Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Introduction to Image Processing:

Global Operators: PCA

Principal Component Analysis

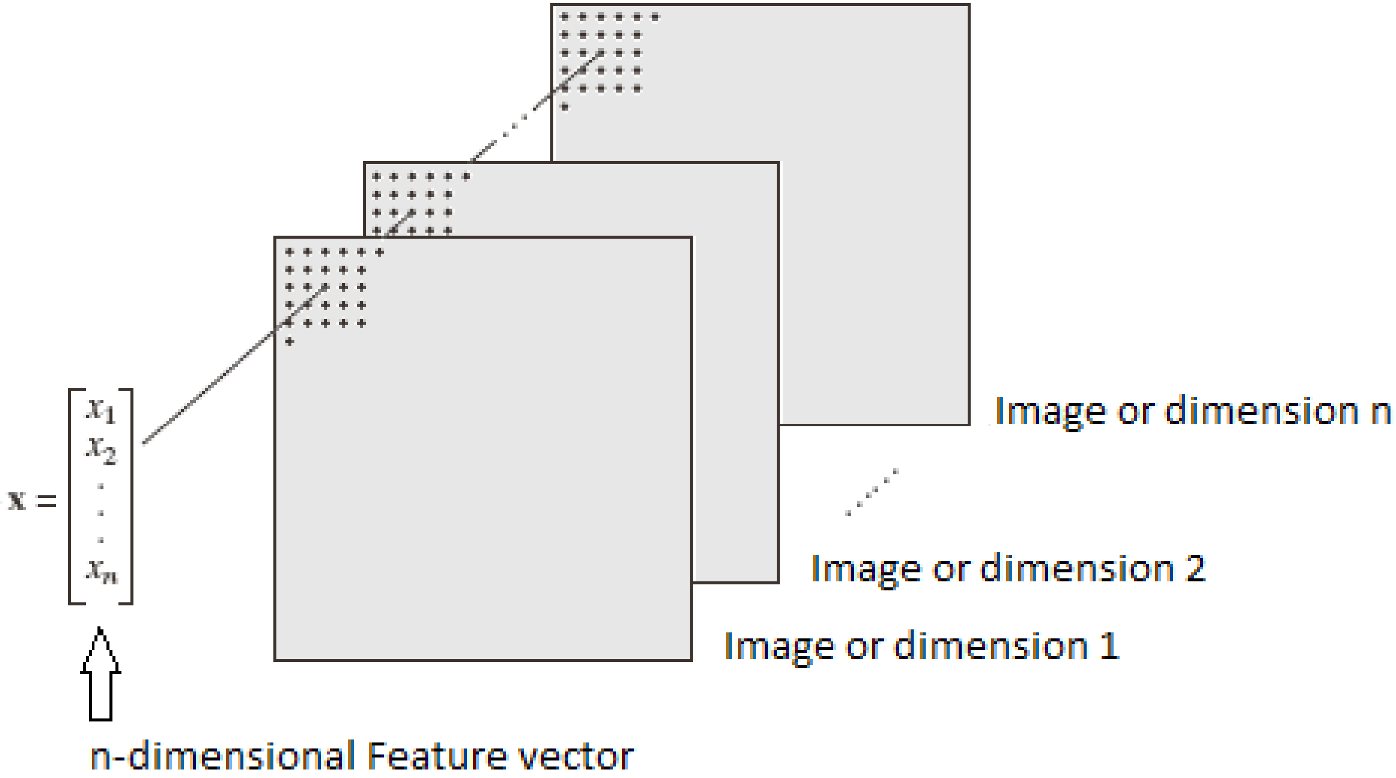
Marcus Hudritsch (hsm4)

Global Operators: Principal Component Analysis

- The **Principal Component Analysis** (**PCA**) or **Principal Component Transform** (**PCT**) transforms a multidimensional (=multivariate) dataset.
- In image processing it is sometimes called **Karhunen-Loève** transform.
- The new **dimensions are ordered by variance**.
- The **variance of an image** can be interpreted as **contrast**.

Global Operators: Principal Component Analysis

- The color channels of an RGB image are e.g. 3 dimensions.
- The PCA is used mostly for higher dimensionality.
- The **PCA is a statistical transform** where a single value must correspond with the values in the other dimensions at the same position:
- All values at the same position form a **feature vector**:



Global Operators: Principal Component Analysis

- The PCA is a simple multiplication of:
  - a linear **matrix *E*** with **eigenvectors in the rows** with the
  - data **matrix *X*** with **n-dimensional feature vectors in columns**

$$X_{PC} = E \cdot X = \begin{bmatrix} \boxed{e_{1x}, e_{1y}, \ldots, e_{1n}} \\ e_{2x}, e_{2y}, \ldots, e_{2n} \\ \ldots, \ldots, \ldots, \ldots \\ e_{mx}, e_{my}, \ldots, e_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_{11}, x_{12}, \ldots, x_{1m} \\ x_{21}, x_{22}, \ldots, x_{2m} \\ \ldots, \ldots, \ldots, \ldots \\ x_{n1}, x_{n2}, \ldots, x_{nm} \end{bmatrix}$$

- There are two methods to derive the transform matrix:
  - **Eigenvalue & eigenvector decomposition with the covariance matrix.**
  - Singular Value Decomposition

Global Operators: Principal Component Analysis

- The PCA has **similarities to the Fourier Transform (FT)**:
  - The **FT** splits a signal into a **weighted sum of othogonal sine & cosine frequencies**.
  - The **PCA** splits a signal up into a **weighted sum of PC vectors**.

PCA: Variance, Covariance & Covariance Matrix

- The **variance** of a 1D dataset X is defined as:

$$var(X) = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2$$

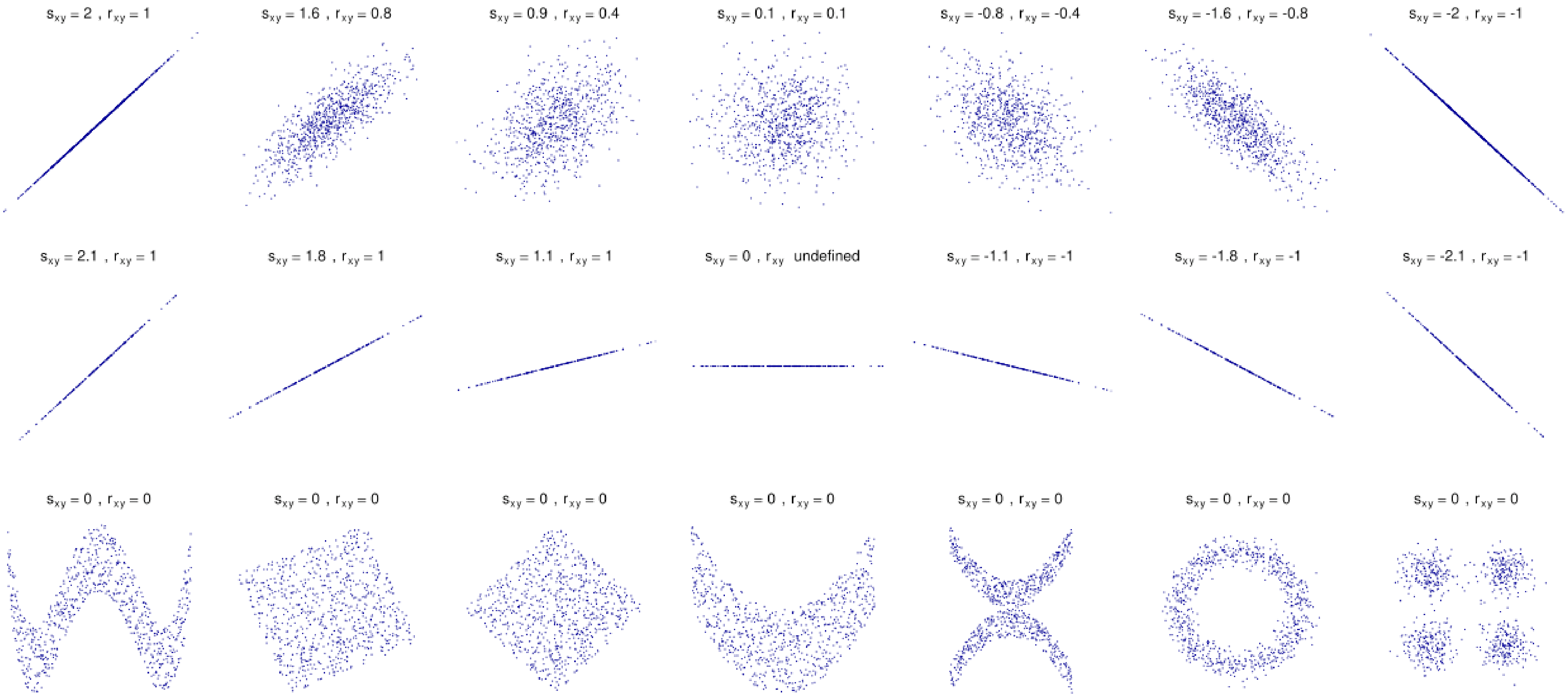
- For two dimensions you can calculate the **covariance**:

$$cov(X, Y) = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})$$

- The **covariance** value can be interpreted as follows:
  - A **positive value** mean a **strong positive correlation**: Low values in X correspond in low values in Y and vice versa.
  - A **negative value** mean a **strong negative correlation**: Low values in X correspond to high values in Y and vice versa.
  - A **zero value** mean **no correlation** at all.

PCA: Variance, Covariance & Covariance Matrix

- The **covariance s_{xy}** of the 2D datasets X & Y:



(The r_{xy} values are the correlation value that is normalized between -1 & 1)

PCA: Variance, Covariance & Covariance Matrix

- The n dimensions you get $\frac{n!}{2(n-2)!}$ covariances.
- For 3 dimensions you get: $cov(X,Y), cov(X,Z), cov(Y,Z)$
- The **covariance matrix** is the assembly of all covariances in a matrix.
- For 2D- and 3D-Data they are:

$$C_{XY} = \begin{bmatrix} cov(X,X), cov(X,Y) \\ cov(Y,X), cov(Y,Y) \end{bmatrix}$$

$$C_{XYZ} = \begin{bmatrix} cov(X,X), cov(X,Y), cov(X,Z) \\ cov(Y,X), cov(Y,Y), cov(Y,Z) \\ cov(Z,X), cov(Z,Y), cov(Z,Z) \end{bmatrix}$$

PCA: Variance, Covariance & Covariance Matrix

- The **covariance matrix** has important properties:
  - It is symmetric: All 'real' covariances appear doubled and mirrored
  - On the diagonal are all positive variances of the single dimensions

$$C_{XYZ} = \begin{bmatrix} cov(X,X), cov(X,Y), cov(X,Z) \\ cov(Y,X), cov(Y,Y), cov(Y,Z) \\ cov(Z,X), cov(Z,Y), cov(Z,Z) \end{bmatrix}$$

- These properties allow to extract the **eigenvectors & eigenvalues**!

PCA: Eigenvectors & Eigenvalues

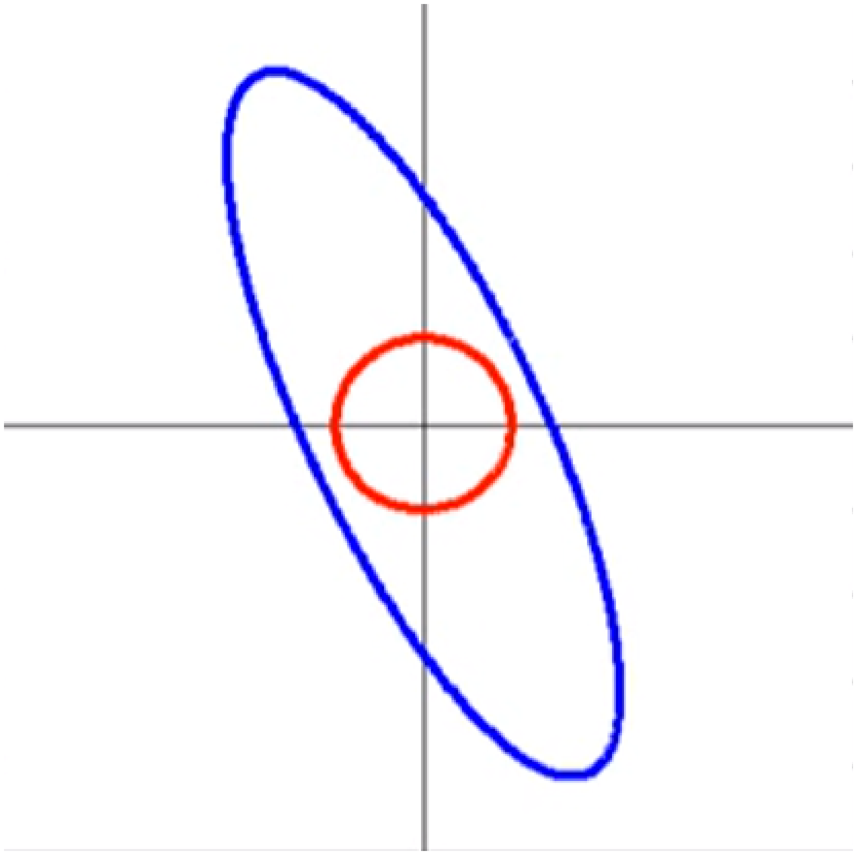
- **Eigenvectors** and **eigenvalues** can be calculated from the covariance matrix.
- An **eigenvector** $\vec{v}$ is a **vector** and is '*eigen*' to a matrix because if you multiply it with the matrix you get the same vector scaled by the **eigenvalue** λ (lambda):

$$A\vec{v} = \lambda \vec{v}$$

PCA: Eigenvectors & Eigenvalues

Geometric interpretation:

- What are the eigenvectors & eigenvalues of the matrix A= $\begin{bmatrix} 1, -2 \\ 1, 4 \end{bmatrix}$
- If we multiply all red points by the matrix we get the blue ellipse:

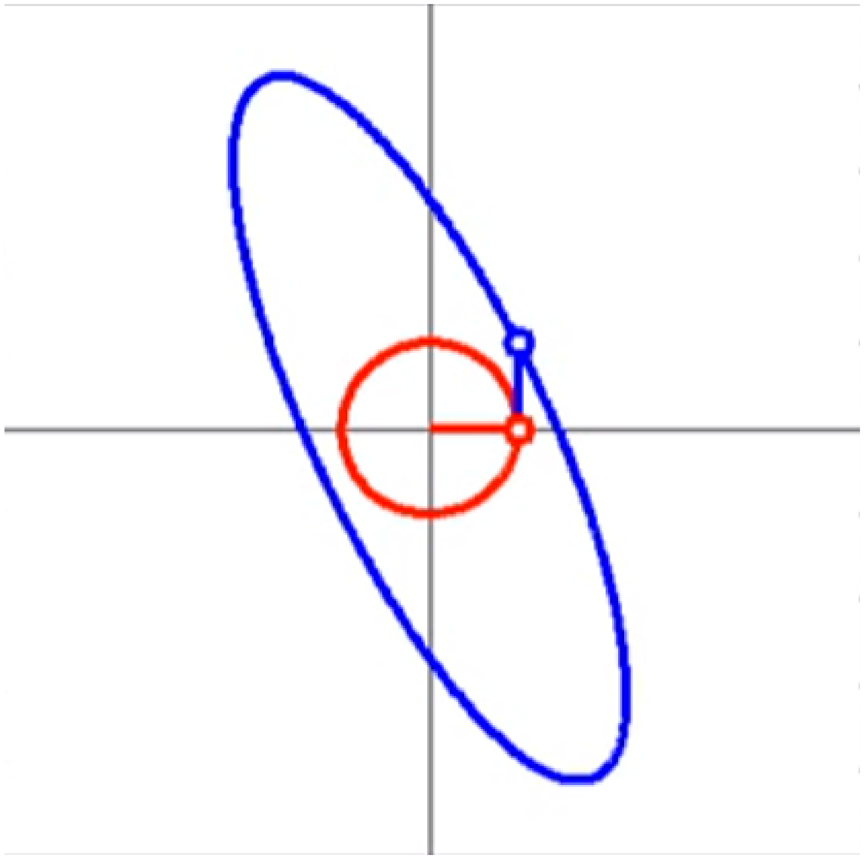


PCA: Eigenvectors & Eigenvalues

Geometric interpretation:

- What are the eigenvectors & eigenvalues of the matrix A= $\begin{bmatrix} 1\,, -2 \\ 1\,, 4 \end{bmatrix}$

- E.g. point [1,0] gets point [1,1]:

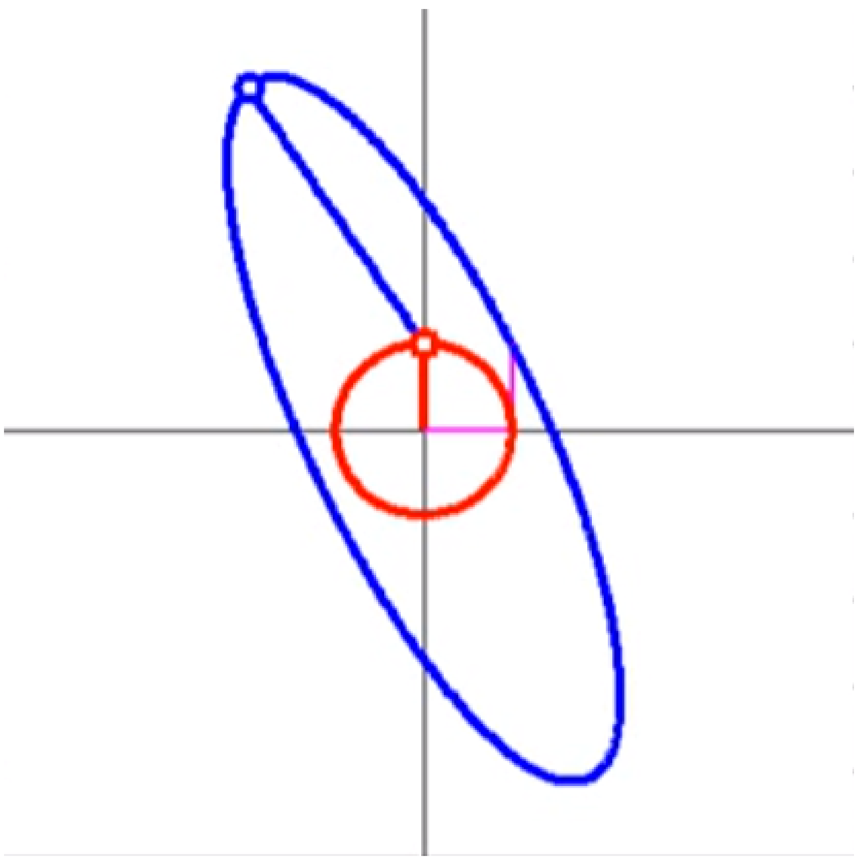


PCA: Eigenvectors & Eigenvalues

Geometric interpretation:

- What are the eigenvectors & eigenvalues of the matrix A= $\begin{bmatrix} 1, -2 \\ 1, 4 \end{bmatrix}$

- or point [0,1] gets point [-2,4]:

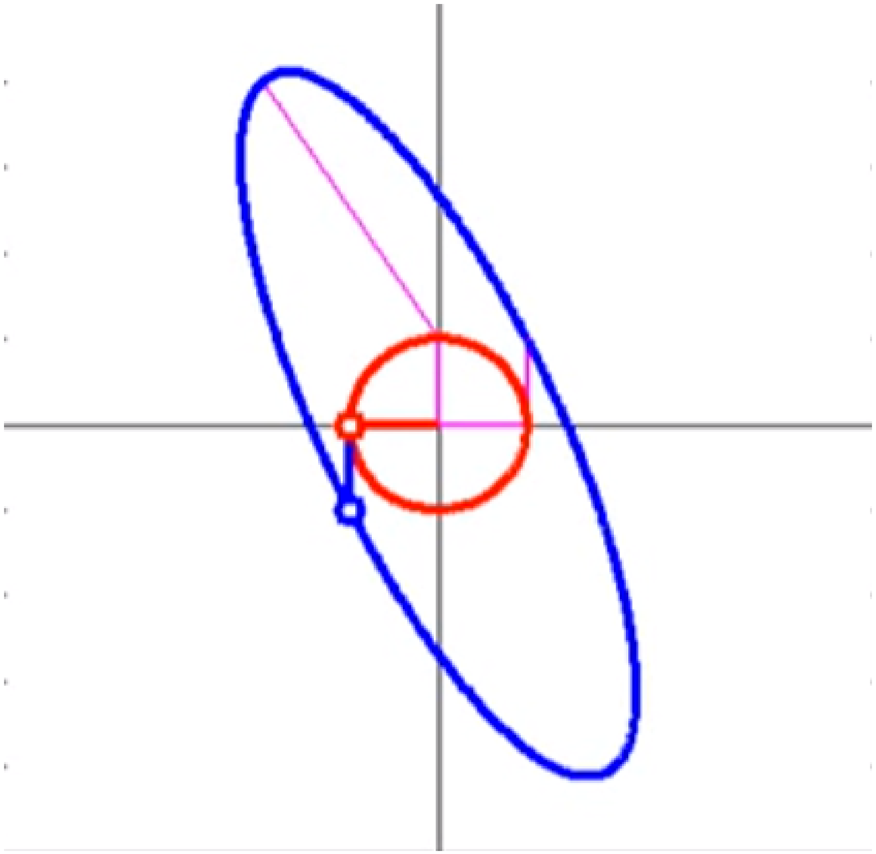


PCA: Eigenvectors & Eigenvalues

Geometric interpretation:

- What are the eigenvectors & eigenvalues of the matrix A= $\begin{bmatrix} 1, -2 \\ 1, 4 \end{bmatrix}$

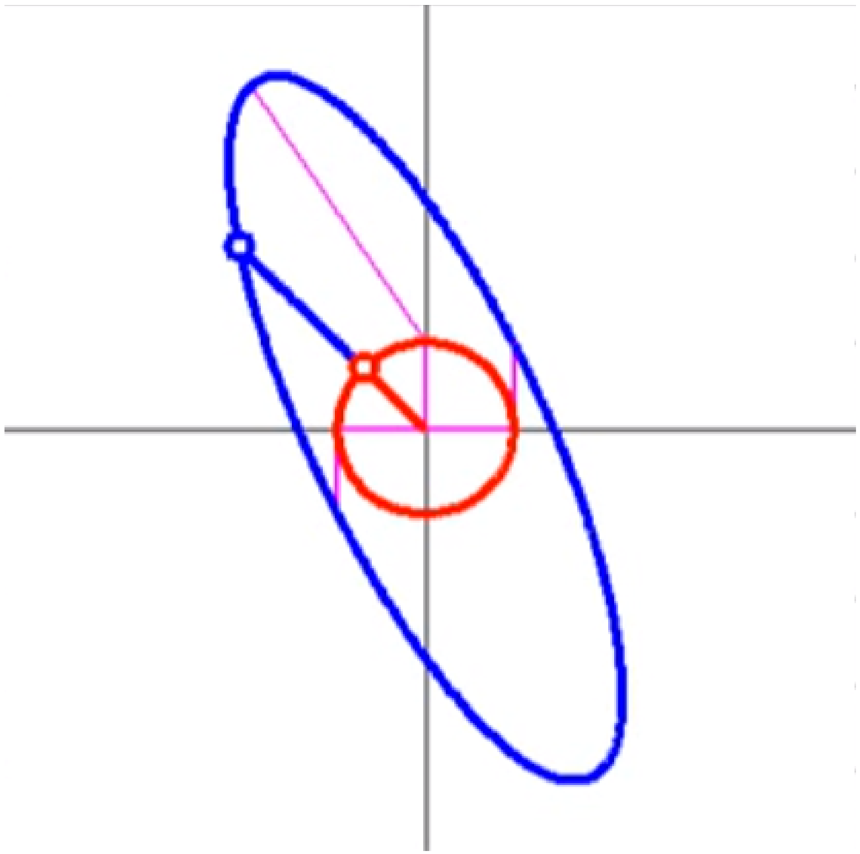
- or point [-1,0] gets point [-1,-1]:



PCA: Eigenvectors & Eigenvalues

Geometric interpretation:

- What are the eigenvectors & eigenvalues of the matrix A= $\begin{bmatrix} 1, -2 \\ 1, 4 \end{bmatrix}$
- but point [-0.71,0.71] gets point [-2.1,2.1]. It has the same direction!

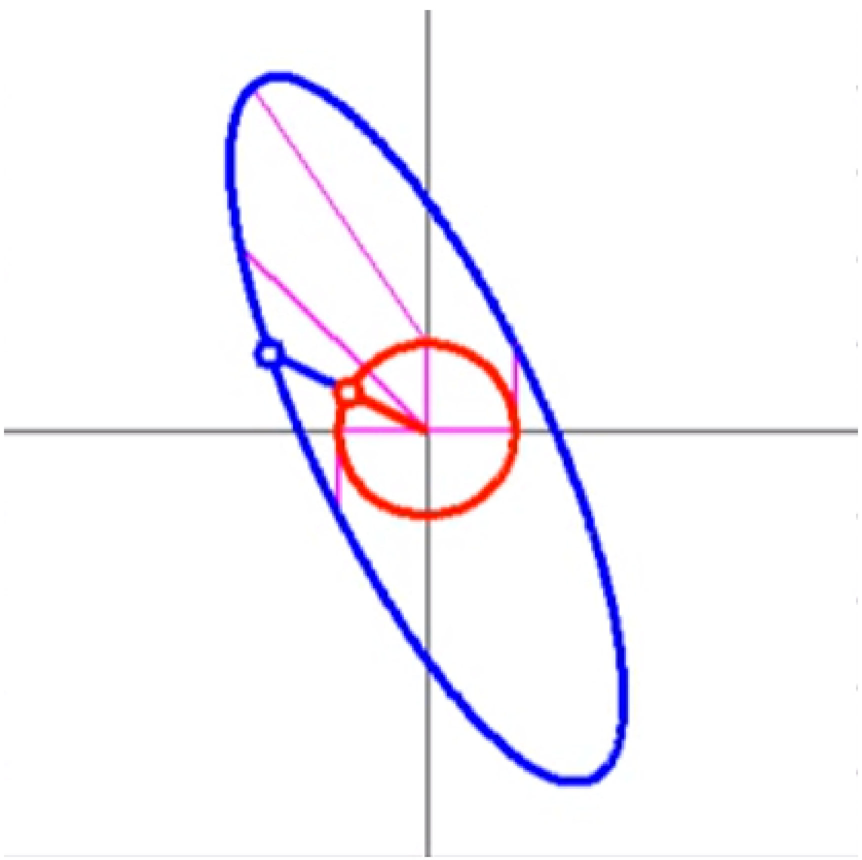


$$\begin{bmatrix} 1, -2 \\ 1, 4 \end{bmatrix} \cdot \begin{bmatrix} -0.71 \\ 0.71 \end{bmatrix} = \begin{bmatrix} -2.1 \\ 2.1 \end{bmatrix} = \underbrace{3}_{\text{eigenvalue}} \underbrace{\begin{bmatrix} -0.71 \\ 0.71 \end{bmatrix}}_{\text{eigenvector}}$$

PCA: Eigenvectors & Eigenvalues

Geometric interpretation:

- What are the eigenvectors & eigenvalues of the matrix A= $\begin{bmatrix} 1, -2 \\ 1, 4 \end{bmatrix}$
- and point [-0.89,0.45] gets point [-1.8,0.89]. It has the same direction!

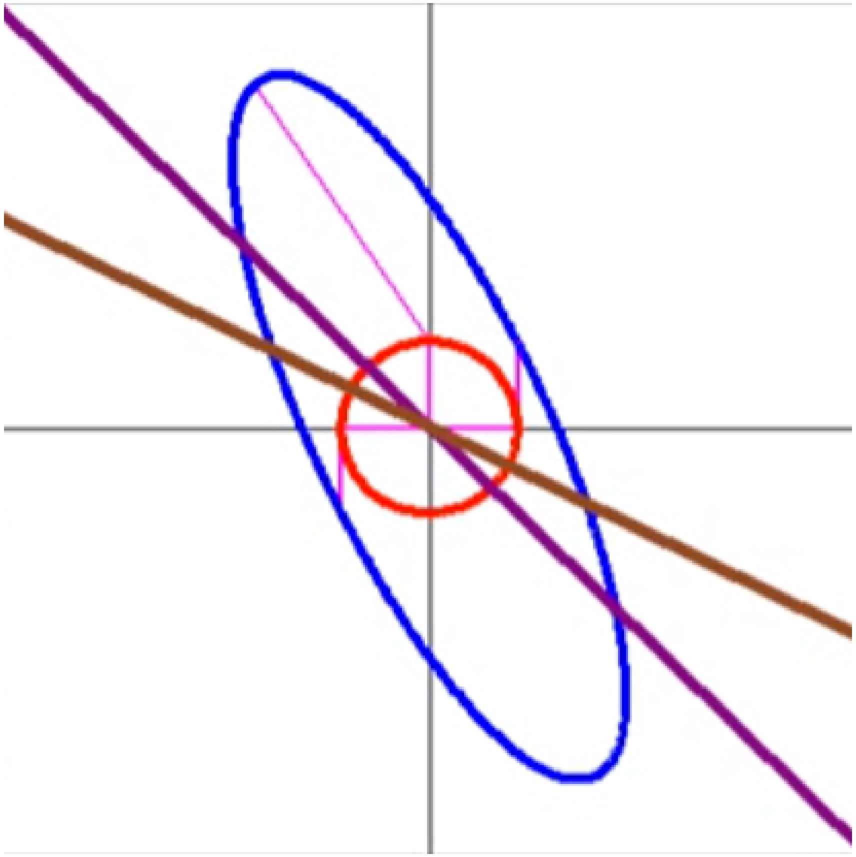


$$\begin{bmatrix} 1, -2 \\ 1, 4 \end{bmatrix} \cdot \begin{bmatrix} -0.71 \\ 0.71 \end{bmatrix} = \begin{bmatrix} -2.1 \\ 2.1 \end{bmatrix} = \overset{\textbf{eigenvalue}}{3} \overset{\textbf{eigenvector}}{\begin{bmatrix} -0.71 \\ 0.71 \end{bmatrix}}$$

$$\begin{bmatrix} 1, -2 \\ 1, 4 \end{bmatrix} \cdot \begin{bmatrix} -0.89 \\ 0.45 \end{bmatrix} = \begin{bmatrix} -1.8 \\ 0.89 \end{bmatrix} = 2 \begin{bmatrix} -0.89 \\ 0.451 \end{bmatrix}$$

PCA: Eigenvectors & Eigenvalues

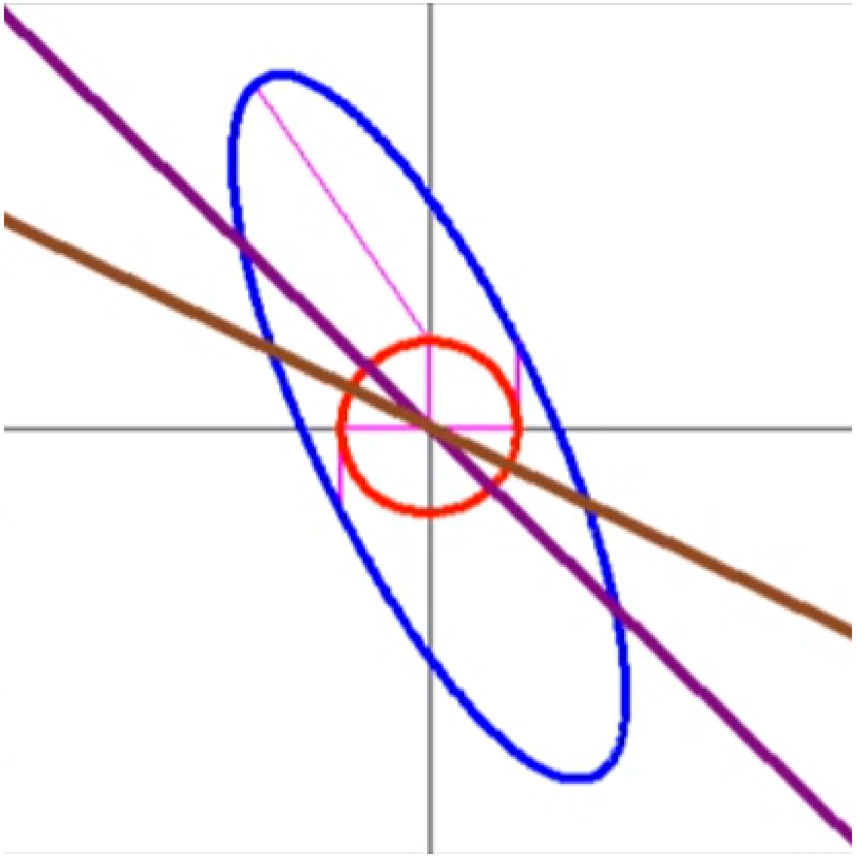
- **Geometric interpretation:**
- What are the eigenvectors & eigenvalues of the matrix A= $\begin{bmatrix} 1, -2 \\ 1, 4 \end{bmatrix}$
- All points on the eigenvector directions remain on it and get scaled by the eigenvalues:



PCA: Eigenvectors & Eigenvalues

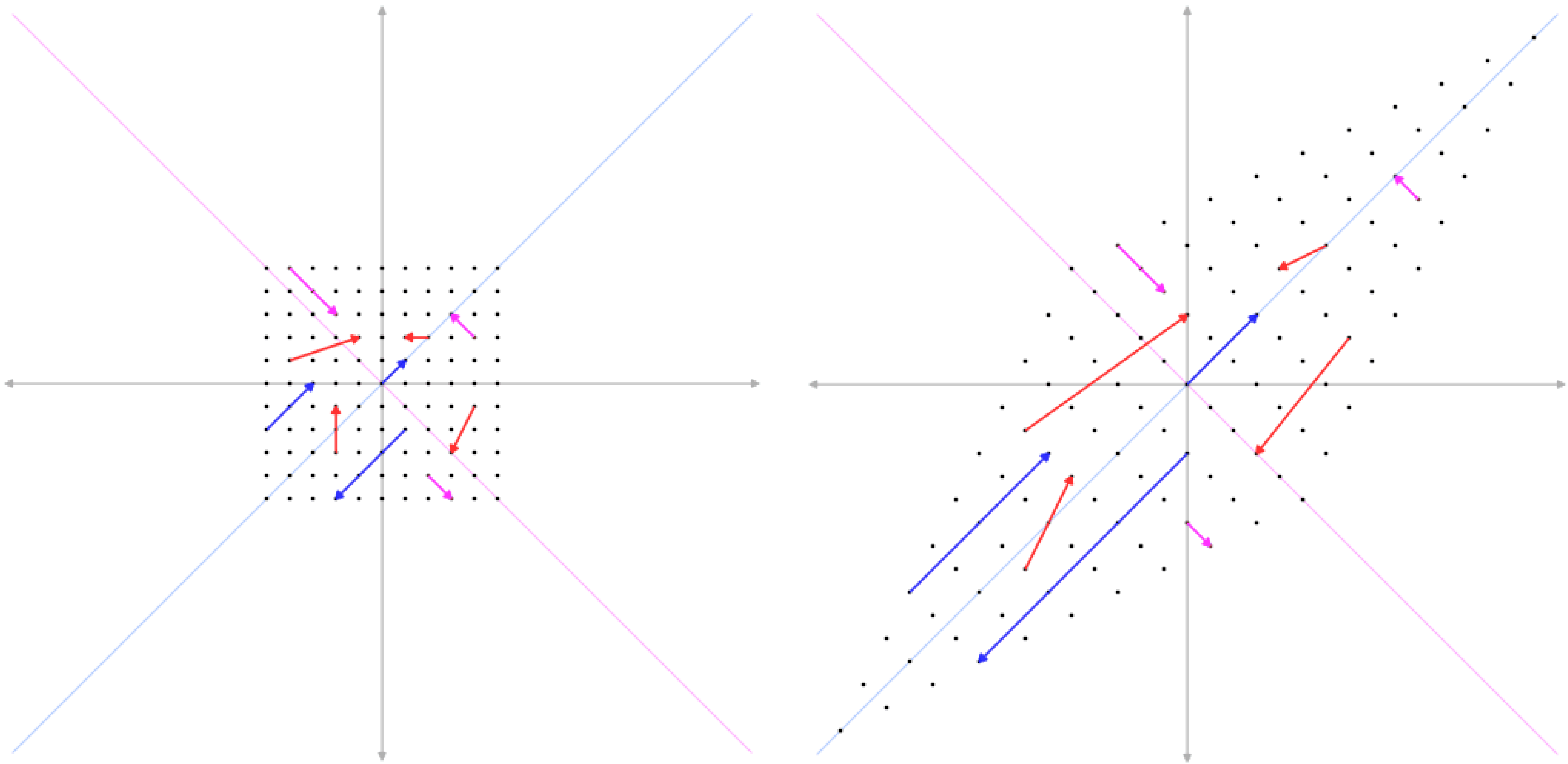
- **Geometric interpretation:**
- What are the eigenvectors & eigenvalues of the matrix A= $\begin{bmatrix} 1, -2 \\ 1, 4 \end{bmatrix}$

- The eigenvectors are here not orthogonal because of the **asymmetric** matrix.



PCA: Eigenvectors & Eigenvalues

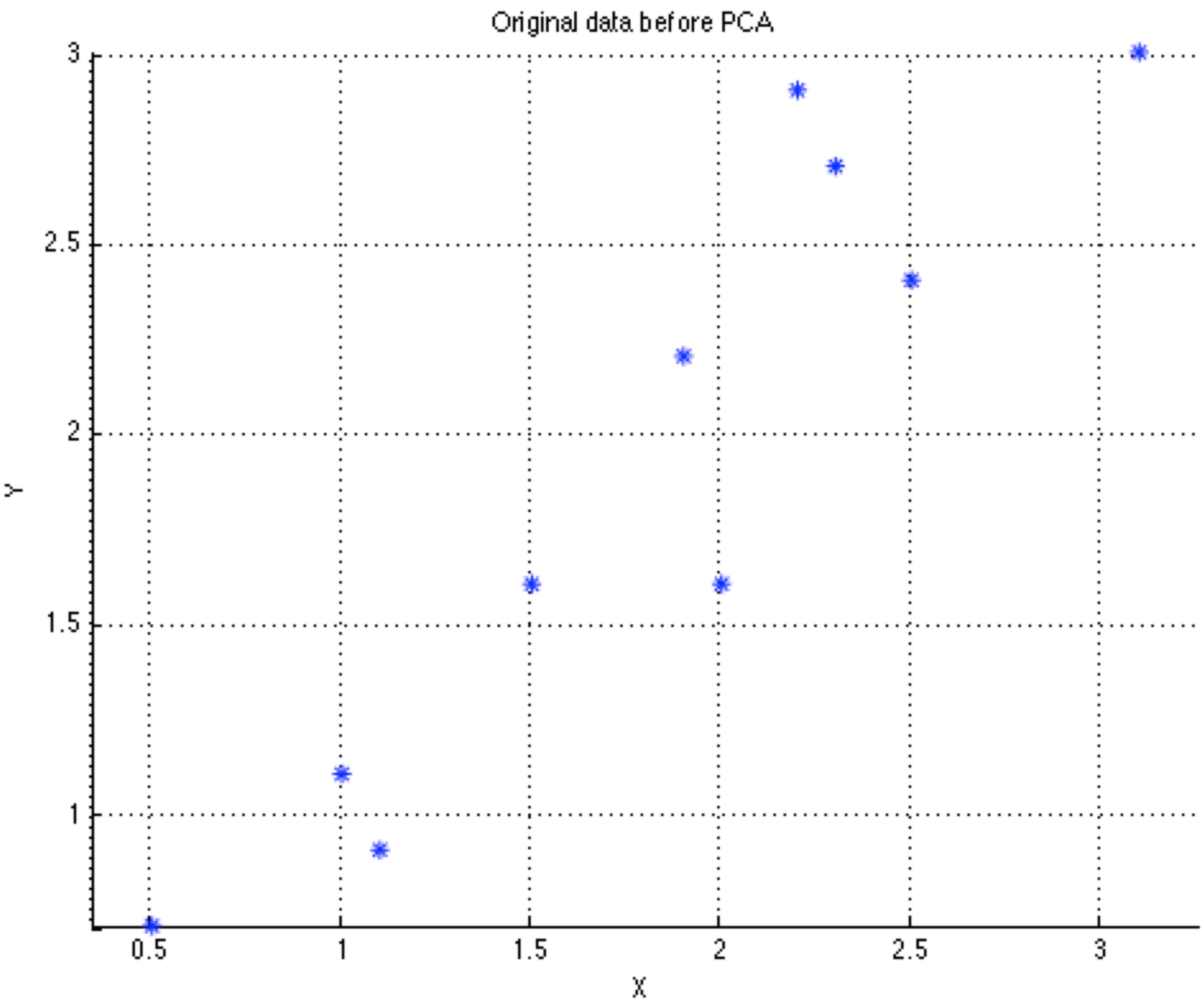
- The transfomration matrix $\begin{bmatrix} 2,1 \\ 1,2 \end{bmatrix}$ preserves the vectors parallel to the eigenvectors:
- Red vectors are not eigenvectors:



Source: Wikipedia

PCA: Simple Example

- For demo purpose we use a very simple 2D dataset (PCAExample1.m)
- X & Y are columns in the data matrix XY
- X & Y Variable are **positive correlated**:



$$XY = \begin{bmatrix} 2.5, 2.4 \\ 0.5, 0.7 \\ 2.2, 2.9 \\ 1.9, 2.2 \\ 3.1, 3.0 \\ 2.3, 2.7 \\ 2.0, 1.6 \\ 1.0, 1.1 \\ 1.5, 1.6 \\ 1.1, 0.9 \end{bmatrix}$$

PCA: Simple Example

- **Step 1: Covariance Matrix, Eigenvectors & Eigenvalues**
  - From the data matrix XY we build the covariance matrix C:

$$C = \begin{bmatrix} 0.6166, 0.6154 \\ 0.6154, 0.7166 \end{bmatrix}$$

  - From the covariance matrix C we extract the eigenvectors & eigenvalues.
  - Both are packed in matrices:

$$eigvec = \begin{bmatrix} -0.7352, 0.6779 \\ 0.6779, 0.7352 \end{bmatrix} \qquad eigval = \begin{bmatrix} 0.0491, 0 \\ 0, 1.2840 \end{bmatrix}$$

PCA: Simple Example

- **Step 2: Extract & Sort Eigenvalues**
- We extract the eigenvalues from the diagonal and sort them:

$$eigval = \begin{bmatrix} 0.0491\,,\,0 \\ 0\,,\,1.2840 \end{bmatrix} \Rightarrow eigval = \begin{bmatrix} 1.2840 \\ 0.0491 \end{bmatrix}$$

- We sort the eigenvectors accordingly:

$$eigvec = \begin{bmatrix} -0.7352\,,0.6779 \\ 0.6779\,,0.7352 \end{bmatrix} \Rightarrow eigvec = \begin{bmatrix} \boxed{\begin{matrix} 0.6779\,, \\ 0.7352 \end{matrix}} \begin{matrix} -0.7352 \\ ,0.6779 \end{matrix} \end{bmatrix}$$

- The eigenvector of the principal component is now in the left column.

PCA: Simple Example

- **Step 3a: PCA with all eigenvectors**
  - We first **subtract** the mean from the dataset XY:

$$XY = \begin{bmatrix} 2.5\,, 2.4 \\ 0.5\,, 0.7 \\ 2.2\,, 2.9 \\ 1.9\,, 2.2 \\ 3.1\,, 3.0 \\ 2.3\,, 2.7 \\ 2.0\,, 1.6 \\ 1.0\,, 1.1 \\ 1.5\,, 1.6 \\ 1.1\,, 0.9 \end{bmatrix} - \begin{bmatrix} 1.81\,,\, 1.91 \end{bmatrix} = \begin{bmatrix} 0.69\,, 0.49 \\ -1.31\,, -1.21 \\ 0.39\,, 0.99 \\ 0.09\,, 0.29 \\ 1.29\,, 1.09 \\ 0.49\,, 0.79 \\ 0.19\,, -0.31 \\ -0.81\,, -0.81 \\ -0.31\,, -0.31 \\ -0.71\,, -1.01 \end{bmatrix}$$

  - Then we **transpose** the shifted dataset:

$$XY0^T = \begin{bmatrix} 0.69\,, -1.31\,, 0.39\,, 0.09\,, 1.29\,, 0.49\,, 0.19\,, -0.81\,, -0.31\,, -0.71 \\ 0.49\,, -1.21\,, 0.99\,, 0.29\,, 1.09\,, 0.79\,, -0.31\,, -0.81\,, -0.31\,, -1.01 \end{bmatrix}$$

PCA: Simple Example

- **Step 3a: PCA with all eigenvectors**
  - We also transpose the eigenvectors.
    The major eigenvector is now in the top row:

$$eigvec = \begin{bmatrix} \boxed{\begin{matrix} 0.6779 \\ 0.7352 \end{matrix}} & \begin{matrix} , -0.7352 \\ , 0.6779 \end{matrix} \end{bmatrix} \Rightarrow eigvec^T = \begin{bmatrix} \boxed{0.6779 , 0.7352} \\ -0.7352 , 0.6779 \end{bmatrix}$$

  - The PCA is now **a simple matrix multiplication**:

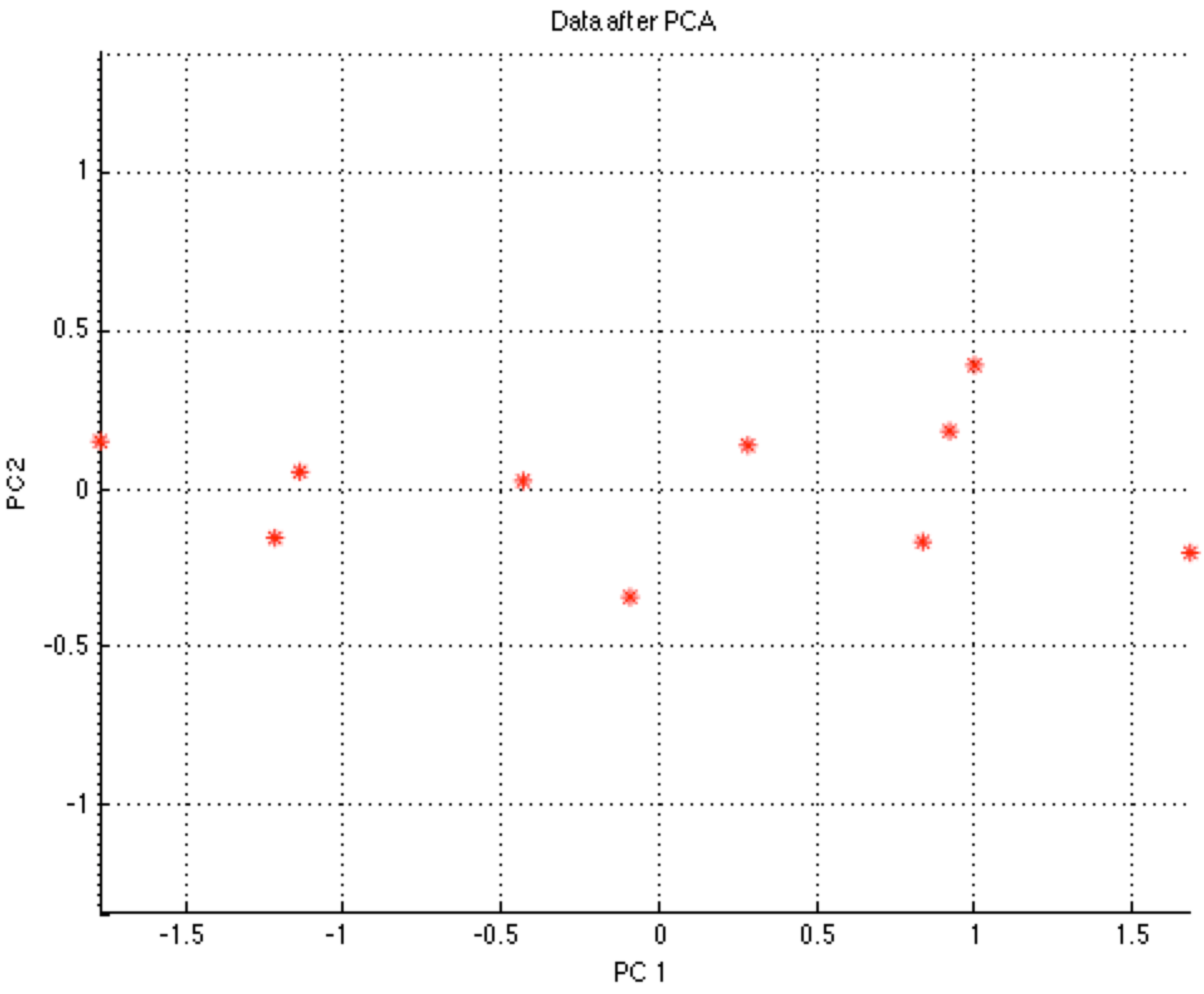
$$XY2 = eigvec^T \cdot XY0^T$$

$$XY2 = \begin{bmatrix} 0.6779 , 0.7352 \\ -0.7352 , 0.6779 \end{bmatrix} \cdot \begin{bmatrix} 0.69 , -1.31 , 0.39 , 0.09 , 1.29 , 0.49 , 0.19 , -0.81 , -0.31 , -0.71 \\ 0.49 , -1.21 , 0.99 , 0.29 , 1.09 , 0.79 , -0.31 , -0.81 , -0.31 , -1.01 \end{bmatrix}$$

$$XY2 = \begin{bmatrix} 0.8280 , -1.7776 , 0.9922 , 0.2742 , 1.6758 , 0.9129 , -0.0991 , -1.1446 , -0.4380 , -1.2238 \\ -0.1751 , 0.1429 , 0.3844 , 0.1304 , -0.2095 , 0.1753 , -0.3498 , 0.0464 , 0.0178 , -0.1627 \end{bmatrix}$$

PCA: Simple Example

- **Step 3a: PCA with all eigenvectors**
  - For visualization we have to transpose it back to a column data matrix.
  - The **x-axis** became the **principal component**.
  - PC1 & PC2 values are now **decorrelated**:



$$XY2 = \begin{bmatrix} 0.8280, -0.1751 \\ -1.7776, 0.1429 \\ 0.9922, 0.3844 \\ 0.2742, 0.1304 \\ 1.6758, -0.2095 \\ 0.9129, 0.1753 \\ -0.0991, -0.3498 \\ -1.1446, 0.0464 \\ -0.4380, 0.0178 \\ -1.2238, -0.1627 \end{bmatrix}$$

PCA: Simple Example

- **Step 3b: PCA with only the major eigenvector**
  - We simply multiply only the major component (the top row) with the dataset $XY0^T$:

$$XY1 = eigvec(:1)^T \cdot XY0^T$$

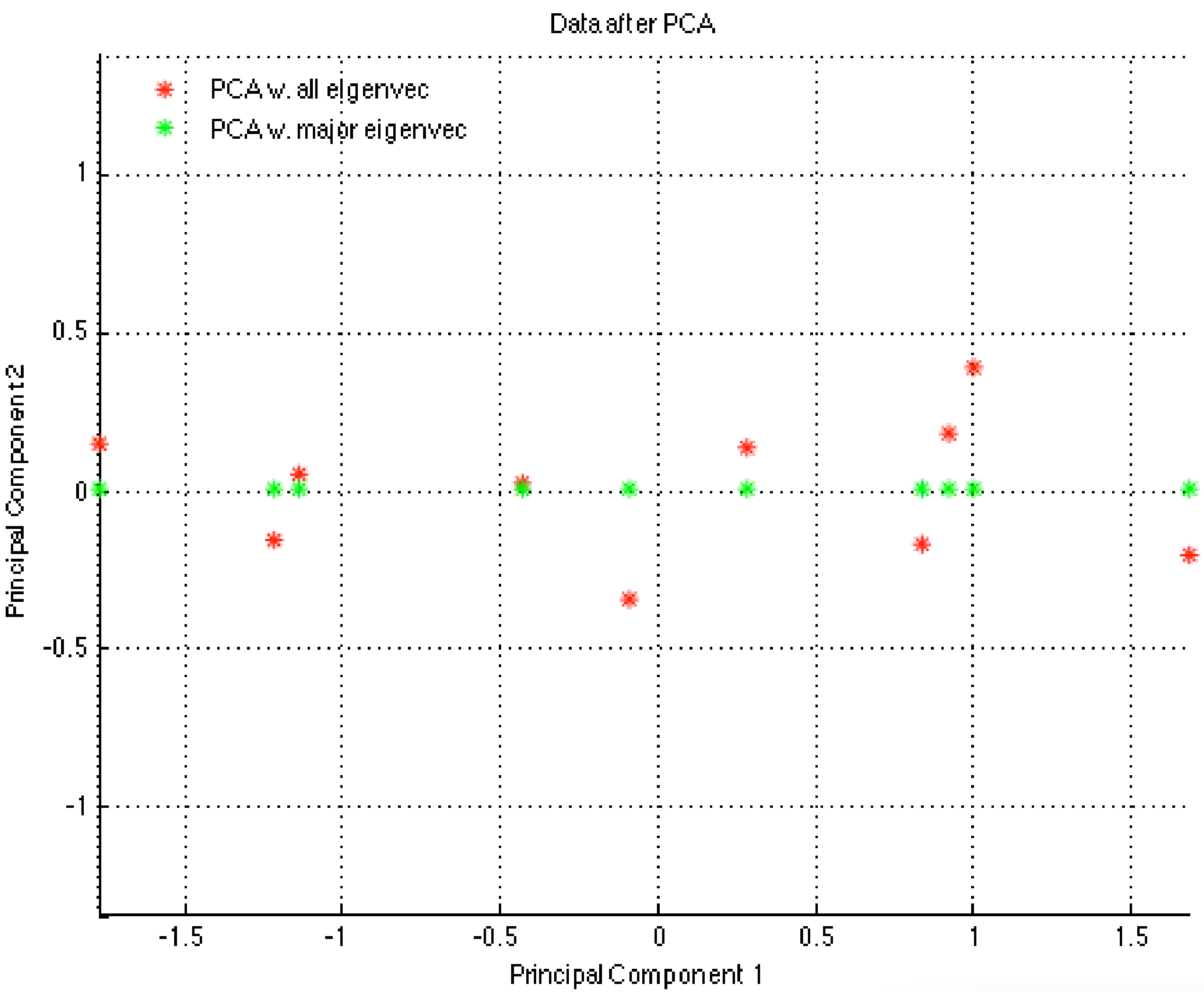
$$XY1 = [0.6779, 0.7352] \cdot \begin{bmatrix} 0.69, -1.31, 0.39, 0.09, 1.29, 0.49, 0.19, -0.81, -0.31, -0.71 \\ 0.49, -1.21, 0.99, 0.29, 1.09, 0.79, -0.31, -0.81, -0.31, -1.01 \end{bmatrix}$$

  - We loose one dimension (the y-values):

$$XY1 = [0.69, -1.31, 0.39, 0.09, 1.29, 0.49, 0.19, -0.81, -0.31, -0.71]$$

PCA: Simple Example

- **Step 3b: PCA with only the major eigenvector**
  - For visualization we have to transpose it back to a column data matrix.
  - **All values** lie on the **x-axis** (the **principal component**):



$$XY1 = \begin{bmatrix} 0.8280 \\ -1.7776 \\ 0.9922 \\ 0.2742 \\ 1.6758 \\ 0.9129 \\ -0.0991 \\ -1.1446 \\ -0.4380 \\ -1.2238 \end{bmatrix}$$

PCA: Simple Example

- **Step 4: Reconstruction**
  - Most transforms make only sense if we can back transform the data.
    - **Foreward** <> **Backward**
    - **Analysis** <> **Synthesis**
    - **PCA**: **P**rincipal **C**omponent **A**nalysis
    - **PCS**: **P**rincipal **C**omponent **S**ynthesis

PCA: Simple Example

- **Step 4: Reconstruction**
  - We simply **use the inverse of the forward eigenvector** matrix.
  - Because it is a **linear transform** we get the inverse by **transposing** it:
    - For the full back transform:

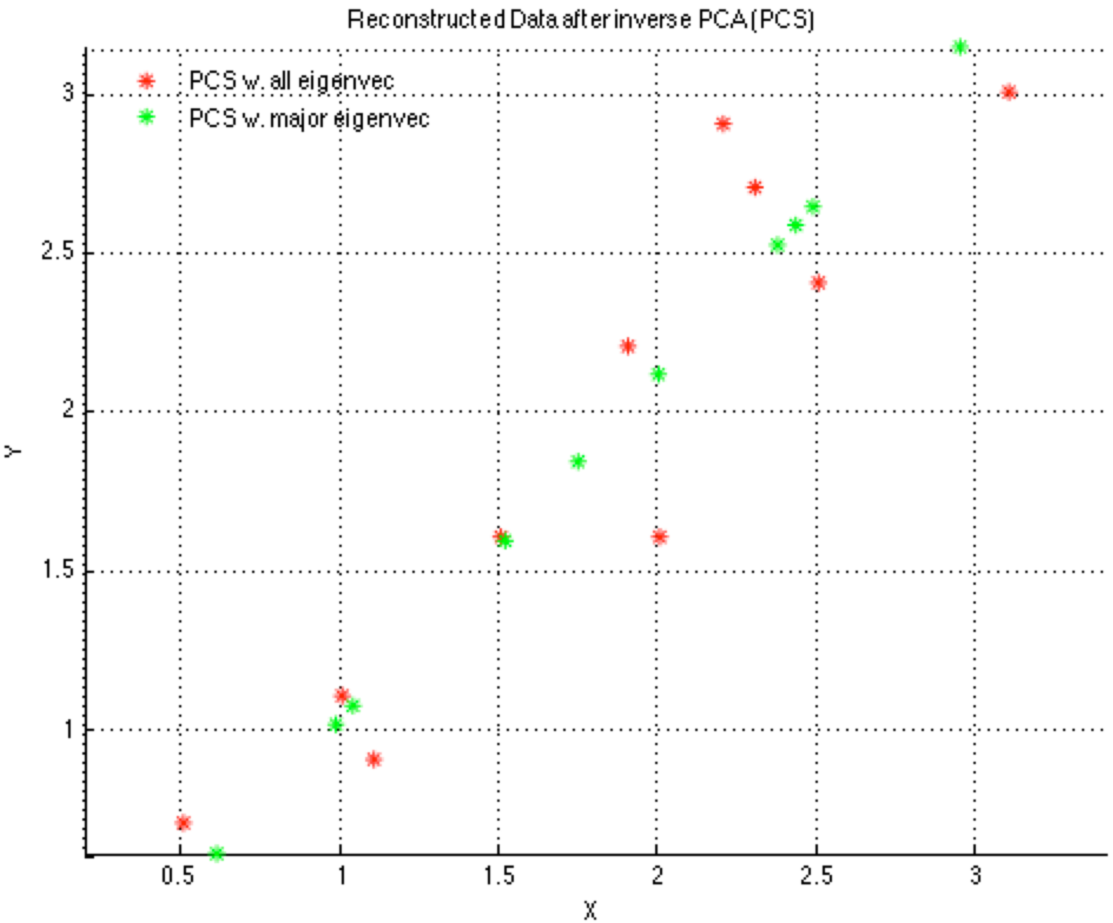
$$inverse\left(eigvec^T\right)=inverse\left(\begin{bmatrix} 0.6779, 0.7352 \\ -0.7352, 0.6779 \end{bmatrix}\right)=\begin{bmatrix} 0.6779, -0.7352 \\ 0.7352, 0.6779 \end{bmatrix}$$

    - For the back transform with only the major eigenvector:

$$inverse\left(eigvec(:1)^T\right)=inverse\left(\begin{bmatrix} 0.6779, 0.7352 \end{bmatrix}\right)=\begin{bmatrix} 0.6779 \\ 0.7352 \end{bmatrix}$$

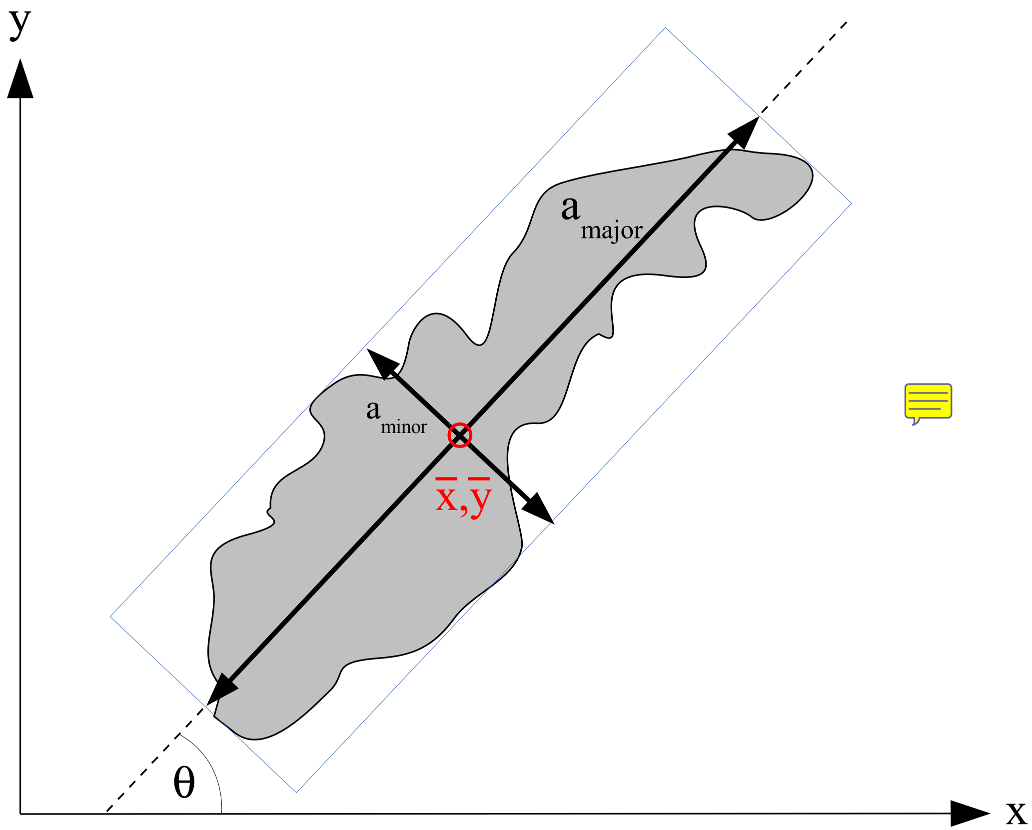
PCA: Simple Example

- **Step 4: Reconstruction**
  - We have to transpose and add back the mean.
  - The original data has no data loss.
  - The principal component data lost its 2nd component:



PCA Applications: Determine Orientation & Axis

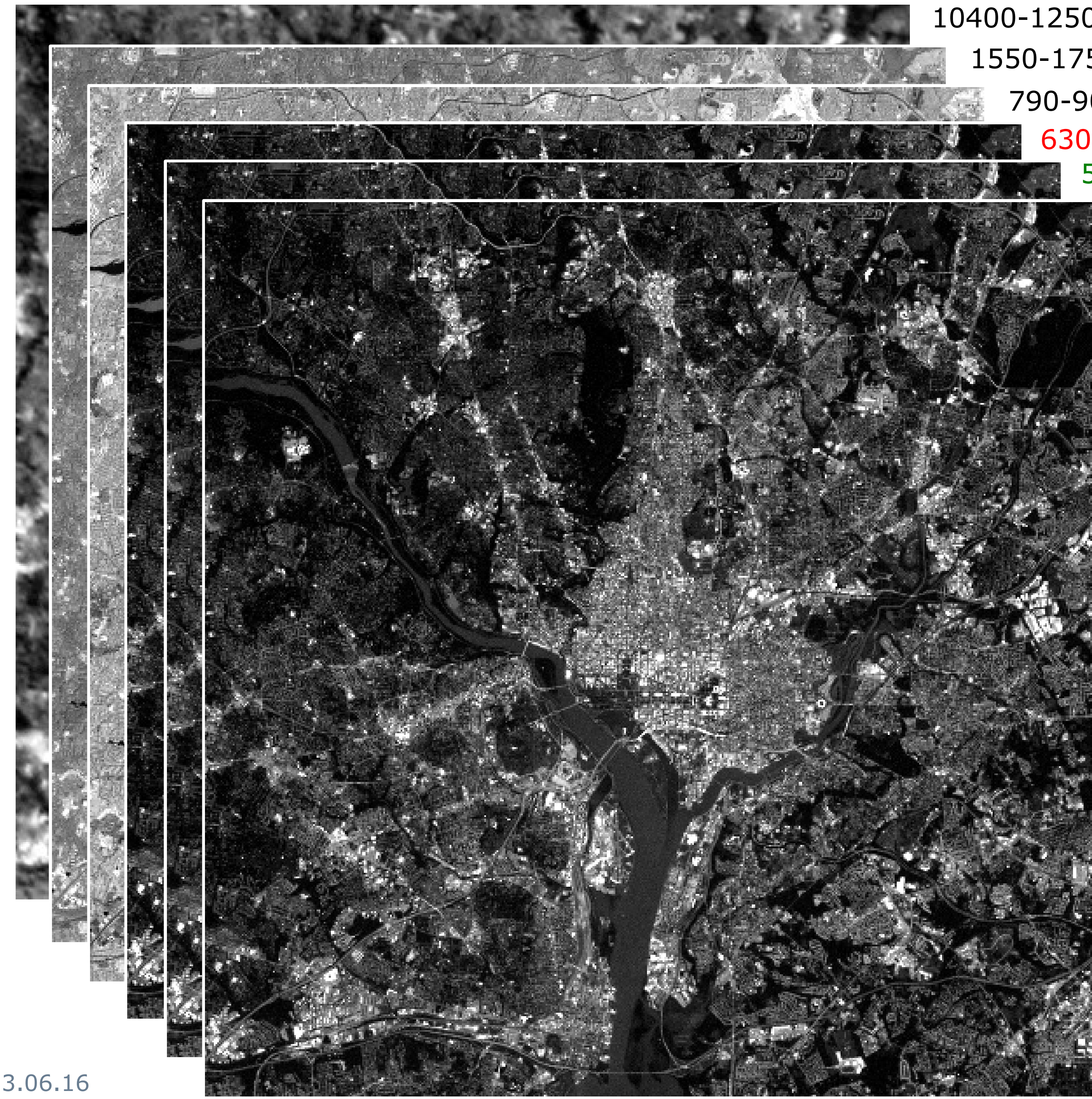
- For a square & symmetric n x n Matrix you get n eigenvectors & values.
- Eigenvectors are allways **orthogonal** & correspond to **the major axis**.
- Eigenvectors are normally returned as **unit vectors**.
- Eigenvalues correspond to **standard deviation σ=1**



See PCAExample2.m

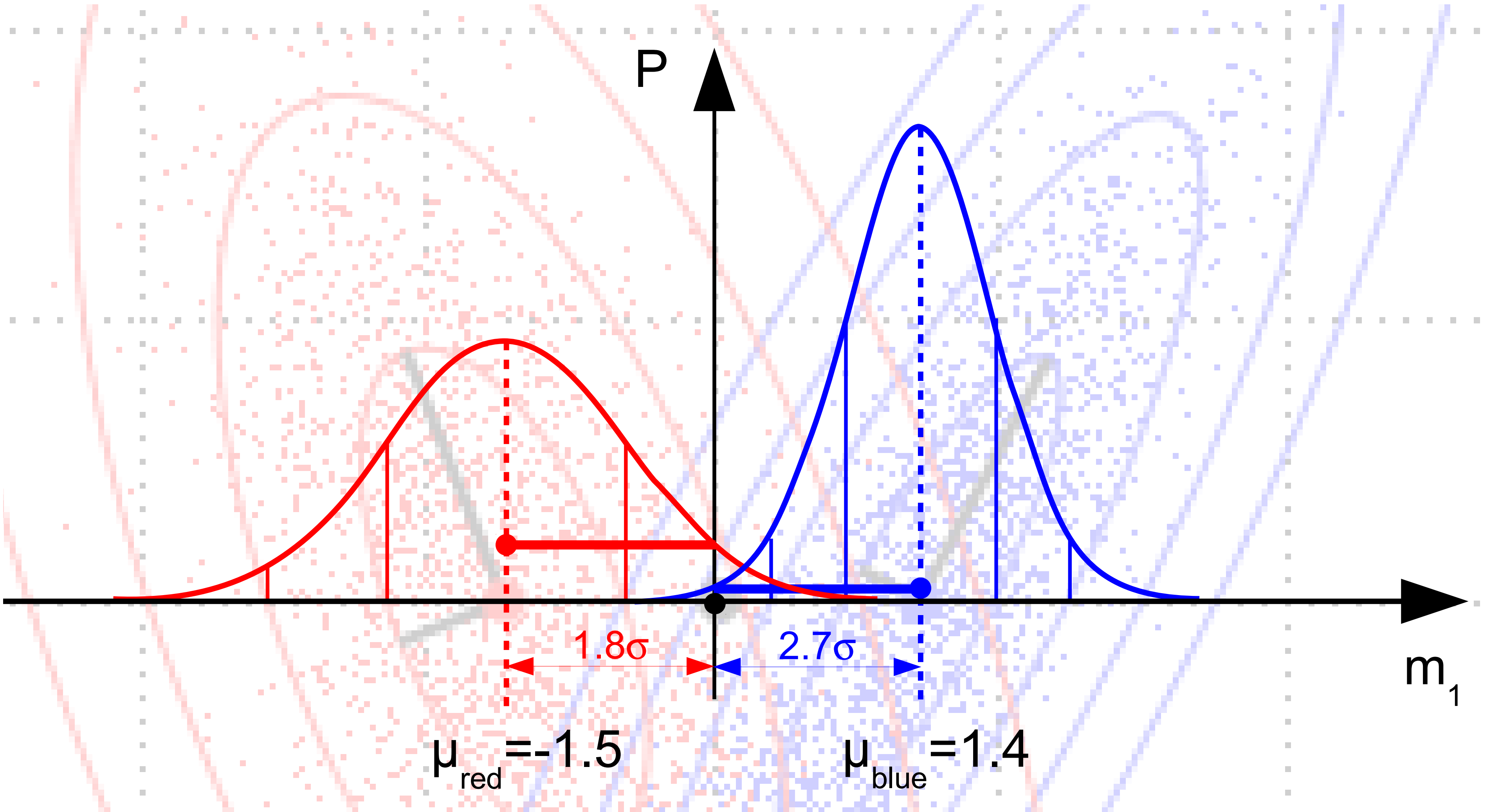
PCA Applications: Dimensionality Reduction

- We can use the PCA to reduce the six-dimensional satellite data:



PCA Applications: Statistical Classification

- We can use the PCA for better classification using the **probability density for the class assignment**:



PCA Applications: Pattern Matching

- We can use the PCA for **Face Recognition**: