references). Most algorithms use the so-called consensus, or co-association, matrix between objects for finding and extending common fragments. There are mathematically deeper approaches using Bayesian or mixture of distributions modeling. In this paper, we do not use any of them, because the concept of Minkowski partition ensemble assumes that there are no meaningful partitions outside of it. Therefore, consensus partition should be one of those constituting the Minkowski partition ensemble. Indeed, we have tried building consensus partitions by using an algorithm from [2,37], which is a version of the approach described in [17]. This usually led to different partitions indeed, but with quite a mediocre cluster recovery results.

Thus, we propose the following routine to select an optimal value of the Minkowski exponent p and determine a Minkowski central partition to be used as a consensus partition:

Choosing an optimal exponent p and central partition S_p

1. *Computing the optimal Minkowski partitions*. For each value of $p = 1.0, 1.1, \ldots, 5.0$, run MWK-means 100 times saving only the run that either (i) maximizes the value of the selected cluster validity index (CH or SW), or (ii) returns the minimum value of W_p. This generates the Minkowski partition ensemble of 41 clusterings.
2. *Computing the Minkowski profile*. Calculate ARI between each pair of Minkowski partitions and define the Minkowski profile as the set of average ARI values between each of the partitions in the Minkowski profile and the rest.
3. *Computing the central Minkowski partition*. Output the central Minkowski partition as a clustering whose average ARI is among the partitions of the Minkowski profile. If there are several partitions of the Minkowski profile that provide the highest value of ARI, select among them the partition that corresponds to the minimum value of the Minkowski exponent p (such a strategy provided the best results in our simulations).
4. *Setting an optimal Minkowski exponent and a consensus partition*. The central Minkowski partition allows one to determine both an optimal exponent p and a consensus partition.

For comparison, we also carry out experiments with the conventional K-means algorithm. There is obviously no need to select a distance exponent in K-means, but one still has to choose here the best partition out of a set of 100 partitions obtained after 100 random starts. To do so, we carry out the above-described routine, but instead of the 41 optimal MWK-means partitions (one for each value of p) we consider the 100 K-means partitions. We compute the ARI between each pair of these 100 K-means partitions, define the profile of the ensemble by computing for each of them the average ARI to the rest, and output the clustering that maximizes the profile. As in [16], in all of our experiments we first consider clustering solutions that have the expected number of clusters. When no such correct clusterings are found by using K-means or MWK-means, we accept those partitions that have been found by these partitioning algorithms regardless of the number of clusters.

We run computational experiments with both real-world and synthetic data. The real-world datasets are those six datasets from the UCI repository that have been used in our previous studies [16], see Table 2.

Among these datasets, there are some with rather clear cluster structure, such as Iris and Wine, as well as some complex datasets for which no conventional classifiers have provided good accuracy results so far, such as Hepatitis and Pima Indians.

We also carry out simulations with synthetic data structures, akin to those presented in our previous works (see for example [16,22]). Our synthetic data are composed of spherical Gaussian clusters so that the covariance matrices are diagonal, with the same diagonal value σ^2 generated randomly at each cluster, and varying between 0.5 and 1.5. All centroid components are generated independently using the standard normal distribution. Cluster cardinalities are generated using a uniform distribution, with a constraint that each generated cluster comprises at least 20 entities. The following GMMs configurations, different in terms of the number of features and clusters, are tested in our study:

- 1000 entities over 8 features constituting 4 clusters (1000x8-4);
- 1000 entities over 10 features constituting 5 clusters (1000x10-5);
- 1000 entities over 12 features constituting 5 clusters (1000x12-5);
- 1000 entities over 20 features constituting 6 clusters (1000x20-6);
- 1000 entities over 30 features constituting 10 clusters (1000x30-10);
- 1000 entities over 40 features constituting 8 clusters (1000x40-8).

Table 2
Real-world datasets from UCI repository used in our experiments.

Dataset	Entities (N)	Features (V)	Clusters (K)
AustraCA	690	14	2
Heart	270	13	2
Hepatitis	155	19	2
Iris	150	4	3
Pima Indians	768	8	2
Wine	178	13	3

It should be noted that not only the feature space dimensions are relatively small at the first three sets of parameters, 8, 10, and 12, but also their relation to the number of clusters is not high either. The space dimension to the number of clusters ratios for these sets are: 8/4=2, 10/5=2, and 12/5=2.4, respectively. This contrasts with the higher ratios at our other parameter combinations: 20/6=3.33, 30/10=3, and 40/8=5. We will see that the cluster recovery results at the latter datasets are much better. For each of these configurations, we generate a hundred different datasets. All results presented further on are the averages taken over the 100 results obtained for each of our configurations.

We standardize each feature by subtracting its mean and dividing it by its range, as shown below:

$$y_{iv} = \frac{y_{iv} - \bar{y_v}}{max(y_v) - min(y_v)}. \quad (9)$$

Often clustering experiments are carried with data standardized using the popular z-score normalization. We think that the above-presented standardization could be a good alternative normalization option [2]. Consider a dataset with two features: a unimodal feature v_1 and a multimodal feature v_2. The standard deviation of v_2 will be higher than that of v_1, leading to lower z-score values of v_2 in comparison to v_1. This means that v_1 would have a higher contribution to clustering in spite of the fact that v_2 has a clearer cluster structure.

Moreover, we carry out additional experiments with the standardized datasets after adding to them noise features. As in our previous studies [16], the values of the noise features are distributed uniformly in the unity range. For all datasets, the number of noisy features inserted is half of the number of the original features.

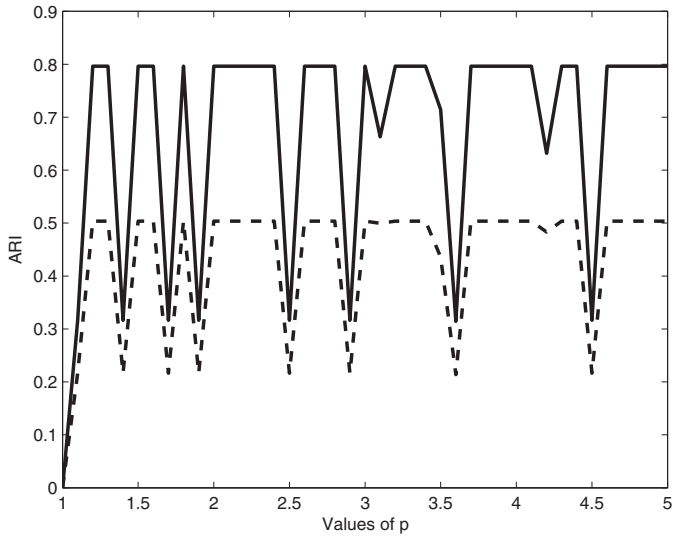

Fig. 2. Adjusted rand index (ARI) of MWK-means applied to the Australian credit approval dataset. The dashed line represents the ARI in relation to the ground truth. The solid line represents the Minkowski profile of this dataset. The optimal Minkowski partition at each value of p was selected using the Silhouette width.

Table 3
Correlations between the Minkowski (or K-means) profiles and the ARI vectors, resulting from the comparison of the obtained partitions to the ground truth, computed for six benchmark datasets from the UCI repository. In the case of K-means, we used a set of 100 partitions obtained from 100 random starts of the algorithm; in the case of MWK-means, we considered a set of 41 optimal partitions (according to the Silhouette width (SW), the Calinski–Harabasz index (CH), and the Minkowski objective function W_p).

	KM	MWK		
		SW	CH	W_p
AustraCA	0.862	0.991	0.991	−0.830
Heart	0.911	0.966	0.899	0.579
Hepatitis	0.886	0.855	0.626	0.928
Iris	0.844	0.984	−0.905	0.961
Pima Indians	0.998	0.949	−0.063	−0.099
Wine	0.738	0.594	0.489	0.957

4. Experimental results

4.1. Relationship between the Minkowski profile and the similarity to ground truth

It appears that the Minkowski profile is closely related to the pre-specified cluster structure of a dataset when the MWK-means partitioning algorithm is used. Specifically, on many real datasets the Minkowski profile closely follows the cluster structure recovered by MWK-means.

For instance, Fig. 2 presents the behaviour of the Minkowski profile and that of the ARI index resulting from the comparison of 41 optimal partitions S_p (at $p = 1.0, 1.1, \ldots, 5.0$, obtained using MWK-means) to the known ground truth partition for the Australian Credit Approval dataset analyzed in many works on data clustering, including [23] and [16]. The striking similarity of the two presented curves is reflected in a very high value of the linear correlation coefficient between them, 0.991.

Table 3 reports the correlation results obtained for the six benchmark datasets from the UCI repository listed above. This table allows us to compare the correlations obtained with traditional K-means and those obtained with our MWK-means algorithm using the Silhouette width (SW) [27], the Calinski–Harabasz (CH) [28]) index, and the Minkowski objective function W_p (Eq. (4)). For each value of p considered in this study, the MWK-means algorithm was carried out 100 times starting at random partitions. Then, the partition maximizing the value of the selected cluster validity index (SW or CH) or minimising the objective function (W_p), at a given value of p, was chosen for calculating the Minkowski profile. The column KM in Table 3 presents the results found by running the conventional K-means algorithm 100 times, also with random initializations (see Section 3). Afterwards, we computed the correlation between the MWK-means (or K-means) profile and the ARI vector resulting from the comparison of the 100 obtained partitions to the ground truth partition. Observing the results presented in Table 3, one can notice that both the traditional K-means and MWK-means used along with the SW cluster validity index provide, in most of the cases, a high correlation between the profile vector and the vector of ARIs resulting from the comparison of the obtained partitions to the ground truth. However, this is not the case of the MWK-means results found using CH and W_p. With the latter partitions, even negative correlation results were obtained for some datasets.

Table 4
Correlations between the Minkowski (or K-means) profiles and the ARI vectors, resulting from the comparison of the obtained partitions to the ground truth, computed for synthetic data. In the case of K-means, we considered a set of 100 partitions obtained from 100 random starts of the algorithm; in the case of MWK-means, we considered a set of 41 optimal partitions according to the SW, CH, and W_p criteria.

		KM	MWK		
			SW	CH	W_p
No noise	1000x8-4	0.315/0.65	0.898/0.19	0.868/0.17	0.938/0.12
	1000x10-5	0.465/0.48	0.938/0.11	0.929/0.10	0.964/0.05
	1000x12-5	0.684/0.35	0.957/0.07	0.953/0.06	0.978/0.02
	1000x20-6	0.799/0.32	0.987/0.03	0.985/0.03	0.986/0.02
	1000x30-10	0.807/0.22	0.994/0.02	0.991/0.02	0.990/0.02
	1000x40-8	0.852/0.23	0.999/0.00	0.997/0.01	0.998/0.00
With noise	1000x8-4	−0.088/0.41	−0.281/0.55	−0.438/0.41	0.613/0.38
	1000x10-5	−0.044/0.40	0.258/0.50	−0.063/0.47	0.827/0.19
	1000x12-5	0.077/0.48	0.733/0.37	0.475/0.46	0.902/0.12
	1000x20-6	0.608/0.32	0.942/0.06	0.930/0.08	0.930/0.12
	1000x30-10	0.616/0.23	0.972/0.06	0.929/0.07	0.984/0.03
	1000x40-8	0.701/0.30	0.958/0.04	0.969/0.03	0.968/0.04

Table 4 reports the average correlation values, obtained for each of the six parameter configurations listed above, between the Minkowski (or K-means) profiles and the ARI vectors resulting from the comparison of the obtained partitions to the ground truth. The obtained standard deviations are also indicated here.

The correlation values presented in Table 4 suggest that the best correlation results have been obtained using MWK-means and the minimum of W_p (Eq. (4)). This trend is particularly noticeable for GMMs with noisy features. One can also observe that the correlations obtained with MWK-means and SW generally follow those obtained with MWK-means and W_p at datasets of larger dimensions. In the GMMs with and without noise, the W_p criterion seems to work better than CH and SW at low-dimensional datasets. Another conclusion which can be drawn from these results is that the second triplet of parameters, 1000x12-5, clearly leads to the increase in the obtained correlations. In general, Table 4 shows quite high correlation values, especially under the SW and W_p scenarios, for both K-means and MWK-means. However, both algorithms fail at small space dimensions under the noise conditions, except for the W_p scenario of MWK-means. At larger space dimensions, the MWK-means results for noisy data show remarkably high correlations under all the three scenarios.

Moreover, we carried out experiments with the Rand, Mirkin, Hubert, and Jaccard, partition similarity indices (for details see

Table 5
Correlations between the Minkowski profiles and the vectors obtained using Jaccard, Hubert, Mirkin and Rand indices, resulting from the comparison of the obtained partitions to the ground truth, computed for synthetic data. The optimal partitions were generated using MWK-means and W_p. We considered 41 optimal partitions (those corresponding to the minimum value of W_p, one for each of the 41 values of p, were selected).

		Jaccard	Hubert/Mirkin/Rand
No noise	1000x8-4	0.920/0.12	0.931/0.13
	1000x10-5	0.959/0.05	0.958/0.06
	1000x12-5	0.976/0.03	0.977/0.03
	1000x20-6	0.981/0.03	0.984/0.03
	1000x30-10	0.985/0.02	0.989/0.02
	1000x40-8	0.996/0.01	0.997/0.01
With noise	1000x8-4	0.529/0.38	0.559/0.40
	1000x10-5	0.793/0.19	0.774/0.29
	1000x12-5	0.890/0.12	0.873/0.23
	1000x20-6	0.915/0.11	0.930/0.14
	1000x30-10	0.981/0.02	0.982/0.03
	1000x40-8	0.949/0.05	0.970/0.04

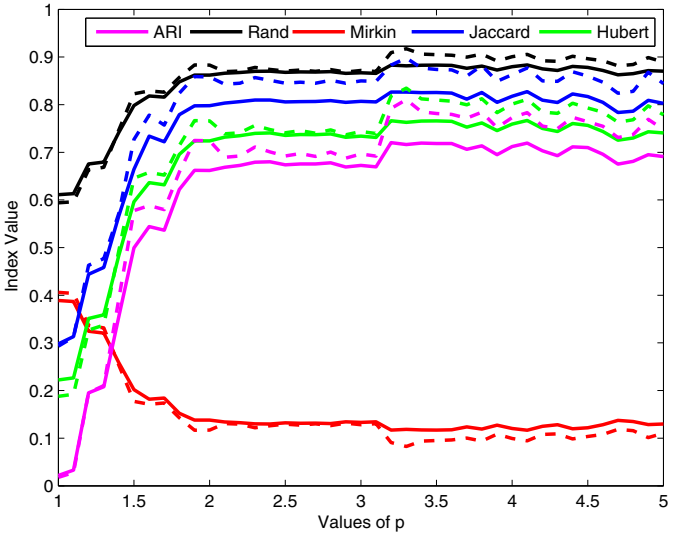

Fig. 3. ARI, Rand, Mirkin's, Jaccard, and Hubert's indices of MWK-means clusterings for a randomly chosen dataset under the configuration 1000x8-4. The dashed lines represent the partition similarity indices in relation to the ground truth. The solid lines represent the Minkowski profiles. The correlation between the values of the two lines is of: 0.9951 (ARI), 0.9946 (Rand), 0.9946 (Mirkin's), 0.9982 (Jaccard), and 0.9946 (Hubert's). Here we used the minimum of the Minkowski objective function, W_p, to select an optimal partition for each value of p.

[21] and references therein), which were used instead of ARI within MWK-means. In these experiments the optimal Minkowski partition at each value of p was selected using W_p. The former three indices are linearly related, which implies that they lead to the same correlation values (see Table 5). The use of the W_p criterion leads to the high correlation values for all of the considered partition similarity indices. Fig. 3 shows the Minkowski profile (solid line) of each index as well as the index value when comparing the partition corresponding to the minimum of W_p to the ground truth (dashed line). This figure presents the results of a randomly chosen dataset under the configuration 1000x8-4.

Overall, these results do show a remarkable affinity between the two series of values associated with elements of the Minkowski partition ensemble chosen under the W_p scenario: (1) the average similarities to the ensemble and (2) the similarity to the ground truth. A similar affinity can be seen at K-means partition ensembles when they are representative of the dataset structure; the effects of noise, however, destroy the balance and K-means partition ensembles fail in this regard under the noise. In contrast, the Minkowski partition ensembles remain representative, especially when the number of clusters is not that high in comparison to the feature space dimension. Therefore, the central Minkowski partition indeed is indicative of both an optimal value of the exponent p and the consensus partition.

4.2. The central Minkowski partition at the UCI repository data

Good affinity between the similarity of a Minkowski MWK-partition to the ground truth and to the Minkowski ensemble is not necessarily an indicator that the central partition is close enough to the ground truth partition. This is illustrated by the results in Table 6 reporting the average ARI values for the six UCI repository datasets with and without noise features. For example, the results obtained using the W_p criterion are rather mediocre here, except those found for the Iris and Wine datasets. The application of our central consensus strategy to traditional K-means when the data were not affected by noise allowed us to generate equal or higher ARI values for five of the six real datasets. Furthermore, in the framework of the MWK-means analysis, the consensus strategy produced competitive or better results in five of the six possible cases when SW was used, and was generally equivalent to the traditional approach when CH was used. When 50% of noise features were added to each dataset, our consensus method using the SW and CH indices generally yielded more stable results than the traditional K-means and MWK-means approaches. The most evident cases of the improvement provided by the consensus MWK-means over the traditional MWK-means include the AustraCC, Hepatitis and PimaIndians datasets when the SW cluster validity index was used. The use of the W_p criterion did not provide any visible improvement in this case.

4.3. Central Minkowski partition at the synthetic data

The tables presented in this section (Tables 7, 8, 9, and 10) are similar to Table 6. They report the ARI values for the generated clusterings in relation to the known ground truth labels. The tables are composed of two main columns. Under “CVI-based”, we provide the ARI values for a given partitioning algorithm (K-means or MWK-means) by simply applying the selected clustering validity index to all of the obtained partitions, and choosing the partition that maximizes the selected CVI. We carried out K-means 100 times per dataset, and MWK-means 100 times for each value of p per dataset. The column “Central” presents the ARI results obtained by applying our central Minkowski partition consensus rule.

Tables 7, 8, and 9, report the results of experiments with MWK-means when using respectively CH, SW, and W_p to choose the optimal Minkowski partition for a given value of p.

The experiments conducted without adding noise features demonstrates that the results generated by the consensus and traditional MWK-means approaches, based on CH and SW, are generally similar (Tables 7 and 8). For instance, with the SW index, the traditional method provides slightly better results in the case of lower numbers of clusters and features, while our central consensus method slightly outperforms the original MWK-means algorithm in the case of higher number of clusters and features. However, when 50% of noise features are added to the synthetic datasets our central consensus strategy, applied in the framework of MWK-means, clearly outperforms the original MWK-means strategy in the case of both CH and SW cluster validity indices. Also, the SW index provides better performances than CH in the context of both original and central consensus clustering strategies. The average optimal value of p usually varies between 2 and 3 in the case of both CH and SW. The results obtained when the minimum of the Minkowski objective function, W_p, was used

Table 6
Results of the experiments with real-world datasets without noise features and with 50% added noise features. The table presents the measures of cluster recovery in terms of Adjusted Rand Index against the known ground truth. The ARI measurements under 'CVI-based' are those for which the resulting clustering was selected based solely on the cluster validity index, where W accounts for the K-means least-squares criterion (Eq. (1)), SW for the Silhouette width, and CH for the Calinski–Harabasz index. The ARI measurements under 'Central' are those obtained using our central Minkowski (or K-means central) consensus rule.

		CVI-based					Central			
		KM			MWK		KM	MWK		
		W	SW	CH	SW	CH		SW	CH	W_p
No noise	AustraCA	0.504	0.499	0.499	0.001	0.504	0.499	0.504	0.504	−0.007
	Heart	0.385	0.423	0.404	0.404	0.404	0.423	0.433	0.376	0.181
	Hepatitis	0.160	0.190	0.141	0.396	0.122	0.268	0.396	0.122	0.355
	Iris	0.716	0.716	0.716	0.716	0.716	0.716	0.745	0.745	0.886
	Pima Indians	0.102	0.011	0.102	0.008	0.096	0.104	0.100	0.100	0.069
	Wine	0.868	0.868	0.868	0.850	0.867	0.915	0.835	0.837	0.787
With noise	AustraCA	0.504	0.499	0.499	0.001	0.504	0.499	0.504	0.504	−0.007
	Heart	0.394	0.423	0.404	0.404	0.376	0.423	0.394	0.367	0.026
	Hepatitis	0.150	0.243	0.122	0.036	0.122	0.293	0.407	0.122	0.417
	Iris	0.529	0.730	0.730	0.445	0.730	0.716	0.445	0.730	0.716
	Pima Indians	0.000	0.011	0.103	0.002	0.104	0.103	0.099	0.100	0.036
	Wine	0.884	0.869	0.847	0.867	0.819	0.882	0.867	0.867	0.788

Table 7
Results of the experiments with MWK-means on synthetic datasets without noise features and with 50% of added noise features. The Calinski–Harabasz (CH) index was used here as CVI. The table presents the measures of cluster recovery in terms of Adjusted Rand Index against the known ground truth and the related average values of the exponent p. The standard deviations of both ARI and p are indicated after a slash.

		CVI-based		Central	
		ARI	p	ARI	p
No Noise	1000x8-4	0.607/0.20	2.306/0.21	0.606/0.20	2.856/0.34
	1000x10-5	0.660/0.18	2.212/0.17	0.664/0.18	2.804/0.39
	1000x12-5	0.776/0.16	2.162/0.16	0.776/0.16	2.904/0.26
	1000x20-6	0.926/0.11	2.050/0.12	0.934/0.08	2.798/0.34
	1000x30-10	0.990/0.01	2.024/0.14	0.986/0.02	2.468/0.47
	1000x40-8	0.995/0.02	2.006/0.06	0.994/0.02	1.838/0.77
With noise	1000x8-4	0.072/0.15	2.524/0.45	0.105/0.17	3.294/0.57
	1000x10-5	0.114/0.15	2.712/0.45	0.183/0.17	2.940/0.71
	1000x12-5	0.288/0.26	2.740/0.52	0.434/0.25	2.392/0.53
	1000x20-6	0.729/0.19	2.226/0.33	0.914/0.12	2.122/0.57
	1000x30-10	0.801/0.10	2.270/0.29	0.903/0.12	2.054/0.54
	1000x40-8	0.993/0.01	1.930/0.18	0.981/0.03	2.222/0.53

Table 8
Results of the experiments with MWK-means on synthetic datasets without noise features and with 50% of added noise features. The Silhouette width (SW) was used here as CVI. The table presents the measures of cluster recovery in terms of Adjusted Rand Index against the known ground truth and the related average values of the exponent p. The standard deviations of both ARI and p are indicated after a slash.

		CVI-based		Central	
		ARI	p	ARI	p
No noise	1000x8-4	0.675/0.19	2.558/0.67	0.665/0.19	2.870/0.39
	1000x10-5	0.712/0.16	2.564/0.65	0.706/0.17	2.936/0.30
	1000x12-5	0.833/0.11	2.404/0.59	0.814/0.14	2.836/0.30
	1000x20-6	0.930/0.07	2.608/0.73	0.933/0.08	2.822/0.30
	1000x30-10	0.974/0.02	2.638/0.75	0.979/0.02	2.538/0.42
	1000x40-8	0.988/0.02	3.382/1.06	0.996/0.01	2.014/0.80
With noise	1000x8-4	0.117/0.20	3.314/0.88	0.152/0.21	3.064/0.48
	1000x10-5	0.246/0.25	3.026/0.72	0.331/0.25	2.688/0.56
	1000x12-5	0.530/0.35	2.528/0.72	0.606/0.29	2.298/0.47
	1000x20-6	0.865/0.14	1.882/0.42	0.893/0.13	2.358/0.36
	1000x30-10	0.939/0.08	2.378/0.53	0.962/0.08	2.356/0.47
	1000x40-8	0.983/0.03	1.870/0.50	0.970/0.04	2.198/0.59

to select optimal partitions show that the W_p criterion clearly outperforms the SW and CH-based central consensus strategies when applied to noisy data, but slightly underperforms when the data do not include noise features (Table 9). Moreover, we conducted similar experiments with the traditional K-means algorithm (Table 10). The results presented in this table suggest that our central consensus rule does not bring any visible advantage in the case of traditional K-means. Here, the classical K-means algorithm is generally more accurate than our consensus strategy, especially when the SW index is used.

The results presented in Tables 7, 8, 9, and 10, as well as the overall simulation graphs in Figs. 4 and 5 suggest that the MWK-means algorithm generally outperforms classical K-means, and it tends to do so with a higher discrimination when the consensus clustering based on our central consensus rule is used. Figs. 4 and 5 summarize the results of our simulations obtained for synthetic data. The presented curves are the averages taken over the correlation (Table 4) and ARI (Tables 7, 8, 9, and 10) values obtained for original and noisy datasets. Fig. 4 shows that the use of the W_p function allows one to obtain very high correlations (Fig. 4a) and good ARI performances (Fig. 4b) even for low-dimensional data. Moreover, very high (i.e., close to 1) correlations between the Minkowski profile and the ARI vectors, resulting from the comparison of the optimal Minkowski partitions to the ground truth, can

Table 9
Results of the experiments with MWK-means on synthetic datasets without noise features and with 50% of added noise features. The minimum of the Minkowski objective function W_p was used here for selecting an optimal partition for each considered value of p. The table presents the measures of cluster recovery in terms of Adjusted Rand Index against the known ground truth and the related average values of the exponent p. The standard deviations of both ARI and p are indicated after a slash. Unlike the previous tables, here we do not report results under 'CVI-based' because the criterion output is not comparable at different values of p. We report solely the results obtained using our central Minkowski consensus rule.

	No noise		With noise	
	ARI	p	ARI	p
1000x8-4	0.604/0.20	3.208/0.43	0.518/0.25	2.650/0.52
1000x10-5	0.635/0.17	3.146/0.45	0.610/0.23	2.398/0.44
1000x12-5	0.743/0.16	3.083/0.35	0.738/0.16	2.462/0.40
1000x20-6	0.882/0.14	2.924/0.43	0.880/0.11	2.548/0.37
1000x30-10	0.944/0.09	2.522/0.49	0.940/0.08	2.416/0.50
1000x40-8	0.970/0.08	2.128/0.74	0.969/0.04	2.258/0.59

be obtained by using the central consensus strategy with any of the three considered optimization criteria (i.e., CH, SW, or W_p) for datasets with large numbers of features ($\geq$ 20 in our case) and clusters ($\geq$ 6 in our case), even in the presence of noise. In terms of ARI (Fig. 5), the proposed central consensus MWK-means algorithm outperforms conventional MWK-means with respect to both

Table 10
Results of the experiments with K-means on synthetic datasets without noise and with 50% of added noise features. The Silhouette width (SW) and the Calinski–Harabasz (CH) index were used here as CVI. The table presents the measures of cluster recovery in terms of Adjusted Rand Index against the known ground truth and the related standard deviations. The results reported under 'CVI-based' are those for which the resulting clustering was selected based solely on CVI. The results reported under 'Central' are those obtained using our K-means central consensus rule.

		CVI-based		Central
		SW	CH	
No noise	1000x8-4	0.649/0.19	0.596/0.20	0.583/0.20
	1000x10-5	0.685/0.18	0.650/0.19	0.619/0.19
	1000x12-5	0.817/0.13	0.769/0.16	0.768/0.16
	1000x20-6	0.933/0.09	0.913/0.12	0.889/0.15
	1000x30-10	0.964/0.07	0.956/0.08	0.932/0.11
	1000x40-8	0.992/0.01	0.980/0.07	0.942/0.12
With noise	1000x8-4	0.053/0.13	0.053/0.12	0.058/0.13
	1000x10-5	0.078/0.11	0.069/0.10	0.064/0.10
	1000x12-5	0.198/0.25	0.152/0.19	0.155/0.20
	1000x20-6	0.445/0.25	0.424/0.22	0.413/0.22
	1000x30-10	0.810/0.12	0.730/0.11	0.746/0.11
	1000x40-8	0.859/0.18	0.789/0.19	0.802/0.22

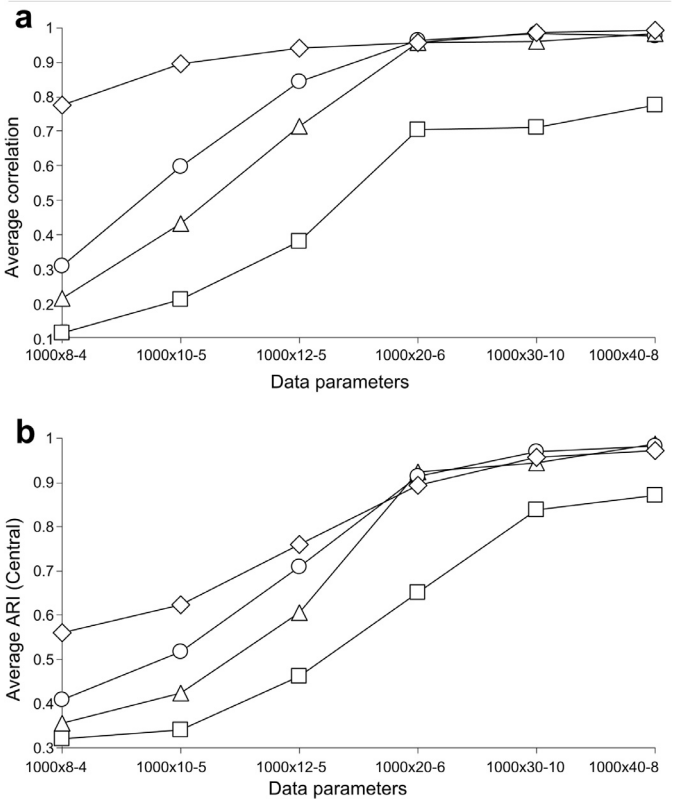

Fig. 4. Average correlation (a) and ARI (b) results obtained by the K-means and MWK-means algorithms for our synthetic data composed of spherical Gaussian clusters. The averages were taken over the results generated for both original and noisy datasets. Our central consensus strategy is represented by open circles (SW-based MWK-means consensus strategy), open triangles (CH-based MWK-means consensus strategy), open rhombuses (W_p-based MWK-means consensus strategy), and open squares (K-means central consensus strategy).

cluster validity indices (CH and SW) used in this study. However, it is not the case of traditional K-means.

5. Conclusion

In this paper, we presented a new way of generating a partition ensemble by employing the framework of Minkowski weighted

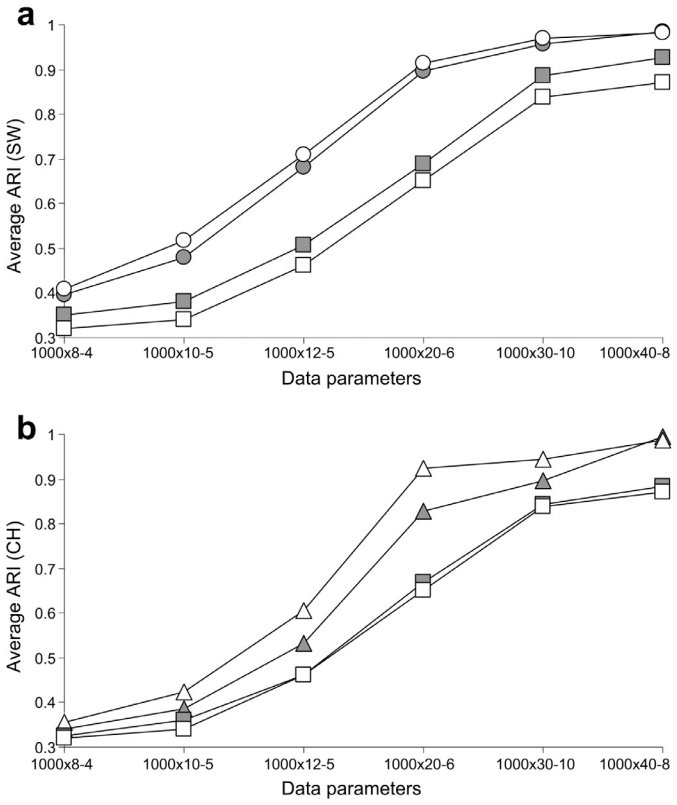

Fig. 5. Average ARI results obtained using SW (a) and CH (b) by the K-means and MWK-means algorithms for our synthetic data composed of spherical Gaussian clusters. The averages were taken over the results generated for both original and noisy datasets. Our central consensus strategy is represented by open circles (SW-based MWK-means consensus strategy), open triangles (CH-based MWK-means consensus strategy), and open squares (K-means central consensus strategy). The CVI-based strategies of MWK-means and K-means are represented by gray circles (SW-based MWK-means strategy), gray triangles (CH-based MWK-means strategy), and gray squares (traditional K-means).

K-Means clustering. In contrast to conventional approaches, the Minkowski partition ensemble satisfies the properties of Completeness, Refinement and Natural diversity discussed in the Introduction section. This allows us to shift the focus from diversity to representativeness: a good partition ensemble should follow the data structure rather than just being simply diverse. The concepts of the Minkowski profile and the central Minkowski partition are introduced to point to a suitable value of the Minkowski exponent p and to a good consensus partition.

In our simulations (see Table 4), we were able to obtain strikingly high correlations between the Minkowski profile and the ARI vector resulting from the comparison of the obtained partitions to the ground truth. For instance, the average correlation for the 100 datasets under the 1000x40-8 configuration was 0.998, when using the W_p criterion (Eq. (4)) to select the optimal Minkowski partition for a given value of p. When adding noise features to the same datasets the correlation was still high, with a value of 0.968. This means that the Minkowski profile can be used for predicting the resemblance of the p-specific partitions to the ground truth and, thus for selecting the optimal value of the Minkowski exponent, p, in the framework of the MWK-means analysis. The resulting central Minkowski partition is defined through a central consensus rule. Furthermore, we showed that the high correlation property also holds for the conventional K-means algorithm, although to a lesser extent, i.e., only for large ratios of the space dimension over the number of clusters.

The results of our simulations, conducted with the Silhouette and Calinski–Harabasz cluster validity indices as well as the Minkowski objective function W_p, original and consensus MWK-means and K-means algorithms, and datasets of different sizes with and without noise features, suggest the central Minkowski partition can potentially provide a good guidance regarding the recovery of an optimal Minkowski exponent and the ground truth clusters, especially in the case when noise features are present in the data, which is typical for most of the real-world data.

Kuncheva and Vetrov [35] looked at the relationship between stability and accuracy with respect to the number of clusters, when investigating whether stability can be used as a CVI. These latter authors proposed a combined stability index, based on the ARI computation, and defined as the sum of the pairwise individual and ensemble stabilities. This index was shown to correlate well enough with the ensemble accuracy [35]. It would be interesting to see in the future whether our Minkowski profile and central Minkowski partition can be used for the same purposes. Thus, the maximum of the Minkowski profile computed over a given interval of values of p and a given interval of numbers of clusters, K, could be viewed as both the ensemble validity estimate and the indicator of the true number of clusters. On the other hand, the middle of the longest constant interval of values of p (i.e., most stable interval; see for example the interval [3.7,4.2] in Fig. 2) could be also used to determine the number of clusters in a dataset.

Of course we feel that the empirical regularity discovered in this paper should be converted into a theoretical one by introducing an adequate mathematical model to both explain the phenomenon and to determine conditions at which it holds.

Acknowledgements

The authors are indebted to the reviewers whose comments helped us to improve both the results and their description. This work was funded by Natural Sciences and Engineering Research Council of Canada (grant no. 2016–06557) and Le Fonds Quebecois de la Recherche sur la Nature et les Technologies (grant no. 173539). Boris Mirkin thanks the Academic Fund Program at the National Research University Higher School of Economics Moscow (grant no. 16-01-0085 supported within the framework of a subsidy granted to the HSE by the Government of the Russian Federation for the implementation of the Global Competitiveness Program in 2016–2017).

References

[1] A. Jain, Data clustering: 50 years beyond k-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[2] B. Mirkin, Clustering: A Data recovery approach, Computer Science and Data Analysis, CRC Press, London, UK, 2012.
[3] D. Steinley, K-Means clustering: a half-century synthesis, Brit. J. Math. Stat. Psychol. 59 (1) (2006) 1–34.
[4] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, California, USA, 1967, pp. 281–297.
[5] J.A. Hartigan, Clustering Algorithms, 99th, John Wiley & Sons, Inc., New York, NY, USA, 1975.
[6] N. Saitou, M. Nei, The neighbor-joining method: a new method for reconstructing phylogenetic trees., Mol. Biol. Evol. 4 (4) (1987) 406–425.
[7] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Advances in Neural Information Processing Systems, MIT Press, 2001, pp. 849–856.
[8] P. Legendre, L.F. Legendre, Numerical Ecology, 3rd, Elsevier, Amsterdam, Netherlands, 2012.
[9] MATLAB, Version 7.10.0 (R2010a), The MathWorks Inc., Natick, Massachusetts, 2010.
[10] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2013.
[11] A. Field, Discovering statistics using SPSS, SAGE Publications, 2005.
[12] E. Jones, T. Oliphant, P. Peterson, et al., SciPy: Open source scientific tools for Python, 2001.
[13] V. Makarenkov, P. Legendre, Optimal variable weighting for ultrametric and additive trees and k-means partitioning, J. Classif. 18 (2) (2001) 245–271.
[14] R.C. de Amorim, A survey on feature weighting based k-means algorithms, J. Classif. 33 (2016).
[15] J.Z. Huang, J. Xu, M. Ng, Y. Ye, Weighting Method for Feature Selection in K-means, in: H. Liu, H. Motoda (Eds.), Computational Methods of Feature Selection, Data Mining & Knowledge Discovery, Chapman & Hall/CRC, 2008, pp. 193–209.
[16] R.C. de Amorim, B. Mirkin, Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering, Pattern Recognit. 45 (3) (2012) 1061–1075.
[17] A. Topchy, A.K. Jain, W. Punch, Combining multiple weak clusterings., in: Third IEEE International Conference on Data Mining, IEEE, 2003, pp. 331–338.
[18] S.T. Hadjitodorov, L.I. Kuncheva, L.P. Todorova, Moderate diversity for better cluster ensembles, Inf. Fusion 7 (3) (2006) 264–275.
[19] M. Pividori, G. Stegmayer, D.H. Milone, Diversity control for improving the analysis of consensus clustering., Inf. Sci. (Ny) 361 (2016) 120–134.
[20] F. Yang, X. Li, Q. Li, T. Li, Exploring the diversity in cluster ensemble generation: random sampling and random projection., Expert Syst. Appl. 41 (10) (2014) 4844–4866.
[21] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (2) (1985) 193–218.
[22] R.C. de Amorim, C. Hennig, Recovering the number of clusters in data sets with noise features using feature rescaling factors, Inf. Sci. (Ny) 324 (2015) 126–145, doi:10.1016/j.ins.2015.06.039.
[23] J.Z. Huang, M.K. Ng, H. Rong, Z. Li, Automated variable weighting in k-means type clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 657–668.
[24] E.Y. Chan, W.K. Ching, M.K. Ng, J.Z. Huang, An optimization algorithm for clustering using weighted dissimilarity measures, Pattern Recognit. 37 (5) (2004) 943–952.
[25] J. Ji, T. Bai, C. Zhou, C. Ma, Z. Wang, An improved k-prototypes clustering algorithm for mixed numeric and categorical data, Neurocomputing 120 (2013) 590–596.
[26] D. Steinley, Profiling local optima in k-means clustering: developing a diagnostic technique., Psychol. Methods 11 (2) (2006) 178.
[27] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.
[28] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, Commun. Stat. Theory Methods 3 (1) (1974) 1–27.
[29] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J.M. Pérez, I. Perona, An extensive comparative study of cluster validity indices, Pattern Recognit. 46 (1) (2012) 243–256.
[30] K.S. Pollard, M.J. Van Der Laan, A method to identify significant clusters in gene expression data, in: Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics, bepress, Orlando, USA, 2002, pp. 318–325.
[31] G.W. Milligan, M.C. Cooper, An examination of procedures for determining the number of clusters in a data set, Psychometrika 50 (2) (1985) 159–179.
[32] U. Von Luxburg, Clustering stability, Found. Trends Mach. Learn. 2 (3) (2010) 235–274.
[33] A. Bertoni, G. Valentini, Discovering multi–level structures in bio-molecular data through the bernstein inequality, BMC Bioinformatics 9 (2) (2008) 1.
[34] R.C. de Amorim, P. Komisarczuk, On initializations for the minkowski weighted k-means, in: Advances in Intelligent Data Analysis XI, in: Lecture Notes in Computer Science, vol. 7619, Springer, 2012, pp. 45–55.
[35] L.I. Kuncheva, D.P. Vetrov, Evaluation of stability of k-means cluster ensembles with respect to random initialization, IEEE Trans. Pattern Anal. Mach. Intell. 28 (11) (2006) 1798–1808.
[36] D. Huang, J. Lai, C.D. Wang, Ensemble clustering using factor graph., Pattern Recognit. 50 (1) (2016) 131–142.
[37] B. Mirkin, A. Shestakov, A note on the effectiveness of the least squares consensus clustering, in: Clusters, Orders, and Trees: Methods and Applications, in: Springer Optimization and Its Applications, vol. 92, Springer, 2014, pp. 181–185.

Renato Cordeiro de Amorim holds a PhD in Computer Science from Birkbeck University of London (2011), and he is currently a Senior Lecturer in Computer Science at the University of Hertfordshire. He has published various papers related to feature weighting as well as unsupervised and semi-supervised learning, with applications in fields such as security, biosignal processing and data mining.

Andrey Shestakov is a PhD student in Computer Science at the Higher School of Economics of Moscow. He works under the supervision of Prof. Boris Mirkin.

Boris Mirkin BSc, MSc and PhD in Computer Science (Saratov State University, Russia), ScD in Engineering (Russian Academy of Sciences, Moscow, 1990). He is an Emeritus Professor of Computer Science at the Department of Computer Science, Birkbeck University of London, and a Full Professor at the School of Data Analysis and Artificial Intelligence, National Research University Higher School of Economics. Research interests: mathematical models and computational algorithms for visualization and clustering of data in molecular biology, genomics, sociology, ecology and other applications.

Vladimir Makarenkov is a Full Professor and Director of a graduate Bioinformatics program at the Department of Computer Science at the UniversitĘ du QuĘbec MontrĘal. His research interests are in the fields of Bioinformatics, Operations Research, Software Engineering, and Mathematical Classification.