Machine Learning and Pattern Recognition Practice Session III

Martin Palazzo

Universite de Technologie de Troyes
Universidad Tecnologica Nacional Buenos Aires
Biomedicine Research Institute of Buenos Aires - Max Planck Partner

martin.palazzo@utt.fr

October 31, 2020

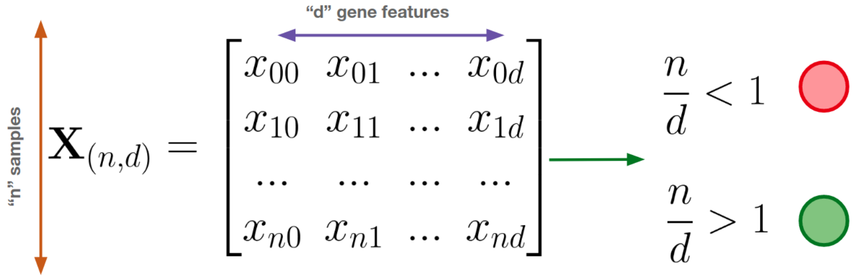
Overview

Disclaimer

The following document has been created as supporting and guiding material during the practical lessons during the Pattern Recognition course of the Master OSS. The official bibliography and theory materials have been distributed previously by the organization committee of the Master OSS.

Dimension Reduction

The curse of dimensionality



"d" gene features
"n" samples
$\mathbf{X}_{(n,d)} = \begin{bmatrix} x_{00} & x_{01} & \dots & x_{0d} \\ x_{10} & x_{11} & \dots & x_{1d} \\ \dots & \dots & \dots & \dots \\ x_{n0} & x_{n1} & \dots & x_{nd} \end{bmatrix}$
$\frac{n}{d} < 1$
$\frac{n}{d} > 1$

Figure: Sample to feature ratio.

Dimension reduction

Our data may lie in a high dimensional space $\mathcal{X}$. Nevertheless it is assumed that the inner structure of the data may lie in a *latent space* of lower dimensionality. For this reason it is possible to learn a transformation $\phi(x)$ that projects the input data in a low dimensional latent space.

Data transformation

$$\phi(x) = z$$

where $\mathcal{X} \in \mathbb{R}^d$, $\mathcal{Z} \in \mathbb{R}^p$ and $d > p$.

Dimension reduction

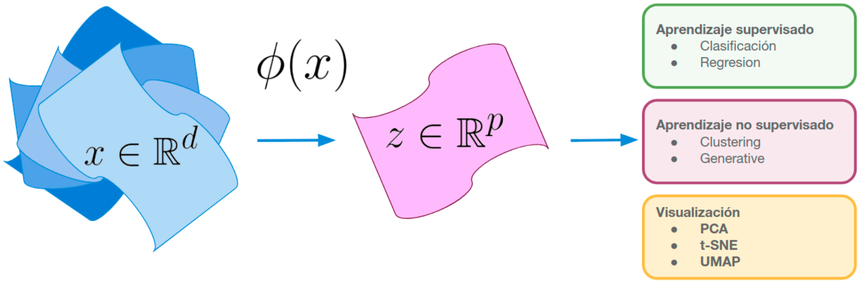


Figure: Dimension reduction

DImension reduction can be used for

- Downstream Supervised learning
- Downstream unsupervised learning
- Data visualization

Dimension reduction

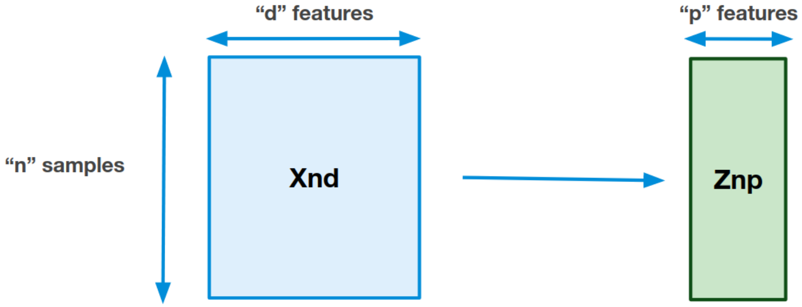


Figure: Dimension reduction

Dimension reduction

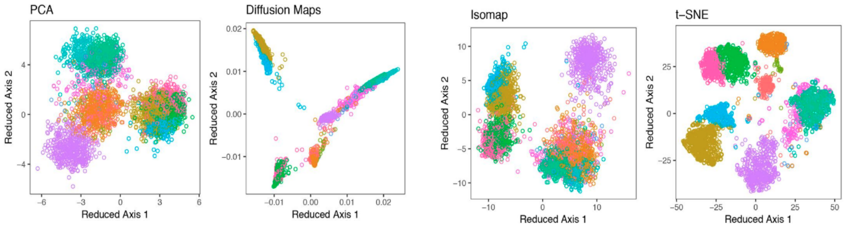


Figure: Dimension reduction

There exists multiple dimensionality reduction methods used for visualization [5].

Random projections

Random projection

The simplest way to learn a low dimensional representation of the high dimensional input data is to project it to a p-dimensional latent space build from random projections where $p < d$.

Random projections

$$z = \langle x, \beta \rangle$$

where $\beta_{ij} \sim U(0,1)$ [6].

Principal Component Analysis (PCA)

Principal Component Analysis

Principal component analysis is a linear transformation of the original input data to compute two main operations:

- Dimensionality reduction
- Data visualization

One of the goals is to reduce the dimensions of a d-dimensional data set by projecting it onto a (p)-dimensional subspace (where $p < d$) in order to filter noisy information, increase the computational efficiency and make data visualization while retaining most of the information.

PCA question

How should be the value of "p" to make an acceptable representation of the data?

Principal Component Analysis

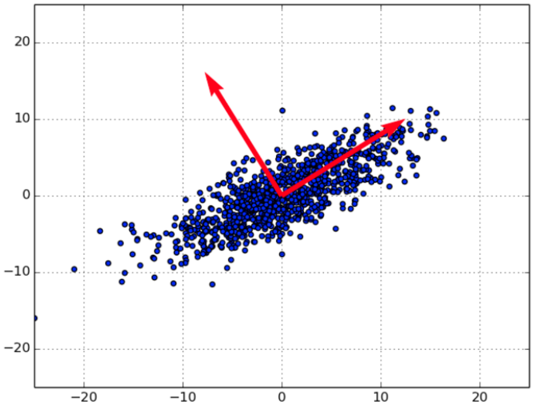


Figure: The PCA projects the data on the direction that maximizes the variability by building a d-dimensional elipse.

Principal Component Analysis

The following steps are required to perform PCA:

1. Standardize the data (mean 0 and std dev 1)
2. Perform eigen-decomposition on the covariance matrix and obtain Eigenvectors and Eigenvalues.
3. Sort eigenvalues in descending order and choose the p eigenvectors that correspond to the p largest eigenvalues where p is the number of dimensions of the new feature subspace.
4. Construct the matrix β $d \times p$ where the original input d-dimensional data will be projected using the selected top p eigenvectors.
5. Transform the original dataset X via β as $z = \langle X\beta \rangle$ to obtain a p-dimensional feature subspace z.

Principal Component Analysis

Once the data is standarized, the covariance matrix is built as follows

covariance between feature "j" and "p"

$$\sigma_{jp} = \frac{1}{n-1} \sum_{1}^{N} (x_{ij} - \bar{x}_j)(x_{ip} - \bar{x}_p)$$

Covariance matrix

$$\Sigma = \frac{1}{n-1}((X - \bar{x}^T)(X - \bar{x}))$$

where $\bar{x} = \sum_{i=1}^{n} x_i$ is a vector where each position is the sample mean of each feature.

Principal Component Analysis

then the eigen-decomposition is done as

eigen-decomposition

Given an $d \times d$ covariate matrix X, if it satisfies the required assumptions it allows the following decomposition

$$\Sigma = \upsilon \lambda \upsilon^{-1} \rightarrow \beta_{dp} \rightarrow X.\beta = z$$

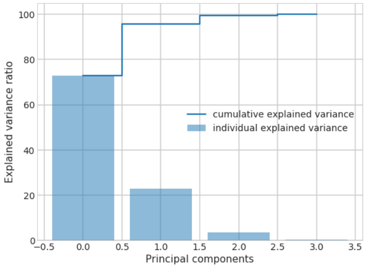
where v is a matrix of eigenvecors and λ a vector of eigenvalues.

Each eigen-vector will be a component and its associated eigen-value will be its importance (in other words, how much information that eigen-vector retains).

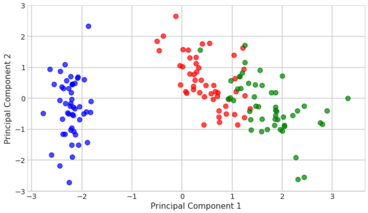
Principal Component Analysis

- In the reduced space generated by the PCA, the eigen-vectors will be the dimensions.
- The top eigen-vectors associated to the top eigen-values are picked and the rest of eigen-vectors are discarded.
- The associated eigen-value of each eigen-vector means the explained variance of it.
- For visualization purposes only the first two component are used, but consider the % of information these represent!!!

Principal Component Analysis



(a) PCA explained variance



(b) PCA projection using the top 2 components

Principal Component Analysis

Exercise

Given a dataset of dimensionality greater than 2, determine which is the best dimensionality reduction if we want to keep up to the 80% of the variability.

Exercise

Given the same dataset, visualize it by using the first two components.

Kernel Functions

Kernel Functions

Kernels are functions that can be thought as similarity measurements between vectors [4]. A kernel is a function which output value reflects the dot product between two vectors in a high dimensional hilbert space mapped by a $\phi(x)$.

Kernel Functions

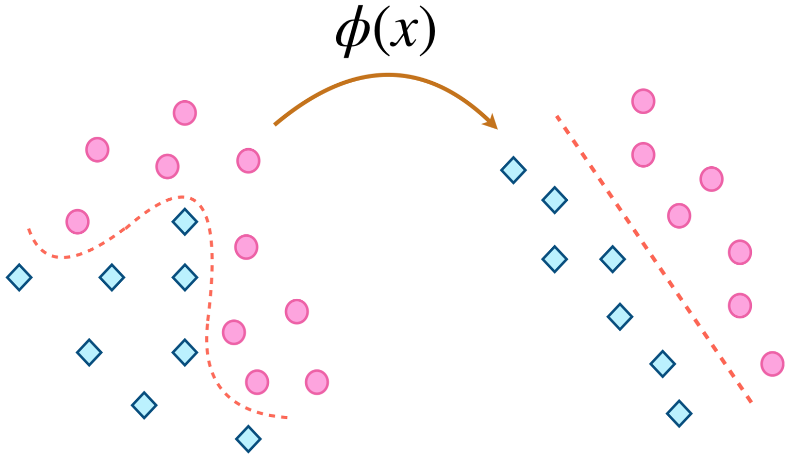


Figure: Reproducing Kernel Hilbert Space

Kernel Functions

Let X be a compact space. The function $k : X \times X \mapsto \mathbb{R}$ is symmetric and describes the mapping ϕ from X to a Reproducing Kernel Hilbert Space (RKHS) H through an inner product [3].

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_H \tag{1}$$

Here ϕ is a function that maps from X to a feature space H

$$\phi : X \mapsto \phi(X) \in H \tag{2}$$

Kernel Functions

Consider the vector of samples x_i belonging to a set S

$$S = \{x_1, , \cdots , x_m\} \tag{3}$$

The Gram Matrix of a Kernel [3] is built from S and is defined as an $m \times m$ matrix G with respective entries $G_{i,j} = \langle x_i, x_j \rangle$. The application of a Reproducing Hilbert Space Kernel function to compute the inner product between training vectors with a feature mapping ϕ allow to compute the following gram matrix:

$$G_{i,j} = \langle \phi\left(x_i\right), \phi\left(x_j\right) \rangle = k\left(x_i, x_j\right) \tag{4}$$

where each position of the Gram matrix if $G_{i,j} = 0$ corresponds to orthogonal training vectors i and j. We refer to high similarity when a pair of training vectors correspond to a value closer to 1 and a lower similarity when the gram matrix coordinates of two pair of vectors is closer to 0.

A summary

The core idea is to apply a transformation/mapping to the input feature vector X. The transformation is denoted as ϕ where ϕ_m corresponds to the m_{th} transformation of X and $m = 1...M$. The mapping $\phi(x)$ is used only for inner products. For this reason it is not necessary to determine the transformation $\phi(x)$ but it is required to know the positive and semi-definite kernel function $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ responsible to reproduce the inner products in the transformed space.

$$k\left(x_i, x_j\right) = \left\langle \phi(x_i), \phi(x_j) \right\rangle_H$$

$$\phi : X \rightarrow H$$

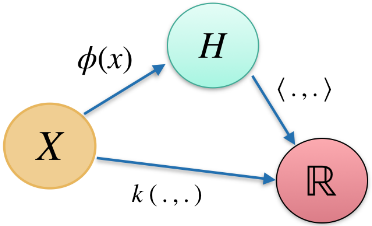


Figure: Reproducing Kernel Hilbert Space

RKHS functions

Linear Kernel

$$k_l(x_i, x_j) = x_i^T x_j + c \tag{5}$$

Polynomial Kernel

$$k_p(x_i, x_j) = (\alpha x_i^T x_j + c)^d \tag{6}$$

Gaussian Kernel

$$k_g(x_i, x_j) = \exp\left(\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) ; \sigma > 0 \tag{7}$$

Kernel operations

Multiple kernels can be taken into account to create better functions that finds an improved similarity between samples. Given two kernels k_1 and k_2 then a third kernel can be obtained by:

$$k_3 = k_1 + k_2$$

$$k_4 = \mu k_1$$

$$k_5 = \mu_1 k_1 + \mu_2 k_2$$

and k_3, k_4 and k_5 are valid kernels.

The gram matrix

$$G(x_1, x_2, x_3, ..., x_n) = \begin{vmatrix} \langle x_1, x_1 \rangle & \langle x_1, x_2 \rangle & ... & \langle x_1, x_n \rangle \\ \langle x_2, x_1 \rangle & ... & ... & ... \\ ... & ... & ... & ... \\ \langle x_n, x_1 \rangle & ... & ... & \langle x_n, x_n \rangle \end{vmatrix}$$

About the matrix

The gram matrix contains on each position the pairwise similarity measurements done by the RKHS function between each pair of samples on the dataset.

The gram matrix

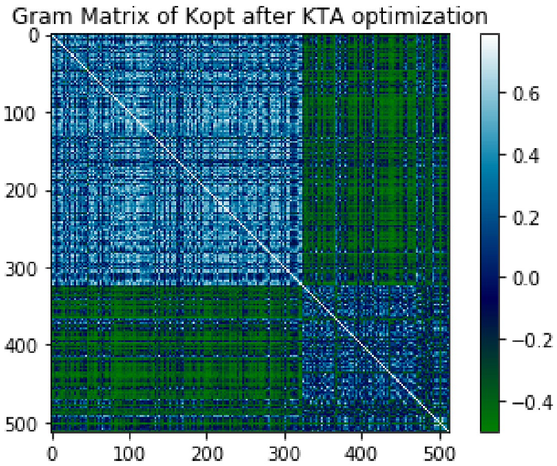


Figure: The gram matrix of a dataset using gaussian kernel and 2 classes.

The gram matrix

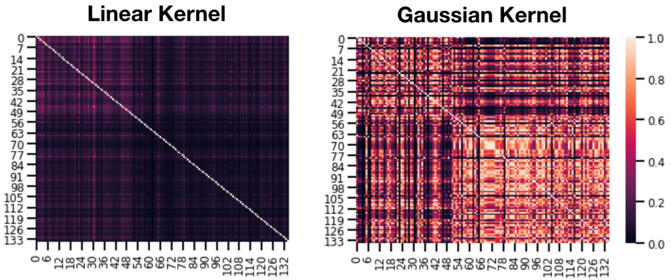


Figure: The gram matrix of the linear and gaussian kernel.

The gram matrix

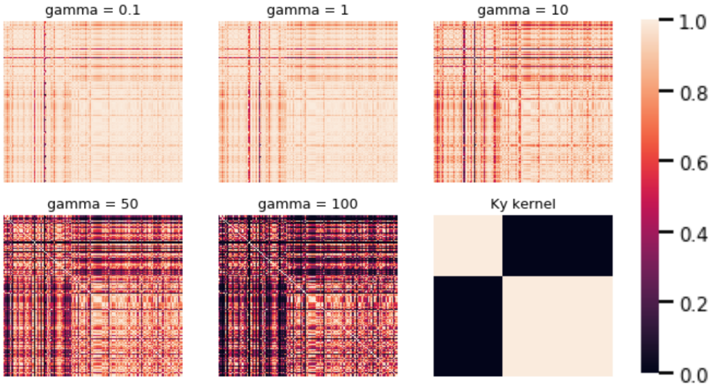
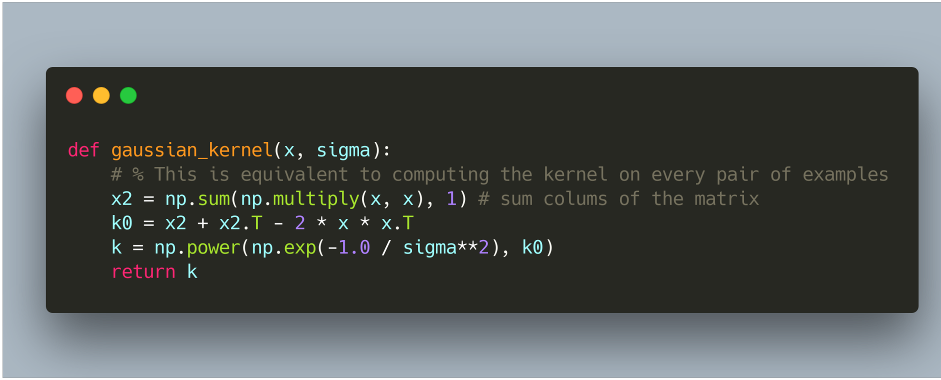


Figure: The gram matrix for different values of gamma parameters in a RBF kernel.

Kernel functions



```python
def gaussian_kernel(x, sigma):
    # % This is equivalent to computing the kernel on every pair of examples
    x2 = np.sum(np.multiply(x, x), 1) # sum colums of the matrix
    k0 = x2 + x2.T - 2 * x * x.T
    k = np.power(np.exp(-1.0 / sigma**2), k0)
    return k
```

Figure: A pairwise gaussian kernel implementation on pyhon

Kernel functions

Exercise

Given a dataset, build the Gram matrix using Gaussian kernels for different values of sigma. Then try to find which sigma parameter is better for a two class problem. Plot the gram matrix.

Exercise

Combine two types of kernels and find if it is possible to improve the gram matrix to obtain a better separability between the two classes.

Kernel PCA

Kernel PCA

- Classic PCA is a linear projection of the samples in a low dimensional space. By taking advantage of kernel funcions a Kernel-PCA finds a non-linear projection of the samples.
- Instead of computing the co-variance matrix in the high dimensional hilbert space, the gram matrix is used.
- The co-variance matrix in the higher dimensional space is not calculated explicitly (kernel trick). Therefore, the implementation of kernel PCA does not yield the principal component axes (in contrast to the standard PCA), but the obtained eigen-vectors can be understood as projections of the data onto the principal components.

Kernel PCA

The eigen-decomposition can be applied on the gram matrix K. A set of eigenvalues U and eigenvectors Δ are obtained. The data is projected to a low dimensional space using the U^* using the first p eigenvectors from the Δ matrix associated to the top p eigen values.

Kernel-PCA

$$K = nU^{-1}\Lambda U \rightarrow Z_{np} = KU^*$$

If the obtained kernel is non-linear therefore the dimension reduction is non-linear as well.

Kernel PCA

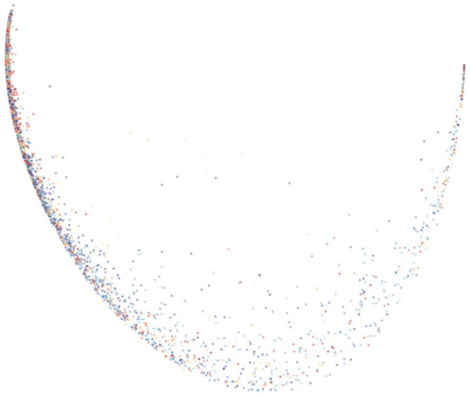


Figure: A Kernel PCA visualization

Bishop, C. M. (2006). Pattern recognition and machine learning. springer.

Friedman, J., Hastie, T., Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.

Shawe-Taylor, J., Cristianini, N. (2004). Kernel methods for pattern analysis. Cambridge university press.

Palazzo, M., Beauseroy, P., Yankilevich, P. (2019). Hepatocellular Carcinoma tumor stage classification and gene selection using machine learning models. Electronic Journal of SADIO (EJS), 18(1), 26-42.

Konstorum, A., Jekel, N., Vidal, E., Laubenbacher, R. (2018). Comparative analysis of linear and nonlinear dimension reduction techniques on mass cytometry data. bioRxiv, 273862.

Bingham, E., Mannila, H. (2001, August). Random projection in dimensionality reduction: applications to image and text data. In

Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 245-250).