



Simplicial and Minimal-Variance Distances in Multivariate Data Analysis

Jonathan Gillard¹ · Emily O’Riordan¹ · Anatoly Zhigljavsky¹

Accepted: 24 December 2021 / Published online: 21 January 2022

© The Author(s) 2022

Abstract

In this paper, we study the behaviour of the so-called k -simplicial distances and k -minimal-variance distances between a point and a sample. The family of k -simplicial distances includes the Euclidean distance, the Mahalanobis distance, Oja’s simplex distance and many others. We give recommendations about the choice of parameters used to calculate the distances, including the size of the sub-sample of simplices used to improve computation time, if needed. We introduce a new family of distances which we call k -minimal-variance distances. Each of these distances is constructed using polynomials in the sample covariance matrix, with the aim of providing an alternative to the inverse covariance matrix, that is applicable when data is degenerate. We explore some applications of the considered distances, including outlier detection and clustering, and compare how the behaviour of the distances is affected for different parameter choices.

Keywords Mahalanobis distance · Scatter · Generalized inverse · Simplices

1 Introduction

The Mahalanobis distance is one of the most useful tools in multivariate data science, underpinning a huge variety of practical data analysis methods. This distance measures the proximity of a point $x \in \mathbb{R}^d$ to a d -dimensional set of points $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^d \times N$. It was introduced in Mahalanobis [27]. The Mahalanobis distance corresponds to the Euclidean distance in the standardized space where variables are uncorrelated

✉ Anatoly Zhigljavsky
zhigljavskyaa@cardiff.ac.uk

Jonathan Gillard
gillardjw@cardiff.ac.uk

Emily O’Riordan
oriordane1@cardiff.ac.uk

¹ School of Mathematics, Cardiff University, Cardiff, UK

and have unit variance. It has applications in cluster analysis [16,44], outlier detection [35], financial settings [40], text classification [39] and image processing [32,45], to name a few.

The Mahalanobis distance requires the inversion of the $d \times d$ sample covariance matrix, which we denote by W . This covariance matrix is commonly singular (or very close to being singular) in high-dimensional settings [13], which often causes instability when computing the Mahalanobis distance and sometimes makes it practically unusable.

The Moore–Penrose pseudo-inverse W^- is commonly used in cases where the covariance matrix is not invertible, see Wei et al. [41] and Lahav et al. [22], for example. This pseudo-inverse is constructed using the nonzero eigenvalues and corresponding eigenvectors of the covariance matrix W , and satisfies the four Moore–Penrose conditions [17]. However, by using *all* nonzero eigenvalues, it can be adversely affected by very small eigenvalues, creating discrepancies between W^- and the true inverse, if it exists [6,19]. The Moore–Penrose pseudo-inverse is unique, and so has no flexibility in these cases where eigenvalues close to zero can cause issues.

Knowing how to deal with high-dimensional data is extremely important, as our ability to collect and store large quantities of data continues to grow. Although there are benefits to having datasets with large amounts of information, analysing and understanding these datasets can present many challenges. For example, the concentration of points in high dimensions can cause distances between them to be (approximately) the same [5], yielding difficulties in determining whether points are ‘close’ or ‘far away’ to each other [1]. Given a high enough dimension, there will exist at least one subspace such that a significant number of points can be classified as being ‘close’ together [46]. Huge datasets create combinatorially intractable search spaces and extracting information or patterns may not be computationally possible. Such datasets also increase the probability of detecting spurious relations [38]. Existing rudimentary data processing algorithms do not scale to scenarios of degenerate data in big dimensions [2].

High-dimensional data is usually highly correlated [9,33]. Measures of proximity that typically work well in low-dimensional settings, such as the Euclidean and Manhattan distance, do not take correlations into consideration when measuring the scatter of the data. Using a distance measure like the Mahalanobis distance accounts for these correlations and therefore produces more accurate and meaningful results in such settings. However, the correlated variables also result in an intrinsic dimensionality that is lower than the given dimensionality of the data [46], causing degeneracy. As such, it is important to have methods that can account for correlations, that can effectively deal with the possibility of singular covariance matrices, and that are not limited by the problems of the Moore–Penrose pseudo-inverse.

A common solution to some of the problems outlined is to use an estimator which produces an invertible sample covariance matrix, rather than using the sample covariance matrix produced by the maximum likelihood estimator (MLE). Examples include methods which shrink the MLE sample covariance matrix towards some other target matrix: Ledoit and Wolf [25] use the identity matrix as the target, whereas Schäfer and Strimmer [37] allow for a user-defined target matrix. However, it is hard to know what the ideal target matrix is [24]. Other methods include shrinking only the largest and smallest eigenvalues [43], tapering off-diagonal elements to zero [3,15], applying ℓ_1

penalties directly to covariance matrix entries [10] and many others. We recommend Fan et al. [12] and Lam [23] for detailed overviews of such methods.

Methods which estimate the inverse covariance matrix directly, such as neighbourhood selection [29], modified Cholesky decompositions [21] and ℓ_1 regularization approaches [7,14], prevent inversion errors and can improve computation time. However, these methods still suffer from the potentially impractical assumption that the inverse covariance matrix should be sparse.

Pronzato et al. [34] introduced a family of so-called k -simplicial distances which, according to the claim in that paper, resemble the squared Mahalanobis distance and could be used in situations where the data is degenerate or close to being degenerate. These k -simplicial distances are defined as averaged volumes of all possible k -dimensional simplices formed by x and all points from the set X , for a user-defined $k \in \{1, \dots, d\}$. The distances are raised to a user-defined power $\delta > 0$. For general $\delta > 0$, the choice $k = 1$ gives the ℓ_δ -distance in \mathbb{R}^d . For $\delta = 2$, the parameter choices $k = 1$ and $k = d$ give distances proportional to the squared Euclidean and Mahalanobis distances, respectively. The choice of parameter k indicates the dimension of the simplices used, and can be chosen in such a way as to avoid the problems of degeneracy and small eigenvalues, which cause issues in other methods, such as the Moore–Penrose pseudo-inverse. When $\delta = 2$, there is a more efficient method of finding the k -simplicial distance using a matrix polynomial in W , rather than through computation of volumes of simplices, which is outlined in Sect. 2. For other values of the parameter δ , we demonstrate a sub-sampling method which improves computational speed, with very little observed change to the distances measured.

We also introduce a new distance, the k -minimal-variance distance, which constructs a matrix polynomial in W of degree $k - 1$. In this method, the constructed polynomial yields a squared distance with minimum variance. Like the k -simplicial distance, the k -minimal-variance distance can be used when W is singular, and produces a distance measure which accounts for correlations in the dataset. We explore the choice of the parameter k , and show that k can be relatively low to produce good results, making the k -minimal-variance distance a quick and viable alternative to the Mahalanobis distance. We will use examples to show how minimizing the variance of the distances can provide a more intuitive distance measure for use with correlated data.

We consider the performance of the k -minimal-variance distance and compare it to the k -simplicial distance, the Euclidean distance and the Mahalanobis distance. We also show in Sect. 5.2 that both distances proposed may produce more accurate results than the Euclidean and Mahalanobis distances when used for clustering applications.

The structure of this paper is as follows: Sect. 2 introduces and further studies the k -simplicial distance formulated in Pronzato et al. [34]. Section 3.1 explores the effects of the parameters k and δ in the k -simplicial distance through numerical examples. Section 3.2 introduces the sub-sampling method which reduces computation time and allows us to consider large sample sizes. Section 3.3 uses outlier detection examples to produce parameter recommendations for the k -simplicial distance. Section 4 introduces the k -minimal-variance distance. We compare the k -simplicial and k -minimal-variance distances to each other in both their efficiency at minimizing variance in Sect. 5.1 and compare their performance at clustering some real-life datasets

against the Mahalanobis and Euclidean distances in Sect. 5.2. We give our conclusions about the distance measures in Sect. 6.

2 *k*-Simplicial Distances

Assume we are given a set of N points in d -dimensions $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^{d \times N}$, with no assumptions on how this set of points has been generated. The sample mean and covariance matrix associated with X are defined, respectively, as follows:

$$\mu = \frac{1}{N} \sum_{j=1}^N x_j, \quad W = \frac{1}{N} \sum_{j=1}^N (x_j - \mu)(x_j - \mu)^\top. \tag{1}$$

The squared Euclidean distance between a point $x \in \mathbb{R}^d$ and the set X is

$$\rho_E^2(x, X) = (x - \mu)^\top (x - \mu).$$

The squared Mahalanobis distance between $x \in \mathbb{R}^d$ and X is defined by

$$\rho_M^2(x, X) = (x - \mu)^\top W^{-1} (x - \mu), \tag{2}$$

where W^{-1} is the inverse of the sample covariance matrix W . If the matrix W is singular then its inverse does not exist, and so the Moore–Penrose pseudo-inverse W^- is often used in place of W^{-1} in (2). A pseudo-inverse is a type of generalized inverse, satisfying some but not necessarily of all the properties of an inverse matrix; the Moore–Penrose pseudo-inverse satisfies the four Moore–Penrose conditions [17]. When we use W^- in the Mahalanobis distance we will call it the pseudo-Mahalanobis distance.

To compute the k -simplicial distance between $x \in \mathbb{R}^d$ and $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^{d \times N}$, we calculate volumes of k -dimensional simplices raised to a given power $\delta > 0$. Let $r \leq d$ be the intrinsic dimension of the dataset X , which is the rank of X when X is considered as a $d \times N$ matrix. The volumes of all k -dimensional simplices are zero for $k > r$ and so it makes no sense to use $k > r$.

Let $\mathcal{V}_k(x, z_1, \dots, z_k)$ be the volume of a k -dimensional simplex with vertices x and $z_1, \dots, z_k \in \mathbb{R}^d$. This volume can be computed by

$$\mathcal{V}_k(x, z_1, \dots, z_k) = \frac{1}{k!} \left| \det(Z^\top Z) \right|^{1/2},$$

where $\det(A)$ is the determinant of the matrix A and $|a|$ is the absolute value of the scalar a [34]. Z is the $d \times k$ matrix with columns $[(z_1 - x) \ (z_2 - x) \ \dots \ (z_k - x)]$. Let

$$J = \left\{ (j_1, \dots, j_k) \in \{1, \dots, N\}^k \mid j_1 < \dots < j_k \right\} \tag{3}$$

be the set of all ordered k -combinations of the indices in $\{1, \dots, N\}$. Define

$$P_{k,\delta}(x, X) = \frac{1}{\binom{N}{k}} \sum_{(j_1, j_2, \dots, j_k) \in J} \mathcal{V}_k^\delta(x, x_{j_1}, \dots, x_{j_k}), \tag{4}$$

which is the average volume of all k -dimensional simplices created by the query point x and points in X , raised to the power of a user-defined scalar $\delta > 0$. For given $\delta > 0$ and $1 \leq k \leq r$, the centre of the set X (that is, the k -simplicial multidimensional median) is defined as

$$\bar{\mu}_{k,\delta} = \arg \min_x P_{k,\delta}(x, X),$$

which may not be uniquely defined for $\delta \leq 1$. Recommendations for these parameters will be discussed in Sects. 3 and 5. We then define the k -simplicial outlyingness function by

$$O_{k,\delta}(x, X) = \frac{P_{k,\delta}(x, X)}{P_{k,\delta}(\bar{\mu}_{k,\delta}, X)} - 1. \tag{5}$$

The function (5) is non-negative, has value 0 at the centre of the sample and is unitless; these are the required properties that an outlyingness function must possess, see Wilks [42]. For any $\delta > 0$, we define the k -simplicial distance (here to the power of δ) from the query point x to the dataset X by

$$\rho_{k,\delta}^\delta(x, X) = c_{k,\delta} O_{k,\delta}(x, X) = c_{k,\delta} \left(\frac{P_{k,\delta}(x, X)}{P_{k,\delta}(\bar{\mu}_{k,\delta}, X)} - 1 \right), \tag{6}$$

where the constant $c_{k,\delta}$ is chosen so that

$$\frac{1}{N} \sum_{j=1}^N \rho_{k,\delta}^2(x_j, X) = 1. \tag{7}$$

The normalization (7) is introduced to ensure consistency of the k -simplicial distances for different k . In the next section we see that, for $\delta = 2$ and all $k \leq r$, this normalization constant is $c_{k,2} = 1/k$. For $\delta \neq 2$, we find constants $c_{k,\delta}$ numerically from (7).

As shown in [34, Theorem 5], for $\delta = 2$ and any eligible k we get $\bar{\mu}_{k,\delta} = \mu$, the sample mean. Moreover, we obtain, similarly to [34, Sect. 3.1]

$$\frac{1}{N} \sum_{j=1}^N P_{k,2}(x_j, X) = (k + 1) P_{k,2}(\mu, X). \tag{8}$$

We then define the squared k -simplicial distance (of order $\delta = 2$) from x to the dataset X as

$$\rho_{k,2}^2(x, X) = \frac{1}{k} O_{k,2}(x, X) = \frac{1}{k} \left(\frac{P_{k,2}(x, X)}{P_{k,2}(\mu, X)} - 1 \right). \tag{9}$$

The difference between (9) and the corresponding definition in [34, Eq. 17] is the introduction of the normalizing constant $1/k$, which provides consistency of the distances for different k in the sense that (7) holds for $\delta = 2$ and all $k = 1, 2, \dots, r$. The equality in (7) with $\delta = 2$ directly follows from (8).

Direct evaluation of the distances given by (9), i.e. by empirical calculation of the volumes of all $\binom{N}{k}$ simplices, can be computationally time-consuming. Below, we consider an alternative method for the case with $\delta = 2$, which is much faster and easier. Reducing this computational time when $\delta \neq 2$ is considered in Sect. 3.2.

Let $\Lambda = \{\lambda_1, \dots, \lambda_d\}$ be the set of eigenvalues of the sample covariance matrix W defined in (1). The elementary symmetric function of degree $k \leq d$ associated with the set Λ is given by

$$e_k(\Lambda) = \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq d} \lambda_{i_1} \dots \lambda_{i_k},$$

with $e_0(\Lambda) = 1$. If $k > r = \text{rank}(X)$ then $e_k(\Lambda) = 0$ and the k -simplicial distance is always 0.

For $k \leq r$, define the function

$$q_k(W) = \sum_{i=0}^{k-1} (-1)^i e_{k-i-1}(\Lambda) W^i \tag{10}$$

and the associated matrix

$$S_k = \frac{q_k(W)}{e_k(\Lambda)}.$$

As follows from [34, Sect. 3.2], for any $k \leq r$,

$$\rho_{k,2}^2(x, X) = \frac{1}{k} (x - \mu)^\top S_k (x - \mu). \tag{11}$$

Note that the matrices $S_k, k = 1, \dots, r$, are polynomials in the covariance matrix W .

Since $S_1 = I_d / \text{trace}(W)$, where I_d is the identity $d \times d$ matrix, for $k = 1$ the squared distance (11) is equal to the squared Euclidean distance divided by the trace of the covariance matrix W :

$$\rho_{1,2}^2(x, X) = (x - \mu)^\top S_1 (x - \mu) = (x - \mu)^\top \frac{q_1(W)}{e_1(\Lambda)} (x - \mu) = \frac{(x - \mu)^\top (x - \mu)}{\text{trace}(W)}.$$

When $k = d$ and W is invertible, we have $S_d = W^{-1}$ and therefore the squared k -simplicial distance (11) is equal to the squared Mahalanobis distance multiplied by a factor $1/d$ (for details see [34, Sect. 3.1]):

$$\rho_{d,2}^2(x, X) = \frac{1}{d} (x - \mu)^\top S_d (x - \mu) = \frac{1}{d} (x - \mu)^\top W^{-1} (x - \mu).$$

We prove the following theorem comparing the variance of the squared Euclidean distance, Mahalanobis distance and k -simplicial distance with $k = 2$ and $\delta = 2$.

Theorem 1 Assume $X = \{x_1, \dots, x_N\}$ is a set of N normally distributed d -dimensional vectors (data points) with sample mean μ and sample covariance matrix W , as defined in (1). Let $\Lambda = \{\lambda_1, \dots, \lambda_d\}$ be the set of eigenvalues of the matrix W , and assume $\text{rank}(X) = r \leq d$. Then

$$\text{Var} \left(\rho_{r,2}^2(x, X) \right) \leq \text{Var} \left(\rho_{2,2}^2(x, X) \right) < \text{Var} \left(\rho_{1,2}^2(x, X) \right),$$

where $\rho_{k,2}^2(x, X)$ is the squared k -simplicial distance between the point x and set X as defined in (6) with $\delta = 2$.

Proof We can write the k -simplicial distance between a point x and set X with $k = 2$, $\delta = 2$ as

$$\rho_{2,2}^2(x, X) = (x - \mu)^\top \frac{S_k}{k} (x - \mu) = (x - \mu)^\top \frac{S_2}{2} (x - \mu) = (x - \mu)^\top \frac{q_2(W)}{2e_2(\Lambda)} (x - \mu)$$

with $q_2(W) = e_1(\Lambda)I_d - W$, from (10). From (24) in Appendix A, the variance of the k -simplicial distance with $k = 2$, $\delta = 2$ may be written as

$$\text{Var} \left(\rho_{2,2}^2(x, X) \right) = 2\text{trace} \left(\left(\frac{S_2}{2} W \right)^2 \right) = \frac{\text{trace} \left(W^2 (e_1(\Lambda)I_d - W)^2 \right)}{2e_2(\Lambda)^2}. \tag{12}$$

Let $\eta_j = \sum_{i \neq j} \lambda_i = \sum_{i=1}^r \lambda_i - \lambda_j = e_1(\Lambda) - \lambda_j$. Consider the second-order elementary symmetric polynomial:

$$e_2(\Lambda) = \sum_{i < j} \lambda_i \lambda_j = \frac{1}{2} \sum_{i \neq j} \lambda_i \lambda_j = \frac{1}{2} \left(\sum_{i=1}^r \sum_{j=1}^r \lambda_i \lambda_j - \sum_{i=1}^r \lambda_i^2 \right) = \frac{1}{2} \sum_{j=1}^r \lambda_j \eta_j.$$

Then (12) can be rewritten as:

$$\begin{aligned} \text{Var} \left(\rho_{2,2}^2(x, X) \right) &= \frac{\text{trace} \left(W^2 (e_1(\Lambda)I_d - W)^2 \right)}{2e_2(\Lambda)^2} \\ &= \frac{\sum_{j=1}^d \lambda_j^2 (e_1(\Lambda) - \lambda_j)^2}{2 \left(\frac{1}{2} \sum_{j=1}^d \lambda_j \eta_j \right)^2} = \frac{2 \sum_{j=1}^r \lambda_j^2 \eta_j^2}{\left(\sum_{j=1}^r \lambda_j \eta_j \right)^2} \end{aligned} \tag{13}$$

and similarly, using (25) from Appendix A, we may write

$$\text{Var} \left(\rho_{1,2}^2(x, X) \right) = \frac{2 \sum_{j=1}^r \lambda_j^2}{\left(\sum_{j=1}^r \lambda_j \right)^2}. \tag{14}$$

Consider the denominator in (14). By the Cauchy–Schwartz inequality,

$$\left(\sum_{j=1}^r \lambda_j\right)^2 = \left(\sum_{j=1}^r 1 \cdot \lambda_j\right)^2 \leq \sum_{j=1}^r 1^2 \sum_{j=1}^r \lambda_j^2 = r \sum_{j=1}^r \lambda_j^2,$$

and so it follows that

$$\text{Var}\left(\rho_{1,2}^2(x, X)\right) = \frac{2 \sum_{j=1}^r \lambda_j^2}{\left(\sum_{j=1}^r \lambda_j\right)^2} \geq \frac{2 \sum_{j=1}^r \lambda_j^2}{r \sum_{j=1}^r \lambda_j^2} = \frac{2}{r} = \text{Var}\left(\rho_{r,2}^2(x, X)\right).$$

Again using the Cauchy–Schwartz inequality for the denominator in (13), we have

$$\left(\sum_{j=1}^r \lambda_j \eta_j\right)^2 \leq r \sum_{j=1}^r \lambda_j^2 \eta_j^2,$$

and so it follows

$$\text{Var}\left(\rho_{2,2}^2(x, X)\right) = \frac{2 \sum_{j=1}^r \lambda_j^2 \eta_j^2}{\left(\sum_{j=1}^r \lambda_j \eta_j\right)^2} \geq \frac{2 \sum_{j=1}^r \lambda_j^2 \eta_j^2}{r \sum_{j=1}^r \lambda_j^2 \eta_j^2} = \frac{2}{r} = \text{Var}\left(\rho_{r,2}^2(x, X)\right).$$

It remains to show that

$$\text{Var}\left(\rho_{1,2}^2(x, X)\right) = \frac{2 \sum_{j=1}^r \lambda_j^2}{\left(\sum_{j=1}^r \lambda_j\right)^2} \geq \frac{2 \sum_{j=1}^r \lambda_j^2 \eta_j^2}{\left(\sum_{j=1}^r \lambda_j \eta_j\right)^2} = \text{Var}\left(\rho_{2,2}^2(x, X)\right). \tag{15}$$

The validity of the inequality in (15) does not depend on the change $\lambda_i \rightarrow c\lambda_i$ for all i and for any constant $c > 0$. Therefore, we can choose $\lambda_1, \lambda_2, \dots, \lambda_r$ such that $\sum_{i=1}^r \lambda_i = 1$ and express quantities in (15) as moments of a random variable ξ concentrated on $[0, 1]$ having values λ_i with probabilities λ_i .

Let $\tau_j = E[\xi^j]$. In this notation, we have the following properties:

$$\begin{aligned} \sum_{j=1}^r \lambda_j &= E[\xi^0] = 1, \\ \sum_{j=1}^r \lambda_j^2 &= E[\xi] = \tau_1, \\ \sum_{j=1}^r \lambda_j \eta_j &= \sum_{j=1}^r \lambda_j (1 - \lambda_j) = 1 - E[\xi] = 1 - \tau_1, \\ \sum_{j=1}^r \lambda_j^2 \eta_j^2 &= \sum_{j=1}^r \lambda_j^2 (1 - \lambda_j)^2 = E[\xi] - 2E[\xi^2] + E[\xi^3] = \tau_1 - 2\tau_2 + \tau_3. \end{aligned}$$

Table 1 Details of datasets generated to be used in examples

Notation	d	r	Eigenvalues
Λ_A	10	9	([100, 4, 3, 2, 1], [0.0001] \times 4, [0])
Λ_B	50	40	([100, 10], [1] \times 5, [0.0001] \times 33, [0] \times 10)
Λ_C	50	22	([100, 100], [1] \times 10, [0.00001] \times 10, [0] \times 28)

500 points are sampled from a multivariate normal distribution with zero mean and diagonal covariance matrix, with eigenvalues given in the table. The rank r of the dataset is also given

Using these properties, the inequality in (15) has the form

$$\tau_1 \geq \frac{\tau_1 - 2\tau_2 + \tau_3}{(1 - \tau_1)^2}.$$

Rearranging gives $\tau_1^3 + 2\tau_2 - 2\tau_1^2 - \tau_3 \geq 0$, which is true for all probability measures on $[0, 1]$. □

3 Applying the k -Simplicial Distance

3.1 Choosing k in the k -Simplicial Distance

The choice of the parameter k is integral to the performance of the k -simplicial distance. In this section, we show how different choices of k affect the distance through experimental results. We use three examples, in each of which we generate $N = 500$ points from a d -dimensional multivariate normal distribution, with zero mean and diagonal covariance matrix. The value of d and the eigenvalues of the covariance matrix used to generate the points are given in Table 1. We use the sample covariance matrix when computing our distance measures, and so the true eigenvalues of the matrix will differ slightly from those in the table.

For given values of $k \leq r = \text{rank}(X)$, we find the k -simplicial distances between all points in the dataset to the dataset X itself, for both $\delta = 2$ and $\delta = 1$. Note that for distances using $\delta = 1$, sub-sampling is used to find the distance, using the method described in Sect. 3.2. We compare the empirical cumulative distribution functions (CDF) produced by the k -simplicial distances in Figs. 1, 2 and 3. For examples with $\delta = 2$ we also consider the squared pseudo-Mahalanobis distance multiplied by $1/r$, which is equal to the k -simplicial distance with $k = r$.

Example 1 Eigenvalues $\Lambda = \Lambda_A$. The CDFs for the distances measured for Dataset A using the k -simplicial distance with $\delta = 2$ are given in Fig. 1a and indicate that the squared Euclidean distance (proportional to the k -simplicial distance with $k = 1, \delta = 2$) produces a large range of distances with high variance, when compared to the distances produced when using other values of k . We see, in the $\delta = 2$ case, low values of k (compared to the rank $r = 9$) begin to converge away from the squared

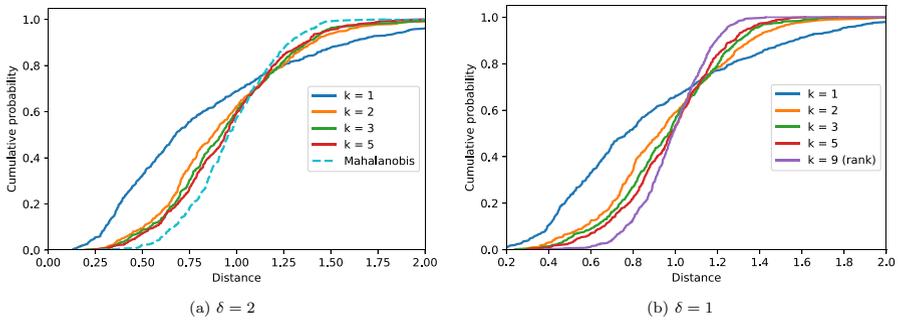


Fig. 1 CDFs of k -simplicial distances with eigenvalues $\Lambda = \Lambda_A$ **a** $\delta = 2$, **b** $\delta = 1$

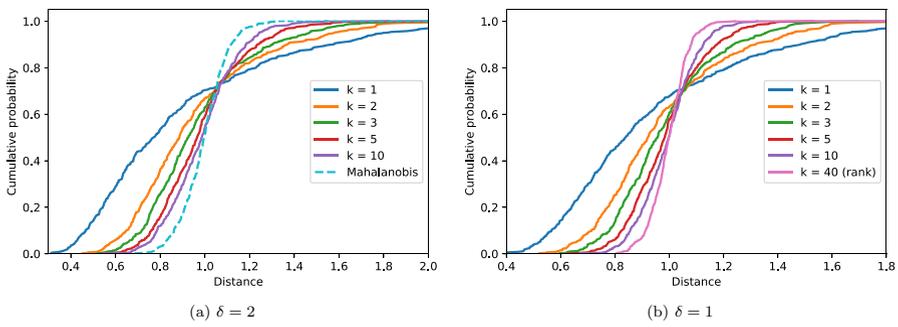


Fig. 2 CDFs of k -simplicial distances with eigenvalues $\Lambda = \Lambda_B$ **a** $\delta = 2$, **b** $\delta = 1$

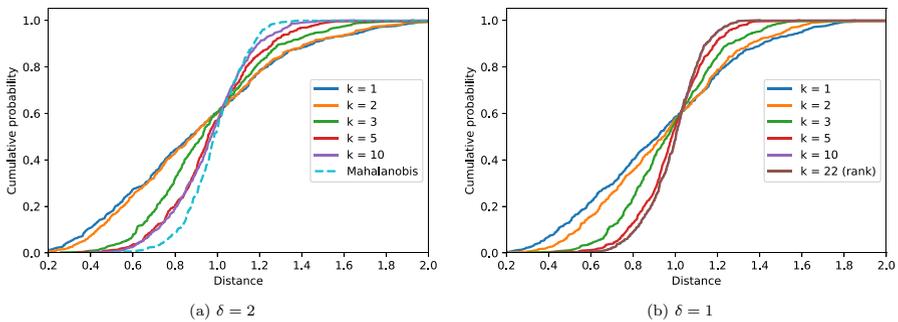


Fig. 3 CDFs of k -simplicial distances with eigenvalues $\Lambda = \Lambda_C$ **a** $\delta = 2$, **b** $\delta = 1$

Euclidean distance, and towards the squared pseudo-Mahalanobis distance quickly. Figure 1b shows a similar pattern for the distance with $\delta = 1$.

Example 2 Eigenvalues $\Lambda = \Lambda_B$. The CDFs for the k -simplicial distances using $\delta = 2$ on Dataset B are given in Fig. 2a. For relatively low values of k (compared to the rank $r = 40$), such as $k = 10$, we see the distances converging to those produced when $k = r$, i.e. the pseudo-Mahalanobis distance in the case where $\delta = 2$. A similar

profile is observed for $\delta = 1$ in Fig. 2b, with CDFs of distances converging towards the CDF with $k = r = 40$ as k increases.

Example 3 Eigenvalues $\Lambda = \Lambda_C$. The CDFs for the distances with $\delta = 2$ are given in Fig. 3a. Again, we see for relatively low values of k (compared to rank $r = 22$) the distance measure converges towards the distance where $k = r$. Note that for the k -simplicial distances with $\delta = 1$, $k = 10$ in Fig. 3b, the CDF lies underneath that of $k = 22$, as the distances produced are so similar.

Figures 1, 2 and 3 all demonstrate that the k -simplicial distance transitions from the squared Euclidean distance multiplied by $1/\text{trace}(W)$ to the squared Mahalanobis distance multiplied by $1/r$ for $\delta = 2$ as k increases. A similar monotonic behaviour is shown for $\delta = 1$. The eigenvalues of the covariance matrix have an effect on what an appropriate choice of k may be. It is important to ensure the most influential dimensions (that is, those with the largest eigenvalues) are all considered, by taking k larger than the number of large eigenvalues.

For example, consider Fig. 3. The two large eigenvalues in Λ_C result in $k = 2$ behaving similarly to $k = 1$, particularly in the $\delta = 2$ case, whereas in Fig. 1, the CDF produced using the distance with $k = 2$ is very different to the CDF where $k = 1$, as there is only one large eigenvalue.

In general, we recommend using a value of k that is larger than the number of ‘large’ eigenvalues the covariance matrix W has, relative to the size of the other eigenvalues. This is easier to see when there is a clear elbow or ‘drop-off’ in the value of the eigenvalues. Otherwise, it can be appropriate to find the k -simplicial distances with several values of k and measure the best value according to some metric appropriate to the task. This is a common method for choosing a parameter value in many parameter-dependent tasks, such as K -means clustering.

Not much performance gain is made by choosing a value of k that also encompasses the smaller eigenvalues. As an example of this, see Fig. 2, where there are seven ‘large’ eigenvalues, 33 ‘small’ eigenvalues and 10 zero eigenvalues. Using $k = 10$ does not give a huge improvement in performance compared to using $k = 5$ (where performance is measured here by the minimizing of variance) but it is computationally more expensive.

3.2 Numerical Computation of k -Simplicial Distances Using Sub-Sampling

When $\delta \neq 2$, the k -simplicial distance is calculated by averaging the volumes of all $\binom{N}{k}$ simplices formed with x and X . This can be computationally intensive, particularly for large N and d . To circumvent this problem, we can sample a subset of the simplices to reduce computation time to milliseconds. The size of the sub-sample of simplices depends on the user’s wish for precision. This size does not have to be large to achieve practically accurate approximations, which will be demonstrated in the examples that follow, where we use less than 0.05% of all possible simplices when we use $k = 3$, and less than 0.0004% when using $k = 4$.

Let J be as defined in (3). To compute the k -simplicial distances, we have to compute the values of $P_{k,\delta}(x, X)$ defined in (4). The procedure to approximate these

values is as follows. For any sampling proportion $\gamma \in [0, 1]$, we form $J^{(\gamma)}$, a subset of J of size $|J^{(\gamma)}| = \lceil \gamma \times \binom{N}{k} \rceil$ and approximate (4) with

$$P_{k,\delta,\gamma}(x, X) = \frac{1}{|J^{(\gamma)}|} \sum_{(j_1, \dots, j_k) \in J^{(\gamma)}} \mathcal{V}_k^\delta(x, x_{j_1}, \dots, x_{j_k}).$$

A simple but efficient way of constructing $J^{(\gamma)}$ consists of taking random samples of size k without replacement from the set $\{1, 2, \dots, N\}$, see Blom [4]. This reduces computation time dramatically, and in examples that follow we see such sub-sampling is highly effective in producing results extremely close to those of the ‘full’ distance measure, in which we average the volumes over all available simplices.

In the following examples we revisit the sets of eigenvalues given in Table 1 and calculate the distances from all points in a dataset to the dataset itself, using the full sample (where possible) and then a smaller sample using 10,000 simplices. We generate data according to the procedure outlined in the beginning of Sect. 3.1. We compare the effect that different sampling sizes have on the distribution of distances calculated though investigating histograms and moments of these distances. Our analysis in Sect. 3.1 indicates that using low values of k gives good performance, and is less computationally intensive than using higher values, so we will use $k = 3$ and $k = 4$ in the sub-sampling examples that follow.

Example 1 Eigenvalues $\Lambda = \Lambda_A$. Figure 4a shows histograms of the distances between all points of dataset A to the dataset itself, as produced by the k -simplicial distance with $\delta = 2$, $k = 3$. The blue solid histogram shows the ‘full’ distances with no sub-sampling (using polynomials), and the orange dotted histogram shows the sub-sampled distance with 10,000 simplices. This is repeated for other parameters in the rest of Fig. 4, as detailed in the captions.

These histograms show that the distribution of distances produced using a small sample of simplices is extremely similar to the distribution of distances produced using the full sample of simplices available. For the examples using $\delta = 1$, we cannot produce the full distance directly as it requires the computation of the volume of $\binom{500}{k}$ simplices, so we compare the distances produced using a sub-sample of 10,000 simplices to the distances when using a larger sub-sample of simplices (1% of the total amount of simplices in the $k = 3$ case, 0.01% in the $k = 4$ case). In both cases, the distribution of the larger sample and the 10,000 simplex sample remain very similar. Table 2a and b also demonstrate this, with the summary statistics remaining close even for small samples.

Example 2 Eigenvalues $\Lambda = \Lambda_B$. We again see in Fig. 5 and Table 3a and b that using a low number of simplices (compared to the full amount of simplices available, or a large sample) produces distances that are mostly the same as the full distance measure. This example illustrates that the sampling method is effective even in cases with a lot of small and zero eigenvalues.

Example 3 Eigenvalues $\Lambda = \Lambda_C$. Figure 6 and Table 4a and b show that the number of small or zero eigenvalues does not influence the performance of the sub-sampling. Overall, we see sub-sampling is an effective way to drastically reduce computation

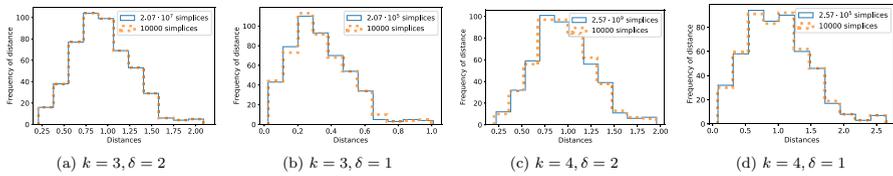


Fig. 4 Histograms to compare the distribution of the k -simplicial distances from all points to the mean for eigenvalues Λ_A with different parameters k and δ for different sampling amounts. Blue, solid histograms show the distances produced using the full sample of simplices when $k = 3$, and a larger sample when $k = 4$. The orange dotted histograms show the distances produced when using a sample of 10,000 simplices.

Table 2 Summary statistics of the distances when (a) $k = 3$, (b) $k = 4$ with eigenvalues Λ_A

	(a) $k = 3$		$\delta = 1$	
	$\delta = 2$			
	$2.07 \cdot 10^7$	10^4	$2.07 \cdot 10^5$	10^4
Mean	1.00	1.00	1.00	1.00
Variance	0.11	0.11	0.27	0.28
Skewness	0.49	0.49	0.81	0.81
Kurtosis	0.31	0.31	0.85	0.87

	(b) $k = 4$		$\delta = 1$	
	$\delta = 2$			
	$2.57 \cdot 10^9$	10^4	$2.57 \cdot 10^5$	10^4
Mean	1.00	1.00	1.00	1.00
Variance	0.10	0.10	0.23	0.23
Skewness	0.38	0.40	0.61	0.60
Kurtosis	0.10	0.12	0.34	0.34

Table headers indicate the number of simplices sampled

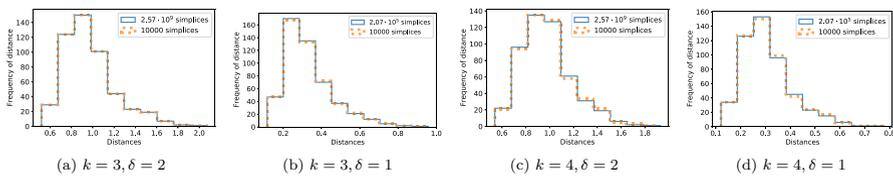


Fig. 5 Histograms to compare the distribution of the k -simplicial distances from all points to the mean for eigenvalues Λ_B with different parameters k and δ for different sampling amounts. Blue, solid histograms show the distances produced using the full sample of simplices when $k = 3$, and a larger sample when $k = 4$. The orange dotted histograms show the distances produced when using a sample of 10,000 simplices.

time while maintaining the same results as the full k -simplicial distance. This means that using the distance with $\delta \neq 2$ is much more accessible than it otherwise would be.

Table 3 Summary statistics of the distances when (a) $k = 3$, (b) $k = 4$ with eigenvalues Λ_B

	(a) $k = 3$		$\delta = 1$	
	$\delta = 2$	$\delta = 1$	$2.07 \cdot 10^7$	10^4
Mean	1.00	1.00	1.00	1.00
Variance	0.06	0.06	0.16	0.16
Skewness	1.08	1.08	1.29	1.29
Kurtosis	1.42	1.42	2.00	1.97

	(b) $k = 4$		$\delta = 1$	
	$\delta = 2$	$\delta = 1$	$2.57 \cdot 10^5$	10^4
Mean	1.00	1.00	1.00	1.00
Variance	0.04	0.04	0.11	0.11
Skewness	0.91	0.91	1.09	1.08
Kurtosis	1.20	1.16	1.58	1.52

Table headers indicate the number of simplices sampled

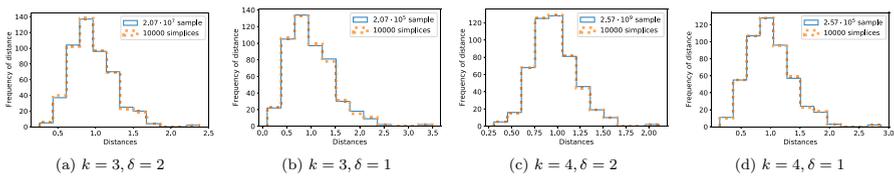


Fig. 6 Histograms to compare the distribution of the k -simplicial distances from all points to the mean for eigenvalues Λ_C with different parameters k and δ for different sampling amounts. Blue, solid histograms show the distances produced using the full sample of simplices when $k = 3$, and a larger sample when $k = 4$. The orange dotted histograms show the distances produced when using a sample of 10,000 simplices.

3.3 Outlier Labelling Example

In this section, we illustrate one potential application of the k -simplicial distance measure. The k -simplicial distance could be a useful tool in identifying outlying points in high-dimensional degenerate datasets, where the Euclidean distance struggles to measure distance meaningfully due to the sparse and correlated nature of the data, and the Mahalanobis relies on the inversion of a matrix possessing many small (and potentially zero) eigenvalues. We investigate how different values of the parameter k and the scalar power δ perform in identifying outliers.

We perform the following experiments. We consider three 10-dimensional examples, each with different sets of data. Each dataset D_i , $i = \{I, II, III\}$, is made up of two clusters: $D_i = D_{i,1} + D_{i,2}$. The first cluster $D_{i,1}$ has 450 points, mean μ_1 as specified in Table 5 and covariance matrix produced by a matrix with eigenvalues as specified in the table, rotated by a rotation matrix. The second cluster $D_{i,2}$ has 50 points, a different mean μ_2 but the same covariance matrix as $D_{i,1}$. By doing this, we

Table 4 Summary statistics of the distances when (a) $k = 3$, (b) $k = 4$ with eigenvalues Λ_C

	(a) $k = 3$			
	$\delta = 2$		$\delta = 1$	
	$2.07 \cdot 10^7$	10^4	$2.07 \cdot 10^5$	10^4
Mean	1.00	1.00	1.00	1.00
Variance	0.08	0.08	0.22	0.22
Skewness	0.81	0.81	1.10	1.11
Kurtosis	1.64	1.64	2.43	2.47
	(b) $k = 4$			
	$\delta = 2$		$\delta = 1$	
	$2.57 \cdot 10^9$	10^4	$2.57 \cdot 10^5$	10^4
Mean	1.00	1.00	1.00	1.00
Variance	0.06	0.06	0.14	0.14
Skewness	0.62	0.62	0.86	0.86
Kurtosis	1.36	1.34	1.82	1.84

Table headers indicate the number of simplices sampled

Table 5 Datasets to be used in outlier labelling

Dataset	Eigenvalues	μ_1	μ_2
<i>I</i>	$\Lambda_I = ([100, 10, 1, 1], [0.00001] \times 5, [0])$	$([0] \times 10)$	$([1] \times 10)$
<i>II</i>	$\Lambda_{II} = ([100, 10, 1, 1], [0.00001] \times 5, [0])$	$([0] \times 10)$	$([0] \times 5, [1] \times 5)$
<i>III</i>	$\Lambda_{III} = ([100, 4, 3, 2, 1], [0.00001] \times 4, [0])$	$([0] \times 10)$	$([1] \times 10)$

Datasets are made of two clusters of different sizes and different means, but the same covariance matrix

test the robustness of the distances against rotations and correlations in the data, as well as its ability to tell two similar but separate clusters apart.

We measure the distance of all points in the dataset D_i to the largest cluster, $D_{i,1}$. We label the furthest 50 points from this cluster $D_{i,1}$ as outliers for each dataset i . We consider how many of the points the k -simplicial distances correctly label as outliers from $D_{i,1}$ for different values of k and δ . If it were to incorrectly label all the outlying points as inliers, we would have a minimum value of 400. If the distance correctly labels all points, we will get a value of 500. Table 6 contains the number of points correctly labelled by the distance measures using different values of k and δ for the k -simplicial distances. In Table 7, we provide the Area Under the Receiver Operating Characteristic Curve (AUC) score for the labels produced by the distances, for different values of k and δ . The AUC score measures the overall performance of a binary classifier, where a score of 1 indicates a perfect labelling and 0.5 is the minimum score [18].

Considering the k -simplicial distance with $\delta = 2$, we see that the values of k which perform best are those slightly larger than the number of ‘large’ eigenvalues. For Dataset *I*, we have 4 ‘large’ eigenvalues and values of $k = 5, 6$ perform best when using $\delta = 2$. Similar results are shown in Datasets *II* and *III*. Larger values of k begin

Table 6 Number of points correctly labelled by the distance measures using different values of k for the k -simplicial distances

k	Dataset I		Dataset II		Dataset III	
	$\delta = 2$	$\delta = 1$	$\delta = 2$	$\delta = 1$	$\delta = 2$	$\delta = 1$
1	412	412	408	408	406	406
2	412	412	418	418	420	416
3	440	444	432	442	436	436
4	492	500	482	498	444	448
5	500	500	500	500	482	496
6	500	500	500	500	500	500
7	414	500	416	500	498	500
8	410	500	408	500	406	500
9	412	500	410	500	406	500

Minimum score is 400, maximum score is 500. Bold values indicate perfect labellings

Table 7 AUC scores for outlier detection when using the k -simplicial distance with different values of k and δ

k	Dataset I		Dataset II		Dataset III	
	$\delta = 2$	$\delta = 1$	$\delta = 2$	$\delta = 1$	$\delta = 2$	$\delta = 1$
1	0.51	0.51	0.49	0.49	0.48	0.48
2	0.51	0.51	0.54	0.54	0.56	0.53
3	0.67	0.69	0.62	0.68	0.64	0.64
4	0.96	1.00	0.90	0.99	0.69	0.71
5	1.00	1.00	1.00	1.00	0.90	0.98
6	1.00	1.00	1.00	1.00	1.00	1.00
7	0.52	1.00	0.53	1.00	0.99	1.00
8	0.50	1.00	0.49	1.00	0.48	1.00
9	0.51	1.00	0.50	1.00	0.48	1.00

Minimum score is 0.5, maximum is 1. Bold values indicate perfect labellings

to break down when $\delta = 2$ as they require the use of the smaller eigenvalues when forming the simplices. This indicates that lower values of k outperform the squared pseudo-Mahalanobis distance multiplied by $1/r$.

Distances using $\delta = 1$ are more robust to the effect of degeneracy. The performance improves as k increases, but unlike the $\delta = 2$ case, there is no breakdown in success once k encompasses the smaller eigenvalues too, making it less sensitive to the choice of k than the distance with $\delta = 2$. These distances were computed using very low sub-sampling amounts, and so there is not considerable computational time disadvantage in using $\delta = 1$ over $\delta = 2$. Overall, this example illustrates that using $\delta = 1$, even with sub-sampling, can give a more stable distance measure than using $\delta = 2$ as k increases, particularly for outlier detection applications.

4 *k*-Minimal-Variance Distances

Assume that X is a normally distributed d -dimensional dataset, with sample mean μ and sample covariance matrix W . We now introduce a family of generalized squared distances from a point x to the set X , in the form

$$\rho_A^2(x, X) = (x - \mu)^\top A(x - \mu), \tag{16}$$

where A is a matrix polynomial in W of user-defined degree $k - 1 \leq r$, where $r = \text{rank}(X)$. From the moments of a quadratic form (see (23) in Appendix A), we have

$$E\left(\rho_A^2(x, X)\right) = \text{trace}(AW), \quad \text{Var}\left(\rho_A^2(x, X)\right) = 2\text{trace}\left((AW)^2\right).$$

For given $k \leq r$, we wish to find the matrix A such that $\text{trace}(AW) = d$ holds and $\text{Var}\left(\rho_A^2(x, X)\right)$ is minimized. The condition $\text{trace}(AW) = d$ ensures the identifiability of a solution from the minimization of the Lagrange function (20) and weights the solution towards W^{-1} (if W^{-1} exists).

We can motivate minimizing the variance of the distances by considering the CDFs in Figs. 1, 2 and 3. The Mahalanobis distance has the smallest variance out of all distances considered, and can be written as a $(d - 1)$ -degree polynomial when the covariance matrix is non-singular. We aim to replicate the minimization of the variance of the distances produced through lower degree polynomials that also work in the case of a singular covariance matrix, making for a quicker and more versatile method.

Let A be a polynomial in W , expressed as

$$A = \sum_{i=0}^{k-1} \theta_i W^i. \tag{17}$$

The first moment of the distance (16) is $E\left(\rho_A^2(x, X)\right) = \sum_{i=0}^{k-1} \theta_i \text{trace}(W^{i+1})$. Note that $\text{trace}(W^i) = \sum_{j=1}^d \lambda_j^i$, where $\lambda_1, \dots, \lambda_d$ are the eigenvalues of W . Then let $\theta = (\theta_0, \theta_1, \dots, \theta_{k-1})^\top$, and define the matrix

$$Y = (y_{ji})_{j=1, \dots, d, i=0, \dots, k-1} = \left(\lambda_j^{i+1}\right)_{j=1, \dots, d, i=0, \dots, k-1}.$$

Using this notation, the variance of the distance (16) can be written using (23) from Appendix A as

$$\text{Var}\left(\rho_A^2(x, X)\right) = 2\theta^\top Y^\top Y \theta. \tag{18}$$

Proposition 1 For given $k \leq r$, let \mathcal{A}_{k-1} be the set of matrices A of the form (17) satisfying the condition $\text{trace}(AW) = d$. The solution to the optimization problem

$$A^* = \arg \min_{A \in \mathcal{A}_{k-1}} \text{Var}\left(\rho_A^2(x, X)\right) = \sum_{i=0}^{k-1} \theta_i^* W^i$$

is given by the set of coefficients

$$\theta^* = \frac{d}{S^\top(Y^\top Y)^{-1}S} (Y^\top Y)^{-1}S,$$

where

$$S^\top = \left(\text{trace}(W^{i+1}) \right)_{i=0}^{k-1} = \left(\text{trace}(W), \text{trace}(W^2), \dots, \text{trace}(W^k) \right).$$

For the matrix A^* , the variance of the distance (16) is given by

$$\text{Var}(\rho_{A^*}^2(x, X)) = 2d^2 / \left(S^\top(Y^\top Y)^{-1}S \right). \tag{19}$$

If the matrix $Y^\top Y$ is non-degenerate, then $(Y^\top Y)^{-}$ becomes $(Y^\top Y)^{-1}$.

Proof The Lagrange function used to minimize the variance (18) with the constraint $\text{trace}(AW) = \sum_{i=0}^{k-1} \theta_i \text{trace}(W^{i+1}) = d$ is

$$\mathcal{L}(\theta, \omega) = \frac{1}{2} \theta^\top Y^\top Y \theta - \omega(\theta^\top S - d). \tag{20}$$

Note that $\theta^\top S = \text{trace}(AW)$.

Differentiating the Lagrange function with respect to θ and setting the result equal to 0, we get $Y^\top Y \theta = \omega S$, which gives: $\theta^* = \omega(Y^\top Y)^{-1}S$. The required value of ω is found from the unbiasedness condition $\text{trace}(AW) = d$, giving $\omega = d/S^\top(Y^\top Y)^{-1}S$. Substituting θ^* into (18) gives the variance of the distance as in (19). \square

5 Experimental Analysis

5.1 Efficiency of k -Minimal-Variance Distances Compared to k -Simplicial Distances

We consider the efficiency of the k -minimal-variance distances (16) with $A = A^*$ as derived in Theorem 1 and k -simplicial distances (11) with $\delta = 2$ relative to the (sometimes pseudo-)Mahalanobis distance. As we consider the Mahalanobis distance multiplied by $1/r$ to align with the k -simplicial distance, we must also consider the k -minimal-variance distance multiplied by $1/d$ for comparability.

We define the efficiency of the k -minimal-variance distances as

$$\text{eff}_{MV}^{(k)} = \frac{\text{Var} \left(\rho_{A^*/d}^2(x, X) \right)}{\text{Var} \left(\rho_{r,2}^2(x, X) \right)} = \frac{2 / \left(S^\top(Y^\top Y)^{-1}S \right)}{2/r}, \tag{21}$$

Table 8 Efficiencies (21) and (22) for different k , with three different sets of eigenvalues of the covariance matrix W given by $\Lambda_i, i = 1, 2, 3$.

k	Λ_1		Λ_2		Λ_3	
	$\text{eff}_{simp}^{(k)}$	$\text{eff}_{MV}^{(k)}$	$\text{eff}_{simp}^{(k)}$	$\text{eff}_{MV}^{(k)}$	$\text{eff}_{simp}^{(k)}$	$\text{eff}_{MV}^{(k)}$
1	1.51	1.51	2.16	2.16	2.06	2.06
2	1.40	1.18	1.64	1.20	1.62	1.22
3	1.32	1.05	1.38	1.02	1.37	1.03
4	1.24	1.01	1.23	1.00	1.21	1.00
5	1.17	1.00	1.13	1.00	1.12	1.00
6	1.12	1.00	1.07	1.00	1.06	1.00
7	1.07	1.00	1.04	1.00	1.02	0.99
8	1.03	1.00	1.01	1.00	1.00	1.00
9	1.01	1.00	1.00	1.00	1.00	1.00
10	1.00	0.00	1.00	0.00	N/A	0.00

with $\text{Var}(\rho_{A^*}^2(x, X))$ derived in (19). We define the efficiency of the k -simplicial distances with $\delta = 2$ as

$$\text{eff}_{simp}^{(k)} = \frac{\text{Var}(\rho_{k,2}^2(x, X))}{\text{Var}(\rho_{r,2}^2(x, X))} = \frac{(2/k^2)\text{trace}((S_k W)^2)}{2/r}, \tag{22}$$

with $\text{Var}(\rho_{k,2}^2(x, X))$ stated in (24) in Appendix A.

We generate $N = 500$ points $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^{d \times N}$ from a d -dimensional multivariate normal distribution with zero mean and diagonal covariance matrix W with eigenvalues $\Lambda = \{\lambda_1, \dots, \lambda_d\}$. We take the eigenvalues to be:

$$\Lambda_1 = (10, 7, 6, 5, 4, 3, 2, 1, 1, 1),$$

$$\Lambda_2 = (10, 4, 3, 2, 1, 1, 1, 1, 1, 1),$$

$$\Lambda_3 = (10, 5, 3, 2, 1, 1, 1, 1, 1, 0).$$

Table 8 demonstrates the high-efficiency of the k -minimal-variance distances even for small k . Note that $k - 1$ is the order of the polynomial in W minimizing (18); in this example, linear and quadratic polynomials perform well. The efficiency of the k -simplicial distances improves as k gets larger but also has variance tolerably close to that of the Mahalanobis distance even for k significantly smaller than r .

For larger dimensions, with covariance matrices possessing a number of zero eigenvalues, the examples are more striking. Table 9 gives the results of performing the same exercise on the datasets described in Sect. 3.1, with eigenvalues given in Table 1. Table 9 shows that the k -minimal-variance distances start to have similar variance to the squared (pseudo-)Mahalanobis distance using much lower values of k than in the k -simplicial distance with $\delta = 2$. For values as low as $k = 2$, we see the variance of the k -minimal-variance distance is much closer to that of the Mahalanobis distance than the k -simplicial distances.

Table 9 Efficiencies (21) and (22) for different k , with three different sets of eigenvalues of the covariance matrix W as given in Table 1 in Sect. 3.1.

k	Λ_A		Λ_B		Λ_C	
	$\text{eff}_{\text{simp}}^{(k)}$	$\text{eff}_{MV}^{(k)}$	$\text{eff}_{\text{simp}}^{(k)}$	$\text{eff}_{MV}^{(k)}$	$\text{eff}_{\text{simp}}^{(k)}$	$\text{eff}_{MV}^{(k)}$
1	7.58	7.58	24.25	24.25	10.00	10.00
2	2.83	2.07	9.79	4.72	9.17	5.55
3	2.09	1.84	5.87	1.95	4.77	1.87
4	1.87	1.80	4.21	1.90	3.20	1.83

Table 10 Real datasets used to evaluate performance of distances when used with K -Means clustering

Dataset	d	N	No. of Clusters
Iris	4	150	3
Wine	13	178	3
Image Seg.	19	210	7
Digits	64	1797	10
Protein	77	1080	8

d is the number of variables in the dataset, and N is the number of observations

We note little performance gain when choosing $k > 3$ in these examples, indicating that A^* even with small k is often a good enough approximation to the inverse of the covariance matrix from the viewpoint of the distance generated by this matrix. From an efficiency perspective, the k -minimal-variance produces better results at a lower computational cost than the k -simplicial distance.

5.2 Comparison of Performances of the K -Means Clustering Algorithm with Different Distances

We compare the performance of the K -Means clustering algorithm [26] when applied using the Euclidean distance, the Mahalanobis distance, the k -minimal-variance distance and the k -simplicial distance (with $\delta = 2$). We do this by applying K -Means to 5 real datasets, obtained from the UCI Machine Learning Repository [11], with the exception of the ‘Digits’ dataset, which was obtained through the Python package sklearn’s data loading functions [31]. The details of these datasets are given in Table 10.

Each dataset was appropriately preprocessed: rows with missing values were removed, and the data was normalized such that each variable has values in range $[0, 1]$. It is important to note that the K used in the K -Means clustering algorithm is used to indicate how many clusters we seek, and is different to the k used in our distance measures. For each dataset, the choice of K in the K -Means algorithm is used as the ‘true’ number of clusters, given in Table 10, as these datasets are all fully labelled.

The K -Means algorithm is classically applied using the Euclidean distance, but research has shown success in applying the algorithm with the Mahalanobis distance to exploit the covariance structure of a dataset (see Gnanadesikan et al. [16], or more

recently [30]). The method of applying K -Means with the Mahalanobis, k -minimal-variance or k -simplicial distances is given in Algorithm 1, following the algorithm given in Melnykov and Melnykov [8]. These distance measures require the covariance matrix of each cluster; and as such we require initial estimates of the clusters. We obtain these initial estimates by performing a few iterations of the K -Means algorithm using the Euclidean distance. Clearly, this initial estimate can have a large influence on the resulting clusters found by the other distances, and so we run the K -Means algorithm 1000 times for each distance.

Algorithm 1: Applying K -Means with the Mahalanobis, k -minimal-variance or k -simplicial distance

Apply K -Means clustering with Euclidean distance for a small number of iterations;
 Calculate the centroids μ_i of each cluster C_i by taking the mean of all points in the cluster
while *Distance between old centroids and new centroids is less than some small chosen tolerance*
do
 Calculate the covariance matrix W_i of each cluster C_i ;
 Calculate the distance from each point to each cluster C_i using the Mahalanobis,
 k -minimal-variance or k -simplicial distance with the corresponding covariance matrix W_i and
 cluster centroid μ_i ;
 Assign each point to its closest cluster;
 Calculate new centroids μ_i for each new cluster
end

As this is a supervised task, we can compare the labels given by K -Means to the ‘true’ labels to assess the performance of the clustering algorithm. We do this using two external evaluation methods, namely the adjusted rand (AR) score [20,31] and the purity score [28]. We use these two different evaluation methods to corroborate the results. The adjusted rand score is calculated as follows.

Let L_T be the vector of true labels, and let L_P be the labels assigned by the K -Means clustering. Define a as the number of pairs of points in the same set in L_T and in the same set in L_P , i.e. the number of points whose labels are the same in L_T and L_P . Define b as the number of pairs of points in different sets in L_T and in different sets in L_P , i.e. the number of points whose labels are different in L_T and L_P . The unadjusted rand score is given by

$$R = \frac{a + b}{\nu},$$

where ν is the total number of possible pairs in the dataset, without ordering. The unadjusted rand score does not account for the possibility that random label assignments can perform well, so we discount the expected rand score $E[R]$ of random labellings by defining the adjusted rand score as

$$AR = \frac{R - E[R]}{\max(R) - E[R]}.$$

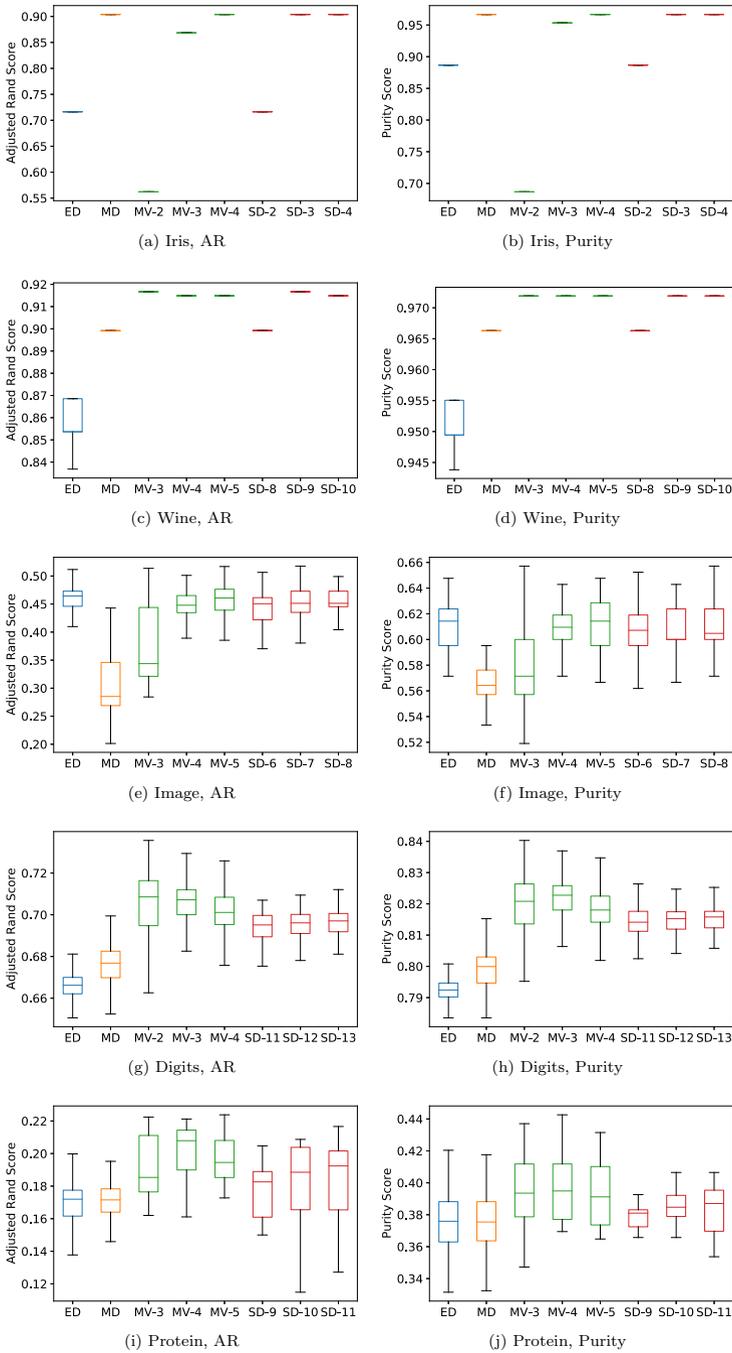


Fig. 7 Adjusted rand scores and purity scores of the clusterings produced by K -Means when using different distance measures. ED: Euclidean distance, MD: Mahalanobis distance, MV- k : Minimal-variance distance with parameter k , SD- k : Simplicial distance with parameter k

Table 11 Iris dataset: Median AR scores (and standard deviations) for each given distance

k	k -simp	k -MV
2	0.716 (0)	0.562 (0)
3	0.904 (0)	0.869 (0)
4	0.904 (0)	0.904 (0)
Euc.	0.716 (0)	
Mah.	0.904 (0)	

Bold figures denote the highest score(s) out of all methods used

The adjusted rand score takes values in $[-1, 1]$, where 1 indicates a perfect matching between L_T and L_P .

To find the purity score, we proceed as follows: let $T = \{t_1, t_2, \dots, t_m\}$ be the set of ‘true’ clusters in the data, and let $P = \{p_1, p_2, \dots, p_K\}$ be the set of predicted clusters. The purity score measures the extent to which a predicted cluster p_i only contains points from a single ‘true’ cluster t_j :

$$\mathcal{P}(T, P) = \frac{1}{N} \sum_{i=1}^K \max_j |p_i \cap t_j|,$$

where N is the total number of points. That is, for each predicted cluster p_i , we count the highest number of points from a single true cluster t_j predicted to be in p_i . These counts are summed and divided by the total number of observations. The purity score takes values in $[0, 1]$, with 1 being a perfect clustering.

Figure 7 gives the AR and purity scores for the K -Means clustering of each dataset using the varying distance measures being considered. For the k -minimal-variance and k -simplicial distances, we show the distances with values of k which produced the highest scores. Note that the pseudo-Mahalanobis distance is used in cases where the data is degenerate. The eigenvalues for each of the datasets in Table 10 can be found in Appendix B. The influence of these eigenvalues on the performance of the different distance measures is important, particularly when choosing values of k . When we discuss eigenvalues being ‘close to zero’, this is in relation to the largest eigenvalue of the dataset. There is no specific threshold for being ‘close to zero’, but the examples should give some intuition about choosing the parameter k .

Table 11, 12 and 13 give the median AR score for each dataset by each method of clustering, and the standard deviation of these scores, with bold values denoting the highest score(s) out of all methods used. Although the AR score and the purity score measure distinct aspects of the success of a clustering, Fig. 7 shows that the same patterns emerge for both evaluation methods, and so we only consider the AR score in the tables.

Iris is a 4-dimensional dataset, with no extreme small eigenvalues in comparison to its largest eigenvalue. Figure 7a, b and Table 11 show that the Mahalanobis distance performs best, joint with the k -simplicial and k -minimal-variance distances when $k = 4$ (recall that these distances are equal to the Mahalanobis distance when $k = d$). The Iris dataset is low-dimensional and full-rank, and hence the Mahalanobis distance can

Table 12 Median AR scores (and standard deviations) for each given dataset and distance

(a) Wine dataset		
<i>k</i>	<i>k</i> -simp	<i>k</i> -MV
2	0.714 (0.000)	0.817 (0.017)
3	0.759 (0.000)	0.917 (0.007)
4	0.759 (0.000)	0.915 (0.007)
5	0.818 (0.005)	0.915 (0.007)
6	0.833 (0.004)	0.915 (0.007)
7	0.899 (0.004)	0.915 (0.007)
8	0.899 (0.006)	0.915 (0.007)
9	0.917 (0.004)	0.915 (0.006)
10	0.915 (0.006)	0.913 (0.005)
11	0.915 (0.006)	0.869 (0.017)
12	0.915 (0.006)	0.899 (0.006)
13	0.899 (0.012)	0.854 (0.000)
Euc.	0.854 (0.01)	
Mah.	0.899 (0.01)	
(b) Image Segmentation dataset		
<i>k</i>	<i>k</i> -simp	<i>k</i> -MV
2	0.360 (0.063)	0.238 (0.084)
3	0.247 (0.074)	0.344 (0.063)
4	0.339 (0.070)	0.448 (0.027)
5	0.392 (0.047)	0.465 (0.027)
6	0.451 (0.051)	0.460 (0.026)
7	0.451 (0.022)	0.456 (0.023)
8	0.451 (0.044)	0.454 (0.021)
9	0.449 (0.047)	0.454 (0.028)
Euc.	0.464 (0.020)	
Mah.	0.285 (0.067)	

Bold figures denote the highest score(s) out of all methods used

use the true inverse of the covariance matrix. This example illustrates the performance gains that can be made in cluster analysis by taking the correlations in the data into consideration. Even when we use $k = 3$ in the k -simplicial and k -minimal-variance distances, we achieve better results than the Euclidean distance.

Figure 7c, d and Table 12a consider the Wine dataset, and show that the Mahalanobis, k -minimal-variance and k -simplicial distances outperforms the Euclidean distance, again highlighting the importance of accounting for correlation. The Wine dataset has some very small eigenvalues compared to its largest eigenvalue, and as such the Moore–Penrose pseudo-inverse is likely to have been adversely impacted [19]. Choosing the k -minimal-variance or k -simplicial distance avoids this impact, and as such produces better clustering results. This example also highlights that the

k -minimal-variance distance performs very well with lower values of k , whereas the k -simplicial distance requires a higher value of k to achieve its best results, as seen before in the efficiency evaluations in Sect. 5.1. This gives better computational-time for the best results when using the k -minimal-variance distance, but does make the distance more sensitive to a too-high choice of k , as seen by the decrease in AR scores in Table 12a.

The Image Segmentation dataset has a number of very large eigenvalues, some eigenvalues very close to zero, and five zero eigenvalues. Figure 7e and f shows that the Mahalanobis distance performs worse than the Euclidean distance, perhaps due to the effect of very small eigenvalues on the Moore–Penrose pseudo-inverse. The k -minimal-variance and k -simplicial distance outperform the Mahalanobis distance here, as they are less likely to be adversely affected by these small eigenvalues. Table 12b shows that the k -minimal-variance distance attains the highest AR score out of all the distances, but does not improve greatly on the Euclidean distance.

The Digits and Protein datasets (Table 13a, and b, respectively) both have a substantial number of small and zero eigenvalues (see Appendix B), indicating why our distances perform better than the pseudo-Mahalanobis distance. The Mahalanobis distance does not add much performance gain compared to the Euclidean distance in these examples, but the correct choice of k in the k -minimal-variance or k -simplicial distances provides improvement.

In these examples, we see that K -Means with the k -minimal-variance distance reaches its best adjusted rand score with relatively low k , whereas the k -simplicial distance needs higher values of k to reach this. However, the k -simplicial distance is less likely to breakdown for too-high a choice of k , as we see in Table 12a and 13a with the k -minimal-variance. For the k -simplicial distance, the values of k that produce the best adjusted rand score roughly match with the number of ‘larger’ eigenvalues in the datasets.

6 Conclusion

In this paper, we have continued the work done by Pronzato et al. [34] in researching the k -simplicial distance. This distance considers the covariance structure of the data, but is less adversely affected by the presence of small eigenvalues than the Mahalanobis distance.

We have studied the choice of the parameter k in detail, through the use of numerical examples, illustrating that too low a choice of k does not give much improvement on the Euclidean distance, and too high a choice of k doesn’t give much performance benefit over lower values, but does increase computational complexity. We recommend a choice of k influenced by the number of ‘large’ eigenvalues present, with respect to the other eigenvalues of the sample covariance matrix. We also discuss the benefits and limitations of different choices of the parameter δ : $\delta = 1$ is more robust, and well suited to applications such as outlier detection, but does not have a fast method of full computation. If $\delta = 2$, we have a significantly faster and easier method of producing the distance, but this is more likely to be influenced by the presence of outliers. We discuss the implementation of sub-sampling simplices, which greatly

Table 13 Median AR scores (and standard deviations) for each given dataset and distance

(a) Digits dataset		
<i>k</i>	<i>k</i> -simp	<i>k</i> -MV
2	0.596 (0.019)	0.709 (0.018)
3	0.620 (0.017)	0.707 (0.017)
4	0.642 (0.019)	0.701 (0.016)
5	0.657 (0.019)	0.696 (0.017)
6	0.663 (0.017)	0.693 (0.016)
7	0.677 (0.019)	0.695 (0.016)
8	0.686 (0.019)	0.691 (0.015)
9	0.690 (0.019)	0.689 (0.142)
10	0.693 (0.019)	0.686 (0.144)
11	0.695 (0.018)	0.689 (0.111)
12	0.696 (0.018)	0.679 (0.230)
13	0.697 (0.018)	0.664 (0.174)
14	0.696 (0.018)	0.605 (0.226)
15	0.696 (0.017)	0.673 (0.166)
16	0.695 (0.017)	0.679 (0.173)
17	0.694 (0.017)	0.658 (0.180)
18	0.692 (0.017)	0.664 (0.222)
Euc.	0.666 (0.012)	
Mah.	0.677 (0.014)	
(b) Protein dataset		
<i>k</i>	<i>k</i> -simp	<i>k</i> -MV
2	0.140 (0.057)	0.130 (0.057)
3	0.140 (0.055)	0.185 (0.020)
4	0.141 (0.050)	0.208 (0.021)
5	0.143 (0.044)	0.195 (0.015)
6	0.156 (0.024)	0.194 (0.019)
7	0.164 (0.024)	0.189 (0.014)
8	0.176 (0.026)	0.183 (0.017)
9	0.183 (0.026)	0.181 (0.027)
10	0.189 (0.028)	0.179 (0.032)
11	0.192 (0.026)	0.178 (0.041)
12	0.194 (0.021)	0.186 (0.058)
13	0.197 (0.020)	0.184 (0.022)
14	0.196 (0.019)	0.185 (0.008)
15	0.191 (0.021)	0.184 (0.026)
Euc.	0.172 (0.012)	
Mah.	0.172 (0.012)	

Bold figures denote the highest score(s) out of all methods used

improves computation time when we are using the distance with $\delta = 1$ with minimal changes to the results of the distance.

We have also introduced a new measure of distance, namely the k -minimal-variance distance. Again, this distance is less affected by the presence of small eigenvalues than the Euclidean and Mahalanobis distances (for appropriate choices of the parameter k) but is highly influenced by the choice of the parameter k which needs to be chosen carefully to ensure good performance of the distance.

Overall, we show that the k -minimal-variance distance is more efficient at minimizing the variance of the distances for lower choices of k (and therefore has better computation time) than the k -simplicial distance, but that the k -simplicial distance is less likely to be negatively affected by too-high a choice of k . We have given several examples where the k -minimal-variance and/or the k -simplicial distances outperform the multivariate distances commonly used. Namely, these proposed distances perform well where other classical distances often fail: when data is correlated, degenerate and has small eigenvalues, as well as possibly zero-valued eigenvalues.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A: Moments and Distributions of Distances

From Rencher and Schaalje [36], let y be a random vector with mean μ and covariance matrix W , and let A be a symmetric matrix of constants, then

$$E\left(y^{\top}Ay\right) = \text{trace}(AW) + \mu^{\top}A\mu.$$

If y is normally distributed with sample mean μ and sample covariance matrix W , then

$$\text{Var}\left(y^{\top}Ay\right) = 2\text{trace}\left((AW)^2\right) + 4\mu^{\top}AWA\mu.$$

Our distances considered in this paper are generalized squared distances of the form

$$\rho_A^2(x, X) = (x - \mu)^{\top}A(x - \mu),$$

where x is normally distributed with sample mean μ and sample covariance matrix W . Therefore we replace y with $(x - \mu)$, which is normally distributed with zero mean,

References

1. Aggarwal CC, Hinneburg A, Keim DA (2001) On the surprising behavior of distance metrics in high dimensional space. International conference on database theory. Springer, Berlin, pp 420–434
2. Agrawal R, et al. (1998) Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, pp 94–105
3. Bickel PJ et al (2008) Regularized estimation of large covariance matrices. *Ann Stat* 36(1):199–227
4. Blom G (1976) Some properties of incomplete U-statistics. *Biometrika* 63(3):573–580
5. Blum A, Hopcroft J, Kannan R (2016) Foundations of data science. Vorabversion eines Lehrbuchs 5:5
6. Bodnar T, Dette H, Parolya N (2016) Spectral analysis of the Moore-Penrose inverse of a large dimensional sample covariance matrix. *J Multivar Anal* 148:160–172
7. Cai T, Liu W, Luo X (2011) A constrained L1 minimization approach to sparse precision matrix estimation. *J Am Stat Assoc* 106(494):594–607
8. Chokniwal A, Singh M (2016) Faster Mahalanobis k-means clustering for Gaussian distributions. In: 2016 International Conference on Advanced Computing and Communication Information (ICACCI), pp 947–952
9. Clarke R et al (2008) The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nat Rev Cancer* 8(1):37–49
10. d'Aspremont A, Banerjee O, El Ghaoui L (2008) First-order methods for sparse covariance selection. *SIAM J Matrix Anal Appl* 30(1):56–66
11. Dua D, Graff C (2017) UCI machine learning repository. <http://archive.ics.uci.edu/ml>
12. Fan J, Liao Y, Liu H (2016) An overview of the estimation of large covariance and precision matrices. *Econ J* 19(1):C1–C32
13. Fisher TJ, Sun X (2011) Improved Stein-type shrinkage estimators for the high-dimensional multivariate normal covariance matrix. *Comput Stat Data Anal* 55(5):1909–1918
14. Friedman J, Hastie T, Tibshirani R (2008) Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9(3):432–441
15. Furrer R, Bengtsson T (2007) Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants. *J Multivar Anal* 98(2):227–255
16. Gnanadesikan R, Harvey JW, Kettenring JR (1993) Mahalanobis metrics for cluster analysis. *Sankhy Indian J Stat A* 55(3):494–505
17. Golub GH, Van Loan CF (2013) Matrix computations. Johns Hopkins University Press, Baltimore
18. Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143(1):29–36
19. Hoyle DC (2010) Accuracy of pseudo-inverse covariance learning—a random matrix theory analysis. *IEEE Trans Pattern Anal Mach Intell* 33(7):1470–1481
20. Hubert L, Arabie P (1985) Comparing partitions. *J Classif* 2(1):193–218
21. Kang X, Deng X (2020) An improved modified Cholesky decomposition approach for precision matrix estimation. *J Stat Comput Simul* 90(3):443–464
22. Lahav A, Talmon R, Kluger Y (2018) Mahalanobis distance informed by clustering. *Inf Inference J IMA* 8(2):377–406
23. Lam C (2020) High-dimensional covariance matrix estimation. *Wiley Interdiscip Rev: Comput Stat* 12(2):1485
24. Lancewicki T, Aladjem M (2014) Multi-target shrinkage estimation for covariance matrices. *IEEE Trans Signal Process* 62(24):6380–6390
25. Ledoit O, Wolf M (2004) A well-conditioned estimator for large-dimensional covariance matrices. *J Multivar Anal* 88(2):365–411
26. Lloyd S (1982) Least squares quantization in PCM. *IEEE Trans Inf Theory* 28(2):129–137
27. Mahalanobis PC (1936) On the generalised distance in statistics. In: Proceedings of the National Institute of Science India. pp 49–55
28. Manning CD, Schütze H, Raghavan P (2008) Introduction to information retrieval. Cambridge University Press, Cambridge
29. Meinshausen N et al (2006) High-dimensional graphs and variable selection with the lasso. *Ann Stat* 34(3):1436–1462
30. Melnykov I, Melnykov V (2014) On K-means algorithm with the use of Mahalanobis distances. *Stat Probab Lett* 84:88–95

31. Pedregosa F et al (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12:2825–2830
32. Perlibakas V (2004) Distance measures for PCA-based face recognition. *Pattern Recognit Lett* 25(6):711–724
33. Perthame E, Friguet C, Causeur D (2016) Stability of feature selection in classification issues for high-dimensional correlated data. *Stat Comput* 26(4):783–796
34. Pronzato L, Wynn H, Zhigljavsky A (2018) Simplicial variances, potentials and Mahalanobis distances. *J Multivar Anal*, pp 276–289
35. Prykhodko S, et al. (2018) Application of the squared Mahalanobis distance for detecting outliers in multivariate non-Gaussian data. In: 2018 14th International conference on advanced trends in radio-electronics, telecommunications and computer engineering (TCSET), pp 962–965
36. Rencher AC, Schaali GB (2008) *Linear models in statistics*, 2nd edn. Wiley-Interscience, New Jersey
37. Schäfer J, Strimmer K (2005) A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Stat Appl Genet Mol Biol* 4(1)
38. Smith MR, Martinez TR (2011) Improving classification accuracy by identifying and removing instances that should be misclassified. In: 2011 International joint conference on neural networks, IEEE, pp 2690–2697
39. Srivastava N, Rao S (2016) Learning-based text classifiers using the Mahalanobis distance for correlated datasets. *Int J Big Data Intell* 3:18–27
40. Stöckl S, Hanke M (2014) Financial applications of the Mahalanobis distance. *Appl Econ Finance* 1(2):78–84
41. Wei X, Huang G, Li Y (2007) Mahalanobis ellipsoidal learning machine for one class classification. In: 2007 International conference on machine learning and cybernetics, vol 6, pp 3528–3533
42. Wilks SS (1960) Multidimensional statistical scatter. *Contrib Probab Stat (Essays in Honor of Harold Hotelling, Olkin, Ingram et al)* pp 486–503
43. Won JH et al (2013) Condition-number-regularized covariance estimation. *J R Stat Soc B (Stat Methodol)* 75(3):427–450
44. Xiang S, Nie F, Zhang C (2008) Learning a Mahalanobis distance metric for data clustering and classification. *Pattern Recognit* 41(12):3600–3612
45. Zhang Y et al (2015) A low-rank and sparse matrix decomposition-based Mahalanobis distance method for hyperspectral anomaly detection. *IEEE Trans Geosci Remote Sens* 54(3):1376–1389
46. Zimek A, Schubert E, Kriegel HP (2012) A survey on unsupervised outlier detection in high-dimensional numerical data. *Stat Anal Data Min ASA Data Sci J* 5(5):363–387

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.