Kernel Principal Component Analysis

Sho Inaba

Outline

- Introduction
  - From Linear to Non - Linear
- Kernel Methods
  - Inner Product in Higher Dimension
- Kernel PCA
  - Constructing the Kernel Principal Components
- Other Applications
  - Where to Go

Outline

- **Introduction**
  - From Linear to Non - Linear
- Kernel Methods
  - Inner Product in Higher Dimension
- Kernel PCA
  - Constructing the Kernel Principal Components
- Other Applications
  - Where to Go

Introduction

- **Data Analysis**

A process of breaking down the given information into understandable forms:

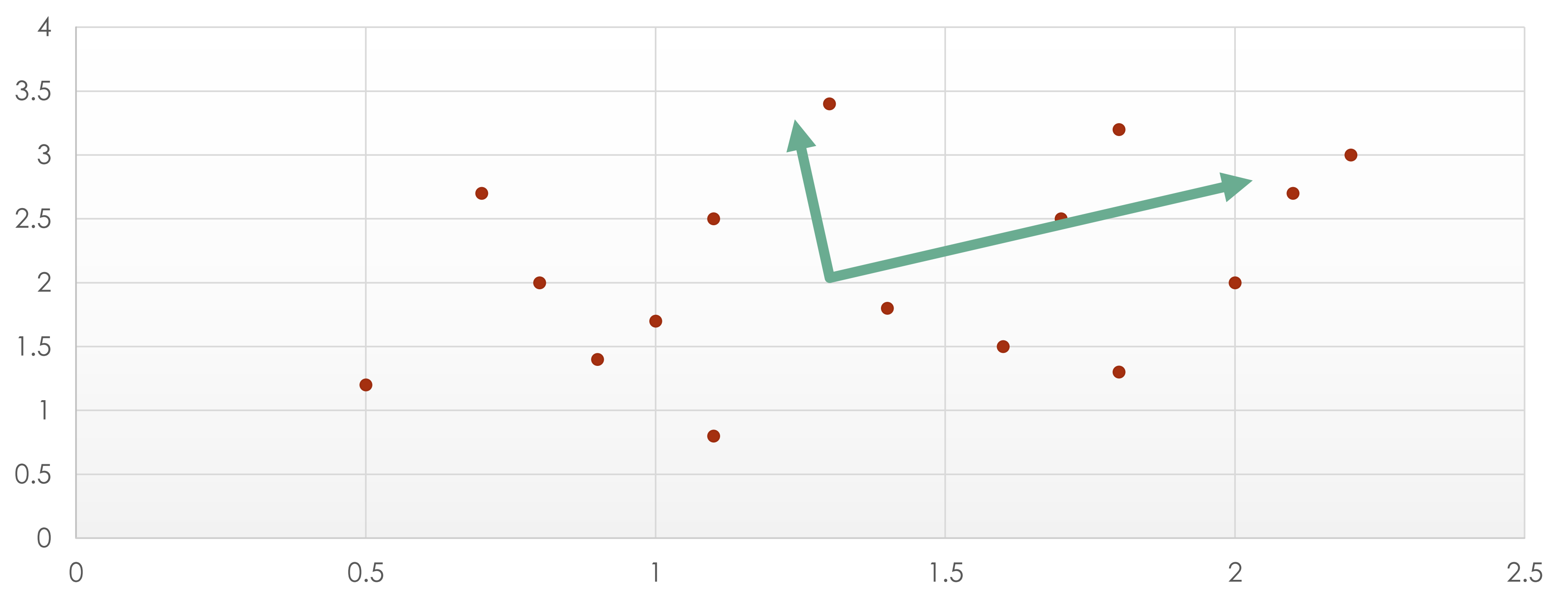
Inspecting, Cleansing, Transforming, Modeling,   and so on…

Inspecting
- Correlation Analysis
- Data Independent Acquisition

Cleansing
- Dimension Reduction
- Noise Reduction

Transforming
- Principal Component Analysis
- Linear Discriminant Analysis

PCA (Principal Component Analysis)



- A technique to find a linear subspace where the new features have the **largest variance**.

Given a dataset $\{x_i \in \mathbb{R}^D\}$ for $i = 1,2,\dots,N$.

Suppose the projection is denoted as $y = Ax$, where an orthonormal $A = (u_1 \quad \cdots \quad u_M)^T$ with $M \le D$ and $u_i \in \mathbb{R}^M$.

Want: $A_{opt} = \arg\max_A tr(S_y)$,

where S_y is the covariance matrix of $\{y_i\}$.

Since A is orthonormal, $tr(S_y) = tr(AS_xA^T) = tr(S_x)$.

Then, we can obtain $S_x v_k = \lambda_k v_k$ can maximize the argument, with $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_D$.

Now, we can approximate x_i as

$$\tilde{x}_i = \sum_{k=1}^{M} (x_i^T u_k) u_k$$

$x_i^T u_k$: k-th principal component, u_k : k-th principal direction.

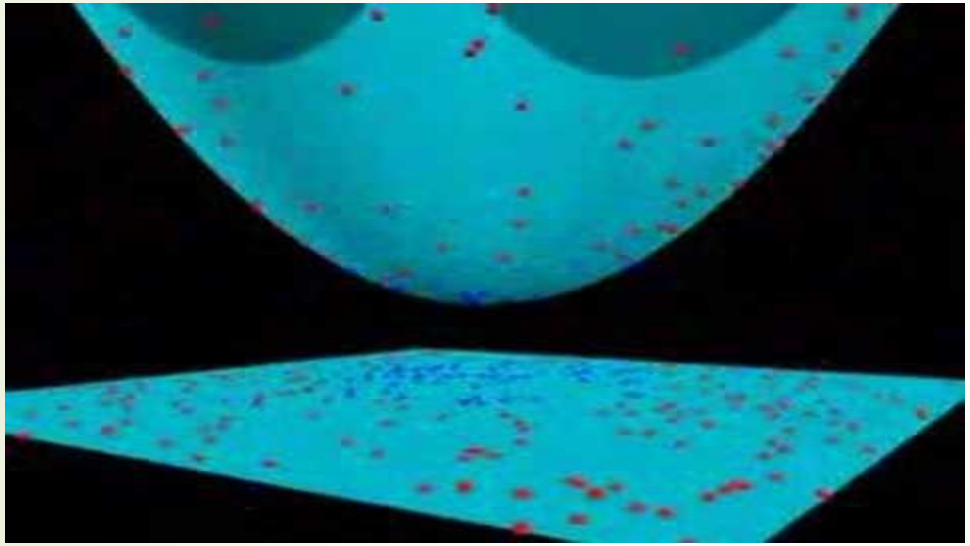
From Linear to Non - Linear

- Limitation of linear methods
  - Existence of Linearly Inseparable Data
  - Every features must be numerical
- Transformation to Higher Dimension
  - Combinatorial Explosion
- Kernel Methods
  - Reduction of computational cost

Outline

- Introduction
  - From Linear to Non - Linear
- **Kernel Methods**
  - Inner Product in Higher Dimension
- Kernel PCA
  - Constructing the Kernel Principal Components
- Other Applications
  - Where to Go

Basic Idea of Kernel Methods



- If a dataset is **not linearly separable**, then let's find a higher dimensional space with **proper** basis.
- Problem

How to find

A link to the video: https://www.youtube.com/watch?v=3liCbRZPrZA

Mapping into Higher Dimension

A non-linear transformation $\phi: \mathbb{R}^D \rightarrow \mathbb{R}^M$ with (usually) $\mathrm{D} \ll M$.

- Unfortunately, there is no particular way to find a proper space.
  - Too many possibility
  - Too much computational cost

- Example: From (x_1, x_2) to a quadratic space $(1, x_1, x_2, x_1^2, x_2^2, x_1 x_2)$

  There are 6 basis in transferred space.
- Example: MNIST dataset to a cubic space.

  There are $\frac{787!}{784!3!} = 80931145$ basis.

Inner Product in a Higher Dimension

- Consider a transformation $K: \mathbb{R}^D \times \mathbb{R}^D (\rightarrow \mathbb{R}^M \times \mathbb{R}^M) \rightarrow \mathbb{R}$ such that
$$K(x, y) = \phi(x)^T \phi(y)$$

The transformation K is called a **kernel** function.

- This inner product space has a lot of nice property.
  - Mercer's theorem

    A kernel matrix $K_{ij} = K(x_i, x_j)$ is positive semidefinite.
  - Representor theorem

    If a kernel matrix K is positive semidefinite, then $\mathrm{w}^{\mathrm{T}} \phi(x)$ can be expressed as a linear combination of $K(x, x_i)$.

Kernel Trick

Original Space
- Linearly inseparable

→

Higher Dimensional Space
- Linearly Separable
- Computationally expensive

→

Kernel Space (Inner Product, Dual Space)
- Constant dimension (The number of samples)

We can solve the optimization problem without knowing an actual ϕ by just using $K(x, x_i)$.

Outline

- Introduction
  - From Linear to Non - Linear
- Kernel Methods
  - Inner Product in Higher Dimension
- **Kernel PCA**
  - Constructing the Kernel Principal Components
- Other Applications
  - Where to Go

PCA to Kernel PCA

In Regular PCA, we evaluate a covariance matrix of the original data.

$$S_x = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T$$

Let ϕ_n be the normalized non-linear transformation from $\mathbb{R}^D$ to $\mathbb{R}^M$. Then, a covariance matrix C of the transferred data can be expressed as

$$C = \frac{1}{N}\sum_{i=1}^{N}\phi_n(x_i)\phi_n(x_i)^T$$

With its eigenvalues and eigenvectors are given by

$$Cv_k = \lambda_k v_k$$

For $k = 1,2,\ldots,M$.

Introducing a Kernel function

$$Cv_k = \lambda_k v_k \Leftrightarrow \frac{1}{N}\sum_{i=1}^{N}\phi_n(x_i)\phi_n(x_i)^T v_k = \lambda_k v_k$$

And eigenvectors of C must be a linear combination of $\phi(x_i)$,

$$v_k = \sum_{i=1}^{N} a_{ki}\phi_n(x_i)$$

Then, we have

$$\frac{1}{N}\sum_{i=1}^{N}\phi_n(x_i)\phi_n(x_i)^T \sum_{j=1}^{N} a_{kj}\phi_n(x_j) = \lambda_k \sum_{i=1}^{N} a_{ki}\phi_n(x_i)$$

By multiplying both sides by $\phi_n(x_l)^T$,

$$\frac{1}{N}\sum_{i=1}^{N}K(x_l, x_i) \sum_{j=1}^{N} a_{kj}K(x_i, x_j) = \lambda_k \sum_{i=1}^{N} a_{ki}K(x_l, x_i)$$

The kernel principal components

$$\frac{1}{N}\sum_{i=1}^{N} K(x_l, x_i) \sum_{j=1}^{N} a_{kj} K(x_i, x_j) = \lambda_k \sum_{i=1}^{N} a_{ki} K(x_l, x_i) \Leftrightarrow K^2 a_k = \lambda_k N K a_k$$

Where $K_{ij} = K(x_i, x_j)$ and $a_k = [a_{k1} \quad \cdots \quad a_{kN}]^T$.

Here, we can solve for a_k by

$$K a_k = \lambda_k N a_k$$

Then, the kernel principle components are

$$y_k = \phi_n(x)^T v_k = \sum_{i=1}^{N} a_{ki} K(x, x_i)$$

By representor theorem.

- If ϕ were not normalized, replace K with $\widetilde{K} = K - 1_N K - K 1_N + 1_N K 1_N$ where 1_N is the $N \times N$ matrix with all elements equal to $\frac{1}{N}$.

Choice of Kernel

- Two major kernels
  - Polynomial Kernel

    $K(x, y) = (x^T y + c)^d$ where $c \geq 0$
  - Gaussian Kernel (Radial Basis Kernel)

    $K(x, y) = \exp(-\|x - y\|^2 / 2\sigma^2)$

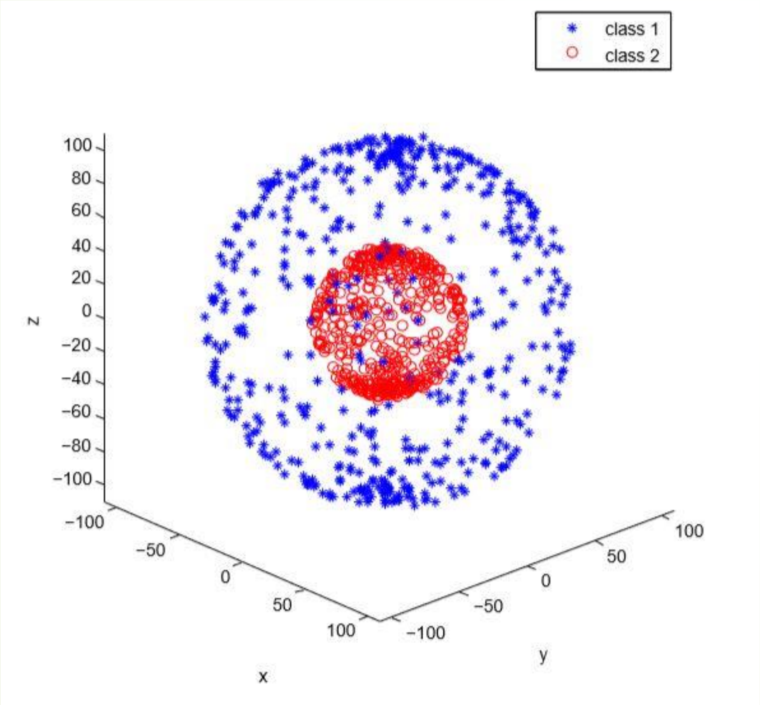
- Other well used kernels
  - Laplace Kernel

    $K(x, y) = \exp(-\alpha \sum_{i=1}^{D} |x_i - y_i|)$
  - Exponential Kernel

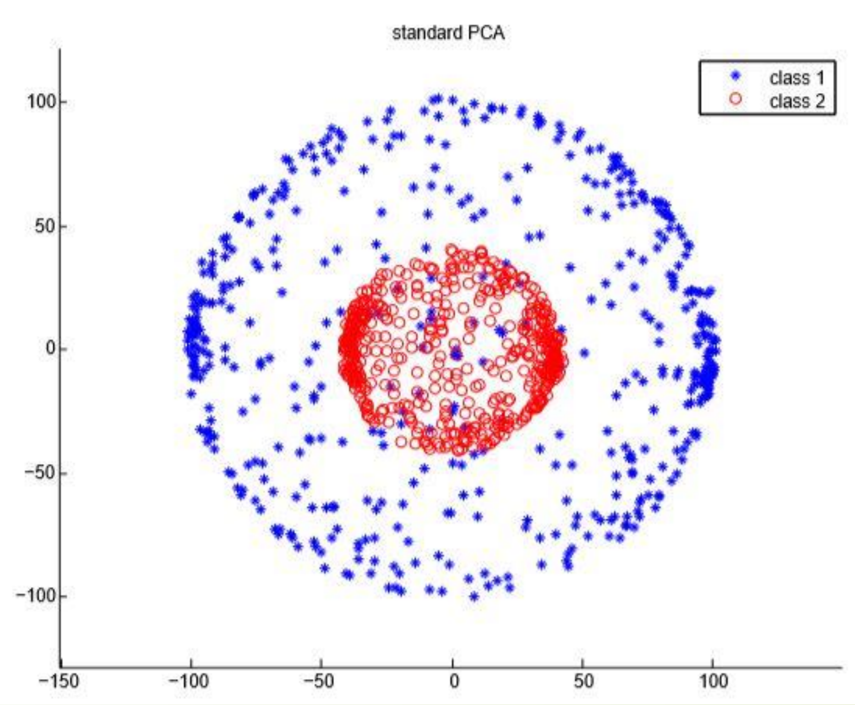
    $K(x, y) = \exp(\beta x^T y)$

Result

Two spherical data

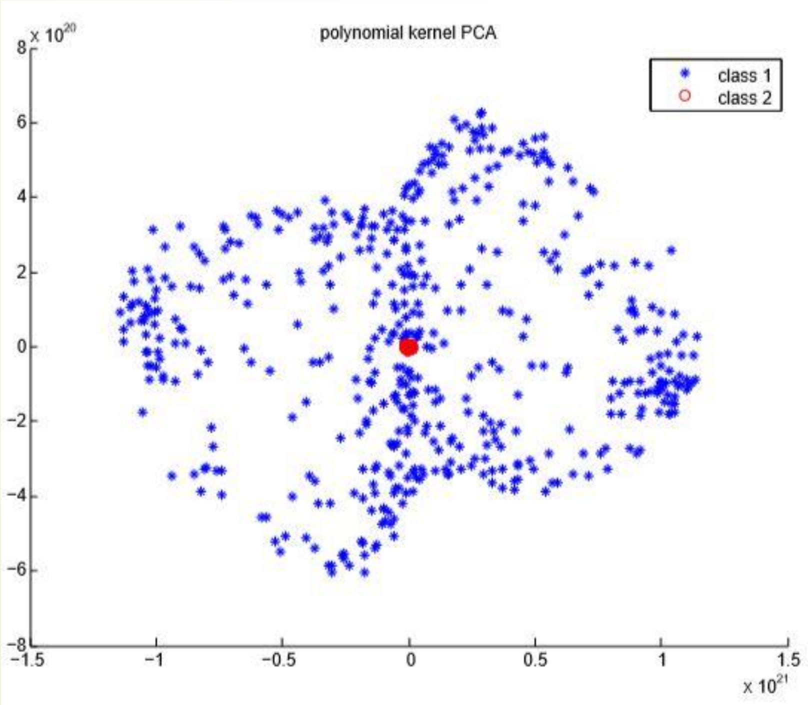


Standard PCA

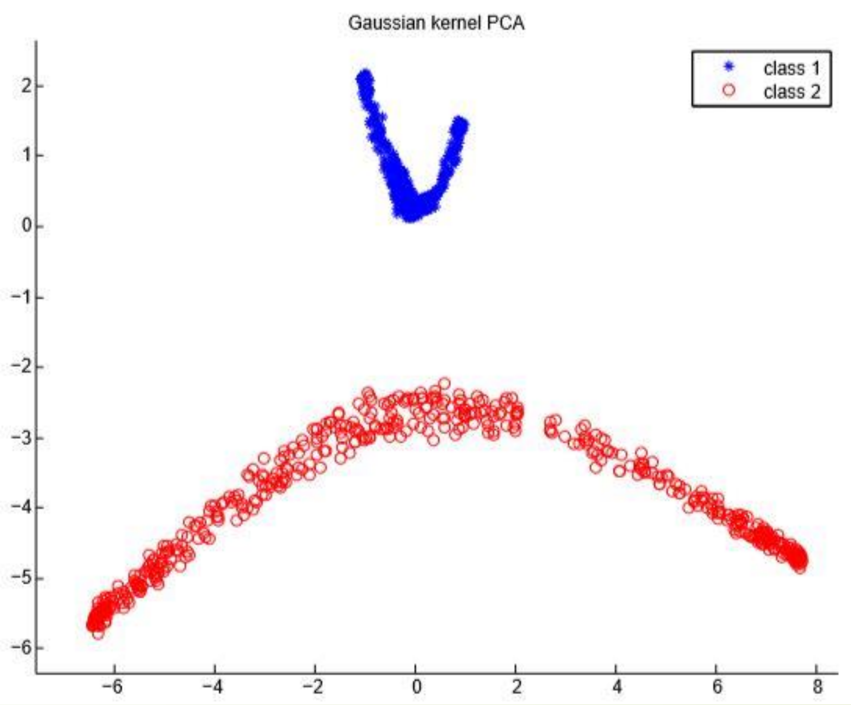


Result of Kernel PCA

Polynomial with Degree 5



Gaussian with $\sigma = 27.8$



Outline

- Introduction
  - From Linear to Non - Linear
- Kernel Methods
  - Inner Product in Higher Dimension
- Kernel PCA
  - Constructing the Kernel Principal Components
- **Other Applications**
  - Where to Go

Other Applications

- Basic kernelizing trick can be applied on linear optimization algorithm.
  - Linear Regression
- We can kernelize any covariance matrix based learning algorithm.
  - LDA (Linear Discriminant Analysis)
  - CCA (Canonical Correspondence Analysis)
- Covariance matrix is not accurate with sparse dataset
  - Sparse Covariance Estimation

Reference

- Wang Quan, Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models, arXiv:1207.3538, 07/2012, web.

Thank you.