CS 3750 Machine Learning
Lecture 10

Principal Component Analysis (PCA)
Singular Value Decomposition (SVD)

Based on slides from Iyad Batal, Eric Strobl & Milos Hauskrecht

Outline

- **Principal Component Analysis (PCA)**
- **Singular Value Decomposition (SVD)**
- **Multi-Dimensional Scaling (MDS)**
- **Non-linear PCA extension:**
  - **Kernel PCA**

Outline

- **Principal Component Analysis (PCA)**
- **Singular Value Decomposition (SVD)**
- **Multi-Dimensional Scaling (MDS)**
- **Non-linear extensions:**
  - **Kernel PCA**

Real-World Data

Real world data and information therein may be:

- **Redundant**
  - One variables may carry the same information as the other variable
  - Information covered by a set of variable may overlap
- **Noisy**
  - Some dimensions may not carry any useful information and the variation in that dimension is purely due to noise in the observations

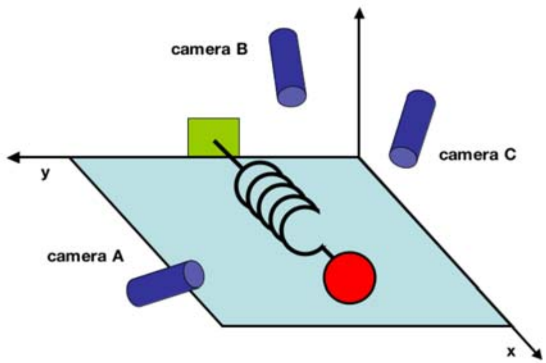
Important questions:

- how to reduce the dimensionality of the data
- what is the intrinsic dimensionality of the data?

Example

Three cameras tracking the movement of a ball on a string in 3D space.

- The ball moves in 2 D space (one dimension is redundant)
- Information collected by 3 cameras overlap.

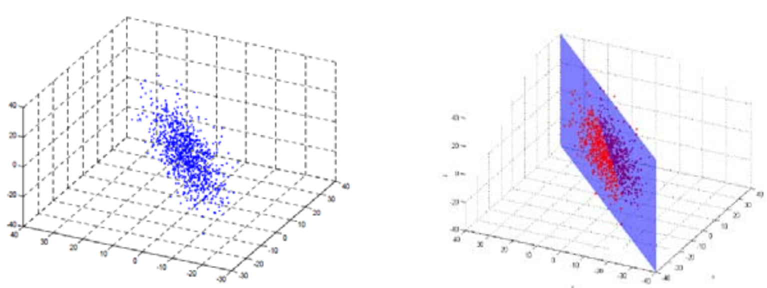


PCA

PCA finds a linear projection of data into orthogonal basis system that has the minimum redundancy and preserves the variance in data.

Applications:

- Identify the intrinsic dimensionality of the data
- Lower dimensional representation of data with the smallest reconstruction error.



PCA/SVD applications

- Dimensionality reduction
- LSI: Latent Semantic Indexing.
- Kleinberg/Hits algorithm
- Google/PageRank algorithm (random walk with restart).
- Image-compression (eigen faces)
- Data visualization (by projecting the data on 2D).

Background: eigenvectors

- If A is a square matrix, a non-zero vector **v** is an eigenvector of A if there is a scalar λ (eigenvalue) such that

$$Av = \lambda v$$

- Example: $\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4\begin{pmatrix} 3 \\ 2 \end{pmatrix}$
- If we think of the squared matrix as a transformation matrix, then multiply it with the eigenvector do not change its direction.

What are the eigenvectors of the identity matrix?

The Covariance Matrix of X

$$C_X = \frac{1}{n-1} X^T X$$

Diagonal terms: variance

Large values = signal

Off-diagonal: covariance

Large values = high redundancy

Covariance matrix is always symmetric

$$C_X{}^T = \frac{1}{n-1}(X^T X)^T = \frac{1}{n-1}(X^T X) = C_X$$

Matrix decomposition

Theorem 1: if square $d \times d$ matrix $\mathbf{S}$ is a real and symmetric matrix ($\mathbf{S} = \mathbf{S}^{\mathrm{T}}$) then

$$\mathbf{S} = \mathbf{V\Lambda V}^{\mathrm{T}}$$

where $\mathbf{V} = [v_1 \quad \cdots \quad v_d]$ are the eigenvectors of $\mathbf{S}$ and $\mathbf{\Lambda} = diag(\lambda_1, \dots, \lambda_d)$ are the corresponding eigenvalues.

Proof:

$$\mathbf{SV} = V\Lambda$$

$$[\mathbf{S}v_1, \mathbf{S}v_2, \ldots \mathbf{S}v_d] = [\lambda_1 v_1, \lambda_2 v_2, \ldots \lambda_d v_d]$$

$$\mathbf{S}VV^{-1} = V\Lambda V^{-1}$$

$$\mathbf{S} = V\Lambda V^T$$

Covariance matrix decomposition

$$\boldsymbol{X^T X = V \Lambda V^T}$$

where:

- $\boldsymbol{V}$ is a matrix of eigenvectors of $\boldsymbol{X^T X}$ (arranged in columns);
- Λ is a diagonal matrix of corresponding eigenvalues

Proof:

$(\boldsymbol{X^T X})\boldsymbol{V} = \boldsymbol{V}\Lambda\boldsymbol{D}$

$(\boldsymbol{X^T X})\boldsymbol{V V^T} = \boldsymbol{V}\Lambda\boldsymbol{V^T}$

$\boldsymbol{X^T X} = \boldsymbol{V}\Lambda\boldsymbol{V^T}$ since eigenvectors are orthonormal

Change of Basis

Assume:

- **X** is an *n* x *d* data matrix
- **Linear (affine) transformation:** $\boldsymbol{A}$

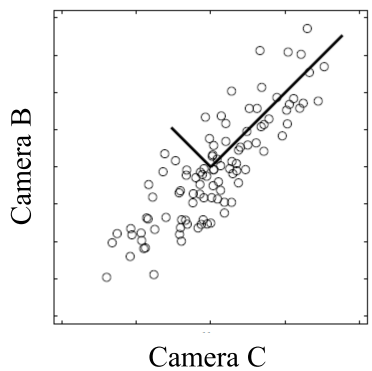
$$\boldsymbol{Y = XA}$$

where

- $\boldsymbol{A}$ is a matrix that transforms $\boldsymbol{X}$ into $\boldsymbol{Y}$
- *Columns* of $\boldsymbol{A}$ are formed by basis vectors that re-express the rows of $\boldsymbol{X}$ in the new coordinate system

Change of Basis

- But, what is the best "basis" vector?
  - **PCA assumption:** the direction with the largest variance



The basis is just the best fit line

Goal and Assumptions of PCA

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{A}$$

Goal: Find the best transformation $\boldsymbol{A}$, so that $\boldsymbol{Y}$ has the minimal noise and redundancy

Assumptions

1) $\boldsymbol{A}$ contains orthonormal basis vectors (makes computations easier)
2) Covariance matrix captures all the information about $\boldsymbol{X}$ (only true for exponential family distributions)

PCA Derivation

- C_Y : Covariance of Y expressed in terms of A

$$C_Y = \frac{1}{n-1} Y^T Y$$
$$= \frac{1}{n-1} (XA)^T (XA)$$
$$= \frac{1}{n-1} A^T X^T X A$$
$$= \frac{1}{n-1} A^T (X^T X) A$$

PCA

- Find the direction for which the variance is maximized:

$$v_1 = argmax_{v1}\ var(Xv_1)$$
Subject to: $v_1{}^T v_1$=1

- Rewrite in terms of the covariance matrix:

$$var(Xv_1) = \frac{1}{N-1}(Xv_1)^T(Xv_1) = v_1{}^T \frac{1}{N-1} X^T X\, v_1 = v_1{}^T C\, v_1$$

- Solve via constrained optimization:

$$L(v_1, \lambda_1) = v_1{}^T C\, v_1 + \lambda_1 (1 - v_1{}^T v_1)$$

PCA

- Constrained optimization:

$$L(v_1, \lambda_1) = v_1^T C\, v_1 + \lambda_1(1 - v_1^T v_1)$$

- Gradient with respect to v_1:

$$\frac{dL(v_1, \lambda_1)}{dv_1} = 2Cv_1 - 2\lambda_1 v_1 \Rightarrow Cv_1 = \lambda_1 v_1$$

This is the eigenvector problem!

- Multiply by v_1^T:

$$\lambda_1 = v_1^T C\, v_1$$

The projection variance is the eigenvalue

PCA Derivation

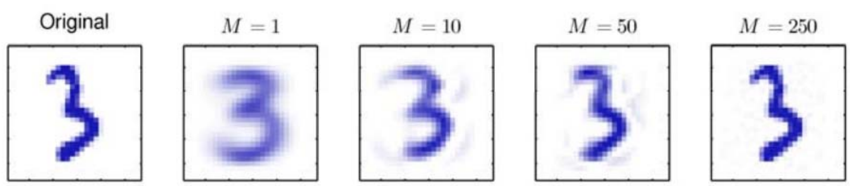
- Assuming $\boldsymbol{A} = \boldsymbol{V}$, i.e. each column is an eigenvector of $\boldsymbol{X^T X}$

$$\begin{aligned} \boldsymbol{C_Y} &= \frac{1}{n-1}\boldsymbol{V^T}(\boldsymbol{X^T X})\boldsymbol{V} \\ &= \frac{1}{n-1}\boldsymbol{V^T}\left(\boldsymbol{V D V^T}\right)\boldsymbol{V} \\ &= \frac{1}{n-1}\boldsymbol{V^T V D V^T V} \\ &= \frac{1}{n-1}\boldsymbol{V^{-1} V D V^{-1} V} \\ &= \frac{1}{n-1}\boldsymbol{D} \end{aligned}$$

After the transformation of $\boldsymbol{X}$ with $\boldsymbol{V}$, the covariance matrix becomes diagonal

PCA as dimensionality reduction

(1) If the data lives in a lower dimensional space d', then some of the eigenvalues in $\boldsymbol{D}$ matrix are set to 0
(2) If we want to reduce the dimensionality of the data from d to some fixed k, we choose the eigenvectors with the k highest eigenvalues – the dimensions that preserve most of the variance in the data
(3) This selection also minimizes the data reconstruction error (so the best k dimensions lead to best error).



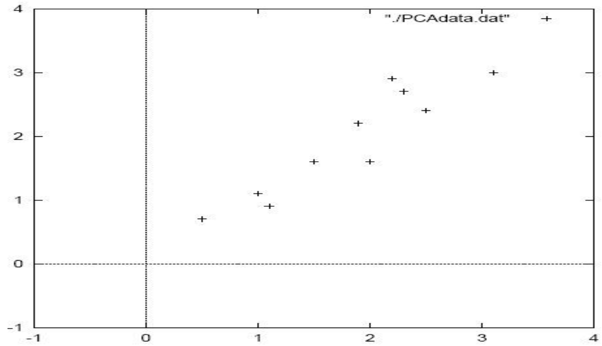
PCA for dimensionality reduction

PCA steps: transform an $N \times d$ matrix X into an $N \times m$ matrix Y:

- **Centralized the data (subtract the mean).**
- **Calculate the $d \times d$ covariance matrix: $C = \frac{1}{N-1} X^T X$ *(different notation from tutorial!!!)***
  - $C_{i,j} = \frac{1}{N-1} \sum_{q=1}^{N} X_{q,i} \cdot X_{q,j}$
  - **$C_{i,i}$ (diagonal) is the variance of variable i.**
  - **$C_{i,j}$ (off-diagonal) is the covariance between variables i and j.**
- **Calculate the eigenvectors of the covariance matrix (orthonormal).**
- **Select m eigenvectors that correspond to the largest m eigenvalues to be the new basis.**

PCA: example

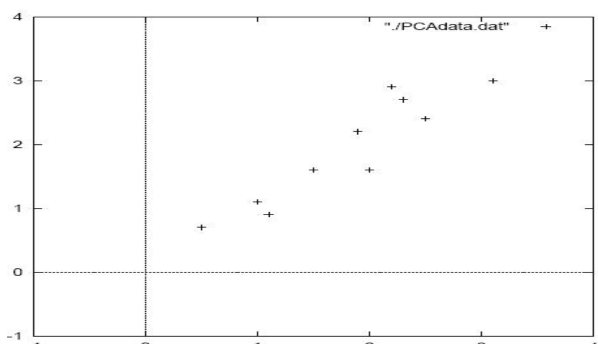
X **: the data matrix with *N=11* objects and *d=2* dimensions.**



PCA: example

➤ ***Step 1: subtract the mean and calculate the covariance matrix C.***

$$C = \begin{pmatrix} 0.716 & 0.615 \\ 0.615 & 0.616 \end{pmatrix}$$



PCA: eexample

- *Step 2: Calculate the eigenvectors and eigenvalues of the covariance matrix:*

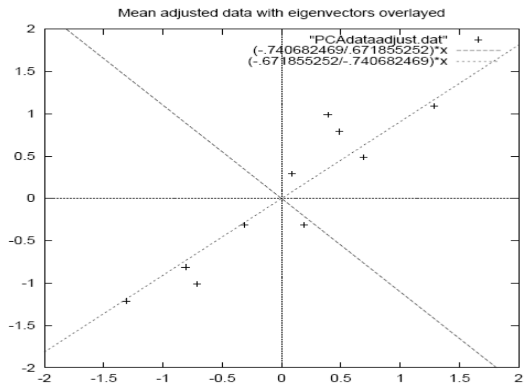
  $\lambda_1 \approx 1.28$, $v_1 \approx [-0.677\ \ -0.735]^T$, $\lambda_2 \approx 0.49$, $v_2 \approx [-0.735\ \ 0.677]^T$

Notice that v_1 and v_2 are orthonormal:

$|v_1|=1$

$|v_2|=1$

$v_1 \cdot v_2 = 0$



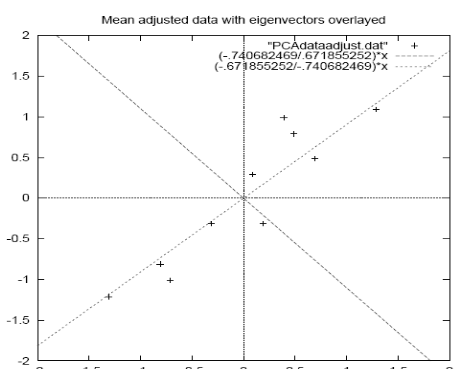
PCA: example

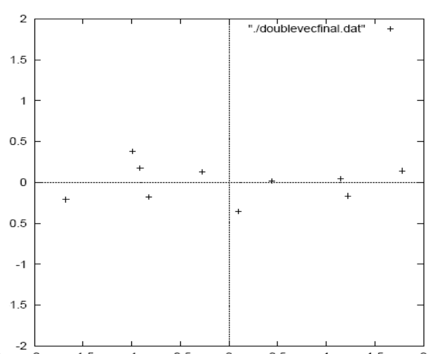
- ***Step 3: project the data***

  Let $V = [v_1, \dots v_m]$ is $d \times m$ matrix where the columns v_i are the eigenvectors corresponding to the largest m eigenvalues

  The projected data: $Y = X V$ is $N \times m$ matrix.

  If m=d (more precisely rank(X)), then there is no loss of information!





PCA: example

➢ *Step 3: project the data*

$\lambda_1 \approx 1.28$, $v_1 \approx [-0.677 \;\; -0.735]^T$, $\lambda_2 \approx 0.49$, $v_2 \approx [-0.735 \;\; 0.677]^T$

The eigenvector with the highest eigenvalue is the **principle component** of the data.

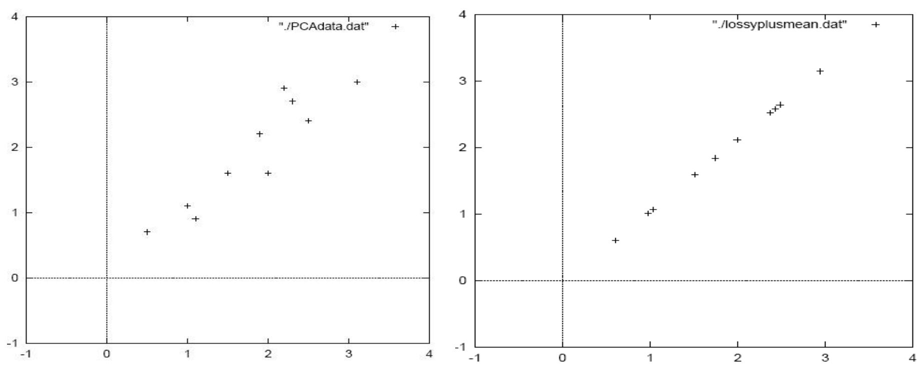
if we are allowed to pick only one dimension, the principle component is the best direction (retain the maximum variance).

Our PC is $v_1 \approx [-0.677 \;\; -0.735]^T$

PCA: example

➢ *Step 3: project the data*

If we select the first PC and reconstruct the data, this is what we get:



We lost variance along the other component (lossy compression!)

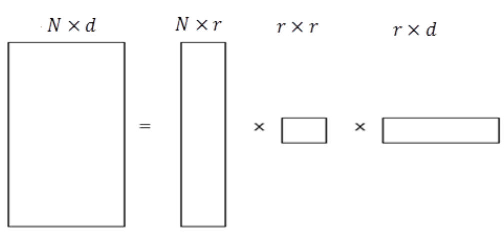
Outline

- Principal Component Analysis (PCA)
- Singular Value Decomposition (SVD)
- Multi-Dimensional Scaling (MDS)
- Non-linear extensions:
  - Kernel PCA

SVD

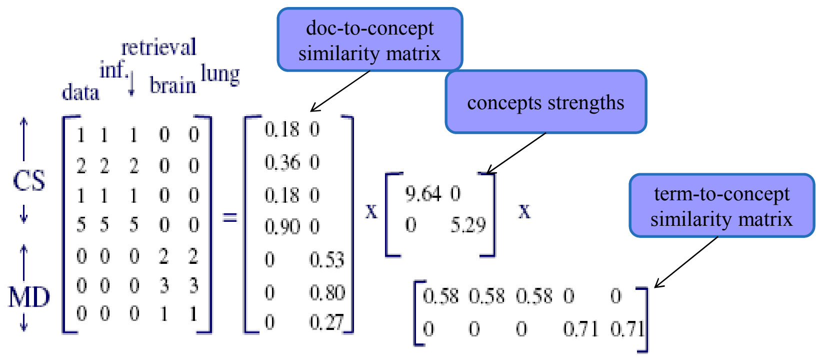
Any $N \times d$ matrix X can be uniquely expressed as:

$$\mathbf{X} = \mathbf{U} \times \mathbf{\Sigma} \times \mathbf{V}^T$$



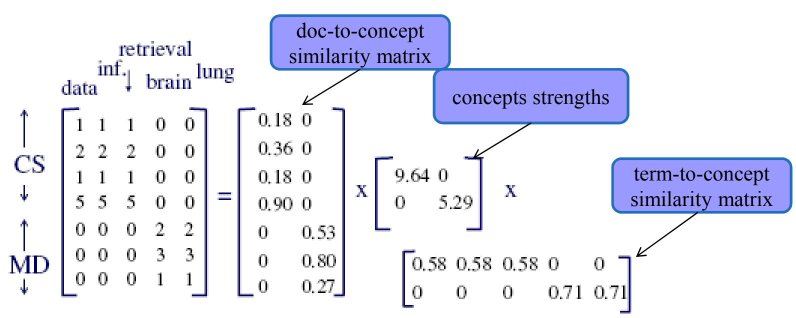
- r is the rank of the matrix X (# of linearly independent columns/rows).
- U is a column-orthonormal $N \times r$ matrix.
- Σ is a diagonal $r \times r$ matrix where the singular values σ_i are sorted in descending order.
- V is a column-orthonormal $d \times r$ matrix.

SVD example



The rank of this matrix r=2 because we have 2 types of documents (CS and Medical documents), i.e. 2 concepts.

SVD example



U: document-to-concept similarity matrix

V: term-to-concept similarity matrix.

Example: $U_{1,1}$ is the weight of CS concept in document d_1, σ_1 is the strength of the CS concept, $V_{1,1}$ is the weight of ‘data’ in the CS concept.

$V_{1,2}$=0 means ‘data’ has zero similarity with the 2nd concept (Medical).

What does $U_{4,1}$ means?

PCA and SVD relation

Theorem: Let $X = U \Sigma V^T$ be the SVD of an $N \times d$ matrix X and $C = \frac{1}{N-1} X^T X$ be the $d \times d$ covariance matrix. The eigenvectors of C are the same as the right singular vectors of X.

Proof:

$X^T X = V \Sigma U^T U \Sigma V^T = V \Sigma \Sigma V^T = V \Sigma^2 V^T$

$C = V \frac{\Sigma^2}{N-1} V^T$

But C is symmetric, hence $C = V \Lambda V^T$ (according to theorem1).

Therefore, the eigenvectors of the covariance matrix are the same as matrix V (right singular vectors) and the eigenvalues of C can be computed from the singular values $\lambda_i = \frac{\sigma_i^2}{N-1}$

Summary for PCA and SVD

Objective: project an $N \times d$ data matrix X using the largest m principal components $V = [v_1, \dots v_m]$.

1. zero mean the columns of X.
2. Apply PCA or SVD to find the principle components of X.

PCA:

I. Calculate the covariance matrix $C = \frac{1}{N-1} X^T X$.

II. V corresponds to the eigenvectors of C.

SVD:

I. Calculate the SVD of $X = U \Sigma V^T$.

II. V corresponds to the right singular vectors.

3. Project the data in an m dimensional space: $Y = X V$

Outline

- Principal Component Analysis (PCA)
- Singular Value Decomposition (SVD)
- **Multi-Dimensional Scaling (MDS)**
- Non-linear extensions:
  - Kernel PCA

MDS

- Multi-Dimensional Scaling [Cox and Cox, 1994] .
- MDS give points in a low dimensional space such that the Euclidean distances between them best approximate the original distance matrix.

  Given distance matrix

$$\Delta := \begin{pmatrix} \delta_{1,1} & \delta_{1,2} & \cdots & \delta_{1,I} \\ \delta_{2,1} & \delta_{2,2} & \cdots & \delta_{2,I} \\ \vdots & \vdots & & \vdots \\ \delta_{I,1} & \delta_{I,2} & \cdots & \delta_{I,I} \end{pmatrix}.$$

  Map input points x_i to z_i such as $||z_i - z_i|| \approx \delta_{i,j}$
- Classical MDS: the norm $\| . \|$ is the Euclidean distance.
- Distances → inner products (Gram matrix) → embedding

  There is a formula to obtain Gram matrix G from distance matrix Δ.

MDS example

Given pairwise distances between different cities (Δ matrix), plot the cities on a 2D plane (recover location)!!

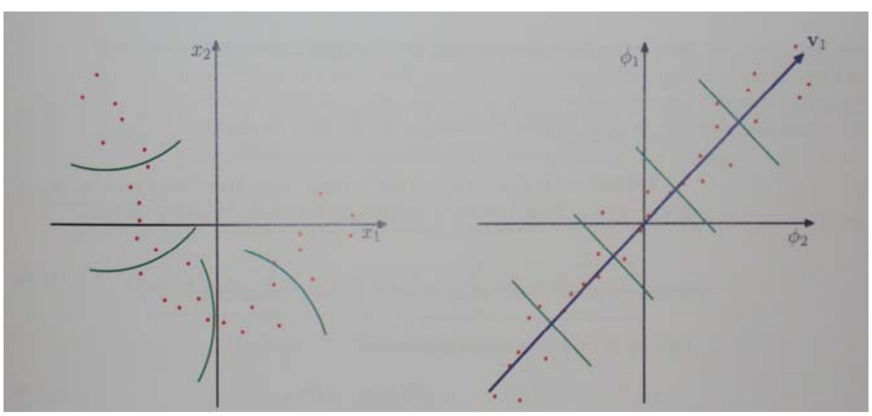
PCA and MDS relation

- Preserve Euclidean distances = retaining the maximum variance.
- *Classical MDS is equivalent to PCA when the distances in the input space are the Euclidean distance.*
- PCA uses the $d \times d$ covariance matrix: $C = \frac{1}{N-1} X^T X$
- MDS uses the $N \times N$ Gram (inner product) matrix: $G = X X^T$
- If we have only a distance matrix (we don't know the points in the original space), we cannot perform PCA!
- Both PCA and MDS are invariant to space rotation!

Kernel PCA

- Kernel PCA [Scholkopf et al. 1998] performs nonlinear projection.
- Given input $(x_1, \dots x_N)$, kernel PCA computes the principal components in the feature space $(\varphi(x_1), \dots \varphi(x_N))$.
- Avoid explicitly constructing the covariance matrix in feature space.
- The kernel trick: formulate the problem in terms of the kernel function $k(x, x') = \varphi(x).\varphi(x')$ without explicitly doing the mapping.
- Kernel PCA is non-linear version of MDS use Gram matrix in the feature space (a.k.a Kernel matrix) instead of Gram matrix in the input space.

Kernel PCA



Original space | A non-linear feature space