# Principal Component Analysis (PCA)

CE-717: Machine Learning
Sharif University of Technology
Spring 2016

Soleymani

# Dimensionality Reduction: Feature Selection vs. Feature Extraction

- Feature **selection**
  - Select a subset of a given feature set
- Feature **extraction**
  - A linear or non-linear transform on the original feature space

$$\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \rightarrow \begin{bmatrix} x_{i_1} \\ \vdots \\ x_{i_{d'}} \end{bmatrix}$$

Feature Selection ($d' < d$)

$$\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ \vdots \\ y_{d'} \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}\right)$$

Feature Extraction

# Feature Extraction

- Mapping of the original data to another space
  - Criterion for feature extraction can be different based on problem settings
    - Unsupervised task: minimize the information loss (reconstruction error)
    - Supervised task: maximize the class discrimination on the projected space

- Feature extraction algorithms
  - Linear Methods
    - Unsupervised: e.g., Principal Component Analysis (PCA)
    - Supervised: e.g., Linear Discriminant Analysis (LDA)
      - Also known as Fisher's Discriminant Analysis (FDA)

# Feature Extraction

- Unsupervised feature extraction:

$$\boldsymbol{X} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix} \Rightarrow \boxed{\text{Feature Extraction}} \Rightarrow$$

A mapping $f: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$
Or
only the transformed data

$$\boldsymbol{X}' = \begin{bmatrix} x'^{(1)}_1 & \cdots & x'^{(1)}_{d'} \\ \vdots & \ddots & \vdots \\ x'^{(N)}_1 & \cdots & x'^{(N)}_{d'} \end{bmatrix}$$

- Supervised feature extraction:

$$\boldsymbol{X} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix} \Rightarrow \boxed{\text{Feature Extraction}} \Rightarrow$$

$$Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$

A mapping $f: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$
Or
only the transformed data

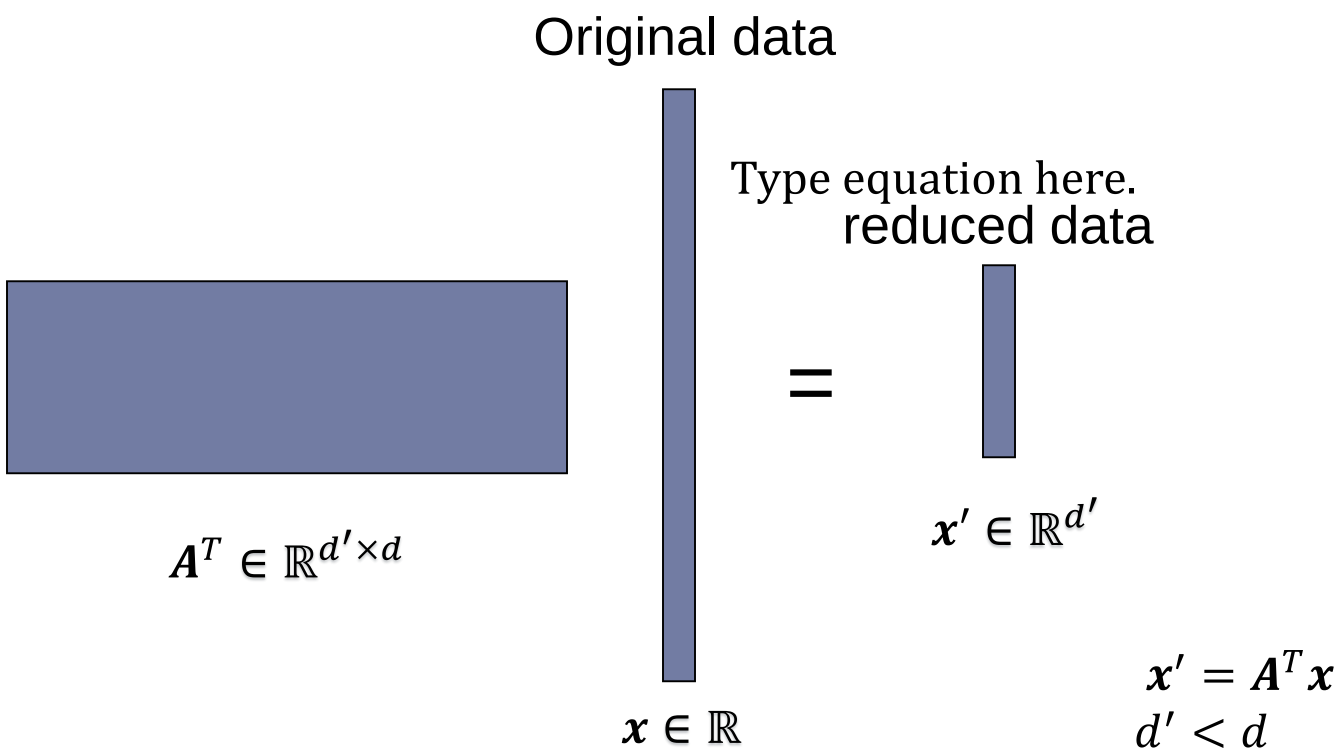
$$\boldsymbol{X}' = \begin{bmatrix} x'^{(1)}_1 & \cdots & x'^{(1)}_{d'} \\ \vdots & \ddots & \vdots \\ x'^{(N)}_1 & \cdots & x'^{(N)}_{d'} \end{bmatrix}$$

# Unsupervised Feature Reduction

- Visualization: projection of high-dimensional data onto 2D or 3D.
- Data compression: efficient storage, communication, or and retrieval.
- Pre-process: to improve accuracy by reducing features
  - As a preprocessing step to reduce dimensions for supervised learning tasks
  - Helps avoiding overfitting
- Noise removal
  - E.g, "noise" in the images introduced by minor lighting variations, slightly different imaging conditions, etc.

# Linear Transformation

- For linear transformation, we find an explicit mapping $f(\boldsymbol{x}) = \boldsymbol{A}^T \boldsymbol{x}$ that can transform also new data vectors.



# Linear Transformation

- Linear transformation are simple mappings

$$\boldsymbol{x}' = \boldsymbol{A}^T \boldsymbol{x} \qquad \boldsymbol{A} = \begin{bmatrix} a_{11} & \cdots & a_{1d'} \\ \vdots & \ddots & \vdots \\ a_{d1} & \cdots & a_{dd'} \end{bmatrix}$$

$\mathbf{a}_1$ $\qquad$ $\mathbf{a}_{d'}$

$\boldsymbol{a}_1^T$

$$\begin{bmatrix} x'_1 \\ \vdots \\ x'_{d'} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{d1} \\ \vdots & \ddots & \vdots \\ a_{1d'} & \cdots & a_{d'd} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$$

$\boldsymbol{a}_{d'}^T$

$$x'_j = \boldsymbol{a}_j^T \boldsymbol{x} \qquad j = 1, \dots, d'$$

# Linear Dimensionality Reduction

- Unsupervised
  - Principal Component Analysis (PCA) [we will discuss]
  - Independent Component Analysis (ICA) [we will discuss]
  - Singular Value Decomposition (SVD)
  - Multi Dimensional Scaling (MDS)
  - Canonical Correlation Analysis (CCA)

# Principal Component Analysis (PCA)

- Also known as Karhonen-Loeve (KL) transform

- Principal Components (PCs): orthogonal vectors that are ordered by the fraction of the total information (variation) in the corresponding directions
  - Find the directions at which data approximately lie
    - When the data is projected onto first PC, the variance of the projected data is maximized

- PCA is an orthogonal projection of the data into a subspace so that the variance of the projected data is maximized.

# Principal Component Analysis (PCA)

- The "best" linear subspace (i.e. providing least reconstruction error of data):
  - Find mean reduced data
  - The axes have been rotated to new (principal) axes such that:
    - Principal axis 1 has the highest variance

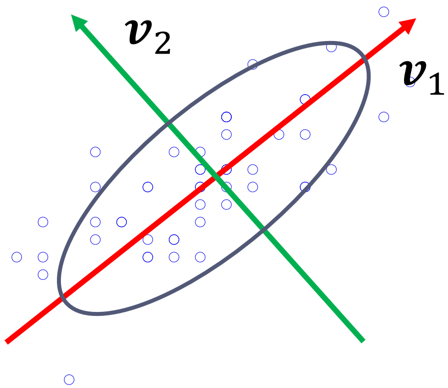
      ....
    - Principal axis i has the i-th highest variance.
  - The principal axes are uncorrelated
    - Covariance among each pair of the principal axes is zero.

- Goal: reducing the dimensionality of the data while preserving the variation present in the dataset as much as possible.

- PCs can be found as the "best" eigenvectors of the covariance matrix of the data points.

# Principal components

- If data has a Gaussian distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the direction of the largest variance can be found by the eigenvector of $\boldsymbol{\Sigma}$ that corresponds to the largest eigenvalue of $\boldsymbol{\Sigma}$



# PCA: Steps

- Input: $N \times d$ data matrix $\boldsymbol{X}$ (each row contain a $d$ dimensional data point)
  - $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}^{(i)}$
  - $\widetilde{\boldsymbol{X}} \leftarrow$ Mean value of data points is subtracted from rows of $\boldsymbol{X}$
  - $\boldsymbol{C} = \frac{1}{N}\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}}$ (Covariance matrix)
  - Calculate eigenvalue and eigenvectors of $\boldsymbol{C}$
  - Pick $d'$ eigenvectors corresponding to the largest eigenvalues and put them in the columns of $\boldsymbol{A} = [\underbrace{\boldsymbol{v}_1}_{\text{First PC}}, \dots, \underbrace{\boldsymbol{v}_{d'}}_{\text{d'-th PC}}]$
  - $\boldsymbol{X}' = \widetilde{\boldsymbol{X}}\boldsymbol{A}$

# Covariance Matrix

$$\boldsymbol{\mu}_x = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_d \end{bmatrix} = \begin{bmatrix} E(x_1) \\ \vdots \\ E(x_d) \end{bmatrix}$$

$$\boldsymbol{\Sigma} = E\left[(\boldsymbol{x} - \boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x)^T\right]$$

- ML estimate of covariance matrix from data points $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$:

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{x}^{(i)} - \widehat{\boldsymbol{\mu}}\right)\left(\boldsymbol{x}^{(i)} - \widehat{\boldsymbol{\mu}}\right)^T = \frac{1}{N}\left(\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}}\right)$$

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} \widetilde{\boldsymbol{x}}^{(1)} \\ \vdots \\ \widetilde{\boldsymbol{x}}^{(N)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^{(1)} - \widehat{\boldsymbol{\mu}} \\ \vdots \\ \boldsymbol{x}^{(N)} - \widehat{\boldsymbol{\mu}} \end{bmatrix} \qquad \widehat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}^{(i)}$$

Mean-centered data

We now assume that data are mean removed and $\boldsymbol{x}$ in the later slides is indeed $\widetilde{\boldsymbol{x}}$

# Correlation matrix

$$\boldsymbol{X} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix}$$

$$\frac{1}{N}\boldsymbol{X}^T\boldsymbol{X} = \frac{1}{N}\begin{bmatrix} x_1^{(1)} & \cdots & x_1^{(N)} \\ \vdots & \ddots & \vdots \\ x_d^{(1)} & \cdots & x_d^{(N)} \end{bmatrix}\begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix}$$

$$= \frac{1}{N}\begin{bmatrix} \sum_{n=1}^{N} x_1^{(n)} x_1^{(n)} & \cdots & \sum_{n=1}^{N} x_1^{(n)} x_d^{(n)} \\ \vdots & \ddots & \vdots \\ \sum_{n=1}^{N} x_d^{(n)} x_1^{(n)} & \cdots & \sum_{n=1}^{N} x_d^{(n)} x_d^{(n)} \end{bmatrix}$$

# Two Interpretations

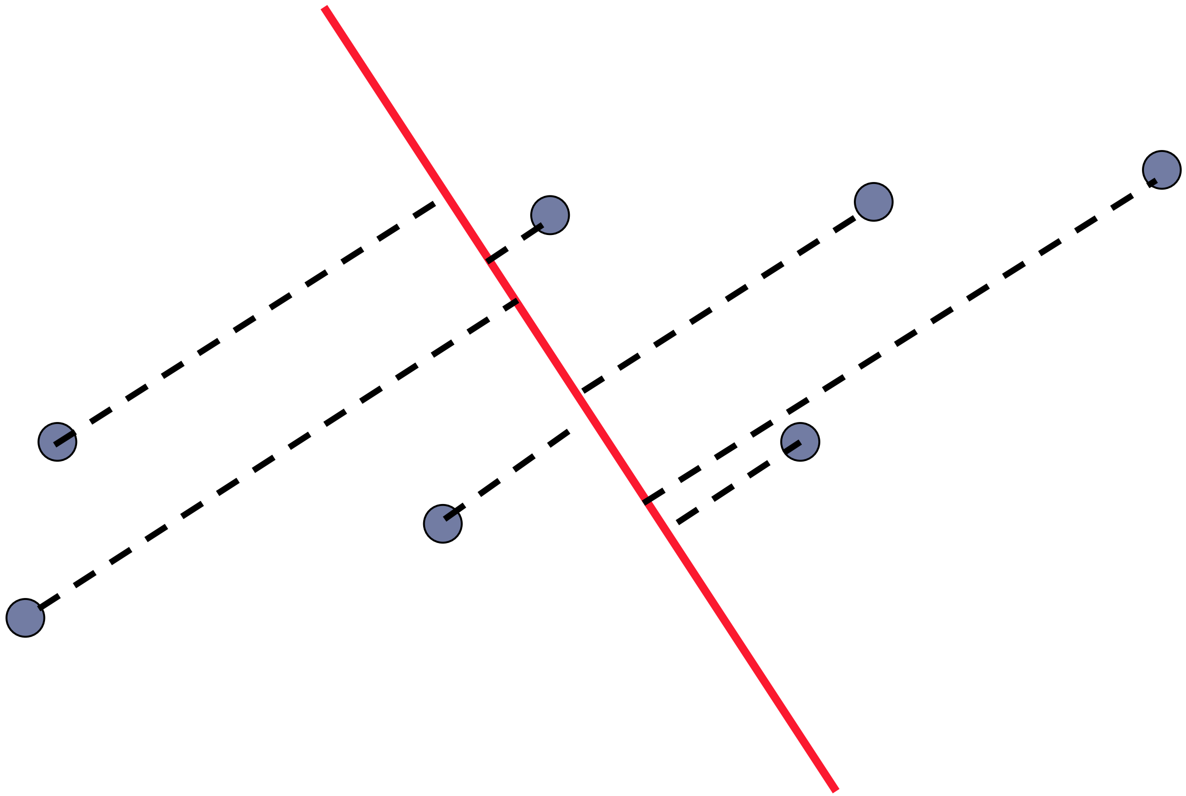
- Maximum Variance Subspace
  - PCA finds vectors v such that projections on to the vectors capture maximum variance in the data
  - $\frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{a}^{T}\boldsymbol{x}^{(n)}\right)^{2} = \frac{1}{N}\boldsymbol{a}^{T}\boldsymbol{X}^{T}\boldsymbol{X}\boldsymbol{a}$

- Minimum Reconstruction Error
  - PCA finds vectors v such that projection on to the vectors yields minimum MSE reconstruction
  - $\frac{1}{N}\sum_{n=1}^{N}\left\|\boldsymbol{x}^{(n)} - \left(\boldsymbol{a}^{T}\boldsymbol{x}^{(n)}\right)\boldsymbol{a}\right\|^{2}$

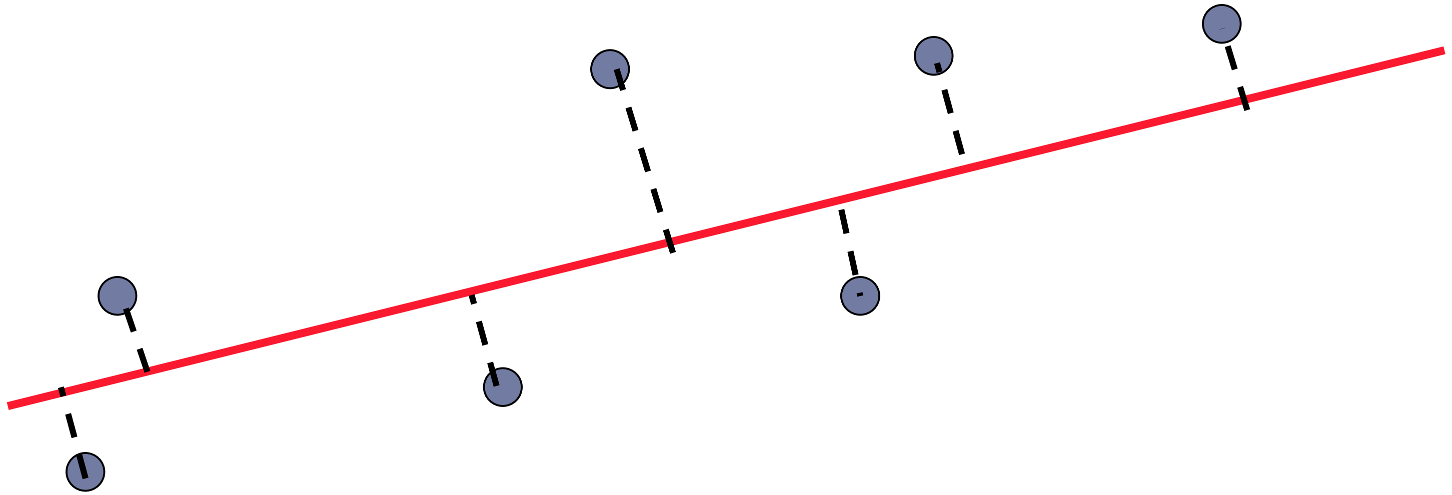
# Least Squares Error Interpretation

- PCs are linear least squares fits to samples, each orthogonal to the previous PCs:
  - First PC is a minimum distance fit to a vector in the original feature space
  - Second PC is a minimum distance fit to a vector in the plane perpendicular to the first PC
  - And so on

# Example

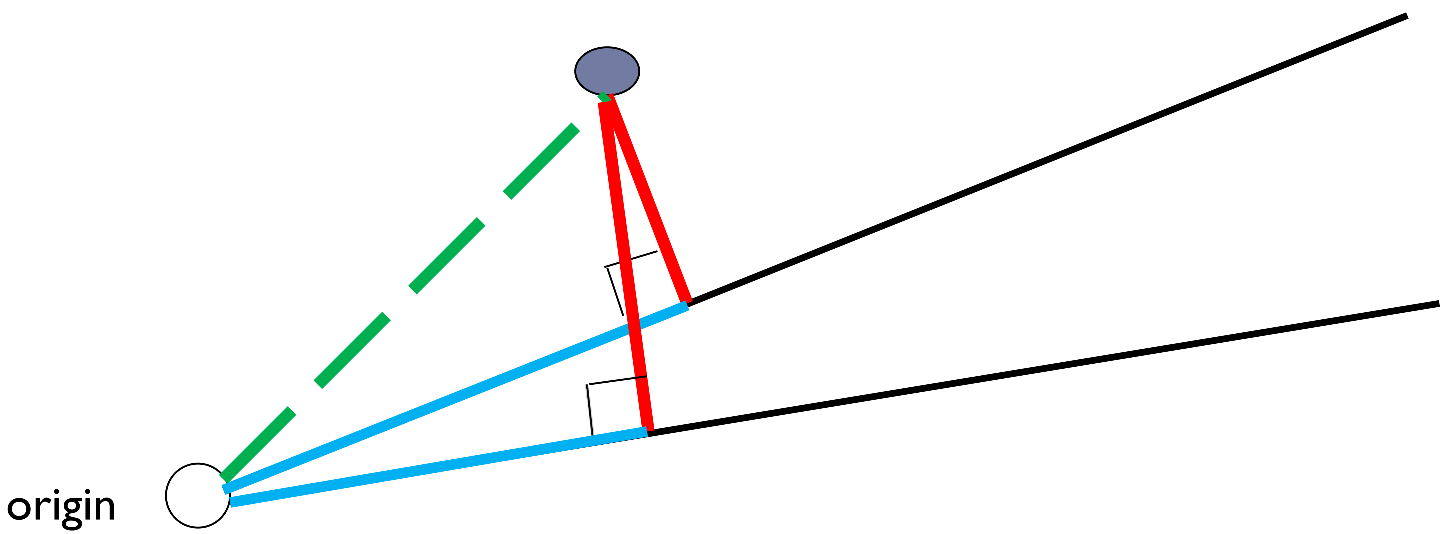


# Example



# Least Squares Error and Maximum Variance Views Are Equivalent (1-dim Interpretation)

- <u>Minimizing sum of square distances to the line</u> is equivalent to <u>maximizing the sum of squares of the projections on that line</u> (Pythagoras).



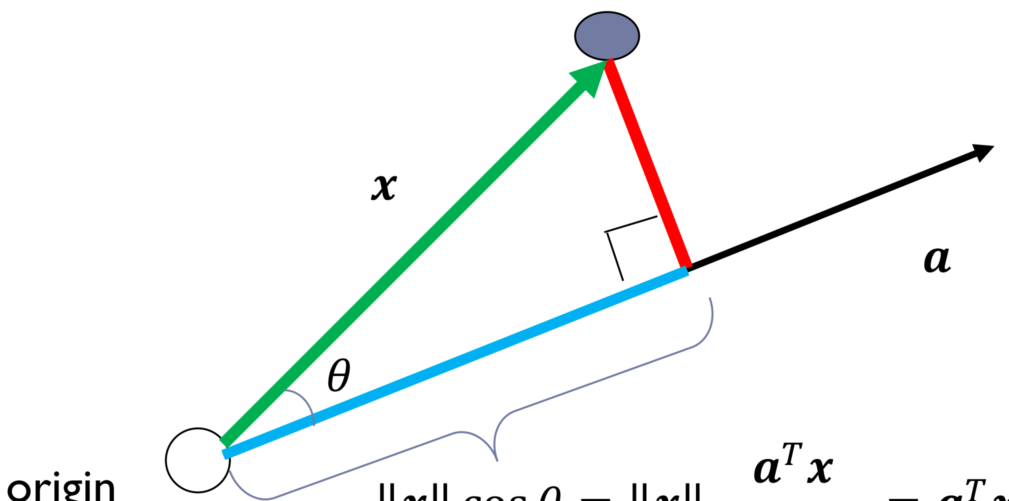

red$^2$+blue$^2$=green$^2$

green$^2$ is fixed (shows the data vector after mean removing)

$\Rightarrow$ maximizing blue$^2$ is equivalent to minimizing red$^2$

# First PC

- The first PC is direction of greatest variability in data
- We will show that the first PC is the eigenvector of the covariance matrix corresponding the maximum eigen value of this matrix.
- If $||\boldsymbol{a}|| = 1$, the projection of a d-dimensional $\boldsymbol{x}$ on $\boldsymbol{a}$ is $\boldsymbol{a}^T\boldsymbol{x}$




$$\|\boldsymbol{x}\| \cos\theta = \|\boldsymbol{x}\| \frac{\boldsymbol{a}^T\boldsymbol{x}}{\|\boldsymbol{x}\|\|\boldsymbol{a}\|} = \boldsymbol{a}^T\boldsymbol{x}$$

# First PC

$$\underset{\boldsymbol{a}}{\text{argmax}} \frac{1}{N} \sum_{n=1}^{N} \left(\boldsymbol{a}^T \boldsymbol{x}^{(n)}\right)^2 = \frac{1}{N} \boldsymbol{a}^T \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{a}$$

$$\text{s.t. } \boldsymbol{a}^T \boldsymbol{a} = 1$$

$$\frac{\partial}{\partial \boldsymbol{a}} \left( \frac{1}{N} \boldsymbol{a}^T \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{a} + \lambda (1 - \boldsymbol{a}^T \boldsymbol{a}) \right) = 0 \Rightarrow \boxed{\frac{1}{N} \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{a} = \lambda \boldsymbol{a}}$$

- $\boldsymbol{a}$ is the eigenvector of sample covariance matrix $\frac{1}{N} \boldsymbol{X}^T \boldsymbol{X}$
- The eigenvalue $\lambda$ denotes the amount of variance along that dimension.
  - Variance= $\frac{1}{N} \boldsymbol{a}^T \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{a} = \boldsymbol{a}^T \left( \frac{1}{N} \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{a} \right) = \boldsymbol{a}^T \lambda \boldsymbol{a} = \lambda$
- So, if we seek the dimension with the largest variance, it will be the eigenvector corresponding to the largest eigenvalue of the sample covariance matrix

# PCA: Uncorrelated Features

$$\boldsymbol{x}' = \boldsymbol{A}^T\boldsymbol{x}$$

$$\boldsymbol{R}_{\boldsymbol{x}'} = E\left[\boldsymbol{x}'\boldsymbol{x}'^T\right] = E[\boldsymbol{A}^T\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{A}] = \boldsymbol{A}^T E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{A} = \boldsymbol{A}^T\boldsymbol{R}_{\boldsymbol{x}}\boldsymbol{A}$$

- If $\boldsymbol{A} = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_d]$ where $\boldsymbol{a}_1, \dots, \boldsymbol{a}_d$ are orthonormal eighenvectors of $\boldsymbol{R}_{\boldsymbol{x}}$:

$$\boldsymbol{R}_{\boldsymbol{x}'} = \boldsymbol{A}^T\boldsymbol{R}_{\boldsymbol{x}}\boldsymbol{A} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{\Lambda}\boldsymbol{A}^T)\boldsymbol{A} = \boldsymbol{\Lambda}$$

$$\Rightarrow \forall i \neq j \ (i, j = 1, \dots, d) \ E\left[\boldsymbol{x}'_i\boldsymbol{x}'_j\right] = 0$$

then mutually uncorrelated features are obtained

  - Completely uncorrelated features avoid information redundancies

# PCA Derivation: Mean Square Error Approximation

- Incorporating all eigenvectors in $\boldsymbol{A} = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_d]$:

$$\boldsymbol{x}' = \boldsymbol{A}^T\boldsymbol{x} \Rightarrow \boldsymbol{A}\boldsymbol{x}' = \boldsymbol{A}\boldsymbol{A}^T\boldsymbol{x} = \boldsymbol{x}$$
$$\Rightarrow \boldsymbol{x} = \boldsymbol{A}\boldsymbol{x}'$$

- $\Longrightarrow$ If $d' = d$ then $\boldsymbol{x}$ can be reconstructed exactly from $\boldsymbol{x}'$

# PCA Derivation: Relation between Eigenvalues and Variances

- The $j$-th largest eigenvalue of $\boldsymbol{R_x}$ is the variance on the $j$-th PC:

$$var(x_j') = \lambda_j$$

$$var(x_j') = E[x_j' x_j']$$
$$= E[\boldsymbol{a}_j^T \boldsymbol{x}\boldsymbol{x}^T \boldsymbol{a}_j] = \boldsymbol{a}_j^T E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{a}_j$$
$$= \boldsymbol{a}_j^T \boldsymbol{R_x} \boldsymbol{a}_j = \boldsymbol{a}_j^T \lambda_j \boldsymbol{a}_j = \lambda_j$$

Eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots$
•The 1st PC is the the eigenvector of the sample covariance matrix associated with the largest eigenvalue
•The 2nd PC $v2$ is the the eigenvector of the sample covariance matrix associated with the second largest eigenvalue
•And so on …

# PCA Derivation: Mean Square Error Approximation

- Incorporating only $d'$ eigenvectors corresponding to the largest eigenvalues $\boldsymbol{A} = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_{d'}]$ $(d' < d)$

- It minimizes MSE between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}'$:

$$J(\boldsymbol{A}) = E[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2] = E[\|\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}'\|^2]$$

$$= E\left[\left\|\sum_{j=d'+1}^{d} x_j' \boldsymbol{a}_j\right\|^2\right]$$

$$= E\left[\sum_{j=d'+1}^{d} \sum_{k=d'+1}^{d} x_j' \boldsymbol{a}_j^T \boldsymbol{a}_k \; x_k'\right] = E\left[\sum_{j=d'+1}^{d} {x_j'}^2\right]$$

$$= \sum_{j=d'+1}^{d} E\left[{x_j'}^2\right] = \sum_{j=d'+1}^{d} \lambda_j$$

Sum of the $d - d'$ smallest eigenvalues

# PCA Derivation: Mean Square Error Approximation

- In general, it can also be shown MSE is minimized compared to any other approximation of $\boldsymbol{x}$ by any $d'$-dimensional orthonormal basis
  - without first assuming that the axes are eigenvectors of the correlation matrix, this result can also be obtained.

- If the data is mean-centered in advance, $\boldsymbol{R}_x$ and $\boldsymbol{C}_x$ (covariance matrix) will be the same.
  - However, in the correlation version when $\boldsymbol{C}_x \neq \boldsymbol{R}_x$ the approximation is not, in general, a good one (although it is a minimum MSE solution)

# PCA on Faces: "Eigenfaces"

- ORL Database

Some Images

# PCA on Faces: "Eigenfaces"

Average face

1st to 10th PCs

For eigen faces
"gray" = 0,
"white" > 0,
"black" < 0

# PCA on Faces:

$\boldsymbol{x}$ is a $112 \times 92 = 10304$ dimensional vector containing intensity of the pixels of this image

**Feature vector=**$[x'_1, x'_2, \ldots, x'_{d'}]$

$x'_i = PC_i^T \boldsymbol{x}$ $\longrightarrow$ The projection of $\boldsymbol{x}$ on the i-th PC

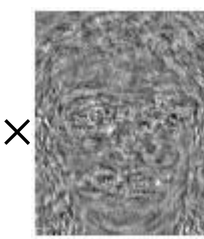
$=$

Average Face

$+x'_1 \times$

$+x'_2 \times$

$+\cdots$ $+x'_{256} \times$



# PCA on Faces: Reconstructed Face

**d'=1**

**d'=2**

**d'=4**

**d'=8**

**d'=16**

**d'=32**

**d'=64**

**d'=128**

**d'=256**

**Original Image**

# Dimensionality Reduction by PCA

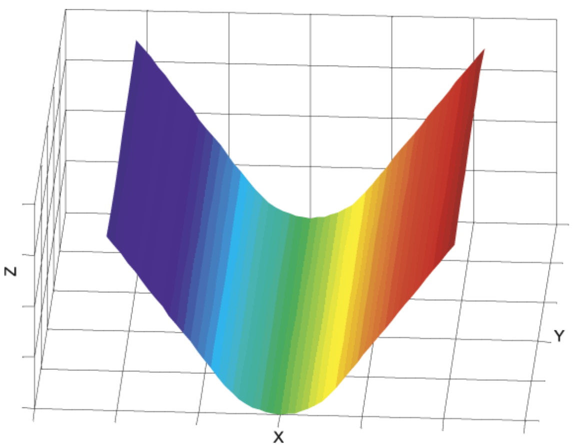
- In high-dimensional problems, data sometimes lies near a linear subspace (small variability around this subspace can be considered as noise)

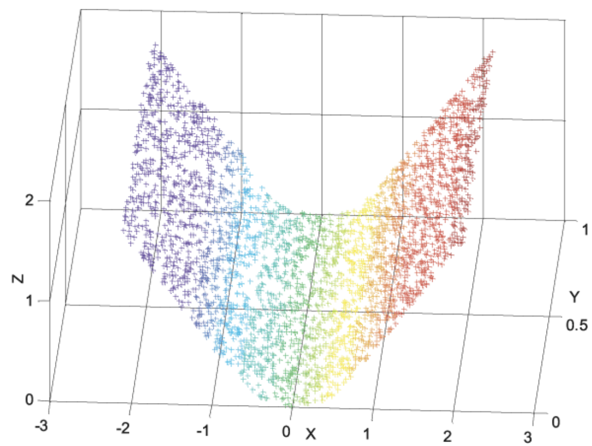
- Only keep data projections onto principal components with large eigenvalue

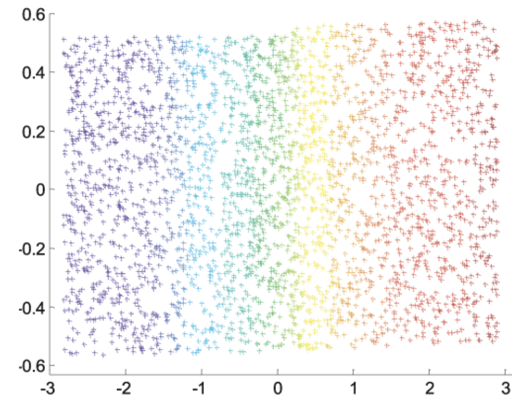
- Might lose some info, but if eigenvalues are small, do not lose much

# Kernel PCA

- Kernel extension of PCA



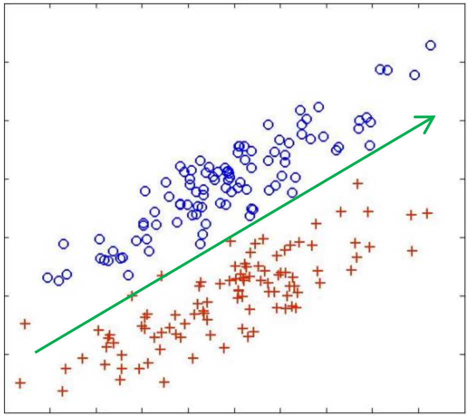



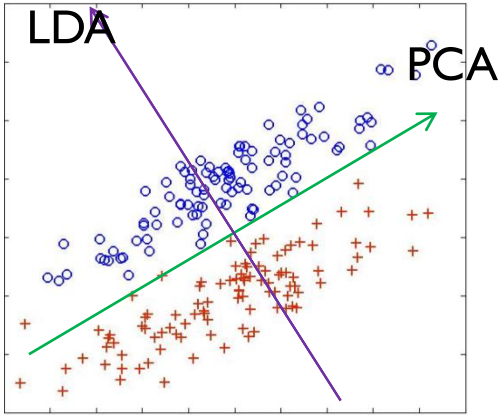




data (approximately) lies on
a lower dimensional non-linear space

# PCA and LDA: Drawbacks

- PCA drawback: An excellent information packing transform does not necessarily lead to a good class separability.
  - The directions of the maximum variance may be useless for classification purpose






- LDA drawback
  - Singularity or under-sampled problem (when $N < d$)
    - Example: gene expression data, images, text documents
  - Can reduces dimension only to $d' \leq C - 1$ (unlike PCA)

# PCA vs. LDA

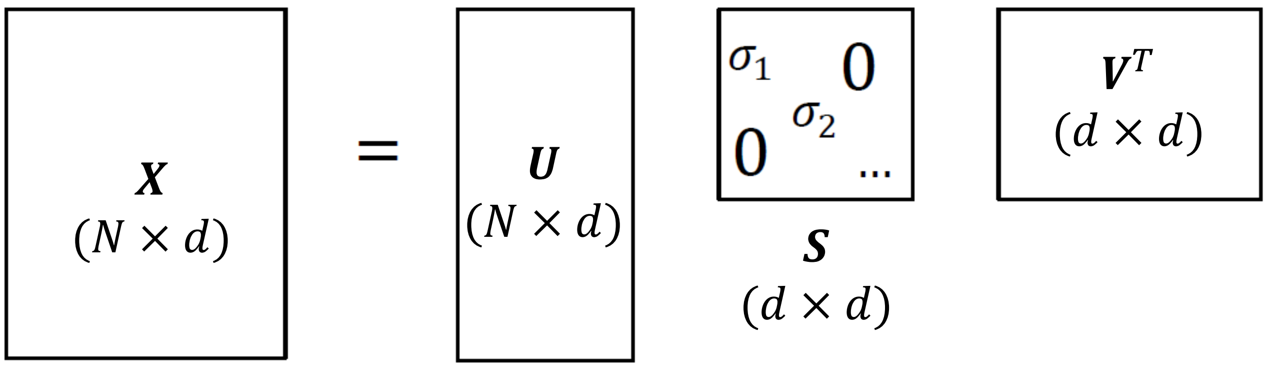
- Although LDA often provide more suitable features for classification tasks, PCA might outperform LDA in some situations:
  - when the number of samples per class is small (overfitting problem of LDA)
  - when the number of the desired features is more than $C - 1$

- Advances in the last decade:
  - Semi-supervised feature extraction
    - E.g., PCA+LDA, Regularized LDA, Locally FDA (LFDA)

# Singular Value Decomposition (SVD)

- Given a matrix $\boldsymbol{X} \in \mathbb{R}^{N \times d}$, the SVD is a decomposition:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$$




$$= \sum_i \sigma_i u_i v_i^T$$

- $\boldsymbol{S}$ is a diagonal matrix with the singular values $\sigma_1, \ldots, \sigma_d$ of $X$.
- Columns of $\boldsymbol{U}, \boldsymbol{V}$ are orthonormal matrices

# Singular Value Decomposition (SVD)

- Given a matrix $\boldsymbol{X} \in \mathbb{R}^{N\times d}$, the SVD is a decomposition:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$$

- SVD of $\boldsymbol{X}$ is related to eigen-decomposition of $\boldsymbol{X}^T\boldsymbol{X}$ and $\boldsymbol{X}\boldsymbol{X}^T$.
  - $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^T$
    - so $\boldsymbol{V}$ contains eigenvectors of $\boldsymbol{X}^T\boldsymbol{X}$ and $\boldsymbol{S}^2$ includes its eigenvalues ($\lambda_i = \sigma_i^2$)
  - $\boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T = \boldsymbol{U}\boldsymbol{S}^2\boldsymbol{U}^T$
    - so $\boldsymbol{U}$ contains eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$and $\boldsymbol{S}^2$ includes its eigenvalues ($\lambda_i = \sigma_i^2$)

- In fact, we can view each row of $\boldsymbol{U}\boldsymbol{S}$ as the coordinates of an example along the axes given by the eigenvectors.

# Independent Component Analysis (ICA)

- PCA:
  - The transformed dimensions will be uncorrelated from each other
  - Orthogonal linear transform
  - Only uses second order statistics (i.e., covariance matrix)

- ICA:
  - The transformed dimensions will be as independent as possible.
  - Non-orthogonal linear transform
  - High-order statistics can also used

# Uncorrelated and Independent

Uncorrelated: $cov(X_1,X_2) = 0$
Independent: $P(X_1,X_2) = P(X_1)P(X_2)$

- Gaussian
  - Independent $\Leftrightarrow$ Uncorrelated
- Non-Gaussian
  - Independent $\Rightarrow$ Uncorrelated
  - Uncorrelated $\nRightarrow$ Independent

# ICA: Cocktail party problem

- Cocktail party problem
  - $d$ speakers are speaking simultaneously and any microphone records only an overlapping combination of these voices.
    - Each microphone records a different combination of the speakers' voices.
  - Using these $d$ microphone recordings, can we separate out the original $d$ speakers' speech signals?

- Mixing matrix $\boldsymbol{A}$:

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$$

- Unmixing matrix $\boldsymbol{A}^{-1}$:

$$\boldsymbol{s} = \boldsymbol{A}^{-1}\boldsymbol{x}$$

$s_j^{(i)}$: sound that speaker $j$ was uttering at time $i$.

$x_j^{(i)}$: acoustic reading recorded by microphone $j$ at time $i$.

# ICA

- Find a linear transformation $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$
- for which dimensions of $\boldsymbol{s} = [s_1, s_2, \dots, s_d]^T$ are statistically independent

$$p(s_1, \dots, s_d) = p_1(s_1)p_2(s_2) \dots p_d(s_d)$$

- Algorithmically, we need to identify matrix $\boldsymbol{A}$ and sources $\boldsymbol{s}$ where $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$ such that the mutual information between $s_1, s_2, \dots, s_d$ is minimized:

$$I(s_1, s_2, \dots, s_d) = \sum_{i=1}^{d} H(s_i) - H(s_1, s_2, \dots, s_d)$$