EE512 – Applied Biomedical Signal Processing

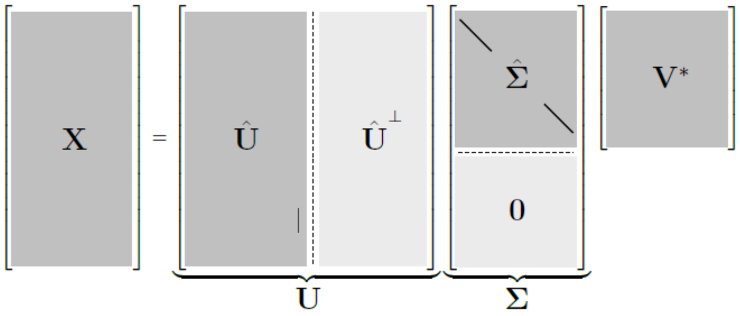
Singular Value Decomposition

Guillaume BONNIER
CSEM Signal Processing Group

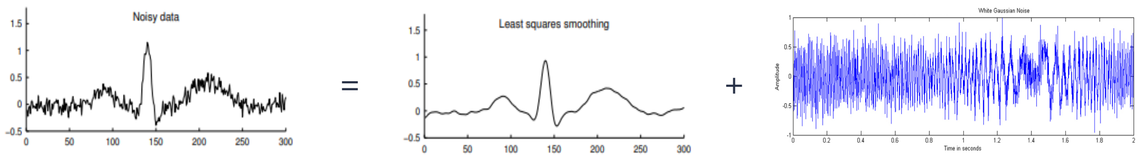
EPFL csem

Outline

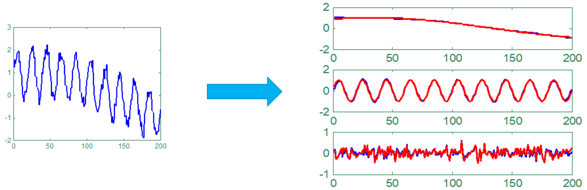
- Singular Value Decomposition overview



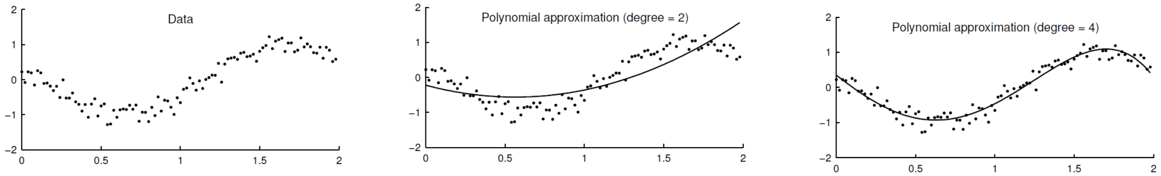
- Mathematical overview
- Noise reduction



- Singular spectrum analysis



- Least square solution using SVD



Introduction (1): SVD overview

- SVD is a **matrix factorization** technique which provides a systematic way to determine a **low-dimensional approximation** to high-dimensional data **in terms of dominant patterns**.

- Mathematical **mapping** to a new coordinate system, similar to Fourier transform, **based on data**. “Tailored” solution to a specific problem.

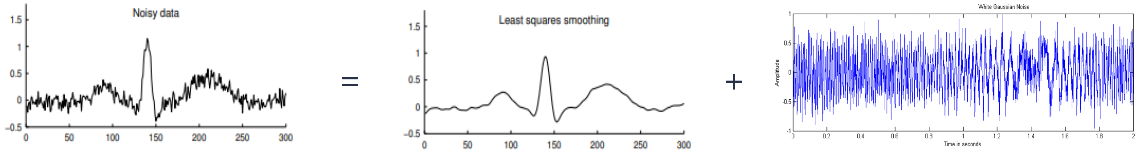
- Based on **simple** linear algebra, **interpretable and scalable**.

Introduction (3): SVD applied examples

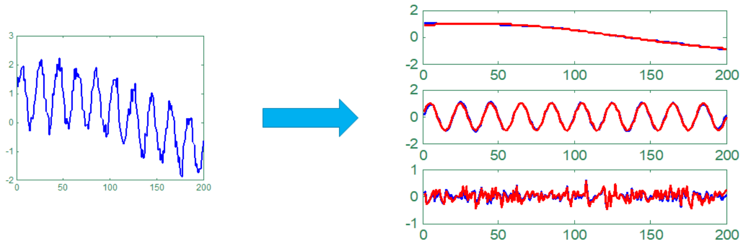
- One of the most widely used technique in **data processing**, **dimensionality reduction** and **high-dimensional statistics**.

  - **Image compression**

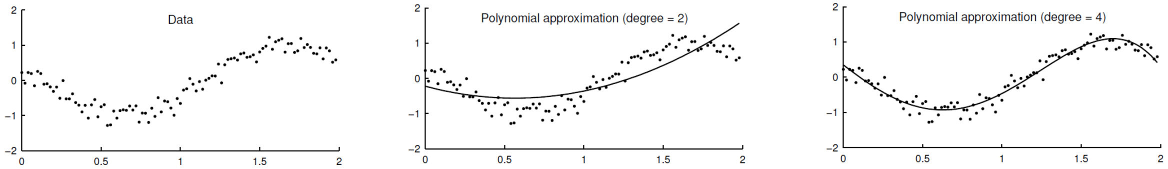
  - **Noise reduction**



  - **Filterless** signal **decomposition**



  - Robust approach for **least square** estimation problems



Mathematical overview: SVD central theorem

SVD central theorem: **any** matrix $\mathbf{X}$ of size $K \times M$ and of rank r can be decomposed as:

$$\mathbf{X} = \mathbf{USV}^T$$

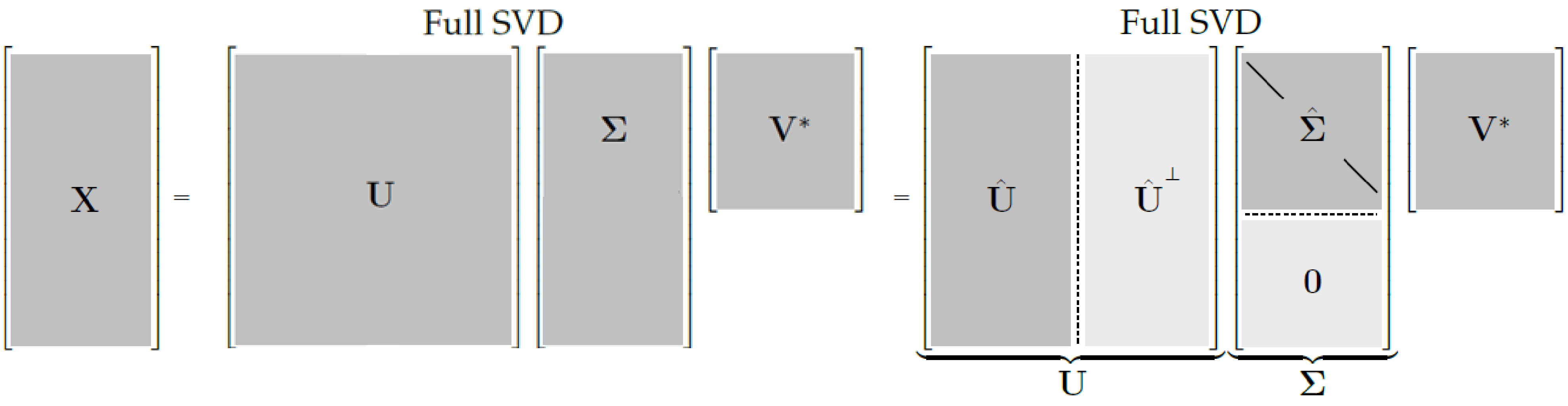
with $\mathbf{U}$ an orthogonal $K \times K$ matrix, $\mathbf{V}$ an **orthogonal** $M \times M$ matrix, and $\mathbf{S}$ a specific $K \times M$ matrix.

$\mathbf{S}$ can be seen as the generalization of a **diagonal** matrix, with:

$$\mathbf{S_{ij} = 0, i \neq j \quad and\ S_{ii} = \sigma_i > 0, i = 1, 2, \ldots, r}$$

- In Matlab:
  [U, S, V] = svd(X);
- In python:
  U, S, V = numpy.linalg.svd(X)

Mathematical overview (2): «Slim» or «economy» SVD

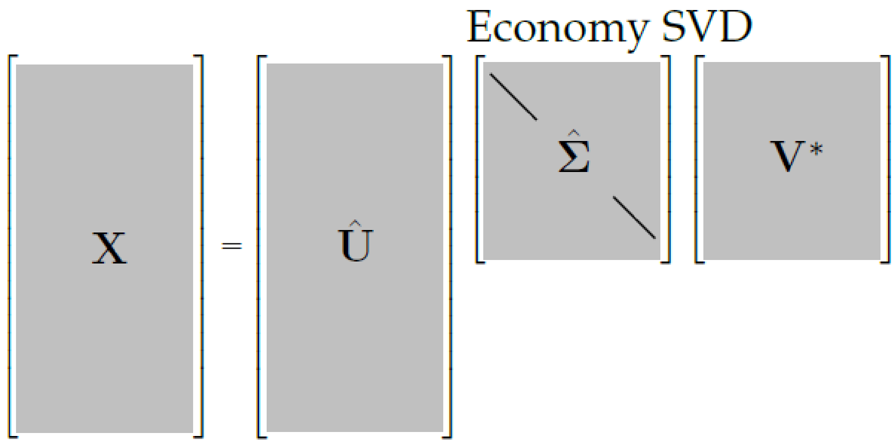


- In the product $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathbf{T}}$, the $K - r$ last lines of $\mathbf{S}$ and the $K - r$ last columns of $\mathbf{U}$ are useless because they interact only with null blocks of $\mathbf{S}$. One may use the ˝slim˝ SVD:

$$\boldsymbol{X} = \widehat{\boldsymbol{U}}\widehat{\boldsymbol{\Sigma}}\boldsymbol{V}^{\boldsymbol{T}}$$

$\widehat{\boldsymbol{U}}$ formed with the first r columns of $\mathbf{U}$.

$\widehat{\boldsymbol{\Sigma}}$ formed with first r lines of $\boldsymbol{\Sigma}$.



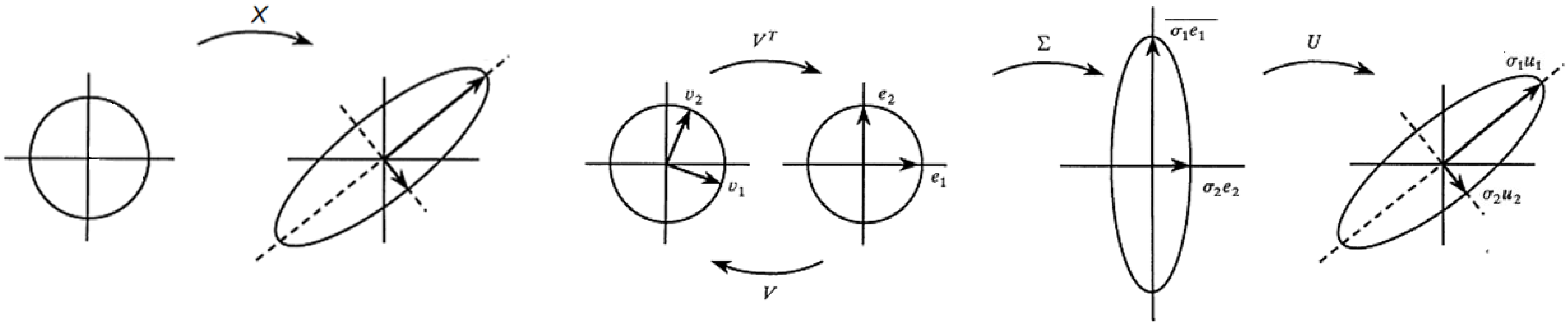
In Matlab: [U, S, V] = svd(X, ‘econ’);

In python: U, S, V = numpy.linalg.svd(X, full_matrices=False)

Mathematical overview (2): Geometrical interpretation

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$$

- U, and V are unitary matrices rotating vectors (preserve angle and length).
- S is rectangular diagonal matrix scaling vectors.

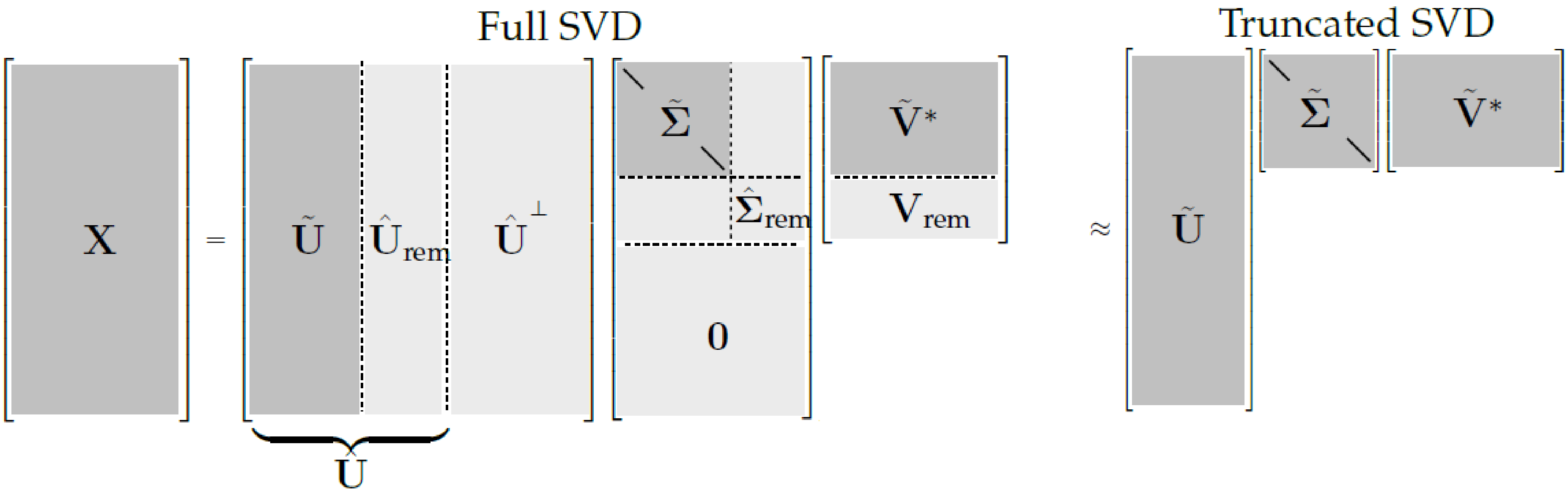


Mathematical overview (3): Dominant correlation interpretation

$$X = USV^T \Rightarrow \begin{cases} XX^T U = US^2 \\ X^T XV = VS^2 \end{cases}$$

- Columns of U are the eigenvectors of the correlation matrix XX^T
- Columns of V are the eigenvectors of the correlation matrix $X^T X$
- Columns of U are hierarchically ordered by **how much correlation they capture** in the columns of X; V similarly captures correlation in the rows of X.
- **The amount of** correlation / **information** captured by U or V component is **characterized by** their corresponding **singular values**.

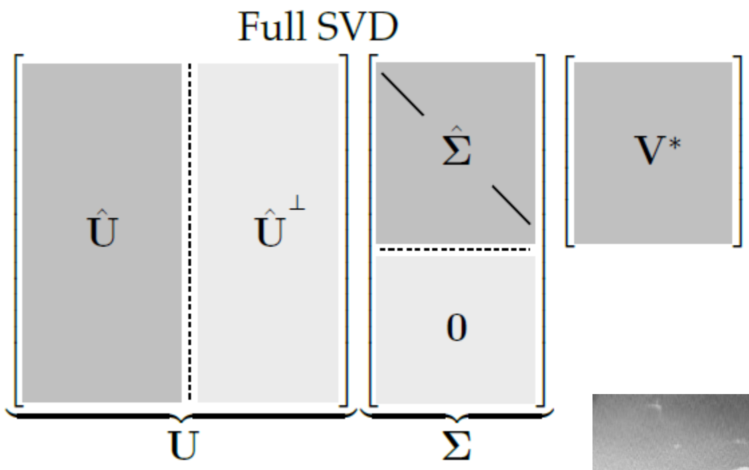
Mathematical overview (4): Low rank approximation



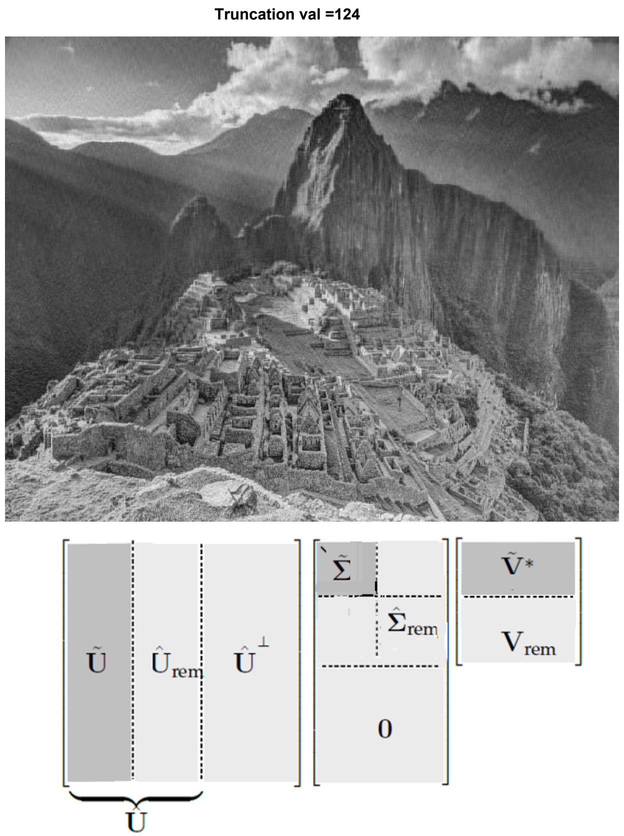
- The **SVD provides** an **optimal low-rank *p*** approximation **to the matrix X** by keeping the leading *p* singular values and vectors and discarding the rest.

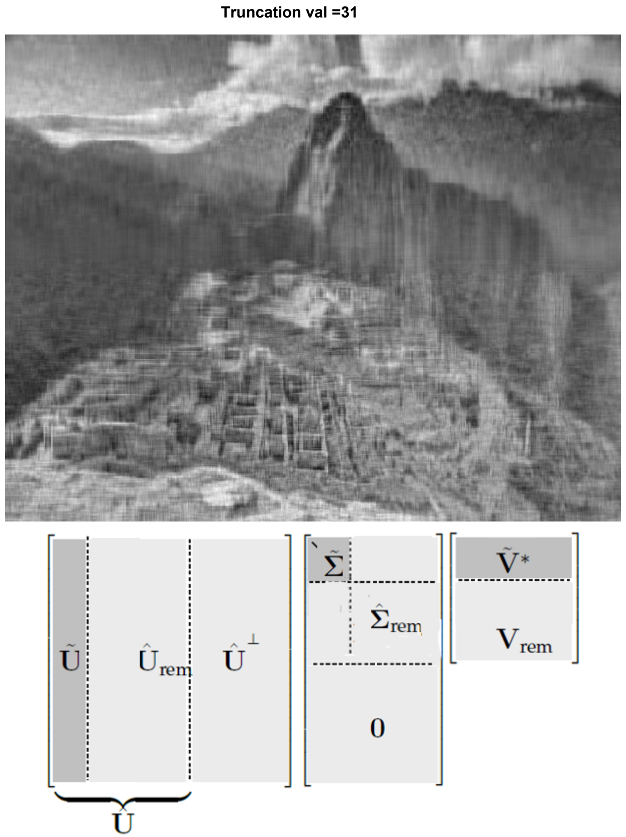
$$\tilde{X} = \sum_{i=1}^{p} \sigma_i u_i v_i^T$$

SVD application: Data compression

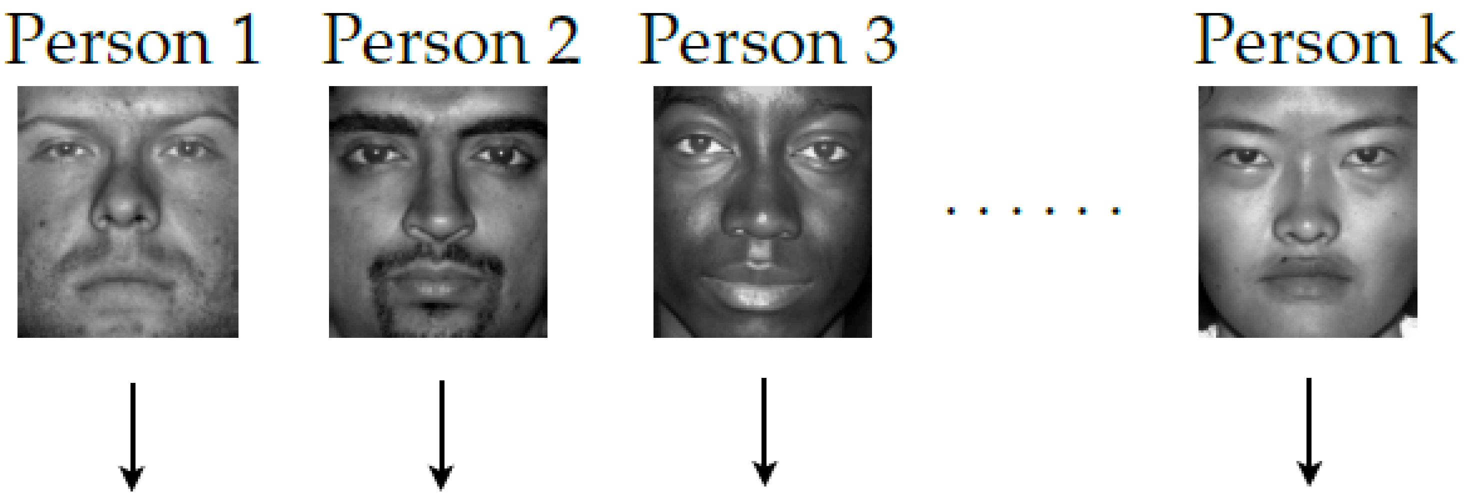



```
>> A=imread('MachuPichu.jpg'); % Load image
>> X=double(rgb2gray(A)); % Convert RBG->grayscale image.
>> [U,S,V] = svd(X); % Apply SVD
>> % Loop over  different truncation values
for r=[size(X, 1) floor(size(X, 1)/16) floor(size(X, 1)/64)]
    % Truncated image
    X_approx = Uj(:,1:r)*Sj(1:r,1:r)*Vj(:,1:r)';
    figure, imagesc(X_approx), axis off, colormap gray
    title(['Truncation val =',num2str(r,'%d')]);
end
```





SVD application: Face recognition example



$$\mathbf{X} = \begin{bmatrix} | & & | & & | & & | \\ \mathbf{x}_1 & \cdots & \mathbf{x}_{k2} & \cdots & \mathbf{x}_{k3} & \cdots\cdots & \mathbf{x}_m \\ | & & | & & | & & | \end{bmatrix}$$

http://databookuw.com/databook.pdf

SVD application: Face recognition example

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$$

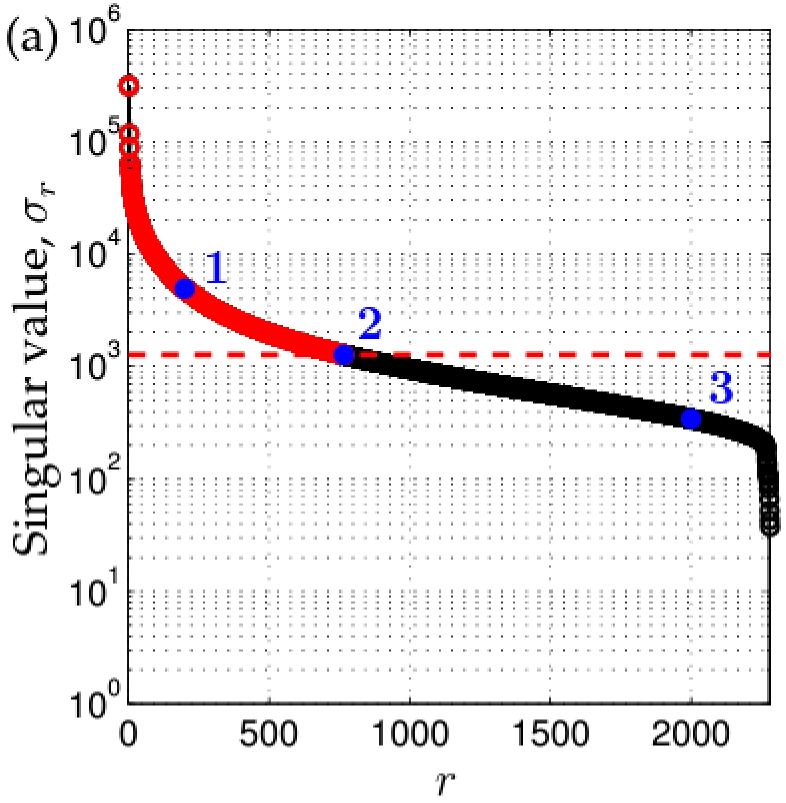
$$\tilde{\mathbf{U}} = \begin{bmatrix} | & | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \ldots & \mathbf{u}_r \\ | & | & | & & | \end{bmatrix}$$

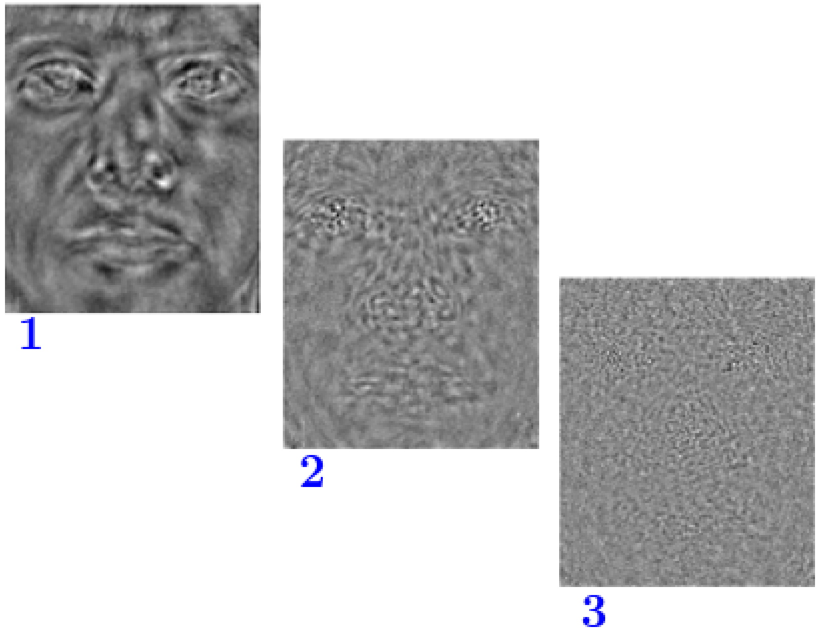
Eigenfaces

http://databookuw.com/databook.pdf

SVD application: Face recognition example

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$$

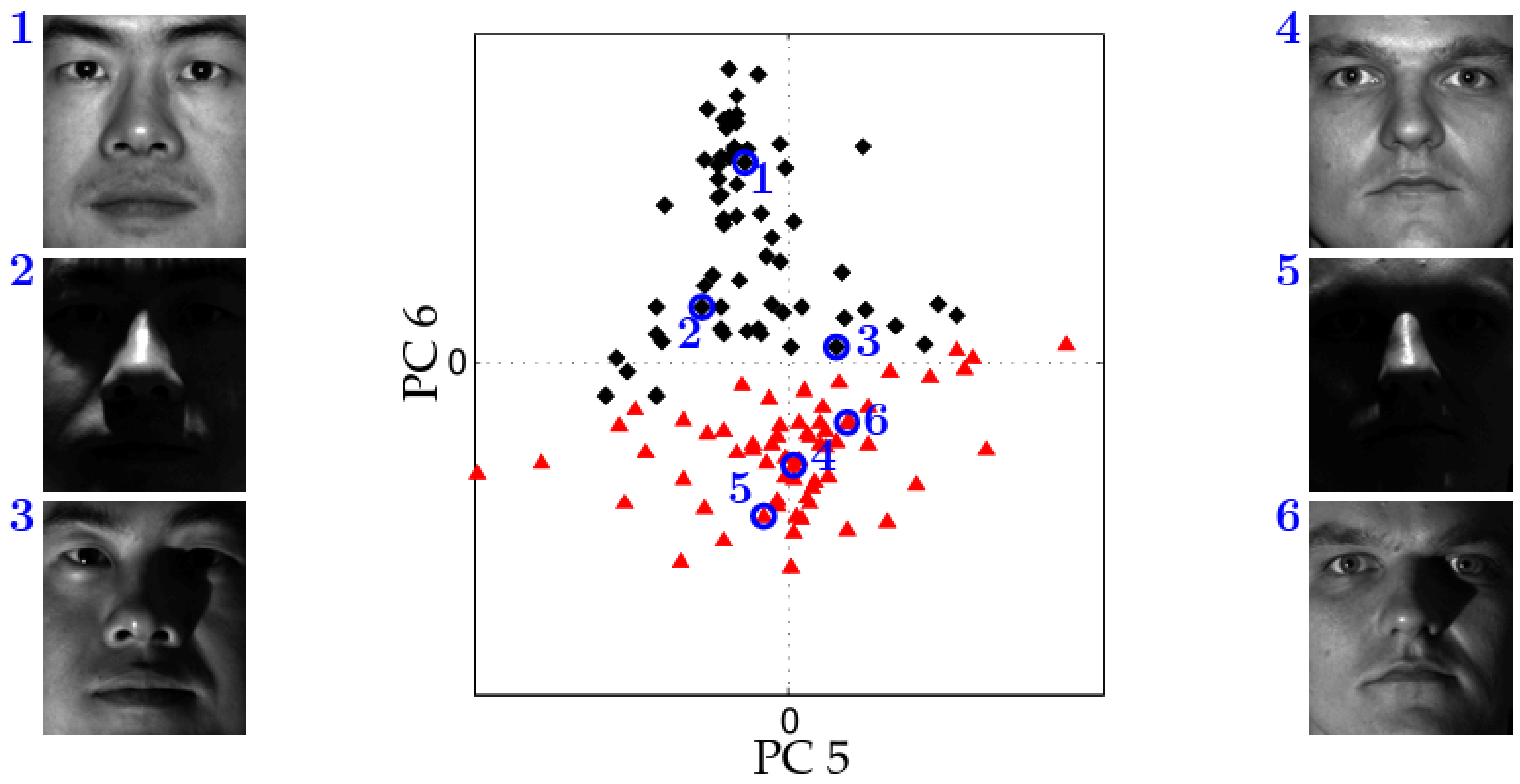




Eigenfaces

http://databookuw.com/databook.pdf

SVD application: Face recognition example

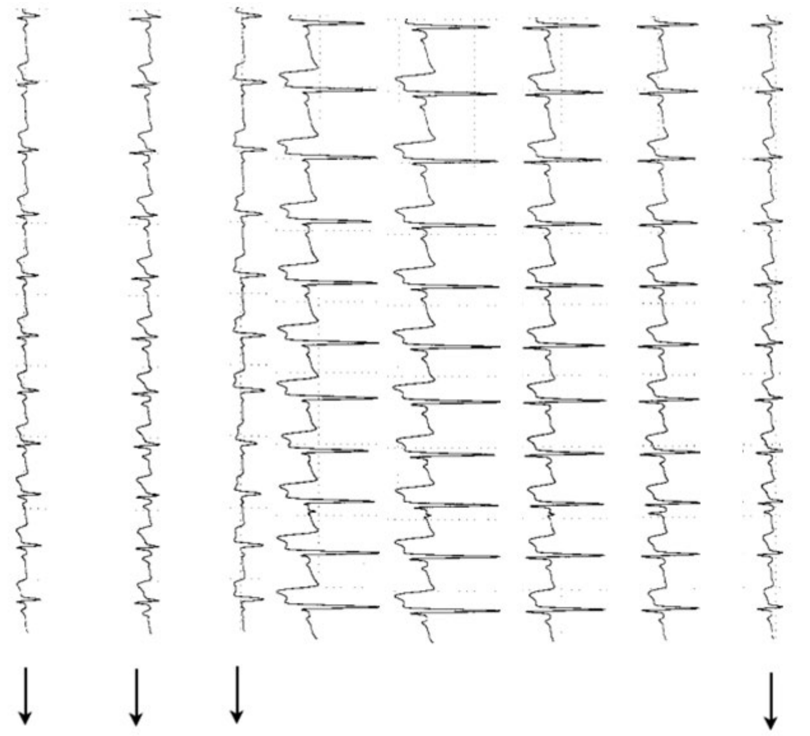


http://databookuw.com/databook.pdf

SVD application: Noise reduction

- We fill the matrix X with one lead signal of *m* samples in each column, giving X, a matrix *m* x 8.

- One observation / measurement corresponds to one lead signal of *m* samples.



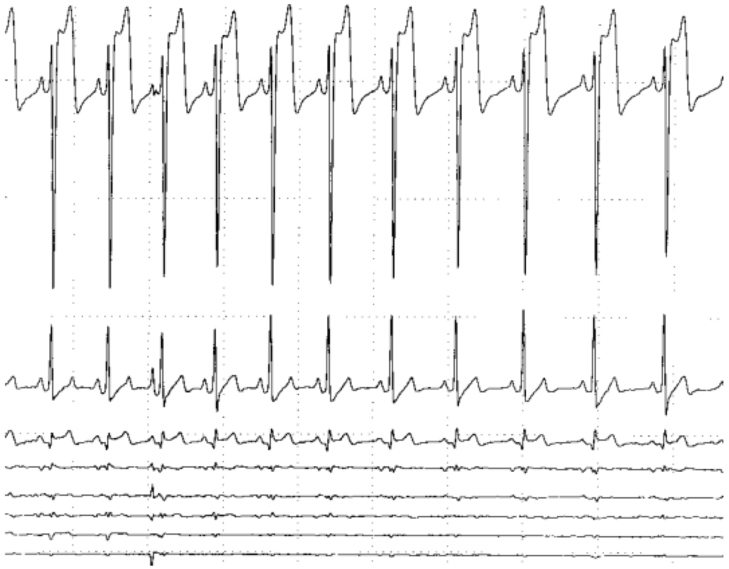
$$\mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & & x_{1,8} \\ \vdots & & & \\ \vdots & & \ddots & \vdots \\ \vdots & & & \\ x_{m,1} & & \cdots & x_{m,8} \end{bmatrix}$$

https://www.researchgate.net/publication/13194658_SVD-based_on-line_ECG_signal_orthogonalization

SVD application: Noise reduction

$$X = U\Sigma V^*$$

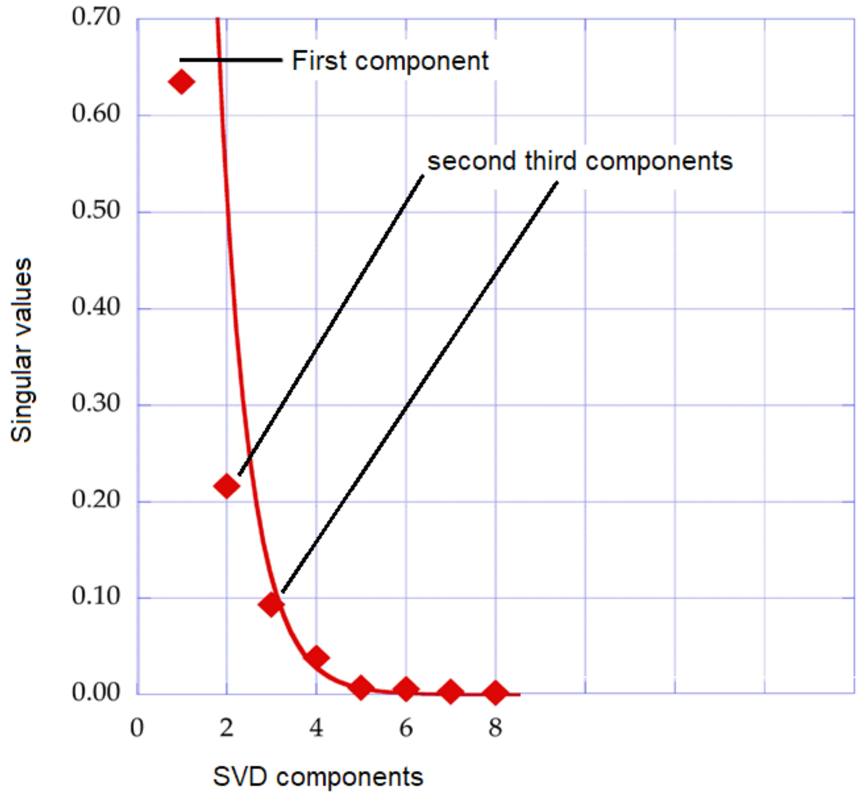
$$U^* X =$$



Projections of signal leads (matrix X)

in the SVD components (columns of matrix U)

subspace



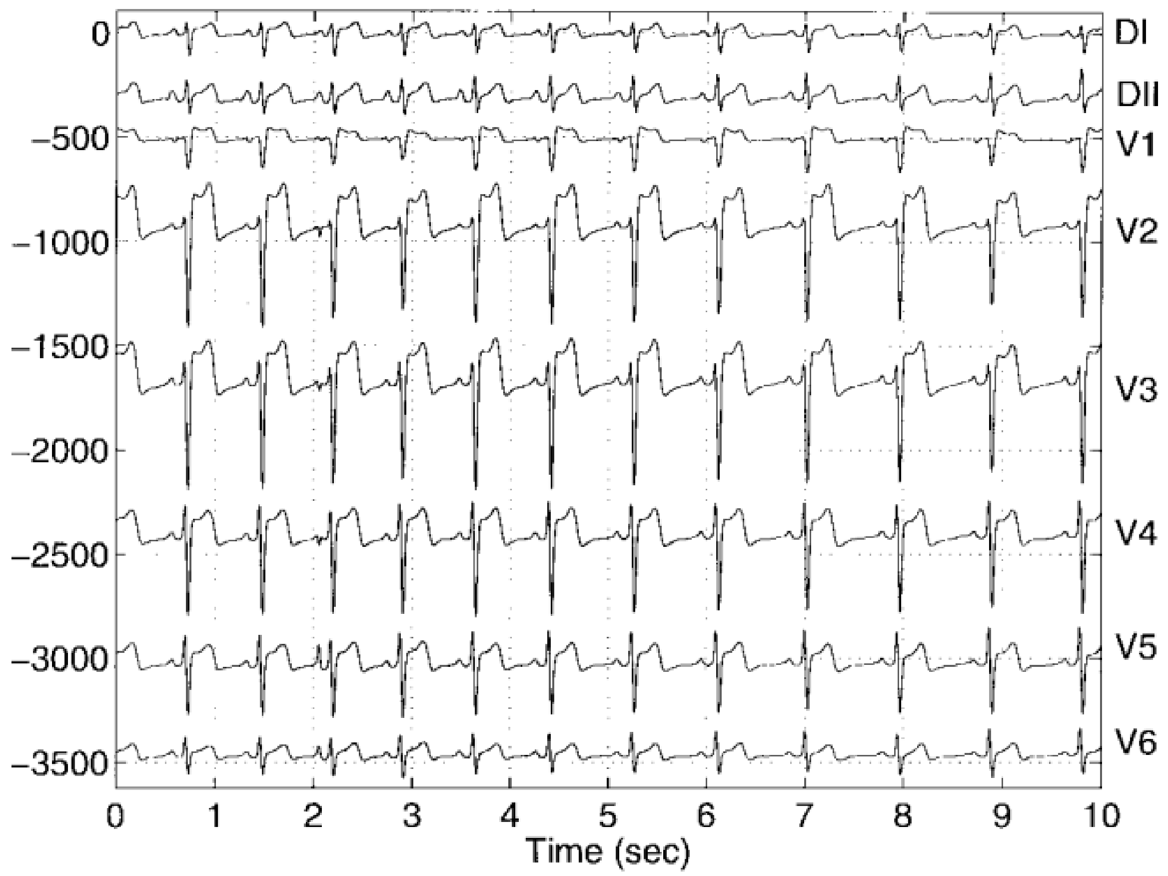
https://www.researchgate.net/publication/13194658_SVD-based_on-line_ECG_signal_orthogonalization

EPFL csem

SVD application: Noise reduction

- Example of 8-leads ECG decomposition:

ECG leads' signals at rest



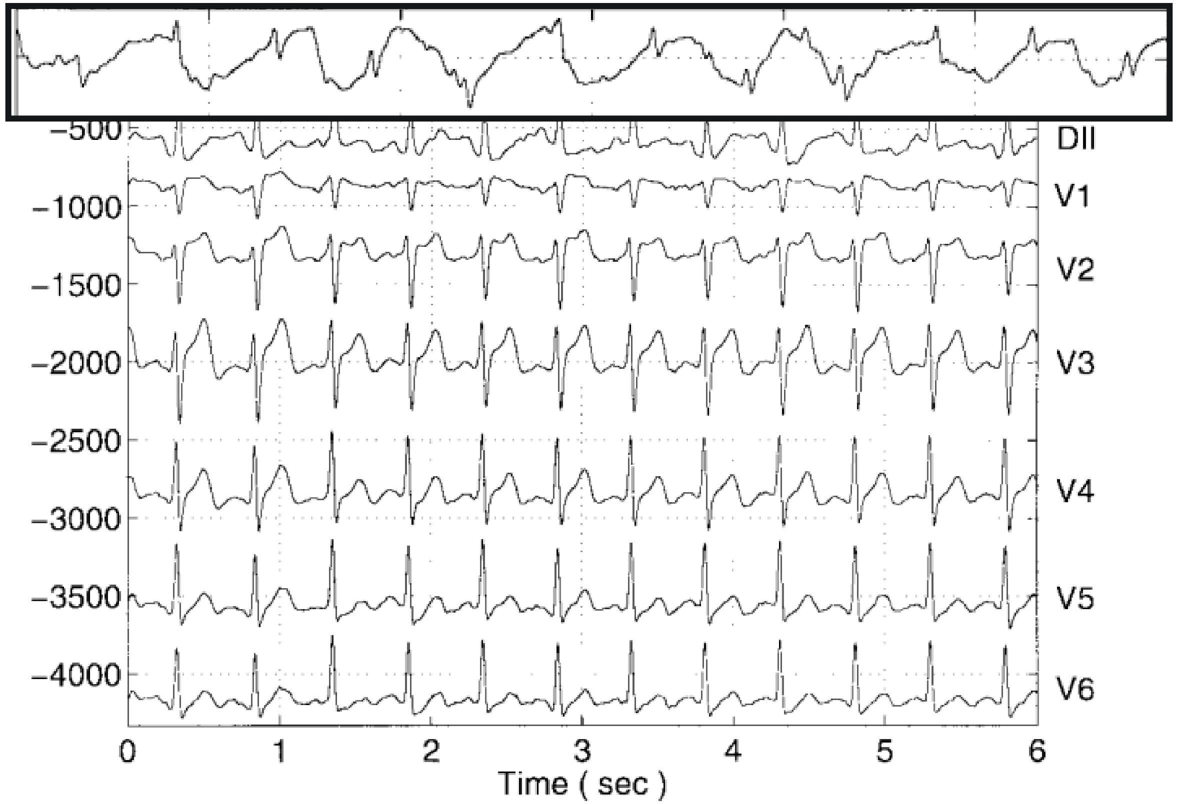
https://www.researchgate.net/publication/13194658_SVD-based_on-line_ECG_signal_orthogonalization

EPFL csem

SVD application: Noise reduction

Noise : ECG with baseline wandering (DI)

ECG leads' signals during exercise

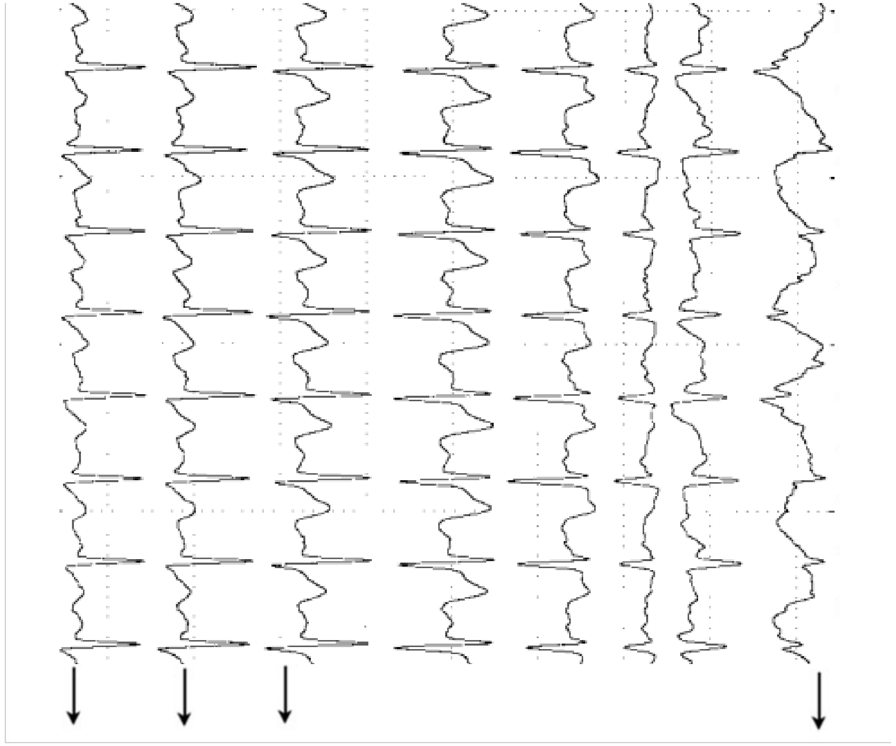


ECG leads' signals at rest

https://www.researchgate.net/publication/13194658_SVD-based_on-line_ECG_signal_orthogonalization

SVD application: Noise reduction

SVD decomposition of X: $X = USV^T$

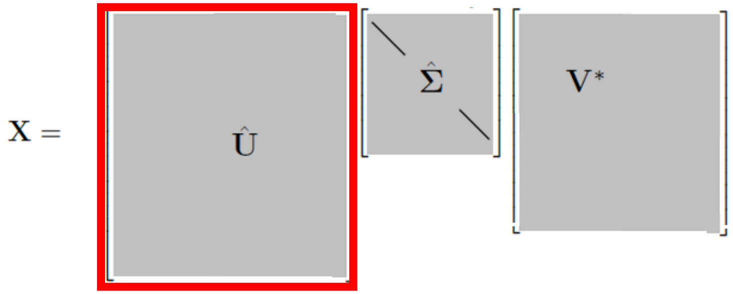


$$\mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,8} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,8} \end{bmatrix}$$

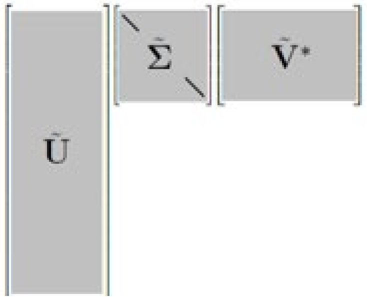
https://www.researchgate.net/publication/13194658_SVD-based_on-line_ECG_signal_orthogonalization

SVD application: Noise reduction

«Slim» SVD

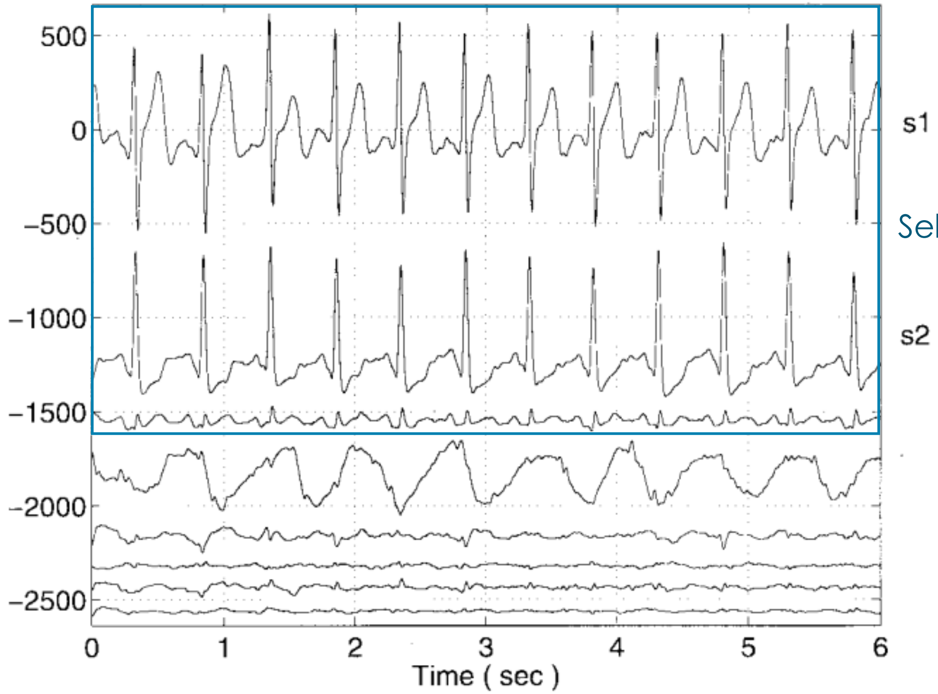


Truncated SVD

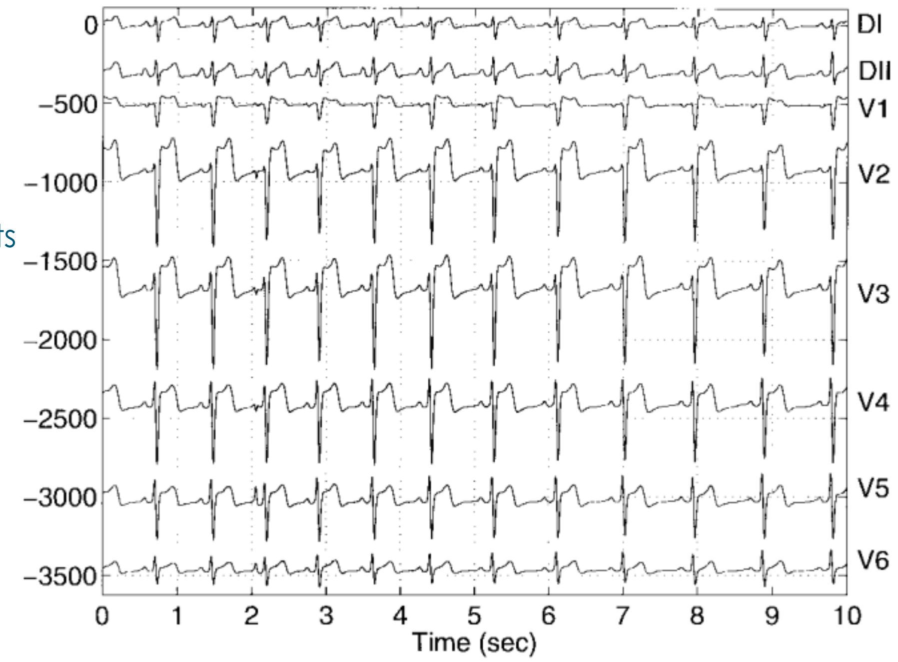


$$= \tilde{X}$$

$$U^{*}X =$$



Select 3 first components

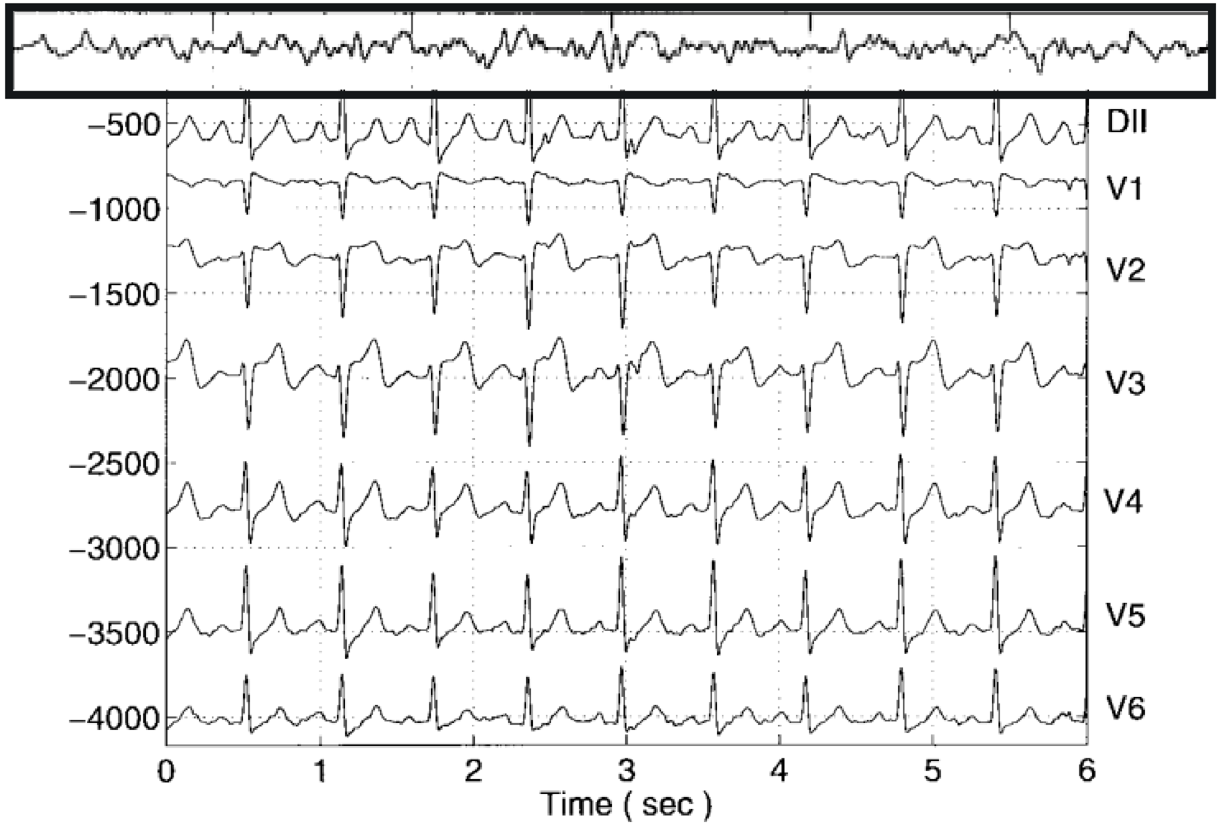


https://www.researchgate.net/publication/13194658_SVD-based_on-line_ECG_signal_orthogonalization

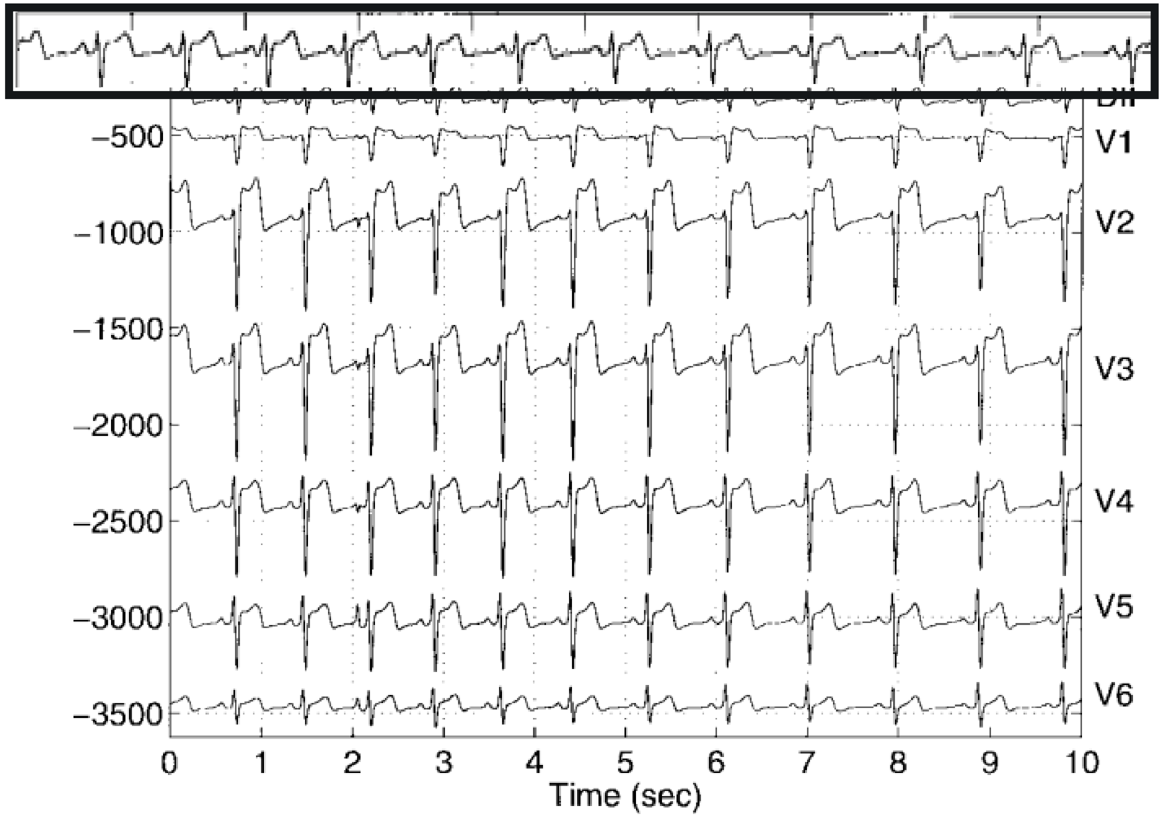
SVD application: Noise reduction

Noise : ECG with electromyogram noise

ECG leads' signals during exercise



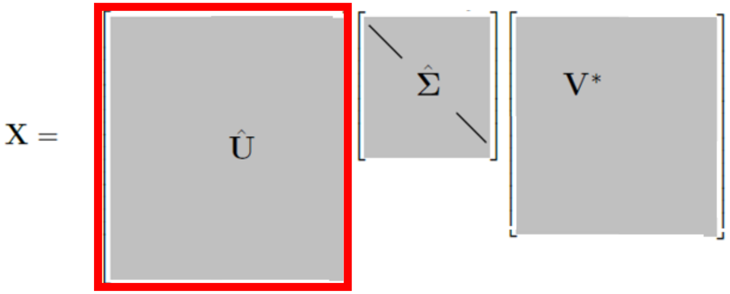
ECG leads' signals at rest



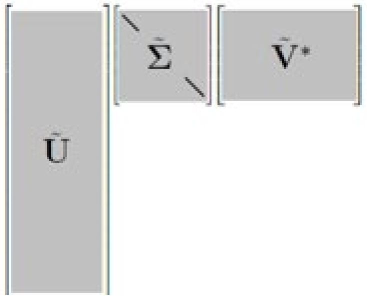
https://www.researchgate.net/publication/13194658_SVD-based_on-line_ECG_signal_orthogonalization

SVD application: Noise reduction

«Slim» SVD

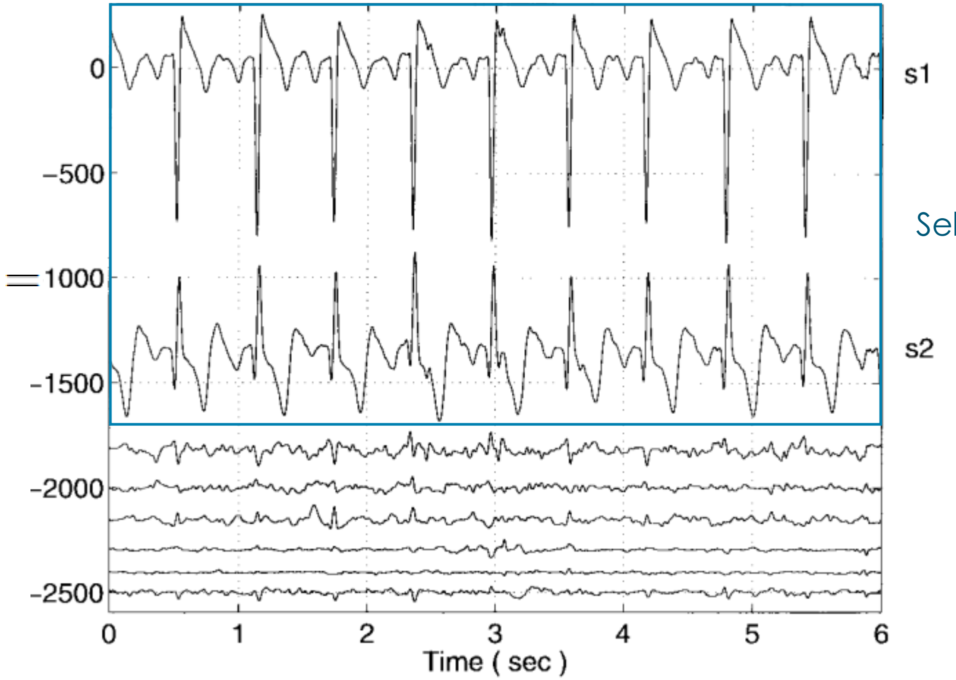


Truncated SVD

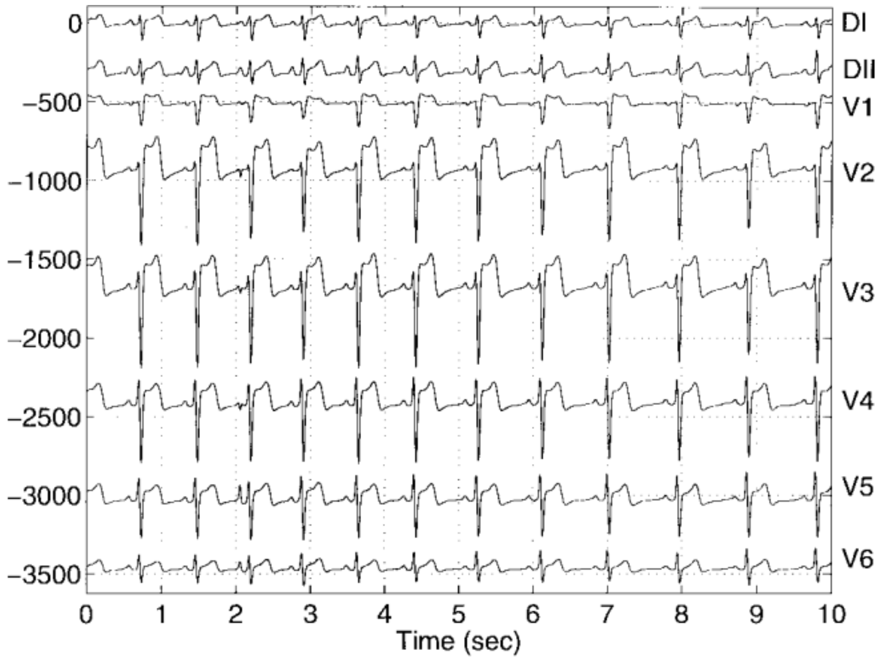


$= \tilde{X}$

$U^* X$



Select 2 first components



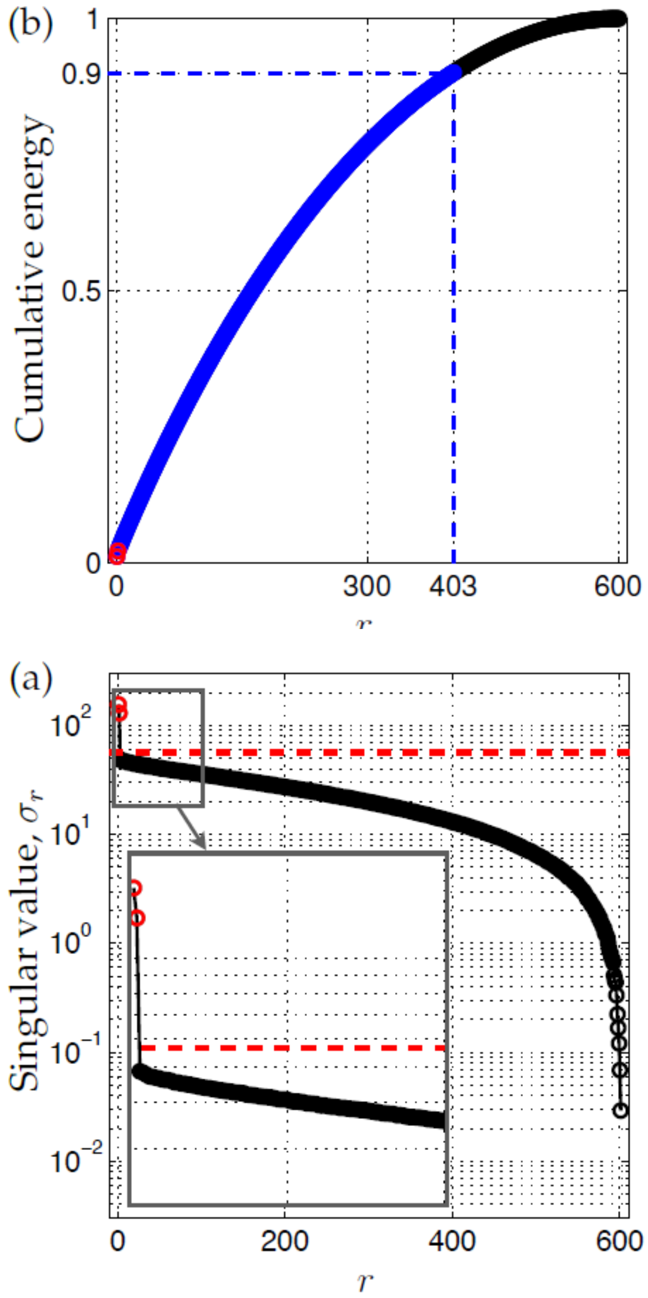
https://www.researchgate.net/publication/13194658_SVD-based_on-line_ECG_signal_orthogonalization

SVD application: Truncation techniques

- Often, one truncates the SVD at an effective rank r_e as the smallest i such that:

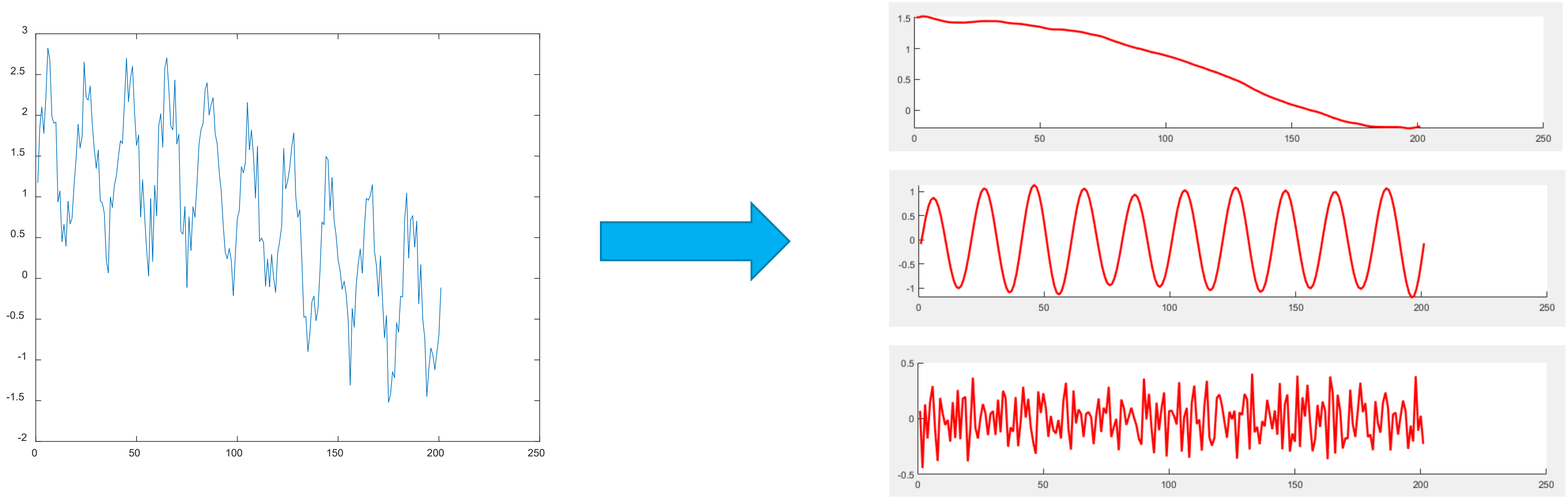
$$\frac{\sigma_1{}^2+\sigma_2{}^2+\sigma_3{}^2+\cdots+\sigma_i{}^2}{\sigma_1{}^2+\sigma_2{}^2+\sigma_3{}^2+\cdots+\sigma_r{}^2} > t$$, *with t = 0.98 for instance*

Other techniques involve identifying "elbows" or "knees" in the singular value distribution.

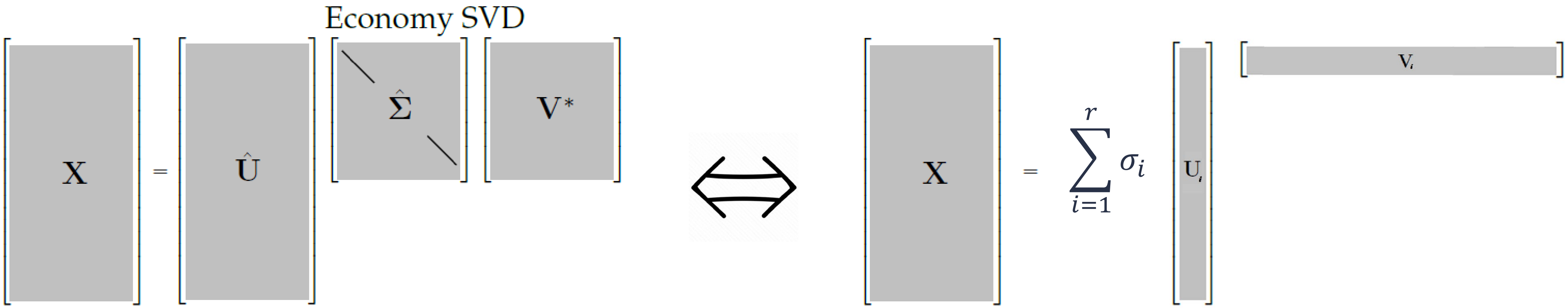


SVD application: Non parametric spectral estimation

- Singular spectrum analysis (SSA) is a principal component analysis that is very suitable for the study of one-dimensional nonlinear time-series data.
- It provides a **decomposition** of a signal **into bandpass components without** the need **to define passbands** a priori.



SVD application: Non parametric spectral estimation



- The SVD can be expressed as follow with r the rank of matrix X:

$$X = \sum_{i=1}^{M} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}$$

SVD application: Non parametric spectral estimation

- Dimensionnality reduction => Noise reduction

$$X = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ \dots + \sigma_{r-1} u_{r-1} v_{r-1}^T + \ \sigma_r u_r v_r^T$$

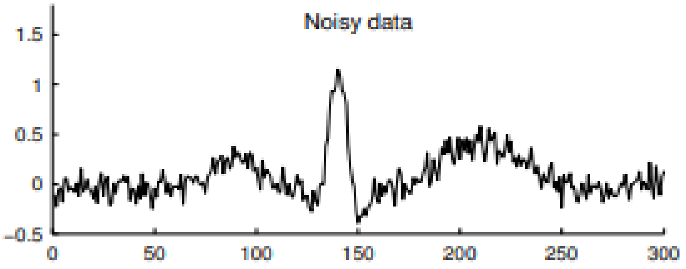
  - Each $u_i v_i^T$ is a basis direction of the matrix X, and σ_i the coordinate in this basis.

  - The amount of X pointing in $u_i v_i^T$ direction is characterized by σ_i.

  - Considering a white noise (same in all direction), the first components have higher SNR than the last components.

SVD application: Non parametric spectral estimation

- From time serie signal to a matrix:



$$X = \begin{bmatrix} x(1) & x(2) & x(3) & \cdots & x(L) \\ x(2) & x(3) & x(4) & \cdots & x(L+1) \\ x(3) & x(4) & x(5) & \cdots & x(L+2) \\ \vdots & \vdots & & & \vdots \\ x(N-L+1) & x(N-L) & \cdots & \cdots & x(N) \end{bmatrix}$$

- Matrix X has Hankel-type structure: it has the same values on all its anti-diagonals

SVD application: Non parametric spectral estimation

- Let's decompose X using SVD:

$$X = \sum_{i=1}^{L} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}$$

- The exterior products $\mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}$ correspond to matrices with the same size as X, that is (N-L+1)×L…

…However, these matrices do not necessarily have a Hankel-type structure.

SVD application: Non parametric spectral estimation

- Singular spectrum analysis central idea: Transform $\mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}$ matrices in Hankel-type matrices, with the weighted sum of the matrices equal to X.

Replace each coefficient on the anti-diagonal in each matrix by the average of the coefficients on this anti-diagonal, and one still gets X.

$$\begin{bmatrix} 5 & 10 \\ 10 & 6 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 4 & 2 \\ 4 & 3 \end{bmatrix} + \begin{bmatrix} 3 & 8 \\ 6 & 4 \\ 2 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 3 \\ 3 & 3 \\ 3 & 3 \end{bmatrix} + \begin{bmatrix} 3 & 7 \\ 7 & 3 \\ 3 & 0 \end{bmatrix}$$

SVD application: Non parametric spectral estimation

- Now the transformed matrices are Hankel-type, and signals $w_i(n)$, $i = 1$, …L, can be extracted. One has:

$$x(n) = \sum_{i=1}^{L} \sigma_i w_i(n)$$

- Note that in the resulting decomposition the component variances depend on the singular values of $\mathbf{X}$. The less $\boldsymbol{v}_i$, $\boldsymbol{u}_i$, ‘explain’ $\mathbf{X}$, the smaller the component.

SVD application: Non parametric spectral estimation

- The only parameter to select is L, the number of columns of $\mathbf{X}$. This choice is mostly based on trial-and-error.

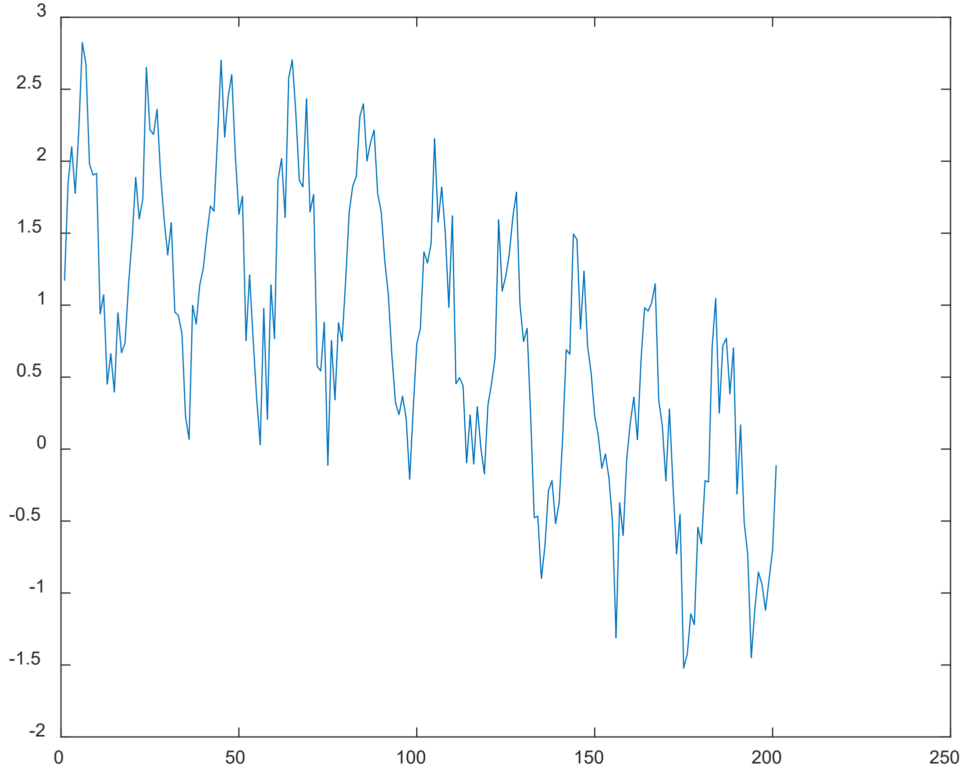
- The components $y_i(.)$ are usually presented in decreasing order of variance, which directly follows the decreasing order of the singular values.

- These components are usually bandpass signals, and their peak frequency can be estimated using non-parametric or parametric spectral estimation.

SVD application: Non parametric spectral estimation

- Example: 200-sample signal

$x(n) = \sin(2\pi * 0.0025 * n + \pi/3) + \sin(2\pi * 0.05 * n)$ + white noise $\sigma^2 = 0.04$



$$X = \sum_{i=1}^{L} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}$$

```
>>[Uhat,Shat,V] = svd(X,'econ');
```

SVD application: Non parametric spectral estimation

- *With* $L = 20$, components variance versus frequency:

$$X = \begin{bmatrix} x(1) & x(2) & x(3) & \cdots & x(L) \\ x(2) & x(3) & x(4) & \cdots & x(L+1) \\ x(3) & x(4) & x(5) & \cdots & x(L+2) \\ \vdots & \vdots & & & \vdots \\ x(N-L+1) & x(N-L) & \cdots & \cdots & x(N) \end{bmatrix}$$

SVD

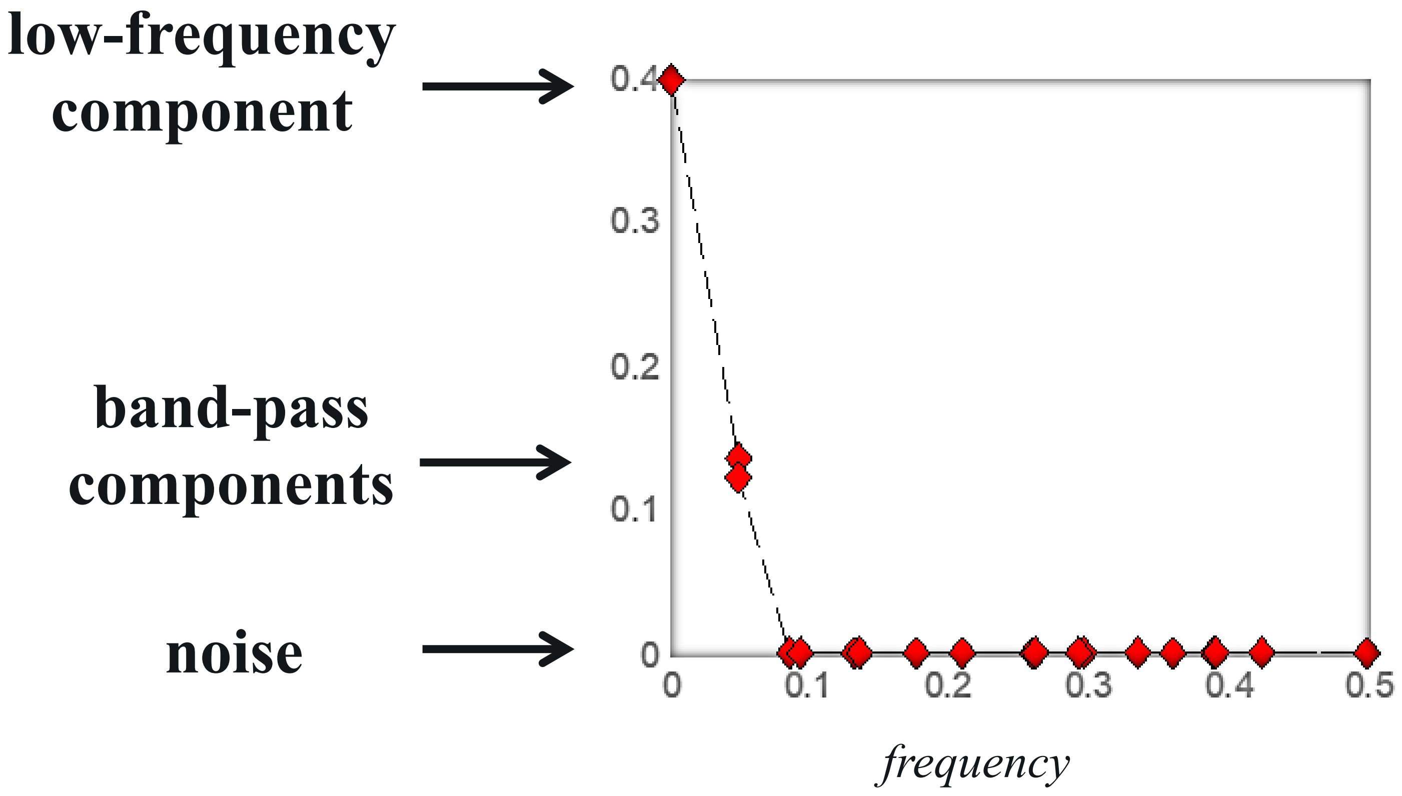
$$X = \sum_{i=1}^{L} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}$$

SSA

$$x(n) = \sum_{i=1}^{L} \sigma_i w_i(n)$$

SVD application: Non parametric spectral estimation

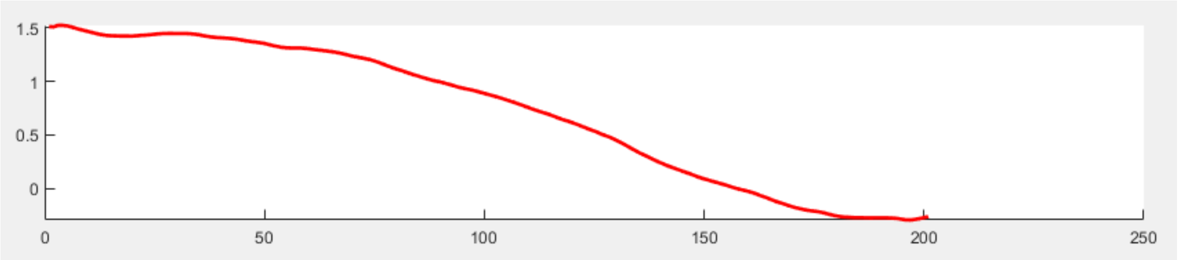
- components variance versus frequency:



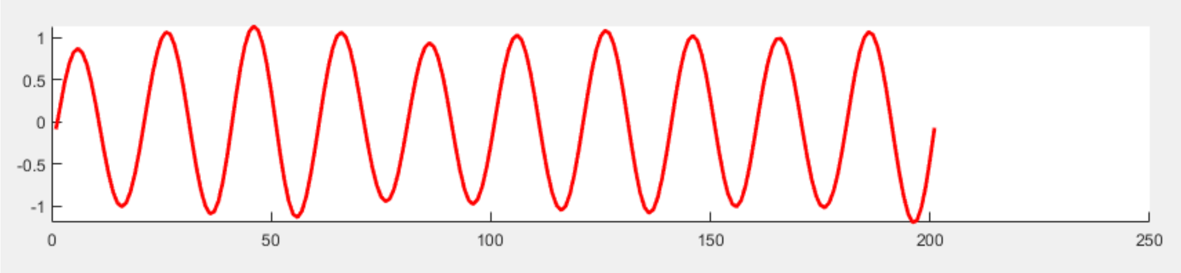
SVD application: Non parametric spectral estimation

- Decomposition: $x(n) = \sum_{i=1}^{20} y_i(n)$

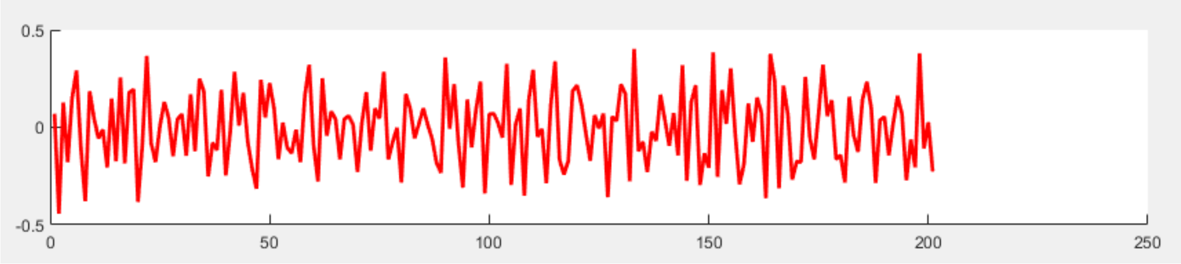
y_1 : $\sin(2\pi*0.0025*n+\pi/3)$



$y_2 + y_3$: $\sin(2\pi*0.05*n)$



$y_4 + \ldots + y_{20}$: white noise



SVD application: Least square solution

- We can use SVD to obtain low-rank approximations to matrices and to perform pseudo-inverses of non-square matrices to solve with respect to $\boldsymbol{c}$ the equation $\mathbf{X}\boldsymbol{c} = \boldsymbol{y}$ in the least-squares sense.

$$c' = \min_{c} ||Xc - y||_2^2$$

SVD application: Least square solution

$$c' = \min_{c} ||Xc - y||_2^2$$

Solution is :

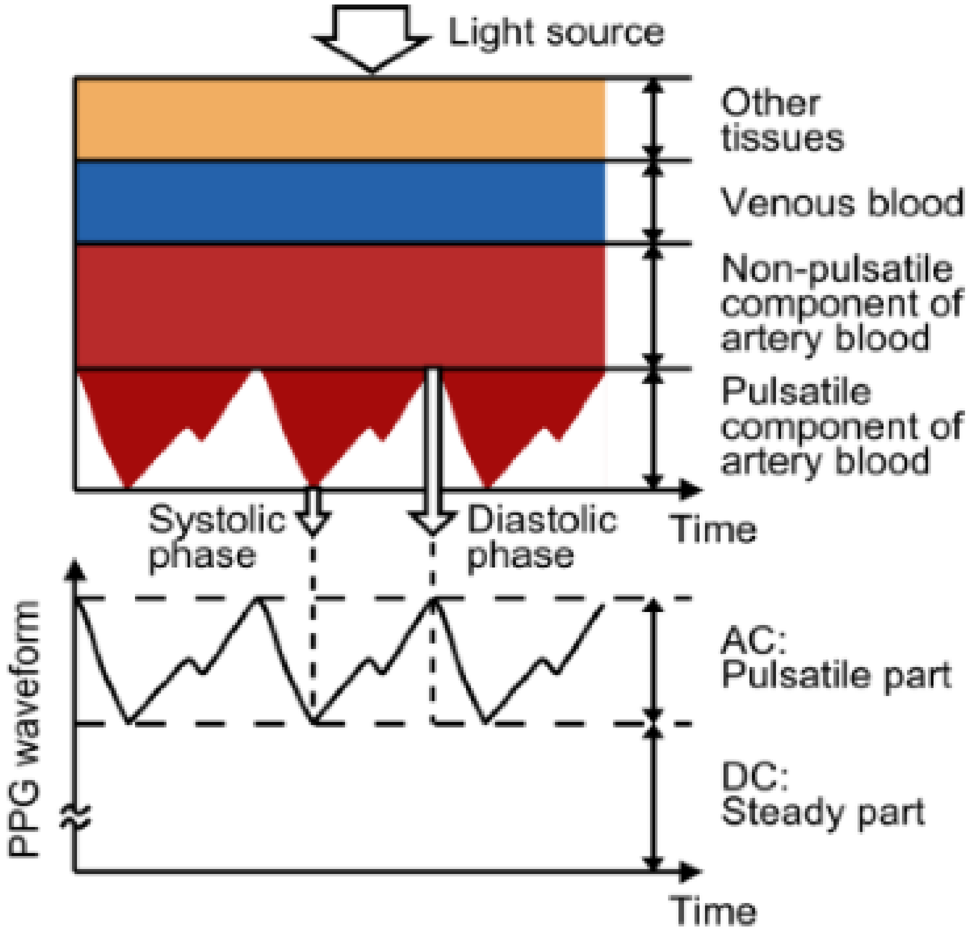
$$c' = X^{+}y$$

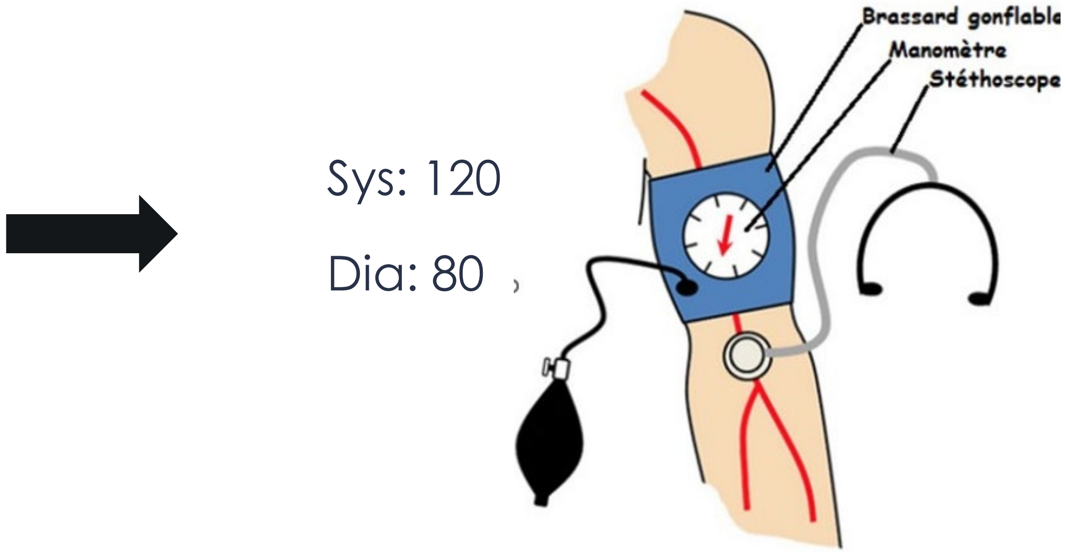
with X^{+} the pseudo-inverse of X from the exact truncated SVD:

$$X^{+} = \tilde{V}S^{+}\tilde{U} = \sum_{i=1}^{r} \frac{1}{\sigma_i} v_i u_i^T \text{ with } S^{+} = \begin{bmatrix} \tilde{\Sigma}^{-1} & 0 \\ 0 & 0 \end{bmatrix}$$

SVD application: Least square solution

- Example: Estimate blood pressure from PPG.





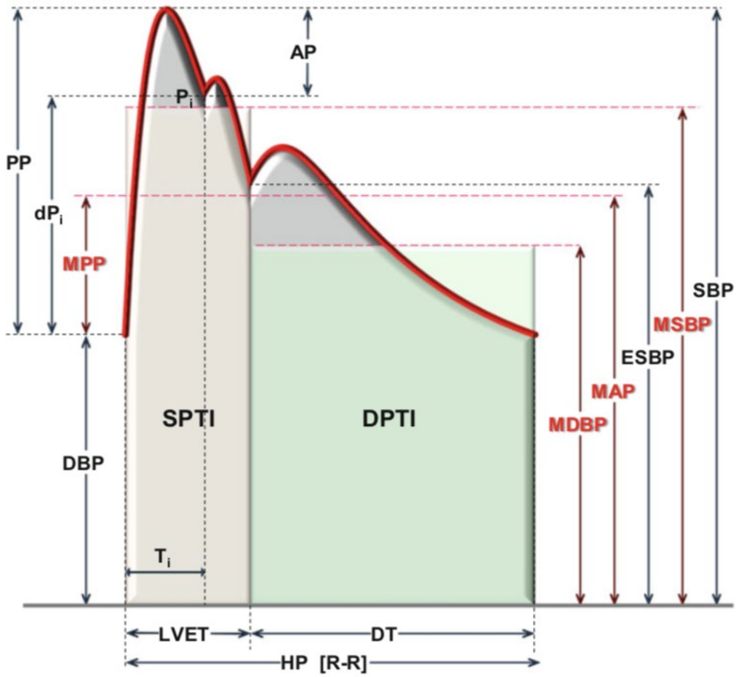
SVD application: Least square solution

- Example: Estimate blood pressure using PPG.

$$Xc = y$$

$$\mathbf{X} = \begin{bmatrix} \cdots\cdots & \mathbf{x}_1 & \cdots\cdots \\ \cdots\cdots & \mathbf{x}_{k_2} & \cdots\cdots \\ \cdots\cdots & \mathbf{x}_{k_3} & \cdots\cdots \\ & \vdots & \\ & \mathbf{x}_m & \end{bmatrix}$$

← S1: Pulse wave features



$$y = \begin{matrix} y_1 \\ \vdots \\ y_m \end{matrix}$$

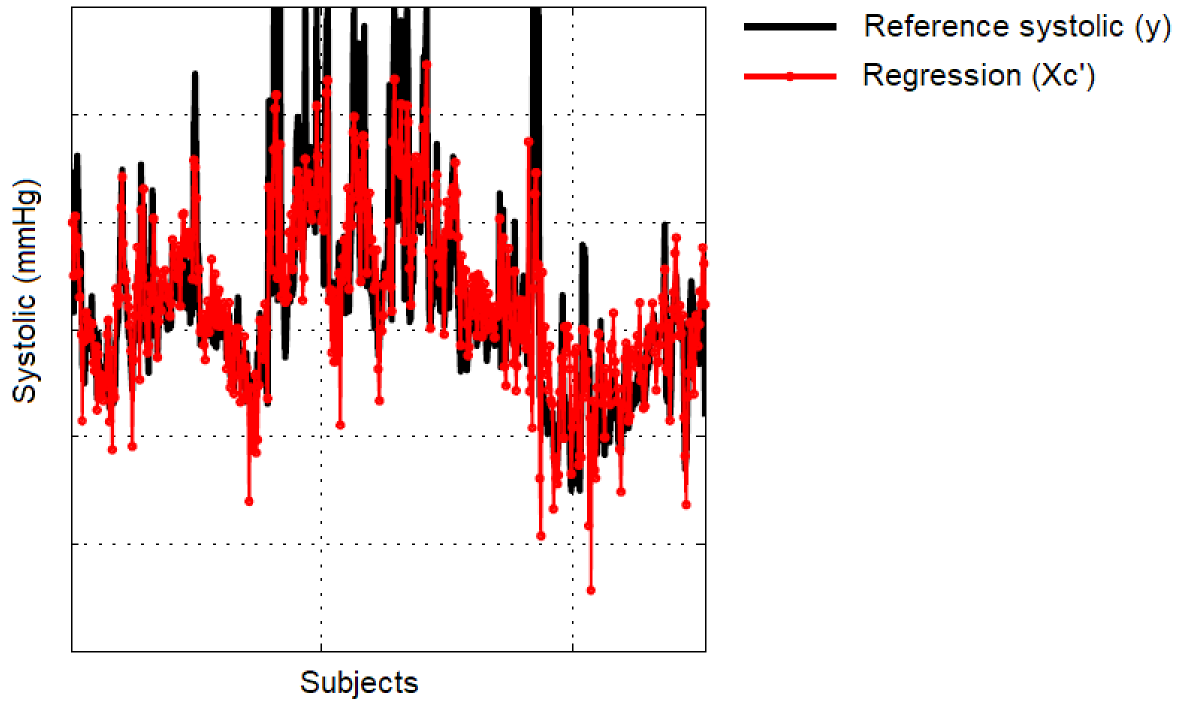
y_1 ← S1: Systolic blood pressure

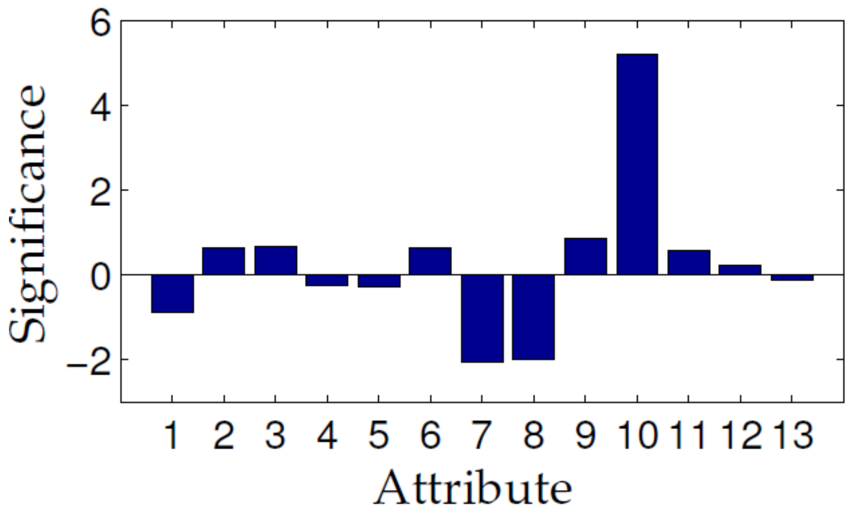
y_m ← Sm: Systolic blood pressure

SVD application: Least square solution

- Example: Estimate blood pressure using PPG.

$$c' = X^{+}y$$





SVD application: Least square solution

- In practice, what happens is that X has columns almost linearly dependent, so $X^T X$ is ill-conditioned. Theoretically, the rank of X is r = M, but one or several singular values are very small. Consequently, in the computation of:

$$X^+ = \sum_{i=1}^{r} \frac{1}{\sigma_i} v_i u_i^T$$

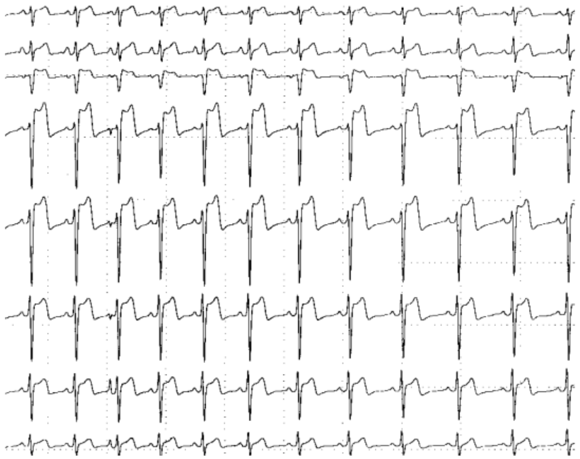
Some denominators are very small, which leads to numerical instability !

SVD data pre-processing: Data alignment

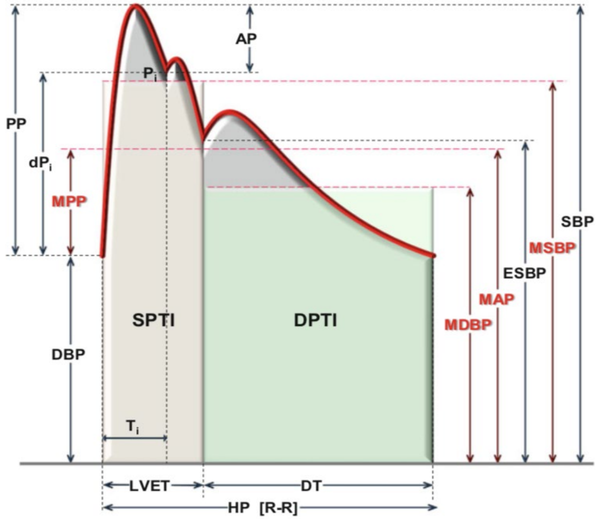
- The alignment of data significantly impacts the rank of the SVD approximation. The SVD essentially relies on a separation of variables between the columns and rows of a data matrix. In many situations, such as when analyzing misaligned data, this assumption breaks down, resulting in an artificial rank inflation.

Example 1: Face Images

Example 2: ECG signals



Example 3: Pulse wave features



SVD data pre-processing: Data normalization

- SVD is sensitive to the average value of the input signals. When the derivations contain a nonzero average SVD algorithm decomposes these as orthogonal components which increases the rank of the data matrix and dimension of the signal space.

- Normalization, i.e. all signals scaled to unit variance, suppresses effects due to amplitude differences in the signals that could affect the singular values.

Conclusions

- SVD provides systematic approach to reduce dimensionality and resolve linear system of equations.

- Truncation, one of the most important decisions when using the SVD, is based on many factors, including specifications on the desired rank of the system, the magnitude of noise, and the distribution of the singular values.

- Singular spectrum analysis (based on SVD) provides a decomposition of a signal into bandpass components without prior knowledge.

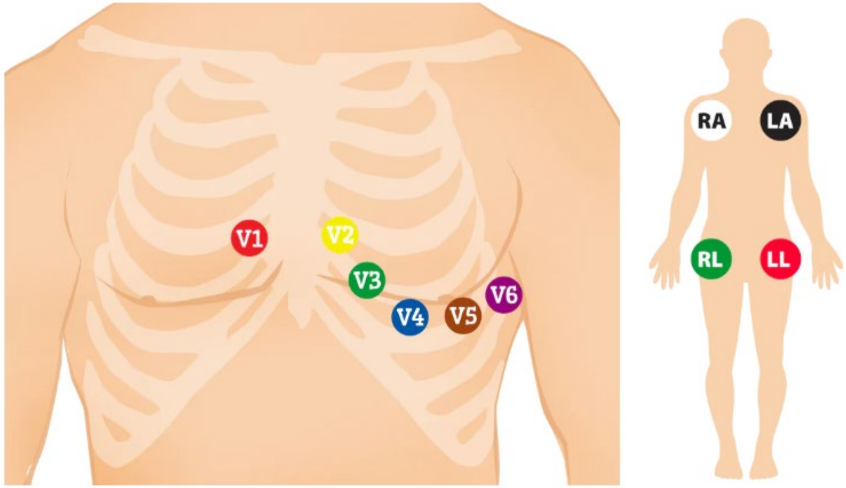
- Data pre-processing, including alignment and normalization, is a key step to perform a SVD.

References

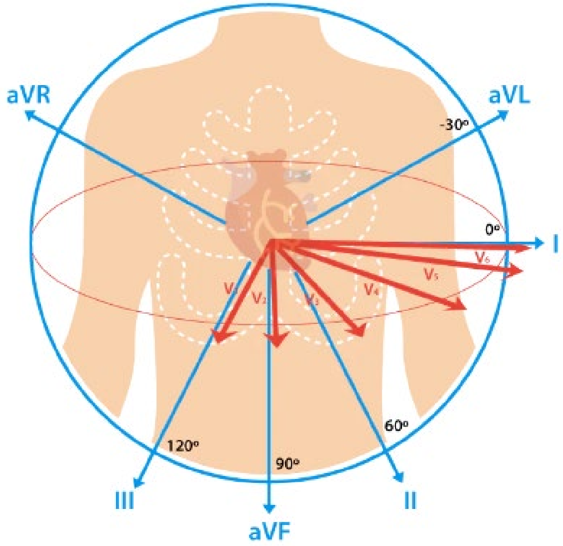
- G. H. Golub and C. F. Van Loan, Matrix Computations, John Hopkins Univ. Press, 3rd Ed., 1996.
- T.K. Moon and W.C. Stirling, Mathematical Methods and Algorithms for Signal Processing, Prentice Hall, 2000.
- N. Golyandina and A. Zhigljavski, Singular Spectrum Analysis for Time Series, Springer, 2013.
- Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Steven L. Brunton, J. Nathan Kutz (https://www.youtube.com/watch?v=gXbThCXjZFM&list=PLMrJAkhIeNNSVjnsviglFoY2nXildDCcv&ab_channel=SteveBrunton, http://databookuw.com/databook.pdf)

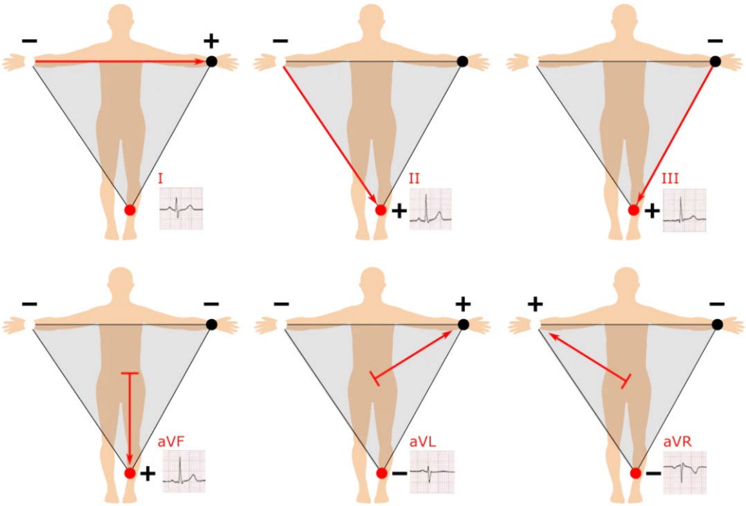
Lab experiment 1: 12-leads ECG

10 electrodes placement of a12-leads ECG



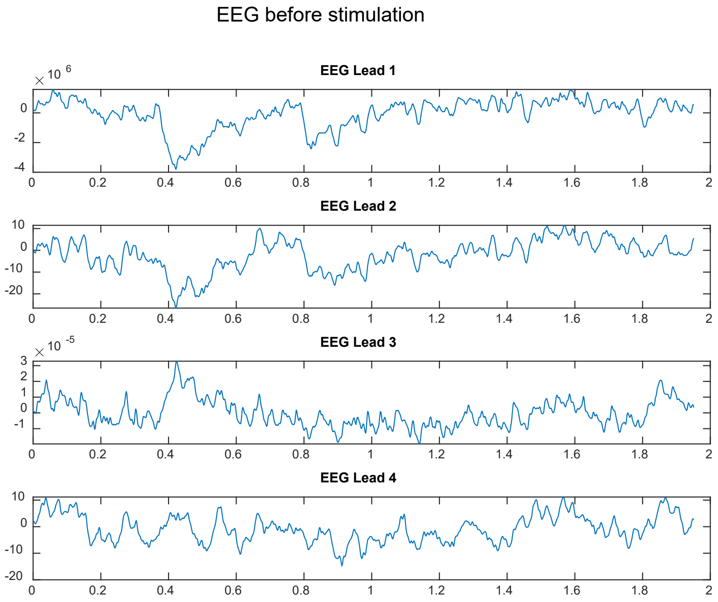
12-leads ECG

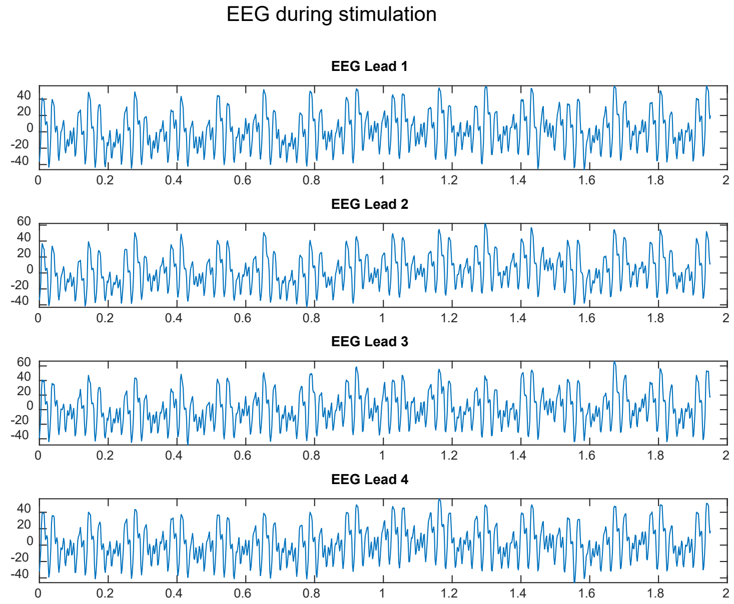


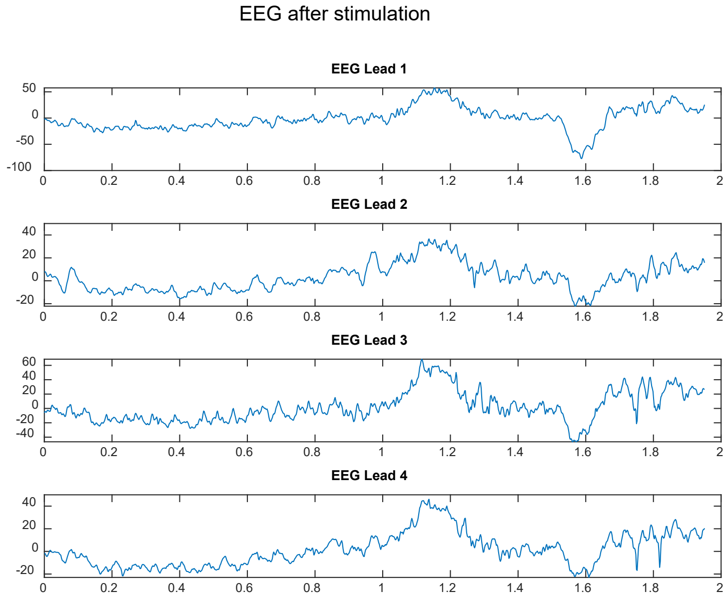


Lab experiment 2: Singular values and process complexity

EEG signals during brain stimulation for parkinson disease.







Lab experiment 3: Drift cancellation and frequency component extraction

PPG and accelerometer signals from a running subject.