Machine Learning & Pattern Recognition

Xin Xin(辛鑫)

xinxin@sdu.edu.cn

https://xinxin-me.github.io/

Unsupervised Feature Extraction

- **Principal Component Analysis (PCA)**

What is feature extraction?

- Feature extraction (dimensionality reduction/feature reduction) refers to the mapping of the original **high-dimensional** data into a **low-dimensional** space.
- Criterion for feature reduction can be different based on different problem setting.
  - ✓ Unsupervised setting: minimize the information loss
  - ✓ Supervised setting: maximize the class discrimination

Feature Extraction VS. Feature Selection

- Feature extraction
  - All original features are used.
  - Transformed features are linear combinations of the original features

- Feature selection
  - Only a subset of the original features are used.

Why Feature Extraction?

- **Visualization**: projection of high-dimensional data onto 2D or 3D
- **Data compression**: efficient storage and retrieval
- **Noise removal**: positive effect on accuracy

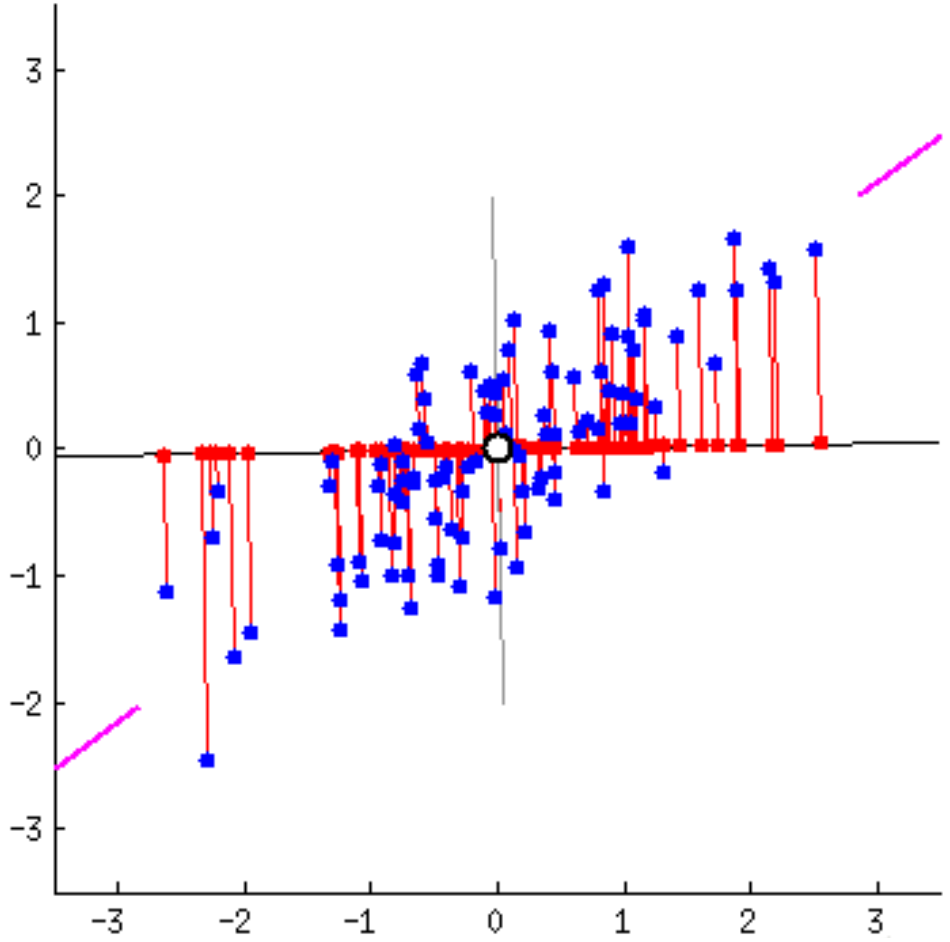
Feature Extraction Algorithms

- **Unsupervised**
  - Principal Component Analysis (PCA)
  - Nonnegative Matrix Factorization (NMF)
  - Independent Component Analysis (ICA) [Reading]
- **Supervised**
  - Linear Discriminant Analysis (LDA)
  - General Graph Embedding (GE) [Reading]
  - Canonical Correlation Analysis (CCA) [Reading, encouraged]
- **Semi-supervised**
  - Research topic [Further study, encouraged]

Principal Component Analysis (PCA)

PCA is a technique that is widely used for applications such as dimensionality reduction, lossy data compression, feature extraction, and data visualization (Jolliffe, 2002).

Map the original high-dimensional data into a low-dimensional space.



Principal Component Analysis (PCA)

Principal component analysis, or PCA, is a technique that is widely used for applications such as dimensionality reduction, lossy data compression, feature extraction, and data visualization (Jolliffe, 2002).

- Two commonly used definitions of PCA
  - Maximum variance formulation
    - The variance of the projected data is maximized.
  - Minimum-error formulation
    - Minimizes the average projection cost

Maximum Variance Formulation

- The central idea of PCA is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set.

- This is achieved by transforming to a new set of variables, the principal components (PCs), which are uncorrelated, and are ordered by the fraction of the total information each retains, so that the first few retain most of the variation present in all of the original variables.

Algebraic Derivation of PCs

Given a sample set of m observations on a vector of d variables

$$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\} \in \mathbb{R}^d$$

Define the first PC of the samples by the linear projection $\boldsymbol{w}_1 \in \mathbb{R}^d$

$$z_{1i} = \boldsymbol{w_1^T x}_i = \sum_{k=1}^{d} w_{1k}\, x_{ik},\ i = 1, \ldots, m$$

where $\boldsymbol{w}_1 = (w_{11}, w_{12}, \ldots, w_{1d})^{\boldsymbol{T}}$ $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^{\boldsymbol{T}}$

$z_1 = \{z_{11}, z_{12}, \ldots, z_{1m}\}$

$\boldsymbol{w}_1$ is chosen such that $var[z_1]$ is maximum.

Algebraic Derivation of PCs

To find $\boldsymbol{w}_1$, first note that

$$var[z_1] = E[(z_1 - \bar{z}_1)^2] = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}_1^{\boldsymbol{T}}\boldsymbol{x}_i - \boldsymbol{w}_1^{\boldsymbol{T}}\bar{\boldsymbol{x}})^2$$

$$= \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{w}_1^{\boldsymbol{T}}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T\boldsymbol{w}_1 = \boldsymbol{w}_1^{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{w}_1$$

where $\boldsymbol{S} = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$ is the covariance matrix.

$\bar{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i$ is the mean.

Covariance and Correlation Coefficient

- For two random variables X and Y,

Covariance $COV[X,Y] = E[\{X - E[X]\}\{Y - E[Y]\}] = E[XY] - E[X]E[Y]$

The extent to which X and Y vary together.

$$|COV[X,Y]| \leq \sqrt{VAR[X]VAR[Y]}$$

Cauchy–Schwarz inequality.
柯西-施瓦茨不等式

- Correlation coefficient ρ (normalized covariance)

$$\rho(X,Y) = \frac{COV[X,Y]}{\sqrt{VAR[X]VAR[Y]}}$$

Interpretation of The Correlation Coefficient ρ

- Correlation coefficient ρ (normalized covariance)

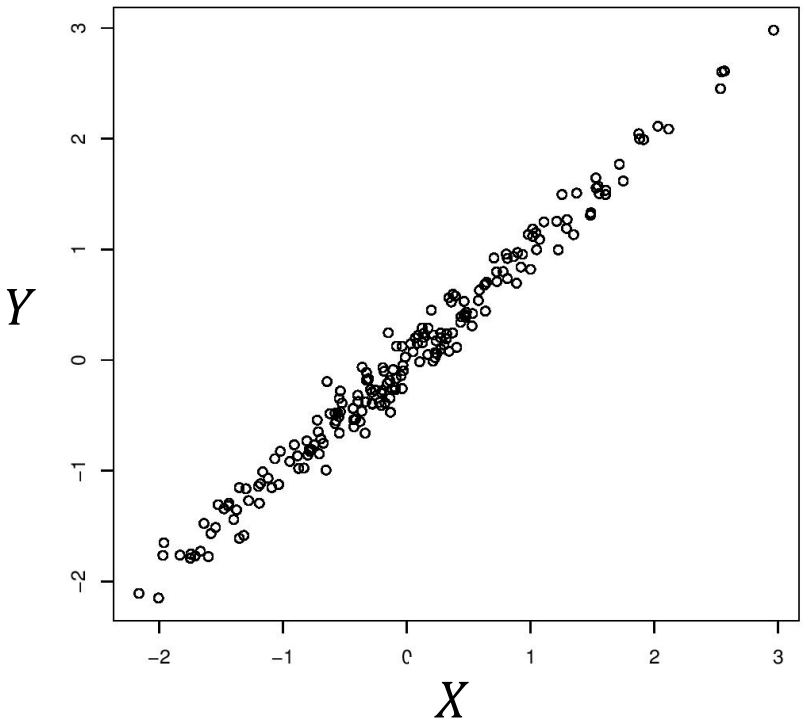
$$\rho(X,Y) = \frac{COV[X,Y]}{\sqrt{VAR[X]VAR[Y]}}$$

- $\rho(X,Y)$ measures the strength and direction of the linear relationship between X and Y.
- If X and Y have non-zero variance, then $\rho(X,Y) \in [-1,1]$.
- Y is a linearly increasing function of X if and only if $\rho(X,Y) = 1$
- Y is a linearly decreasing function of X if and only if $\rho(X,Y) = -1$
- X and Y are uncorrelated, if and only if $\rho(X,Y) = 0$

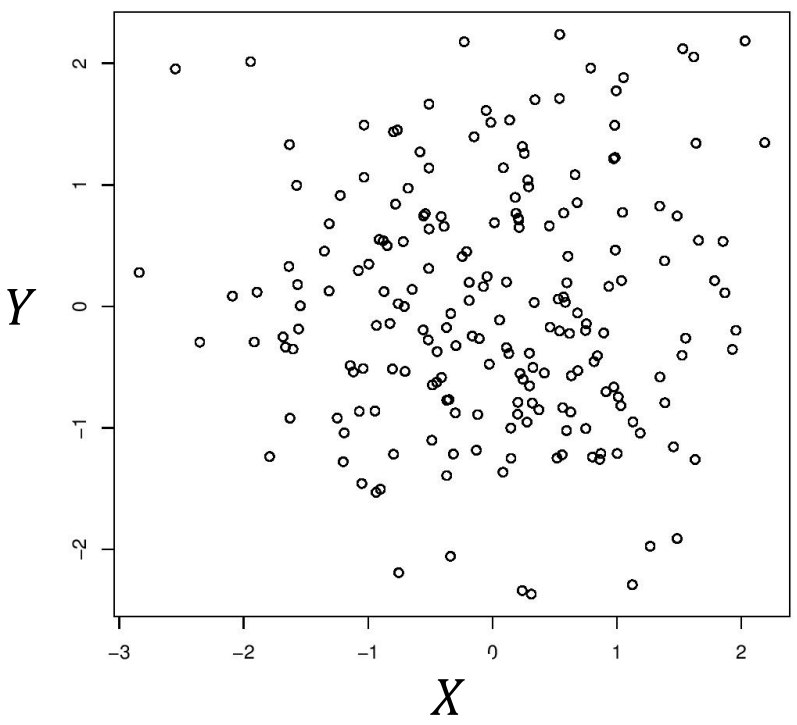
Interpretation of The Correlation Coefficient ρ

- Y is a linearly increasing function of X if and only if $\rho(X,Y)=1$
- Y is a linearly decreasing function of X if and only if $\rho(X,Y)=-1$
- X and Y are uncorrelated, if and only if $\rho(X,Y)=0$

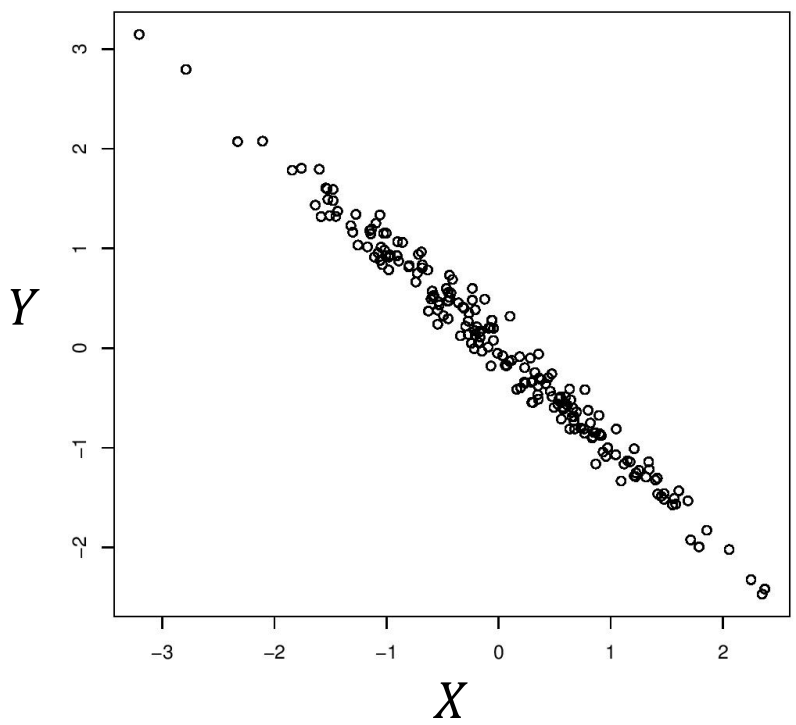
$\rho(X,Y)=0.99$



$\rho(X,Y)=0$



$\rho(X,Y)=-0.99$



Covariance Matrix

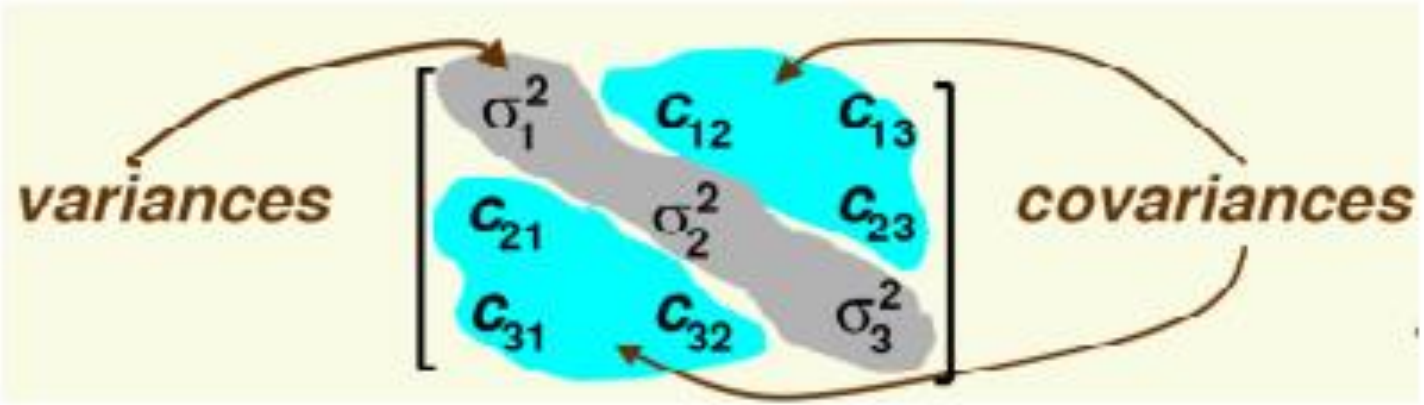
- Given random vector, $\vec{\mathbf{X}} = [\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{N}]^\mathrm{T}$, we define,

Mean vector $$E[\boldsymbol{X}] = [E[X_1], E[X_2], \ldots, E[X_N]]^T = [\mu_1 \mu_2 \ldots \mu_N] = \boldsymbol{\mu}$$

Covariance matrix

$$COV[\boldsymbol{X}] = \boldsymbol{\Sigma} = E[(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X} - \boldsymbol{\mu})^T]$$

$$= \begin{bmatrix} E[(X_1 - \mu_1)(X_1 - \mu_1)] & \ldots & E[(X_1 - \mu_1)(X_N - \mu_N)] \\ & \ddots & \\ E[(X_N - \mu_N)(X_1 - \mu_1)] & \ldots & E[(X_N - \mu_N)(X_N - \mu_N)] \end{bmatrix} = \begin{bmatrix} {\sigma_1}^2 & \ldots & c_{1N} \\ & \ldots & \\ c_{N1} & \ldots & {\sigma_N}^2 \end{bmatrix}$$



Covariance Matrix

- The covariance matrix indicates the tendency of each pair of dimensions (features) in a random vector to vary together, i.e., to co-vary.

- **Important Properties**
  - If x_i and x_k tend to increase together, then $c_{ik} > 0$
  - If x_i tends to decrease when x_k increases, then $c_{ik} < 0$
  - If x_i and x_k are uncorrelated, then $c_{ik} = 0$
  - $|c_{ik}| \leq \sigma_i \sigma_k$, where σ_i is the standard deviation of x_i
  - $c_{ii} = \sigma_i{}^2 = \mathrm{VAR}(x_i)$
  - Symmetric: $c_{ji} = c_{ij}$
  - Positive semi-definite:
    - Eigenvalues are nonnegative
    - Determinant is nonnegative, $|C| \geq 0$

Covariance Matrix

- You are given the heights and weights of a certain set of individuals in unknown units. Which one of the following four matrices is the most likely to be the sampled covariance matrix?

(a) $\begin{bmatrix} 1.232 & 0.867 \\ -0.867 & 2.791 \end{bmatrix}$

(b) $\begin{bmatrix} 1.232 & -0.867 \\ -0.867 & 2.791 \end{bmatrix}$

(c) $\begin{bmatrix} 1.232 & 0.867 \\ 0.867 & 2.791 \end{bmatrix}$

(d) $\begin{bmatrix} 1.232 & 3.307 \\ 3.307 & 2.791 \end{bmatrix}$

Algebraic Derivation of PCs

To find $\boldsymbol{w}_1$, first note that

$$var[z_1] = E[(z_1 - \bar{z}_1)^2] = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}_1^T\boldsymbol{x}_i - \boldsymbol{w}_1^T\bar{\boldsymbol{x}})^2$$

$$= \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{w}_1^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T\boldsymbol{w}_1 = \boldsymbol{w}_1^T\boldsymbol{S}\boldsymbol{w}_1$$

where $\boldsymbol{S} = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$ is the covariance matrix.

$\bar{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i$ is the mean.

The covariance matrix $\boldsymbol{S}$ is symmetric.

- The eigenvectors must be orthogonal（正交） to one another.
- The eigenvalues of $\boldsymbol{S}$ must all be ≥ 0

Algebraic Derivation of PCs

To find $\boldsymbol{w}_1$ that maximizes $var[z_1]$ subject to $\boldsymbol{w}_1^T\boldsymbol{w}_1$=1

We use the Lagrange multiplier, we maximize the following function:

$$L = \boldsymbol{w}_1^T\boldsymbol{S}\boldsymbol{w}_1 + \lambda(1 - \boldsymbol{w}_1^T\boldsymbol{w}_1)$$

$$\Rightarrow \quad \frac{\partial L}{\partial \boldsymbol{w}_1} = \boldsymbol{S}\boldsymbol{w}_1 - \lambda\boldsymbol{w}_1 = 0 \quad \Rightarrow \quad (\boldsymbol{S} - \lambda\boldsymbol{I}_d)\boldsymbol{w}_1 = 0$$

Eigenvectors and Eigenvalues

- **Definition:** $\boldsymbol{v}$ is an eigenvector of matrix $\boldsymbol{A} \in \mathbb{R}^{m*m}$ if there exists a scalar λ, such that:

$$\boldsymbol{Av} = \lambda \boldsymbol{v}$$

$\boldsymbol{v}$**:** an eigenvector (nonzero vector)
λ: the corresponding eigenvalue

- **Computation**

$$\boldsymbol{Av} = \lambda \boldsymbol{v} \qquad (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{v} = \boldsymbol{0}$$

$$\boldsymbol{v} \neq \boldsymbol{0} \quad \Rightarrow \qquad |\boldsymbol{A} - \lambda \boldsymbol{I}| = 0$$

Eigenvectors and Eigenvalues

- **Definition:** $\boldsymbol{v}$ is an eigenvector of matrix $\boldsymbol{A} \in \mathbb{R}^{m*m}$ if there exists a scalar λ, such that:

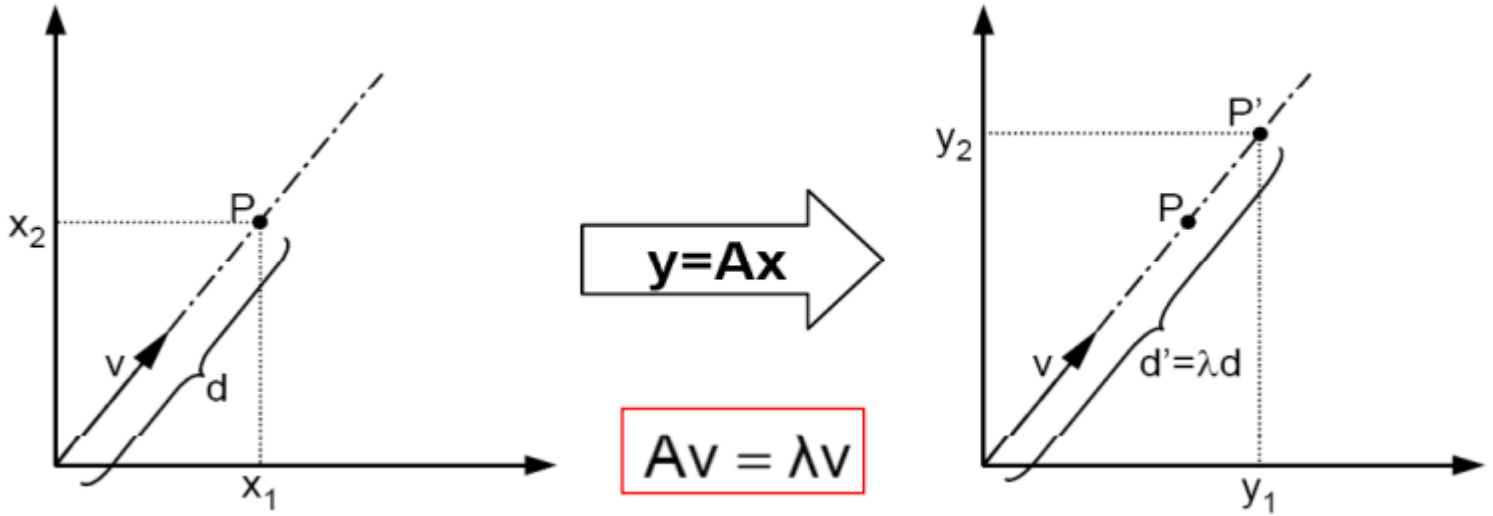
$$\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}$$

$\boldsymbol{v}$: an eigenvector
λ: the corresponding eigenvalue

- **Note**
  - $tr(\boldsymbol{A}) = \sum_i \lambda_i$
  - $|\boldsymbol{A}| = \prod_i \lambda_i$
  - If λ is an eigenvalue of the matrix $\boldsymbol{A}$, then λ^2 is an eigenvalue of $\boldsymbol{A}^2$. $(\boldsymbol{A}^2 = \boldsymbol{A}\boldsymbol{A})$
  - If λ is an eigenvalue of the matrix $\boldsymbol{A}$, then λ is an eigenvalue of $\boldsymbol{A}^T$.

Eigenvectors and Eigenvalues

- Intepretation: an eigenvector represents an invariant direction in the vector space.
  - Any point lying on the direction defined by $\boldsymbol{v}$ remains on that direction.
  - Its magnitude is multiplies by the corresponding eigenvalue λ



Algebraic Derivation of PCs

To find $\boldsymbol{w}_1$ that maximizes $var[z_1]$ subject to $\boldsymbol{w}_1^T\boldsymbol{w}_1$=1

We use the Lagrange multiplier, we maximize the following function:

$$L = \boldsymbol{w}_1^T\boldsymbol{S}\boldsymbol{w}_1 + \lambda(1 - \boldsymbol{w}_1^T\boldsymbol{w}_1)$$

$$\Rightarrow \quad \frac{\partial L}{\partial \boldsymbol{w}_1} = \boldsymbol{S}\boldsymbol{w}_1 - \lambda\boldsymbol{w}_1 = 0 \quad \Rightarrow \quad (\boldsymbol{S} - \lambda\boldsymbol{I}_d)\boldsymbol{w}_1 = 0$$

Therefore $\boldsymbol{w}_1$ is an eigenvector of $\boldsymbol{S}$, λ is the associated eigenvalue.

Which eigenvector should we choose?

Algebraic Derivation of PCs

If we recognize that the quantity to be maximized

$$L = \boldsymbol{w}_1^T \boldsymbol{S} \boldsymbol{w}_1 = \lambda$$

then we should choose λ to be as big as possible.

So $\lambda = \lambda_1$ is the largest eigenvalue, $\boldsymbol{w}_1$ is the corresponding eigenvector.

We have got the 1st PC. Now, let's go to the 2nd PC.

Algebraic Derivation of PCs

The 2nd PC, $\boldsymbol{w_2^T x}$ maximize $\boldsymbol{w_2^T S w_2}$ subject to being uncorrelated with $\boldsymbol{w_1^T x}$

The uncorrelation constraint can be expressed as

$$COV\left(\boldsymbol{w_1^T x}, \boldsymbol{w_2^T x}\right) = \boldsymbol{w_1^T S w_2} = \boldsymbol{w_1^T} \lambda_1 \boldsymbol{w_2} = \lambda_1 \boldsymbol{w_1^T w_2} = 0$$

Using Lagrange multiplier, we will maximize

$$\boldsymbol{w_2^T S w_2} + \lambda_2\left(1 - \boldsymbol{w_2^T w_2}\right) - \varphi \boldsymbol{w_1^T w_2}$$

Algebraic Derivation of PCs

Differentiation *w.r.t.* $\boldsymbol{w}_2$ yields,

$$\frac{d(\boldsymbol{w}_2^T \boldsymbol{S} \boldsymbol{w}_2 + \lambda_2(1 - \boldsymbol{w}_2^T \boldsymbol{w}_2) - \varphi \boldsymbol{w}_1^T \boldsymbol{w}_2)}{d\boldsymbol{w}_2} = \boldsymbol{0}$$

$$2\boldsymbol{S}\boldsymbol{w}_2 - 2\lambda_2 \boldsymbol{w}_2 - \varphi \boldsymbol{w}_1 = \boldsymbol{0}$$

If we left multiply $\boldsymbol{w}_1$,

$$\boldsymbol{w}_1^T \boldsymbol{S} \boldsymbol{w}_2 - \lambda_2 \boldsymbol{w}_1^T \boldsymbol{w}_2 - \frac{\varphi}{2} \boldsymbol{w}_1^T \boldsymbol{w}_1 = 0$$

what does it imply?

Algebraic Derivation of PCs

$$\boldsymbol{w}_1^T \boldsymbol{S} \boldsymbol{w}_2 - \lambda_2 \boldsymbol{w}_1^T \boldsymbol{w}_2 - \frac{\varphi}{2} \boldsymbol{w}_1^T \boldsymbol{w}_1 = 0$$

$$0 \quad - \quad 0 \quad - \frac{\varphi}{2} \quad 1 \quad = 0$$

φ must be zero and then we re-check the derivative,

$$\boldsymbol{S} \boldsymbol{w}_2 - \lambda_2 \boldsymbol{w}_2 - \varphi \boldsymbol{w}_1 = \boldsymbol{0} \quad \Rightarrow \quad \boldsymbol{S} \boldsymbol{w}_2 - \lambda_2 \boldsymbol{w}_2 = \boldsymbol{0}$$

The same strategy of choosing $\boldsymbol{w}_2$ to be the eigenvector associated with the second largest eigenvalue λ_2 yields the second PC.

Algebraic Derivation of PCs

This process can be repeated for $k = 1, \ldots, p$ yielding up to p different eigenvectors of $\boldsymbol{S}$ along with the corresponding eigenvalues $\lambda_1, \ldots \lambda_p$

The variance of each of the PCs are given by

$$var[z_k] = var\left[\boldsymbol{w}_k^T \boldsymbol{x}\right] = \lambda_k, k = 1, \ldots, p$$

Algebraic Derivation of PCs

- Main steps for computing PCs:
  - Form the covariance matrix $\boldsymbol{S}$.
  - Compute its eigenvectors: $\{\boldsymbol{w}_i\}_{i=1}^{d}$
  - The first p eigenvectors $\{\boldsymbol{w}_i\}_{i=1}^{p}$ form the p PCs
  - The transformation $\boldsymbol{G}$ consists of the p PCs

$$\boldsymbol{G} \leftarrow \left[\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_p\right] \in \mathbb{R}^{d \times p}$$

$$\boldsymbol{y} = \boldsymbol{G}^{\boldsymbol{T}} \boldsymbol{x} \in \mathbb{R}^{p}$$

Principal Component Analysis (PCA)

Principal component analysis, or PCA, is a technique that is widely used for applications such as dimensionality reduction, lossy data compression, feature extraction, and data visualization (Jolliffe, 2002).

- Two commonly used definitions of PCA
  - Maximum variance formulation
    - The variance of the projected data is maximized.
  - Minimum-error formulation
    - Minimizes the average projection cost

Minimum-error Formulation

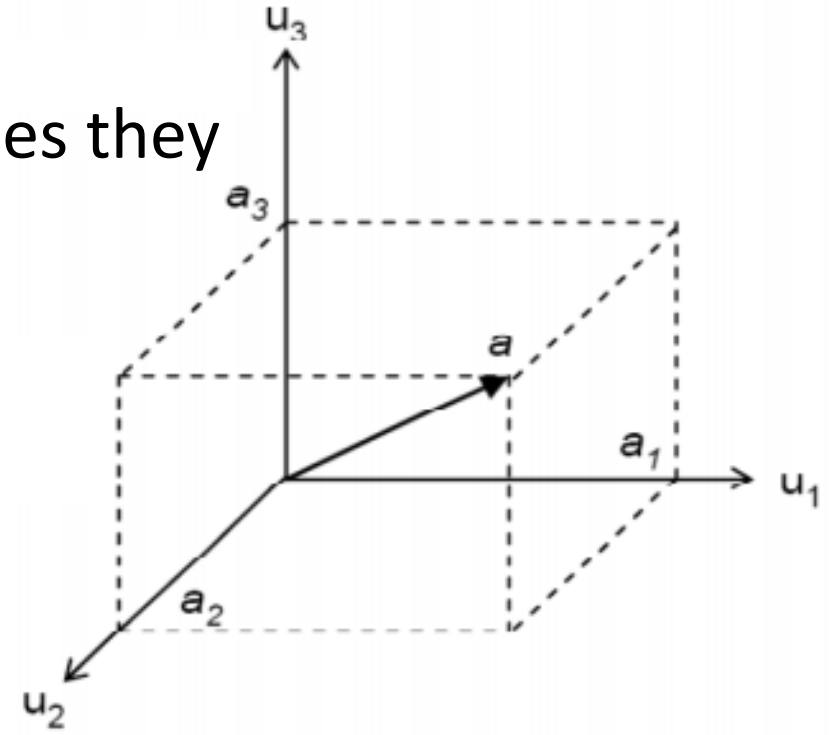
- **Basis**
  - A set of vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n\}$ are called a ***basis*** for a vector space $\mathbb{R}^n$ if any vector $\boldsymbol{x} \in \mathbb{R}^n$ can be written as a linear combination of $\{\boldsymbol{u}_i\}$

$$\boldsymbol{x} = a_1\boldsymbol{u}_1 + a_2\boldsymbol{u}_2 + \cdots + a_n\boldsymbol{u}_n$$

  - $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n$ are linearly **independent** implies they form a basis and vice versa.

$$a_1\boldsymbol{u}_1 + a_2\boldsymbol{u}_2 + \cdots + a_n\boldsymbol{u}_n = \mathbf{0} \Rightarrow a_k = 0$$

  - A basis $\{\boldsymbol{u}_i\}$ is **orthonormal** if
    - Basis vectors are pairwise orthogonal
    - Have unit length, i.e., $|\boldsymbol{u}_i| = 1$.



Minimum-error Formulation

Here, we first introduce a complete orthonormal set of basis vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_d\} \in \mathbb{R}^d$, where

$$\boldsymbol{u}_i^T \boldsymbol{u}_j = \delta_{ij} \ （0 \ or \ 1）$$

Each data point can be represented by a linear combination of the basis vectors:

$$\boldsymbol{x}_n = \sum_{i=1}^{d} \alpha_{ni} \boldsymbol{u}_i$$

$$\alpha_{ni} = \boldsymbol{x}_n^T \boldsymbol{u}_i \qquad \boldsymbol{x}_n = \sum_{i=1}^{d} (\boldsymbol{x}_n^T \boldsymbol{u}_i) \boldsymbol{u}_i$$

Minimum-error Formulation

- Our goal is to approximate $\boldsymbol{x}_n \in \mathbb{R}^d$ with $\tilde{\boldsymbol{x}}_n \in \mathbb{R}^p$, $p < d$ (projection onto a lower-dimensional subspace).
- The p-D subspace can be represented, without loss of generality, by the first p of the basis vectors, and we approximate each sample $\boldsymbol{x}_n$ by

$$\tilde{\boldsymbol{x}}_n = \sum_{i=1}^{p} z_{ni}\,\boldsymbol{u}_i + \sum_{i=p+1}^{d} b_i\,\boldsymbol{u}_i$$

- $\{z_{ni}\}$ depend on the particular data point.
- $\{b_i\}$ are constants that are the same for all data points.

We aim to minimize the loss introduced by the dimensionality reduction, i.e., the squared distance between the original $\boldsymbol{x}_n$ and approximation $\tilde{\boldsymbol{x}}_n$

$$J = \frac{1}{m}\sum_{n=1}^{m} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2$$

Minimum-error Formulation

To minimize $J = \frac{1}{m}\sum_{n=1}^{m} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2$ $\quad \tilde{\boldsymbol{x}}_n = \sum_{i=1}^{p} z_{ni}\,\boldsymbol{u}_i + \sum_{i=p+1}^{d} b_i\,\boldsymbol{u}_i$

- Taking the derivative w.r.t. z_{nj} and setting to zero, we have

$$z_{nj} = \boldsymbol{x}_n^T \boldsymbol{u}_j, \qquad j = 1, \ldots, p$$

- Taking the derivative w.r.t. b_j and setting to zero, we have

$$b_j = \bar{\boldsymbol{x}}^T \boldsymbol{u}_j, \qquad j = p+1, \ldots, d$$

Leaving for Your homework.

- Substituting z_{nj} and b_j, and make use of $\boldsymbol{x}_n = \sum_{i=1}^{d} (\boldsymbol{x}_n^T \boldsymbol{u}_i)\,\boldsymbol{u}_i$, we have

$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \sum_{i=p+1}^{d} \{(\boldsymbol{x}_n - \bar{\boldsymbol{x}})^T \boldsymbol{u}_i\}\,\boldsymbol{u}_i$$

$$x_n - \hat{x}_n = x_n - \sum_{i=1}^{p} z_{ni} u_i - \sum_{i=p+1}^{d} b_i u_i.$$

$$= x_n - \sum_{i=1}^{p} x_n u_i u_i - \sum_{i=p+1}^{d} \bar{x}^T u_i u_i$$

$$= \sum_{i=1}^{d} x_n^T u_i u_i - \sum_{i=1}^{p} x_n u_i u_i - \sum_{i=p+1}^{d} \bar{x}^T u_i u_i$$

$$= \sum_{i=1}^{p} x_n^T u_i u_i + \sum_{i=p+1}^{d} x_n^T u_i u_i - \sum_{i=1}^{p} x_n u_i u_i - \sum_{i=p+1}^{d} \bar{x}^T u_i u_i.$$

$$= \sum_{i=p+1}^{d} (x_n^T - \bar{x}^T) u_i u_i = \sum_{i=p+1}^{d} (x_n - \bar{x})^T u_i u_i.$$

Minimum-error Formulation $\tilde{\boldsymbol{x}}_n = \sum_{i=1}^{p} z_{ni}\,\boldsymbol{u}_i + \sum_{i=p+1}^{d} b_i\,\boldsymbol{u}_i$

$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \sum_{i=p+1}^{d} \{(\boldsymbol{x}_n - \bar{\boldsymbol{x}})^T \boldsymbol{u}_i\}\,\boldsymbol{u}_i$$

- $\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n$ lies in the space orthogonal to the principal subspace, why?
- Then

$$\min J = \frac{1}{m}\sum_{n=1}^{m} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2 = \frac{1}{m}\sum_{n=1}^{m}\sum_{i=p+1}^{d} (\boldsymbol{x}_n^T\boldsymbol{u}_i - \bar{\boldsymbol{x}}^T\boldsymbol{u}_i)^2 = \sum_{i=p+1}^{d} \boldsymbol{u}_i^T \boldsymbol{S}\boldsymbol{u}_i$$

- Let us try to obtain some intuition about the result by considering the case of $d = 2$ and $p = 1$.

Minimum-error Formulation

$$J = \sum_{i=p+1}^{d} \boldsymbol{u}_i^T \boldsymbol{S} \boldsymbol{u}_i,\ d = 2, p = 1$$

- We choose $\boldsymbol{u}_2$ to minimize $J = \boldsymbol{u}_2^T \boldsymbol{S} \boldsymbol{u}_2$.
- Using Lagrange multiplier, we know $\boldsymbol{u}_2$ should be the eigenvector corresponding to the smaller of the two eigenvalues.
- Thus we should choose the principle subspace to be aligned with the eigenvector having the larger eigenvalue.

- For the case when the eigenvalues are equal, any choice of principal direction will give rise to the same value of J.

Minimum-error Formulation

- Generally, the solution to the minimization of J is obtained by choosing the $\{\boldsymbol{u}_i\}$ to be the eigenvectors of the covariance matrix given by $\boldsymbol{S}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i, i = 1, \ldots, d.$
- The corresponding value of the loss measure is then given by

$$J = \sum_{i=p+1}^{d} \lambda_i$$

- Eigenvectors defining the principal subspace are those corresponding to the p largest eigenvalues.

Eigenvalue Decomposition

- Given a **square** matrix with m linearly **independent** eigenvectors $\boldsymbol{A} \in \mathbb{R}^{m*m}$, we have an eigenvalue decomposition

$$\boldsymbol{AV} = \boldsymbol{V\Lambda} \qquad \boldsymbol{A} = \boldsymbol{V\Lambda V^{-1}}$$

- **Note**
  - Columns of $\boldsymbol{V}$ are eigenvectors of $\boldsymbol{A}$
  - Diagonal elements of $\boldsymbol{\Lambda}$ are eigenvalues of $\boldsymbol{A}$

$$\boldsymbol{\Lambda} = diag(\lambda_1, \lambda_2, \dots, \lambda_m), \lambda_i \geq \lambda_{i+1}$$

Eigenvalue Decomposition

- **Note**
- If $\boldsymbol{A}$ is non-singular
  - All eigenvalues are non-zero
- If $\boldsymbol{A}$ is real and symmetric
  - All eigenvalues are *real*.

$$\text{If } |\boldsymbol{A} - \lambda \boldsymbol{I}| = 0 \text{ and } \boldsymbol{A} = \boldsymbol{A}^{\boldsymbol{T}} \Rightarrow \lambda \in \mathbb{R}$$

  - The eigenvectors for distinct eigenvalues are *orthogonal*.

$$\boldsymbol{A}\boldsymbol{v}_{\{1,2\}} = \lambda_{\{1,2\}}\boldsymbol{v}_{\{1,2\}} \text{ and } \lambda_1 \neq \lambda_2 \Rightarrow \boldsymbol{v}_1 \cdot \boldsymbol{v}_2 = 0$$

- If A is positive definite, then all eigenvalues are *positive*.

$$\forall \boldsymbol{w} \in \mathbb{R}^n, \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{A}\boldsymbol{w} > 0, \text{ then if } \boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v} \Rightarrow \lambda > 0$$

Singular Value Decomposition (SVD)

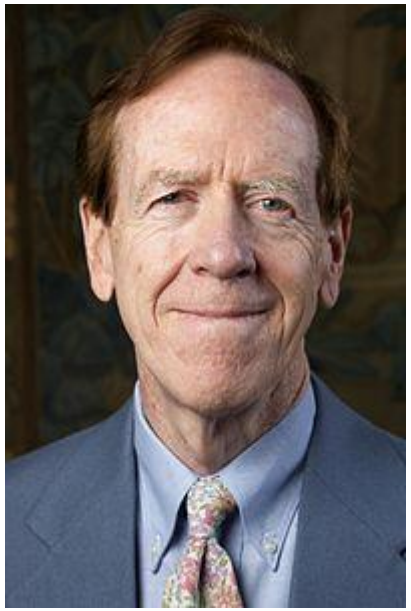
- In practice, we compute the PCs via singular value decomposition (SVD) on the centered data matrix.

Singular Value Decomposition (SVD)

- **"The highpoint of linear algebra"---** *Gillert Strang*

Awards and honors [edit]

- Rhodes Scholar (1955)
- Alfred P. Sloan Fellow (1966–1967)
- Chauvenet Prize, Mathematical Association of America (1976)
- Honorary Professor, Xian Jiaotong University, China (1980)
- American Academy of Arts and Sciences (1985)
- Honorary Fellow, Balliol College, Oxford University (1999)
- Honorary Member, Irish Mathematical Society (2002)
- Award for Distinguished Service to the Profession, Society for Industrial and Applied Mathematics (2003)
- Lester R. Ford Award (2005)[3]
- Von Neumann Medal, US Association for Computational Mechanics (2005)
- Haimo Prize, Mathematical Association of America (2007)[4]
- Su Buchin Prize, International Congress (ICIAM, 2007)
- Henrici Prize (2007)
- National Academy of Sciences (2009)
- Doctor Honoris Causa, University of Toulouse (2010)
- Fellow of the American Mathematical Society (2012)[5]
- Doctor Honoris Causa, Aalborg University (2013)
- Fellow of the Society for Industrial and Applied Mathematics (2009) [6]



1934~

SVD

rank: the number of linearly independent rows (or columns)

$$0 \leq rank(\boldsymbol{A}) \leq min(m, n)$$

- Any $m \times n$ matrix $\boldsymbol{A}$ of rank r can be decomposed into: $\boldsymbol{A} = \boldsymbol{U\Sigma V^T}$
  - For $m > n$

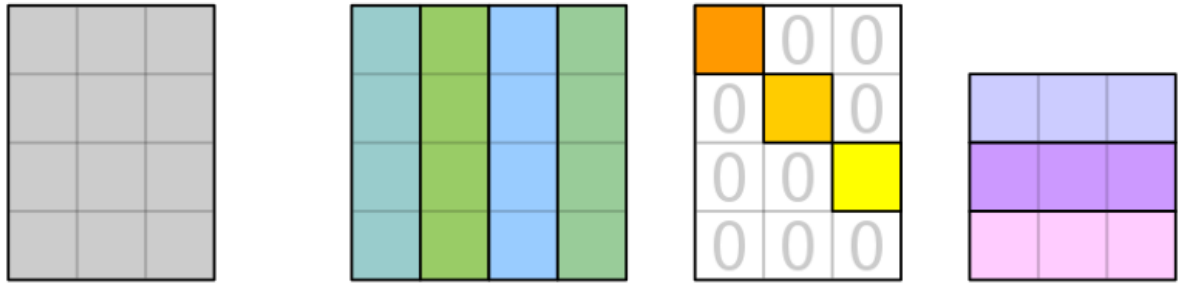
$\boldsymbol{U}_{m\times m}$ $\boldsymbol{\Sigma}_{m\times n}$ $\boldsymbol{V}_{n\times n}$

$$\mathbf{A} = \begin{bmatrix} \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{u}_1 & \cdots & \mathbf{u}_r & \mathbf{u}_{r+1} & \cdots & \mathbf{u}_m \\ \vdots & & \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} \cdots & \mathbf{v}_1^\top & \cdots \\ & \vdots & \\ \cdots & \mathbf{v}_r^\top & \cdots \\ \cdots & \mathbf{v}_{r+1}^\top & \cdots \\ & \vdots & \\ \cdots & \mathbf{v}_n^\top & \cdots \end{bmatrix}$$

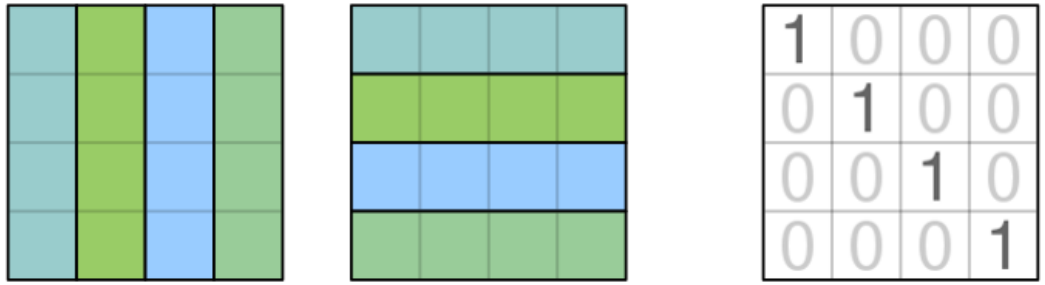
- Orthogonal matrix $\boldsymbol{U}$: $\boldsymbol{U^T U} = \boldsymbol{I}_{m\times m}$, and $\boldsymbol{UU^T} = \boldsymbol{I}_{m\times m}$.
  - The columns of $\boldsymbol{U}$ are defined as *left singular vectors*.
- Orthogonal matrix $\boldsymbol{V}$: $\boldsymbol{V^T V} = \boldsymbol{I}_{n\times n}$, and $\boldsymbol{VV^T} = \boldsymbol{I}_{n\times n}$.
  - The columns of $\boldsymbol{V}$ are defined as *right singular vectors*.
- Rectangular diagonal matrix $\boldsymbol{\Sigma}$: we arrange the *singular values* as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

An orthogonal matrix is a square matrix whose columns and rows are orthogonal unit vectors (orthonormal vectors).

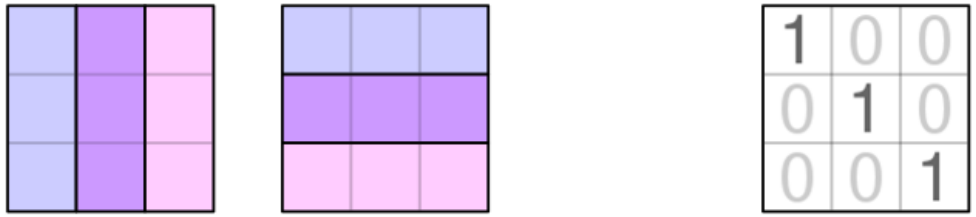
SVD



$$\underset{m\times n}{\mathbf{A}} = \underset{m\times m}{\mathbf{U}}\ \underset{m\times n}{\mathbf{\Sigma}}\ \underset{n\times n}{\mathbf{V}^*}$$



$$\mathbf{U}\ \mathbf{U}^* = \mathbf{I}_m$$



$$\mathbf{V}\ \mathbf{V}^* = \mathbf{I}_n$$

Singular Value Decomposition (SVD)

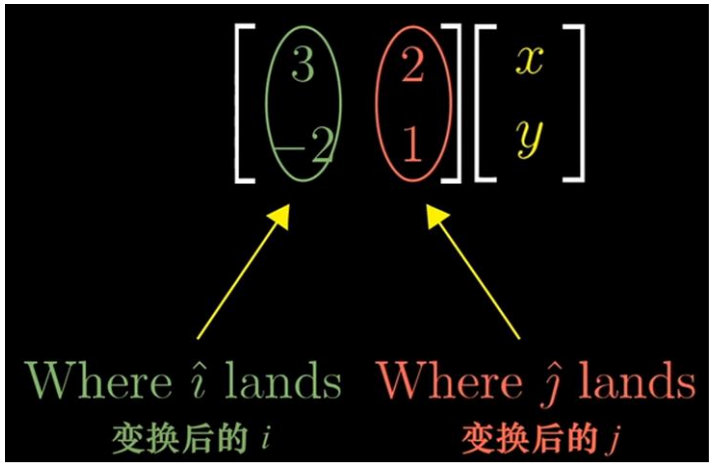
- $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices. The columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ are basis vectors in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively.

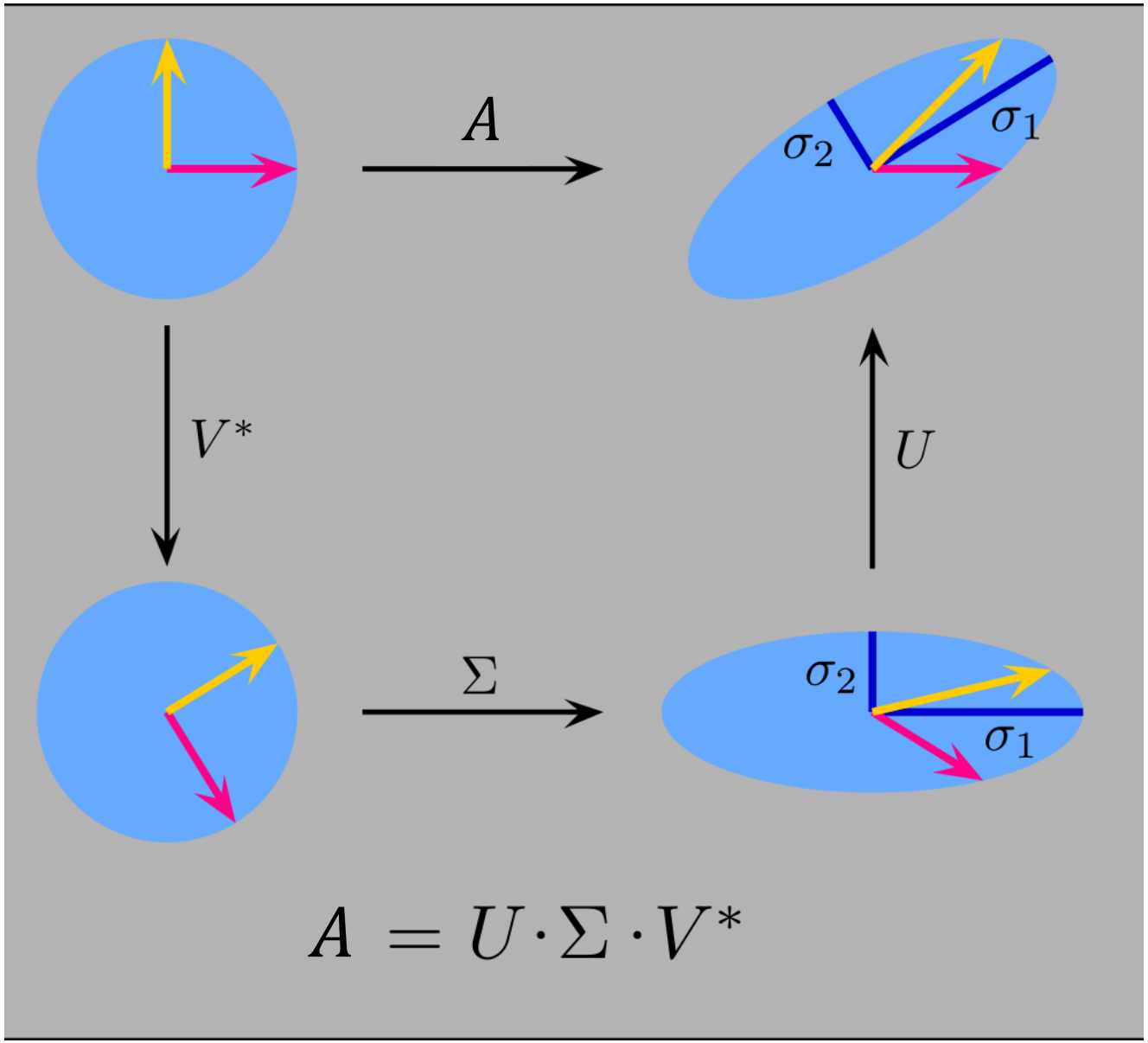
The linear transformation $\boldsymbol{A}$ can be interpreted as a composition of three geometrical transformations: a rotation or reflection ($\boldsymbol{V^T}$), followed by a coordinate-by-coordinate scaling ($\boldsymbol{\Sigma}$), followed by another rotation or reflection ($\boldsymbol{U}$).

Singular Value Decomposition (SVD)

Illustration of the SVD of a real 2×2 matrix $\boldsymbol{A = U\Sigma V^T}$.

- **Top**: The action of $\mathbf{A}$.
- **Left**: The action of $\boldsymbol{V^T}$, a rotation.
- **Bottom**: The action of $\boldsymbol{\Sigma}$, a scaling by the singular values σ_1 horizontally and σ_2 vertically.
- **Right**: The action of $\boldsymbol{U}$, another rotation.





Singular Value Decomposition (SVD)

Given the matrix $\boldsymbol{A} = \begin{bmatrix} 3 & 1 \\ 2 & 1 \end{bmatrix}$

$$\boldsymbol{U} = \begin{bmatrix} 0.8174 & -0.5760 \\ 0.5760 & 0.8174 \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 3.8643 & 0 \\ 0 & 0.2588 \end{bmatrix} \qquad \boldsymbol{V^T} = \begin{bmatrix} 0.9327 & 0.3606 \\ -0.3606 & 0.9327 \end{bmatrix}$$

Let us focus on the transformation of the original basis vectors by $\boldsymbol{A}$

$$\boldsymbol{e_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \boldsymbol{e_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

First, by rotation $\boldsymbol{V^T}$, we have

$$\boldsymbol{V^T e_1} = \begin{bmatrix} 0.9327 \\ -0.3606 \end{bmatrix} \qquad \boldsymbol{V^T e_2} = \begin{bmatrix} 0.3606 \\ 0.9327 \end{bmatrix}$$

Singular Value Decomposition (SVD)

Given the matrix $\boldsymbol{A} = \begin{bmatrix} 3 & 1 \\ 2 & 1 \end{bmatrix}$

$$\boldsymbol{U} = \begin{bmatrix} 0.8174 & -0.5760 \\ 0.5760 & 0.8174 \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 3.8643 & 0 \\ 0 & 0.2588 \end{bmatrix} \qquad \boldsymbol{V^T} = \begin{bmatrix} 0.9327 & 0.3606 \\ -0.3606 & 0.9327 \end{bmatrix}$$

Second, by $\boldsymbol{\Sigma}$, we scale $\boldsymbol{V^T e_1}$ and $\boldsymbol{V^T e_2}$ along with the original coordinate system ($\boldsymbol{e_1}$, $\boldsymbol{e_2}$)

$$\boldsymbol{\Sigma V^T e_1} = \begin{bmatrix} 3.6042 \\ -0.0933 \end{bmatrix} \qquad \boldsymbol{\Sigma V^T e_2} = \begin{bmatrix} 1.3935 \\ 0.2414 \end{bmatrix}$$

Last, by rotation $\boldsymbol{U}$, we have

$$\mathbf{U\Sigma V^T e_1} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \qquad \mathbf{U\Sigma V^T e_2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Singular Value Decomposition (SVD)

- Any $m \times n$ matrix $\boldsymbol{A}$ of rank r can be decomposed into: $\boldsymbol{A} = \boldsymbol{U\Sigma V^T}$
  - For $m > n$

$$\underset{\boldsymbol{U}_{m\times m}}{\begin{bmatrix} \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{u}_1 & \cdots & \mathbf{u}_r & \mathbf{u}_{r+1} & \cdots & \mathbf{u}_m \\ \vdots & & \vdots & \vdots & & \vdots \end{bmatrix}} \underset{\boldsymbol{\Sigma}_{m\times n}}{\begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}} \underset{\boldsymbol{V}_{n\times n}}{\begin{bmatrix} \cdots & \mathbf{v}_1^\top & \cdots \\ & \vdots & \\ \cdots & \mathbf{v}_r^\top & \cdots \\ \cdots & \mathbf{v}_{r+1}^\top & \cdots \\ & \vdots & \\ \cdots & \mathbf{v}_n^\top & \cdots \end{bmatrix}}$$

- **Special Properties**:
  - The columns of $\boldsymbol{U}$ (i.e., left singular vectors) are eigenvectors of $\boldsymbol{AA^T}$.
  - The columns of $\boldsymbol{V}$ (i.e., right singular vectors) are eigenvectors of $\boldsymbol{A^TA}$.
  - Eigenvalues $\lambda_1, \ldots, \lambda_r$ of $\boldsymbol{AA^T}$ are the eigenvalues of $\boldsymbol{A^TA}$.
  - Singular value $\sigma_i = \sqrt{\lambda_i}$.

Singular Value Decomposition (SVD)

- **Prove that the columns of V (i.e., right singular vectors) are eigenvectors of A^TA.**

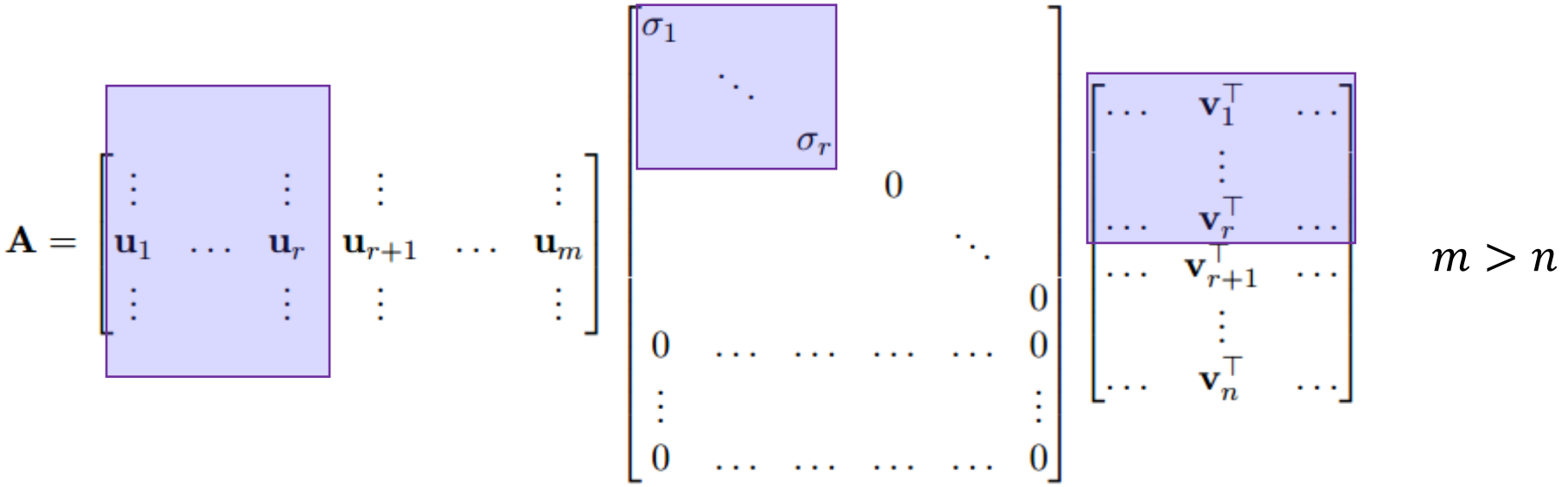
$A^TA = (U\Sigma V^T)^T U\Sigma V^T = V\Sigma^T\Sigma V^T$ This is the eigen-decomposition of A^TA.

V is the eigenvector matrix of A^TA, and $\Sigma^T\Sigma$ is the eigenvalue matrix of A^TA, i.e., singular values are positive square roots of eigenvalues.

Theorem1: Every symmetric matrix M is orthogonally diagonalizable, i.e., there exists an orthogonal matrix Q (i.e., $Q^T = Q^{-1}$) such that $Q^TMQ = D$ (i.e., $M = QDQ^T$) is a diagonal matrix. (https://en.wikipedia.org/wiki/Diagonalizable_matrix)

Compact SVD

- Only the $r = rank(\boldsymbol{A})$ column vectors of $\boldsymbol{U}$ and r row vectors of $\boldsymbol{V}^{\boldsymbol{T}}$ corresponding to the non-zero singular values $\boldsymbol{\Sigma}_r$ are calculated.

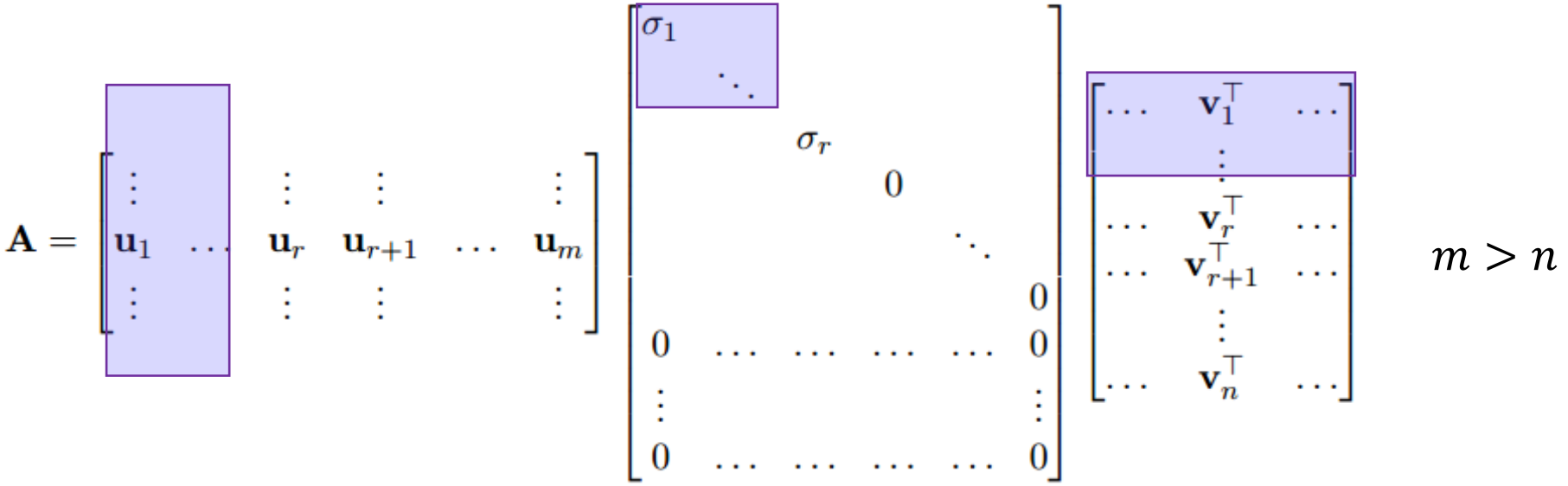


- Economy version $\boldsymbol{A} = \underbrace{\boldsymbol{U}_r}_{m \times r}\underbrace{\boldsymbol{\Sigma}_r}_{r \times r}\underbrace{\boldsymbol{V}_r{}^{\boldsymbol{T}}}_{r \times n}$ $\quad \boldsymbol{\Sigma}_r = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$

Any information loss?

Truncated SVD

- Only k column vectors of $\boldsymbol{U}$ and k row vectors of $\boldsymbol{V^T}$ corresponding to the non-zero singular values $\boldsymbol{\Sigma}_k$ are calculated, $0 < k < r, r = rank(\boldsymbol{A})$.



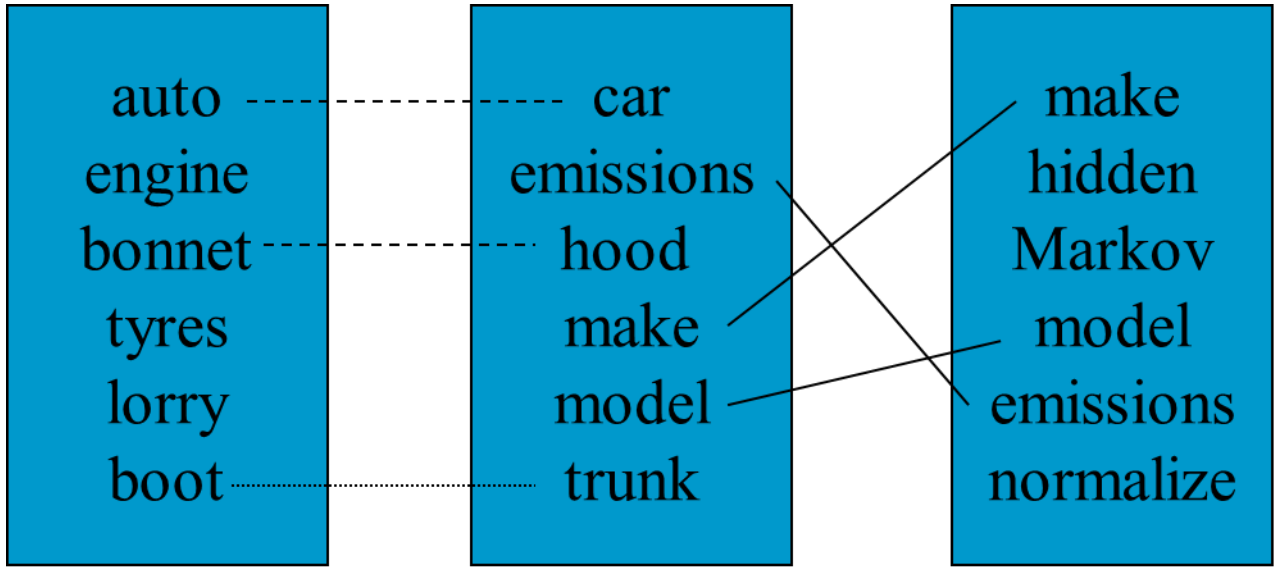
- More Economical $\boldsymbol{A} = \underbrace{\boldsymbol{U}_k}_{m \times k}\underbrace{\boldsymbol{\Sigma}_k}_{k \times k}\underbrace{\boldsymbol{V}_k^{\boldsymbol{T}}}_{k \times n}$ $\boldsymbol{\Sigma}_k = \mathrm{diag}(\sigma_1, \ldots, \sigma_k)$

Truncated SVD is no longer an exact decomposition of the original matrix.

SVD Application1-Latent Semantic Indexing (LSI)

- LSI was proposed to address two problems with the vector space model
  - **synonymy**: many ways to refer to the same object, e.g. *car* and *automobile*
    - leads to poor recall (*recall*: portion of the target items that the system selected)
  - **polysemy**: most words have more than one distinct meaning, e.g. *model*, *python*, *chip*
    - leads to poor precision (*precision*: portion of selected items that the system got right)

Index Words	Titles								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				



Synonymy

Will have small cosine

but are related

Polysemy

Will have large cosine

but not truly related

SVD Application1-Latent Semantic Indexing (LSI)

- LSI is a technique that projects queries and documents into a space with “latent” semantic dimensions.
- In the latent semantic space, a query and a document can have high cosine similarity even if they don’t share any terms, as long as their terms are semantically similar in a sense.

Index Words	Titles								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				

A

$k = 3$

book	0.15	-0.27	0.04
dads	0.24	0.38	-0.09
dummies	0.13	-0.17	0.07
estate	0.18	0.19	0.45
guide	0.22	0.09	-0.46
investing	0.74	-0.21	0.21
market	0.18	-0.30	-0.28
real	0.18	0.19	0.45
rich	0.36	0.59	-0.34
stock	0.25	-0.42	-0.28
value	0.12	-0.14	0.23

U_k

3.91	0	0
0	2.61	0
0	0	2.00

Σ_k

T1	T2	T3	T4	T5	T6	T7	T8	T9
0.35	0.22	0.34	0.26	0.22	0.49	0.28	0.29	0.44
-0.32	-0.15	-0.46	-0.24	-0.14	0.55	0.07	-0.31	0.44
-0.41	0.14	-0.16	0.25	0.22	-0.51	0.55	0.00	0.34

V_k^T

SVD Application2-Pseudoinverse

- SVD can be used for computing the pseudoinverse of a matrix.

$$\boldsymbol{A = U\Sigma V^T}$$

$$\mathbf{A} = \begin{bmatrix} \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{u}_1 & \cdots & \mathbf{u}_r & \mathbf{u}_{r+1} & \cdots & \mathbf{u}_m \\ \vdots & & \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} \cdots & \mathbf{v}_1^\top & \cdots \\ & \vdots & \\ \cdots & \mathbf{v}_r^\top & \cdots \\ \cdots & \mathbf{v}_{r+1}^\top & \cdots \\ & \vdots & \\ \cdots & \mathbf{v}_n^\top & \cdots \end{bmatrix}$$

$$\boldsymbol{A^\dagger = V\Sigma^\dagger U^T}$$

$\boldsymbol{\Sigma^\dagger}$: pseudoinverse of $\boldsymbol{\Sigma}$ by replacing every non-zero diagonal entry by its reciprocal and transposing the matrix.

$$\begin{bmatrix} \frac{1}{\sigma_1} & & & & 0 \cdots 0 \\ & \ddots & & & \vdots \quad \vdots \\ & & \frac{1}{\sigma_r} & & \vdots \quad \vdots \\ & & & 0 & \vdots \quad \vdots \\ & & & & \ddots \quad \vdots \quad \vdots \\ & & & & 0 \; 0 \cdots 0 \end{bmatrix}$$

$$\boldsymbol{AA^\dagger A = U\Sigma V^T V\Sigma^\dagger U^T U\Sigma V^T = U\Sigma\Sigma^\dagger\Sigma V^T = U\Sigma V^T = A}$$

SVD Application3-PCA

- In practice, we compute the PCs via singular value decomposition (SVD) on the centered data matrix.

- Form the centered data matrix:

$$\boldsymbol{X} = [\,(\boldsymbol{x}_1 - \overline{\boldsymbol{x}}); \ldots; (\boldsymbol{x}_m - \overline{\boldsymbol{x}})] \in \mathbb{R}^{d \times m}$$

- Compute its SVD:

$$\boldsymbol{X} = \boldsymbol{U}_{d\times d}\boldsymbol{D}_{d\times m}(\boldsymbol{V}_{m\times m})^{\boldsymbol{T}}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices, $\boldsymbol{D}$ is a diagonal matrix.

SVD Application3-PCA

- Note that the scatter/covariance matrix can be written as

$$\boldsymbol{S} = \boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}} = \boldsymbol{U}\boldsymbol{D}^{\boldsymbol{2}}\boldsymbol{U}^{\boldsymbol{T}} \qquad \boldsymbol{X} = \boldsymbol{U}_{d\times d}\boldsymbol{D}_{d\times m}(\boldsymbol{V}_{m\times m})^{\boldsymbol{T}}$$

- The eigenvectors of $\boldsymbol{S}$ are the columns of $\boldsymbol{U}$ and the eigenvalues are the diagonal elements of $\boldsymbol{D}^{\boldsymbol{2}}$.

- Take only a few significant eigenvalue-eigenvector pairs $p \ll d$. The new reconstructed sample from low-dim space is:

$$\hat{\boldsymbol{x}}_{\boldsymbol{i}} = \overline{\boldsymbol{x}} + \boldsymbol{U}_{d\times p}\left(\boldsymbol{U}_{d\times p}\right)^{\boldsymbol{T}}(\boldsymbol{x}_{\boldsymbol{i}} - \overline{\boldsymbol{x}})$$

SVD Application4-Reconstruction

Original

Reconstruction using 50 PCs

$$\hat{\boldsymbol{x}}_{\boldsymbol{i}} = \overline{\boldsymbol{x}} + \boldsymbol{U}_{d\times p}\left(\boldsymbol{U}_{d\times p}\right)^{\boldsymbol{T}}(\boldsymbol{x}_{\boldsymbol{i}} - \overline{\boldsymbol{x}})$$

Advantages of Using SVD for PCA

- No need to compute the covariance matrix $\boldsymbol{S} = \boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}$
- Numerically more accurate, since the formation of $\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}$ can cause loss of precision.
  - For example, the Läuchli matrix:

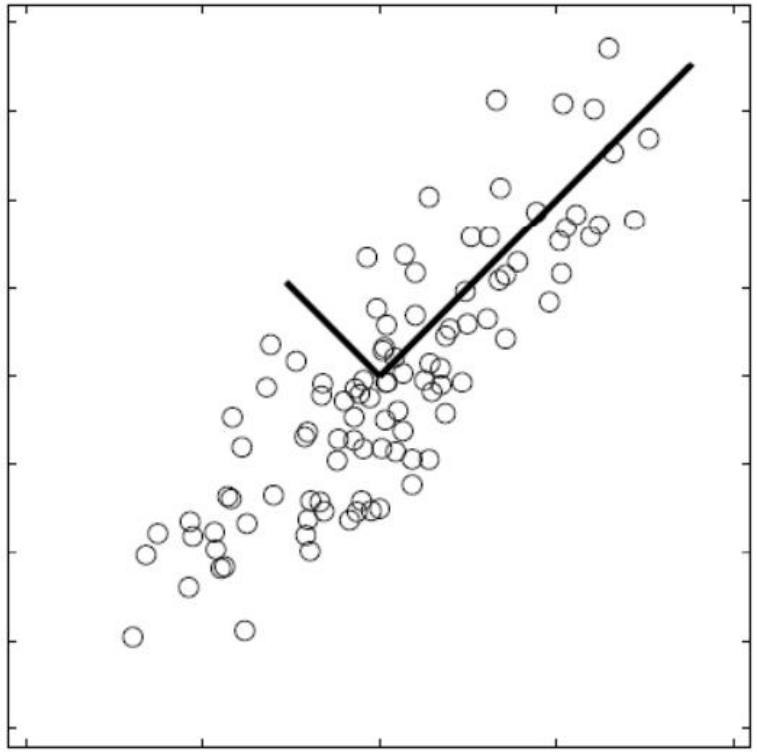
$$\begin{pmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{pmatrix}^{T}$$

where ϵ is a tiny number.

Visualize PCs

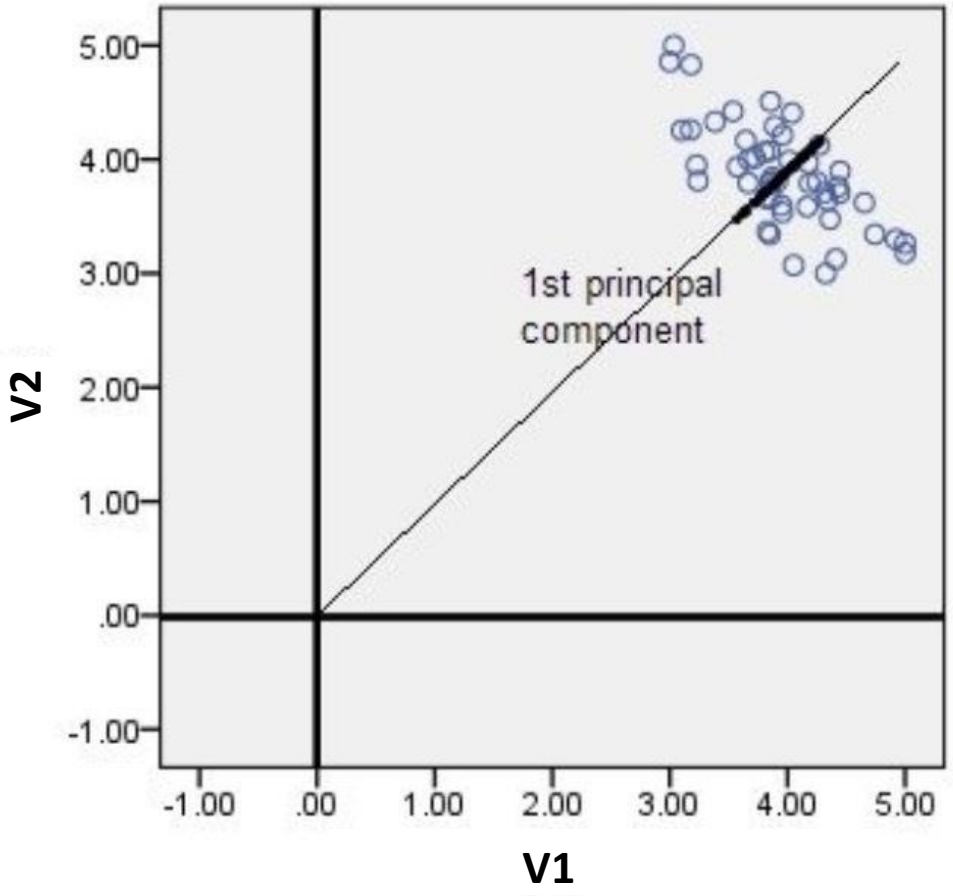
The columns of $\boldsymbol{U}$ are eigenvectors of $\boldsymbol{S} = \boldsymbol{A}\boldsymbol{A}^{\boldsymbol{T}}$.

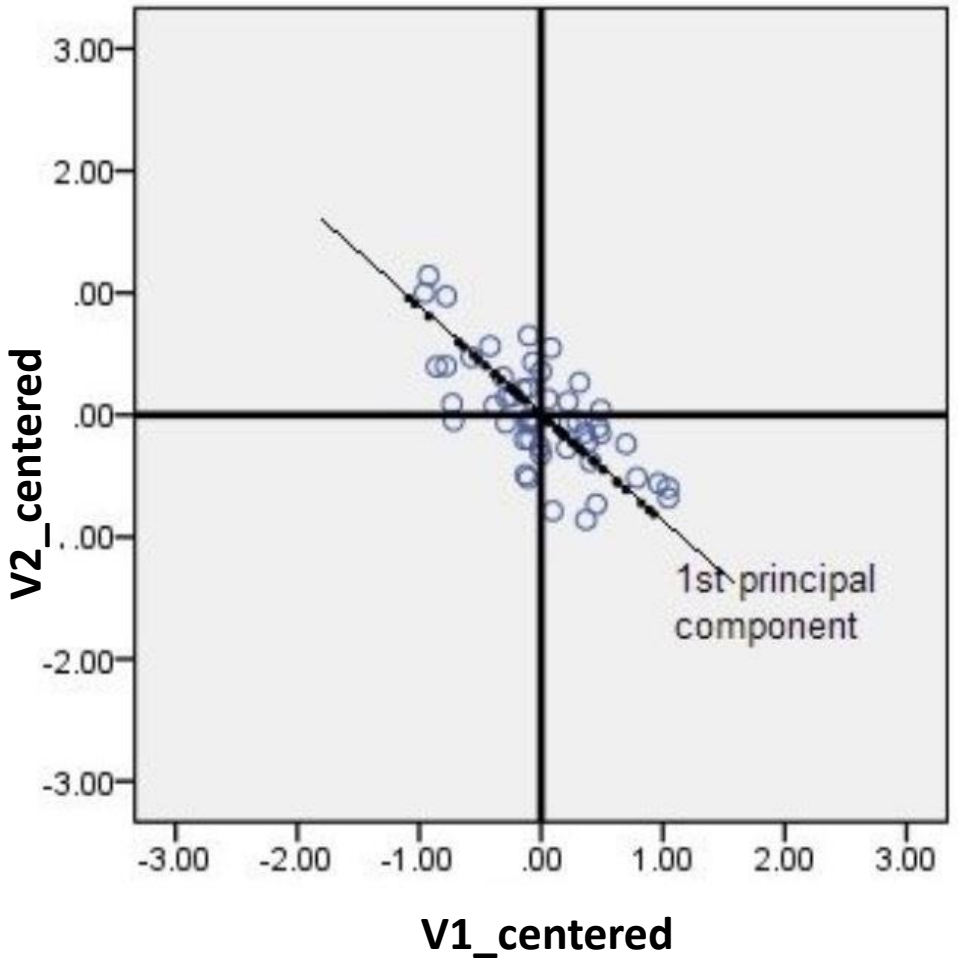
$$\boldsymbol{y} = \boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{x}$$



Data points are represented in a rotated orthogonal coordinate system: the origin is the mean of the data points and the axes are provided by the eigenvectors.

The Necessity of Centralization





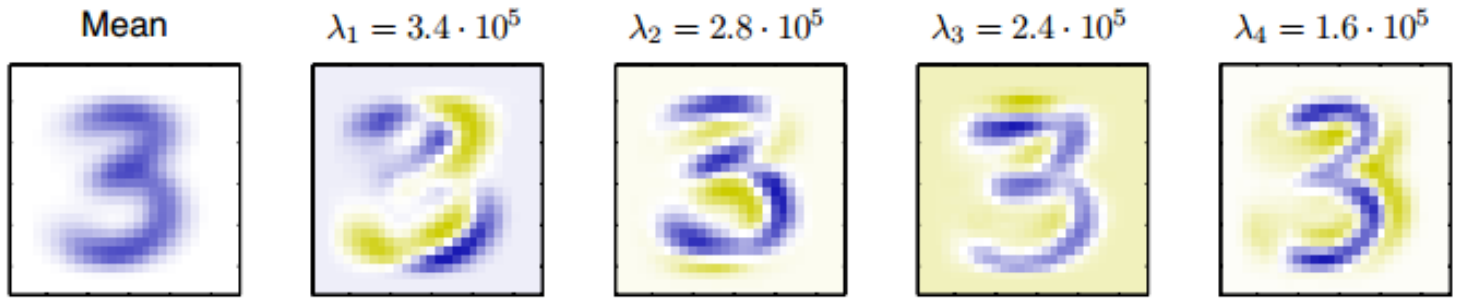
How Many PCs to Keep?

To choose p based on percentage of energy to retain, we can use the following criterion (smallest p):

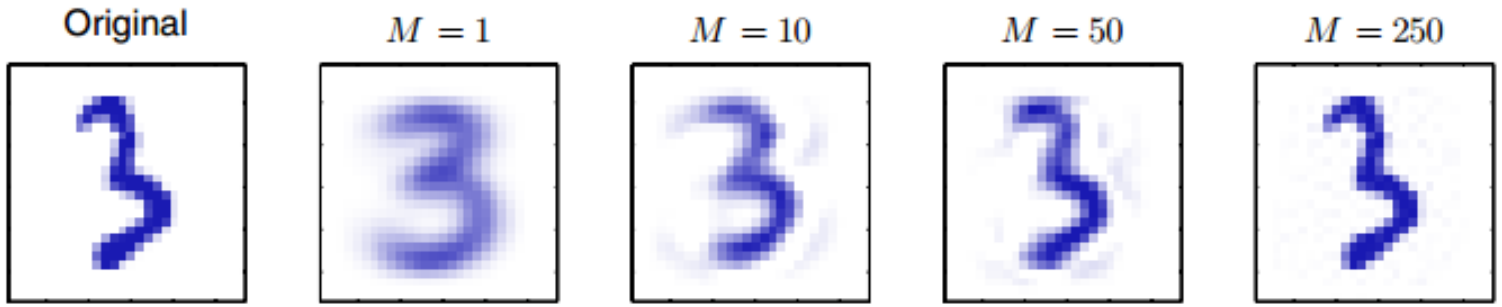
$$\frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq Threshold \quad (e.g., 0.95)$$

PCA-Applications

Data Compression



The mean vector $\overline{\mathbf{x}}$ along with the first four PCA eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_4$ for the off-line digits data set, together with the corresponding eigenvalues.



An original example from the off-line digits data set together with its PCA reconstructions obtained by retaining M principal components for various values of M. As M increases the reconstruction becomes more accurate and would become perfect when $M = D = 28 \times 28 = 784$.

We represent the eigenvectors as images of the same size as the data points.

PCA-Applications

Data Preprocessing

- The goal is **not** dimensionality reduction but rather the transformation of a data set in order to **standardizing** the data.
- Important in allowing subsequent pattern recognition algorithms to be applied successfully to the data set.
- Typically, it is done when the original variables are measured in different order of magnitudes or have significantly different variability.

PCA-Applications

Data Preprocessing

- The goal is **not** dimensionality reduction but rather the transformation of a data set in order to **standardizing** the data.

Traditionally, we can made a linear re-scaling of the individual variables such that each variable had zero mean and unit variance.

$$\frac{x_{ni} - \bar{x}_i}{\sigma_i}$$

However, using PCA we can make a more substantial normalization of the data to give it zero mean and unit covariance, so that variables become decorrelated.

PCA-Applications

Data Preprocessing

- The goal is **not** dimensionality reduction but rather the transformation of a data set in order to **standardizing** the data.

We first write the eigenvalue decomposition $\boldsymbol{SU} = \boldsymbol{U\Lambda}$

Then for each data point, we define $\boldsymbol{y}_n = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{U}^T(\boldsymbol{x}_n - \bar{\boldsymbol{x}})$

Clearly, $\{\boldsymbol{y}_n\}$ have zero mean, and we thus have the covariance matrix

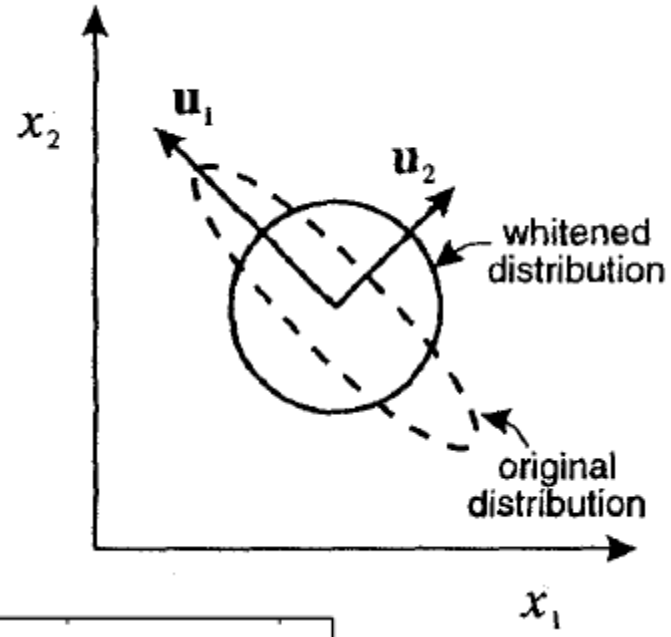
$$\frac{1}{m}\sum_{n=1}^{m}\boldsymbol{y}_n\boldsymbol{y}_n^T = \frac{1}{m}\sum_{n=1}^{m}\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{U}^T(\boldsymbol{x}_n - \bar{\boldsymbol{x}})(\boldsymbol{x}_n - \bar{\boldsymbol{x}})^T\boldsymbol{U}\boldsymbol{\Lambda}^{-\frac{1}{2}} = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{U}^T\boldsymbol{S}\boldsymbol{U}\boldsymbol{\Lambda}^{-\frac{1}{2}} = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{\Lambda}\boldsymbol{\Lambda}^{-\frac{1}{2}} = \boldsymbol{I}$$

This operation is known as *whitening* or *sphereing* the data

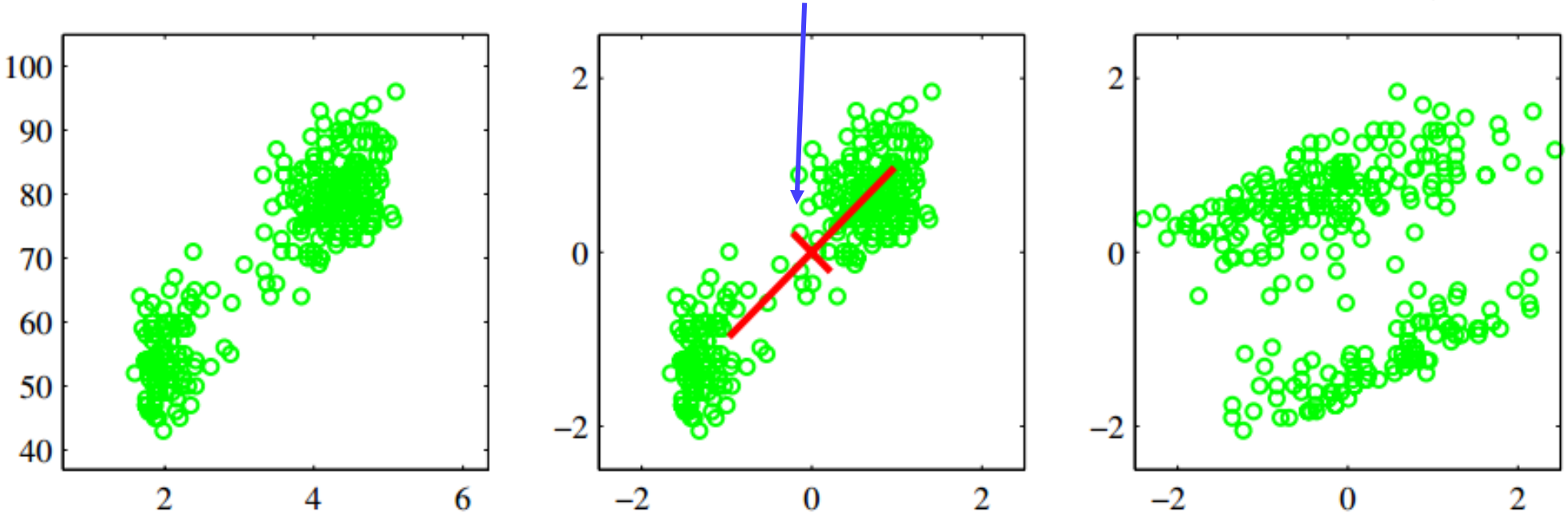
PCA-Applications

Data Preprocessing

- The goal is **not** dimensionality reduction but rather the transforr
  order to **standardizing** the data.



Principal axes of this normalized data set.



Original data

$$\frac{x_{ni} - \bar{x}_i}{\sigma_i}$$

$$\boldsymbol{y}_n = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{U}^T(\boldsymbol{x}_n - \bar{\boldsymbol{x}})$$

PCA-Applications

Data Visualization

- Each data point is projected onto a two-dimensional principal subspace.

PCA and Classification

- Classification with PCA
  - Project both training and testing data into the PCs space
  - For each testing sample, use Nearest Neighbor for classification
  - Issue: accuracy is sensitive to the number of PCs

- PCA may not be always an optimal feature extraction technique for classification.
  - Suppose there are C classes in the training data
  - PCA is based on the sample covariance which characterizes the scatter of the entire data set, irrespective of class-membership.
  - The projection axes chosen by PCA might not provide good discrimination power.

Summary

- Two commonly used definitions of PCA
  - Maximum variance formulation
  - Minimum-error formulation
- Covariance Matrix
  - Symmetric, Positive semi-definite
- Lagrange Multiplier
  - Constrained system → Unconstrained system
  - Critical points
- Data Point Reconstruction
- SVD
  - U, V orthogonal matrix
- PCA Applications