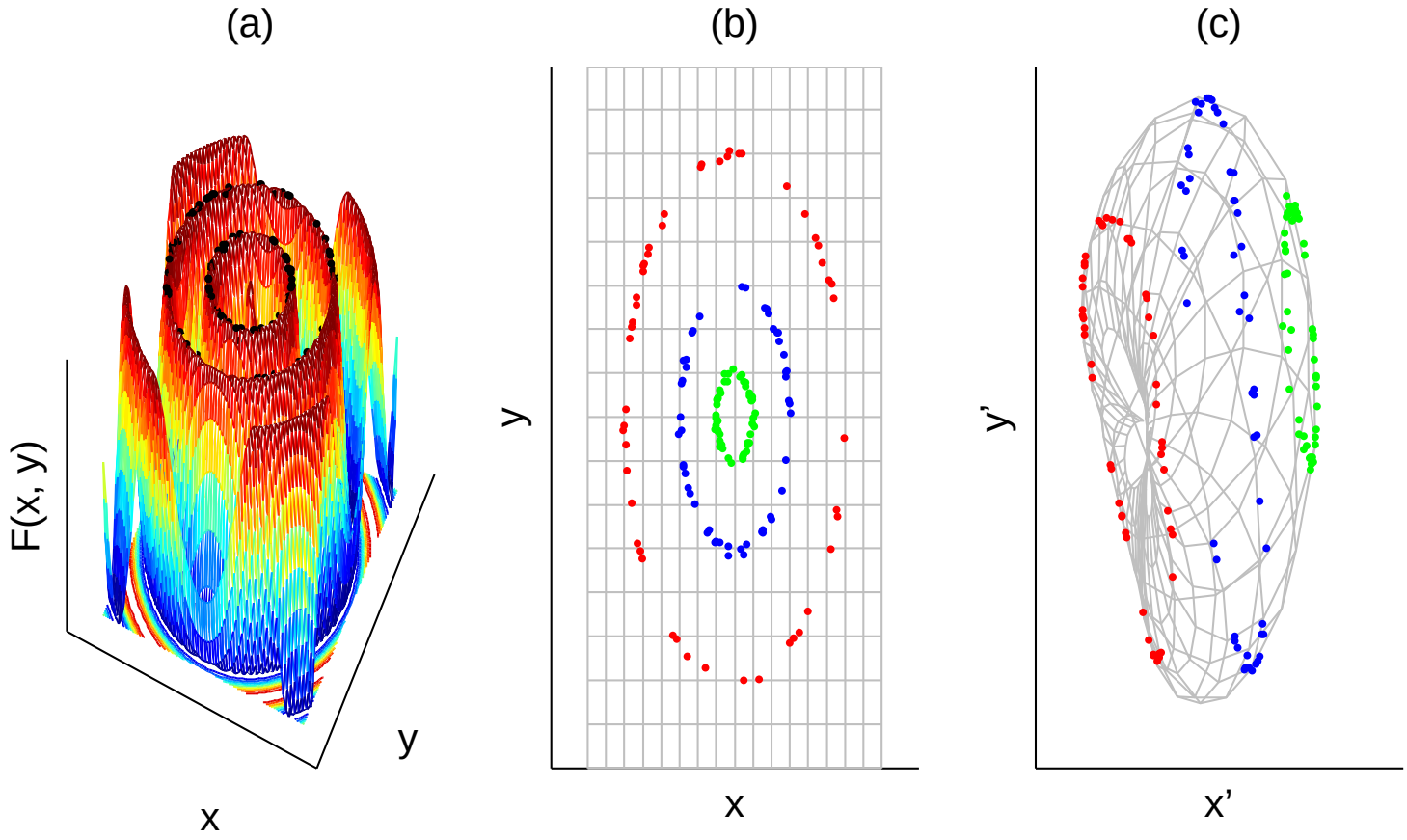
Metody redukcji wymiarowości danych
Kernel Principal Component Analysis

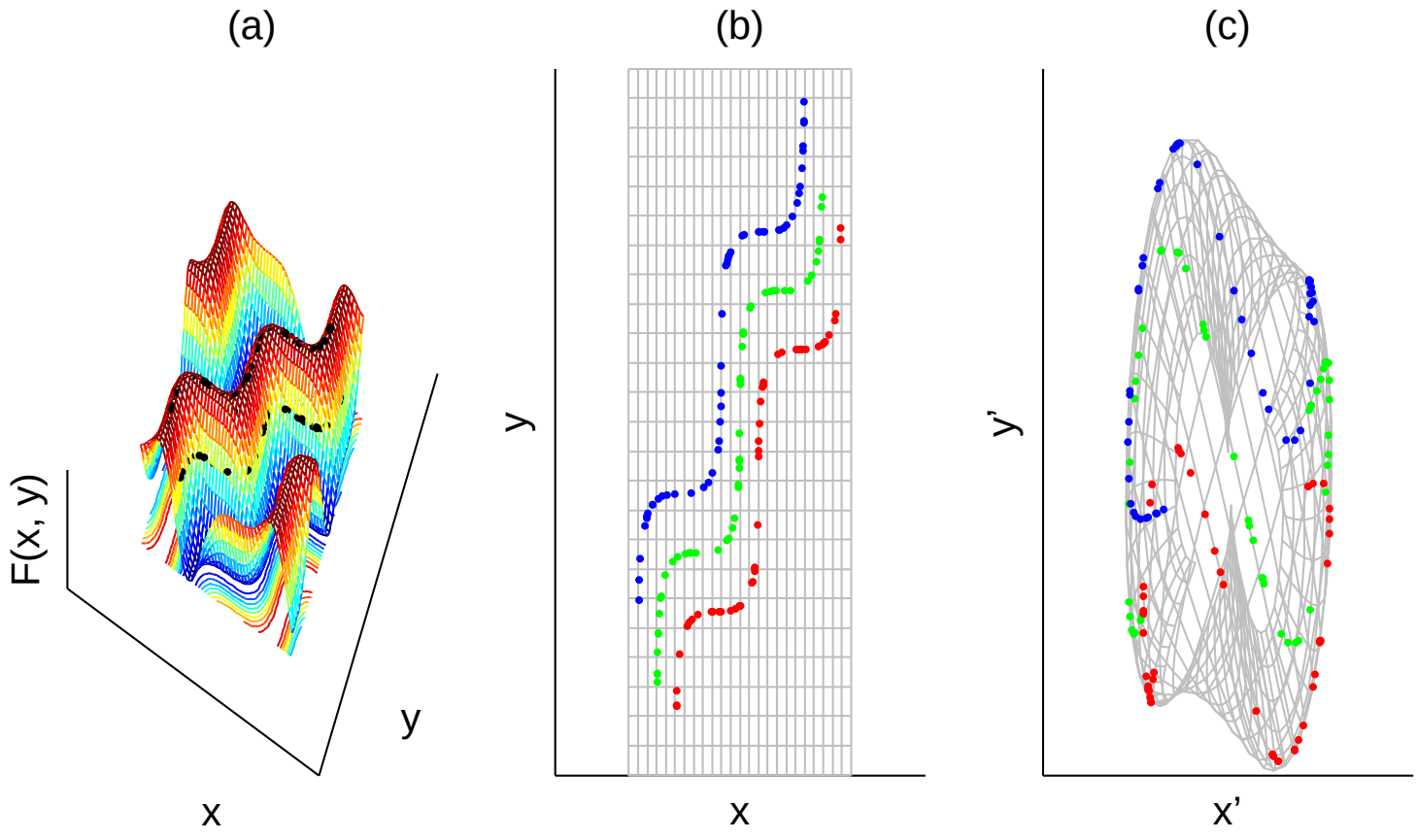
Piotr Lipiński

wykład z eksploracji danych

Introduction



Introduction



Overview of KPCA

- transform the search space to a highly dimensional data space, where it is easier to separate data points, i.e. to simplify the geometry of the manifold
- reduce the dimensionality of the highly dimensional mapping of the data sample by transforming it to a new lower dimensional search space

Implementation

Let $\Phi : \Omega \to \Pi$ be a mapping, not necessarily linear, from the original search space Ω to a hypothetical highly dimensional data space Π, which may be even of the infinite dimensionality, assumed only to be a Hilbert space. At the beginning, assume that the data sample $\mathcal{D}$ mapped by the transformation Φ is centered, i.e.

$$\sum_{i=1}^{N} \Phi(\mathbf{x}_i) = 0, \tag{1}$$

where $\Phi(\mathbf{x}_i)$ is the mapping of the data point $\mathbf{x}_i$ in the data space Π.

Implementation, continued

In order to try to reduce the dimensionality of the mapped data sample in the highly dimensional data space Π, one may try to perform the classic Principal Component Analysis (PCA) and to find eigenvalues and eigenvectors of the covariance matrix

$$\mathbf{\Sigma} = \frac{1}{N} \sum_{I=1}^{N} \Phi(\mathbf{x}_I) \Phi(\mathbf{x}_I)^T. \tag{2}$$

It is obvious that all the eigenvectors must be linear combinations of $\Phi(\mathbf{x}_1)$, $\Phi(\mathbf{x}_2)$, ..., $\Phi(\mathbf{x}_N)$ and lie in the span of $\Phi(\mathbf{x}_1)$, $\Phi(\mathbf{x}_2)$, ..., $\Phi(\mathbf{x}_N)$, so the eigenvalue equation

$$\lambda \mathbf{v} = \mathbf{\Sigma} \mathbf{v}, \tag{3}$$

where λ denotes the eigenvalue and $\mathbf{v}$ denotes the corresponding eigenvector of the matrix $\mathbf{\Sigma}$, is equivalent to a system of linear equations

$$\Phi(\mathbf{x}_i)^T (\lambda_k \mathbf{v}_k) = \Phi(\mathbf{x}_i)^T (\mathbf{\Sigma} \mathbf{v}_k), \qquad \text{for each } i = 1, 2, \ldots, N, \tag{4}$$

for $k = 1, 2, \ldots, K$, where $\lambda_1, \lambda_2, \ldots, \lambda_K$ are non-zero eigenvalues and $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K$ are corresponding eigenvectors of the matrix $\mathbf{\Sigma}$.

Implementation, continued

As each eigenvector $\mathbf{v}_k$ is a linear combination of $\Phi(\mathbf{x}_1)$, $\Phi(\mathbf{x}_2)$, ..., $\Phi(\mathbf{x}_N)$, let

$$\mathbf{v}_k = \sum_{j=1}^{N} \alpha_{kj} \Phi(\mathbf{x}_j), \tag{5}$$

where $\boldsymbol{\alpha}_k = (\alpha_{k1}, \alpha_{k2}, \ldots, \alpha_{kN})^T \in \mathbb{R}^N$ is a vector of linear coefficients. Therefore, the system of linear equations (4) reduces to

$$\lambda_k \sum_{j=1}^{N} \alpha_{kj} \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) = \sum_{j=1}^{N} \alpha_{kj} \Phi(\mathbf{x}_i)^T \boldsymbol{\Sigma} \Phi(\mathbf{x}_j), \qquad \text{for each } i = 1, 2, \ldots \tag{6}$$

and using (2) to

$$\lambda_k \sum_{j=1}^{N} \alpha_{kj} \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) = \frac{1}{N} \sum_{j=1}^{N} \sum_{l=1}^{N} \alpha_{kj} \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_l) \Phi(\mathbf{x}_l)^T \Phi(\mathbf{x}_j), \tag{7}$$

Implementation, continued

thus after defining a matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ with elements $k_{ij} = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$, the system of linear equations (4) is equivalent to

$$N\lambda_k \boldsymbol{\alpha}_k \mathbf{K} = \mathbf{K}^2 \boldsymbol{\alpha}_k. \tag{8}$$

Finally, it may be shown that the solution to the system of linear equations (4) are the eigenvalues $N\lambda_k$ and corresponding eigenvectors $\boldsymbol{\alpha}_k$ of the matrix $\mathbf{K}$. Therefore, the mappings $\Phi(\mathbf{x}_i)$ of data points $\mathbf{x}_i$ may be reduced to $\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{iq}) \in \mathbb{R}^K$ by projecting $\Phi(\mathbf{x}_i)$ onto the K eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_q$ corresponding to the K non-zero eigenvalues, so

$$y_{ik} = \mathbf{v}_k^T \Phi(\mathbf{x}_i) = \sum_{j=1}^{N} \alpha_{kj} \Phi((\mathbf{x}_j)^T \Phi((\mathbf{x}_i) = \sum_{j=1}^{N} \alpha_{kj} k_{ji}. \tag{9}$$

Implementation, continued

Concerning the assumption (1) that mappings are centered, it may be relaxed with transforming the arbitrary kernel matrix $\mathbf{K}$ into

$$\mathbf{K} = \mathbf{K} - \mathbf{1K} - \mathbf{K1} + \mathbf{1K1}, \tag{10}$$

where $\mathbf{1}$ is a matrix of ones divided by N of the same size than the matrix $\mathbf{K}$.

Implementation, continued

In the literature, a number of different kernel functions has been proposed, among the others, the polynomial kernels, the gaussian or radial basis kernels, the sigmoid kernels and the kernels constructed from particular mappings. We focus on gaussian kernels defined by the gaussian kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$:

$$k(\mathbf{x}, \mathbf{y}) = \exp(-\frac{||\mathbf{x} - \mathbf{y}||^2}{2\sigma^2}), \tag{11}$$

where σ is a parameter and the kernel matrix $\mathbf{K}$ defined by $k_{ij} = k(\mathbf{x}_i, \mathbf{y}_j)$.