Project 2 NLA: SVD applications

The goal of this project is to discuss three common applications of the SVD decomposition: LS problem, graphics compression and PCA.

1. Least Squares problem

Let us consider the linear *projector* (that is, a linear map $P : \mathbb{R}^n \to \mathbb{R}^n$ such that $P^2 = P$) giving the projection onto the linear subspace

$$V = \mathrm{Im}(P) = \{y \in \mathbb{R}^n \text{ for which } \exists x \in \mathbb{R}^n \text{ s.t. } y = P(x)\}.$$

If the matrix representing P is symmetric the projector P is *orthogonal.*

Given a nontrivial subspace $V \subset \mathbb{R}^n$ there exists a unique orthogonal projector P_V onto V. Given $A \in \mathbb{R}^{m\times n}$ the following properties hold:

1. $P_{\mathrm{Im}(A)} = AA^+$.
2. $P_{\mathrm{Ker}(A)} = I - A^+A$.

Now, consider the LS problem as follows: of all the vectors x which minimize $\|Ax - b\|$, which is the shortest (that is, the one with $\|x\|_2$ minimum)? That is, we look for the minimum norm solution of the least squares problem.

Note that one has $Ax \approx b$, but $b \notin \mathrm{Im}(A)$ in general. This motivates to use the orthogonal projector onto $\mathrm{Im}(A)$ and consider the problem $Ax = AA^+b$ instead. Then

$$Ax = AA^+b \Leftrightarrow A(x - A^+b) = 0 \Leftrightarrow x - A^+b \in \mathrm{Ker}(A) \Leftrightarrow \exists w \in \mathbb{R}^n \text{ s.t. } x - A^+b = (I - A^+A)w$$

hence

$$x = A^+b + (I - A^+A)w$$

where $w \in \mathbb{R}^n$ is an arbitrary vector. Taking $w = 0$ we minimize $\|x\|_2$, then

$$\boxed{x = A^+b}$$

is the LS solution with minimum norm.

Write a program to solve the LS problem using SVD. Compute the LS solution for the datasets `datafile` and `datafile2.csv` that were used in `pr4: QR factorization and least square problems`. Compare the results using SVD with those obtained from the QR solution of the LS problems.

2. Graphics compression

1. The SVD factorization has the property of giving the best low rank approximation matrix with respect to the Frobenius and/or the 2-norm to a given matrix. State properly the previous statement and write down the corresponding proofs for the Frobenius norm and the 2-norm.

2. Use the previous results to obtain a lossy compressed graphic image from a .jpeg graphic file. A .jpeg graphic file can be read as a matrix using the function `scipy.ndimage.imread()`. Use SVD decomposition to create approximations of lower rank to the image. Compare different approximations. The function `scipy.misc.imsave()` can be useful to save the approximated graphic files as .jpeg. The code must generate different compressed files for a given graphic file. Hence, to organize the output files, the name of the compressed file must reflect the percentage of the Frobenius norm captured in each compressed file. Use different .jpeg images (of different sizes and having letters or pictures) and compare results.

   Remarks:

   (a) If the image is a color one, you get the three matrices obtained from the three components (RGB) of the .jpeg file. If you only use the first column-matrix you will obtain a grey picture instead.

   (b) Even if SVD is not a very effective procedure for graphic compression (unless the image data is highly correlated) it has many applications in practical situations. For example, see [1] for applications to facial recognition. Also SVD is used for denoising images (image deblurring problem) in many contexts, see for example [2] and references therein.

3. Principal component analysis (PCA)

Main idea: Principal component analysis is a technique to detect the main components of a data set in order to reduce into fewer dimensions retaining the relevant information. Let $X \in \mathbb{R}^{m\times n}$ a data set *with zero mean*, that is, the matrix formed by n observations of m variables (or observables). Below we denote the m variables as $x_1, \dots, x_m$. The elements of X are denoted as usual by x_{ij} meaning that it contains the value of the observable i of the j-th observation experiment.

A *principal component* is a linear combination of the variables so that maximizes the variance. More concretely, one looks for a combination

$$z_{1,j} = a_{1,1}x_{1,j} + \cdots + a_{1,m}x_{m,j}, \quad j = 1, \dots, n.$$

Denote by $a_1 = (a_{1,1}, \dots, a_{1,m})^t$ the vector of coefficients of the combination. These are chosen so that $\|a_1\|_2 = 1$. The variance of z_1 is given by $a_1^t C_X a_1$, where $C_X = \frac{1}{n-1}XX^t \in \mathbb{R}^{m\times m}$ is the covariance matrix[1]. Then, one selects a_1 to maximize the variance of z_1. With this choice z_1 becomes the first principal component. To obtain the second principal component, one looks for a combination

$$z_{2,j} = a_{2,1}x_{1,j} + \cdots + a_{2,m}x_{m,j}, \quad j = 1, \dots, n.$$

being $a_2 = (a_{2,1}, \dots, a_{2,m})$, $\|a_2\|_2 = 1$. One requires a_2 to maximize the variance of z_2 (i.e. maximizes $a_2^t C_X a_2$) subject to the property of being orthogonal to a_1 (i.e. $a_2^t a_1 = 0$). This gives the second principal component. One proceeds similarly to compute the other principal components. At the end, one ends up with coefficient vectors $a_1, a_2, \dots, a_n$ that provide the principal components $z_1, \dots, z_n$.

Relation with eigenvalues/eigenvectors of the covariance matrix. Let $v_1, \dots, v_p$ the non-zero eigenvectors of C_X. The set $\{v_1, \dots, v_p, e_{p+1}, \dots, e_m\}$ becomes an orthogonal[2] basis of $\mathbb{R}^m$ so that the covariance matrix C_X becomes diagonal in this basis. Note that highly correlated variables become concentrated in few components in this basis (many components become near zero). Let $\lambda_1, \dots, \lambda_p$ the eigenvalues of C_X in decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$. By renaming the eigenvectors we assume that λ_i corresponds to the eigenvector v_i of C_X. Moreover we also assume that $\|v_i\|_2 = 1$. Then $v_1 = a_1$ is the direction with maximum variance, $v_2 = a_2$ is the dimension of maximum variance subject to the orthogonal subspace to a_1, and so on. Hence,

the principal components are the eigenvectors of C_X.

The v_i direction accounts for a ratio of $\lambda_i / \sum_{j=1}^{p} \lambda_j$ of the total variance of the data set X.

Covariance vs. Correlation matrix. In the previous explanation we implicitly assumed that the observables are measured in comparable physical units. In this case one performs PCA analysis on the covariance matrix (hence one center the data for each observation by subtracting the mean of the observations for each variable). Otherwise it can be useful to standarize data (to a normal $N(0,1)$, that is, one centers the data by centering it and dividing it by the standard deviation of the observation for each variable) and look for eigenvalues/eigenvectors of the *correlation matrix* instead. Note that one assumes in this approach that the data is Gaussian distributed.

[1]For a data set with zero mean the covariance matrix is simply $C_X = \frac{1}{n-1}XX^t$

[2]Because C_X is symmetric.

Computing the eigenvalues and eigenvectors of the covariance matrix. The construction of the covariance matrix $C_X = \frac{1}{n-1} X X^t$ is highly numerically unstable. In order to avoid numerical instabilities one can use the SVD decomposition. If one considers

$$Y = \frac{1}{\sqrt{n-1}} X^t$$

then $Y^t Y = C_X$. Then, the reduced SVD decomposition of Y is

$$Y = USV^t$$

where $U \in \mathbb{R}^{n \times r}$, $S \in \mathbb{R}^{r \times r}$ and $V \in \mathbb{R}^{r \times m}$, being $r = \text{rank}(Y)$.

By definition of SVD we have the following properties:

i) The singular values s_i are such that $s_i^2 = \lambda_i$ (in decreasing order). If $\text{var}_T = \sum_i \lambda_i$ accounts for the total variance, then s_i^2/var_T accounts for the portion of the total variance in each of the principal components.

ii) The matrix V contains the eigenvectors of $Y^t Y = C_X$ as columns, hence the principal components as a function of the old variables (i.e. the coefficients of the combination, also called *loadings* in the PCA context).

iii) In the new PCA coordinates the data is given by $V^t X$.

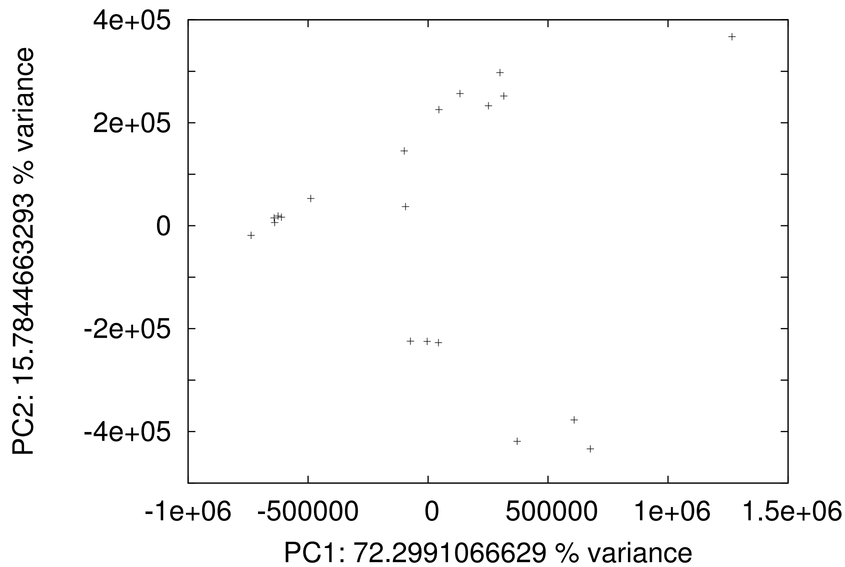
We will apply the previous PCA analysis to two different datasets.

1. The file `example.dat` contains a dataset of 16 observations of 4 variables. Perform PCA analysis using both the covariance matrix and the correlation matrix. The code must write down the portion of the total variance accumulated in each of the principal components, the standard deviation of each of the principal components and the expression of the original dataset in the new PCA coordinates.

2. The file `RCsGoff.csv` contains data from the experiment reported in [3]. Each observation consists in measuring the amount of a total number of 58581 genes. There are a total of 20 observations grouped by day of observation. The code must perform a PCA analysis on the covariance matrix. The output file must contain rows with the following format

   Sample,PC1,PC2,...,PC20,Variance

   where Sample stands for day0_rep1,...,day18_rep3 (i.e. the different observations) and PCi stands for the coordinate of the principal component of the observation. Finally variance is the portion of the total variance accumulated in each of the principal components. To compare with, below there is the plot of the first two principal components.



The memory should include a discussion about the number of principal components needed to explain the data sets (using for example the Kaiser rule, Scree plot, the 3/4 of the total variance rule, or any other method; explain the main idea of the methods used).

References

[1] N. Muller, L. Magaia and B.M. Herbst. Singular Value Decomposition, Eigenfaces, and 3D reconstruction. SIAM review, 46(3):518–545. `https://epubs.siam.org/doi/pdf/10.1137/S0036144501387517`

[2] T. Workalemahu. Singular Value Decomposition in Image Noise Filtering and Reconstruction. Thesis, Georgia State University, 2008. `https://scholarworks.gsu.edu/cgi/viewcontent.cgi?article=1051&context=math_theses`.

[3] M. Sauvageau et al. *Multiple knockout mouse models reveal lincRNAs are required for life and brain development.* eLife 2013;2:e01749, December 31, 2013. `http://dx.doi.org/10.7554/eLife.01749`.