Queen Mary University of London

EBU6018
Advanced Transform Methods

Week 4.1 – Linear Transform Coding

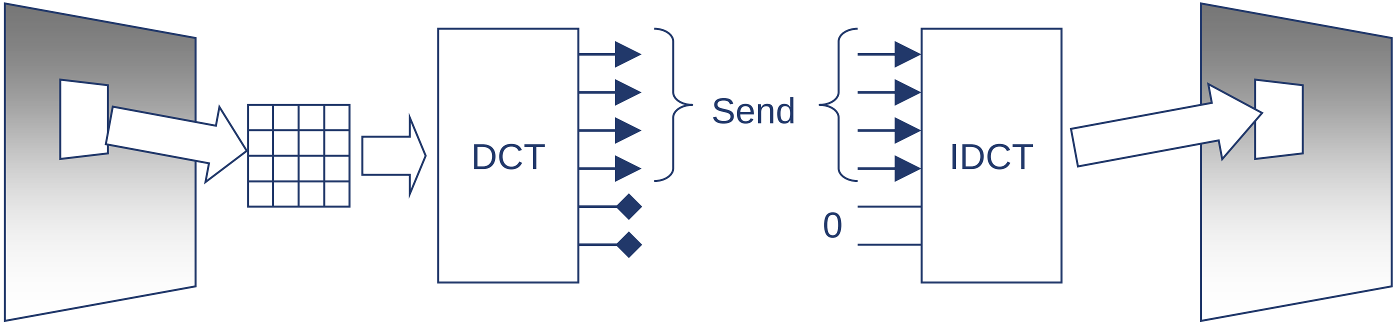
Dr Yixuan Zou

Lecture Outline

- Linear Transform Coding
- Principal Component Analysis (PCA)
- Karhunen-Loeve Transform (KLT)
- KLT/PCA practical applications
  - Image compression
  - Facial recognition

Linear Transform Coding

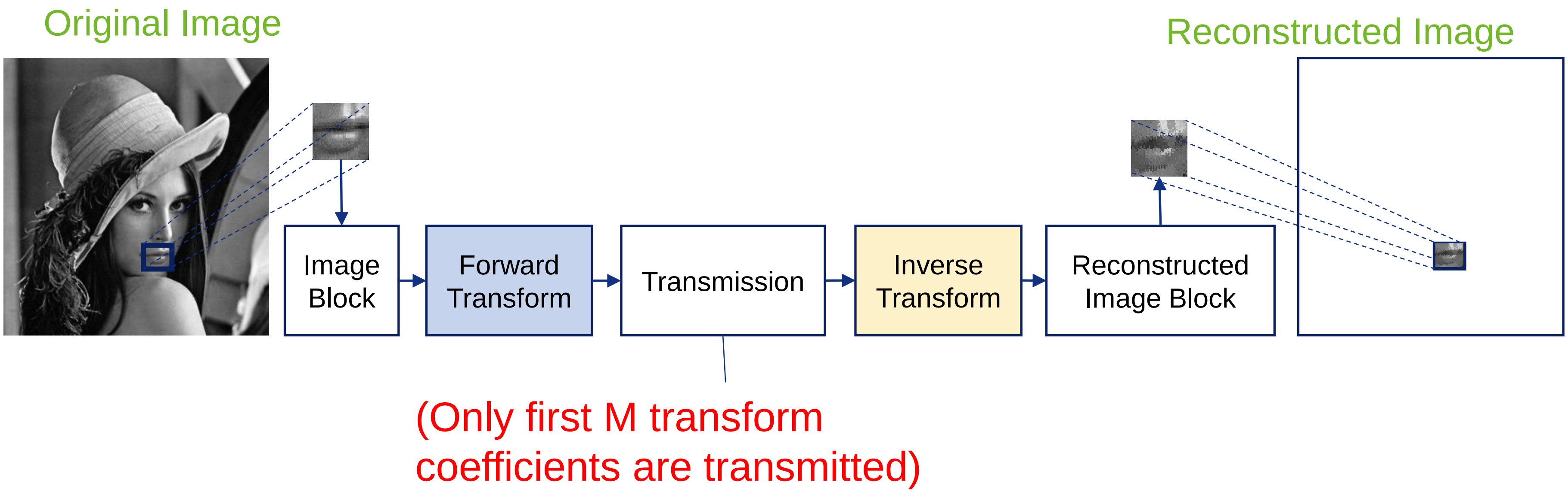
- Discrete Cosine Transform (DCT) is a type of linear transform coding
- Advantages of DCT
  - most energy concentrated in a few coefficients, so
  - can discard some coeffs, while keeping most of signal



- Fourier transform, wavelet transform can also achieve this, depending on the signal.
- This type of application is known as Linear Transform Coding

Linear Transform Coding

- General procedures of linear transform coding:



Linear Transform Coding

- General procedures of linear transform coding (in words):

  1. **Divide image (or signal) into P blocks of N pixels (samples).**
     - The k-th block is now an $N-$dimensional vector:
       $$\mathbf{x}_k = \left(x_{1,k},\ x_{2,k},\ \dots,\ x_{N,k}\right)^T$$
     - The image (signal) is now a sequence of vectors $\{\mathbf{x}_k\}$.
  2. **Transform $\mathbf{x}_k$ by multiplying by a linear matrix**
     $$\mathbf{y}_k = \mathbf{A}\mathbf{x}_k$$
  3. **Transmit the first M coefficients $\hat{\mathbf{y}}_k = \left(y_{1,k}, \dots, y_{M,k}\right)^T$**, discarding the remaining $N-M$ coeffs $y_{M+1,k} \cdots y_{N,k}$
  4. **Reconstruct the image block** using another matrix
     $$\hat{\mathbf{x}}_k = \mathbf{B}\hat{\mathbf{y}}_k$$
  5. Repeat steps 2-4 for each image block to transmit the whole image

Linear Transform Coding

- The transmission error is calculated as
  - $J = E(|\mathbf{x} - \hat{\mathbf{x}}|^2)$
  - Mean squared error (MSE)
  - $E(v)$ is the expected value (mean) of v
- We want to choose A (forward transform matrix) and B (inverse transform matrix) to minimize J
- <u>Easy case</u>: if we keep all the coefficients, we have
  - $\hat{\mathbf{y}} = \mathbf{y}$, hence $\hat{\mathbf{x}} = \mathbf{B}\hat{\mathbf{y}} = \mathbf{B}\mathbf{y} = \mathbf{B}A\mathbf{x}$
  - Therefore, $\hat{\mathbf{x}} = \mathbf{x}$ (zero error) if $\mathbf{BA} = \mathbf{I}$, i.e., $\mathbf{B} = \mathbf{A}^{-1}$

Principal Component Analysis (PCA) – The basis for KLT

- Multivariate procedure
- Main use of PCA is to reduce dimensionality of a data set while retaining as much information as is possible.
- Finds a projection of the observations onto orthogonal axes contained in the space defined by the original variables.
- Correlated variables transformed into uncorrelated variables, known as principal components
    - Ordered by reducing variability.
    - Uncorrelated variables are linear combinations of original variables
- Computes compact, optimal description of data set.
- Rotates data so that maximum variabilities projected onto the axes
- Rotation of existing axes to new positions in the space defined by the original variables.

Principal Component Analysis (PCA)

1. **First principal component** is the combination of variables that explains the greatest amount of variation
   - ✓ Contains the maximum amount of variation
2. **Second principal component** defines the next largest amount of variation and is independent to the first principal component
   - ✓ Contains the maximum amount of variation unexplained by and orthogonal to the first
3. Third axis contains the maximum amount of variation orthogonal to the first and second axis
   - ✓ Contains the maximum amount of variation unexplained by and orthogonal to the first and second axes

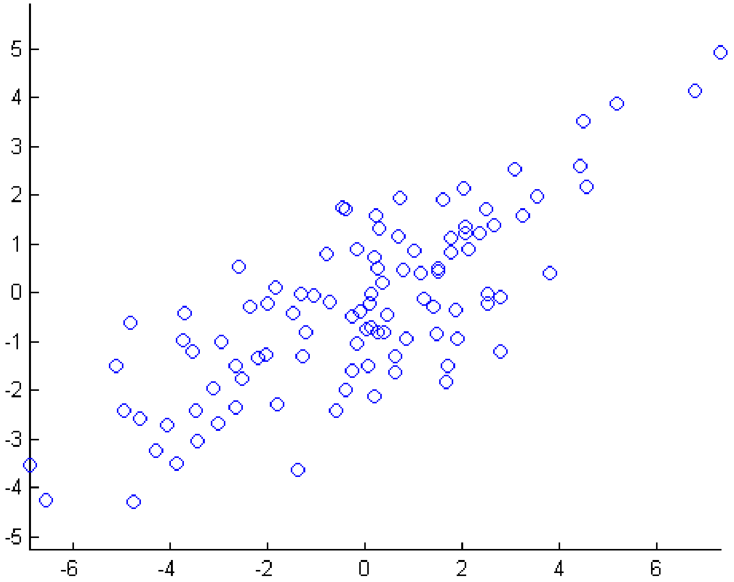
… Last new axis which is the last amount of variation left

- ✓ Can be removed with minimum loss of real data

- ❑ Can be as many principal components as there are variables
- ❑ No correlation between the new variables defined by rotation

Principal Components Visualized

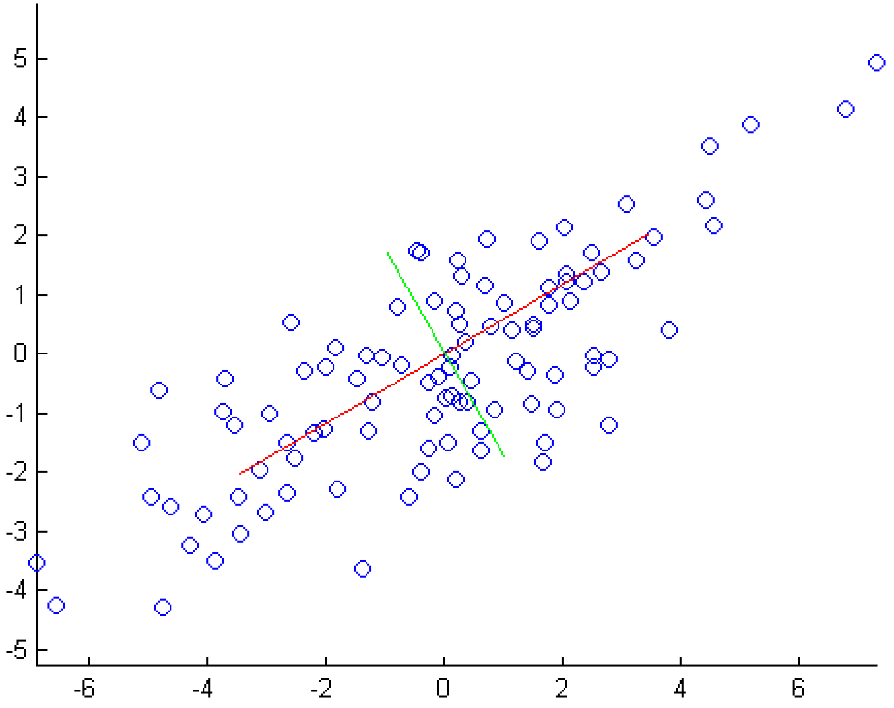
- Consider a sample of a population



$$X = \begin{array}{cc} x_1 & x_2 \\ \boxed{\begin{array}{cc} -3.0139 & -2.6748 \\ 0.1810 & -2.1227 \\ -6.5559 & -4.2424 \\ 1.5960 & 1.9161 \\ 2.7825 & -1.2017 \\ 2.0292 & 2.1421 \\ -2.5200 & -1.7787 \\ 0.2340 & -0.8122 \\ 0.8263 & -0.9407 \\ -4.6227 & -2.5955 \\ -4.3052 & -3.2307 \\ -2.6013 & 0.5435 \\ 3.5409 & 1.9755 \\ 5.1704 & 3.8629 \\ \vdots & \vdots \end{array}} \end{array}$$

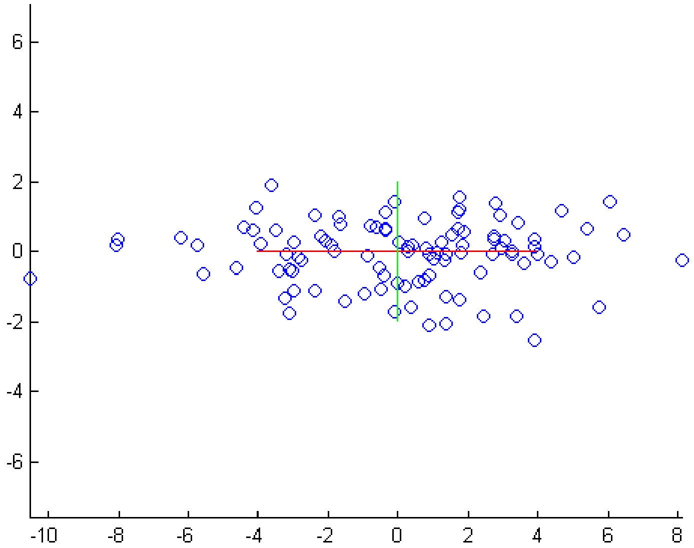
Principal Components Visualized

- red line represents direction of first principal component
  - line of greatest variation
- green line is direction of second principal component
  - perpendicular to red line.
- When there are more than 2 dimensions,
  - next component along line of next greatest variation



Principal Components Visualized

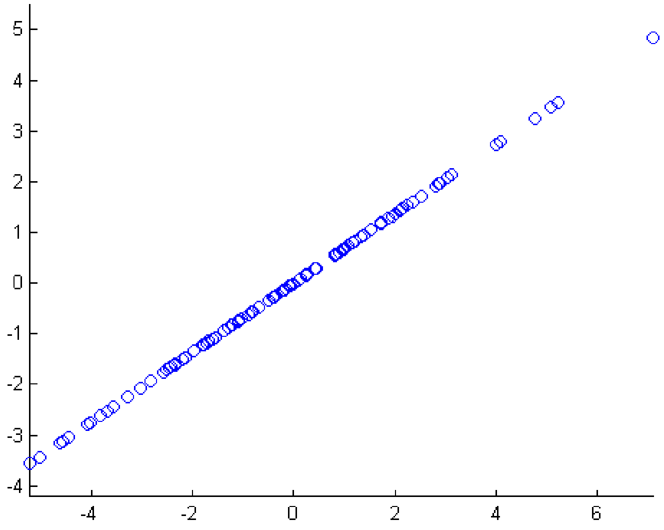
- By multiplying original data by principal components, data rotated so that principal components lie along axes



This is the projection onto the new orthogonal axis

Principal Components Visualized

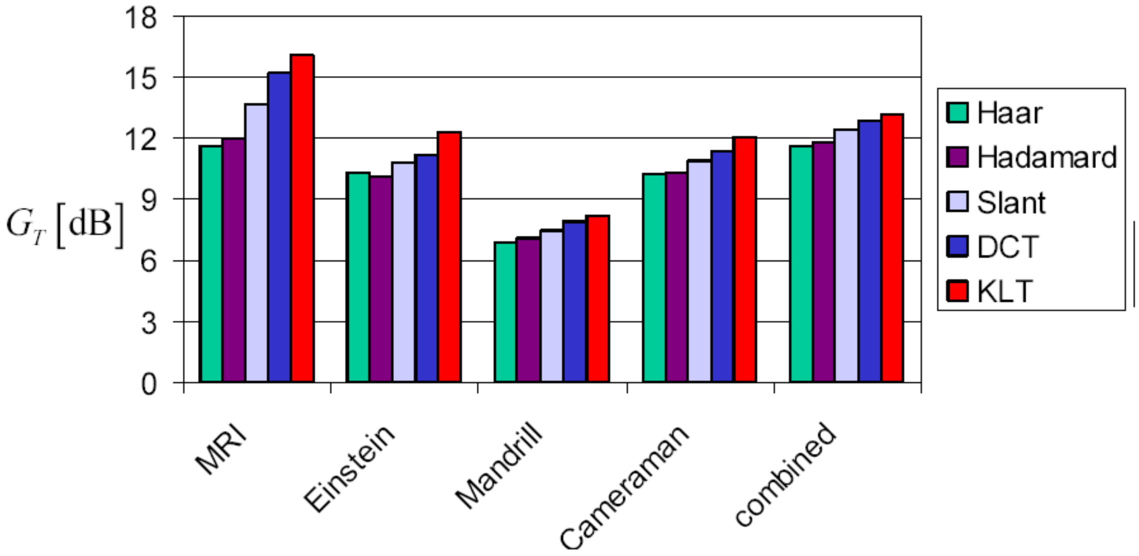
- By dropping the second principal component, the projection back to the original axis is



Karhunen Loève Transform (KLT)

- The generalized transform method for both discrete and continuous data
  - ❑ PCA is the discrete version of KLT
- Basis functions of KLT are eigenvectors of the covariance matrix R_{xx} of the input signal x
  - ❑ Values of the basis vectors (eigenvectors) depend on the data set, i.e., basis vectors are not fixed.
- KLT yields uncorrelated transform coefficients.
- Optimal linear transform for keeping the subspace that has largest variance
  - ❑ Achieves optimum energy concentration.
  - ❑ KLT maximizes coding gain (best “quality” of compressed data for a given level of compression)

Karhunen Loève Transform (KLT) – Coding Gain



* All using 8x8 transform

Eigenvalues/Eigenvectors

- **What** is an Eigenvalue?
  - Eigenvalue comes from German *Eigenwart*
    - *Eigen* means ‘own’
    - *Wart* means ‘value’
  - Eigenvalues can be termed as characteristic values
- **What** is an Eigenvector?
  - Eigenvectors are non-zero vectors that do not change the direction when any linear transformation is applied. They only change by a scalar factor.
- Relationships between eigenvalues and eigenvectors:
  - Let A be a linear transformation from a vector space V
  - The eigenvectors of A are the non-zero vectors $x \in V$, such that $Ax = \lambda x$ for some scalar λ.
  - We refer λ as the eigenvalue associated with eigenvector x

Eigenvalues/Eigenvectors

- **Why** are we interested in eigenvalues/eigenvectors?
  - To decompose matrices into eigenvalues and eigenvectors
  - To understand the key information (principal components) contained by the matrix
  - To compress large matrices through feature extraction algorithms, e.g., PCA.

Eigenvalues/Eigenvectors

- **How** to find the eigenvectors and the eigenvalues of matrix A?
  - To calculate the eigenvalues, find λ that satisfies

$$|A - \lambda I| = 0$$

(I is the identity matrix)

  - To calculate the eigenvectors, find v that satisfies

$$(A - \lambda I)v = 0$$

(Based on the knowledge of λ)

- Only square matrices have eigenvalues/eigenvectors.
- The number of eigenvalues is equal to the dimension of the square matrix
  - i.e., For a $N \times N$ matrix A, it has N eigenvalues
  - The eigenvalues can be repeated

Eigenvalues/Eigenvectors - Examples

- 2 x 2 Matrices
  - Matrix: $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$
  - Eigenvalues: $|A - \lambda I| = \left| \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = \begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = 3 - 4\lambda + \lambda^2 = 0$

    $$\Rightarrow \lambda_1 = 1, \lambda_2 = 3$$
  - Eigenvectors:
    - For $\lambda_1 = 1$, $(A - \lambda_1 I)v_1 = 0 \Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} v_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow v_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$
    - For $\lambda_2 = 3$, $(A - \lambda_2 I)v_2 = 0 \Rightarrow \underbrace{\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} v_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{} \Rightarrow v_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

They are orthogonal!

Any vector that satisfies this relationship is an eigenvector

Variance/Covariance

- **What** is variance/covariance?
  - **Variance** refers to the spread of a data set around its mean value
  - **Covariance** refers to the measure of the directional relationship between two random variables.
  - Variances are always positive-valued, whereas covariances can be either positive or negative
    - A positive covariance indicates two variables are positively-related and they move in the same direction
      - E.g. The number of hours spent studying for an exam and the scores obtained in the exam
    - A negative covariance indicates two variables are inversely-related and they move in the opposite direction

Variance/Covariance

- **How** to compute variance/covariance?
  - **Variance** of dataset $x = \{x_1, \dots, x_N\}$ of size N:

$$Var_x = \frac{\sum_{i=1}^{N}(x_i - x_{ave})^2}{N-1} \qquad x_{ave} = \frac{\sum_{i=1}^{N} x_i}{N}$$

  - **Covariance** between variables x and y of size N:

$$Cov_{x,y} = \frac{\sum_{i=1}^{N}(x_i - x_{ave})(y_i - y_{ave})}{N-1}$$

    - Divide by (N-1) if we have a sample of a population. If the dataset is the complete population, then divide by N.
    - In general, the variance of a complete population will be greater than the variance of a sample. Hence, the variance of the sample is an estimation of the variance of the population.

Covariance Matrix

- **What** is the covariance matrix?
  - It measures how much two variables vary together in a population
  - It describes the relationship (covariance) between every two variables in a dataset
  - If a dataset contains N variables, the covariance matrix has a dimension of $N \times N$
- **Why** is covariance matrix important?
  - It reflects the redundant variables of a population
  - All eigenvectors of a covariance matrix are orthogonal to each other, allowing for the principal component analysis (PCA) to be applied

Covariance Matrix

- **How** to compute the covariance matrix?
  - The **covariance matrix** of a dataset containing two variables is computed by

$$\begin{bmatrix} Var_{x_1} & Cov_{x1,x_2} \\ Cov_{x_2,x_1} & Var_{x_2} \end{bmatrix}$$

  - The **covariance matrix** of a dataset containing $N > 2$ variables is computed by

$$R_{xx} = \begin{bmatrix} Var_{x_1} & Cov_{x_1,x_2} & \cdots & Cov_{x_1,x_N} \\ Cov_{x_2,x_1} & Var_{x_2} & \cdots & Cov_{x_2,x_N} \\ \vdots & \vdots & \ddots & \vdots \\ Cov_{x_N,x_1} & Cov_{x_N,x_2} & \cdots & Var_{x_N} \end{bmatrix}$$

  - It is a symmetric square matrix

Karhunen Loève Transform (KLT) - Procedures

$$\boxed{\mathbf{X} = [\vec{x}_0, \vec{x}_1, \ldots \vec{x}_{N-1}]}$$

*Each row of X stands for one variable

1. Find mean vector for input data

$$E(\mathbf{X}) = \frac{1}{N}\sum_{i=0}^{N-1} \vec{x}_i$$

2. Find covariance matrix

$$\mathbf{R_{XX}} = \frac{1}{N-1}\sum_{i=0}^{N-1} (\vec{x}_i - E(\vec{x}))(\vec{x}_i - E(\vec{x}))^T$$

Karhunen Loève Transform (KLT) - Procedures

$$\boxed{\mathbf{X} = [\vec{x}_0, \vec{x}_1, \ldots \vec{x}_{N-1}]}$$

3. Find eigenvalues of the covariance matrix

$$|\mathbf{R}_{\mathbf{XX}} - \lambda \mathbf{I}| = 0$$

4. Find eigenvectors of the covariance matrix

$$(\mathbf{R}_{\mathbf{XX}} - \lambda_i \mathbf{I})\vec{\varphi}_i = 0$$

Queen Mary University of London

Karhunen Loève Transform (KLT) - Procedures

$$\boxed{\mathbf{X} = [\vec{x}_0, \vec{x}_1, \ldots \vec{x}_{N-1}]}$$

5. Normalise the eigenvectors

$$\vec{\varphi}^*{}_i = \frac{\vec{\varphi}_i}{|\vec{\varphi}_i|} \qquad \text{so that } \langle \vec{\varphi}_i, \vec{\varphi}_i \rangle = 1$$

- Each column of φ is an eigenvecter, arranged in descending order of their eigenvalue

6. Transform the input

$$\mathbf{Y} = \varphi^T \mathbf{X}, \text{ where } \varphi = [\vec{\varphi}^*{}_1, \vec{\varphi}^*{}_2, \ldots]$$

To check, find covariance matrix of $\mathbf{Y}$

$$\mathbf{R_{YY}} = \frac{1}{N-1} \sum_{i=0}^{N-1} (\vec{y}_i - E[\vec{y}])(\vec{y}_i - E[\vec{y}])^T$$

- Data X is rotated to Y so that principal components lie along axes
- Covariances between the new variables should be zero

Karhunen Loève Transform (KLT) - Procedures

$$\boxed{\mathbf{X} = [\vec{x}_0, \vec{x}_1, \ldots \vec{x}_{N-1}]}$$

To compress the data (Method 1)

1. Set last row vector(s) of $\mathbf{Y}$ to 0

$$\mathbf{Y} \rightarrow \mathbf{Y}'$$

2. Apply inverse transform

$$\mathbf{X}' = \varphi \mathbf{Y}'$$

Karhunen Loève Transform (KLT) - Procedures

$$\boxed{\mathbf{X} = [\vec{x}_0, \vec{x}_1, \ldots \vec{x}_{N-1}]}$$

To compress the data (Method 2)

1. Set last column vector(s) of φ to 0

$$\varphi \rightarrow \varphi'$$

2. Apply transformation to $\mathbf{X}$

$$\mathbf{Y}' = \varphi'^T \mathbf{X}$$

Achieve at the same $\mathbf{Y}'$ as in method 1

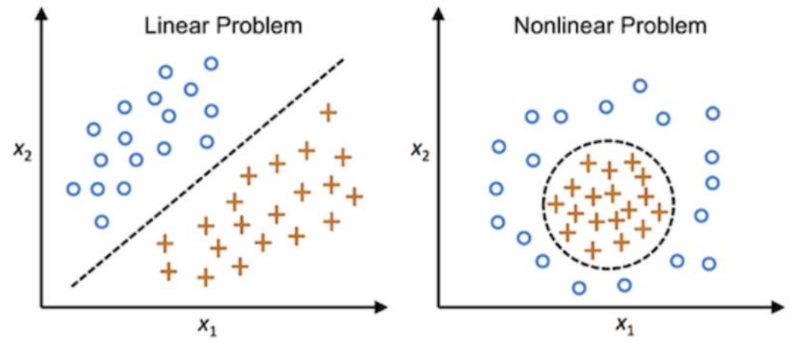
3. Apply inverse transform

$$\mathbf{X}' = \varphi' \mathbf{Y}'$$

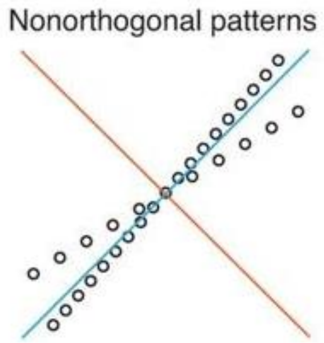
Drawbacks of KLT

- Examples where KLT may not work well:

Non-linear Pattern



Non-orthogonal Pattern



[1] "Introduction to Kernal PCA with Python", *Python and R Tips*, link.
[2] "Principal Component Analysis", *Devopedia*, link

Drawbacks of KLT

KLT is theoretically optimal (in the MSE sense). (KLT maximises the coding gain, i.e. maximises the SNR after a given level of compression.)

BUT, it has practical difficulties:

- Estimate of correlation can be unwieldy
- Solution of eigenvector decomposition is computationally intensive (i.e. slow)
- Calculation of forward and inverse transforms is $O(MN)$ for each image block
- Transmission of data-dependent basis $\mathbf{A}$ is required
- The technique is linear, therefore any non-linear correlation between variables will not be captured.

In comparison, turns out that DCT is fixed, and

- a good approximation to KLT for typical images,
- needs no eigenvalue decomposition, and
- transform is $O(N \log N)$.

Karhunen Loève Transform (KLT) - Quiz

Question 1

What is the correct order of KLT based on the labelled steps?

a. FEDBAC
b. EFDBAC
c. FEBDAC
d. EFBDAC

A. Normalize the eigenvectors

B. Find the eigenvectors of the covariance matrix

C. Transform the input

D. Find the eigenvalues of the covariance matrix

E. Calculate the covariance matrix

F. Calculate the mean vector

Karhunen Loève Transform (KLT) - Quiz

Question 1

What is the correct order of KLT based on the labelled steps?

a. FEDBAC Correct !
b. EFDBAC
c. FEBDAC
d. EFBDA[C]

1. Calculate the mean vectors before the covariance matrix
2. Calculate the eigenvalues before the eigenvectors

A. Normalize the eigenvectors

B. Find the eigenvectors of the covariance matrix

C. Transform the input

D. Find the eigenvalues of the covariance matrix

E. Calculate the covariance matrix

F. Calculate the mean vector

Karhunen Loève Transform (KLT) - Quiz

<u>Question 2</u>

Which of the following is true?

a. KLT is a non-linear transformation
b. KLT maximizes coding gain
c. KLT reduces the number of variables in the dataset
d. KLT is easy to compute

Karhunen Loève Transform (KLT) - Quiz

Question 2

Which of the following is true?

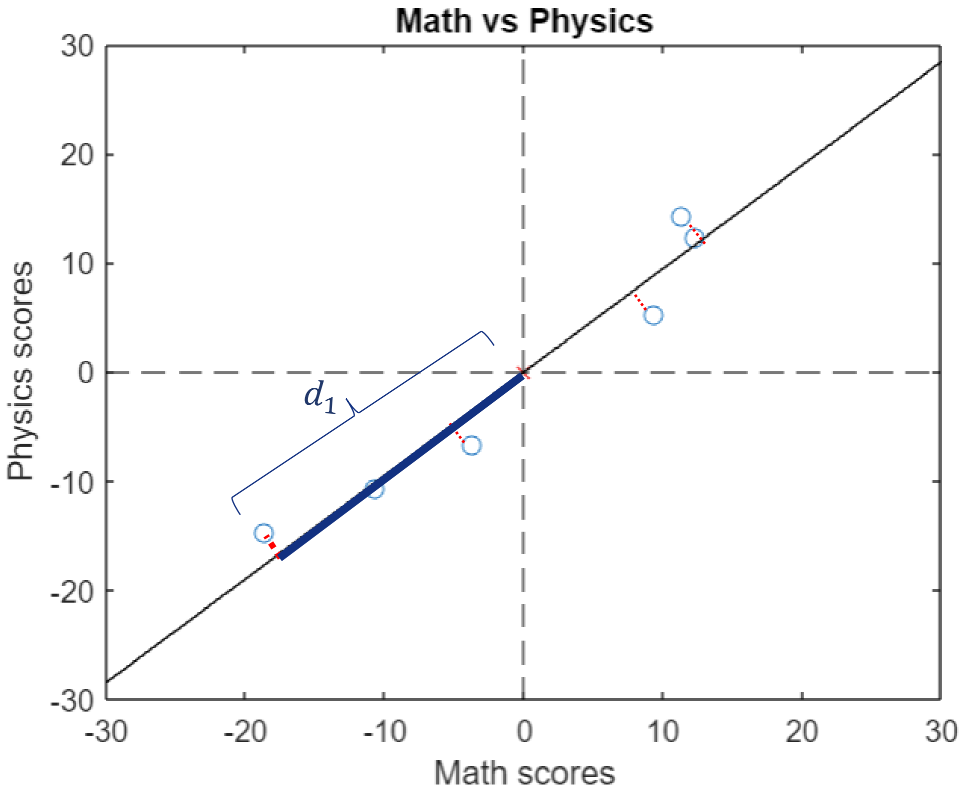
KLT is linear

a. KLT is a non-linear transformation
b. KLT maximizes coding gain   Correct !
c. KLT reduces the number of variables in the dataset
d. KLT is easy to compute

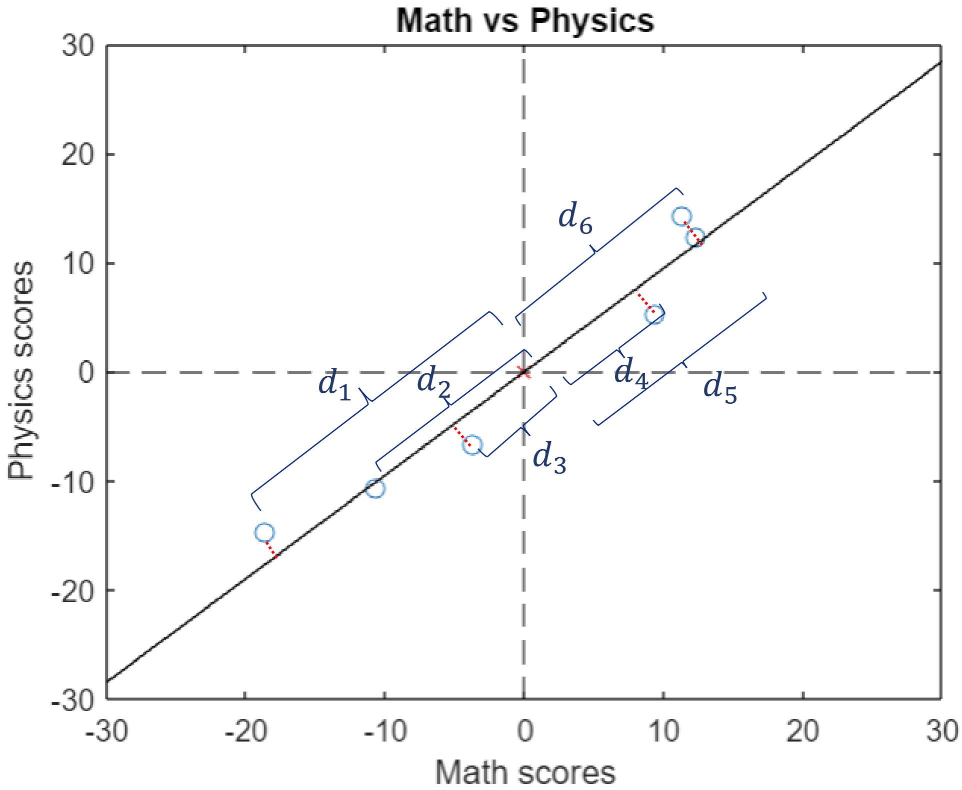
KLT is slow…

KLT reduces the number of PCs not variables

Visualize eigenvalues



Visualize eigenvalues



- Eigenvalue of this principal component is the sum of squared distance when data points are projected onto the line

$$\lambda_1 = d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$$

- Hence, the eigenvalue can be interpreted as the amount of variation that is explained by the principal component

KLT - Example

- Determine the KLT of the following 2D data set.

$$\mathbf{X}=\begin{bmatrix} 2 & 4 & 5 & 5 & 3 & 2 \\ 2 & 3 & 4 & 5 & 4 & 3 \end{bmatrix}$$

(Assume this is a sample of a larger population)
(Each row represents a feature/variable)

- Then, replace the elements of the least principal component of the output by zero and then perform the inverse KLT

Queen Mary University of London

KLT - Example

$$\mathbf{X} = \begin{bmatrix} 2 & 4 & 5 & 5 & 3 & 2 \\ 2 & 3 & 4 & 5 & 4 & 3 \end{bmatrix}$$

*Each row corresponds to a variable

1. Mean vector for input data

$$E(\vec{x}) = \frac{1}{6}\begin{bmatrix} 2+4+5+5+3+2 \\ 2+3+4+5+4+3 \end{bmatrix} = \begin{bmatrix} 3.5 \\ 3.5 \end{bmatrix}$$

2. Covariance matrix

$$\mathbf{R}_{\mathbf{XX}} = \frac{1}{5}\sum_{i=0}^{5}(\vec{x}_i - \begin{bmatrix} 3.5 \\ 3.5 \end{bmatrix})(\vec{x}_i - \begin{bmatrix} 3.5 \\ 3.5 \end{bmatrix})^T$$

$$= \frac{1}{5}\left\{\begin{bmatrix} -1.5 \\ -1.5 \end{bmatrix}\begin{bmatrix} -1.5 & -1.5 \end{bmatrix} + \ldots + \begin{bmatrix} -1.5 \\ -.5 \end{bmatrix}\begin{bmatrix} -1.5 & -.5 \end{bmatrix}\right\}$$

$$= \frac{1}{5}\left\{\begin{bmatrix} 2.25 & 2.25 \\ 2.25 & 2.25 \end{bmatrix} + \ldots + \begin{bmatrix} 2.25 & 0.75 \\ 0.75 & 0.25 \end{bmatrix}\right\} = \begin{bmatrix} 1.9 & 1.1 \\ 1.1 & 1.1 \end{bmatrix}$$

KLT - Example

$$\mathbf{X} = \begin{bmatrix} 2 & 4 & 5 & 5 & 3 & 2 \\ 2 & 3 & 4 & 5 & 4 & 3 \end{bmatrix}$$

*Each row corresponds to a variable

3. Find eigenvalues

$$0 = \left|\mathbf{R}_{\mathrm{XX}} - \lambda \mathbf{I}\right| = \begin{vmatrix} 1.9-\lambda & 1.1 \\ 1.1 & 1.1-\lambda \end{vmatrix}$$

$$\lambda^2 - 3\lambda + 0.88 = 0$$

$$\lambda_1 = 2.67, \lambda_2 = 0.33$$

4. Find eigenvectors

$$\begin{bmatrix} -0.77 & 1.1 \\ 1.1 & -1.57 \end{bmatrix} \begin{bmatrix} \varphi_{11} \\ \varphi_{21} \end{bmatrix} = 0 \qquad \begin{bmatrix} 1.57 & 1.1 \\ 1.1 & 0.77 \end{bmatrix} \begin{bmatrix} \varphi_{12} \\ \varphi_{22} \end{bmatrix} = 0$$

$$\Rightarrow \varphi_{11} = 1.43\varphi_{21} \qquad \Rightarrow \varphi_{12} = -0.7\varphi_{22}$$

KLT - Example

$$\mathbf{X} = \begin{bmatrix} 2 & 4 & 5 & 5 & 3 & 2 \\ 2 & 3 & 4 & 5 & 4 & 3 \end{bmatrix}$$

*Each row corresponds to a variable

5. Normalise and solve

(From previous step: $\varphi_{11} = 1.43\varphi_{21},\ \varphi_{12} = -0.7\varphi_{22}$)

$$\langle \vec{\varphi}_1, \vec{\varphi}_1 \rangle = 1 \rightarrow {\varphi_{11}}^2 + {\varphi_{21}}^2 = 1$$
$$\langle \vec{\varphi}_2, \vec{\varphi}_2 \rangle = 1 \rightarrow {\varphi_{12}}^2 + {\varphi_{22}}^2 = 1$$

$$\Rightarrow \varphi = \begin{bmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{bmatrix} = \begin{bmatrix} 0.82 & -0.57 \\ 0.57 & 0.82 \end{bmatrix}$$

6. Transform the input

$$\mathbf{Y} = \varphi^T \mathbf{X} = \begin{bmatrix} 0.82 & 0.57 \\ -0.57 & 0.82 \end{bmatrix} \begin{bmatrix} 2 & 4 & 5 & 5 & 3 & 2 \\ 2 & 3 & 4 & 5 & 4 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 2.78 & 4.99 & 6.38 & 6.95 & 4.74 & 3.35 \\ 0.5 & 0.18 & 0.43 & 1.25 & 1.57 & 1.32 \end{bmatrix}$$

KLT - Example

$$\mathbf{X} = \begin{bmatrix} 2 & 4 & 5 & 5 & 3 & 2 \\ 2 & 3 & 4 & 5 & 4 & 3 \end{bmatrix}$$

Original data

$$\mathbf{Y} = \begin{bmatrix} 2.78 & 4.99 & 6.38 & 6.95 & 4.74 & 3.35 \\ 0.5 & 0.18 & 0.43 & 1.25 & 1.57 & 1.32 \end{bmatrix}$$

Transformed data (projection onto orthogonal space)

Check: Covariance of Transformed Data

$$\mathbf{R}_{\mathbf{YY}} = \frac{1}{N-1}\sum_{i=0}^{N-1}(\vec{y}_i - E[\vec{y}])(\vec{y}_i - E[\vec{y}])^T = \begin{bmatrix} 2.67 & 0 \\ 0 & 0.33 \end{bmatrix}$$

Total "energy" is proportional to 2.67+0.33 = 3.00

- 89% (2.67/3 *100%) of energy along the first axis, 11% on the second.
- Covariance between two new variables is zero, i.e., uncorrelated.

KLT - Example

$$\mathbf{X} = \begin{bmatrix} 2 & 4 & 5 & 5 & 3 & 2 \\ 2 & 3 & 4 & 5 & 4 & 3 \end{bmatrix}$$

Original data

$$\mathbf{Y} = \begin{bmatrix} 2.78 & 4.99 & 6.38 & 6.95 & 4.74 & 3.35 \\ 0.5 & 0.18 & 0.43 & 1.25 & 1.57 & 1.32 \end{bmatrix}$$

Transformed data
(projection onto orthogonal space)

Now suppose we do dimensionality reduction and remove second coordinate.

$$\mathbf{Y'} = \begin{bmatrix} 2.78 & 4.99 & 6.38 & 6.95 & 4.74 & 3.35 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

KLT - Example

$$\mathbf{X} = \begin{bmatrix} 2 & 4 & 5 & 5 & 3 & 2 \\ 2 & 3 & 4 & 5 & 4 & 3 \end{bmatrix}$$

Original data

- Inverse transform of reduced data

$$\mathbf{X}' = \varphi \mathbf{Y}' = \begin{bmatrix} 0.82 & -0.57 \\ 0.57 & 0.82 \end{bmatrix} \begin{bmatrix} 2.78 & 4.99 & 6.38 & 6.95 & 4.74 & 3.35 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2.28 & 4.1 & 5.23 & 5.7 & 3.89 & 2.75 \\ 1.58 & 2.84 & 3.64 & 3.96 & 2.7 & 1.91 \end{bmatrix}$$

- Error

$$J = E(|\mathbf{X} - \mathbf{X}'|^2)$$

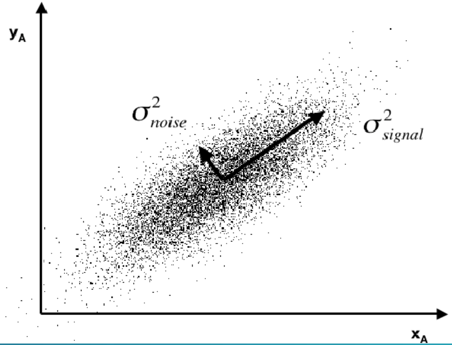
This MSE could be calculated, but without knowing the context of the application of the initial data would be meaningless.

Queen Mary University of London

Some further points on KLT

- KLT is not suitable for low-dimensional datasets that contains a small number of variables.
  - In image processing, each pixel is counted as a variable. A typical image contains hundreds/thousands of variable.

- KLT is not suitable for non-linear datasets because the nonlinearity cannot be explained by the orthogonal principal components. Hence, all principal components will carry large weights. Compression will be very lossy.



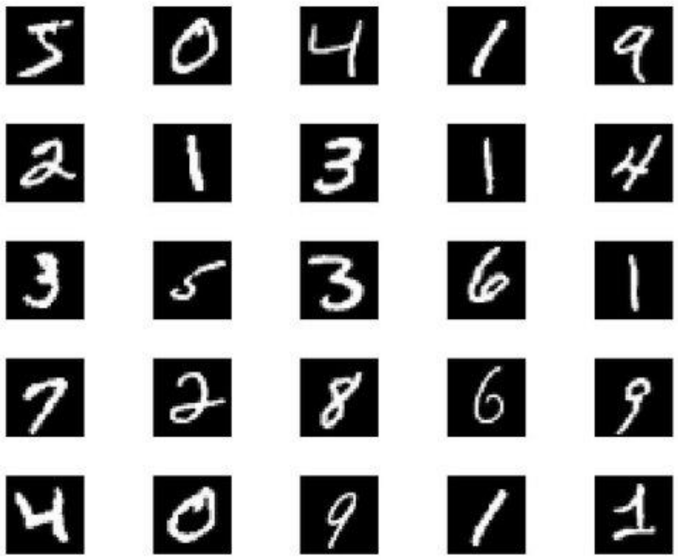
- KLT can be designed for image compression, facial recognition, financial analysis, denoising, etc…

Practical Applications of KLT

- Image Compression
- Facial Recognition

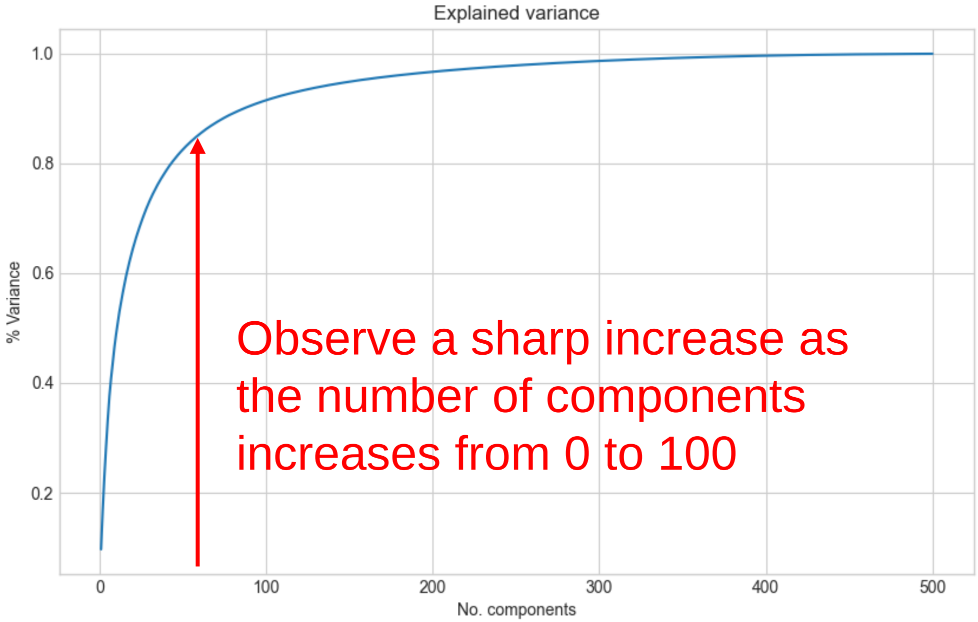
KLT (PCA) for image compression

- The well-known MINST dataset
- Each image contains 28x28 = 784 pixels
- KLT (PCA) will result in 784 principal components
- To compress the image, we eliminate the less important principal components



KLT (PCA) for image compression

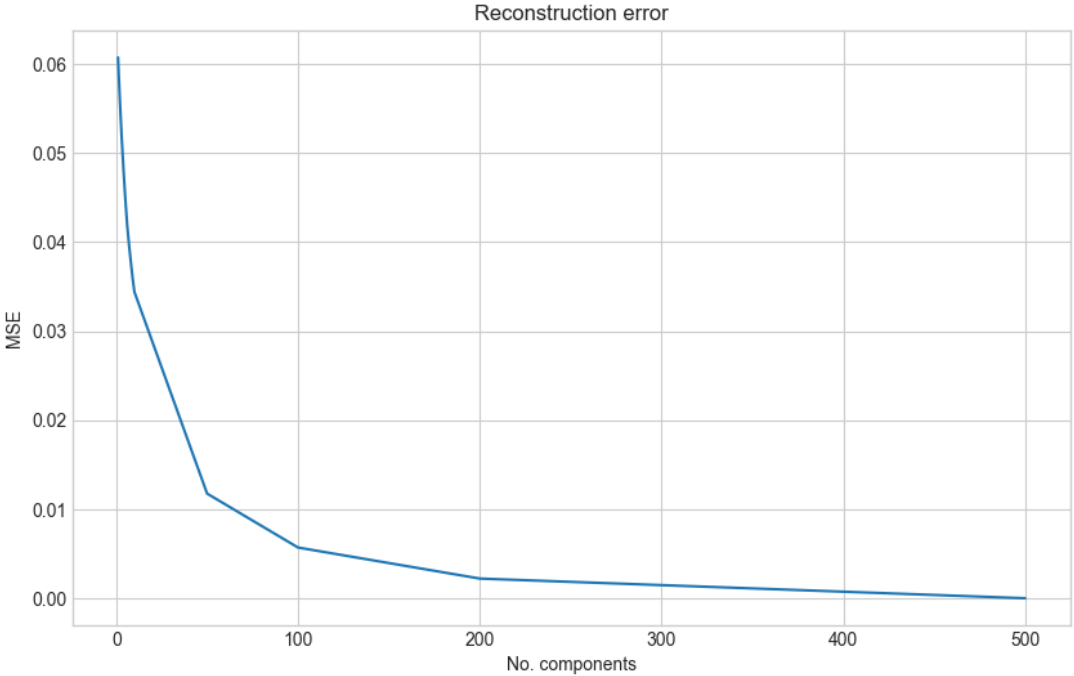
- Plot of % variance vs. number of principal components:



- % variance is associated with the eigenvalues of each eigenvector (principal component)
- Larger eigenvalues = more variance explained
- Principal components are ordered and added in descending order of their eigenvalues

KLT (PCA) for image compression

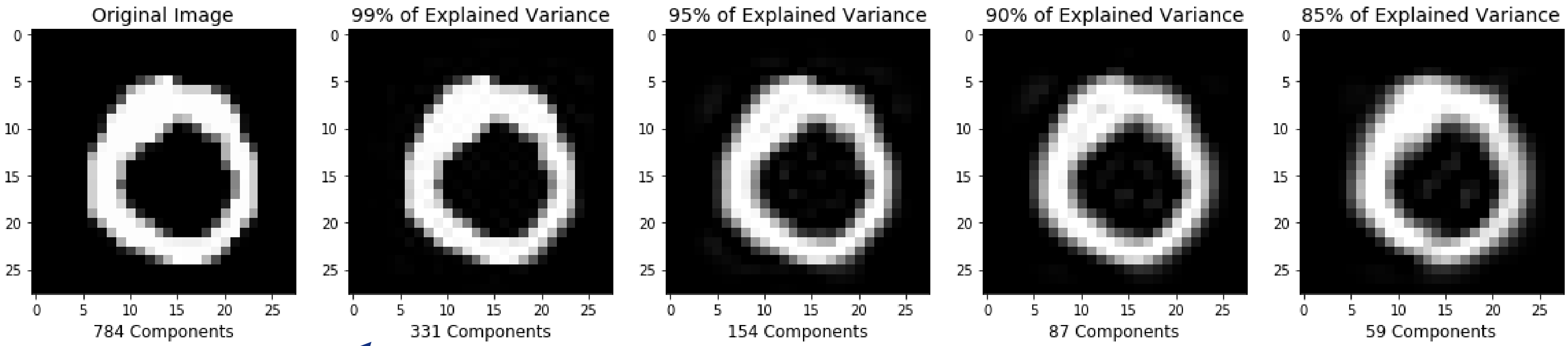
- Plot of mean square error (MSE) vs. number of principal components:



There is a trade-off between complexity and quality

KLT (PCA) for image compression

- If we limit the proportion of variance contained in the compressed image:



453 (57.7%) principal components are removed if we only keep 99% of the variance

≈ Transmission cost is reduced by 57.7% at the cost of 1% information loss

KLT (PCA) for image compression

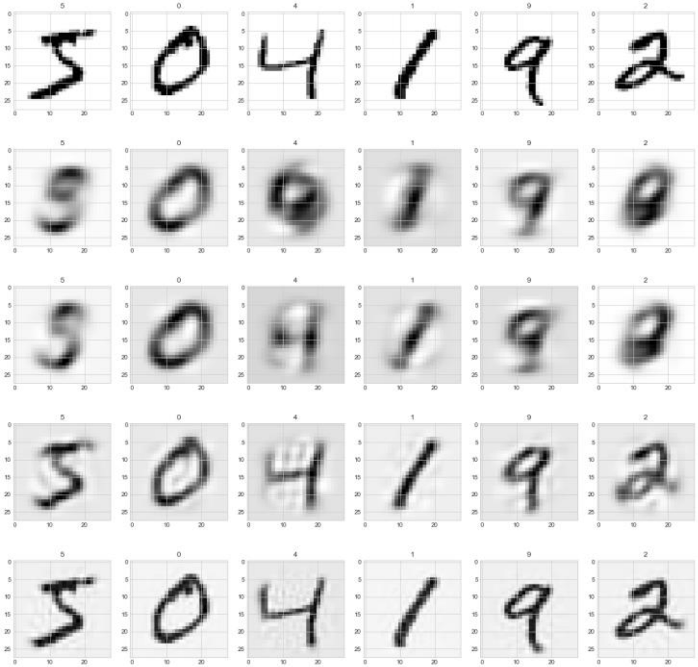
Original (784 components):

Reconstructed with 5 components:

Reconstructed with 10 components:

Reconstructed with 50 components:

Reconstructed with 200 components:



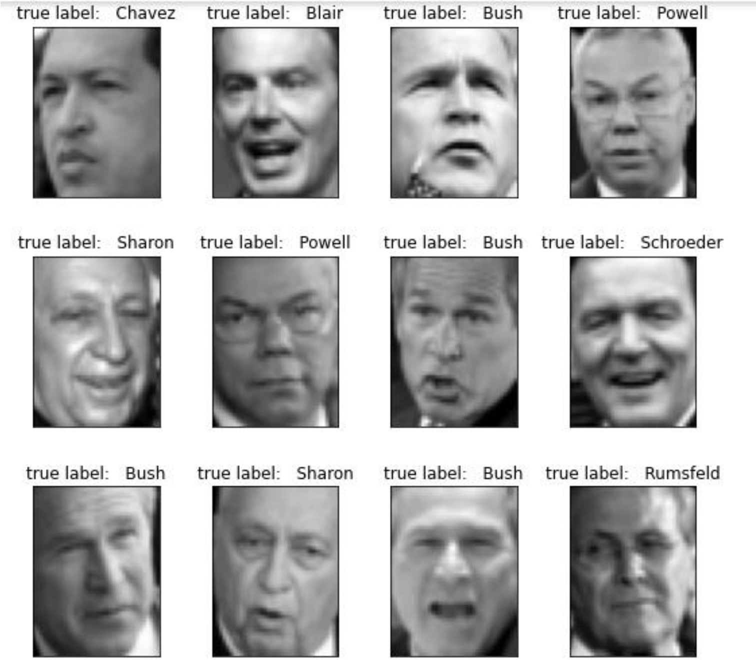
[2] Image compression - part 1. - PCA | Richard Stanton (richard-stanton.com)

Practical Applications of KLT

- Image Compression
- Facial Recognition

KLT (PCA) for Facial Recognition

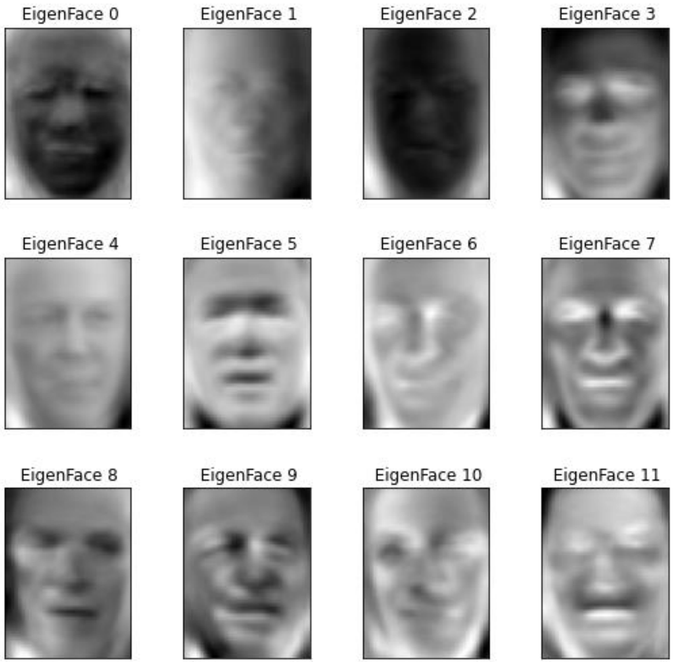
- The Labelled Faces in the Wild (LFW) dataset
- Each image contains 125x94 = 11,750 pixels
- For facial recognition, we use top 150 eigenvectors, known as eigenfaces
- The intuition is to calculate the weight of each eigenface based on the image. Then, perform classification algorithm based on the weights.



[2] 'ML | Face Recognition Using PCA Implementation', GeeksforGeeks, https://www.geeksforgeeks.org/ml-face-recognition-using-pca-implementation/

KLT (PCA) for Facial Recognition - Eigenfaces

- Eigenface = Eigenvector
- Eigenfaces has the same length as the image
- If you reshape the eigenfaces to the same dimension as the images, you will see an abstract face as shown in the right
- Each eigenface describes a certain characteristic of the human face
- Different combination of eigenfaces adds up to a different human face



[2] 'ML | Face Recognition Using PCA Implementation', GeeksforGeeks, https://www.geeksforgeeks.org/ml-face-recognition-using-pca-implementation/

KLT (PCA) for Facial Recognition

- Use the set of eigenvalues as image identifier
- During training, obtain the unique combination of eigenvalues for each person
- During testing, calculate the set of eigenvalues of the given image. Then, find the person with the most similar set of eigenvalues
- Given an image Γ_j, the average face Φ, and an eigenface u_i, the eigenvalue w_i is computed by

$$w_i = u_i(\Gamma_i - \Phi)$$

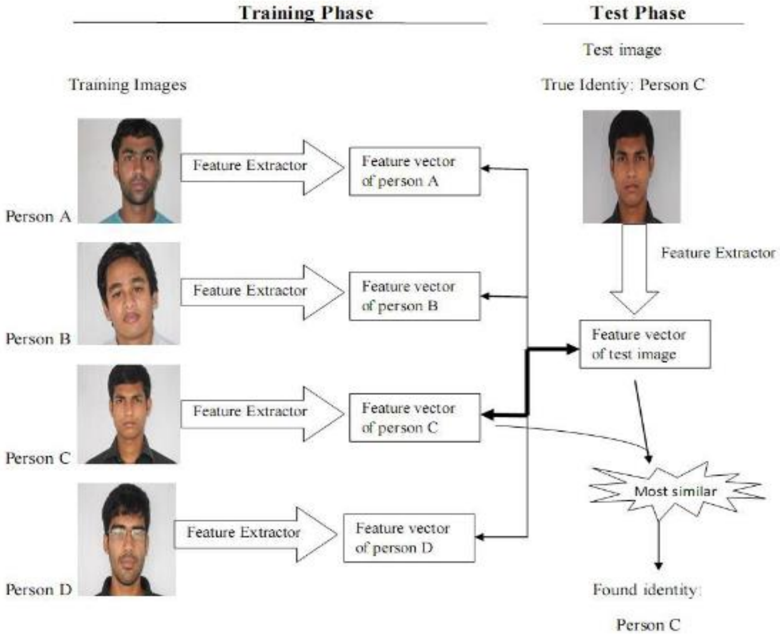


Fig. 1: An example procedure of using PCA for facial recognition

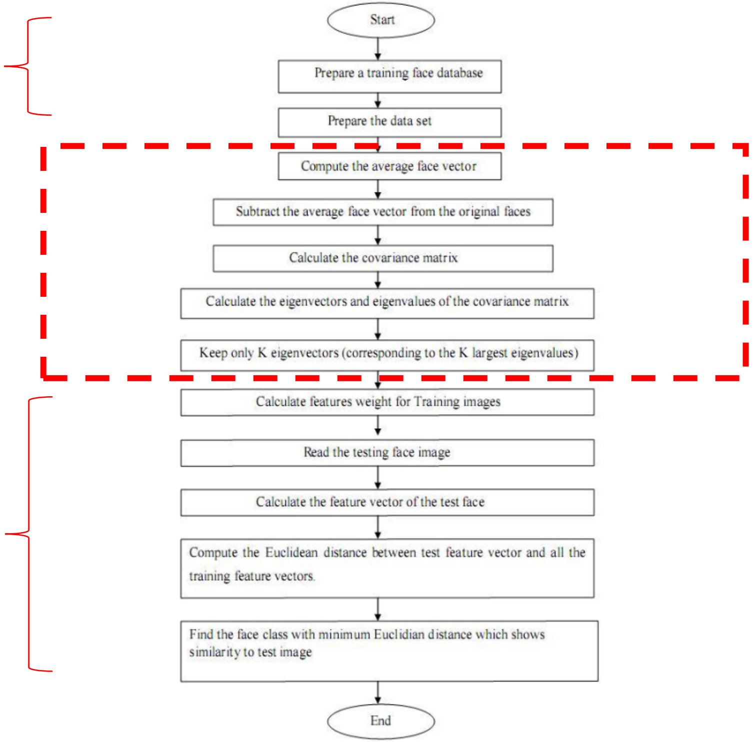
[3] Paul, Liton & Suman, Abdulla. (2012). Face recognition using principal component analysis method. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). 1. 135-139.

KLT (PCA) for Facial Recognition

Data preparation:

Exact procedures of KLT:
(to obtain the eigenfaces)

Compare the set of eigenvalues of the dataset and the test image:



[3] Paul, Liton & Suman, Abdulla. (2012). Face recognition using principal component analysis method. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). 1. 135-139.

Summary

- Practical applications of KLT (non-assessed)
  1. Image compression
     - By keeping the top N eigenvectors that explain the majority of variance in the image
     - In the MNIST data, the number of eigenvectors can be reduced from 784 to 331 at the cost of 1% reduction in explained variance
  2. Facial recognition
     - Eigenvectors can be visualized as eigenfaces
     - Each person is associated to a unique set of eigenvalues
     - Facial recognition is achieved by comparing the set of eigenvalues in the image to the dataset

Queen Mary
University of London