Contents lists available at ScienceDirect

Journal of Computational Mathematics and Data Science

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

A review on the selection criteria for the truncated SVD in Data Science applications

Antonella Falini

Computer Science Department, Via Edoardo Orabona 4, Bari, 70125, Italy

ARTICLE INFO

Keywords:
SVD
Truncated SVD
Component selection
Image processing

ABSTRACT

The Singular Value Decomposition (SVD) is one of the most used factorizations when it comes to Data Science applications. In particular, given the big size of the processed matrices, in most of the cases, a truncated SVD algorithm is employed. In the following manuscript, we review some of the state-of-the-art approaches considered for the selection of the number of components (i.e., singular values) to retain to apply the truncated SVD. Moreover, three new approaches based on the Kullback–Leibler divergence and on unsupervised anomaly detection algorithms, are introduced. The revised methods are then compared on some standard benchmarks in the image processing context.

1. Preliminary notions

The Singular Value Decomposition (SVD) is a matrix factorization technique that was discovered over 100 years ago, independently by Eugenio Beltrami (1835–1899) and Camille Jordan (1838–1921), see [1]. At the same time, James Joseph Sylvester (1814–1897), Erhard Schmidt (1876–1959), and Hermann Weyl (1885–1955) also discovered the SVD using different methods, see [1]. The development in the 1960s of practical methods for computing the SVD transformed the field of numerical linear algebra. One method of particular note is the Golub and Reinsch algorithm from [2]. We refer to the documentation for the Linear Algebra Package (LAPACK) by Anderson et al. [3] and Golub and Van Loan [4] for details on current algorithms to calculate the SVD for dense, structured, or sparse matrices.

For the following properties and definitions summary we refer to Bai et al. [5] and Eldén [6].

Theorem 1.1. *Given a matrix $A \in \mathbb{R}^{m\times n}$, it is possible to factorize it as:*

$$A = U\Sigma V^{\top}, \tag{1}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthonormal and $\Sigma \in \mathbb{R}^{m\times n}$ has elements $\sigma_{ij} = 0$ for $i \neq j$ and for $i = j$ has elements $\sigma_{ii} = \sigma_i$, with

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0, \quad p = \min\{m, n\}.$$

The columns of the matrices U and V are called left and right singular vectors, respectively. While the diagonal elements of Σ are said singular values.

Other versions of the SVD are often used in practice, especially when it comes to reduce memory storage or noise removal.

Definition 1.1 (*Thin SVD*)**.** Denoting $p = \min\{m, n\}$, U_p be the submatrix obtained by selecting the first p columns of U, in symbols $U_p := [u_1, \dots, u_p]$, V_p be the submatrix obtained by selecting the first p columns of V, $V_p := [v_1, \dots, v_p]$ and $\Sigma_p := diag(\sigma_1, \dots, \sigma_p)$, then A can be factorized as

$$A = U_p \Sigma_p V_p^{\top}.$$

E-mail address: antonella.falini@uniba.it.

https://doi.org/10.1016/j.jcmds.2022.100064
Received 10 March 2022; Received in revised form 26 July 2022; Accepted 15 October 2022

Definition 1.2 (*Compact SVD*)**.** Let $r := rank(A) \leq \min\{m, n\}$, then the following,

$$A = U_r \Sigma_r V_r^\top,$$

with $U_r := [u_1, \dots, u_r]$, $V_r := [v_1, \dots, v_r]$ and $\Sigma_r := diag(\sigma_1, \dots, \sigma_r)$, is the so called compact SVD of the matrix A.

Definition 1.3 (*Truncated SVD*)**.** Chosen k such that $k < rank(A)$, then we can construct the so called truncated SVD,

$$A_k = U_k \Sigma_k V_k^\top,$$

with $U_k := [u_1, \dots, u_k]$, $V_k := [v_1, \dots, v_k]$ and $\Sigma_k := diag(\sigma_1, \dots, \sigma_k)$.

Proposition 1.1. *The 2-norm of a matrix A is equal to the biggest singular value σ_1.*

Theorem 1.2. *Given $A \in \mathbb{R}^{m\times n}$ and k such that $k < rank(A)$, then the following approximation problems,*

$$\min_{rank(Z)=k} \|A - Z\|_2 \tag{2}$$

$$\min_{rank(Z)=k} \|A - Z\|_F \tag{3}$$

have as solution the matrix $Z := U_k \Sigma_k V_k^\top$. Moreover, the minimum of problem (2) is given by σ_{k+1}, while the minimum of problem (3) is $\left(\sum_{i=k+1}^{p} \sigma_i^2\right)^{1/2}$, with $p = \min\{m, n\}$ and $\|\cdot\|_F$ denoting the Frobenius norm.

The SVD gives an orthonormal basis for the four fundamental subspaces of a matrix:

Theorem 1.3. *Denoting with r the rank of the given matrix $A \in \mathbb{R}^{m\times n}$, the following hold:*

1. *The singular vectors $u_1, \dots, u_r$ are an orthonormal basis in $R(A)$, the range of A.*
2. *The singular vectors $v_{r+1}, \dots, v_n$ are an orthonormal basis in $N(A)$, the null space of A.*
3. *The singular vectors $v_1, \dots, v_r$ are an orthonormal basis in $R(A^\top)$.*
4. *The singular vectors $u_{r+1}, \dots, u_m$ are an orthonormal basis in the $N(A^\top)$.*

The SVD is strongly related also to the so called *principal component analysis* (PCA). The PCA is defined as an orthogonal linear transformation that transforms the assayed data to a new coordinates system to preserve as much of the data's variations as possible. Indeed the first component is the direction over which to project the data to preserve their variance, the second component is orthogonal to the first one and maximizes the residual variance of the projected data, and so on, see [7]. Given the SVD of a data matrix A, we can factorize the matrix $A^\top A$ in the following way:

$$\begin{aligned} A^\top A &= (U\Sigma V^\top)^\top (U\Sigma V) \\ &= V\Sigma^\top U^\top U\Sigma V^\top \\ &= V\Sigma^\top \Sigma V^\top \\ &= V\hat{\Sigma}^2 V^\top, \end{aligned}$$

where by $\hat{\Sigma}^2$ we denote a square diagonal matrix having as entries the squared singular values of A. We can note that the right singular vectors V are equivalent to the eigenvectors of $A^\top A$, while the singular values of A are the square root of the eigenvalues of $A^\top A$.

Remark 1.1. Notice that the same computation can be carried out on the matrix $AA^\top$, producing similar observations for the left singular vectors.

When the PCA is performed on the data A, usually the so called covariance matrix is constructed. The covariance matrix for A is given by centering every column vector with respect to its mean value, and storing the new elements into a matrix $\tilde{A}$. After that, the covariance matrix can be computed as $\tilde{A}^\top \tilde{A}$.

As a final part we observe that the column vectors of the matrices U and V are orthonormal, and henceforth they define an m-frame and an n-frame, respectively, for the corresponding Stiefel manifolds. In particular, following the notation from [8], taking the matrix U, the Stiefel manifold defined by the selected k-frames column vectors of U, over the scalar field $\mathbb{R}$ can be defined as,

$$Y_{m,k}(\mathbb{R}) = \{Y \in \mathbb{R}^{m\times k} | Y^\top Y = I\}.$$

Geometrically, an element of the Stiefel manifold can be pictured as a set of orthogonal, unit-length vectors that are rigidly connected to one another. Although the elements of the Stiefel manifold can be represented by $m \times k$ matrices, the effective dimension of the manifold is less than mk due to the constraints that need to be imposed. The first column vector must satisfy a single constraint: the unit-length constraint. The second column vector must satisfy two constraints: unit length and orthogonality to the first column vector. The third column vector must additionally be orthogonal to the second column vector, and so on. Therefore the effective dimension of the Stiefel manifold can be computed as,

$$d := mk - 1 - 2 - \cdots k = mk - \frac{k(k+1)}{2}.$$

These last notions will be useful in Section 2.7, when probabilistic approaches will be revised.

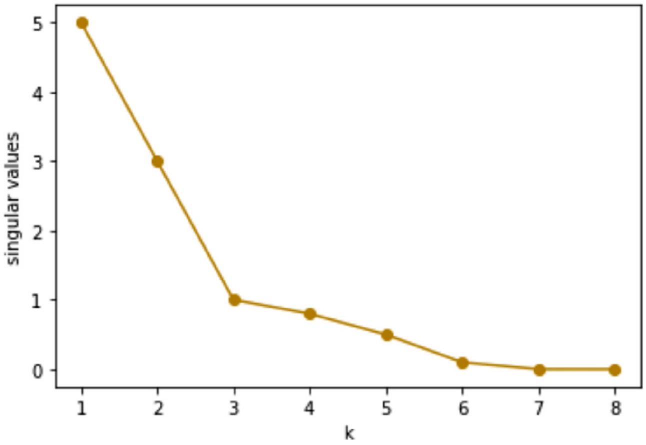


Fig. 1. Scree plot: the characteristic elbow can be identified at $k = 3$, in this case.

2. Methodologies

Selecting the cutoff value k defines the central model order selection problem of the truncated SVD. In the following section we revise the state-of-the-art approaches to identify the number k to truncate the SVD and to still get a significant approximation.

2.1. Scree plots

The *scree-plot* was introduced by Cattell [9]. It consists in drawing the singular values in a Cartesian coordinate system and then k is chosen as the "elbow" of the graph, see Fig. 1. An alternative to this approach is given by plotting the logarithm of the singular values and in this case k is chosen as the value where the diagram almost becomes a straight line. In certain cases of few singular values with rather big gaps between the biggest and the smallest, the scree plot is rather informative. However, in general, it is quite subjective where to cut off the retained singular values.

2.2. Guttman-Kaiser rule

In its original formulation, see [10,11], this rule assesses to retain all those singular values which exceed 1. There are also some simple modifications where either the mean values or a specific threshold value is sought and hence, only the σ_i bigger than the chosen threshold are kept. Although being a very simple and intuitive criterion, it often overestimates the number of singular values to keep, besides being again quite subjective to the choice of the threshold parameter. Moreover, it was argued by Efron [12] and Lambert et al. [13] that besides being arbitrary, it also ignores the error due to sampling. For these reasons these authors proposed the *bootstrap Guttman–Kaiser* rule.

2.3. Hard threshold for singular values

The following method is proposed in [14]. The authors introduce a technique able to estimate the best low-rank approximation given by a hard thresholding of the singular values. In particular, the singular values smaller than τ^* are set to zero with,

$$\tau^* = \omega * \sigma_{med}, \tag{4}$$

σ_{med} median of the given singular values and ω a suitable weight. For all the details we refer to Donoho and Gavish [14] and to the references therein.

2.4. Randomized approaches

Randomized range-finder

Randomized range finder algorithms have the general scheme reported in Algorithm 1.

The goal of Algorithm 1 is to produce an orthonormal matrix Q of size $m \times \ell$, with $\ell \ll \min\{m, n\}$ be the number of singular values of A that exceed a given tolerance ε, such that

$$\|(I - QQ^*)A\| \le \varepsilon, \tag{5}$$

```
Data: Matrix $A$ of size $m \times n$, and an integer $\ell$.
Result: Orthonormal matrix $Q$ of size $m \times \ell$ approximating $R(A)$.
Generate an $n \times \ell$ Gaussian random matrix $\Omega$;
Compute the $m \times \ell$ matrix $Y = A\Omega$;
Compute the QR factorization of $Y$;
```

Algorithm 1: Randomize range finder

with $\|\cdot\|$ denoting the ℓ_2 norm, and the operator $*$ be the adjoint of a matrix. The number of columns ℓ that the algorithm needs to reach this threshold is usually slightly larger than the minimal rank k of the smallest basis satisfying the condition above. In practice usually, ℓ is not known, hence its value is determined by adaptively iterating algorithm 1, see [15]. Once the matrix Q is computed such that Eq. (5) holds, it is easy to construct an approximate singular value decomposition of the input matrix A, with Algorithm 2.

```
Data: Matrix $A$ of size $m \times n$, and an orthonormal matrix $Q$ of size $n \times \ell$, such that Eq. (5) holds.
Result: Matrices $U, \Sigma, V$ such that $A \approx U\Sigma V^*$.
Construct the matrix $B = Q^* A$;
Compute the SVD of $B$, such that $B = \widetilde{U}\Sigma V^*$;
Define $U := Q\widetilde{U}$.
```

Algorithm 2: Approximate SVD.

Flipping sign

The second approach described here can be derived from the technique described in [16]. In particular, the authors propose a fast computation of a low-rank approximation when the singular values of the original matrix are significantly greater than the ones of a random matrix of the same size. In order to compute how many singular values should be retained we could proceed as follows.

Given the matrix A, we point-wise multiply every element by a random -1, $+1$, getting the new matrix $\tilde{A}$. After this type of multiplication the Frobenius norm of A does not change, while the 2-norm of A could change whether A contains structure or not. Denoting with A_{-k} the matrix computed by selecting the last $m-k$ columns of U, the last $n-k$ rows of $V^\top$ and a conformal block-size from the matrix Σ, we obtain the matrix $\tilde{A}_{-k}$ in the same manner by randomly multiplying every element by -1, $+1$. The number k is selected such that

$$\frac{\|A_{-k}\|_2 - \|\tilde{A}_{-k}\|_2}{\|A_{-k}\|_F} \quad \text{is small.}$$

Thus, the matrix A_{-k} will contain only noise, see [16] for additional details. More in general, the issue of selecting a good number of singular values to retain is tightly connected to the construction of low-rank approximations of the original matrix, non necessarily derived by using the truncated SVD algorithm, see [17,18], and references therein for additional details.

Parallel analysis

The last method described in this section is due to Horn [19] and it is called *parallel analysis*. Horn introduced parallel analysis after considering Kaiser rule. Kaiser developed the greater-than-one rule following a formal treatment by Guttman (1954), but, it is essentially an asymptotical and theoretical lower bound to the number of true and reliable structural dimensions at the population level. At the sample level, Horn reasoned that one would expect to see eigenvalues greater than and less than 1 simply because of "sample bias".

Indeed, what we can empirically observe is the so called *reference curve*, which "deviates" from the constant line $\lambda = 1$ according to the magnitude of the sampling error. Hence, only the eigenvalues which exceed the intersection point, between the line $\lambda = 1$ and the reference curve, should be retained.

The reference curve is established by randomly generating a large number of correlation matrices (K). Usually, K should be between 30 to 50 replications per empirical correlation matrix. This accounts for the expression, "parallel analysis". Later on, a principal component analysis is conducted on each of these correlation matrices, yielding K sets of $1, 2, 3, \dots, k$ eigenvalues. The average values of the first, second, and so on till k eigenvalues are then calculated over the K matrices and are plotted as the reference curve on the same graph as the empirical data. The number of components to retain is indicated by the point at which the eigenvalue plot of the empirical data intersects that one of the reference curve.

2.5. Entropy based selection

The entropy-criterion was firstly introduced by Shannon [20] related to the information theory framework. In particular, for a given set of macroscopic variables, the entropy measures the degree to which the probability of the system is spread out over different possible microstates. The entropy E of a discrete r.v. X measures the amount of uncertainty associated to X. Given the set of x values that can be assumed by the variable X, and denoting by $p(x) := P(X = x)$ the probability that the r.v. X assumes value x, then the entropy associated to X can be quantified as,

$$E(X) := -\sum_{x} p(x)\log(p(x)). \tag{6}$$

Using Eq. (6), Alter et al. [21] propose the SVD-based entropy of a given dataset A defined as Eq. (7),

$$E(A) = -\frac{1}{\log(r)}\sum_{j=1}^{r} f_j \log(f_j), \tag{7}$$

with

$$f_j := \frac{\sigma_j^2}{\sum_{i=1}^{r}\sigma_i^2}. \tag{8}$$

If $E(A) = 1$, or in general, a high entropy means that all the singular values give the same contribution. Otherwise, $E(A) = 0$ corresponds to what is usually identified as a rank 1 problem. In general, a low entropy means that the most important information can be explained by the first eigenvectors. In general the entropy is computed in order to understand how many "components" are needed. To be able to reduce this number a very intuitive approach is given by identifying those singular values which are enough to get a certain percentage of the computed entropy. For instance, if we are interested in preserving the 70% of $E(A)$, then an heuristic selection method is given by:

Find the smallest integer k such that $E_k(A) = 0.7E(A)$, within a given tolerance; where $E_k(A)$ is computed by only using the first k addenda in (7). Usually the entropy defined as in Eq. (7) is employed when both, the covariance matrix and the PCA are used.

Following the ideas of Roberts et al. [22] and Sabatini [23], another possibility is to compute the entropy defined as:

$$E_{SVD}(A) = -\frac{1}{\log(r)}\sum_{j=1}^{r}\bar{\sigma}_j\log(\bar{\sigma}_j), \tag{9}$$

where $\bar{\sigma}_j := \sigma_j / \sum_{i=1}^{r}\sigma_i$ are considered as "normalized" singular values.

In particular, Sabatini observes that more complex systems are characterized by a spread of energy away from the first singular values that can be computed as:

$$-\sum_{i=1}^{r}\bar{\sigma}_j\log(\bar{\sigma}_j).$$

In this work we are only interested in revising the available criteria to select the first k biggest singular values, but, in different contexts, it is relevant to extract certain features rather than others. For this type of application, we refer to Varshavsky et al. [24] and Banerjee and Pal [25] for criteria based on the SVD and entropy computation, and in general to the review in [26] for unsupervised techniques.

2.6. Variance based methods

Given a dataset A, analyzing its "variance" requires the preliminary step to standardize its variables. This means that every variable will be transformed to have zero mean and variance equal to one. With the term "total variance" we informally refer to the sum of variances observed in the dataset A. The total variance should be computed as the trace of the covariance matrix $A^\top A$, after standardization, hence, it is directly proportional to the sum of squares of the singular values of A. There are two criteria based on the proportion of variance we need to account for.

Total variance

In the first case, we recall that the quantity f_j defined in (8) expresses the contribution of the singular value σ_j to the total variance. Then, we can decide to retain every singular value which accounts to at least 5% or 10% of the total variance, see [27].

Cumulative percentage of total variance

The second criterion is based on the cumulative percentage of the total variance, see [7], Chapter 6. We can retain the first k singular values such that the cumulative percentage of the total variance is bigger than a threshold value t^* between $70\% - 90\%$.

Find k such that,

$$\frac{\sum_{i=1}^{k}\sigma_i^2}{\sum_{i=1}^{r}\sigma_i^2} > t^*.$$

This criterion is sometimes also referred to as *energy-preservation*, see [28], Chapter 11.

Remark 2.1. Obviously every f_j factor can be computed using rank-1 matrices as,

$$f_j = \frac{\|u_j\sigma_j v_j^\top\|_F}{\|A\|_F^2} = \frac{\sigma_j^2}{\sum_i \sigma_i^2}.$$

Hence, every f_j can also be seen as the variance explained by every *mode* j in each column feature.

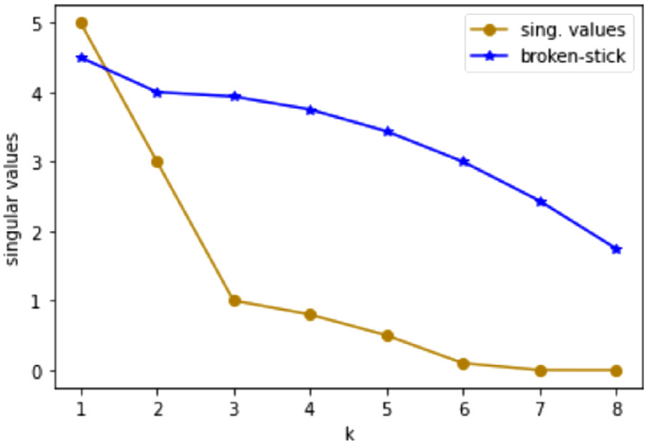


Fig. 2. Singular values and the broken-stick model.

Broken-stick model

The last method proposed in this subsection is the so called *Broken-stick* model, introduced by Frontier [29]. When the total variance is randomly divided amongst the various components, then the expected distribution of the squares of singular values will follow a broken-stick distribution, see Fig. 2. The important singular values to be retained are the ones which lie above the broken-stick model. Frontier, Legendre and Legendre [30] provide a table of eigenvalues based on the broken-stick distribution, but, following [7,31], given a stick of unit length, broken at random into k segments, the expected length of the kth longest segment can be easily computed as:

$$b_k := \frac{1}{r}\sum_{i=k}^{r}\frac{1}{i}.$$

2.7. Probabilistic approaches

Following [32], the problem of selecting the main principal components can be straightforwardly transferred to the problem of how many singular values should be retained. Minka suggests a Bayesian model selection by computing the *evidence* for the suggested (reduced) model. In practice we need to compute the probability of the data given the model, i.e.,

$$p(A|M) = \int_{\theta} p(A|\theta)P(\theta|M)\,d\theta, \tag{10}$$

where A is the given dataset, M is the constructed model and θ are (at this stage) the unknown parameters for the model M. We want to select a subspace of dimension k such that the probability expressed in (10) is maximal. Assuming the least informative priors (in order to avoid a biased model) and approximating the integral in (10) via the Laplace's method [33], denoting with $p(D|k)$ the probability of the data given the subspace of dimension k, after some computation, Minka arrives to the following expression:

$$p(D|k) \approx p(U)\left(\prod_{j=1}^{k}\sigma_j^2\right)^{\frac{N}{2}} \hat{v}^{\frac{-N(m-k)}{2}}(2\pi)^{\frac{(d+k)}{2}}\,det(A_Z)^{-\frac{1}{2}}N^{-\frac{k}{2}}. \tag{11}$$

In the above expression, $p(U)$ is the prior probability density for U that can be computed via Equation (20) in [32], as the area of the Stiefel manifold spanned by the orthogonal k frames of U. The number N denotes the number of features in the dataset A (i.e., the number of columns of A), m is the dimension of every feature (i.e., the number of rows of A), while the scalar $\hat{v}$ is computed as,

$$\hat{v} = \frac{\sum_{j=k+1}^{m}\sigma_j^2}{m-k}.$$

Finally, d is the dimension of the considered Stiefel manifold and $det(A_Z)$ is the determinant of the Hessian matrix of the integrand function in (10) which happens to be diagonal and thus its determinant can be computed as the product of its diagonal entries, see Eq. (27) in [32]. A simplification of expression (11) leads to the method proposed by Kass and Raftery [33],

$$p(D|k) \approx \left(\prod_{j=1}^{k}\sigma_j^2\right)^{N/2} \hat{v}^{\frac{-N(m-k)}{2}} N^{-(d+k)/2}.$$

Obviously, by changing the assumed priors and/or the approximation technique to evaluate the integral in (10), plenty of other methods can be derived. We refer to Kass and Raftery [33] and Minka [32] for completeness.

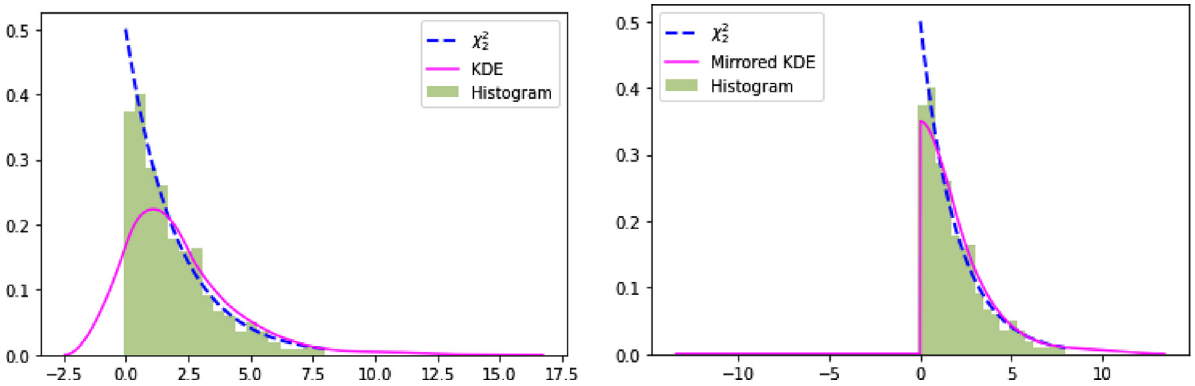


Fig. 3. Left: the kernel density estimate (KDE) for a bounded distribution assumes the density to continuously decrease to reach 0 for samples before the boundary. Right: mirrored KDE with respect to the boundary $x = 0$.

Log-likelihood based model

In [34], the authors construct a model for the singular values and they estimate the position of the "gap" or the "elbow" by maximizing a profile likelihood function. Zhu and Ghodsi assume that the singular values are drawn from two different distributions: one for the significant components and the other for the noise components. The maximum log-likelihood, which corresponds to the best choice for k, is then determined empirically, by comparing the profile log-likelihood functions for $k = 1, \dots, r$. This method makes more robust the search for the "elbow" in the Cattel's scree plot technique, and as shown by the authors, it seems reliable enough in several data-science benchmarks.

Kullback–Leibler divergence approach

The last method in this section is firstly introduced here. The idea comes from the quantities f_j as defined in (8). It is straightforward to observe that:

- f_j are all positives and in particular $0 < f_j < 1$, $\forall j$,
- $\sum_{j=1}^{r} f_j = 1$,

hence, we can think of every f_j as a sample from a continuous probability density distribution. We can construct a continuous model $p(x)$ for f_j with $j = 1, \dots, r$ and $r - 1$ continuous models $q_z(x)$, $z = 1, \dots, r - 1$, by grouping together the discrete samplings as: for $z = 1$, we take $j = 1, 2$; for $z = 2$ we take $j = 1, 2, 3$; for $z = 3$, we take $j = 1, 2, 3, 4$, and so on. Since we want to generate smooth continuous models we are going to use linear kernel density estimates. The kernel density estimator (KDE) has the general form:

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right). \tag{12}$$

In expression (12), h is a smoothing parameter, x_i are the discrete available samples and K is a positive function with the following properties:

- It is symmetric, i.e., $K(\mathbf{u}) = K(-\mathbf{u})$;
- $\int_{-\infty}^{+\infty} K = 1$;
- It is monotonically decreasing: $K'(\mathbf{u}) < 0$ when $\mathbf{u} > 0$;
- It has zero expected value: $\mathbb{E}[K] = 0$.

Specifically, in our case, since we choose a linear kernel (sometimes referred to as "triangular" due to the characteristic shape), the derived expression for K satisfies:

$$K(x, h) \propto 1 - \frac{x}{h}, \text{ if } x < h.$$

In order to construct a suitable density estimate we need to perform some "mirroring" with respect to the boundary, since we are dealing with a bounded distribution, i.e., the samples $0 < f_j < 1$. In general, given a bounded distribution, the estimated kernel density will produce a function which is continuously decreasing to zero for samples smaller than the boundary, see Fig. 3 (left). In order to avoid this behavior we can use a mirroring technique. In particular, if $\hat{g}_h(x)$ is the original KDE, then $\hat{g}^*_h(x) = \hat{g}_h(2a - x)$ is the new KDE obtained by mirroring the data about $x = a$. In our case $a = 0$, see Fig. 3 (right).

Once the smooth models are constructed, we are going to compute the Kullback–Leibler divergence KL_z between $p(x)$ and $q_z(x)$, for $z = 1, \dots, r - 1$:

$$KL_z := \int_X p(x) \log\left(\frac{p(x)}{q_z(x)}\right) dx. \tag{13}$$

Table 1
Acronyms for the implemented methods.

Acronym	Criteria
Scree	Cattell's scree plot
GK	Guttman–Kaiser rule
Bs	Broken-stick model
HT	Hard thresholding
Range finder	Randomized range-finder
Tot Var	Total variance
Cum Perc Var	Cumulative percentage of Tot Var
E-70	70% of total entropy
E-90	90% of total entropy
E-100	100% of total entropy
E$_{SVD}$-70	70% of total E$_{SVD}$
E$_{SVD}$-90	90% of total E$_{SVD}$
E$_{SVD}$-100	100% of total E$_{SVD}$
KL	Kullback–Leibler divergence approach
KIF	Kaiser-Isolation forest
KMIF	KMeans-Isolation forest

In expression (13), the set X is the domain for $p(x)$ and the integral in (13) is in practice evaluated by using the trapezoidal rule.

The Kullback–Leibler divergence is a common "distance" measure in data-science application in order to assess the similarity of two given distribution functions, see [35]. In our case we are going to select k as the index z for which $q_z(x)$ is most similar to $p(x)$.

The idea behind this method is to find for which k we get a similar density function to the one obtained by using r singular values, where r denotes the rank of our problem. Therefore, with this methodology we are going to construct a method to estimate the so called "numerical rank" of the problem.

2.8. Methods based on unsupervised anomaly detection

In this last subsection we describe two novel techniques firstly introduced here. Unsupervised anomaly detection refers to those methods that are able to identify outliers, and hence, behaviors that deviate from classical trends, without using labeled data.

Kaiser-Isolation Forest (KIF)

This criterion can be seen as a modification of the Kaiser rule where the threshold will be automatically decided according to the isolation forest algorithm, see [36]. In particular, every observation is isolated by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature. Since recursive partitioning can be represented by a tree structure, the number of splittings required to isolate a sample is equivalent to the path length from the root node to the terminating node. The path length, averaged over a forest of such random trees, is a measure of normality. Random partitioning produces noticeably shorter paths for anomalies, see [37]. From this algorithm, applied to f_j, for $j = 1, \dots, r$, we expect to get the biggest as well as the smallest f_j to be identified as anomalies. Obviously, since the f_j are sorted in descending order, we select k as the last index sequentially preceding a gap in the indices of the identified anomalies. Moreover, according to the type of assayed dataset, we can assume an apriori knowledge of the percentage of expected outliers. For example, in case we need to remove some Gaussian noise, the lesser components are selected, the highest will be the chances to succeed in our task. In other cases, like for example face-recognition task, on the contrary, it is important to not underestimate the number k, see for example [34], thus a good percentage of samples (i.e., at least 10%) should be considered as anomalous.

Kmeans-Isolation Forest (KMIF)

In the same spirit of Zhu and Ghodsi [34], for this method we also assume the samplings f_j to be generated by two different distributions: one for the important components and one for noise. This splitting is done in practice by using a KMeans clustering algorithm. The data are partitioned into two clusters such that the so called within-cluster-variance is minimal. Since KMeans is an unsupervised approach, the two produced clusters will have no labels. Hence, the cluster containing f_1, i.e., the smallest of the f_j which corresponds to the biggest σ_j, will be processed by the isolation forest algorithm in order to detect the anomalous samples. As in the KIF method, also in this case, we select k as the last index sequentially preceding a gap in the indices of the identified anomalies.

3. Applications

In this section we analyze three common benchmarks in the context of image processing by using the criteria reported in the previous section that were easily implementable. In particular, in Table 1 we report the acronyms and the corresponding used methods.

The implementation[1] is written in Python and uses libraries `scikit` see [38], `scipy` see [39], `numpy` see [40] and `KDEpy`.[2]

[1] The codes are freely accessible here: https://github.com/AntonellaFalini/AntonellaFalini.

[2] https://github.com/tommyod/KDEpy.

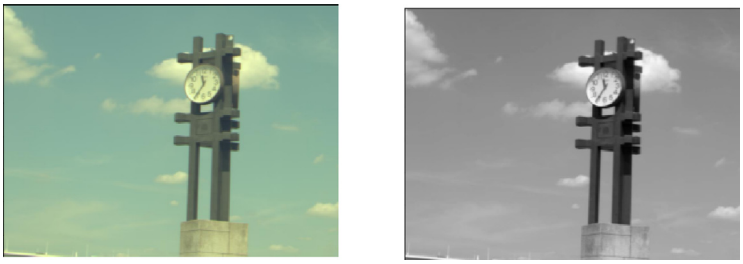


Fig. 4. Left: Selected RGB image. Right: Gray-scale resolution.

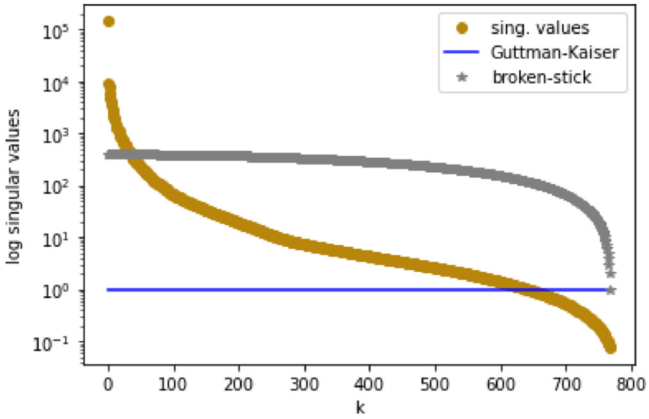


Fig. 5. Logarithmic diagram for the singular values of the assayed image.

Table 2
Results for the image compression task.

Criteria	k	Error
Scree	300	4.5e−4
GK	600	6.9e−5
Bs	60	5.6e−3
HT	244	6.8e−4
Range finder	100	5.8e−3
Tot Var	1	1.2e−1
Cum Perc Var	1	1.2e−1
E-70	8	4.2e−2
E-90	10	3.6e−2
E-100	11	3.3e−2
E_{SVD}-70	170	1.3e−3
E_{SVD}-90	219	8.3e−4
E_{SVD}-100	244	6.8e−4
KL	58	5.8e−3
KIF	38	9.5e−3
KMIF	31	1.3e−2

Image compression

The first analyzed task concerns the so called image-compression. We select one RGB image $\mathcal{I}$, see Fig. 4 (left), from the HS-SOD dataset,[3] which is usually employed to perform saliency detection tasks, see [41]. The goal here is to try to use the smallest k in order to achieve a good approximation of the original image. In order to perform the SVD, as a first step the selected image $\mathcal{I}$ is transformed into a gray-scale picture I, see Fig. 4 (right). Secondly, we compute the SVD and we run the criteria reported in Table 2 in order to select the best k. The size of I is 768 × 1024 and the rank of the gray-scale image is equal to 768. Due to a rather big magnitude of the first singular value compared with the others, we draw the logarithmic scree plot, see Fig. 5, where we also reported the Guttman–Kaiser rule (GK) and the broken-stick (Bs) approach.

[3] https://github.com/gistairc/HS-SOD.

Table 3
Results for the eye-iris recognition task. In bold the best criteria for every metric.

Criteria	k	Euclidean	Canberra	CaEu
Scree	20	80.0%	64.4%	77.7%
GK	180	**82.2%**	48.9%	**82.2%**
Bs	6	48.9%	46.7%	46.7%
HT	50	80.0%	68.9%	**82.2%**
Tot Var	180	**82.2%**	51.1%	77.8%
Cum Perc Var	50	80.0%	62.2%	80.0%
E-70	80	75.6%	60.0%	80.0%
E-90	102	80.0%	55.6%	80.0%
E-100	114	80.0%	57.8%	80.0%
E$_{SVD}$-70	116	80.0%	51.1%	75.6%
E$_{SVD}$-90	149	80.0%	51.1%	82.2%
E$_{SVD}$-100	165	80.0%	51.1%	**82.2%**
KL	12	73.3%	62.2%	64.4%
KIF	16	71.1%	64.4%	68.9%
KMIF	30	80.0%	**73.3%**	**82.2%**

The accuracy of the reconstructed image is computed by using the relative error:

$$Error := \frac{\|I - U_k \Sigma_k V_k^\top\|_2}{\|I\|_2}. \tag{14}$$

By looking at the Table 2, the best approximation is achieved by using the Guttman–Kaiser rule, since in this case a rather big number of k is chosen. In fact, in general, this is not a very good approach, since the main goal here is trying to approximate the image I by using a small number for k and still producing an image that would be visually similar to the original one. In Fig. 6 we also display the output images.

Although according to Table 2 the $70\% - 90\% - 100\%$ of the entropy computed according to Eq. (7) seemed to give a very good compromise between the selected value for k and the computed relative error, in Fig. 6, second row, the produced output looks very blurry in all three cases. Hence, by taking into account the relative error, the visual output, and the goal of choosing k small, for this task, the best criterion is BS (60 components, 5.6e−3 error), followed by KL (58 components, 5.8e−3 error), KIF (38 components, 9.5e−3 error), range-finder (100 components, 5.8e−3 error) and KMIF (31 components, 1.3e−2 error).

Remark 3.1. For this example, the total variance, as well as the cumulative percentage variance, gets already the 98% of contribution from the first singular value, due to its big magnitude compared to the others. Unfortunately, retaining only one singular value is not enough to accurately approximate the original image; indeed, as shown in Fig. 6, in the resulting image the information is completely lost.

Eye iris recognition

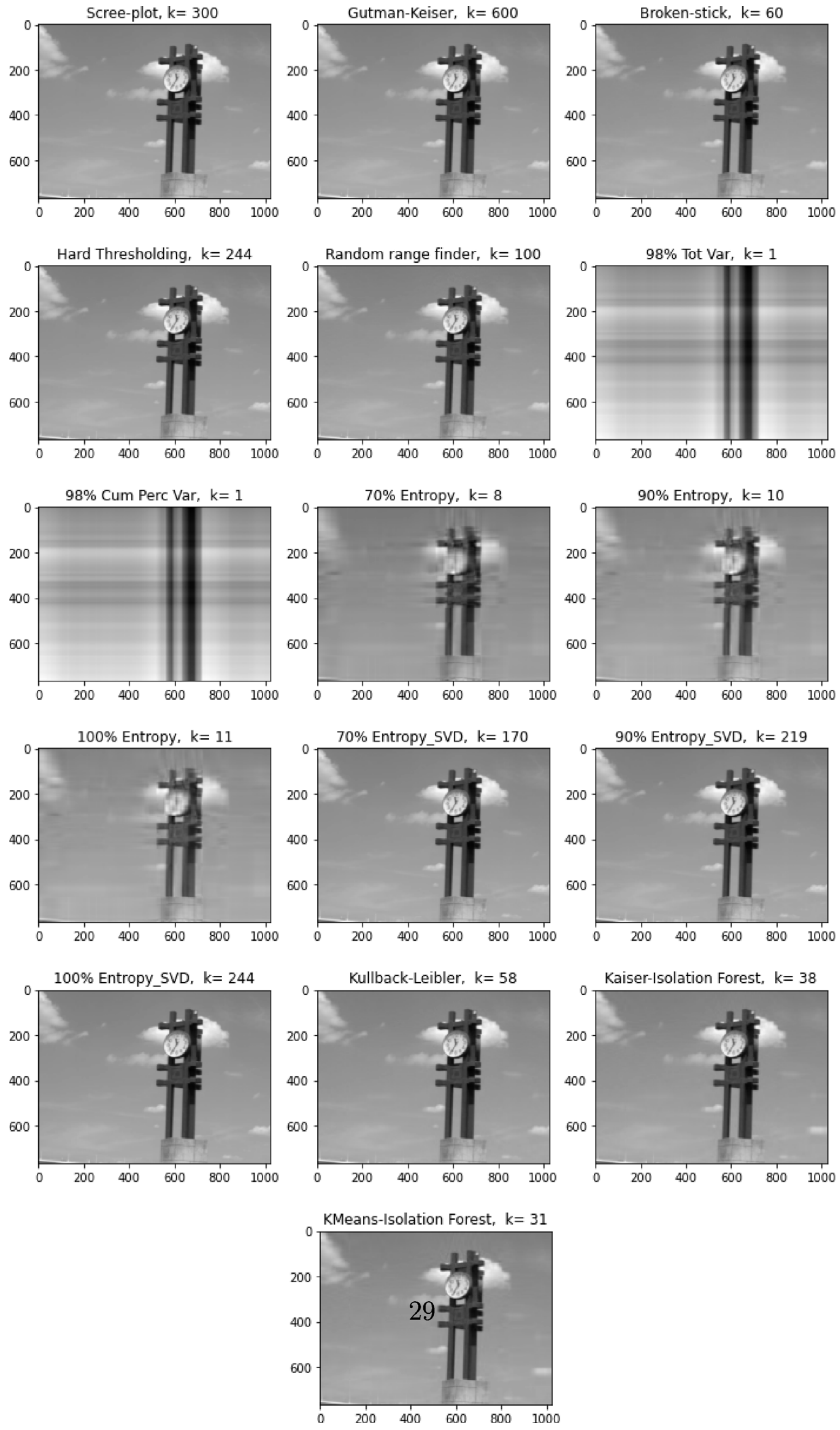
Methods of human identification using biometric features like fingerprints, face, voice, and iris are widely studied. A human eye iris has its unique structure given by pigmentation spots, furrows, and other features that are stable throughout life (see [42]). The iris can be hardly forged, replaced or copied. This makes the iris a suitable object for the identification of persons. Iris recognition seems to be more reliable than other biometric techniques like face recognition, see [43]. In the following benchmark we use the public dataset MultiMedia University (MMU).[4] The dataset contains 5 scans for every eye (i.e., left and right one) per person. For a total of 45 people. Each image is a gray-scale picture of size 240 × 320 pixels. In order to perform eye-iris recognition we select four pictures for every left eye and we collect them together to form a training set. Every image, thought as a matrix, is vectorized to become a column vector. All the training vectors are then collected together to form the term matrix of size $p \times 180$, with $p = 240 \times 320$. In Fig. 7 we show the first elements forming the training set, before vectorization.

As it can be observed by looking at Fig. 7, besides the different shapes, also the light conditions vary and some pictures are actually mirrorized. The recognition task is performed on the test set which is formed by selecting one left-eye per person. After building the term matrix, we factorize it by principal component analysis using a standard algorithm in the face-recognition context, see e.g., [44].

We briefly summarize here the applied algorithm:

1. A “mean-image” I_m is computed amongst the training set images.
2. The dataset $\Omega = I_i - I_m$ is constructed, for $i = 1, \dots, 180$ and every column is centered with respect to its mean value, obtaining $\widetilde{\Omega}$.
3. $\widetilde{\Omega}$ is projected into the coordinates system identified by the principal components (selected by varying the selection criterion).
4. For every test image I_t, the image $I_t - I_m$ is computed and the new image $\widetilde{I}_t := (I_t - I_m) - \mu$, is considered, with μ being the mean value of $I_t - I_m$.

[4] https://www.kaggle.com/naureenmohammad/mmu-iris-dataset.


Scree-plot, k= 300

Gutman-Keiser, k= 600

Broken-stick, k= 60

Hard Thresholding, k= 244

Random range finder, k= 100

98% Tot Var, k= 1

98% Cum Perc Var, k= 1

70% Entropy, k= 8

90% Entropy, k= 10

100% Entropy, k= 11

70% Entropy_SVD, k= 170

90% Entropy_SVD, k= 219

100% Entropy_SVD, k= 244

Kullback-Leibler, k= 58

Kaiser-Isolation Forest, k= 38

KMeans-Isolation Forest, k= 31

Fig. 6. Results of the image compression task.

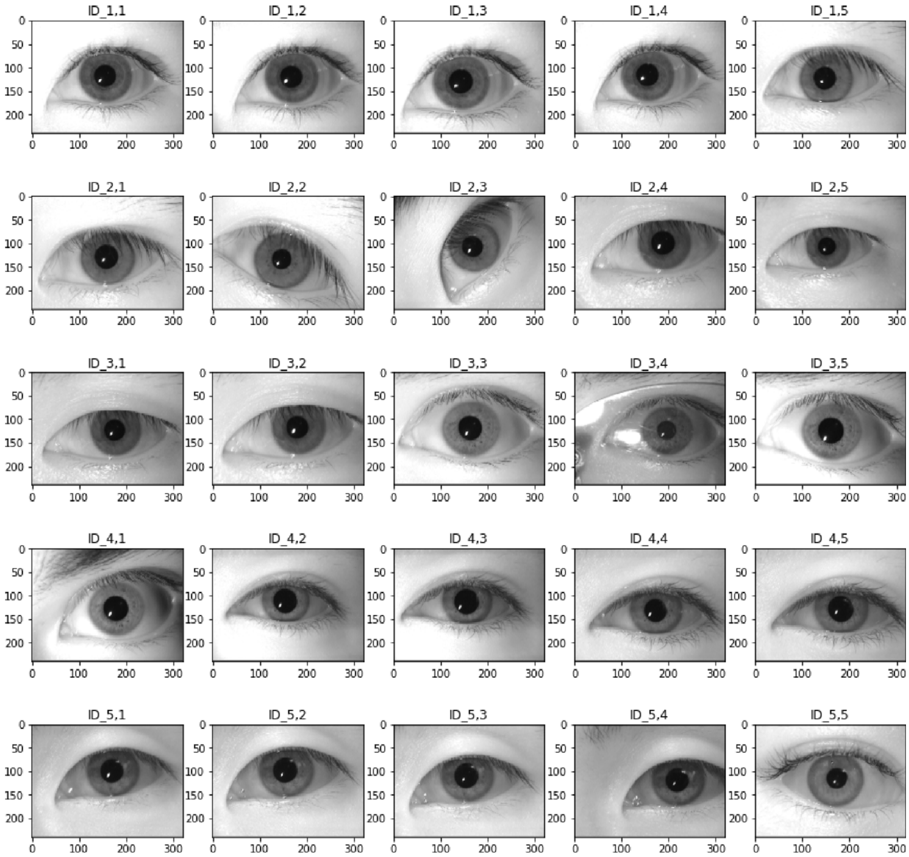


Fig. 7. The first few images stored in the training set for the eye-iris recognition task.

5. The image $\widetilde{I}_t$ is projected on the same principal components.
6. The distance between the two projections is computed.
7. The index corresponding to the smallest distance identifies the class for the considered $\widetilde{I}_t$.

For this algorithm there are two main ingredients: a suitable number of principal components should be correctly identified and a suitable metric function should be employed. In particular, looking at the results reported in Table 3, we see how by using the classical Euclidean distance (Eu), the criteria adopting a large number of components are able to correctly classify a bigger number of images. This is in line with the theory: the more components are used, the closer we get to the actual rank of the problem. With respect to this distance, the best criteria are GK and Tot Var, which are using all the 180 components. On the other hand, the main goal here is to correctly classify the images by adopting a small number of components. By using a more sensible metric, like the Canberra one, we see how the criteria that use a large number for k, as well as the ones using a too small number for k, are penalized. With this metric, the best criterion is the KMIF which hits a 73% of correctly classified images. In order to obtain the results in line with the theory and at the same time, in line with the aim to keep k small, we merge the two metrics into a new one, firstly introduced here. More in details, given two vectors $x, y \in \mathbb{R}^n$, the Canberra distance (Ca) is defined as,

$$Ca(x, y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}.$$

Then, the new metric denoted as $CaEu$ is introduced:

$$CaEu(x, y) = \min\{Eu(x, y), Ca(x, y)^4\}.$$

According to the distance $CaEu$, the best results, in terms of correctly classified images, are obtained with GK, HT, E$_{SVD}$-100 and KMIF. The GK criterion uses all the 180 components and E$_{SVD}$-100 uses 165 components. Hence, the best method is KMIF that uses only 30 components and gives the same percentage of correctly classified images. It is worth mentioning that for this task the range finder algorithm could not be successfully ran due to the size of the term matrix. This type of issue has been mentioned by the authors themselves, together with other technicalities that should be taken into account according to the assayed matrix, see [15].

It is also worth mentioning that the primary goal of this review is to compare the different criteria used for the k value selection and not the different algorithms of the literature about the specific task of biometric recognition. Although other more advanced

Table 4
Results for the noise removal task.

Criteria	k	Error
Scree	20	0.27
GK	300	0.60
Bs	1	0.26
HT	2	0.23
Range finder	100	0.49
Tot Var	300	0.60
Cum Perc Var	89	0.47
E-70	77	0.45
E-90	99	0.49
E-100	110	0.50
E_{SVD}-70	197	0.57
E_{SVD}-90	253	0.59
E_{SVD}-100	282	0.60
KL	27	0.31
KIF	9	0.21
KMIF	9	0.21

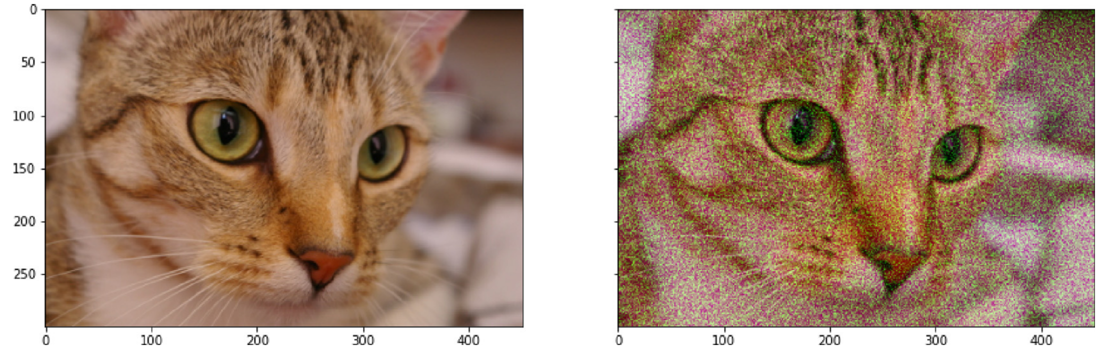


Fig. 8. Image considered for the noise removal task.

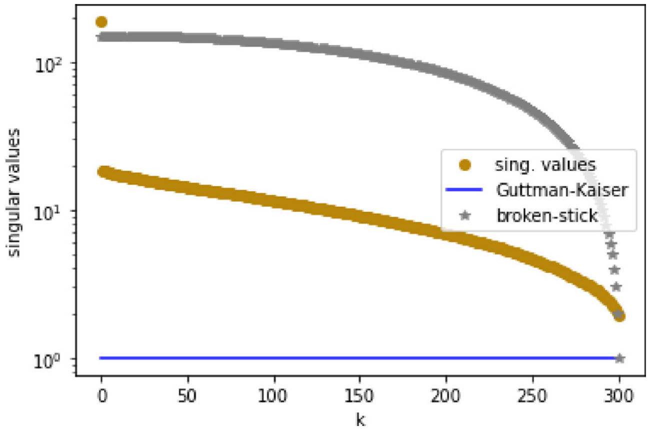


Fig. 9. Logarithmic singular values diagram for the assayed image in the task of noise removal.

techniques exist to tackle this problem, here, we are intended only to use a plain classical PCA algorithm. Therefore, we are not interested in the actual percentage of success, but we are interested in observing which criterion with a relative small k is able to get the highest percentage with the considered algorithm.

Noise removal

In this last example, we address the Gaussian noise removal task. In this case selecting the biggest k singular values is justified since the noise in the data perturbs the small eigenvalues, whereas the first k components supposedly capture the underlying structure of the data. We take one of the RGB images provided by the scikit-image gallery and we manually add some Gaussian noise into the green channel, see Fig. 8. Then, the SVD is performed on the channel affected by the noise and k is selected by varying the

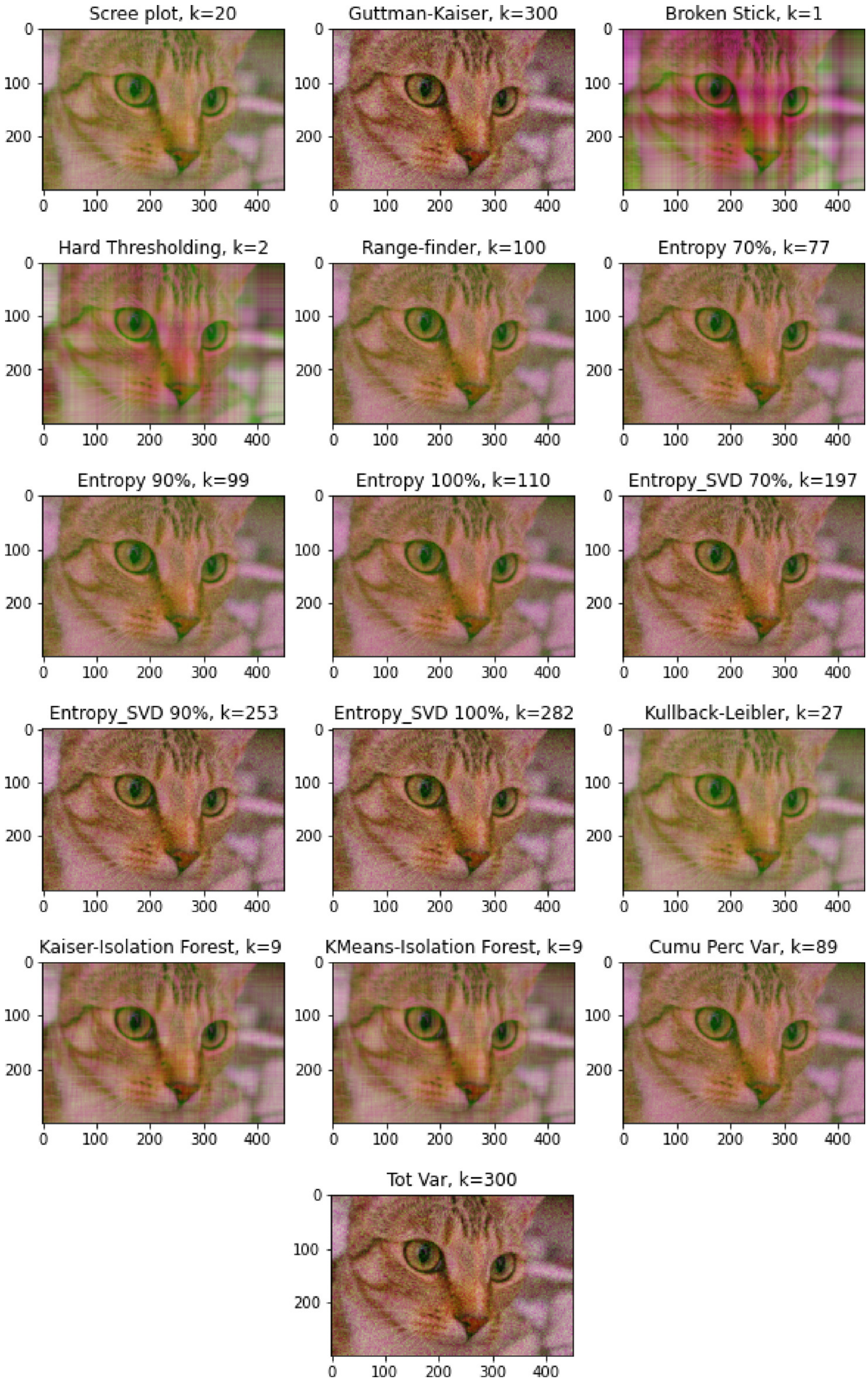


Fig. 10. Results for the noise removal by choosing k with different criteria.

chosen criterion, see Table 4. In Fig. 9 we report the logarithmic scree plot together with the Guttman–Kaiser rule and the broken stick approach. The results are displayed in Fig. 10.

By looking at Table 4 on the right most column we report also the relative error given by Eq. (14). The smallest error is achieved by the HT method that uses only 2 components. It follows Bs and KMIF. Unfortunately, by looking at the visual output in Fig. 10, it is evident how both, HT and BS fail to provide a high quality picture, as the final output looks very corrupted. Hence, in this case, the best results in terms of k value and visual output are provided by using $20 \leq k \leq 100$, i.e., Scree, Range Finder, Cum Perc Var, E-70, E-90, KL, KIF, KMIF. Bs chooses the most thrifty model with $k = 1$, but the final image looks very much corrupted (first row, third image in Fig. 10). On the contrary, KIF and KMIF have the smallest error with Error = 0.21 and $k = 9$, but the image is very smoothed-out thereby some details are lost. They are followed by HT with Error = 0.23 and 2 components and BS with Error = 0.26 with 1 component. Unfortunately, by looking at the visual output in Fig. 10, it is evident how the latter two methods fail to provide a high quality picture, as the final output looks very corrupted. Hence, in this case, the best results in terms of k value and visual output are provided by using $20 \leq k \leq 100$, i.e., Scree, Range Finder, Cum Perc Var, E-70, E-90, KL, KIF, KMIF. For the methods that use $k > 100$, i.e., GK, E-100, ESV D70, ESV D90, ESV D100, Tot Var, although certain details look better captured, a strong noise component is still visible, especially in the background of the image.

Remark 3.2 (*Final Comments*)**.** In this final remark we observe that, every time a scree plot or a logarithmic scree plot has a well identifiable elbow, the criteria based on the subjective level (e.g., Cattel's scree plot, Guttman–Kaiser rule and broken-stick) are still the best ones to be used. This unfortunately is not the standard in Big Data applications, when thousands of images should be processed at the same time. Hence, it is fundamental to rely on automatic selection criteria for the best value of k.

4. Conclusions

In this paper we reviewed the main approaches used to compute the number of singular values to retain when a truncated SVD algorithm is employed. SVD is usually adopted in many image-processing tasks, performed in an unsupervised fashion. In this review, we analyzed three typical benchmarks: image compression, image classification and noise removal. The criteria tested here are the ones which were easily implementable or for which well established libraries were already available, besides the methods based on the Kullback–Leibler divergence and anomaly detection, newly introduced here by the author.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The author is grateful to Professor Francesca Mazzia and Doctor Cristiano Tamborrino for the many fruitful discussion. This work is funded by PON Project AIM 1852414 CUP H95G18000120006 ATT 1. The author is also a member of the INdAM Research group GNCS.

References

[1] Stewart GW. On the early history of the singular value decomposition. SIAM Rev 1993;35(4):551–66.
[2] Golub GH, Reinsch C. Singular value decomposition and least squares solutions. In: Linear algebra. Springer; 1971, p. 134–51.
[3] Anderson E, Bai Z, Bischof C, Blackford LS, Demmel J, Dongarra J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, et al. LAPACK users' guide. SIAM; 1999.
[4] Golub GH, Van Loan CF. Matrix computations. JHU Press; 2013.
[5] Bai Z, Demmel J, Dongarra J, Ruhe A, van der Vorst H. Templates for the solution of algebraic eigenvalue problems: a practical guide. SIAM, Philadelphia; 2000.
[6] Eldén L. Numerical linear algebra in data mining. Acta Numer 2006;15:327–84.
[7] Jolliffe IT. Principal component analysis for special types of data. Springer; 2002.
[8] Pourzanjani AA, Jiang RM, Mitchell B, Atzberger PJ, Petzold LR. Bayesian inference over the Stiefel manifold via the Givens representation. Bayesian Anal 2021;16(2):639–66.
[9] Cattell RB. The scree test for the number of factors. Multivar Behav Res 1966;1(2):245–76.
[10] Guttman L. Some necessary conditions for common-factor analysis. Psychometrika 1954;19(2):149–61.
[11] Kaiser HF, Dickman KW. Analytic determination of common factors. In: American psychologist, vol. 14. AMER psychological assoc 750 First st NE, Washington, DC 20002-4242; 1959, p. 425.
[12] Efron B. Bootstrap methods: another look at the jackknife. In: Breakthroughs in statistics. Springer; 1992, p. 569–93.
[13] Lambert ZV, Wildt AR, Durand RM. Assessing sampling variation relative to number-of-factors criteria. Educ Psychol Measur 1990;50(1):33–48.
[14] Donoho DL, Gavish M. The optimal hard threshold for singular values is $4/\sqrt{3}$. 2013.
[15] Halko N, Martinsson P-G, Tropp JA. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev 2011;53(2):217–88.
[16] Achlioptas D, McSherry F. Fast computation of low-rank matrix approximations. J ACM 2007;54(2):9–es.
[17] Davenport MA, Romberg J. An overview of low-rank matrix recovery from incomplete observations. IEEE J Sel Top Sign Proces 2016;10(4):608–22.
[18] Ltaief H, Sukkari D, Guyon O, Keyes D. Extreme computing for extreme adaptive optics: The key to finding life outside our solar system. In: Proceedings of the platform for advanced scientific computing conference. 2018, p. 1–10.
[19] Horn JL. A rationale and test for the number of factors in factor analysis. Psychometrika 1965;30(2):179–85.
[20] Shannon CE. A mathematical theory of communication. Bell Syst Tech J 1948;27(3):379–423.
[21] Alter O, Brown PO, Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. Proc Natl Acad Sci 2000;97(18):10101–6.

[22] Roberts SJ, Penny W, Rezek I. Temporal and spatial complexity measures for electroencephalogram based brain-computer interfacing. Med Biol Eng Comput 1999;37(1):93–8.
[23] Sabatini A. Analysis of postural sway using entropy measures of signal complexity. Med Biol Eng Comput 2000;38(6):617–24.
[24] Varshavsky R, Gottlieb A, Linial M, Horn D. Novel unsupervised feature filtering of biological data. Bioinformatics 2006;22(14):e507–13.
[25] Banerjee M, Pal NR. Feature selection with SVD entropy: Some modification and extension. Inform Sci 2014;264:118–34.
[26] Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF. A review of unsupervised feature selection methods. Artif Intell Rev 2020;53(2):907–48.
[27] Suhr DD. Principal component analysis vs. exploratory factor analysis. SUGI 30 Proc 2005;203:230.
[28] Leskovec J, Rajaraman A, Ullman JD. Mining of massive data sets. Cambridge University Press; 2020.
[29] Frontier S. Étude de la décroissance des valeurs propres dans une analyse en composantes principales: Comparaison avec le modèle du bâton brisé. J Exp Mar Biol Ecol 1976;25(1):67–75.
[30] Legendre P, Legendre L. Numerical ecology. Elsevier; 2012.
[31] Jackson DA. Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches. Ecology 1993;74(8):2204–14.
[32] Minka T. Automatic choice of dimensionality for PCA. Adv Neural Inf Process Syst 2000;13:598–604.
[33] Kass RE, Raftery AE. Bayes factors. J Amer Statist Assoc 1995;90(430):773–95.
[34] Zhu M, Ghodsi A. Automatic dimensionality selection from the scree plot via the use of profile likelihood. Comput Statist Data Anal 2006;51(2):918–30.
[35] Kullback S, Leibler RA. On information and sufficiency. Ann Math Stat 1951;22(1):79–86.
[36] Liu FT, Ting KM, Zhou Z-H. Isolation forest. In: 2008 Eighth Ieee international conference on data mining. IEEE; 2008, p. 413–22.
[37] Liu FT, Ting KM, Zhou Z-H. Isolation-based anomaly detection. ACM Trans Knowl Discov Data (TKDD) 2012;6(1):1–39.
[38] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine learning in python. J Mach Learn Res 2011;12:2825–30.
[39] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İ, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, and SciPy 10 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 2020;17:261–72. http://dx.doi.org/10.1038/s41592-019-0686-2.
[40] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE. Array programming with NumPy. Nature 2020;585(7825):357–62. http://dx.doi.org/10.1038/s41586-020-2649-2.
[41] Imamoglu N, Oishi Y, Zhang X, Ding G, Fang Y, Kouyama T, Nakamura R. Hyperspectral image dataset for benchmarking on salient object detection. In: 2018 tenth international conference on quality of multimedia experience (QoMEX). IEEE; 2018, p. 1–3.
[42] Daugman JG. High confidence visual recognition of persons by a test of statistical independence. IEEE Trans Pattern Anal Mach Intell 1993;15(11):1148–61.
[43] Daugman J. Statistical richness of visual phase information: update on recognizing persons by iris patterns. Int J Comput Vis 2001;45(1):25–38.
[44] Turk M, Pentland A. Eigenfaces for recognition. J Cogn Neurosci 1991;3(1):71–86.