ELEDIA@UniTN - University of Trento
Dept. of Civil, Environmental, and Mechanical Engineering
Via Mesiano 77, I-38123 Trento, Italy
E-mail: contact@eledia.org
Web: www.eledia.org/eledia-unitn

Software Exercise:

Principal Component Analysis

Dr. Marco SALUCCI

Day 1 - August 29th, 2022

2022 ELEDIA@ICAM PhD Summer Schools
Machine Learning & AI Methods
Theory, Techniques, and Advanced Engineering Applications
29 Aug. - 02 Sept. 2022, Trento, Italy (Onsite and Online)

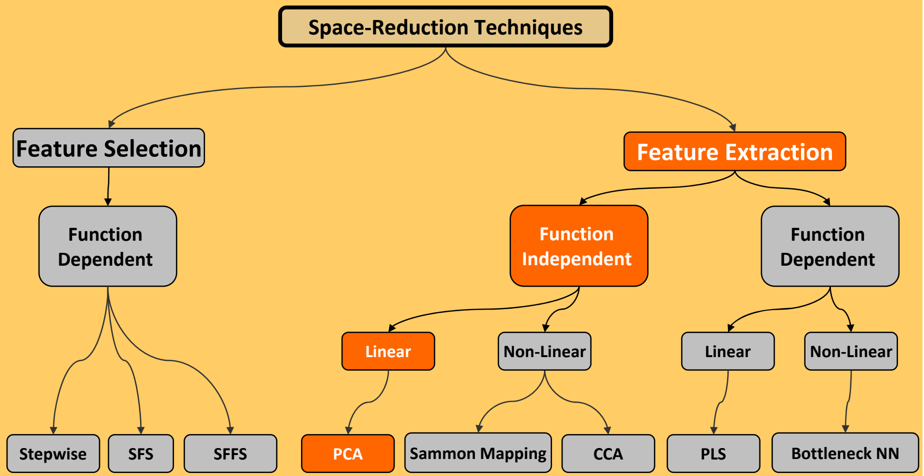
Copyright Notice

Download Link

https://storage.eledia.org/download/2216988075

username: **student**
password: **MATERIALE**

Principal Component Analysis (PCA)

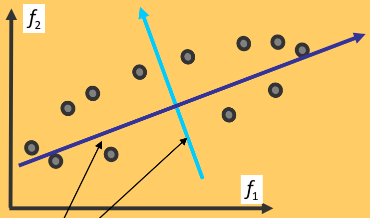


PCA - Basic Idea (1/2)

Idea Given P N-dimensional samples $\mathbf{F} = \{\mathbf{f}_p, p = 1,\ldots,P\}$ find the $H \le N$ **principal components** of data, i.e., the **directions** where there is the **largest variance**

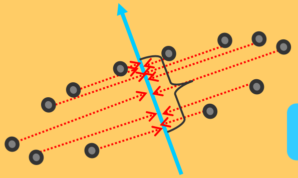
Examle: N=2, H=2

Disposition of samples in the original 2D space



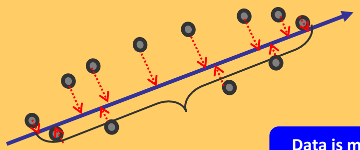
Two possible projection directions

Projection over first "trial" direction



Data is less variable along this direction

Projection over second "trial" direction

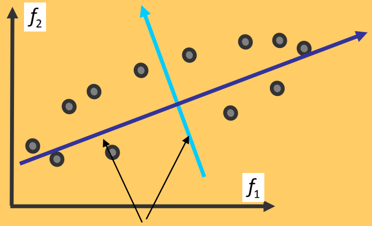


Data is more variable along this direction

PCA - Basic Idea (2/2)

Idea Given P N-dimensional samples $\mathbf{F} = \{\mathbf{f}_p, p = 1,\ldots,P\}$ find the $H \le N$ principal components of data, i.e., the **directions** where there is the **most variance**

Disposition of samples in the original 2D space



Two possible projection directions

Which directions?

Compute the **N eigenvectors** of the **covariance matrix** in the original input space

How much variance?

Compute the **eigenvalues** associated to each eigenvector: larger eigenvalue means larger variance

Feature extraction: Project input samples onto the H<N eigenvectors {W_h, h=1,…,H} showing the largest eigenvalues (the principal components)

$W_h; h = 1,\ldots,H$ Basis of the *J*-dimensional subspace

How to compute?

PCA – Computation of The Basis

Linear Transformation

New variables, Old variables, Basis

$$\underset{[P\times H]}{\mathbf{Z}} = \underset{[P\times N]}{\mathbf{F}} \times \underset{[N\times H]}{\mathbf{W}}$$

$$\underline{W} = \begin{bmatrix} W_{1,1} & W_{2,1} & \cdots & W_{H,1} \\ W_{1,2} & W_{2,2} & \cdots & W_{H,2} \\ \vdots & \vdots & \cdots & \vdots \\ W_{1,N} & W_{2,N} & \cdots & W_{H,N} \end{bmatrix}$$

$\underline{W}_1 \quad \underline{W}_2 \quad \underline{W}_H$

How to Compute? The basis vectors {W_h; h=1,…,H} are the **eigenvectors** of the **covariance matrix** associated to the *H* **highest eigenvalues**

Covariance Matrix

$$\mathbf{C} = \frac{1}{P-1}\sum_{p=1}^{P}\left(\mathbf{f}_p - \overline{\mathbf{f}}\right)\left(\mathbf{f}_p - \overline{\mathbf{f}}\right)^T$$

Average vector

$$\overline{\mathbf{f}} = \frac{1}{S}\sum_{s=1}^{S}\mathbf{f}_p$$

Properties

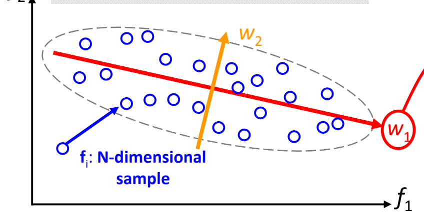
- {W_h; h=1,…,H} are called **principal components**
- {W_h; h=1,…,H} are mutually **orthogonal**
- W_1 has the **largest possible variance** (followed by W_2, …, W_H)

PCA – 2-D Example

EXAMPLE:

Number of unknowns: N = 2

Dimension of the output space: H = 1

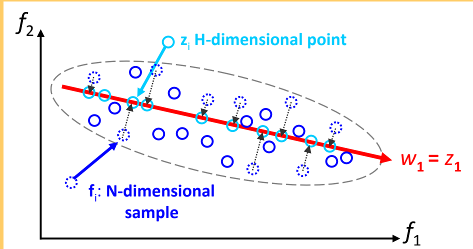


Eigenvector $\mathbf{w}_1$ corresponds to the highest eigenvalue

Greatest variance of the data lies on the first principal component $\mathbf{w}_1$

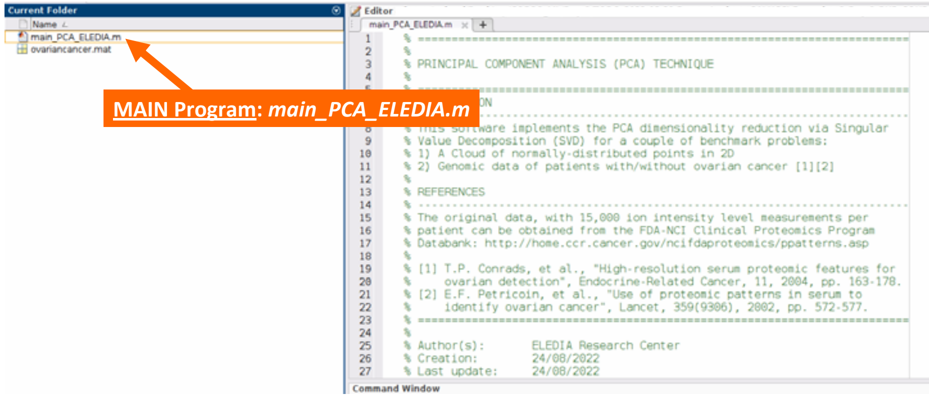
Project samples onto the principal component $\mathbf{w}_1$

- The total number of eigenvectors is equal to N
- The eigenvectors are mutually orthogonal



MATLAB Simulation - Initialization

- Run MATLAB
- Extract the provided **.zip** file and open the created folder from the "Current Folder" window
- Open the ***main_PCA_ELEDIA.m*** script



Input Parameters

```
% INPUT PARAMETERS
% ================================================================
% Benchmark selection
% BENCHMARK = 1: Normal-distributed cloud of points in 2D
% BENCHMARK = 2: Ovarian cancer genomic data
BENCHMARK = 1;

% [BENCHMARK=1] Cloud of points in 2D
% ----------------------------------------------------------------
% Number of points
NUM_POINTS = 10000;

% Center of the data (average value)
X1_AVG = 2;
X2_AVG = 1;

% Standard deviation of the data along the principal axes
P1_SIGMA = 2;
P2_SIGMA = 0.5;

% Rotation angle of the data [deg]
THETA_ROTATION_DEG = 60;

% Number of output dimensions (to show projected data)
NUM_OUTPUT_DIM = 1;
```

Benchmark 1: Normally distributed random data in 2D

Number of samples (S)

Geometric center of the data cloud (average)

Standard deviation of the data cloud along the two principal axes

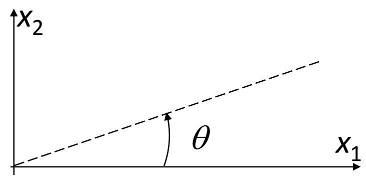
Rotation of the data cloud

Number of extracted features (H)

Data Generation: Rotation Matrix

```
% Compute the rotation matrix
theta = THETA_ROTATION_DEG/180*pi;
R = [cos(theta) -sin(theta);
     sin(theta) cos(theta)];
```

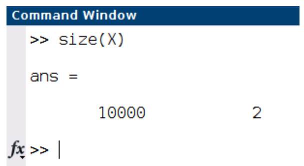
$$\underline{\underline{R}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$



Data Generation: Generate a Cloud of Points

```
% Generate the cloud of points (shift to average and stretch to sigma)
X = R*diag(sig)*randn(2,NUM_POINTS) + diag(xC)*ones(2,NUM_POINTS);
X = X';
```

$$\underline{\underline{X}} = \overbrace{\begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ \vdots & \vdots \\ x_1^{(S)} & x_2^{(S)} \end{bmatrix}}^{\text{N=2 Variables}} \left.\vphantom{\begin{bmatrix} x \\ x \\ \vdots \\ x \end{bmatrix}}\right\} \text{S=10.000 Samples}$$

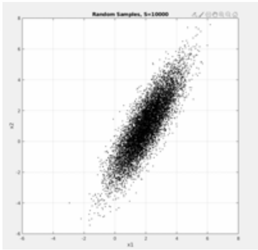


Plot Original Data in 2D Space

```
% Plot 1: Original data with estimated std deviation and PCs
% ----------------------------------------------------------------------
figure('units','normalized','outerposition',[0 0 1 1]);
subplot(1,2,1);

% Original data
scatter(X(:,1),X(:,2),'k.','LineWidth',2);

hold on;
box on;
grid on;
axis([-6 8 -6 8]);
pbaspect([1 1 1]);
xlabel('x1');
ylabel('x2');
title(sprintf('Random Samples, S=%d, Avg=[%.2f,%.2f], Sigma=[%.2f,%.2f]', ...
    NUM_POINTS, X1_AVG, X2_AVG, X1_SIGMA, X2_SIGMA));

return
```

Run the Code

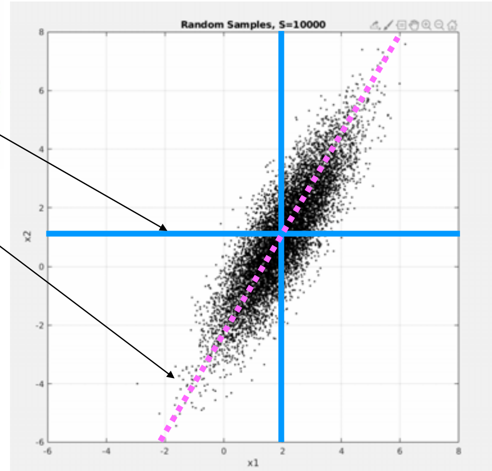
Run



Plot Original Data in 2D Space

```
% Number of points
NUM_POINTS = 10000;

% Center of the data (average value)
X1_AVG = 2;
X2_AVG = 1;

% Standard deviation of the data along the principal axes
P1_SIGMA = 2;
P2_SIGMA = 0.5;

% Rotation angle of the data [deg]
THETA_ROTATION_DEG = 60;
```

Data has a maximum variance along rotation angle (θ=60 [deg])



Center and Normalize Data

```
% Compute the average of the input data (S rows)
Xavg = mean(X,1);

% Center the data to the origin (subtract average)
X_cent = X - ones(NUM_POINTS,1)*Xavg;

% Normalize the centered data
X_cent_norm = X_cent /sqrt(NUM_POINTS);
```

```
% Center of the data (average value)
X1_AVG = 2;
X2_AVG = 1;
```

Command Window

```
>> Xavg

Xavg =

    2.0122    1.0267

fx >> |
```

$$\underline{\underline{X}} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ \vdots & \vdots \\ x_1^{(S)} & x_2^{(S)} \end{bmatrix}$$

Average of n-th variable

$$\overline{x}_n = \frac{1}{S}\sum_{s=1}^{S} x_n^{(s)}; \quad n = 1,\dots,N$$

$$\overline{x}_1 = \frac{1}{S}\sum_{s=1}^{S} x_1^{(s)} \qquad \overline{x}_2 = \frac{1}{S}\sum_{s=1}^{S} x_2^{(s)}$$

$$\underline{\underline{X}}' = \frac{1}{\sqrt{S}} \begin{bmatrix} \left(x_1^{(1)} - \overline{x}_1\right) & \left(x_2^{(1)} - \overline{x}_2\right) \\ \left(x_1^{(2)} - \overline{x}_1\right) & \left(x_2^{(2)} - \overline{x}_2\right) \\ \vdots & \vdots \\ \left(x_1^{(S)} - \overline{x}_1\right) & \left(x_2^{(S)} - \overline{x}_2\right) \end{bmatrix}$$

Optional normalization to make results independent on S

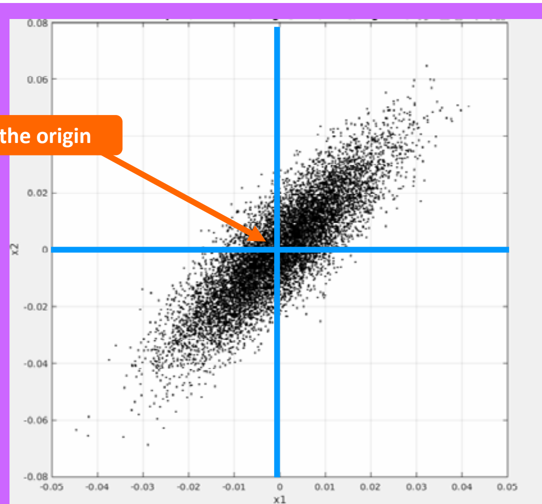
Center and Normalize Data

$$\underline{\underline{X}}' = \frac{1}{\sqrt{S}} \begin{bmatrix} \left(x_1^{(1)} - \overline{x}_1\right) & \left(x_2^{(1)} - \overline{x}_2\right) \\ \left(x_1^{(2)} - \overline{x}_1\right) & \left(x_2^{(2)} - \overline{x}_2\right) \\ \vdots & \vdots \\ \left(x_1^{(S)} - \overline{x}_1\right) & \left(x_2^{(S)} - \overline{x}_2\right) \end{bmatrix}$$

Optional code

```
scatter(X_cent_norm(:,1),X_cent_norm(:,2),'k.','LineWidth',2);
```

Now data is shifted to the origin



Compute the PCA through SVD

```
% Compute the PCA using the SVD
[U,S,V] = svd(X_cent_norm,'econ');
```

Singular Value Decomposition (SVD)

$$\underbrace{\underline{\underline{X}}'}_{S\times N} = \underbrace{\underline{\underline{U}}}_{S\times N}\,\underbrace{\underline{\underline{\Sigma}}}_{N\times N}\,\underbrace{\underline{\underline{V}}^T}_{N\times N}$$

Left singular vectors — Singular values — Right singular vectors

Singular values

$$\underline{\underline{\Sigma}} = \begin{bmatrix} \xi_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \xi_N \end{bmatrix} \qquad \xi_1 > \xi_2 > \dots > \xi_N$$

Meaning?

SVD Interpretation: Singular Values

Singular Value Decomposition (SVD)

$$\underbrace{\underline{\underline{X}}'}_{S\times N} = \underbrace{\underline{\underline{U}}}_{S\times N}\,\underbrace{\underline{\underline{\Sigma}}}_{N\times N}\,\underbrace{\underline{\underline{V}}^T}_{N\times N}$$

Left singular vectors — Singular values — Right singular vectors

```
% Compute the PCA using the SVD
[U,S,V] = svd(X_cent_norm,'econ');
```

Command Window

```
>> S

S =

    1.9963         0
         0    0.5028

fx >> |
```

Indicates that the first principal component / direction catches the largest amount of variance of the data …

$$\underline{\underline{\Sigma}} \approx \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}$$

```
% Standard deviation of the data along the principal axes
P1_SIGMA = 2;
P2_SIGMA = 0.5;
```

… As we know (we generated these data!)

SVD Interpretation: Right Singular Vectors

Singular Value Decomposition (SVD)

$$\underbrace{\underline{\underline{X}}'}_{S\times N} = \underbrace{\underline{\underline{U}}}_{S\times N}\,\underbrace{\underline{\underline{\Sigma}}}_{N\times N}\,\underbrace{\underline{\underline{V}}^T}_{N\times N}$$

Left singular vectors — Singular values — Right singular vectors

```
% Compute the PCA using the SVD
[U,S,V] = svd(X_cent_norm,'econ');
```

Command Window

```
>> V

V =

    0.4980   -0.8672
    0.8672    0.4980

fx >> |
```

$$\underline{\underline{V}} = \begin{bmatrix} v_1^{(1)} & v_2^{(1)} \\ v_1^{(2)} & v_2^{(2)} \end{bmatrix}$$

$\underline{V}_1$ $\underline{V}_2$

Colums of V indentify the two orthogonal directions of maximum variance in the data: the Principal Components

Indeed, it is almost equal to our rotation matrix!

```
>> R

R =

    0.5000   -0.8660
    0.8660    0.5000

fx >> |
```


SVD Interpretation: Right Singular Vectors

Singular Value Decomposition (SVD)

$$\underbrace{\underline{\underline{X}}'}_{S\times N} = \underbrace{\underline{\underline{U}}}_{S\times N}\,\underbrace{\underline{\underline{\Sigma}}}_{N\times N}\,\underbrace{\underline{\underline{V}}^T}_{N\times N}$$

Left singular vectors — Singular values — Right singular vectors

```
% Compute the PCA using the SVD
[U,S,V] = svd(X_cent_norm,'econ');
```

Command Window

```
>> V

V =

    0.4980   -0.8672
    0.8672    0.4980

fx >> |
```

$\underline{V}_1$ $\underline{V}_2$

$\left\|\underline{V}_1\right\| = 1$

Command Window

```
>> norm(V(:,1))

ans =

     1

fx >> |
```

$\left\|\underline{V}_2\right\| = 1$

Command Window

```
>> norm(V(:,2))

ans =

     1

fx >> |
```

The two columns are unit vectors since their norm is 1…

$\underline{V}_1 \bullet \underline{V}_2 = 0 \Rightarrow \underline{V}_1 \perp \underline{V}_2$

Command Window

```
>> dot(V(:,1),V(:,2))

ans =

     0

fx >> |
```

…and they are orthogonal since their scalar product is 0

PCA: Alternative Computation

```
% Note: the same result can be obtained also passing through the
% covariance matrix of the data
% -----------------------------------------------------------------------
% Compute the covariance matrix
C = cov(X_cent_norm);

% Compute the eigenvectors and eigenvalues
[V_unsorted,D_unsorted] = eig(C);

% Sort the eigenvectors from largest to smallest
[sorted_eigenvalues, Index] = sort([D_unsorted(1,1) D_unsorted(2,2)], 'descend');
D = diag(sorted_eigenvalues);
V = [V_unsorted(:,Index(1)), V_unsorted(:,Index(2))];
```

Covariance Matrix of Data

Compute eigenvectors/eigenvalues

Sort eigenvalues from maximum to minimum and arrange eigenvectors accordingly

Produces same outcome as SVD

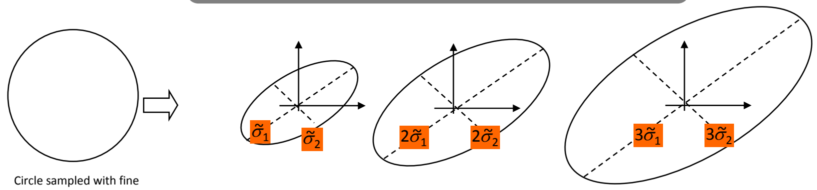
Plot Estimated Standard Deviation

```
%return
% Draw three circles centered on the data average and stretched by
% estimated sigma (1*sigma, 2*sigma, and 3*sigma)
angles = (0:.01:1)*2*pi;

% Estimated std deviation
Xstd = V*S*[cos(angles); sin(angles)];

% Draw ellipses
plot(Xavg(1)+Xstd(1,:),   Xavg(2) +   Xstd(2,:),'r-', 'LineWidth', 2);
plot(Xavg(1)+2*Xstd(1,:), Xavg(2) + 2*Xstd(2,:),'g-', 'LineWidth', 2);
plot(Xavg(1)+3*Xstd(1,:), Xavg(2) + 3*Xstd(2,:),'b-', 'LineWidth', 2);

return
```

Sample the angular range [0:2π]

Scale the principal components (V) by estimated std deviations (S)

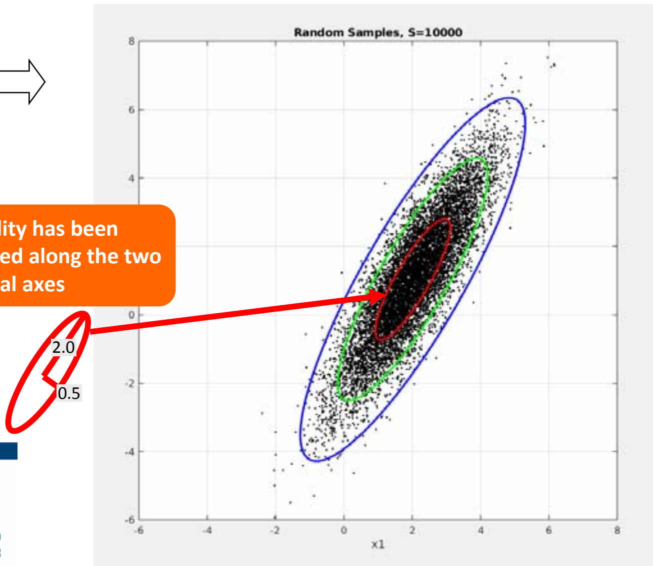
Plot 3 circles centered on data and stretched by estimated 1,2,3 std deviations



Circle sampled with fine angular step (0.01 rad)

Plot Estimated Standard Deviation (±σ, ±2σ, ±3σ)

Run the Code



Data variability has been correctly estimated along the two principal axes

$$\underline{\underline{\Sigma}} \approx \begin{bmatrix} \boxed{2} & 0 \\ 0 & \boxed{0.5} \end{bmatrix}$$

```
Command Window
>> S

S =

    1.9963         0
         0    0.5028

fx >>
```

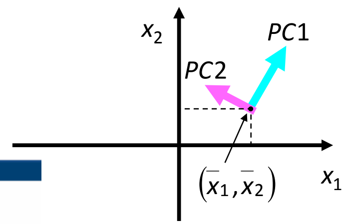

Plot the Two Principal Components

```
%return
% Plot principal components V(:,1)S(1,1) and V(:,2)S(2,2)
plot([Xavg(1) Xavg(1)+V(1,1)*S(1,1)], [Xavg(2) Xavg(2)+V(2,1)*S(1,1)],...
    'c-', 'LineWidth', 2);
plot([Xavg(1) Xavg(1)+V(1,2)*S(2,2)], [Xavg(2) Xavg(2)+V(2,2)*S(2,2)],...
    'm-', 'LineWidth', 2);

legend('Samples', '\sigma', '2*\sigma', '3*\sigma', 'PC1', 'PC2');

return
```

```
Command Window
>> V

V =

    0.4980   -0.8672
    0.8672    0.4980

fx >>
```

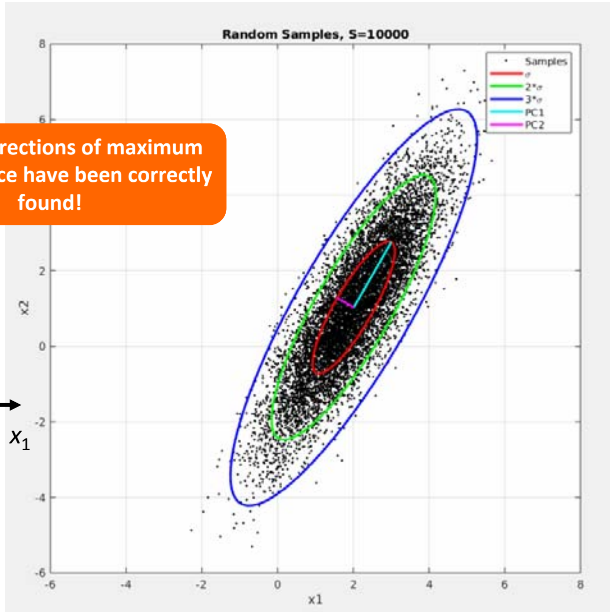
```
Command Window
>> S

S =

    1.9963         0
         0    0.5028

fx >>
```

```
Command Window
>> Xavg

Xavg =

    2.0122    1.0267

fx >>
```

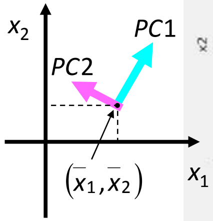


For plotting, we scale each vector by the corresponding std deviation

Plot the Two Principal Components



The two directions of maximum data variance have been correctly found!

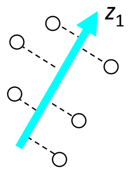


Project Data Onto Principal Components

```
% Project data onto first principal component
Projection_1 = zeros(NUM_POINTS,1);
for i=1:NUM_POINTS
    Projection_1(i) = X_cent_norm(i,:)*V(:,1);
end

% Project data onto second principal component
Projection_2 = zeros(NUM_POINTS,1);
for i=1:NUM_POINTS
    Projection_2(i) = X_cent_norm(i,:)*V(:,2);
end
```

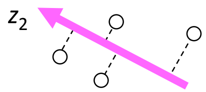
$$\underline{z}_1 = \underline{\underline{X}}' \times \underline{v}_1$$

$S \times 1$ $S \times N$ $N \times 1$



$$\underline{z}_2 = \underline{\underline{X}}' \times \underline{v}_2$$

$S \times 1$ $S \times N$ $N \times 1$



Plot Projected Data

```
%return
% Plot 2: Projected data onto PCs
% -------------------------------------------------------------
subplot(1,2,2);

switch NUM_OUTPUT_DIM
    case 1
        % Plot projected samples onto PC1 (sorted)
        plot(sort(Projection_1), 'cx', 'LineWidth',3);

        xlabel('Sample Index, S');
        ylabel('Projection on PC1');
        title('Projected Data, H=1');
    case 2
        % Plot projected samples onto 2D space (PC1,PC2)
        scatter(Projection_1,Projection_2,'k.','LineWidth',2);

        pbaspect([1 1 1]);
        axis([-0.1 0.1 -0.1 0.1]);
        xlabel('Projection on PC1');
        ylabel('Projection on PC2');
        title('Projected Data, H=2');
    otherwise
        error('Can project only to 1 or 2 PCs!');
end
box on;
grid on;
```

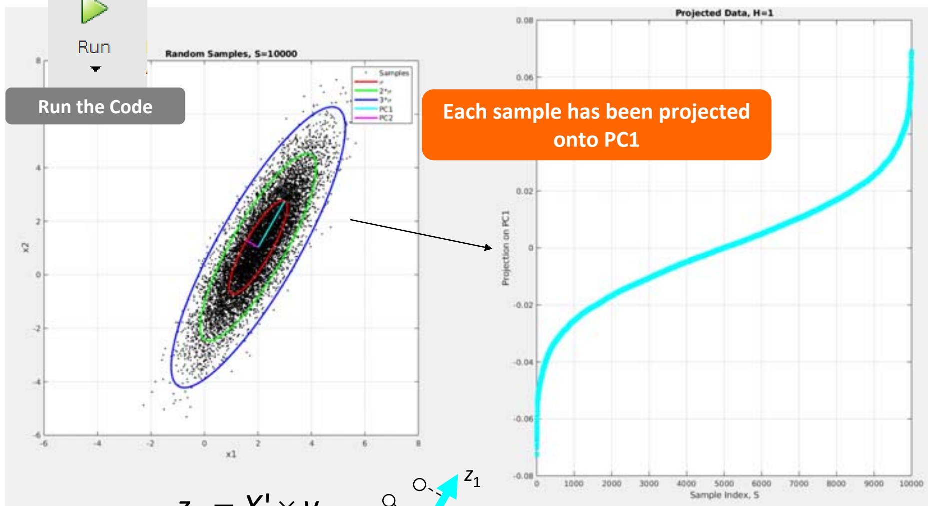
H=1

H=2

Plot Projected Data (H=1)

Run

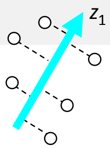
Run the Code



Each sample has been projected onto PC1

$$\underline{z}_1 = \underline{\underline{X}}' \times \underline{v}_1$$

$S \times 1$ $S \times N$ $N \times 1$

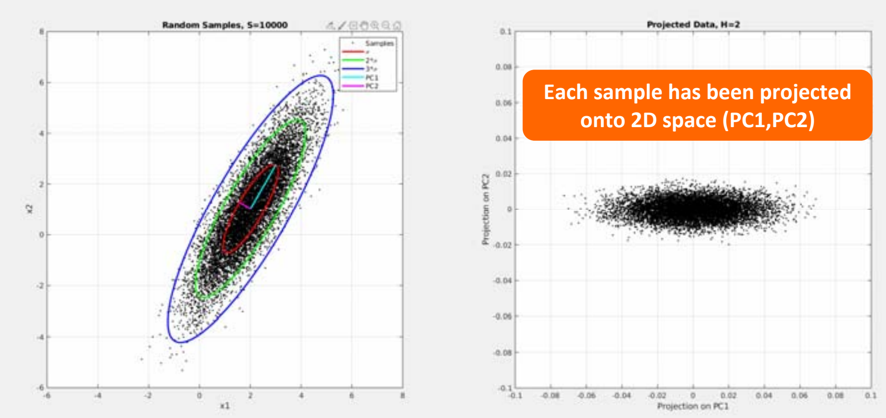


Plot Projected Data (H=2)

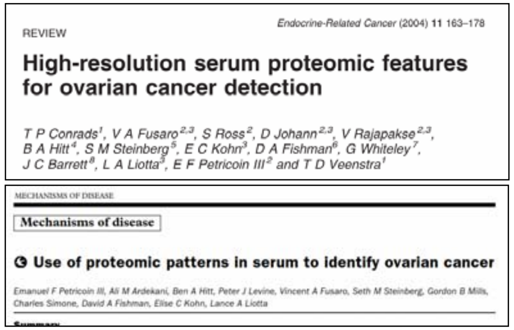
```
% Number of output dimensions (to show projected data)
NUM_OUTPUT_DIM = 2;
```

Run

Run the Code



PCA@Work: Ovarian Cancer Dataset


Endocrine-Related Cancer (2004) 11 163–178

REVIEW

High-resolution serum proteomic features for ovarian cancer detection

T P Conrads[1], V A Fusaro[2,3], S Ross[2], D Johann[2,3], V Rajapakse[2,3], B A Hitt[4], S M Steinberg[5], E C Kohn[3], D A Fishman[6], G Whiteley[7], J C Barrett[8], L A Liotta[3], E F Petricoin III[2] and T D Veenstra[1]

MECHANISMS OF DISEASE

Mechanisms of disease

Use of proteomic patterns in serum to identify ovarian cancer

Emanuel F Petricoin III, Ali M Ardekani, Ben A Hitt, Peter J Levine, Vincent A Fusaro, Seth M Steinberg, Gordon B Mills, Charles Simone, David A Fishman, Elise C Kohn, Lance A Liotta

Background New technologies for the detection of early-stage ovarian cancer are urgently needed. Pathological changes within an organ might be reflected in proteomic patterns in serum. We developed a bioinformatics tool and

Methods Proteomic spectra were generated by mass spectroscopy (surface-enhanced laser desorption and ionisation). A preliminary "training" set of spectra derived from analysis of serum from 50 unaffected women and 50 patients with ovarian cancer were analysed by an iterative searching algorithm that identified a proteomic pattern that completely discriminated cancer from non-cancer. The discovered pattern was then used to classify an independent set of 116 masked serum samples: 50 from women with ovarian cancer, and 66 from unaffected women or those with non-malignant disorders.

PCA@Work: Ovarian Cancer Dataset

```
% INPUT PARAMETERS
% ========================================================
% Benchmark selection
% BENCHMARK = 1: Normal-distributed cloud of points in 2D
% BENCHMARK = 2: Ovarian cancer genomic data
BENCHMARK = 2;

% BENCHMARK 2: OVARIAN CANCER DATA
% ***********************************************
% Load ovarian cancer data
% -----------------------------------------------
load ovariancancer.mat;
return
```

Dataset Contents

"obs": Genomic data

$$\underline{\underline{X}} = \underbrace{\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_N^{(1)} \\ x_1^{(2)} & x_2^{(2)} & & x_N^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(S)} & x_2^{(S)} & \cdots & x_N^{(1)} \end{bmatrix}}_{\substack{N=4000 \\ \text{features}}} \left.\vphantom{\begin{bmatrix}a\\a\\a\\a\end{bmatrix}}\right\} \begin{matrix}S=216 \\ \text{patients}\end{matrix}$$

```
>> size(obs)

ans =

   216   4000
```

"grp": Associated labels

```
grp =

  216×1 cell array

    {'Cancer'}
    {'Cancer'}
    {'Cancer'}
    {'Cancer'}
    {'Cancer'}
    {'Cancer'}
    {'Cancer'}
    {'Cancer'}
    {'Cancer'}
    {'Cancer'}
    {'Cancer'}
    {'Cancer'}
    {'Cancer'}
    {'Normal'}
    {'Normal'}
    {'Normal'}
    {'Normal'}
    {'Normal'}
    {'Normal'}
    {'Normal'}
    {'Normal'}
    {'Normal'}
    {'Normal'}
    {'Normal'}
    {'Normal'}
    {'Normal'}
    {'Normal'}
```

95 Patients healthy (Normal)

121 Patients with Cancer

How to visualize 4000 features and check if there exist any relation with cancer?

Apply the PCA on the Dataset

"obs": Genomic data

$$\underline{\underline{Y}} = \underbrace{\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_N^{(1)} \\ x_1^{(2)} & x_2^{(2)} & & x_N^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(S)} & x_2^{(S)} & \cdots & x_N^{(1)} \end{bmatrix}}_{\substack{N=4000 \\ \text{features}}} \left.\vphantom{\begin{bmatrix}a\\a\\a\\a\end{bmatrix}}\right\} \begin{matrix}S=216 \\ \text{patients}\end{matrix}$$

```
% BENCHMARK 2: OVARIAN CANCER DATA
% ******************************************
% Load ovarian cancer data
% ------------------------------------------
load ovariancancer.mat;
return

% Compute PCA using SVD
% ------------------------------------------
[U,S,V] = svd(obs,'econ');
```

Command Window

```
>> size(V)

ans =

        4000          216

fx >>
```

```
>> size(S)

ans =

   216   216
```

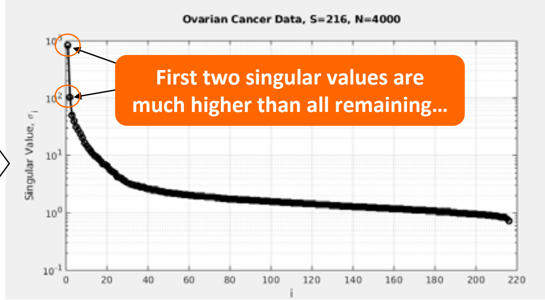
What can we infer from singular values?

Plot of the Singular Values

```
% Plot 1: Singular values
% ------------------------------------------------------------------
figure('units','normalized','outerposition',[0 0 1 1]);
subplot(2,2,1);

semilogy(diag(S), 'k-o', 'LineWidth', 2);

grid on;
xlabel('i');
ylabel('Singular Value, \sigma_i');
axis([0 220 1E-1 1E3]);
title(sprintf('Ovarian Cancer Data, S=%d, N=%d\n', size(obs,1), size(obs,2)));

return
```

Run the Code

Run

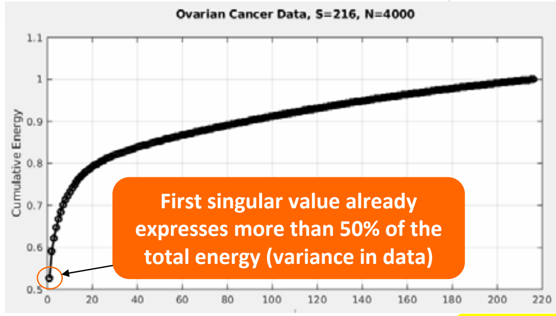


Plot Singular Values Cumulative Energy

```
%return
% Plot 2: Cumulative energy of singular values
% ------------------------------------------------------------------
subplot(2,2,2);

plot(cumsum(diag(S))./sum(diag(S)), 'k-o', 'LineWidth', 2);

grid on;
xlabel('i');
ylabel('Cumulative Energy');
axis([0 220 0.5 1.1]);
title(sprintf('Ovarian Cancer Data, S=%d, N=%d\n', size(obs,1), size(obs,2)));

return
```

Run the Code

Run



Meaning?

Plot Project Data (H=1)

```
% Number of output dimensions (to show projected data)
NUM_OUTPUT_DIM = 1;

%return

% Plot 3: Projected data
% ------------------------------------------------------------------
subplot(2,2,3:4);
hold on;

switch NUM_OUTPUT_DIM
    case 1
        % Plot projected samples onto PC1 (sorted)
        [Projs,Indexes] = sort(Projection_1);
        for i=1:size(obs,1)
            if(strcmp(grp{Indexes(i)}, 'Cancer'))
                plot(i,Projs(i), 'rx', 'LineWidth',3);
            else
                plot(i,Projs(i), 'bo', 'LineWidth',3);
            end
        end
        grid on;
        box on;
        axis([0 220 -80 -10]);
        xlabel('Sample Index, s');
        ylabel('Projection on PC1');
        title('Projected Data, H=1');
        h(1) = plot(NaN,'rx', 'LineWidth',3);
        h(2) = plot(NaN,'bo', 'LineWidth',3);
        legend(h, 'Cancer','Healthy');
```

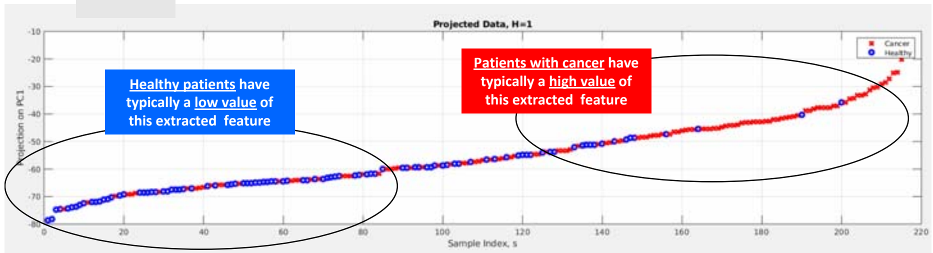
Sort projections onto PC1

For each point, plot with x if cancer, plot with o if healty

Plot Project Data (H=1)

Run the Code

Run

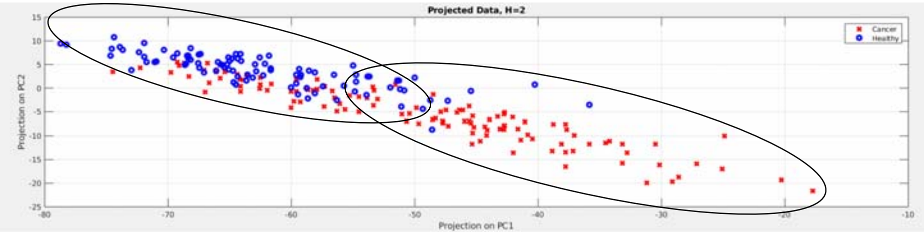


The first extracted feature is already very informative on the probability of having cancer!

What About 2nd and 3rd PCs?

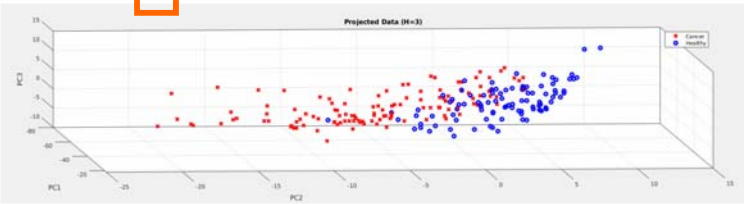
Plot Project Data (H=2 & H=3)

```
% Number of output dimensions (to show projected data)
NUM_OUTPUT_DIM = 2;
```



Easy to see that data can be easily clustered into 2 regions

```
% Number of output dimensions (to show projected data)
NUM_OUTPUT_DIM = 3;
```



ELEDIA@UniTN - University of Trento
Dept. of Civil, Environmental, and Mechanical Engineering
Via Mesiano 77, I-38123 Trento, Italy
E-mail: contact@eledia.org
Web: www.eledia.org/eledia-unitn

Software Exercise:
Principal Component Analysis

Dr. Marco SALUCCI

Day 1 - August 29th, 2022

2022 ELEDIA@ICAM PhD Summer Schools
Machine Learning & AI Methods
Theory, Techniques, and Advanced Engineering Applications
29 Aug. - 02 Sept. 2022, Trento, Italy (Onsite and Online)

Additional Information

Contact Point: **Marco Salucci**
ELEDIA Research Center Board Member
Assistant Professor @ University of Trento (Trento - Italy)

E-mail: marco.salucci@eledia.org
marco.salucci@unitn.it
Web-site: www.eledia.org

summer-schools@eledia.org

https://www.eledia.org
https://twitter.com/ELEDIAResearch
https://www.facebook.com/eledianet
https://www.linkedin.com/company/eledianet
https://www.instagram.com/eledianet
WeChat ID: eledianet