PCA and kernel PCA

Kaiwen Chang
01/03/2017

Outline

I. PCA

1. Definitions
2. Solution
3. Example

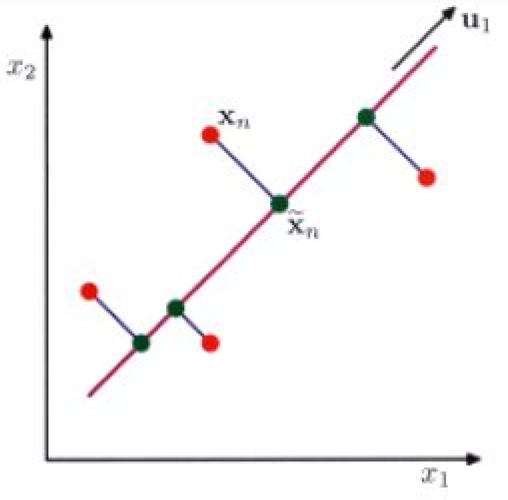
II. Kernel PCA

1. Why kernel
2. Solution
3. Example

I. PCA

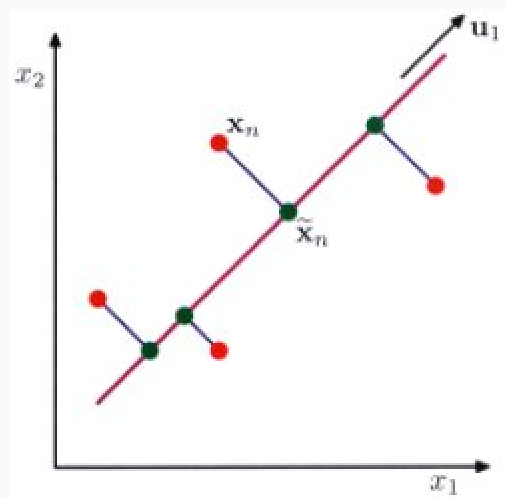
1. Definitions

1. The orthogonal projection of the data onto a lower dimensional linear space (principal subspace), such that the **variance** of the projected data is maximized



Magenta line: principal subspace

2. The linear projection that minimizes the average projection cost (the mean squared distance between the data points and their projections)



Blue line: projection error

2. Solution

Data set $\{\mathbf{x}_n\}$ n = 1,2,...,N

$\mathbf{x}_n$: column vector with dimension D

Aim: project data onto a space with dimension M < D, while maximizing the variance of projected points

Projection of $\mathbf{x}_n$: $\mathbf{u}_1^{\mathrm{T}}\mathbf{x}_n$

($\mathbf{u}_1$: the direction of space, dimension D)

Mean of projected data: $\mathbf{u}_1^{\mathrm{T}}\overline{\mathbf{x}}$ $\overline{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$

Variance: $$\frac{1}{N}\sum_{n=1}^{N}\left\{\mathbf{u}_1^{\mathrm{T}}\mathbf{x}_n - \mathbf{u}_1^{\mathrm{T}}\overline{\mathbf{x}}\right\}^2 = \mathbf{u}_1^{\mathrm{T}}\mathbf{S}\mathbf{u}_1$$

Covariance matrix: $$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \overline{\mathbf{x}})(\mathbf{x}_n - \overline{\mathbf{x}})^{\mathrm{T}}$$

Maximize variance $\mathbf{u}_1^{\mathrm{T}}\mathbf{S}\mathbf{u}_1$ with constraint $\mathbf{u}_1^{\mathrm{T}}\mathbf{u}_1 = 1.$

Maximize $$\mathbf{u}_1^{\mathrm{T}}\mathbf{S}\mathbf{u}_1 + \lambda_1\left(1 - \mathbf{u}_1^{\mathrm{T}}\mathbf{u}_1\right)$$

λ_1: lagrange multiplier

Setting the derivative to zero:

$$\mathbf{S}\mathbf{u}_1 = \lambda_1\mathbf{u}_1$$

$\mathbf{u}_1$: an eigenvector of S

$$\mathbf{u}_1^{\mathrm{T}}\mathbf{S}\mathbf{u}_1 = \lambda_1$$

Largest eigenvalue $\longrightarrow$ maximum variance

$\mathbf{u}_1$: first principal component

Additional principal component $\mathbf{u}_n$: choosing a new direction among all possible directions orthogonal to $\mathbf{u}_1$, $\mathbf{u}_2$, ... $\mathbf{u}_{n-1}$, that maximize the projected variance

Optimal linear projection: eigenvectors $\mathbf{u}_1$, $\mathbf{u}_2$, ... $\mathbf{u}_M$ of covariance matrix S corresponding to M largest eigenvalues

Summary

1. Evaluating the mean and covariance matrix **S** of data set
2. Finding the M eigenvectors of S corresponding to M largest eigenvalues

Computation cost: $O(MD^2)$

Example: http://sebastianraschka.com/Articles/2014_pca_step_by_step.html

$$\widetilde{\mathbf{x}}_n = \sum_{i=1}^{M} z_{ni}\mathbf{u}_i + \sum_{i=M+1}^{D} b_i\mathbf{u}_i$$

$$J = \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}_n - \widetilde{\mathbf{x}}_n\|^2$$

3. Example

Applications:

Dimensionality reduction

Lossy data compression

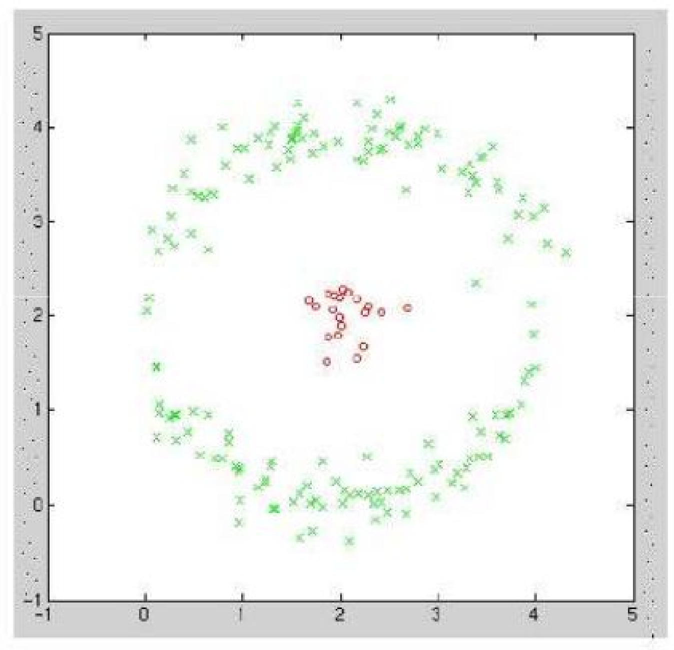
Feature extraction

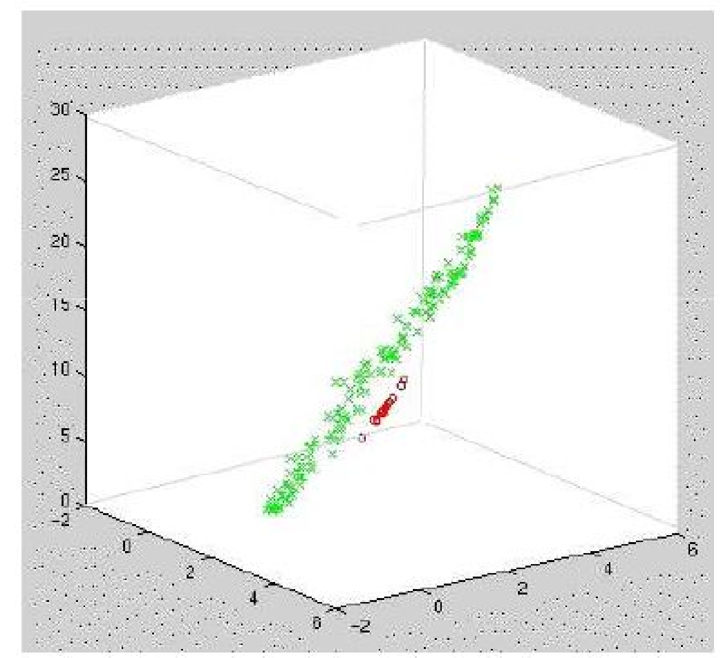
Data visualization

http://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_iris.html#sphx-glr-auto-examples-decomposition-plot-pca-iris-py

II. Kernel PCA

1. Why kernel





Not linear

Kernel trick $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$

2. Solution

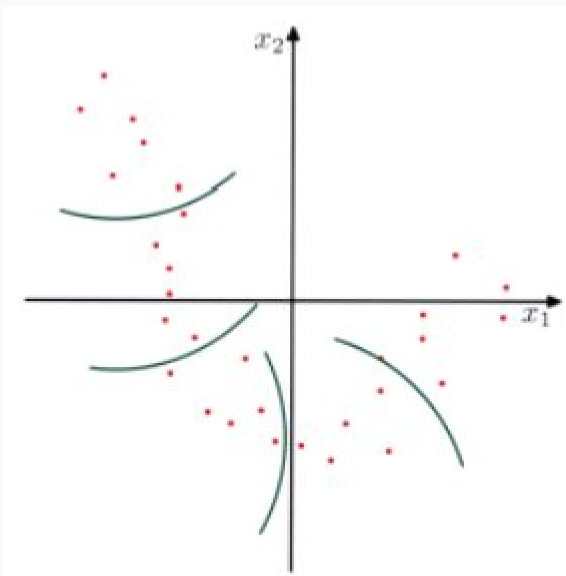
$\sum_n \mathbf{x}_n = \mathbf{0}.$

Covariance matrix $$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^{\mathrm{T}}$$

$\phi(\mathbf{x})$: a nonlinear transformation into an M-dimensional feature space (M > D)

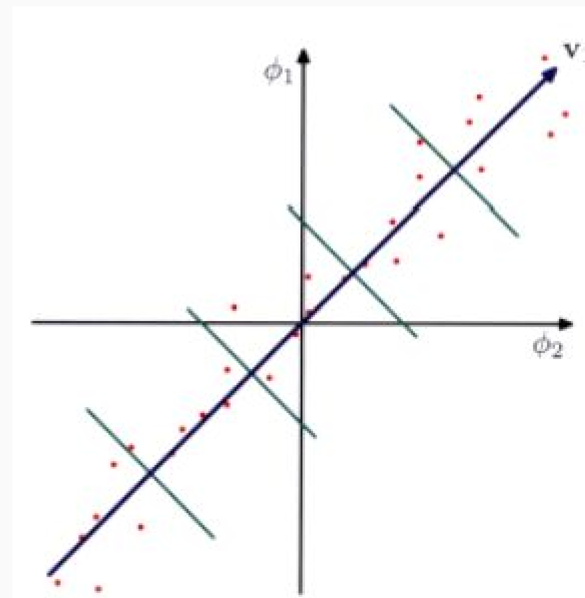
$\mathbf{x}_n \longrightarrow \phi(\mathbf{x}_n)$

Perform standard PCA in feature space



Original data space

Nonlinear projection



Feature space

Principal components

Assume $\sum_n \phi(\mathbf{x}_n) = \mathbf{0}$.

Covariance matrix in feature space:

$$\mathbf{C} = \frac{1}{N} \sum_{n=1}^{N} \phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^{\mathrm{T}}$$

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

$$\frac{1}{N} \sum_{n=1}^{N} \phi(\mathbf{x}_n) \left\{\phi(\mathbf{x}_n)^{\mathrm{T}}\mathbf{v}_i\right\} = \lambda_i \mathbf{v}_i$$

$$\mathbf{v}_i = \sum_{n=1}^{N} a_{in}\phi(\mathbf{x}_n)$$

Goal: solve the problem without having to work explicitly in the feature space

$$\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}}\sum_{m=1}^{N}a_{im}\boldsymbol{\phi}(\mathbf{x}_m) = \lambda_i\sum_{n=1}^{N}a_{in}\boldsymbol{\phi}(\mathbf{x}_n)$$

Kernel function: $k(\mathbf{x}_n, \mathbf{x}_m) = \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_m)$

Multiply by $\boldsymbol{\phi}(\mathbf{x}_l)^{\mathrm{T}}$

$$\frac{1}{N}\sum_{n=1}^{N}k(\mathbf{x}_l, \mathbf{x}_n)\sum_{m=1}^{m}a_{im}k(\mathbf{x}_n, \mathbf{x}_m) = \lambda_i\sum_{n=1}^{N}a_{in}k(\mathbf{x}_l, \mathbf{x}_n)$$

$$\mathbf{K}^2\mathbf{a}_i = \lambda_i N\mathbf{K}\mathbf{a}_i$$

a_i : N dimensional column vector with a_{in}

Kernel matrix

$$\mathbf{K} = \phi(\mathbf{X})\phi(\mathbf{X})^{\mathrm{T}} = \begin{bmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \kappa(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_N) \\ \kappa(\mathbf{x}_2, \mathbf{x}_1) & \kappa(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_N, \mathbf{x}_1) & \kappa(\mathbf{x}_N, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

$$\mathbf{K}\mathbf{a}_i = \lambda_i N \mathbf{a}_i$$

Normalization

$$1 = \mathbf{v}_i^{\mathrm{T}}\mathbf{v}_i = \sum_{n=1}^{N}\sum_{m=1}^{N} a_{in}a_{im}\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_m) = \mathbf{a}_i^{\mathrm{T}}\mathbf{K}\mathbf{a}_i = \lambda_i N \mathbf{a}_i^{\mathrm{T}}\mathbf{a}_i$$

Projection

$$y_i(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{v}_i = \sum_{n=1}^{N} a_{in}\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) = \sum_{n=1}^{N} a_{in}k(\mathbf{x}, \mathbf{x}_n)$$

Centralizing projected data:

$$\widetilde{\phi}(\mathbf{x}_n) = \phi(\mathbf{x}_n) - \frac{1}{N}\sum_{l=1}^{N}\phi(\mathbf{x}_l)$$

Element of the Gram matrix

$$\begin{aligned}
\widetilde{K}_{nm} &= \widetilde{\phi}(\mathbf{x}_n)^{\mathrm{T}}\widetilde{\phi}(\mathbf{x}_m) \\
&= \phi(\mathbf{x}_n)^{\mathrm{T}}\phi(\mathbf{x}_m) - \frac{1}{N}\sum_{l=1}^{N}\phi(\mathbf{x}_n)^{\mathrm{T}}\phi(\mathbf{x}_l) - \frac{1}{N}\sum_{l=1}^{N}\phi(\mathbf{x}_l)^{\mathrm{T}}\phi(\mathbf{x}_m) + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{l=1}^{N}\phi(\mathbf{x}_j)^{\mathrm{T}}\phi(\mathbf{x}_l) \\
&= k(\mathbf{x}_n, \mathbf{x}_m) - \frac{1}{N}\sum_{l=1}^{N}k(\mathbf{x}_l, \mathbf{x}_m) - \frac{1}{N}\sum_{l=1}^{N}k(\mathbf{x}_n, \mathbf{x}_l) + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{l=1}^{N}k(\mathbf{x}_j, \mathbf{x}_l)
\end{aligned}$$

In matrix notation $\widetilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N\mathbf{K} - \mathbf{K}\mathbf{1}_N + \mathbf{1}_N\mathbf{K}\mathbf{1}_N$

1_N: NxN matrix with each element 1/N

1. Pick a kernel

Gaussian $\boxed{K(\vec{x},\vec{x}') = \exp\left(-\beta \|\vec{x} - \vec{x}'\|^2\right)}$

Polynomial $\boxed{K(\vec{x},\vec{x}') = (1 + \vec{x} \cdot \vec{x}')^p}$

2. Construct the normalized kernel matrix of data (NxN)

$$\widetilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N\mathbf{K} - \mathbf{K}\mathbf{1}_N + \mathbf{1}_N\mathbf{K}\mathbf{1}_N$$

3. Solve an eigenvalue problem: $\widetilde{\mathbf{K}}\mathbf{a}_i = \lambda_i N \mathbf{a}_i$

4. Represent the data point i as

$$y_i(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{v}_i = \sum_{n=1}^{N} a_{in}\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) = \sum_{n=1}^{N} a_{in}k(\mathbf{x},\mathbf{x}_n)$$

3. Example

Example: http://sebastianraschka.com/Articles/2014_kernel_pca.html

RBF: radial basis function

http://scikit-learn.org/stable/auto_examples/plot_digits_pipe.html#sphx-glr-auto-examples-plot-digits-pipe-py

Reference

1. Pattern recognition and machine learning. Christopher M. Bishop. 2006
2. Lecture: kernel PCA. Unsupervised learning 2011. Rita Osadchy.
3. Implementing a Principal Component Analysis
   http://sebastianraschka.com/Articles/2014_pca_step_by_step.html
4. Kernel tricks and nonlinear dimensionality reduction via RBF kernel PCA
   http://sebastianraschka.com/Articles/2014_kernel_pca.html