Computational Statistics
https://doi.org/10.1007/s00180-022-01277-6

ORIGINAL PAPER

Polynomial whitening for high-dimensional data

Jonathan Gillard[1] · Emily O'Riordan[1] · Anatoly Zhigljavsky[1]

Received: 19 August 2021 / Accepted: 17 August 2022

Abstract

The inverse square root of a covariance matrix is often desirable for performing data whitening in the process of applying many common multivariate data analysis methods. Direct calculation of the inverse square root is not available when the covariance matrix is either singular or nearly singular, as often occurs in high dimensions. We develop new methods, which we broadly call *polynomial whitening*, to construct a low-degree polynomial in the empirical covariance matrix which has similar properties to the true inverse square root of the covariance matrix (should it exist). Our method does not suffer in singular or near-singular settings, and is computationally tractable in high dimensions. We demonstrate that our construction of low-degree polynomials provides a good substitute for high-dimensional inverse square root covariance matrices, in both $d < N$ and $d \geq N$ cases. We offer examples on data whitening, outlier detection and principal component analysis to demonstrate the performance of the proposed method.

Keywords Whitening · Covariance · Mahalanobis · Scatter · Generalized inverse

1 Introduction

Let $X \in \mathbb{R}^{d \times N}$ be a matrix of data, with N observations in d dimensions. We denote the empirical d-dimensional mean vector and the empirical $d \times d$ covariance matrix of X by μ and Σ respectively, and make no other assumptions about the generation or structure of the data. In this paper we consider transformations of X of the form $X_A = A(X - \mu)$, where A is a $d \times d$ matrix, with the aim of whitening the data X. Data whitening is a transformation of the data intended

✉ Jonathan Gillard
gillardjw@cardiff.ac.uk

Emily O'Riordan
oriordane1@cardiff.ac.uk

Anatoly Zhigljavsky
zhigljavskyaa@cardiff.ac.uk

[1] School of Mathematics, Cardiff University, Cardiff, UK

Published online: 08 September 2022

to decorrelate and standardize the variables. Fully decorrelated data possesses a diagonal covariance matrix, and standardized data has unit variance for each variable (Hossain 2016). Applying a whitening transformation both decorrelates and standardizes the data, so that in the case of non-degenerate data, the covariance matrix of the whitened data X_A will be the identity matrix. By removing the simple elliptic structure of the data through such whitening transformations, we can uncover more interesting and complex structures in the data that may have previously been hidden in correlations, such as clusters or outliers (Li and Zhang 1998). Furthermore, the orthogonality of whitened variables can improve computational time and performance of many statistical methods (Koivunen and Kostinski 1999; Huang et al. 2020; Zuber and Strimmer 2009).

Examples of existing whitening methods include the so-called Mahalanobis whitening defined by

$$X_{\Sigma^{-1/2}} = \Sigma^{-1/2}(X - \mu),$$

which is popularly used to whiten data before performing many classical methods of multivariate analyses (Kessy et al. 2018). The transformed data $X_{\Sigma^{-1/2}}$ has zero-valued mean and the $d \times d$ identity matrix I_d as the covariance matrix. The success of the Mahalanobis whitening depends on the ability to compute $\Sigma^{-1/2}$ in a way that is both accurate and stable.

It is common for big, high-dimensional data to be close to degeneracy/low-rank (Udell and Townsend 2019) yielding unstable computations of $\Sigma^{-1/2}$, with numerous examples of this problem observed in: recommender systems data (Zhou et al. 2008; Li et al. 2016); finance (Bai and Shi 2011); medicine (Schuler et al. 2016); genomics (Wu et al. 2015); and social networks (Liben-Nowell and Kleinberg 2007). These issues also arise in generalized mixture models (Xiao 2020); multiple regression (Healy 1968; Hoang and Baraille 2012); adaptive algorithms (Baktash et al. 2017); and linear discriminant analysis (Ye and Xiong 2006). This is because variables often possess (approximate) linear dependencies, resulting in a covariance matrix Σ that is singular, or very close to singularity. As such, the inverse of the covariance matrix therefore does not exist or is at least unstable, and it becomes inadvisable or impossible to calculate $\Sigma^{-1/2}$. Consequently Mahalanobis whitening, and many other methods which directly rely on the inverse of the covariance matrix (such as those described in the survey of the recent paper (Kessy et al. 2018)), are not recommended.

Nevertheless, it has been demonstrated that applying Mahalanobis whitening prior to clustering or outlier detection (to give just two out of many possible examples) often results in better empirical results. This has been observed in several practical examples (Zafeiriou and Laskaris 2008; Shi et al. 2015). Theoretically, Mahalanobis whitening underpins weighted least squares (Seber and Lee 2012), PCA (Jolliffe 1986; Hyvärinen and Oja 2000), canonical correlation analysis (Härdle and Simar 2007) and most of the array of classic multivariate statistics methods (Li and Zhang 1998; Malsiner-Walli et al. 2016). Crucially, decorrelated and standardized data greatly simplifies both theoretical and practical multivariate data analysis (Agostinelli and Greco 2019; Anaya-Izquierdo et al.

2011; Martens et al. 2003; Thameri et al. 2011; Chen et al. 2015; Yang and Jin 2006).

The need to find stable ways of calculating (or approximating) Σ^{-1} in settings when X is (close to) degeneracy is evidenced in other application domains. In high-dimensional examples one often would like to use the Mahalanobis distance (Mahalanobis 1936) to measure the 'scatter' or 'spread' of the data (Pronzato et al. 2017, 2018), as a basis for proximity-dependent techniques such as clustering (Zuanetti et al. 2019) and Approximate Bayesian Computation (ABC) (Akeret et al. 2015). But again, for the reasons outlined earlier, the covariance matrix may be singular or ill-conditioned. ABC is used to find estimates of distribution parameters by simulating data over a parameter space (informed by some prior) and finding simulated data closest to the observed data. The ideal measure of closeness is to use the Mahalanobis distance, but the practical need to use ABC is often informed by degeneracy of the observed data, rendering the construction of a computationally tractable distance measure one of the fundamental problems of applying ABC (Wegmann et al. 2009; Beaumont 2019; Prangle 2017).

Recent literature has shown that whitening can also be used to improve the training of neural networks (Huang et al. 2018). Often, normalization is used in such training, rather than whitening, due to ill-conditioned problems (Luo 2017) and the great expense of computing a large inverse square root covariance matrix (Ioffe and Szegedy 2015), despite whitening being preferable if it is possible (Huang et al. 2020).

Naturally, many methods attempt to circumvent the aforementioned problems by use of the Moore-Penrose pseudo-inverse Σ^{-} to Σ, and then to take the square root of Σ^{-} if needed. However, in the case of high-dimensional data, it has been shown that the Moore-Penrose pseudo-inverse is not always suitable (Hoyle 2011; Bodnar et al. 2016; Bickel and Levina 2004), particularly when there are small eigenvalues. This is a problem appearing in several branches of mathematics, and work on this topic can be found in the statistics literature (which tend to use shrinkage-type estimators (Ledoit and Wolf 2004; Fisher and Sun 2011; Ito et al. 2015)and/or assume sparsity of the covariance matrix (Cai et al. 2011, 2016; Janková and van de Geer 2017)) and in the linear algebra/numerical analysis literature. In this latter work, algorithms (alternating projections (Higham and Strabić 2016) or Newton-type (Qi and Sun 2011; Higham 2008)) to compute 'substitute' covariance matrices are developed. Challenges in the successful use of these algorithms include computational tractability, stability, and the ability to find good starting-points. Our work differs from this in that we provide explicit formulae for the construction of our substitute to $\Sigma^{-1/2}$ and we are able to quickly generate a family of whitening matrices based on the order of our polynomials.

In this paper, we introduce polynomial whitening, and what we call the minimal-variance polynomial matrix, to be used in place of the square root of the inverse of the empirical covariance matrix. In view of the celebrated Cayley-Hamilton theorem (Cayley 1858; Hamilton 1853) the true inverse of a full-rank $d \times d$ matrix Σ can be calculated through a $d-1$ degree polynomial in Σ. In Gillard et al. (2022), it is shown that an alternative to the inverse of a matrix can be found using low degree polynomials. Our work follows on from this, as we consider polynomials of low degree in Σ to now provide an alternative to the square root of its inverse. Our method is applicable in cases where the true inverse square root of the covariance matrix does not exist, making it a viable alternative for degenerate and close-to degenerate datasets. Parameter options also allow for a trade-off between data whitening accuracy and time complexity.

The main practical focus of this paper is in data whitening, but in view of the discussion above, we envisage other settings where our work may be useful. The structure of the paper is as follows. Section 2 introduces the form of our matrix polynomial, and the optimization problem we solve in order to obtain an alternative to the inverse square root of the covariance matrix. The main theorem of this paper which is studied in later examples is given in Sect. 2.3. We address different parameter choices in Sect. 2.4, and in Sect. 2.6 we discuss using our procedure in conjunction with random projection methods, which can be useful when dealing with very high-dimensional data. Examples applying our method to data whitening, outlier detection and dimension reduction are given in Sect. 3, before we conclude the paper in Sect. 4.

2 The minimal-variance polynomial

2.1 Covariance matrix of transformed data

The mean vector $E(X_A)$ and covariance matrix $\mathscr{D}(X_A)$ of $X_A = A(X - \mu)$ are respectively:

$$E(X_A) = 0_d\,, \quad \mathscr{D}(X_A) = A\Sigma A^\top\,,$$

where 0_d is the d-dimensional vector of zeroes (Mathai and Provost 1992). Data transformed by Mahalanobis whitening, $X_{\Sigma^{-1/2}} = \Sigma^{-1/2}(X - \mu)$, has covariance matrix

$$\mathscr{D}\big(X_{\Sigma^{-1/2}}\big) = \Sigma^{-1/2}\Sigma\Sigma^{-1/2} = I_d\,,$$

where I_d is the $d \times d$ identity matrix. The total variation in X_A is given by $\mathrm{trace}\big(\mathscr{D}(X_A)\big) = \mathrm{trace}(A\Sigma A^\top)$, and for the Mahalanobis whitening the total variation in $X_{\Sigma^{-1/2}}$ is given by $\mathrm{trace}\big(\mathscr{D}\big(X_{\Sigma^{-1/2}}\big)\big) = \mathrm{trace}(I_d) = d$.

2.2 The minimal-variance polynomial alternative to $\Sigma^{-1/2}$

Let $\theta = \big(\theta_0, \theta_1, \dots, \theta_{k-1}\big)^\top$ and $\Sigma_{(k)} = \big(\Sigma^0, \Sigma^1, \dots, \Sigma^{k-1}\big)^\top$. We define A_k to be a $(k-1)$-degree matrix polynomial in Σ, of the form:

$$A_k = \sum_{i=0}^{k-1} \theta_i \Sigma^i = \theta^\top \Sigma_{(k)}\,. \tag{1}$$

For a chosen integer k such that $k - 1 < d$, our objective is to find the k coefficients of the matrix polynomial, denoted $\theta = \big(\theta_0, \theta_1, \dots, \theta_{k-1}\big)^\top$ in (1), so that the total variation of the transformed data $X_{A_k} = A_k(X - \mu)$, is minimized, subject to suitable constraints. For further intuition as to why we minimize the total variation, see Gillard et al. (2022).

Let $s_j = \text{trace}(\Sigma^j)$, $S_{(i,k)} = \left(s_i, s_{i+1}, \dots, s_{i+k-1}\right)$, and define the matrix

$$M_{(k)} = \begin{pmatrix} s_1 & s_2 & \cdots & s_k \\ s_2 & s_3 & \cdots & s_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_k & s_{k+1} & \cdots & s_{2k-1} \end{pmatrix}.$$

We seek to minimize the total variation of the transformed data $X_{A_k} = A_k(X - \mu)$, which is given by:

$$\begin{aligned} \text{trace}\left(\mathscr{D}(X_A)\right) &= \text{trace}\left(A_k \Sigma A_k^\top\right) \\ &= \text{trace}\left(\sum_{i=0}^{k-1} \theta_i \Sigma^i \Sigma \sum_{j=0}^{k-1} \theta_j \Sigma^j \right) \\ &= \theta^\top M_{(k)} \theta. \end{aligned}$$

To ensure non-trivial solutions to the minimization of the total variation, we introduce a constraint. There are a number of options for this constraint; here we consider constraints of the form

$$\text{trace}\left(A_k \Sigma^\alpha\right) = \text{trace}\left(\Sigma^{\alpha - 1/2}\right) \tag{2}$$

for a scalar value α. This can be written in the above notation as

$$\theta^\top S_{(\alpha,k)} = s_{\alpha-1/2}.$$

A constraint of this form ensures that the minimal-variance polynomial matrix A_k has similar qualities to $\Sigma^{-1/2}$, in the cases where this matrix exists. The constraint (2) will be revisited after the following theorem.

2.3 Constructing the minimal-variance polynomial

Theorem 1 *Let $X \in \mathbb{R}^{d \times n}$ be a d-dimensional dataset with n observations, having empirical mean μ and empirical covariance matrix Σ. For $k - 1 < d$, the matrix polynomial $A_k = \sum_{i=0}^{k-1} \theta_i \Sigma^i = \theta^\top \Sigma_{(k)}$ such that* $\text{trace}\left(\mathscr{D}(X_{A_k})\right)$ *is minimized, subject to the constraint $\theta^\top S_{(\alpha,k)} = s_{\alpha-1/2}$, has coefficients given by*

$$\hat{\theta} = \frac{s_{\alpha-1/2}}{S_{(\alpha,k)}^\top M_{(k)}^{-1} S_{(\alpha,k)}} M_{(k)}^{-1} S_{(\alpha,k)}\,. \tag{3}$$

Proof We will minimize $\frac{1}{2}\text{trace}(\mathscr{D}(X_{A_k}))$ subject to the constraint (2), where the constant $1/2$ is introduced to simplify calculations. The Lagrange function $\mathscr{L}(\theta, \omega)$ with Lagrange multiplier ω is given by

$$\mathscr{L}(\theta, \omega) = \frac{1}{2}\theta^\top M_{(k)}\theta - \omega(\theta^\top S_{(\alpha,k)} - s_{\alpha-1/2}).$$

We minimize the Lagrange function by differentiating with respect to θ and setting the result equal to 0, which gives:

$$M_{(k)}\theta = \omega S_{(\alpha,k)}$$

and we can therefore rearrange to find θ:

$$\hat{\theta} = \omega M_{(k)}^{-1} S_{(\alpha,k)}. \tag{4}$$

Let $\omega = \omega_k$. We can find the value of ω_k by substituting (4) into the constraint $\theta^\top S_{(\alpha,k)} = s_{\alpha-1/2}$ giving

$$\omega_k = \frac{s_{\alpha-1/2}}{S_{(\alpha,k)}^\top M_{(k)}^{-1} S_{(\alpha,k)}}.$$

Thus, the vector of coefficients of the polynomial (1) which minimizes $\text{trace}\big(\mathscr{D}(X_{A_k})\big)$ subject to the constraint (2) is given by (3). We call this polynomial the minimal-variance polynomial. □

When evaluating the polynomial, we recommend forming the matrix powers by iteratively multiplying by Σ, or using Horner's method for polynomial evaluation. Both of these methods are outlined in Sect. 4.2 of Higham (2008).

2.4 How to choose parameter values in the minimal-variance polynomial

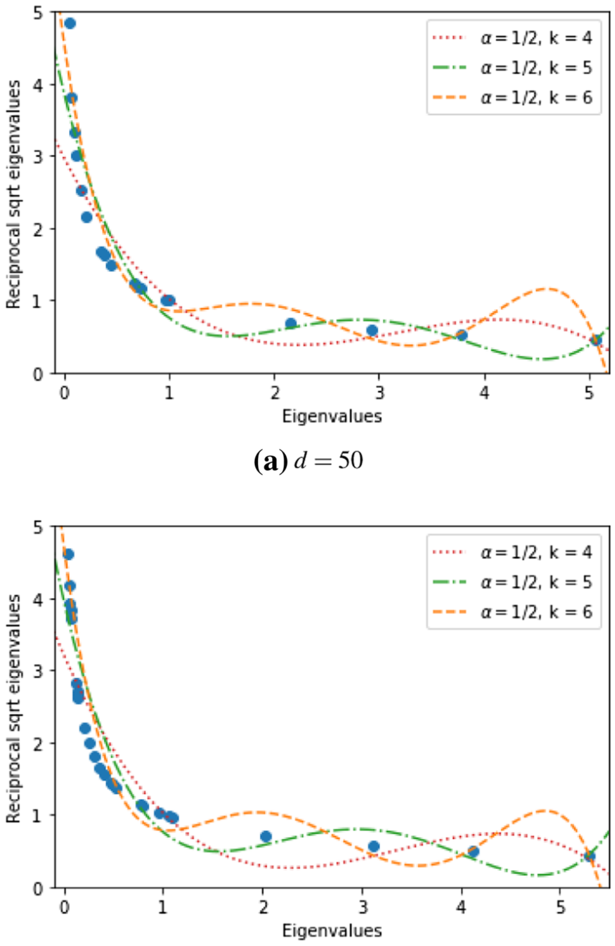
2.4.1 Choice of the parameter α in the constraint (2)

We studied the outcomes of polynomial whitening with different values of α in the constraint (2). Theoretically, any value of α will produce an alternative whitening matrix. Empirical investigations showed that the polynomial with $\alpha = 1/2$ performed particularly well, in terms of data whitening success, stability and computational cost. Using this value of α is equivalent to applying the constraint

$$\text{trace}\big(A_k\Sigma^{1/2}\big) = \text{trace}\big(I_d\big) = d. \tag{5}$$

Our analysis found that, when using this constraint, data was approximately whitened using a relatively low value of k (when compared to the value of the dimension d of the dataset).

Figure 1a considers a 50-dimensional dataset, with 5 eigenvalues greater than 1, 30 eigenvalues between 0 and 1, and 15 zero eigenvalues. Figure 1b considers a



(a) $d = 50$

(b) $d = 150$

Fig. 1 The minimal-variance polynomial fit to simulated eigenvalues (blue, values given in Appendix 1.1) for datasets with **a** 50 dimensions and **b** 150 dimensions. Parameters used are $\alpha = 1/2$ and $k = 4$ (red, dotted line), $k = 5$ (green, dash-dot line) and $k = 6$ (orange dashed line)

150-dimensional dataset, with 5 eigenvalues greater than 1, 100 eigenvalues between 0 and 1, and 45 zero eigenvalues. The eigenvalues of these datasets are given in Appendix 1.1. These eigenvalues have been chosen to create a degenerate example which the Moore-Penrose pseudo-inverse would struggle to deal with well. We did so by setting roughly $d/3$ eigenvalues equal to zero, and letting the nonzero eigenvalues taper towards zero, making the rank of the dataset unclear. We plot in blue dots the nonzero eigenvalues of the dataset on the horizontal axis, and the reciprocal square root of the nonzero eigenvalues on the vertical axis. We then plot the corresponding minimal-variance polynomial with $\alpha = 1/2$ in degree k as follows. Find the coefficients $\theta = (\theta_0, \theta_1, \dots, \theta_{k-1})^\top$ of the minimal-variance polynomial using (3), and write the polynomial as in (1), replacing the matrix Σ with a symbol t:

$$p(t) = \theta_0 t^0 + \theta_1 t^1 + \dots \theta_{k-1} t^{k-1}.$$

We can then plot this polynomial $p(t)$ for different values of t. In Fig. 1, we consider polynomials of degree 3, 4 and 5, (recalling that using the value k results in a $(k-1)$-degree polynomial). The degree of these polynomials are much lower than the dimensionality of the datasets, yet still provide a good approximate fit to the inverse square root of the given eigenvalues.

The constraint (5) with $\alpha = 1/2$ works well in the case of non-degenerate data (when Σ is essentially non-singular), but requires some simple tuning for degenerate or nearly-degenerate data, which has been applied here. This tuning will be discussed in Sect. 2.5.

2.4.2 Choice of the parameter k to determine the degree of the minimal-variance polynomial

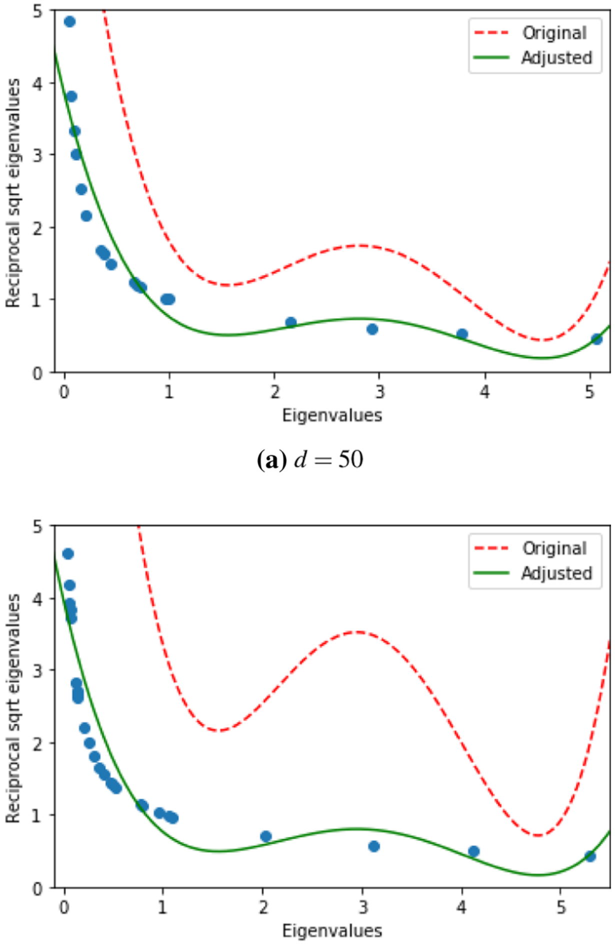
The true inverse square root of a full-rank covariance matrix can be written as a $(d-1)$-degree polynomial using the characteristic polynomial. The minimal-variance polynomial with parameter k forms a $(k-1)$-degree polynomial, as defined in (1). As k increases, the polynomial can approximate the square root of the characteristic polynomial more accurately, but is more costly to compute. Therefore, the choice of the parameter k is often a trade-off between accuracy and cost.

However, as k increases, so does the opportunity for instability in the polynomial, particularly when working in high dimensions (see Table 3 for an example of this, and Sect. 3.1 for more of a discussion on this topic). As such, keeping k relatively low is not only beneficial for cost, but for stability. Furthermore, choosing low values of k results in a good approximation for the inverse square root of the covariance matrix, as can be seen by the polynomial fit to the eigenvalues in Fig. 1. This will be further demonstrated in the numerical examples in Sect. 3.

In this paper, we often run the same experiments multiple times with different values of k, and use a problem-specific metric to identify the best value of k for that dataset. For example, in Sect. 3.1, we use the Wasserstein metric to compare the whitened data to the standard normal distribution, as well as a sum-of-squares-based metric. We then choose the value of k which produces the lowest values for these metrics. This is similar to methods used in many parameterized methods, such as using scree plots or silhouette scores to judge the best number of clusters to use in a clustering algorithm. It may often be best to apply minimal-variance whitening for multiple values of k to the dataset, and then inspect the empirical covariance matrix of the transformed data to see which value of k has performed best.

2.5 Constraint adjustment for rank-deficient data

We provide an adaptation to the method when using $\alpha = 1/2$ to make it suitable for use when Σ is singular, as hinted at in discussions about the choice of α. When using the constraint (5) with $\alpha = 1/2$, the polynomial aims to ensure that the trace of $A_k \Sigma^{1/2}$ is equal to d. We propose that this trace should aim to equal r, the rank of the covariance matrix, in a similar way to the Moore-Penrose pseudo-inverse Σ^{-} having the property that $\operatorname{trace}\big((\Sigma^{-})^{1/2}\Sigma^{1/2}\big) = r$. However, for matrices with many small eigenvalues, r is hard to calculate (Vidal and Favaro 2014), and often approximations of r are based on arbitrary eigenvalue thresholding or subjective elbow plots (Kishore Kumar and Schneider 2017).



(a) $d = 50$

(b) $d = 150$

Fig. 2 The minimal-variance polynomial with $k = 5$ fit to eigenvalues (blue, the same as in Fig. 1) before (red, dashed line) and after (green, solid line) adjustment for rank-deficiency, as described in Sect. 2.5

In cases where Σ is not full-rank, we propose an adjustment to modify our constraint (2) without the need to calculate r directly. We will illustrate our adjustment using the two examples in Fig. 2, which plot the nonzero eigenvalues and nonzero reciprocal square root eigenvalues in the same way as in Fig. 1. The datasets used in Fig. 2 are the same as those in Fig. 1, details of which are given in Sect. 2.4 and Appendix 1.1.

We first calculate the minimal-variance polynomial using the constraint (5). This is shown in Figs. 2a and b as the red, dashed line. Although the polynomials take the correct shape, they are clearly placed too high and do not fit the plot of the inverse square root eigenvalues. We adjust the polynomial by multiplying by some constant c between 0 and 1, which ensures a better fit of the polynomial. Our method of choosing a value of c is as follows.

Let $\Lambda = \{\lambda_1, \dots, \lambda_d\}$ be the set of all eigenvalues of Σ, and let $\tilde{\Lambda} = \{\lambda_i \in \Lambda : \lambda_i \neq 0\}$ be the set of all nonzero eigenvalues of Σ. In the case of very large dimensions d, computation of eigenvalues λ_i is certainly out of reach; in this case, as will be discussed in Sect. 2.6, we suggest to project the data to a low-dimensional space and use the set of eigenvalues for the low-dimensional version of the data. The constant c can be found in any number of ways which minimizes the distance between the polynomial $p(\lambda)$ and the target values $1/\lambda^{1/2}$, for $\lambda \in \tilde{\Lambda}$. We use the value c^* from

$$c^* = \underset{c\in(0,1]}{\arg\min} \sum_{\lambda\in\tilde{\Lambda}} w(\lambda)[c \cdot p(\lambda) - \lambda^{-0.5}]^2$$

where $w(\lambda)$ is a suitable weight function. That is, we seek to minimize the weighted sum of squares between the polynomial and the reciprocal square root of the nonzero eigenvalues. The optimal value of the adjustment constant c is then found to be

$$c^* = \frac{\sum_{\lambda\in\tilde{\Lambda}} w(\lambda)\lambda^{-0.5}p(\lambda)}{\sum_{\lambda\in\tilde{\Lambda}} w(\lambda)p(\lambda)^2}.$$

In Fig. 2 (and all other examples in this paper), we have used $w(\lambda) = \lambda$, and in general we recommend this. However, the choice of w can be altered to give a different fit for the data given. If the user is more concerned about fitting the polynomial to the larger eigenvalues, they may decide to use $w(\lambda) = \lambda^i$ with $i > 1$, for example.

The adjusted polynomials (given by the green solid line) clearly fit the desired points much more successfully than the original polynomials. However, if this adjustment is not performed, the data transformed by the polynomial whitening matrix A_k will still be approximately isotropic, so the adjustment is not necessary if equal variance is of variables is sufficient. This adjustment has been applied to all examples that follow in this paper.

This adjustment to the constraint can also be used to detect the singularity of a matrix. Let us first consider the case with $d < N$. If Σ is full rank, and k is chosen appropriately, the value c^* will be equal to (or very close to) 1, as the minimal-variance polynomial is aiming to make $\mathrm{trace}\big(A_k\Sigma\big) = d$, which is correct in the case of full-rank Σ. If the matrix Σ is not full-rank, c^* will be less than 1. To illustrate this, Table 1 gives two $d < N$ examples. A d-dimensional

Table 1 The adjustment value c^* for different configurations of the dimension d, number of observations N, rank of true population covariance matrix R and rank of sample covariance matrix r

Dataset	d	N	R	r	c^*
1	100	1000	100	100	1.00
2	100	1000	50	50	0.50
3	100	50	100	50	0.50
4	100	50	50	50	0.50
5	100	50	30	30	0.30

dataset with N observations is generated using a covariance matrix generated with rank R. Further details on how these datasets were generated is given in Appendix 1.1. The empirical covariance matrix of the dataset has rank $r = \min(d, N, R)$, and is used to find the minimal-variance polynomial matrix with $k = 10$, and the constraint adjustment c^* is given. In dataset 1, the empirical matrix has full rank $r = d$, so $c^* = 1$. In dataset 2, the 'true' covariance matrix has rank $R = 50$, $d = 100$ and $N = 1000$, therefore the empirical covariance matrix has rank $r = \min(d, N, R) = 50$. This produces a constraint adjustment value of $c^* = 0.50 < 1$, so we know the empirical covariance matrix Σ is singular.

We now consider cases with $d \geq N$ through the use of three examples. Dataset 3 in Table 1 has 100 dimensions and only 50 observations. The 'true' covariance matrix used to generate this dataset is full-rank $R = 100$, but the empirical covariance matrix has rank $r = \min(d, N, R) = 50$. Therefore, using the empirical covariance matrix in the minimal-variance polynomial matrix gives adjustment value $c^* = 0.50$, informing us that this dataset is degenerate. Dataset 4 also has $d = 100$, $N = 50$, and the 'true' covariance matrix now has rank $R = 50$. The adjustment value is therefore less than 1: $c^* = 0.50$. The final example we consider again has dimension $d = 100$ and number of observations $N = 50$, but the 'true' covariance matrix has rank $R = 30$. The empirical covariance matrix therefore has rank $r = \min(d, N, R) = 30$, and the adjustment value is $c^* = 0.30$. In all these examples, $c^* < 1$, as the empirical covariance matrix Σ will never be full-rank in $d < N$ examples.

2.6 Applications to extremely high-dimensional data

Given a dataset X with extremely high-dimension d, say $d = 1{,}000{,}000$, finding the minimal-variance polynomial matrix can be too costly and time-intensive. We can instead sample some variables from X to produce a 'representative' dataset $\tilde{X}$ in a much smaller dimension $\tilde{d}$. This representative dataset can be found through random samples of the variables in X, or projection to a lower dimensional space (see Bingham and Mannila (2001), Blum et al. (2014)). We can proceed with calculating the covariance matrix $\tilde{\Sigma}$ of $\tilde{X}$, and use $\tilde{\Sigma}$ to produce the minimal-variance polynomial alternative to $\tilde{\Sigma}^{-1/2}$:

$$\tilde{A_k} = \theta_0 I + \theta_1 \tilde{\Sigma} + \cdots + \theta_{k-1} \tilde{\Sigma}^{k-1} . \qquad (6)$$

We can then replace the $\tilde{d}$-dimensional matrix $\tilde{\Sigma}$ in (6) with the d-dimensional covariance matrix Σ to obtain the minimal-variance polynomial matrix A_k. This can be used to whiten the original large dataset X, and is much cheaper than finding the minimal-variance polynomial matrix directly.

For large datasets, it may be that we don't know the eigenvalues exactly, but can approximate the distribution of the eigenvalues. If this is the case, we can sample $\tilde{d}$ eigenvalues from this distribution using the inverse cumulative distribution function. We will illustrate this using the Marchenko-Pastur distribution, as this distribution is known to model the eigenvalues of the sample covariance matrix of a random matrix as $d, N \to \infty$. Figure 3 considers an example with $d = 10,000$ and $N = 15,000$, and the probability density function (PDF) of the Marchenko-Pastur distribution with these parameters is shown by the red line. The histogram represents a random sample of 300 eigenvalues, and shows such a sample models the distribution well.

Similar methods can be used in the case where $d \geq N$. We can reduce d to a value smaller than N by sampling the real eigenvalues, if they are known or can be calculated. Alternatively, we can sample the eigenvalues from an assumed distribution as described above.

3 Numerical examples

3.1 Whitening data using minimal-variance polynomials

3.1.1 Data with $d < N$

We begin the numerical examples by whitening several synthetic and real datasets using the minimal-variance polynomial. The details of these datasets are given in Table 2. The four synthetic datasets (D1, D2, D3, D4) are sampled from a

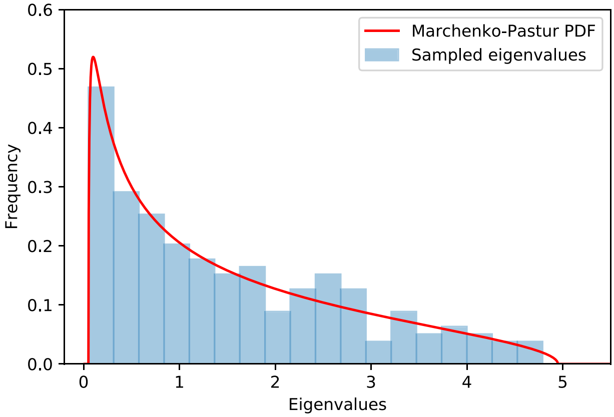


Fig. 3 Sampling of eigenvalues from the Marchenko-Pastur distribution. The red line indicates the Marchenko-Pastur PDF, when $d = 10,000$ and $N = 15,000$. The histogram shows the spread of the 300 sampled values from this distribution

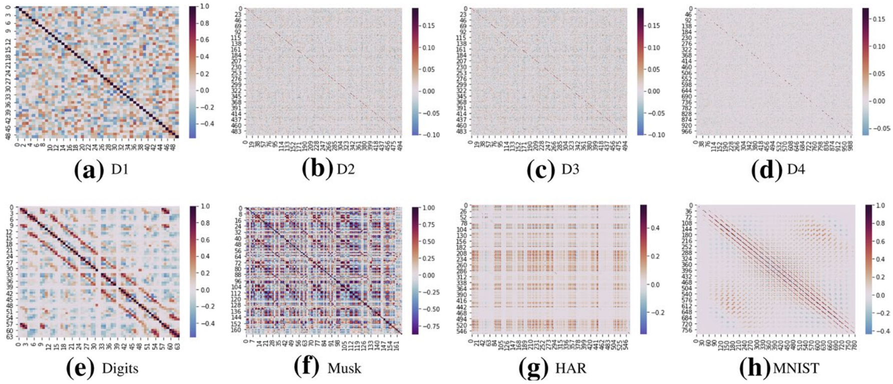
Table 2 Datasets used in Sect. 3.1.1, their dimension d and number of observations N

Dataset	d	N
D1	50	250
D2	100	500
D3	500	2500
D4	1000	5000
Digits	64	1797
Musk	168	6598
HAR	561	10299
MNIST	784	70000

Gaussian distribution $\mathcal{N}_d(0, \Sigma)$ with $N = 5 \times d$ observations, where the covariance matrices Σ are produced as follows. Generate d eigenvalues $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_d\}$ from the Wishart distribution and produce a random $d \times d$ orthogonal matrix Q. Let L be the matrix with the eigenvalues Λ on the diagonal and zeroes elsewhere, then let $\Sigma = Q^\top LQ$.

The real datasets 'Digits', 'Musk' and 'HAR' (Human Activity Recognition) (Anguita et al. 2013) were obtained from the UCI Machine Learning repository (Dua and Graff 2017). The 'MNIST' dataset (LeCun et al. 2010) was obtained from the OpenML database (Vanschoren et al. 2013).

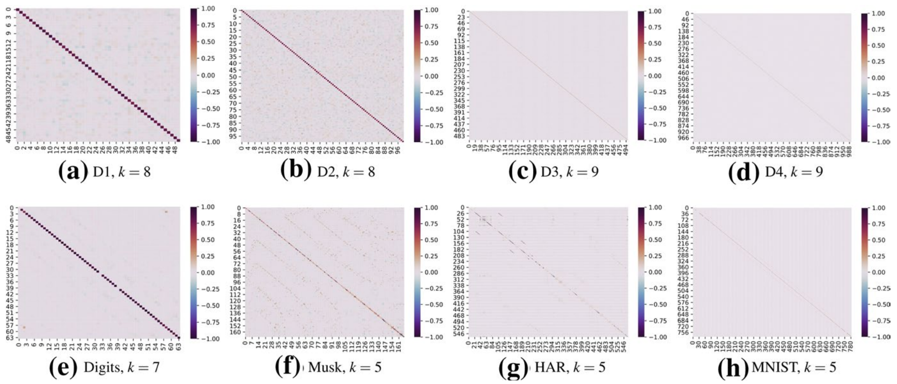
In some cases, it can be beneficial to rescale the data so that each variable has zero mean and unit variance, before finding the minimal-variance polynomial matrix. If rescaling the data provides less extreme eigenvalues in the covariance matrix, this scaling is likely to improve the performance of the polynomial whitening. The heatmaps in Fig. 4 show the covariance matrices of the datasets, and the distribution of the eigenvalues of these covariance matrices are given in



(a) D1 **(b)** D2 **(c)** D3 **(d)** D4

(e) Digits **(f)** Musk **(g)** HAR **(h)** MNIST

Fig. 4 Heatmaps of the covariance matrix of each dataset detailed in Table 2 before minimal-variance polynomial whitening. Datasets corresponding to Figures (a), (e), (f) and (h) are scaled to have unit variance, to improve performance of polynomial whitening. These heatmaps show the covariance matrix after this scaling



(a) D1, $k = 8$ **(b)** D2, $k = 8$ **(c)** D3, $k = 9$ **(d)** D4, $k = 9$

(e) Digits, $k = 7$ **(f)** Musk, $k = 5$ **(g)** HAR, $k = 5$ **(h)** MNIST, $k = 5$

Fig. 5 Heatmaps of the covariance matrix of the datasets in Table 2 after minimal-variance polynomial whitening. The value of k used in constructing the minimal-variance polynomial is given in the caption for each dataset

Table 3 The Wasserstein scores (7), denoted W_{A_kX}, which measure the distance between the polynomial-whitened dataset A_kX and the standard normal distribution $\mathcal{N}(0, I)$ for each dataset

Dataset	W_X	W_{A_3X}	W_{A_4X}	W_{A_5X}	W_{A_6X}	W_{A_7X}	W_{A_8X}	W_{A_9X}	$W_{A_{10}X}$
D1	0.455	0.179	0.119	0.090	0.075	0.060	**0.057**	0.071	0.139
D2	0.634	0.358	0.301	0.246	0.220	0.181	**0.160**	0.200	0.225
D3	0.866	0.718	0.678	0.631	0.601	0.578	0.544	**0.520**	0.552
D4	0.812	0.585	0.524	0.465	0.425	0.393	0.360	**0.336**	0.365
Digits	0.361	0.137	0.101	0.073	0.066	**0.058**	0.071	0.107	0.381
Musk	0.949	0.574	0.450	**0.373**	1.123	2.022	0.989	0.990	0.991
HAR	0.885	0.772	0.794	**0.586**	3.892	0.998	0.998	0.998	0.998
MNIST	0.612	0.405	0.341	**0.296**	0.597	1.077	1.039	1.566	4.563

Values in bold indicate the lowest Wasserstein score W_{A_kX} over all k for a given dataset

Appendix 1.2. Figure 4 shows a lot of nonzero off-diagonal values in the heatmaps, indicating that these datasets are highly correlated.

We can measure the proximity of the transformed data $X_{A_k} \sim \mathcal{N}_d(0, \mathcal{S})$ to the standard normal distribution $\mathcal{N}_d(0, I)$ using the Wasserstein metric (Givens and Shortt 1984):

$$W(X_{A_k}) = \left(d + \mathrm{trace}(\mathcal{S}) - 2\mathrm{trace}\left(\mathcal{S}^{1/2}\right)\right)/d, \tag{7}$$

where we divide by d here to account for the difference in the dimensions of each dataset.

The heatmaps in Fig. 5 show the covariance matrix of the transformed data $X_{A_k} = A_kX$ for each dataset, illustrating that the correlations between variables have been approximately whitened. The value of k used in these heatmaps is chosen as the value of k which gives the lowest Wasserstein score, as given in Table 3. The

Wasserstein scores in Table 3 show that, in general, as the value of k increases, the transformed data is closer to the standard normal distribution, as desired. In some cases, such as the Musk dataset, higher values of k begin to show an increase in the Wasserstein score, indicating the whitening transformation is less successful than when using lower values of k. This is likely due to numerical instability, as the minimal-variance polynomial aims to fit itself to extremely small eigenvalues, causing erratic behaviour in the polynomial. As such, it is recommended to use lower values of k which provide a more reliable alternative to the inverse square root of the covariance matrix, or to compute several minimal-variance polynomial matrices for different k and use the one that best satisfies some metric, such as the Wasserstein score.

As indicated by the Wasserstein scores in Table 3 and the heatmaps in Fig. 5, we produce an effective alternative to the inverse square root of the covariance matrix using a polynomial of degree significantly lower than the dimension of the dataset.

The Wasserstein measure concerns itself with the diagonal values of the covariance matrix, as it is calculated using traces. We can consider it as a measure of standardization, rather than whitening. We therefore need to measure the extent to which the data has been decorrelated. The heatmaps in Fig. 5 show that the off diagonals of the covariance matrix of the transformed data are close to zero, indicating good decorrelation. Another way we can measure this is by considering the sum of squares of the off-diagonal entries of the covariance matrix of the transformed data. In Table 4, let SS_{A_kX} be the sum of squares of the off-diagonal entries of the covariance matrix of the whitened dataset A_kX.

The sum of squares values in Table 4 decrease as k increases, until a certain value of k, much like the Wasserstein scores. Given we would like this value to be as small as possible, we see the value of k that gives the optimum sum of squares value for each dataset is close to value of k that gives the optimum Wasserstein score for each dataset. Therefore, when the data has been successfully standardized, it has also been decorrelated well.

Table 13 in Appendix 3 shows the average time taken to produce the minimal-variance polynomial matrices for each dataset for each value of k considered, over

Table 4 The sum of squares, denoted SS_{A_kX}, of the off diagonal values of the covariance matrix of the polynomial-whitened dataset A_kX for each dataset

Dataset	SS_X	SS_{A_3X}	SS_{A_4X}	SS_{A_5X}	SS_{A_6X}	SS_{A_7X}	SS_{A_8X}	SS_{A_9X}	$SS_{A_{10}X}$
D1	10.104	2.891	2.801	2.442	2.120	2.095	**1.738**	1.846	2.971
D2	10.879	6.955	5.869	5.830	5.123	5.088	5.052	**4.723**	5.784
D3	20.042	19.686	18.547	18.444	16.141	15.758	15.652	**15.391**	16.441
D4	31.882	21.596	21.378	21.087	20.887	20.235	20.459	19.971	**19.243**
Digits	11.095	2.636	2.118	1.961	1.469	**1.178**	1.665	1.877	3.999
Musk	58.266	5.029	6.965	**6.635**	31.327	127.561	0.640	0.510	0.411
HAR	33.836	3.095	**1.349**	1.438	20.434	1.386	1.458	1.533	1.613
MNIST	74.745	11.023	11.016	**10.614**	13.614	58.666	38.451	280.832	1661.624

Values in bold indicate the lowest value of SS_{A_kX} over all k for a given dataset

100 runs. The time taken increases as the dimensionality d of the dataset increases, and as the parameter k increases. However, the procedure to calculate the matrices only takes a matter of seconds, even for 1000-dimensional datasets. This time performance could be improved much further by implementing parallel computing methods.

3.1.2 Data with $d \geq N$

It is increasingly common for data to have higher dimensionality than number of observations in many fields, such as genetic microarrays, medical imaging and chemometrics (Hall et al. 2005). Such data is clearly rank-deficient, with $r \leq N < d$, and thus the sample covariance matrix of such data is always singular, rendering many multivariate data analysis methods unusable, including data whitening. Minimal-variance polynomial whitening is applicable in such cases, as illustrated by the following examples.

We consider four synthetically generated datasets and four real datasets, detailed in Table 5. The first two synthetic datasets, E1 and E2, are sampled from a Gaussian distribution $\mathscr{N}_d(0, \Sigma)$, where the covariance matrices Σ are produced as follows. Like the $d < N$ case, we generate d eigenvalues, and produce a random $d \times d$ orthogonal matrix Q. Let L be the matrix with the eigenvalues Λ on the diagonal and zeroes elsewhere, then let $\Sigma = Q^\top LQ$. The third synthetic dataset, E3, is generated to copy the example in Wang and Fan (2017): a multivariate Gaussian with population covariance matrix with diagonal entries $[50, 20, 10] + [1] * 47$. This creates a spiked eigenvalue model, which is of interest in HDLSS datasets (Aoshima and Yata 2018). The fourth dataset uses a covariance matrix with eigenvalues generated from a random uniform distribution between 0 and 1, to produce a non-sparse set of eigenvalues. The madelon dataset was obtained from the UCI Machine Learning Repository (Dua and Graff 2017). The raw madelon dataset has 4400 observations, greater than the 500 features, so we sampled only the first 250 observations to create the madelon† dataset with $d > N$. The yeast dataset is a real genomic dataset with 2284 features and 17 observations (Tavazoie et al. 1999; Vanschoren et al. 2013). The third real dataset is a genomic dataset on colon cancer data (Alon et al. 1999), used by (Yata and Aoshima 2013) as an example of a spiked

Table 5 Datasets used in Sect. 3.1, their dimension d and number of observations N

Dataset	d	N
E1	500	50
E2	1000	50
E3	500	50
E4	1000	500
Madelon†	500	250
Yeast	2884	17
Colon	2000	40
DB-emails	242	64

The madelon† dataset is a subsample of the true madelon dataset, with only the first 250 observations considered eigenvalue model. This dataset includes two clusters which represent tumorous and non-tumorous colons; we only consider the former cluster here. The DB-emails dataset is a 'bag-of-words' representation of a collection of emails (Filannino 2011). Note that the madelon† , yeast and colon datasets have been scaled to have unit variance. The empirical eigenvalues of all datasets are given in Appendix 1.2.

Successful whitening of these datasets would result in a covariance matrix with r eigenvalues equal to 1, and $d - r$ eigenvalues equal to 0. We performed Moore-Penrose whitening on the four datasets in Table 5 by pre-multiplying the data by the Moore-Penrose inverse of the square root of the covariance matrix. We also performed minimal-variance polynomial whitening on the datasets as described in Sect. 2.

Figure 6 compares the distribution of the eigenvalues of the covariance matrices after Moore-Penrose whitening and minimal-variance polynomial whitening. The eigenvalues are scaled such that the maximum eigenvalue is equal to 1. The first three synthetic datasets show that using minimal-variance whitening returns a dataset with eigenvalues only equal to 0 and 1, whereas using Moore-Penrose whitening gives a dataset with a spread of eigenvalues between 0 and 1. Figure 6d shows that minimal-variance whitening may not achieve perfect whitening, but that it is still more successful than the Moore-Penrose whitening method.

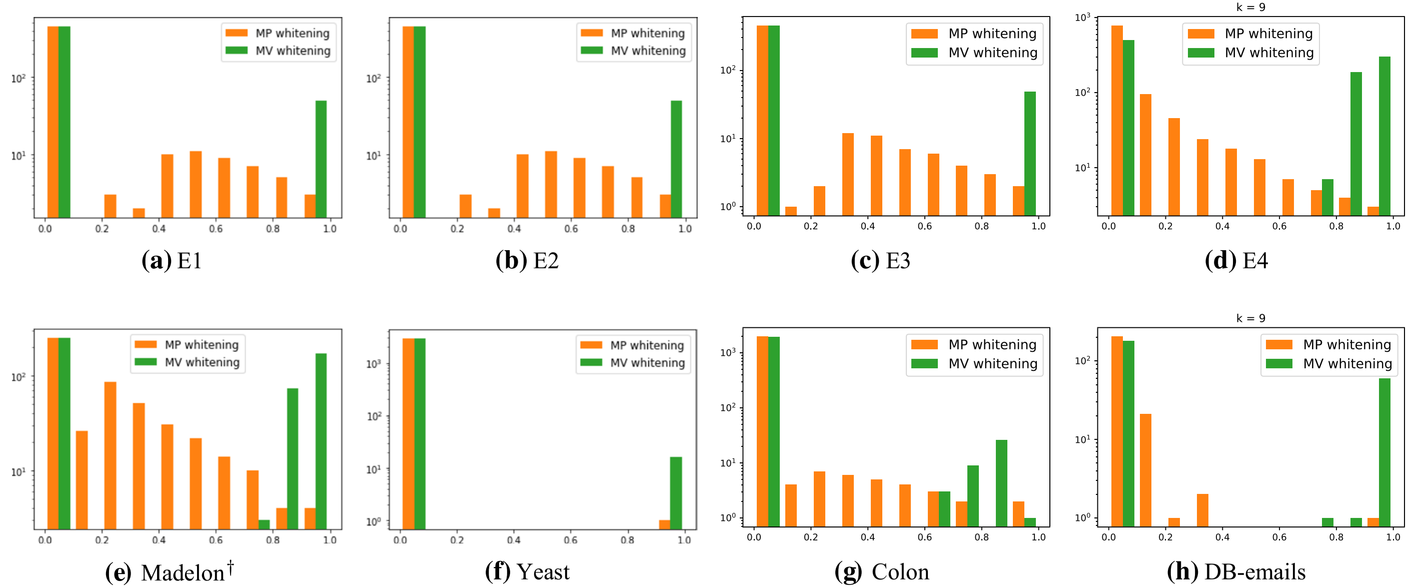
Figure 6f considers the Yeast dataset, and shows that both Moore-Penrose whitening and minimal-variance whitening return only eigenvalues of value 0 or 1. However, the Moore-Penrose whitening gives one eigenvalue equal to 1, and the rest 0 (or very close to 0). When using the minimal-variance whitening, the dataset has rank equal to the original dataset ($r = 16$ in this case). The madelon† , colon and DB-emails datasets are not whitened perfectly by either method, but the eigenvalues are much more dispersed when using Moore-Penrose whitening compared to minimal-variance polynomial whitening, whereas we seek eigenvalues only valued at 0 and 1, ideally.

3.2 Comparison to other whitening methods

Due to rotational freedom, there are infinitely many whitening matrices of the form $W = Q\Sigma^{-1/2}$, where Q is orthogonal and satisfies $Q^\top Q = I_d$ (Kessy et al. 2018).

Let us define some decompositions of the covariance matrix Σ, beginning with $\Sigma = V^{1/2}PV^{1/2}$, where V is the diagonal variance matrix and P is the correlation matrix. Let $\Sigma = U\Delta U^\top$ be the eigendecomposition of the covariance matrix, with U the matrix of eigenvectors and Δ the diagonal matrix of eigenvalues. Analogously, define the eigendecomposition $P = GOG^\top$ of the correlation matrix. We also define the Cholesky decomposition of the inverse covariance matrix $LL^\top = \Sigma^{-1}$, when Σ^{-1} exists.

Five whitening procedures are identified by Kessy et al. (2018) to be unique in fulfilling a given objective function. Most of these objective functions used in the paper are based on the cross-covariance matrix $\boldsymbol{\Phi}$ and the cross-correlation matrix $\boldsymbol{\Psi}$ between the original data X with covariance Σ and the whitened data X_W:



(a) E1 **(b)** E2 **(c)** E3 **(d)** E4

(e) Madelon† **(f)** Yeast **(g)** Colon **(h)** DB-emails

Fig. 6 Log-scale histograms, showing the eigenvalues of the covariance matrix after the data has been whitened by Moore-Penrose (MP) whitening (orange histogram), and minimal-variance polynomial (MV) whitening with $k = 9$ (green histogram), for each of the four datasets given in Table 5

$$\Phi = \mathrm{cov}(X_W, X) = W\Sigma,$$
$$\Psi = \mathrm{corr}(X_W, X) = \Phi V^{-1/2}.$$

In the following example, we will compare polynomial whitening to the three procedures from this paper that share our goal of whitening the data while changing as little else as possible. We do not consider the other methods in this paper, as these methods aim to maximize compression of variance into the first few variables of the whitened data. Although polynomial whitening performed relatively well in these scenarios, this is not the aim of our method. The three types of whitening we will consider alongside polynomial whitening are given below.

Mahalanobis whitening (MW) $\boldsymbol{W} = \boldsymbol{\Sigma}^{-1/2}$. Mahalanobis whitening is found to be the unique whitening procedure which maximizes $\mathbf{trace}(\boldsymbol{\Phi})$, the average cross-covariance between each variable of the original and the newly transformed data. This is equivalent to minimizing the total squared distance between the original data X and the whitened data X_W, ensuring the whitened data is as similar as possible to the original data.

Mahalanobis-cor whitening (MCW) $\boldsymbol{W} = \boldsymbol{P}^{-1/2}\boldsymbol{V}^{-1/2}$. Mahalanobis whitening can be affected by the differences in the scales of variables. To avoid this issue we may use a scale-invariant version, known as Mahalanobis-correlation whitening. The Mahalanobis-correlation whitening method maximizes the cross-correlation $\mathbf{trace}(\boldsymbol{\Psi})$ between each variable of the standardized original data $V^{-1/2}X$ and the whitened data X_W. Doing this is shown to be equivalent to minimizing the squared distance between $V^{-1/2}X$ and X_W.

Cholesky whitening (CW) $W = L^{\top}$. Cholesky whitening is the only whitening procedure fulfilling the constraint of producing lower-triangular cross-covariance and cross-correlation matrices with positive diagonal entries. It does not result from fulfilling an objective function like the above methods, but rather from satisfying this constraint.

We evaluate the performance of these different whitening procedures by applying them to a dataset and considering the different objective functions in $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$. First, as in Kessy et al. (2018), we apply the whitening methods to the 4-dimensional Iris dataset (Fisher et al. 1936) in Table 6. Given the dataset's low dimension and well-conditioned covariance matrix, polynomial whitening (**PW** in the table) with $k = d = 4$ produces exactly the same results as Mahalanobis whitening. We also perform polynomial-cor whitening (**PCW**), where the data is standardized and polynomial whitening is performed using the correlation matrix P. This produces the same results as Mahalanobis-cor whitening.

Table 6 A comparison of different whitening methods applied to the Iris dataset, using metrics identified by Kessy et al. (2018)

	MW	MCW	CW	PW $k = 4$	PCW $k = 4$
$\mathrm{tr}(\hat{\phi})$	**2.9829**	2.8495	1.9369	**2.9829**	2.8495
$\mathrm{tr}(\hat{\psi})$	3.0742	**3.1914**	2.5331	3.0742	**3.1914**

Bold entries identify the best result for each metric

The polynomial whitening method is more effectively used when applied to higher dimensional datasets with singular or near-singular covariance matrices. As such, we repeat the above exercise with a different dataset. For the purposes of this example, we are unable to use a dataset which has a singular covariance matrix, as the Mahalanobis and Cholesky whitening methods are not usable in this case. We use the Wisconsin Breast Cancer dataset (Wolberg et al. 1992), which we have pre-standardized to give improved results from all methods. This dataset has dimension $d = 32$ and has a covariance matrix which could be considered ill-conditioned (see Appendix 1.4 for details on the eigenvalues). Table 7 shows that polynomial whitening outperforms Mahalanobis whitening, using both the covariance and correlation matrix.

3.3 The effect of different pre-processing methods on outlier detection algorithms

Outlier detection algorithms often require that data is pre-processed before the algorithm can be applied. It has been shown by Campos et al. (2016) that the normalization of datasets will often lead to a better performance of outlier detection algorithms.

Here we replicate a study described in Kandanaarachchi et al. (2020). The authors produced a collection of labelled benchmark datasets to be used for evaluating outlier detection algorithm performance. They evaluated the performance of various algorithms when used after applying different normalization methods to these datasets. Performance of an algorithm was measured using the area under the Receiver Operator Characteristic (ROC) curve, which compares the labels of an observation (‘inlier’ or ‘outlier’) produced by the algorithm to the ‘true’ labels. They found that two types of normalization method performed differently (dependent on data set and outlier detection method):

‘Min–Max’ normalization Each variable v of a dataset is normalized to only have values in the range [0, 1]:

$$\frac{v - \min(v)}{\max(v) - \min(v)},$$

where $\min(v)$ and $\max(v)$ are the minimum and maximum values of the variable v, respectively.

‘Median-IQR’ normalization Each variable v is transformed to

Table 7 A comparison of different whitening methods applied to the Wisconsin Breast Cancer dataset, using metrics identified by Kessy et al. (2018)

	MW	MCW	CW	PW $k = 6$	PCW $k = 6$
$\mathrm{tr}(\hat{\phi})$	21.0193	21.1282	14.5409	**24.8036**	23.1984
$\mathrm{tr}(\hat{\psi})$	20.9651	21.0737	14.5034	21.7396	**24.7396**

Bold entries identify the best result for each metric

$$\frac{v - \text{median}(v)}{\text{IQR}(v)},$$

where median(v) and IQR(v) are the median and inter-quartile range of the variable v, respectively.

We consider the following four different outlier detection methods from the Python package PyOD (Zhao et al. 2019):

1. KNN: K-Nearest Neighbours
2. LOF: Local Outlier Factor
3. COF: Connectivity-based Outlier Factor
4. FastABOD: Fast Angle Based Outlier Detection

Further details of each of these algorithms are provided in Campos et al. (2016). All of the above methods require a parameter choice K (different to the polynomial degree parameter k referred to throughout this paper) to set the so-called neighbourhood size, and a contamination value C to indicate how many observations the algorithm should label as outliers. We let $K = 0.1 \times N$, where N is the number of observations in the dataset. The parameter C is equal to the percentage of outliers given by the 'true' labels.

For a dataset D, an outlier detection method o and a pre-processing method z, we denote the area under the ROC curve as $AUC(D, o, z)$. For each outlier detection method o listed above, we say a dataset D 'prefers' a pre-processing method z if $AUC(D, o, z) \geq AUC(D, o, y)$ for all other pre-processing methods y. We evaluate the AUC score for transformations $A_k D$ using (3) by taking the maximum AUC score over all k considered.

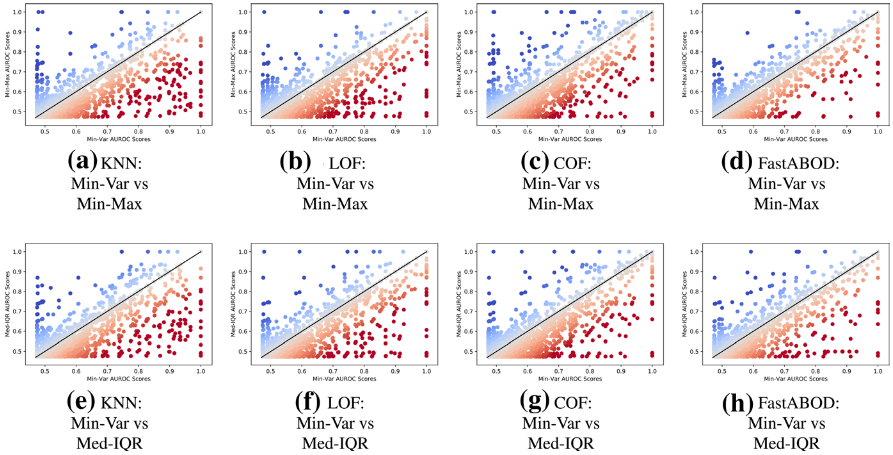
We tested the outlier detection methods with each pre-processing method on 7667 real datasets, as used in Kandanaarachchi et al. (2020). The datasets ranged from dimension 3 to dimension 359, and the number of observations in a dataset ranged from 44 to 5396. We impose no structural assumptions on the datasets for our method or the other normalization methods.

Table 8 shows the percentage of datasets that prefer each pre-processing method for each of the given outlier detection algorithms. The results in this table indicate that the polynomial whitening method outperforms the two normalization methods.

The scatter graphs in Fig. 7 compare the minimal-variance polynomial whitening to the normalization methods considered individually. Each point represents a

Table 8 The percentage of datasets that give higher AUC scores for the pre-processing technique (given in the column), by outlier detection method (given in the row)

Outlier Detection Method	Min–Var	Min–Max	Median-IQR
KNN	40.12%	30.70%	29.17%
LOF	41.29%	30.09%	28.61%
COF	42.26%	29.39%	28.34%
FastABOD	39.17%	31.16%	29.67%



(a) KNN: Min-Var vs Min-Max

(b) LOF: Min-Var vs Min-Max

(c) COF: Min-Var vs Min-Max

(d) FastABOD: Min-Var vs Min-Max

(e) KNN: Min-Var vs Med-IQR

(f) LOF: Min-Var vs Med-IQR

(g) COF: Min-Var vs Med-IQR

(h) FastABOD: Min-Var vs Med-IQR

Fig. 7 Scatter graphs plotting the AUC scores of outlier detection algorithms when performed using the minimal-variance polynomial whitening 'Min–Var' on the horizontal axis, and the AUC scores when using **a**–**d** 'Min–Max' or **e**–**h** 'Med-IQR' normalizations on the vertical axis. Points in red indicate a dataset where using Min–Var produced a better score than the alternative method, and points in blue indicate a dataset where using the alternative method produced a better score

dataset, and the diagonal line indicates those datasets where the two methods give equal AUC scores. Points below this line, in red, indicate that the minimal-variance whitening method outperformed the other method considered. A numerical breakdown of these scatter graphs is given in Table 9. Much like Table 8, Table 9 shows the percentage of datasets that prefer each pre-processing method, but shows a pairwise comparison.

Table 10 shows the amount of datasets out of the total 7667 (and the percentage) for which the pre-processing methods produce *strictly* better results, for each outlier detection method. It is clear that the minimal-variance method performs as well as (and often better than) the techniques often used to preprocess datasets before applying common outlier detection methods.

Table 9 The percentage of datasets for which the given pre-processing method (given in the column) produces AUC scores better than the alternative method in the adjacent column, for different outlier detection methods (given in the row). I.e. 34.4% of datasets produced higher AUC scores when using Min–Var than when using Min–Max, for the KNN outlier detection method. This differs from Table 8 in that it is a pairwise comparison of the pre-processing methods

	Min–Var vs Min–Max			Min–Var vs Med-IQR		
	Min–Var	Min–Max	Equal	Min–Var	Med-IQR	Equal
KNN	34.4%	15.9%	49.7%	35.8%	14.0%	50.1%
LOF	37.3%	14.1%	48.6%	38.1%	12.8%	49.2%
COF	40.6%	16.1%	43.3%	41.8%	14.8%	43.4%
FastABOD	32.2%	15.1%	52.7%	33.8%	12.8%	53.4%

Table 10 The number (and percentage) of datasets for which the given pre-processing method (in the column) produces AUC scores *strictly* better than the other methods, for each outlier detection method (in the row)

	Min–Var	Min–Max	Med-IQR
KNN	2195, 29%	742, 10%	632, 8%
LOF	2338, 30%	772, 10%	604, 8%
COF	2460, 32%	811, 11%	689, 9%
FastABOD	1950, 25%	705, 9%	519, 7%

3.4 Principal component analysis

Principal component analysis (PCA) is a popular dimension-reduction technique, as it reduces a dataset to a chosen dimension p while retaining the greatest amount of variance from the original dataset as possible. PCA finds p linear combinations of the variables of the dataset, giving p new compressed variables with maximal variance. As such, it is highly sensitive to the variances of the variables in the dataset. If one variable is measured on a much larger scale than the others, this variable will be likely have much greater variance, and therefore be given much more weight in a linear combination than the other variables (Jolliffe and Cadima 2016). To prevent this, it is good practice to standardize the variables to ensure they are all measured on the same scale.

We compare two methods of standardization prior to performing PCA: (Moore-Penrose) Mahalanobis standardization, which is most commonly used before PCA, and minimal-variance standardization. In Mahalanobis standardization, let the variable $v_i \in X$ have mean μ_i and standard deviation σ_i. We then consider the dataset made up of the variables

$$z_i = \frac{(v_i - \mu_i)}{\sigma_i}$$

for $i \in \{1, \dots, d\}$. If $\sigma_i = 0$, we use Moore-Penrose Mahalanobis (MPM) standardization, in which we find the square root of the Moore-Penrose inverse of the covariance matrix Σ^-, and then use $(\Sigma^-)_{i,i}$ (i.e. the ith diagonal entry of Σ^-) in place of σ_i.

In minimal-variance (MV) standardization, we find the minimal-variance polynomial matrix A_k, and use the values on the diagonal of A_k, denoted $(A_k)_{i,i}$, in place of σ_i:

$$w_i = \frac{(v_i - \mu_i)}{(A_k)_{i,i}}.$$

Note that this is different to minimal-variance whitening, in that we only use the diagonal of the minimal-variance polynomial matrix to perform the transformation. We do this to align our method with the Mahalanobis standardization method.

Our method of comparing the different standardization methods for PCA is as follows. In the following sections, we consider 1000 generated datasets, each with K clusters. For each dataset, we consider three versions: let X be the original dataset, X_{MPM} be the MPM standardized dataset and X_{MV} be the MV standardized dataset. For each version, we find the data given by the PCA transformation for a given number of principal components, and then perform the K-means clustering algorithm (Lloyd 1982). This is repeated for 1000 different datasets, and the results are given in the following sections for different types of data.

3.4.1 Data with $d < N$

We first consider the impact of the different standardization methods on PCA for data with $d < N$. For each of the 1000 datasets, we generate 3 clusters from multivariate Gaussian distributions $X^{(i)}$, $i = \{1, 2, 3\}$ with dimension $d = 100$, where the parameters $\mu^{(i)}$, $\Sigma^{(i)}$ and $N^{(i)}$ denote the mean, covariance matrix and number of observations in cluster $X^{(i)}$. The details of these parameters are given in Table 11. The eigenvalues of each $\Sigma^{(i)}$ taper off towards zero gradually. This creates a degenerate dataset with a rank that is hard to identify, a situation which the Moore-Penrose inverse struggles to deal with well.

In this example, we make parameter choices based on the relative size of the eigenvalues of a dataset compared to the maximum eigenvalue. Let $\Lambda = \{\lambda_1, \dots, \lambda_d\}$ be the set of eigenvalues of a dataset, let $\lambda_{\max}$ be the largest eigenvalue in Λ, and let $\bar{\lambda}$ be the mean of the eigenvalues in Λ.

Let $p = p(\Lambda)$ be the number of principal components that we wish to reduce a dataset to using PCA. For each dataset, the parameter $p(\Lambda)$ is chosen to be the number of eigenvalues in Λ greater than the mean eigenvalue $\bar{\lambda}$:

$$p(\Lambda) = \sum_{i=1}^{d} \mathbb{1}_{\lambda_i > \bar{\lambda}},$$

as commonly used in practice (Abdi and Williams 2010).

The parameter $k = k(\Lambda)$ for the minimal-variance polynomial will chosen based on the number of scaled eigenvalues $\pi_i = \lambda_i / \lambda_{\max}$ that are bigger than a given threshold t:

Table 11 Details of clusters of datasets used for PCA and K -means examples in Sect. 3.4.1

i	$\mu^{(i)}$	Eigenvalues of $\Sigma^{(i)}$	$N^{(i)}$
1	$[0, \dots, 0]$	$[100, 50, 0.9^1, 0.9^2, 0.9^3, \dots]$	166
2	$[1, \dots, 1]$	$[100, 50, 0.8^1, 0.8^2, 0.8^3, \dots]$	166
3	$[0] * 33 + [1] * 64$	$[100, 50, 0.8^1, 0.8^2, 0.8^3 \dots]$	168

All datasets have dimension $d = 100$

$$k(\Lambda) = \sum_{i=1}^{d} \mathbb{1}_{\pi_i > t}.$$

In the examples that follow we use $t = 0.1$.

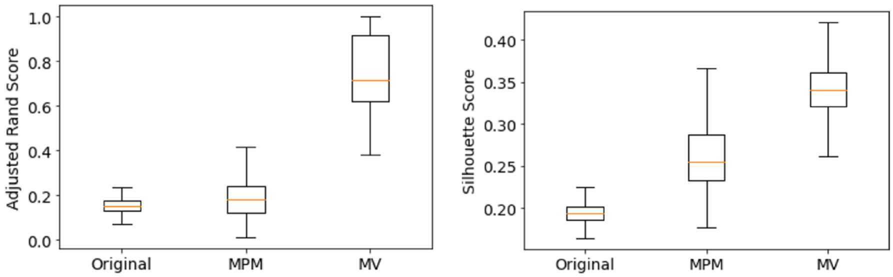
The K-means clustering algorithm aims to assign each point within a dataset to a cluster, by estimating the distances from each point to the estimated centre-point of a cluster of points. For more information on the algorithm, see Jain (2010). We consider how well the K-means clustering algorithm performs after applying PCA, given the different standardization methods. We use the adjusted rand (AR) score (Hubert and Arabie 1985; Steinley 2004) of the cluster labels provided by K-means to judge how well the algorithm has found the correct clusterings. An AR score of 0 indicates random labellings, and an AR score of 1 means the clusters were perfectly labelled by the algorithm.

We also give the silhouette scores (Rousseeuw 1987) of the methods depending on the standardization methods. The silhouette score of a clustering indicates how well separated the clusters are. A score of 1 indicates well-distinguished clusters, whereas a score of -1 tells us that clusters have been incorrectly assigned. A higher silhouette score tells us that the standardization method and PCA have retained cluster structure well.

Figure 8 shows that using the MPM standardization gives a slight improvement on using no standardization. However, using minimal-variance standardization before applying PCA and K-means clustering results in vastly better AR scores, as well as better silhouette scores.

3.4.2 Data with $d \geq N$

We modify the above example slightly to help us consider the case where $d \geq N$. In such circumstances, PCA can sometimes perform poorly due to difficulties in finding the eigenvectors of the covariance matrix correctly (Aoshima et al. 2018). As



(a) Adjusted Rand Score

(b) Silhouette Score

Fig. 8 **a** Adjusted Rand scores and **b** Silhouette scores of the labellings made by the K-means algorithm after PCA, which was applied to 1000 datasets with: no standardization (Original); Moore-Penrose Mahalanobis (MPM) standardization; minimal-variance (MV) standardization such, we now compare the standardization methods when applied before 3 different methods of dimension reduction:

1. Classical PCA (Pearson 1901)
2. Cross-Data PCA (CD-PCA) (Yata and Aoshima 2010)
3. Noise-Reduction PCA (NR-PCA) (Yata and Aoshima 2012)

The latter two methods were formulated for performing dimension reduction on HDLSS data, and can avoid the difficulties sometimes faced by PCA in such settings. Examples given in these papers show promising results for eigenvalue estimation and dimension reduction in high dimensions. For more details on the implementation of these methods, see the papers referenced above.

As in Sect. 3.4.1, we consider 1000 different datasets, each with $d = 1000$ and $N = 430$. Each dataset is generated as a mixture of four multivariate Gaussian distributions $X_i \sim \mathcal{N}_d\left(\mu_i, \Sigma_i\right)$, $i = \{1, 2, 3, 4\}$. The population parameters of each cluster are given in Table 12.

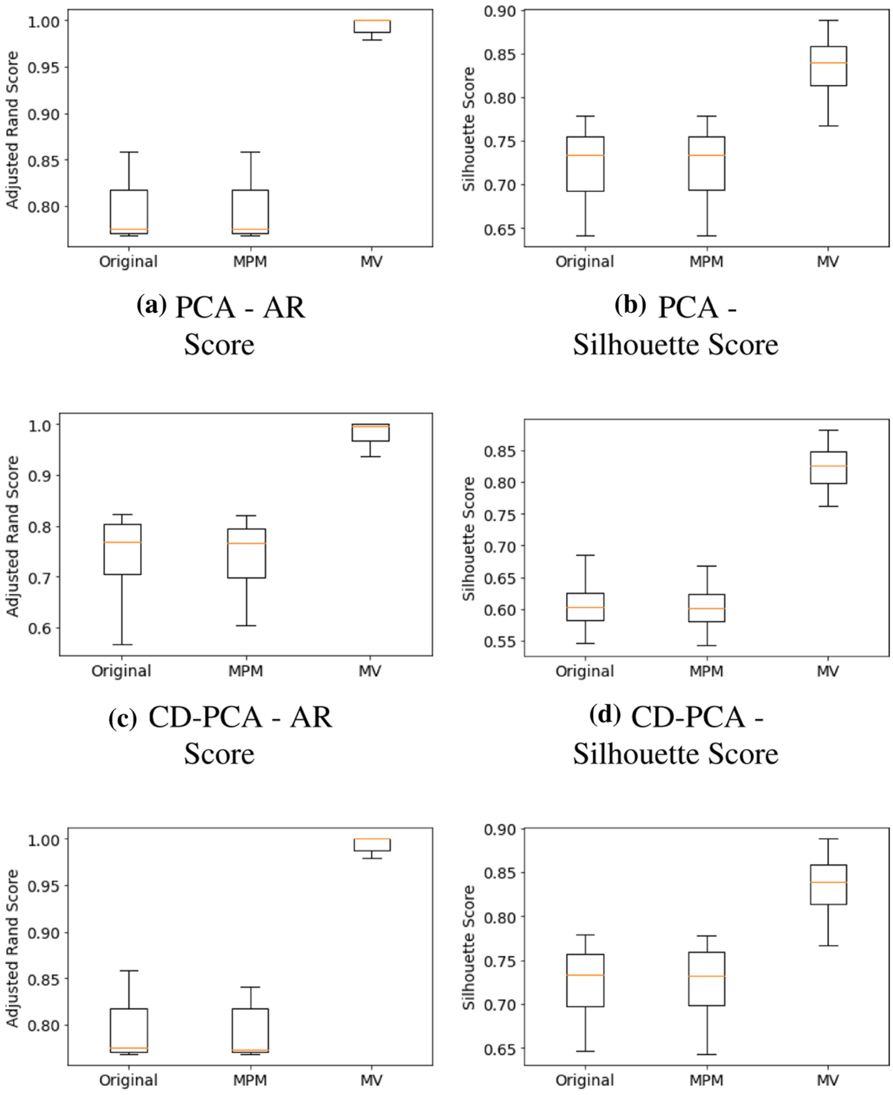
Figure 9 shows boxplots of the AR scores and silhouette scores of the labels given by K-means clustering, after applying one of the standardization methods and one of the three dimension reduction methods. Across all types of PCA, we see that MPM standardization gives very similar results to the datasets with no standardization. On the other hand, the MV standardization method provides a large improvement for all three methods of dimension reduction. The clustering results are better when MV standardization has been used, as indicated by the boxplots of AR scores in Fig. 9a, c and e. We see that cluster separation is also better when using MV standardization with dimension reduction, as the silhouette scores are much higher for all three dimension reduction methods.

Across the three dimension reduction techniques considered, the minimal-variance standardization method is clearly very useful in those cases where standardization would improve dimension reduction algorithms (or other multivariate data analysis methods), as it behaves similarly to the Moore-Penrose Mahalanobis standardization method, but does not struggle in cases where the rank is unclear and there are many small eigenvalues.

Table 12 Details of clusters of datasets used for PCA and K-means examples in Sect. 3.4.2

i	$\mu^{(i)}$	Eigenvalues of $\Sigma^{(i)}$	$N^{(i)}$
1	$[0, \dots, 0]$	$[100, 50, 0.9^1, 0.9^2, 0.9^3, \dots]$	133
2	$[1, \dots, 1]$	$[100, 50, 0.8^1, 0.8^2, 0.8^3, \dots]$	133
3	$[0] * 333 + [1] * 667$	$[100, 50, 0.8^1, 0.8^2, 0.8^3 \dots]$	134
4	$[1, \dots, 1]$	$[100, 50, 0.1^1, 0.1^2, 0.1^3 \dots]$	30

The datasets have $d = 1000$ and $N = 430$



(a) PCA - AR Score

(b) PCA - Silhouette Score

(c) CD-PCA - AR Score

(d) CD-PCA - Silhouette Score

(e) NR-PCA - AR Score

(f) NR-PCA - Silhouette Score

Fig. 9 Adjusted Rand scores and Silhouette scores of the labellings made by the K-means algorithm after PCA, CD-PCA and NR-PCA which was applied to 1000 datasets with: no standardization (Original); Moore-Penrose Mahalanobis (MPM) standardization; minimal-variance (MV) standardization

Fig. 10 Eigenvalues of the datasets used in Sect. 3.1.1. A dataset with a * in its caption has been rescaled such that each variable has zero mean and unit variance, and hence the eigenvalues of the correlation matrix are given

▶

4 Conclusion

We have developed a method of constructing polynomials in the empirical covariance matrix to provide an alternative to the inverse square root of a covariance matrix, particularly suitable to degenerate (or close to degenerate) data in high dimensions. The minimal-variance polynomial whitening method aims at minimizing the total variation in a transformed dataset, and in doing so it provides a dataset that has been decorrelated and standardized. We have demonstrated the potential applications of these polynomial matrices by considering whitening, outlier detection and principal component analysis for both $d < N$ and $d \geq N$ cases. We have discussed and given recommendations for the choice of the parameter k which dictates the degree of the polynomial, as well as an alternative constraint and adjustments. We also suggested a method to reduce computational time in extremely high-dimensional cases, ensuring that this method can be applied in such scenarios.

Appendix 1

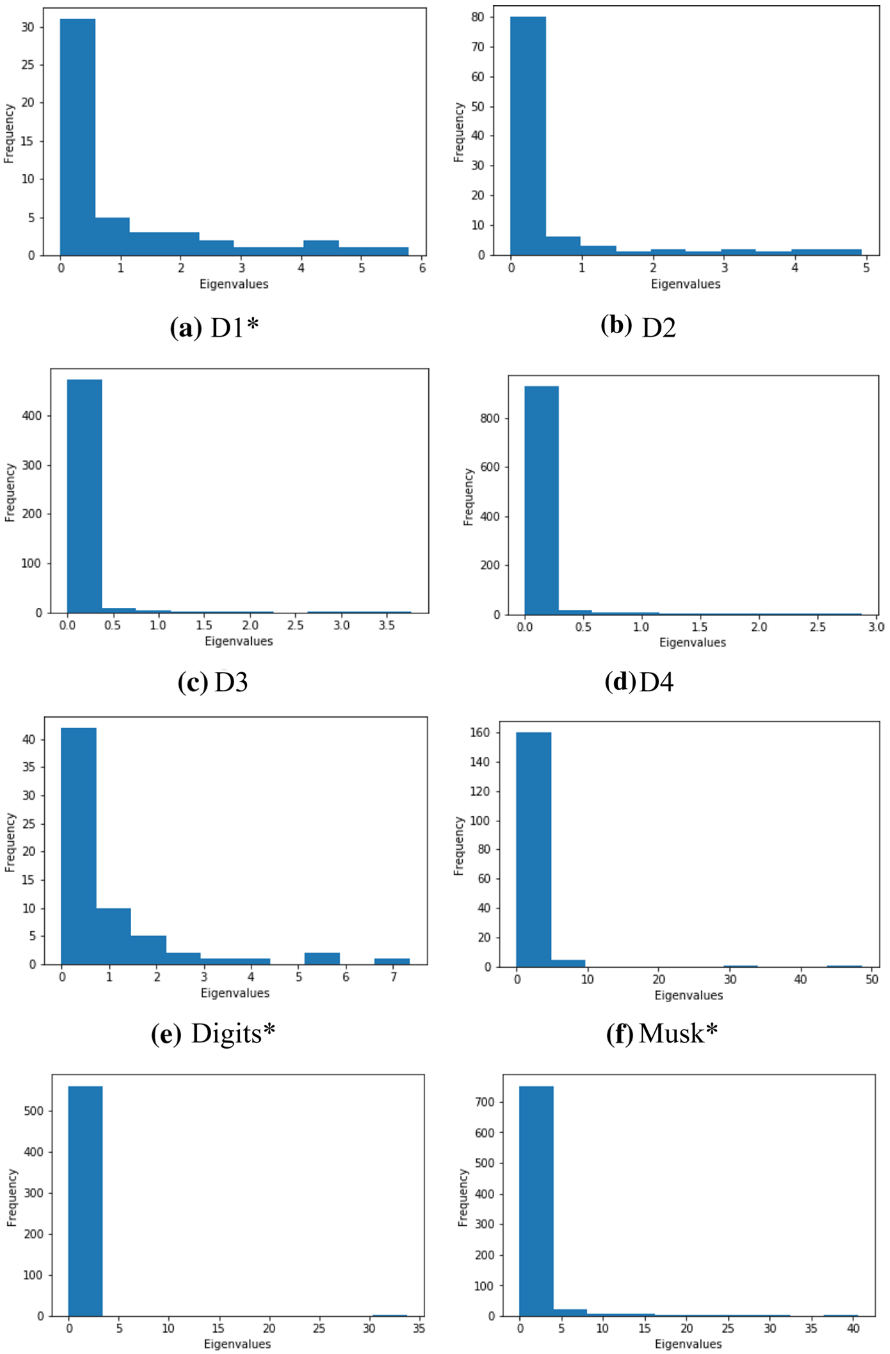
Details of the datasets in Section 2.4 and Section 2.5

The eigenvalues of the datasets used in Figs. 1 and 2 are given below.

$d = 50$ eigenvalues: [5.0, 4.0, 3.0, 2.0, 1.0, 1.0, 0.7, 0.7, 0.6, 0.4, 0.4, 0.4, 0.2, 0.2, 0.1, 0.1, 0.07, 0.04, 0.03, 0.01, 0.009, 0.007, 0.003, 0.001, 0.0009, 0.0002, 1e-05, 4e-08, 1e-08, 8e-12, 1e-16, 8e-17, 1e-22, 6e-23, 8e-28, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].

$d = 150$ eigenvalues: [5.0, 4.0, 3.0, 2.0, 1.0, 1.0, 1.0, 0.8, 0.8, 0.5, 0.5, 0.5, 0.4, 0.4, 0.3, 0.3, 0.2, 0.2, 0.1, 0.1, 0.1, 0.08, 0.08, 0.07, 0.06, 0.05, 0.04, 0.04, 0.03, 0.02, 0.02, 0.02, 0.007, 0.007, 0.006, 0.006, 0.005, 0.005, 0.002, 0.002, 0.0004, 0.0002, 0.0001, 0.0001, 3e-05, 2e-05, 2e-05, 1e-05, 8e-06, 7e-06, 5e-06, 4e-06, 4e-06, 2e-06, 7e-07, 6e-07, 2e-07, 2e-07, 3e-08, 3e-08, 2e-08, 5e-10, 4e-10, 1e-10, 8e-11, 6e-11, 3e-11, 2e-11, 4e-12, 2e-12, 3e-13, 3e-13, 2e-13, 9e-14, 1e-14, 2e-15, 1e-15, 8e-16, 4e-16, 3e-17, 1e-17, 1e-17, 7e-18, 1e-18, 1e-18, 5e-19, 2e-19, 4e-20, 3e-20, 2e-20, 4e-22, 8e-25, 3e-25, 8e-28, 5e-31, 2e-37, 6e-38, 2e-38, 9e-40, 1e-41, 5e-43, 9e-50, 1e-52, 9e-65, 8e-105, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].

The datasets used in Table 1 in Sect. 2.5 were generated using Python code as follows. For the given values of d, N and R:



(a) D1*

(b) D2

(c) D3

(d) D4

(e) Digits*

(f) Musk*

(g) HAR

(h) MNIST*

Fig. 11 Eigenvalues of the datasets used in Section 3.1.2. A dataset with a * in its caption has been ▶ rescaled such that each variable has zero mean and unit variance, and hence the eigenvalues of the correlation matrix are given

```
true_sigma = np.diag([np.random.rand() for _
  in range(R)] + [0] * (d - R))
  X = np.random.multivariate_normal(np.zeros(d),
  sigma, N).T
  empirical_sigma = np.cov(X)
```

Eigenvalues of the datasets in Section 3.1

Datasets in Section 3.1.1, with $d < N$

The histograms in Fig. 10 give the distribution of the eigenvalues of the datasets used in Sect. 3.1.1, detailed in Table 2.

Datasets in Section 3.1.2, with $d \geq N$

The histograms in Fig. 11 give the distribution of the eigenvalues of the datasets used in Sect. 3.1.2, detailed in Table 5.

Time to compute the minimal-variance polynomial in Section 3.1

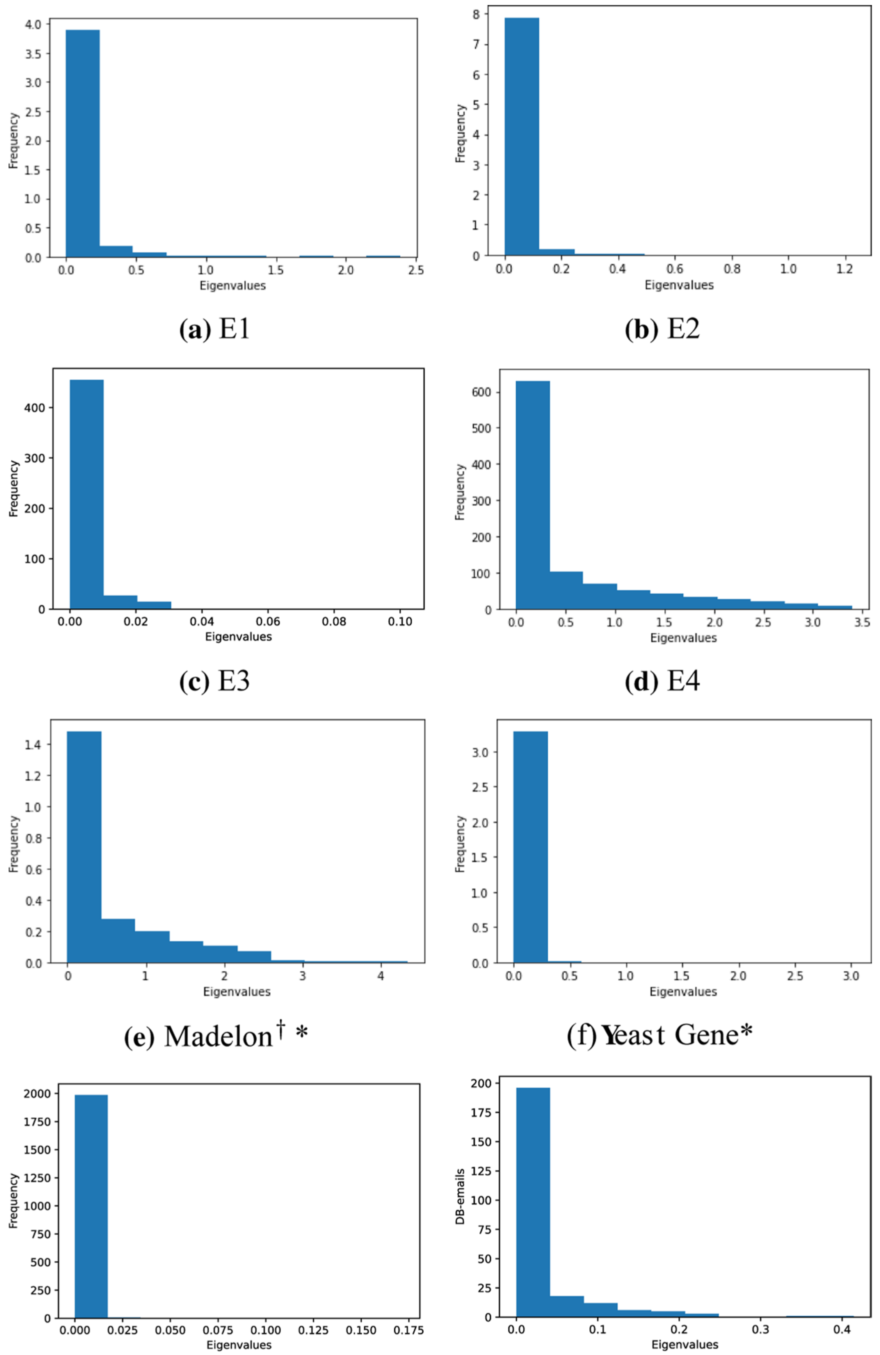
Table 13 gives the time it took to calculate the minimal-variance polynomial (in seconds) for each dataset used in Sect. 3.1, for each different value of k.

Eigenvalues of the datasets in Section 3.2

In Sect. 3.2, we compare different whitening methods with the minimal-variance polynomial whitening method by applying them to the Iris dataset and the Wisconsin breast cancer dataset (the latter of which we have scaled to improve performance). The eigenvalues of these datasets are given below:

Eigenvalues of Iris: [4.2282, 0.2427, 0.0782, 0.0238]

Eigenvalues of Wisconsin Breast Cancer: [9.8005, 8.2868, 3.3664, 2.2588, 1.5496, 1.4151, 1.1688, 0.9771, 0.5900, 0.5073, 0.4427, 0.3733, 0.3303, 0.2486, 0.2024, 0.1211, 0.1064, 0.0798, 0.0737, 0.0519, 0.0452, 0.0369, 0.0302, 0.0250, 0.0226, 0.0186, 0.0144, 0.0125, 0.0058, 0.0026, 0.0010, 0.0004].



(a) E1

(b) E2

(c) E3

(d) E4

(e) Madelon† *

(f) Yeast Gene*

(g) Colon*

(h) DB-emails

Table 13 Time taken to calculate A_k in seconds for each dataset (average over 100 runs)

Dataset	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$	$k = 10$
D1	0.04	0.04	0.05	0.04	0.04	0.06	0.04	0.05
D2	0.25	0.23	0.23	0.21	0.20	0.23	0.26	0.27
D3	1.35	1.43	1.72	2.08	1.98	2.00	2.59	3.11
D4	4.16	4.73	5.91	6.62	8.47	9.77	12.02	14.36
Digits	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Musk	0.21	0.22	0.22	0.23	0.25	0.26	0.29	0.31
HAR	1.98	2.15	2.34	2.62	3.81	3.54	3.83	4.42
MNIST	6.24	6.55	7.24	8.20	10.11	12.29	12.71	14.35

Acknowledgements The authors are very grateful to both reviewers for many constructive comments which have led to significant improvements.

References

Abdi H, Williams LJ (2010) Principal component analysis. WIREs Comput Stat 2(4):433–459

Agostinelli C, Greco L (2019) Weighted likelihood estimation of multivariate location and scatter. TEST 28(3):756–784

Akeret J, Refregier A, Amara A, Seehars S, Hasner C (2015) Approximate Bayesian computation for forward modeling in cosmology. J Cosmol Astropart Phys 08:043

Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc Natl Acad Sci 96(12):6745–6750

Anaya-Izquierdo K, Critchley F, Vines K et al (2011) Orthogonal simple component analysis: a new, exploratory approach. Ann Appl Stat 5(1):486–522

Anguita D, Ghio A, Oneto L et al. (2013) A public domain dataset for human activity recognition using smartphones. In: Esann, vol 3, p 3

Aoshima M, Yata K (2018) Two-sample tests for high-dimension, strongly spiked eigenvalue models. Stat Sin 43–62

Aoshima M, Shen D, Shen H, Yata K, Zhou YH, Marron J (2018) A survey of high dimension low sample size asymptotics. Aust N Z J Stat 60(1):4–19

Bai J, Shi S (2011) Estimating high dimensional covariance matrices and its applications. Ann Econ Finance 12(2):199–215

Baktash E, Karimi M, Wang X (2017) Covariance matrix estimation under degeneracy for complex elliptically symmetric distributions. IEEE Trans Veh Technol 66(3):2474–2484

Beaumont MA (2019) Approximate Bayesian computation. Annu Rev Stat Appl 6(1):379–403

Bickel PJ, Levina E (2004) Some theory for Fisher's linear discriminant function, naive Bayes', and some alternatives when there are many more variables than observations. Bernoulli 10(6):989–1010

Bingham E, Mannila H (2001) Random projection in dimensionality reduction: applications to image and text data. In: Proc. seventh ACM SIGKDD int. conf. knowl. discov. data min., pp 245–250

Blum A, Hopcroft J, Kannan R (2014) Foundations of data. Science. https://doi.org/10.13140/2.1.5115.0726

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

Cai T, Liu W, Luo X (2011) A constrained $\ell 1$ minimization approach to sparse precision matrix estimation. J Am Stat Assoc 106(494):594–607

Cai TT, Ren Z, Zhou HH et al (2016) Estimating structured high-dimensional covariance and precision matrices: optimal rates and adaptive estimation. Electron J Stat 10(1):1–59

Campos G, Zimek A, Sander J et al (2016) On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Min Knowl Discov 30(4):891–927

Cayley A (1858) II. A memoir on the theory of matrices. Philos Trans R Soc Lond 31:17–37

Chen RB, Guo M, Härdle WK, Huang SF (2015) COPICA-independent component analysis via copula techniques. Stat Comput 25(2):273–288

Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml

Filannino M (2011) Dbworld e-mail classification using a very small corpus. The University of Manchester

Fisher RA et al (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7(2):179–188

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

Gillard J, O'Riordan E, Zhigljavsky A (2022) Simplicial and minimal-variance distances in multivariate data analysis. J Stat Theory Pract 16(1):1–30

Givens CR, Shortt RM (1984) A class of Wasserstein metrics for probability distributions. Mich Math J 31(2):231–240

Hall P, Marron JS, Neeman A (2005) Geometric representation of high dimension, low sample size data. J R Stat Soc Ser B (Statistical Methodology) 67(3):427–444

Hamilton WR (1853) Lectures on quaternions. Hodges Smith

Härdle W, Simar L (2007) Applied multivariate statistical analysis, vol 22007. Springer, Berlin

Healy M (1968) Multiple regression with a singular matrix. J R Stat Soc C (Appl Stat) 17(2):110–117

Higham NJ (2008) Functions of matrices: theory and computation. SIAM

Higham NJ, Strabić N (2016) Anderson acceleration of the alternating projections method for computing the nearest correlation matrix. Numer Algorithms 72(4):1021–1042

Hoang HS, Baraille R (2012) A regularized estimator for linear regression model with possibly singular covariance. IEEE Trans Autom Control 58(1):236–241

Hossain M (2016) Whitening and coloring transforms for multivariate Gaussian random variables. Proj Rhea 3

Hoyle DC (2011) Accuracy of pseudo-inverse covariance learning-a random matrix theory analysis. IEEE Trans Pattern Anal Mach Intell 33(7):1470–1481

Huang L, Yang D, Lang B, Deng J (2018) Decorrelated batch normalization. In: Proc. IEEE conf. comput. vis. pattern recognit., pp 791–800

Huang L, Zhao L, Zhou Y, Zhu F, Liu L, Shao L (2020) An investigation into the stochasticity of batch whitening. In: Proc. IEEE conf. comput. vis. pattern recognit., pp 6439–6448

Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218

Hyvärinen A, Oja E (2000) Independent component analysis: algorithms and applications. IEEE Trans Pattern Anal Mach Intell 13(4–5):411–430

Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Int. conf. mach. learn, PMLR, pp 448–456

Ito T, Kubokawa T et al. (2015) Linear ridge estimator of high-dimensional precision matrix using random matrix theory. Tech Repore F-995 CIRJE Fac Econ, Univ Tokyo

Jain AK (2010) Data clustering: 50 years beyond k-means. Pattern Recognit Lett 31(8):651–666

Janková J, van de Geer S (2017) Honest confidence regions and optimality in high-dimensional precision matrix estimation. TEST 26(1):143–162

Jolliffe I (1986) Principal component analysis. Springer Verl, Berlin

Jolliffe IT, Cadima J (2016) Principal component analysis: a review and recent developments. Philos Trans R Soc A Math Phys Eng Sci 374(2065):20150202

Kandanaarachchi S, Muñoz MA, Hyndman RJ, Smith-Miles K (2020) On normalization and algorithm selection for unsupervised outlier detection. Data Min Knowl Discov 34(2):309–354

Kessy A, Lewin A, Strimmer K (2018) Optimal whitening and decorrelation. Am Stat 72(4):309–314

Kishore Kumar N, Schneider J (2017) Literature survey on low rank approximation of matrices. Linear Multilinear Algebra 65(11):2212–2244

Koivunen A, Kostinski A (1999) The feasibility of data whitening to improve performance of weather radar. J Appl Meteorol 38(6):741–749

LeCun Y, Cortes C, Burges C (2010) MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist

Ledoit O, Wolf M (2004) A well-conditioned estimator for large-dimensional covariance matrices. J Multivar Anal 88(2):365–411

Li D, Chen C, Lv Q, Yan J, Shang L, Chu S (2016) Low-rank matrix approximation with stability. In: Proc. 33rd int. conf. mach. learn., PMLR, vol 48, pp 295–303

Li G, Zhang J (1998) Sphering and its properties. Indian J Stat A Sankhyā 119–133

Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. J Am Soc Inf Sci Technol 58(7):1019–1031

Lloyd S (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137

Luo P (2017) Learning deep architectures via generalized whitened neural networks. In: Int. conf. mach. learn, PMLR, pp 2238–2246

Mahalanobis PC (1936) On the generalised distance in statistics. Proc Natl Inst Sci India 49–55

Malsiner-Walli G, Frühwirth-Schnatter S, Grün B (2016) Model-based clustering based on sparse finite Gaussian mixtures. Stat Comput 26(1–2):303–324

Martens H, Høy M, Wise BM, Bro R, Brockhoff PB (2003) Pre-whitening of data by covariance-weighted pre-processing. J Chemom J Chemom Soc 17(3):153–165

Mathai AM, Provost SB (1992) Quadratic forms in random variables: theory and applications. Dekker, New York

Pearson K (1901) LIII. On lines and planes of closest fit to systems of points in space. Lond Edinb Dublin Philos Mag J Sci 2(11):559–572

Prangle D (2017) Adapting the ABC distance function. Bayesian Anal 12(1):289–309

Pronzato L, Wynn HP, Zhigljavsky AA (2017) Extended generalised variances, with applications. Bernoulli 23(4A):2617–2642

Pronzato L, Wynn HP, Zhigljavsky AA (2018) Simplicial variances, potentials and Mahalanobis distances. J Multivar Anal 168:276–289

Qi H, Sun D (2011) An augmented Lagrangian dual approach for the H-weighted nearest correlation matrix problem. IMA J Numer Anal 31(2):491–511

Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65

Schuler A, Liu V, Wan J, Callahan A, Udell M, Stark DE, Shah NH (2016) Discovering patient phenotypes using generalized low rank models. Biocomput. In: Proc. pac. symp. World Scientific, pp 144–155

Seber GA, Lee AJ (2012) Linear regression analysis, vol 329. John Wiley & Sons, USA

Shi X, Guo Z, Nie F, Yang L, You J, Tao D (2015) Two-dimensional whitening reconstruction for enhancing robustness of principal component analysis. IEEE Trans Pattern Anal Mach Intell 38(10):2130–2136

Steinley D (2004) Properties of the hubert-arable adjusted rand index. Psychol Methods 9(3):386

Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, Church GM (1999) Systematic determination of genetic network architecture. Nat Genet 22(3):281–285

Thameri M, Kammoun A, Abed-Meraim K, Belouchrani A (2011) Fast principal component analysis and data whitening algorithms. In: Int. workshop syst. signal process. their appl. WOSSPA IEEE, pp 139–142. IEEE

Udell M, Townsend A (2019) Why are big data matrices approximately low rank? SIAM J Math Data Sci 1(1):144–160

Vanschoren J, van Rijn JN, Bischl B, Torgo L (2013) OpenML: networked science in machine learning. SIGKDD Explor 15(2):49–60

Vidal R, Favaro P (2014) Low rank subspace clustering (LRSC). Pattern Recognit Lett 43:47–61

Wang W, Fan J (2017) Asymptotics of empirical eigenstructure for high dimensional spiked covariance. Ann Stat 45(3):1342

Wegmann D, Leuenberger C, Excoffier L (2009) Efficient approximate Bayesian computation coupled with Markov chain Monte Carlo without likelihood. Genetics 182(4):1207–1218

Wolberg WH, Street WN, Mangasarian OL (1992) Breast cancer Wisconsin (diagnostic) data set. UCI Mach Learn Repos [http://www.archive-ics-uci-edu/ml/]

Wu D, Wang D, Zhang MQ, Gu J (2015) Fast dimension reduction and integrative clustering of multi-omics data using low-rank approximation: application to cancer molecular classification. BMC Genom 16(1):1022

Xiao Z (2020) Efficient GMM estimation with singular system of moment conditions. Stat Theory Relat Fields 4(2):172–178

Yang L, Jin R (2006) Distance metric learning: a comprehensive survey. Mich State Univ 2(2):4

Yata K, Aoshima M (2010) Effective PCA for high-dimension, low-sample-size data with singular value decomposition of cross data matrix. J Multivar Anal 101(9):2060–2077

Yata K, Aoshima M (2012) Effective PCA for high-dimension, low-sample-size data with noise reduction via geometric representations. J Multivar Anal 105(1):193–215

Yata K, Aoshima M (2013) PCA consistency for the power spiked model in high-dimensional settings. J Multivar Anal 122:334–354

Ye J, Xiong T (2006) Null space versus orthogonal linear discriminant analysis. In: Proc. 23rd int. conf. mach. learn., pp 1073–1080

Zafeiriou S, Laskaris N (2008) On the improvement of support vector techniques for clustering by means of whitening transform. IEEE Signal Process Lett 15:198–201

Zhao Y, Nasrullah Z, Li Z (2019) PyOD: a Python toolbox for scalable outlier detection. J Mach Learn Res 20(96):1–7

Zhou Y, Wilkinson D, Schreiber R, Pan R (2008) Large-scale parallel collaborative filtering for the Netflix prize. In: Int. conf. algorithmic appl. manag. Springer, pp 337–348

Zuanetti DA, Müller P, Zhu Y, Yang S, Ji Y (2019) Bayesian nonparametric clustering for large data sets. Stat Comput 29(2):203–215

Zuber V, Strimmer K (2009) Gene ranking and biomarker discovery under correlation. Bioinform 25(20):2700–2707