Manifold Learning – Modern Approaches for Dimensionality Reduction

Generalized Principal Component Analysis

Talk: Alexander Pohl

Manifold Learning

- “The simplest description of manifold learning is that it is a class of algorithms for recovering a low-dimensional manifold embedded in a high-dimensional ambient space” [MF12, p. 1]
- Reduction of dimensionality without - or an insignificant amount - of loss of information contained in a given dataset
- Extraction of important features to make algorithms computationally cheap and memory efficient
- Fundamental methods such as Principal Component Analysis (PCA) and Multidimensional Scaling (MDS) are restricted to linear embeddings
- Expansion to techniques which can capture nonlinear, low-dimensional structures of high-dimensional data

Presentation Outline

(1) Principal Component Analysis in the complete data case
   (1) Summary of theoretical basics
   (2) Principal Component Analysis
   (3) R-packages for practical implementations
   (4) Analysis of datasets

(2) Principal Component Analysis in the incomplete data case
   i. Iterative PCA algorithm
   ii. NIPALS algorithm
   iii. R-packages for practical implementations
   iv. Analysis of datasets

(3) Nonlinear extensions
   i. Nonlinear PCA
   ii. Kernel PCA
   iii. R-packages for practical implementations
   iv. Analysis of datasets

(4) Conclusions

PRINCIPAL COMPONENT ANALYSIS IN THE COMPLETE DATA CASE

Summary of theoretical basics – Schur decomposition

Be $A \in \mathbb{R}^{n \times n}$ a matrix. If the characteristic polynomial $\mathcal{X}_A$ can be factorized in the following form:

$$\mathcal{X}_A = (\lambda_1 - x) * \cdots * (\lambda_n - x)$$

Then there exists an orthogonal matrix $\mathrm{U} \in \mathbb{R}^{n \times n}$, such that

$$U^T A U = \Sigma = \begin{pmatrix} \lambda_1 & \cdots & * \\ 0 & \ddots & \vdots \\ 0 & 0 & \lambda_n \end{pmatrix}$$

Note:

- The diagonal elements of Σ representing the eigenvalues of A
- In the special case of A being a normal matrix, the resulting matrix Σ is a diagonal matrix and the procedure is then also known as spectral decomposition
- The columns of U, $u_1, \dots, u_n$ representing the eigenvectors for the corresponding eigenvalues $\lambda_1, \dots, \lambda_n$

Summary of theoretical basics – Singular Value Decomposition (SVD)

Be $A \in \mathbb{R}^{m \times n}$ an arbitrary real matrix. There exist orthogonal matrices U $\in \mathbb{R}^{m \times m}$ and V $\in \mathbb{R}^{n \times n}$, such that:

$$U^T A V = \Sigma = diag(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m \times n}$$

$$\Leftrightarrow A = U \Sigma V^T, \quad \text{with } p := \min(m, n),\ \sigma_1 \geq \dots \geq \sigma_p$$

Properties:

- Denote u_i and v_i the columns of U and V respectively. It applies:

$$A v_i = \sigma_i u_i \ \wedge \ A^T u_i = \sigma_i v_i \qquad , \forall\, i = 1, \dots, p$$

- Be $\sigma_1 \geq \dots \geq \sigma_r \geq \sigma_{r+1} = \dots = \sigma_p = 0$. It applies:

$$rang(A) = r$$

$$Ker(A) = span(v_{r+1}, \dots, v_p), \quad Im(A) = span(u_1, \dots, u_r)$$

- The squared singular values correspond to the eigenvalues of $A^T A$ and $A A^T$ with the associated eigenvectors being $v_1, \dots, v_p$ and $u_1, \dots, u_p$ respectively

Principal Component Analysis

- Objective:
  - Fitting a low-dimensional affine subspace / linear manifold of dimension $d \ll D$ to a set of points $\{x_1, \dots, x_N\} \in \mathbb{R}^D$
  - Thereby preserving most of the information of the given dataset
  - SVD provides an optimal solution to the PCA problem

So called "principal components" $y \in \mathbb{R}^d$ of $x \in \mathbb{R}^D$ are defined as the d uncorrelated linear components of x:

$$y_i = u_i^T x \in \mathbb{R}, \qquad u_i \in \mathbb{R}^D, i = 1, \dots, d$$

such that the variance of y_i is maximized subject to:

$$u_i^T u_i = 1 \wedge Var[y_1] \geq \dots \geq Var[y_d] > 0$$

First principal component y_1 is received by seeking for $u_1^* \in \mathbb{R}^D$ through solving:

$$\boxed{\boldsymbol{u_1^* = \max_{u_1 \in \mathbb{R}^D} Var(u_i^T x), \qquad s.t.\ \ u_i^T u_i = 1}}$$

Principal Component Analysis

Note:

$$Var(u_i^T x) = E\left(\left(u_i^T x\right)^2\right) = E\left(u_i^T x x^T u_i\right) = u_i^T \Sigma_x u_i$$

And therefore the constrained optimization problem can be written as:

$$u_1^* = \max_{u_1 \in \mathbb{R}^D} u_1^T \Sigma_x u_1 , \qquad s.t.\ \ u_i^T u_i = 1$$

The constrained optimization problem can be solved by the method of Lagrange multipliers.

The solution is given as:

$$\boxed{\underbrace{\boldsymbol{\Sigma_x u_1 = \lambda_1 u_1}}_{\textbf{Eigenvalue equation}} \qquad \wedge \qquad \boldsymbol{u_1^T u_1 = 1}}$$

Note:

$$\lambda_1 = Var[u_1^T x] = Var[y_1]$$

Principal Component Analysis

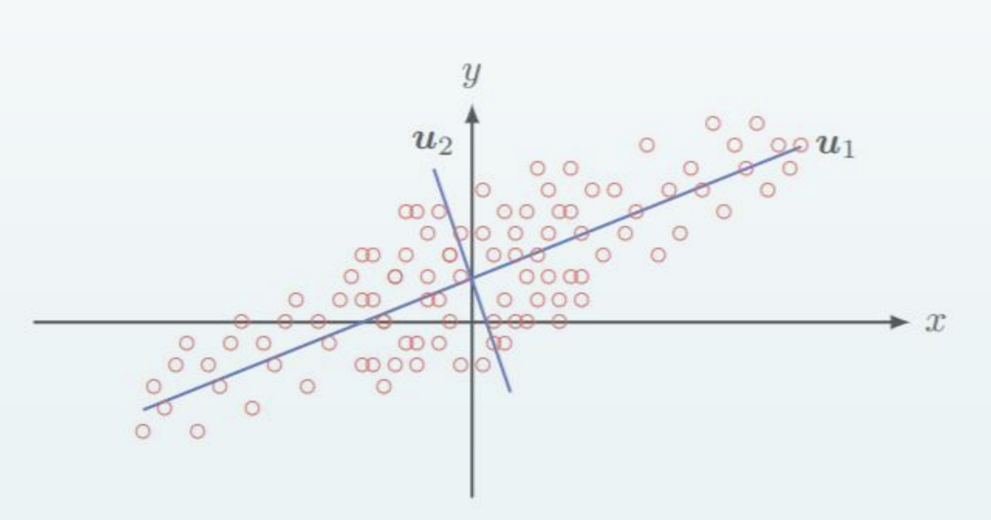
The solution of the Lagrangian for the further principal axes also results in eigenvalue equations. Therefore simultaneous calculation of the principal components can be applied:

$$\boxed{\boldsymbol{\Sigma_y} = \boldsymbol{E}[\boldsymbol{y}\boldsymbol{y}^{\boldsymbol{T}}] = \boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{E}[\boldsymbol{x}\boldsymbol{x}^{\boldsymbol{T}}]\boldsymbol{U}}$$

Note:

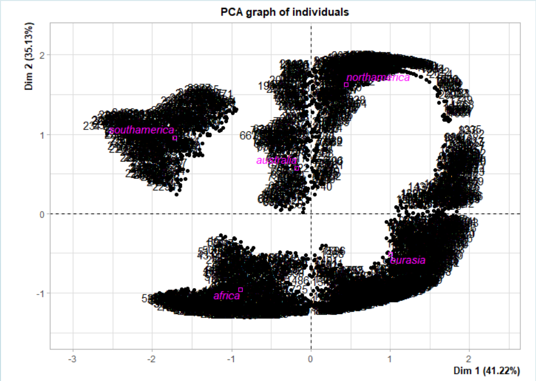
In the case of a high-dimensional data Matrix XX^T one can compute the singular vectors and singular values of X instead

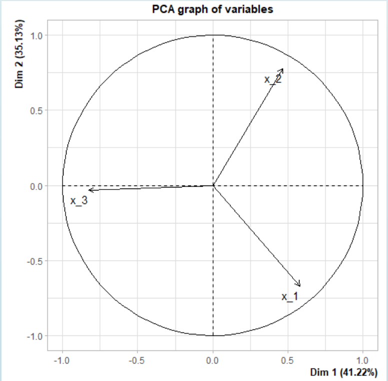
⇒ SVD: $\mathrm{X} = U\Sigma V^T$



R-packages for practical implementations

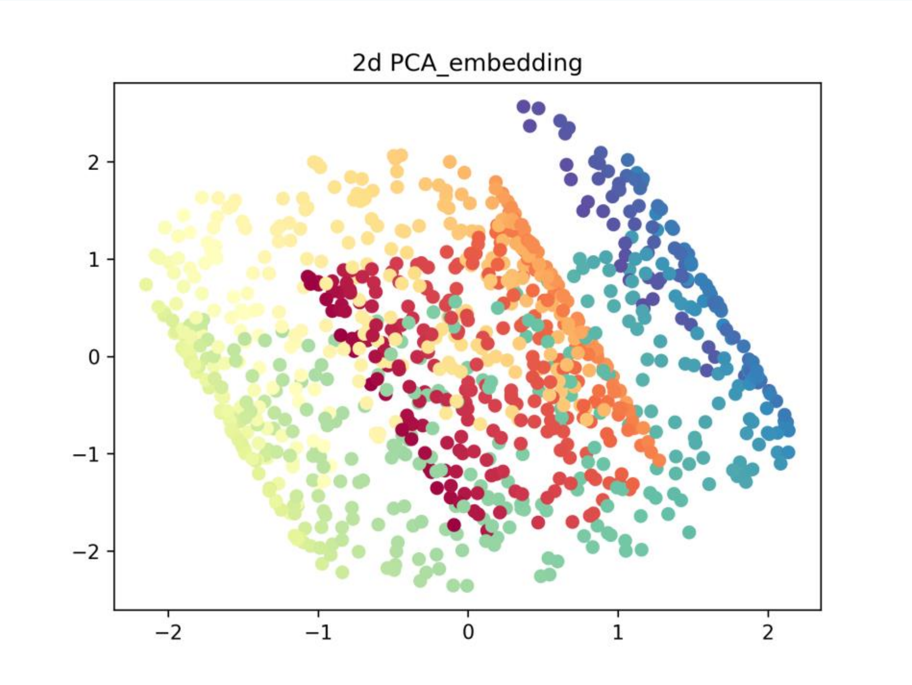
- **'stats' – package [RCore20]:**
  - function 'prcomp': uses SVD and variance computation $\frac{1}{N-1}\sum_{i=1}^{N}\|x_i - \bar{x}\|^2$
  - function 'princomp': uses spectral decomposition on the correlation or covariance matrix with default divisor $\frac{1}{N}$
- **'FactoMineR' – package [LJH08]:**
  - function 'PCA': Performs PCA with supplementary individuals, supplementary quantitative variables and supplementary categorical variables. Returns the individuals factor map and the variables factor map.



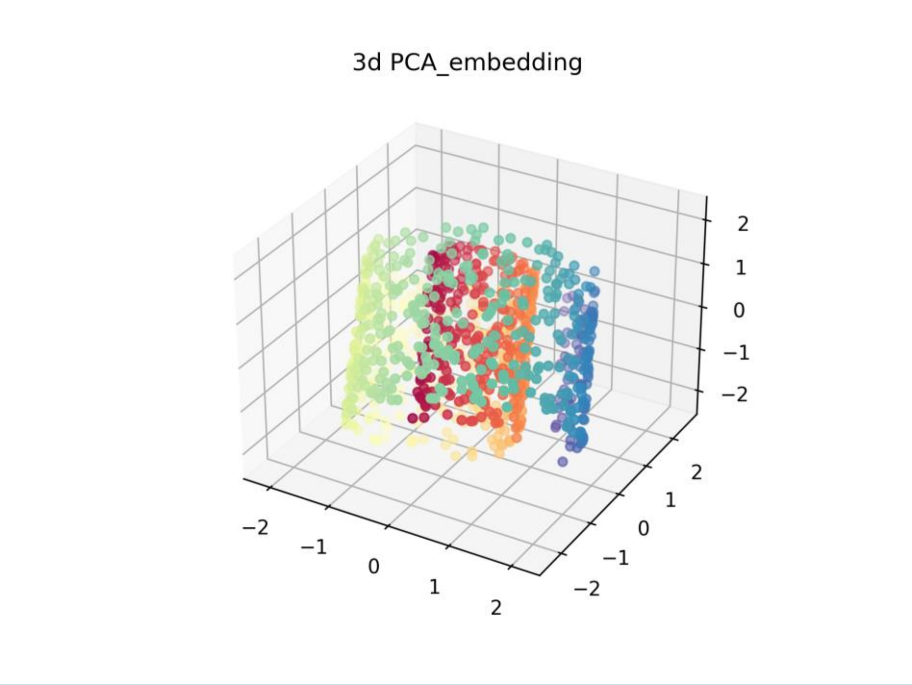


Analysis of datasets - Swissroll

Area under the R_{NX} curve ≈ 0.52

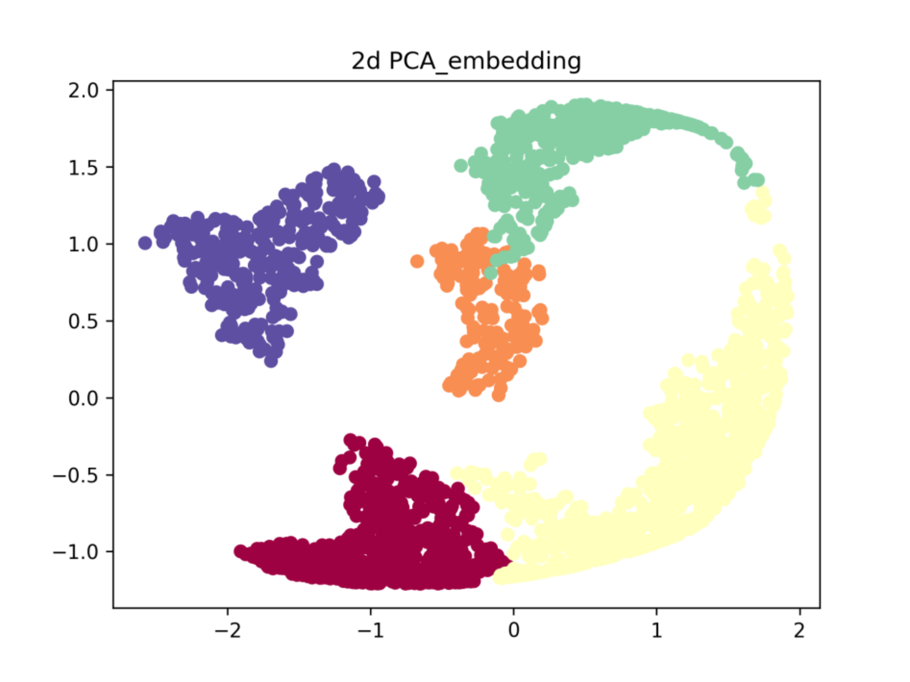


Area under the R_{NX} curve ≈ 0.95

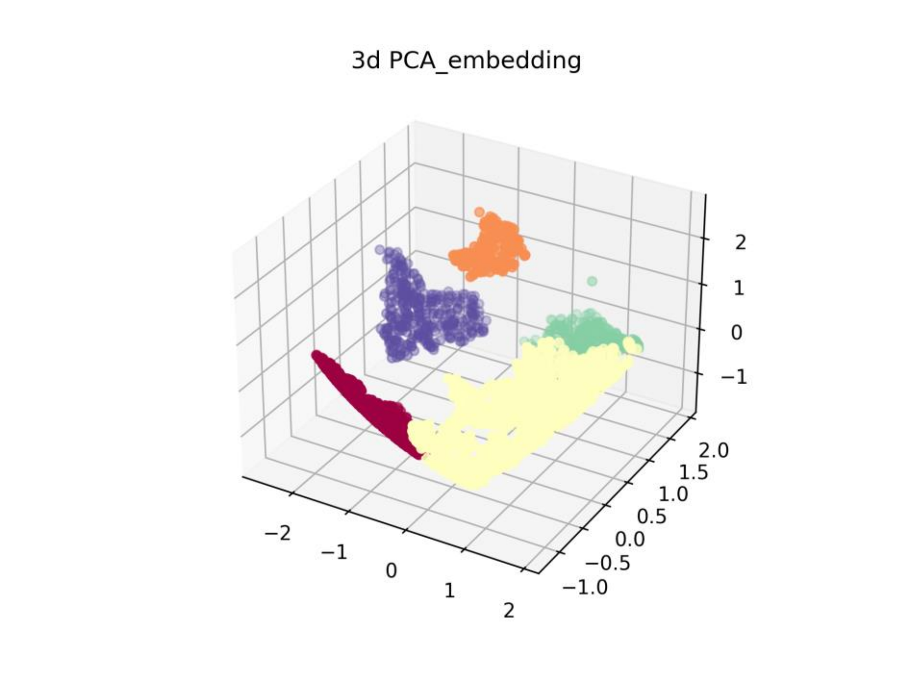


Analysis of datasets - World

Area under the R_{NX} curve ≈ 0.62

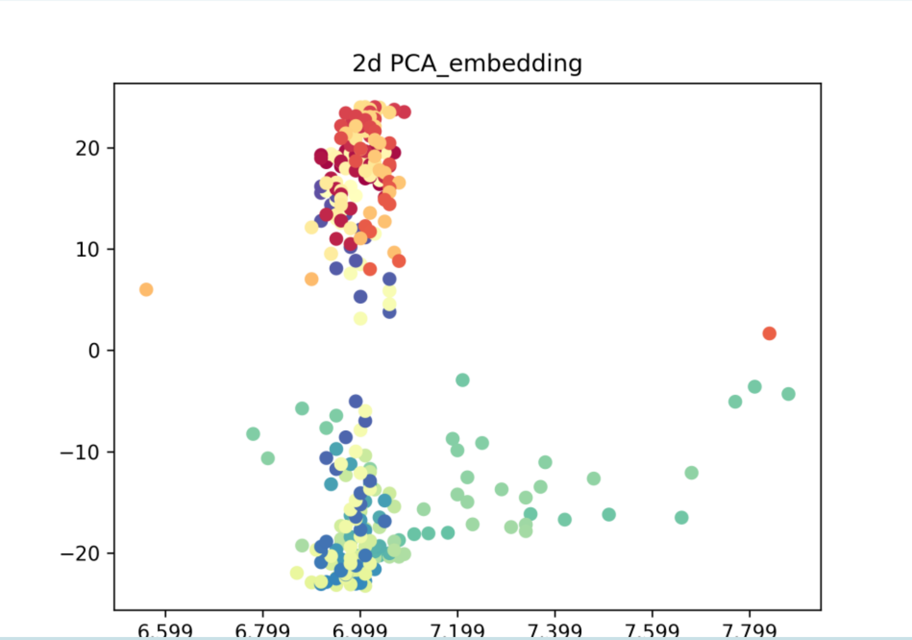


Area under the R_{NX} curve ≈ 0.73

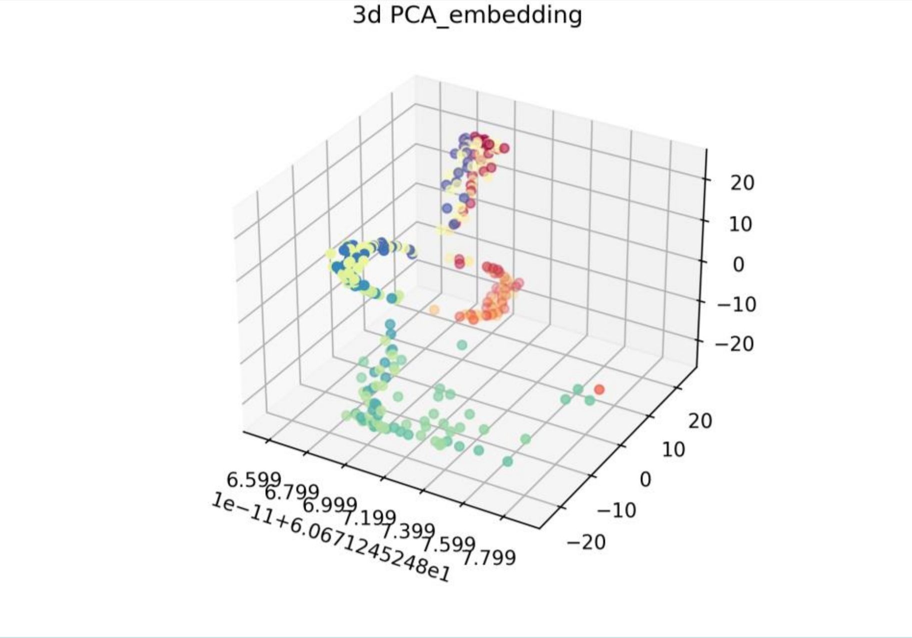


Analysis of datasets – Clock

Area under the R_{NX} curve ≈ 0.29



Area under the R_{NX} curve ≈ 0.73



Principal Component Analysis in the Incomplete Data Case

The incomplete data case

- Missing values are ubiquitous in practice and can occur for a number of reasons
- Grouped into types of missingness: MCAR, MAR, NMAR
- Many statistical methods such as PCA can not be directly applied to the incomplete data case
- Simple single imputation techniques can suffer from several drawbacks:
  - Mean imputation preserves the mean of the imputed variable but reduces its variance and can distort the correlation with other variables
  - Imputation by regression accounts for the relationship between variables but marginal and joint distribution of the variables can still be distorted
  - Imputed values are considered as observed values and the uncertainty of the prediction is therefore not reflected in the subsequent analyses

Can lead to underestimated standard errors of the parameters and overoptimistic tests and confidence intervals!

Incomplete data case – fixed effect PCA model

PCA can also be explained as estimation of a fixed effect model:

$$\boldsymbol{x_i} = \boldsymbol{\mu} + \boldsymbol{U y_i} + \boldsymbol{\varepsilon_i}, \boldsymbol{\varepsilon_i} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma^2 I})$$

In the case of missing values this leads to minimizing a weighted least squares criterion:

$$\boxed{\sum_{i=1}^{N}\sum_{k=1}^{D} w_{i,k}\left(x_{i,k} - \mu_k - \sum_{l=1}^{d} y_{i,l}u_{k,l}\right)^2}$$

There exists no explicit solution to this minimization problem, therefore its necessary to resort to iterative algorithms.

The iterative PCA algorithm

- Focus on best possible estimation of the parameters and their variance and **not** to provide best prediction of the missing values
- Imputation of the missing values is achieved during the estimation process
- Corresponds to an expectation maximization algorithm
- Takes into account the similarities between individuals as well as dependencies between variables
- Standardization should be executed after each iteration
- Can suffer from overfitting especially with increasing amount of missingness and dimensionality

The iterative PCA algorithm

(1) Initialize $X^{(0)}$: Replace missing values via mean imputation

(2) For $t = 0, \dots, T$ or until a certain stopping criteria:

a) Perform PCA on the completed dataset to estimate parameters $\widehat{\mu^{(t)}}, \widehat{U^{(t)}}, \widehat{y^{(t)}}$

b) Keep only dimensions $1, \dots, d$

c) Calculate $\widehat{X^{(t)}} = \widehat{\mu^{(t)}} + \widehat{U^{(t)}}\widehat{y^{(t)}}$

d) Impute missing values and keep observed values: $X^{(t+1)} = W * X^{(t)} + (1 - W)\widehat{X^{(t)}}$

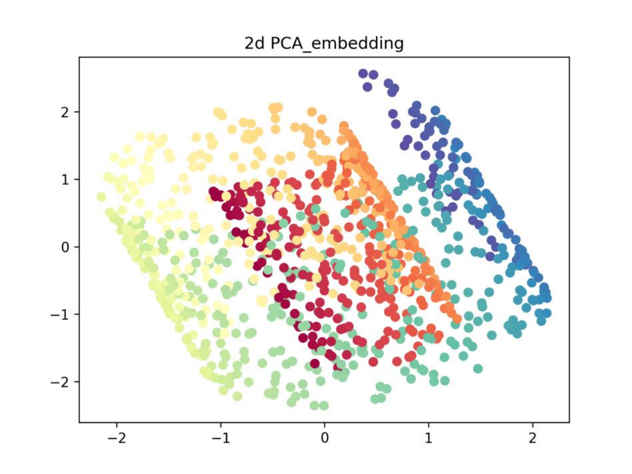
Regularized iterative PCA algorithm

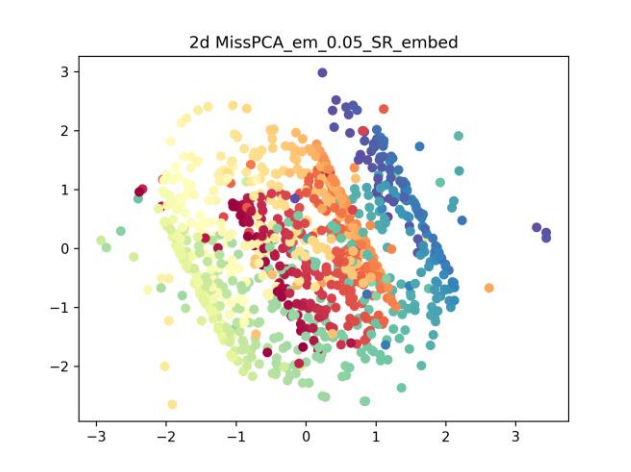
- Overfitting can be reduce by decreasing the number of dimensions
- Can in return result in loss of information
- Regularized version uses shrinkage method to overcome the overfitting problem an replacement of the imputation step:

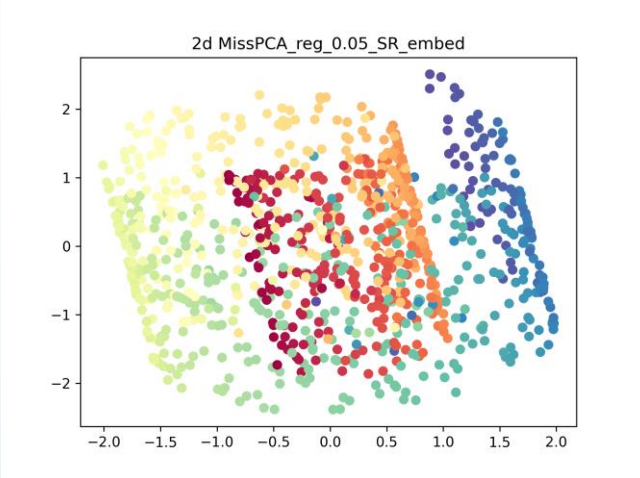
$$\widehat{X_{i,k}^{(t)}} = \widehat{\mu^{(t)}}_k + \sum_{l=1}^{d} \left(1 - \frac{\widehat{\sigma^2}}{\widehat{\lambda}_l}\right) \widehat{y^{(t)}}_{i,l} \widehat{U^{(t)}}_{k,l} \, , \qquad \widehat{\sigma^2} := \frac{1}{D-d} \sum_{l=d+1}^{D} \widehat{\lambda}_l$$

Regularization comes down to shrinking the coordinates of the individuals towards the origin.

Analysis of datasets - Swissroll

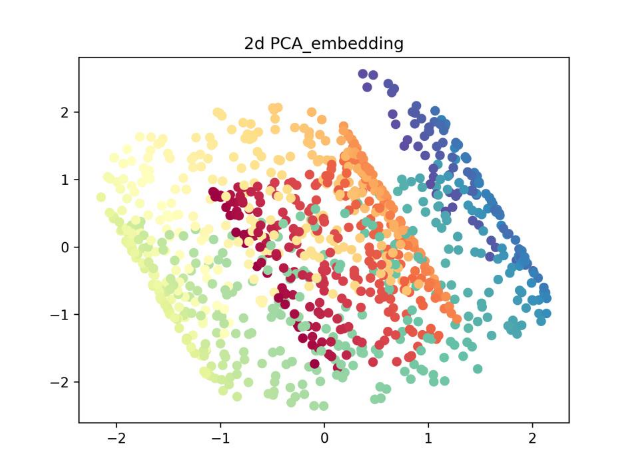




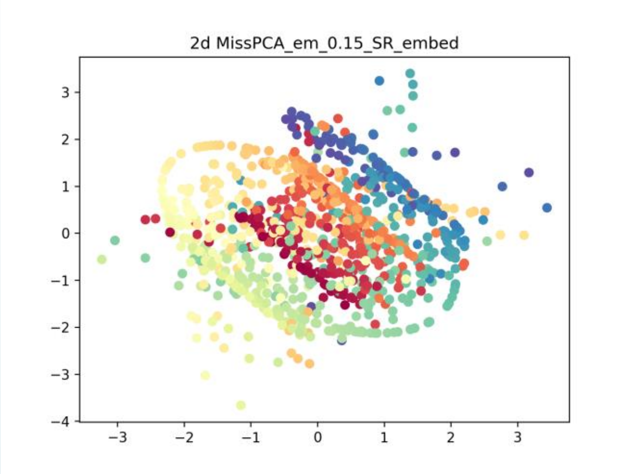


Area under the R_{NX} curve ≈ 0.52; Area under the R_{NX} curve ≈ 0.43; Area under the R_{NX} curve ≈ 0.43;

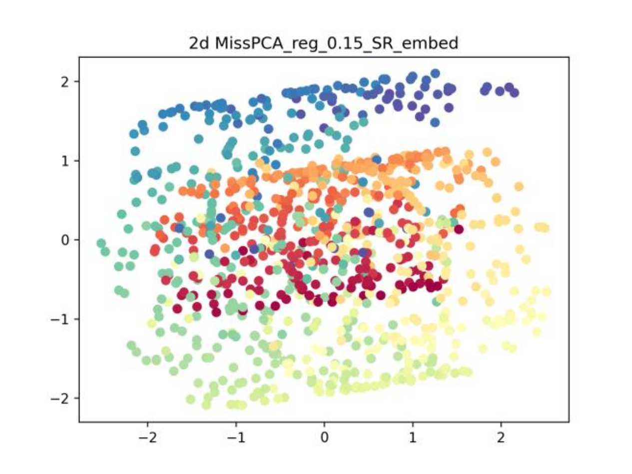
Analysis of datasets - Swissroll



Area under the R_{NX} curve ≈ 0.52;

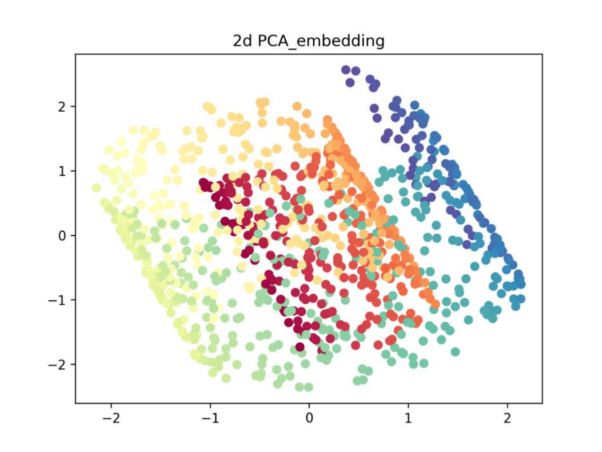


Area under the R_{NX} curve ≈ 0.30;

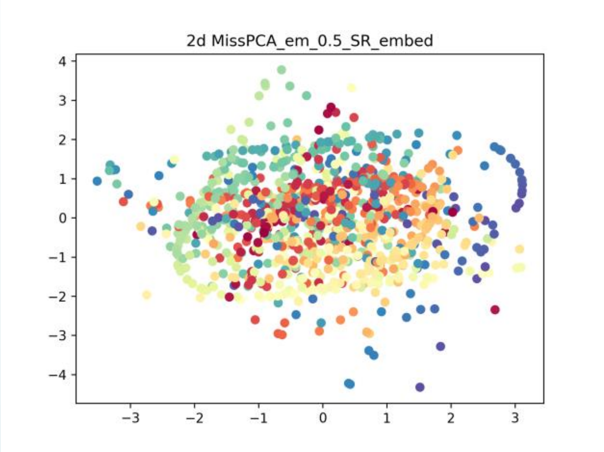


Area under the R_{NX} curve ≈ 0.33;

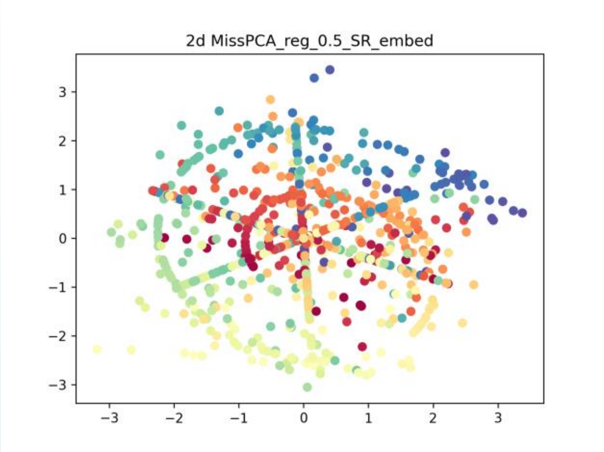
Analysis of datasets - Swissroll



Area under the R_{NX} curve ≈ 0.52;



Area under the R_{NX} curve ≈ 0.10;



Area under the R_{NX} curve ≈ 0.13

The NIPALS algorithm

Nonlinear **I**terative **P**artial **L**east **S**quares (NIPALS)

- Uses alternating least squares method
- Two weighted simple linear regressions are alternated to receive the first principal component
- Following dimensions are obtained by applying the same method to the residual matrix
- Can handle a small percentage of missing data (MAR / MCAR) by skipping those elements in the estimation process

Drawbacks:

(-) unstable estimates with large variability

(-) not minimizing some explicit criterion

(-) can not perform standardized PCA with missing values

The NIPALS algorithm

Find the principal components through the decomposition $X = YU^T$ based on the linear regression model: x $= yu^T + \varepsilon$

(1) Set $\mathrm{h} = 1$ and $X_h = X$

(2) Choose y_h as any column of X_h

(3) Iterate:

a) Compute loadings $u_h = \frac{X_h^T y_h}{y_h^T y_h}$ (projection of X on y)

b) Let $u_h = \frac{u_h}{\sqrt{u_h^T u_h}}$ (scaling)

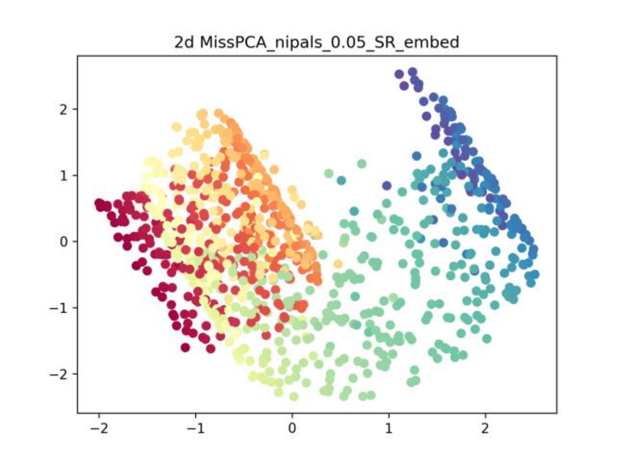
c) Compute scores $y_h = \frac{X_h u_h}{u_h^T u_h}$ (projection of X on u)

(4) Set $X_{h+1} = X_h - y_h u_h^T$ and $h = h + 1$

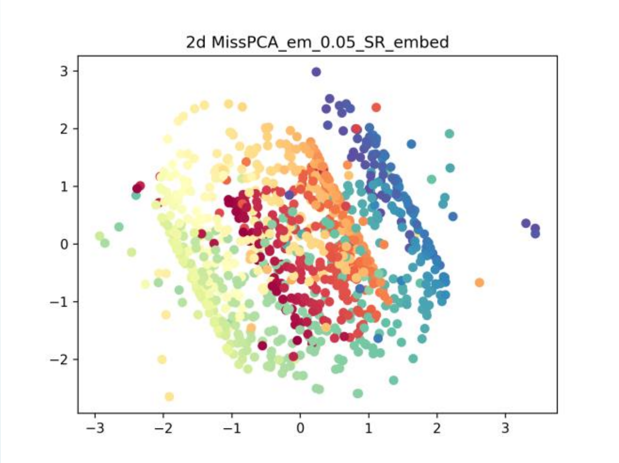
In the incomplete data case the corresponding missing elements y_{hi} and u_{hk} must be skipped in the calculation of the loadings and scores respectively!

Can result in convergence problems if the number of missing values increases.

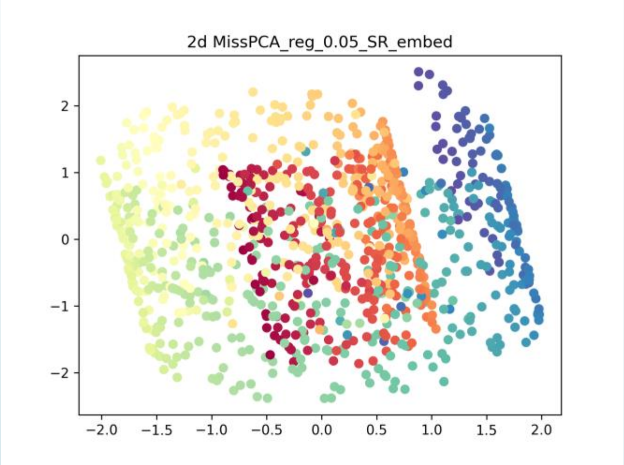
NIPALS vs Iterative PCA



Area under the R_{NX} curve ≈ 0.41;



Area under the R_{NX} curve ≈ 0.43;



Area under the R_{NX} curve ≈ 0.43;

R-packages for practical implementations

- **'missMDA' – package [JH16]:**
  - function 'imputePCA': Impute the missing entries of a mixed data using the iterative PCA algorithm (method="EM") or the regularized iterative PCA algorithm (method="Regularized"). Outputs the observed data matrix and the imputed data matrix

- **'pcaMethods' – package [WH07]:**
  - function 'pca': Performs standard PCA as well as PCA with missing data. Includes a variety of PCA methods:

    (1) "Classical" PCA via SVD

    (2) NIPALS

    (3) Bayesian PCA: An iterative method using a Bayesian model to handle missing values

    (4) PPCA: An iterative method using a probabilistic model to handle missing values

Nonlinear Extensions

Nonlinear PCA (NPCA)

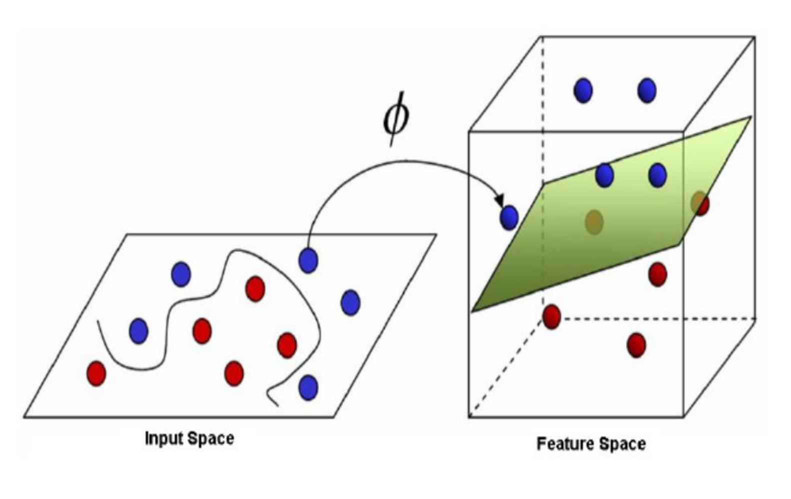
If the inherent structure of a given dataset lies not in or close to a linear or affine subspace of $\mathbb{R}^D$ the PCA method will not be able to project to a low-dimensional subspace which captures the structure of the data in a satisfactory manner.

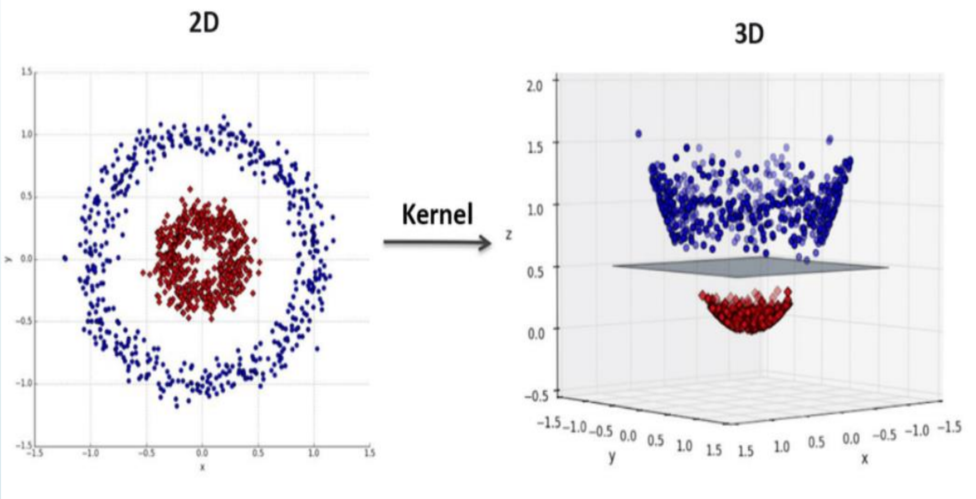
Idea:

It exists a nonlinear mapping $\phi: \mathbb{R}^D \longrightarrow \mathcal{H}$ into a higher-dimensional space $\mathcal{H}$ in such a way that the embedded data lies in a linear manifold / affine subspace of $\mathcal{H}$.

So instead of applying the PCA method directly in the input space one first maps the data into the so called feature space $\mathcal{H}$ and performs PCA in the feature space in a second step.

Nonlinear PCA





https://medium.com/@KunduSourodip/finding-non-linear-decision-boundary-in-svm-a89a97a006d2

https://www.researchgate.net/figure/Non-linear-classifier-using-Kernel-trick-16_fig4_340610860

Nonlinear PCA

<u>Mapping example:</u>

Given a set of points $(x_1, x_2) \in \mathbb{R}^2$ lying in a conic of the form

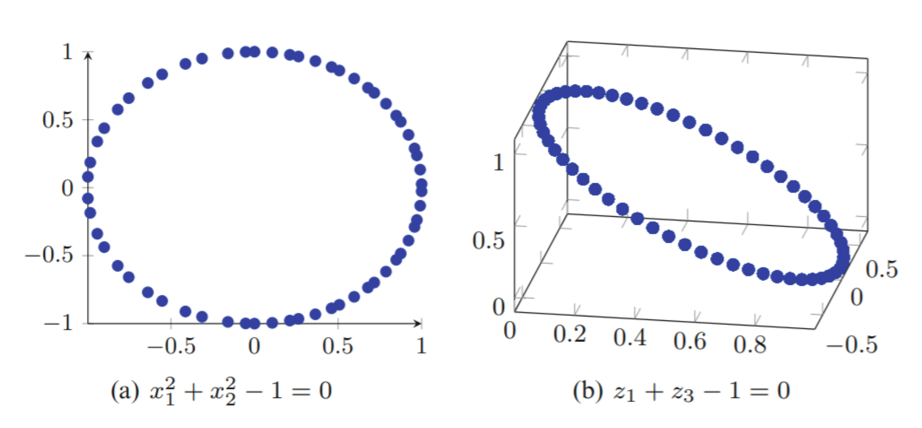
$$ax_1^2 + bx_1x_2 + cx_2^2 + d = 0, \qquad a, b, c, d \in \mathbb{R}$$

Define $\phi: \mathbb{R}^2 \to \mathbb{R}^3$

$$\phi(x_1, x_2) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right) = (z_1, z_2, z_3)$$

Then the conic in $\mathbb{R}^2$ transforms into an affine subspace in $\mathbb{R}^3$:

$$az_1 + \frac{b}{\sqrt{2}}z_2 + cz_3 + d = 0, \qquad a, b, c, d \in \mathbb{R}$$



[VI16, p. 127]

Nonlinear PCA

Recap:

$$\boldsymbol{\phi}\colon \mathbb{R}^{\boldsymbol{D}} \longrightarrow \mathbb{R}^{\boldsymbol{M}} \qquad \boldsymbol{\Sigma}_{\boldsymbol{\phi}(\boldsymbol{x})}\boldsymbol{u}_{\boldsymbol{i}} = \boldsymbol{\lambda}_{\boldsymbol{i}}\boldsymbol{u}_{\boldsymbol{i}}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\phi}(\boldsymbol{x})} = \sum_{\boldsymbol{j}=1}^{\boldsymbol{N}} \left(\boldsymbol{\phi}(\boldsymbol{x}_{\boldsymbol{j}}) - \overline{\boldsymbol{\phi}}\right)\left(\boldsymbol{\phi}(\boldsymbol{x}_{\boldsymbol{j}}) - \overline{\boldsymbol{\phi}}\right)^{\boldsymbol{T}} \stackrel{\text{def}}{=} \boldsymbol{\Phi}\boldsymbol{\Phi}^{\boldsymbol{T}}$$

Challenges:

- How to find an appropriate mapping such that the embedded data becomes approximately linear?
- The dimension of the feature space of an already high dimensional dataset can become enormous such that computations become costly if not unfeasible!

Kernel PCA (KPCA) – Kernel Trick

Computation of the nonlinear principal components relies only on inner products of the features.

Could there be an efficient "shortcut" computation:

$$
\begin{tikzcd}
\boldsymbol{x_i} \arrow[r] & \boldsymbol{\phi(x_i)} \arrow[dr] & \\
& & \boldsymbol{\phi(x_i)^T\phi(x_j)} \\
\boldsymbol{x_j} \arrow[r] & \boldsymbol{\phi(x_j)} \arrow[ur] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
\boldsymbol{x_i} \arrow[dr] & \\
& \boldsymbol{k(x_i\,,x_j)} \\
\boldsymbol{x_j} \arrow[ur] &
\end{tikzcd}
$$

Kernel PCA – Mercer's Theorem and Kernels

Definition (kernel function)

Let $\phi: \mathbb{R}^D \longrightarrow \mathbb{R}^M$ be an embedding function. The kernel function $k: \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$

Of two vectors $x_1, x_2 \in \mathbb{R}^D$ is defined to be the inner product of their features.

Theorem

Mercer's theorem states, that every continuous kernel function on a space $\mathcal{X}$ with the following property's:

(1) Symmetry: $k(x_i, x_j) = k(x_j, x_i) \quad \forall\, x_i, x_j \in \mathcal{X}$

(2) Positive (semi-) definiteness: For each finite subset of data points $\{x_1, \dots, x_n\}$ the kernel matrix $K \in \mathbb{R}^{n \times n}$ with $K_{i,j} := k(x_i, x_j)$ is positive semi-definite can always be associated with an embedding function ϕ.

Kernel PCA – Kernel functions

- Every non-negative constant function is a kernel
- Linear kernel: $k(x_i, x_j) = {x_i}^T x_j$ => PCA results as special case of KPCA
- Polynomial kernel : $k(x_i, x_j) = ({x_i}^T x_j + c)^n$, $c \geq 0,\ n \in \mathbb{N}$
- Gaussian (RBF) kernel: $k(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right)$
- Hyperbolic tangent kernel: $k(x_i, x_j) = tanh(\alpha {x_i}^T x_j + c),\ \alpha, c \in \mathbb{R}$
- (…)

Kernel PCA – SUM AND PRODUCT KERNELS

Given kernels k_1, k_2 it applies:

(1) $\alpha * k_1, \alpha > 0$ is a kernel

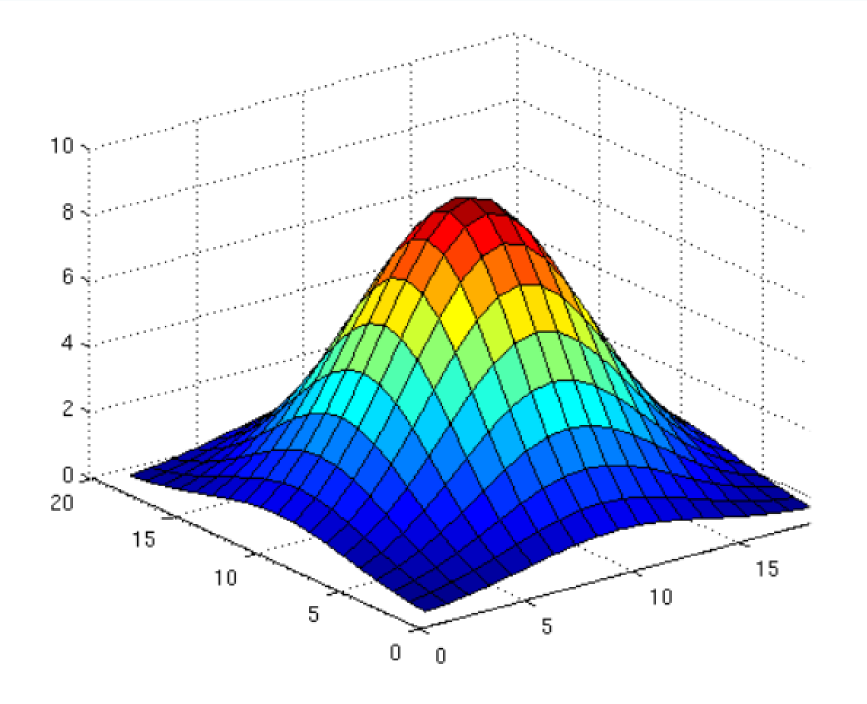
(2) $k_1 + k_2$ is a kernel

(3) $k_1 * k_2$ is a kernel

(4) $(k_1)^n, n \in \mathbb{N}$ is a kernel

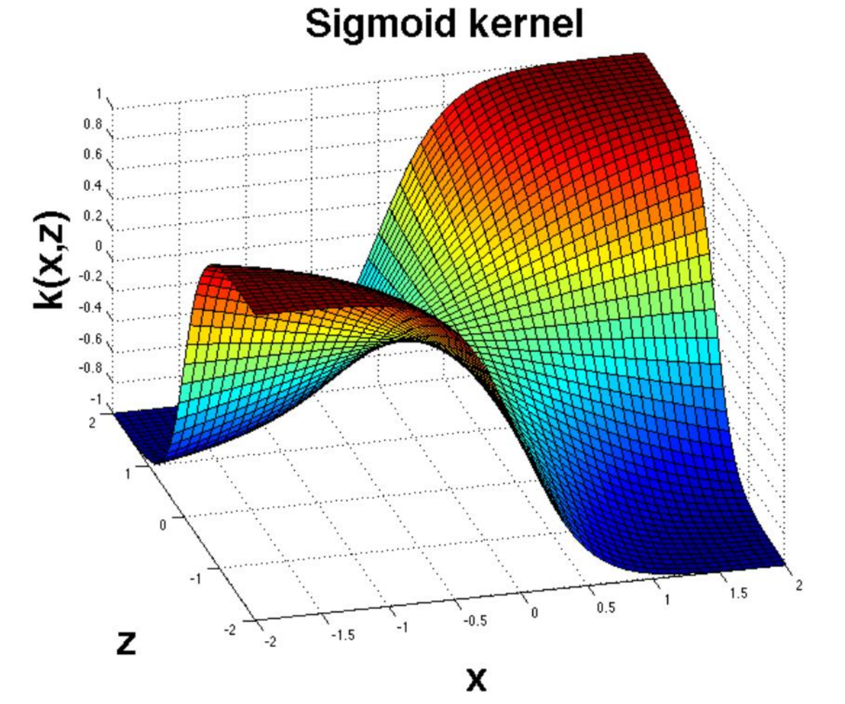
Kernel PCA – Kernel functions

Gaussian kernel function



https://stackoverflow.com/questions/12606048/2d-3d-plot-of-image-processing-filters

Sigmoid kernel function



https://datascience.stackexchange.com/questions/10479/on-the-properties-of-hyperbolic-tangent-kernel

Nonlinear PCA - Algorithm

Input: A set of points $\{x_1, \dots, x_N\} \subset \mathbb{R}^D$, and a map $\phi: \mathbb{R}^D \longrightarrow \mathbb{R}^M$ or a symmetric positive (semi-) definite kernel function $k: \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$.

(1) Compute the centered embedded data matrix Φ or the centered kernel $\tilde{k}$

(2) Compute the centered kernel matrix

$$\tilde{K} = \Phi^T \Phi \text{ or } \tilde{k}(x_i, x_j) \in \mathbb{R}^{N \times N}$$

(3) Compute the eigenvectors $u_i \in \mathbb{R}^N$:

$$\tilde{K} u_i = \lambda_i u_i$$

(4) For every data point x, its i_{th} nonlinear principal component is given by:

$$y_i = u_i^T \Phi^T (\phi(x) - \bar{\phi}) \text{ or } u_i^T \left[\tilde{k}(x_1, x), \dots, \tilde{k}(x_N, x)\right]^T, i = 1, \dots, d$$

Output: A set of points $\{y_1, \dots, y_N\} \subset \mathbb{R}^d$, where $y_{i,j}$ is the i_{th} nonlinear principal component of x_j.

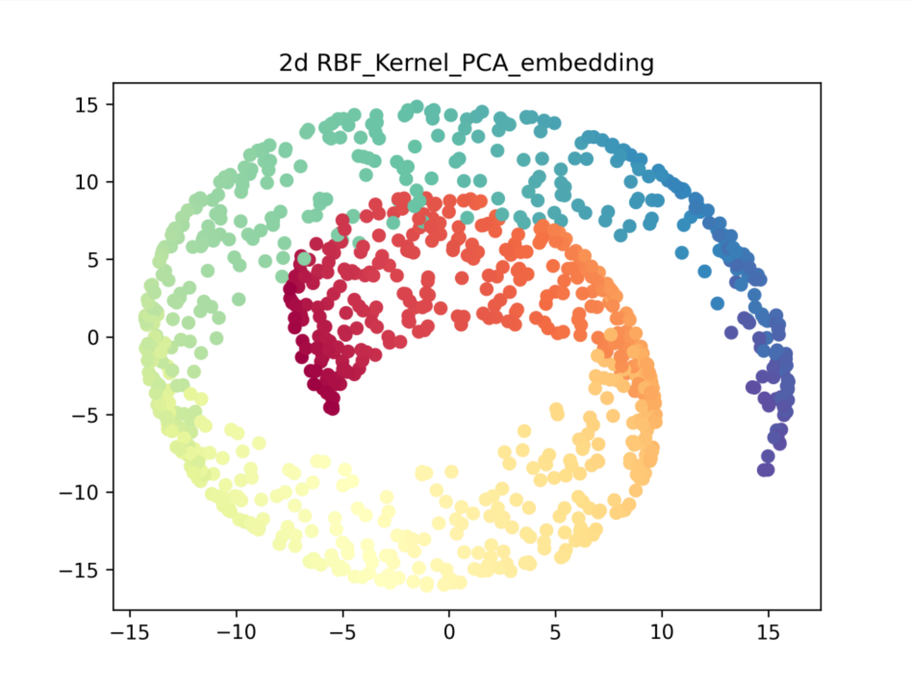
R-packages for practical implementations

- **‘kernlab’ – package [KASH04]:**
  - function ‘kpca’: Compute's kernel pca with a broad range of kernel functions provided or a user defined function.

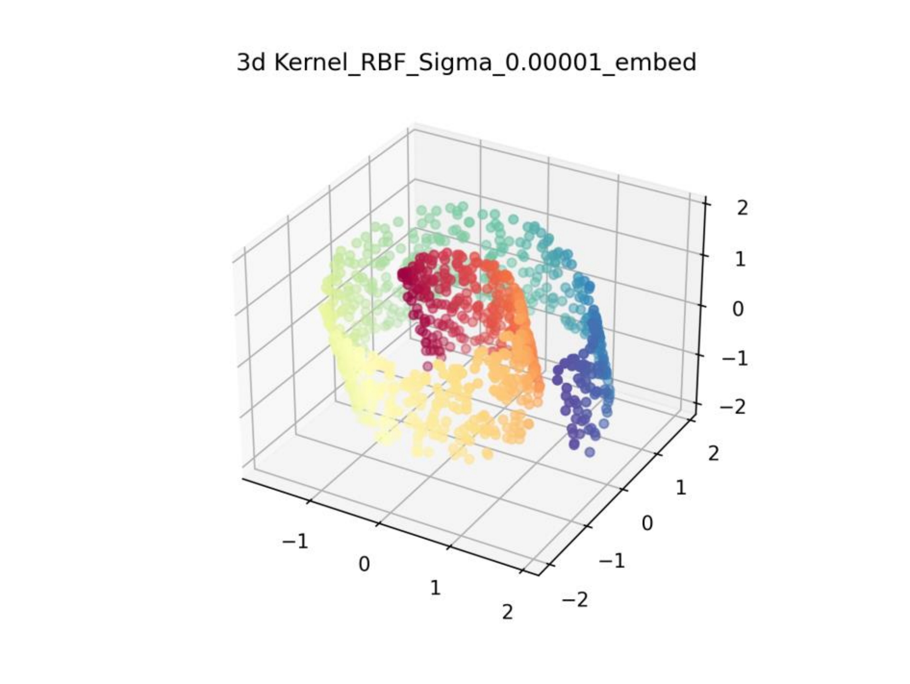
    Outputs an S4 object containing the principal component vectors along with the corresponding eigenvalues.

Analysis of datasets - Swissroll

Area under the R_{NX} curve ≈ 0.56

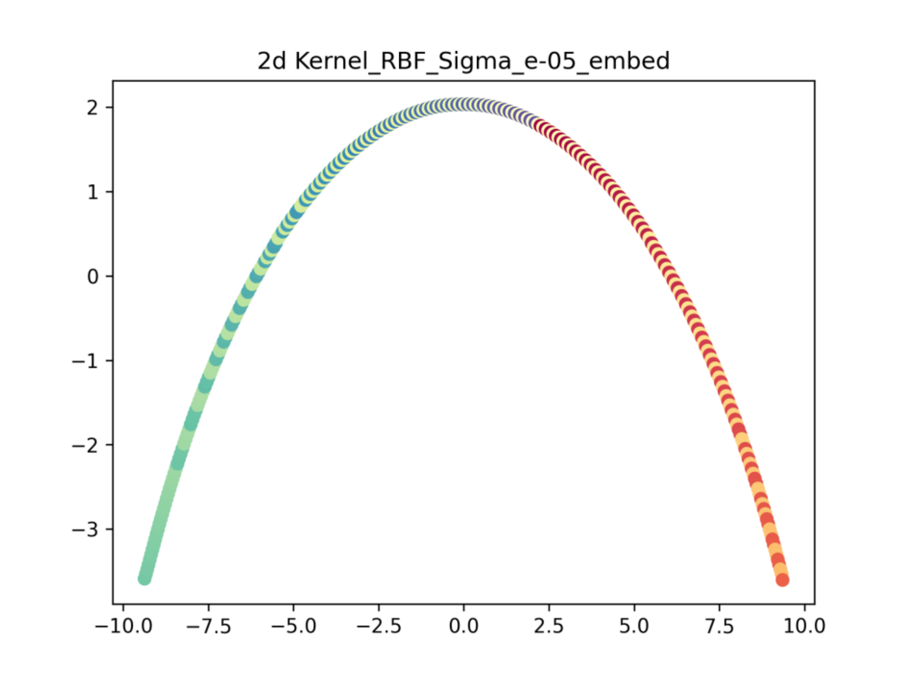


Area under the R_{NX} curve ≈ 0.999

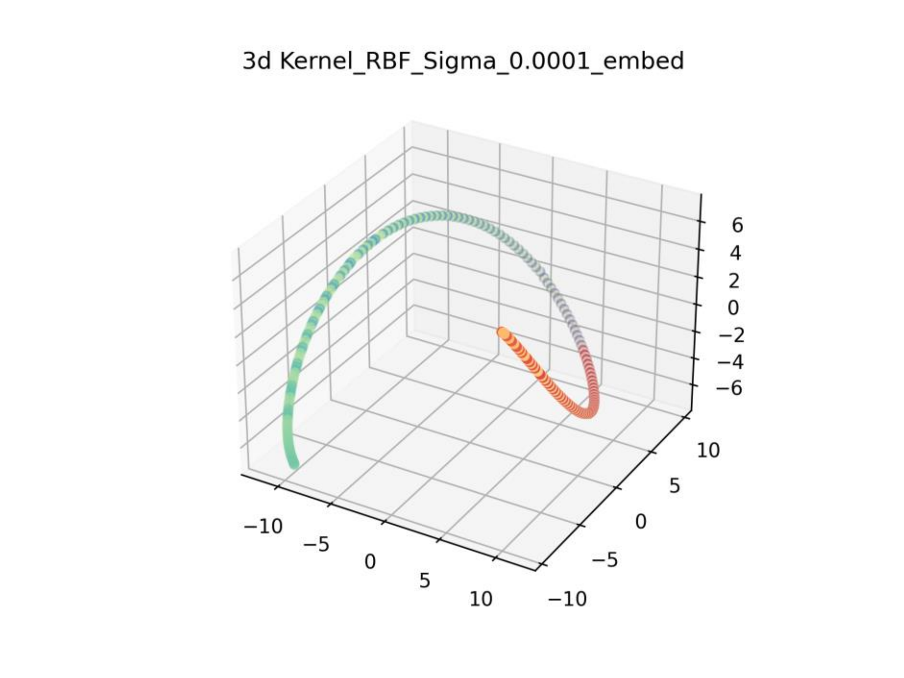


Analysis of datasets - Clock

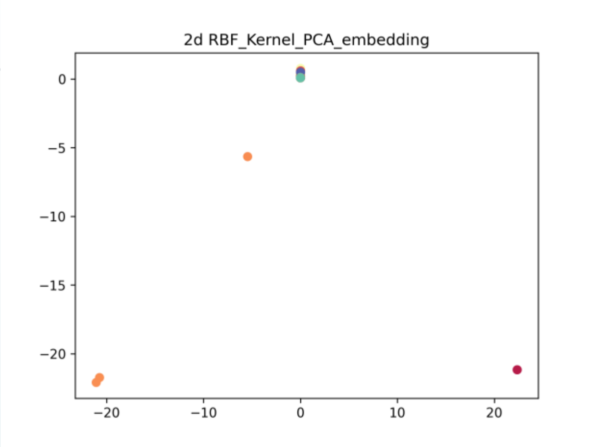
Area under the R_{NX} curve ≈ 0.93

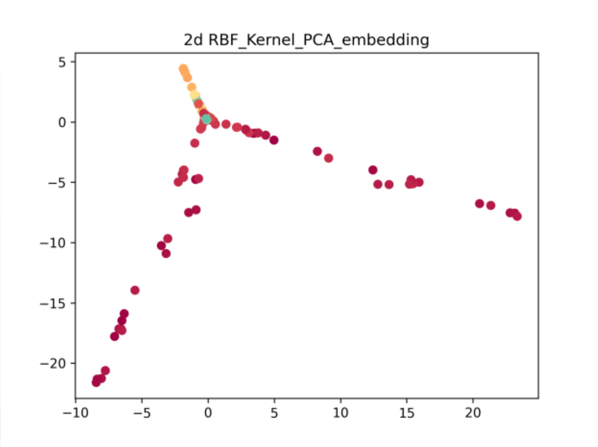


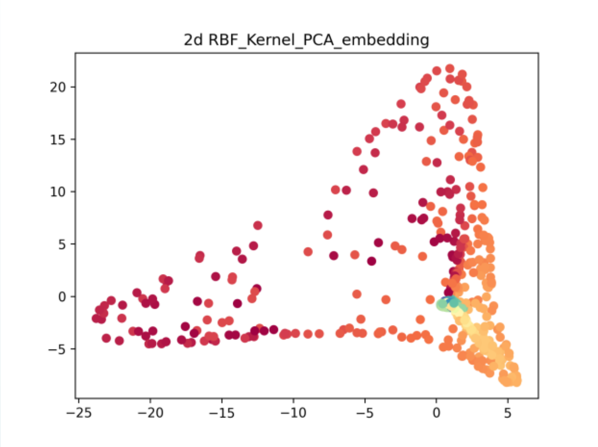
Area under the R_{NX} curve ≈ 0.91

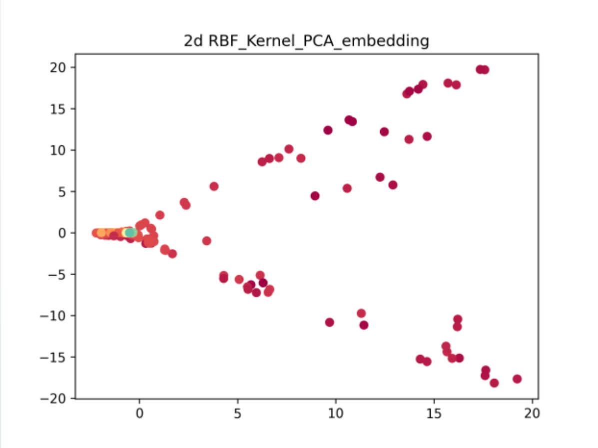


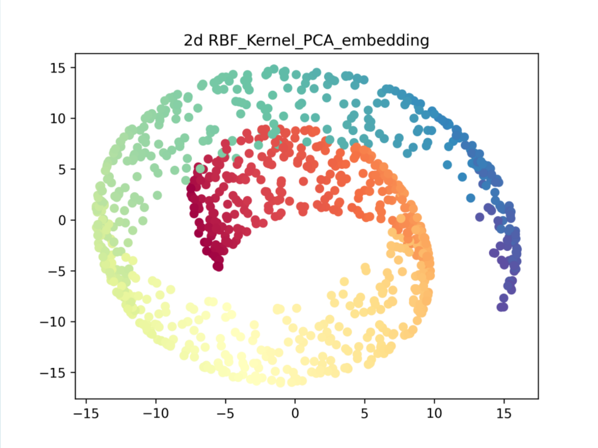
Parameter tuning – Gaussian kernel

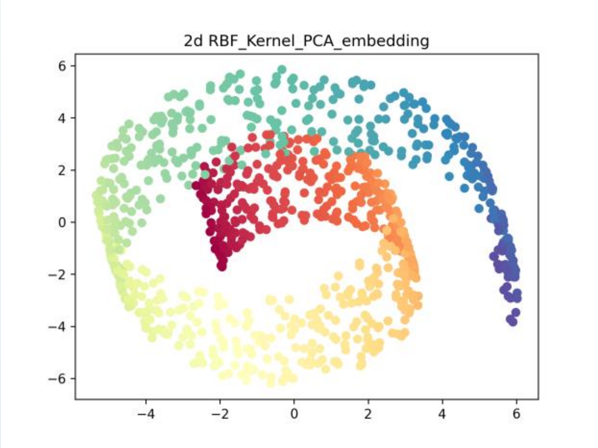












Conclusion

- PCA can capture the structure of a low-dimensional embedding as long as there exists a linear subspace and therefore exists mainly linear correlation between variables
- Iterative algorithms can help to perform PCA in the incomplete data case
- Results should always be compared to PCA results on the observed values
- The higher the degree of missingness the higher the uncertainty in the estimated parameters
- Nonlinear extensions like KPCA can help to capture the nonlinear structure of an low-dimensional embedding
- Choice of kernel functions and parameter tuning can be a difficult task and different results should always be compared
- Are there extensions for nonlinear PCA in the incomplete data case?

R - packages

[JH16] Julie Josse, Francois Husson (2016). missMDA: A Package for Handling Missing Values in Multivariate Data Analysis. Journal of Statistical Software, 70(1), 1-31. doi:10.18637/jss.v070.i01

[KASH04] Alexandros Karatzoglou, Alex Smola, Kurt Hornik, Achim Zeileis (2004). kernlab - An S4 Package for Kernel Methods in R. Journal of Statistical Software 11(9), 1-20. URL http://www.jstatsoft.org/v11/i09/

[LJH08] Sebastien Le, Julie Josse, Francois Husson (2008). FactoMineR: An R Package for Multivariate Analysis. Journal of Statistical Software, 25(1), 1-18. 10.18637/jss.v025.i01

[RCore20] R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org

[WH07] Stacklies, W., Redestig, H., Scholz, M., Walther, D. and Selbig, J. pcaMethods -- a Bioconductor package providing PCA methods for incomplete data. Bioinformatics, 2007, 23, 1164-1167

Literature

[JH12] Josse, J. and Husson, F. (2012), Handling missing values in exploratory multivariate data analysis methods, Journal de la Société Française de Statistique, Vol. 153 No. 2 79-99

[KA14] Karpfinger, C. (2014), Höhere Mathematik in Rezepten, DOI 10.1007/978-3-642-37866-9, Berlin: Springer Verlag

[LE18] Lessig, C. (2018) Wissenschaftliches Rechnen Singulärwertzerlegung, available at: http://graphics.cs.uni-magdeburg.de/teaching/2018/wr/lectures/svd.pdf (Accessed 27/02/21)

[MF12] Yunqian, M. and Yun, F. (2012) Manifold Learning Theory and Applications, New York: CRC Press Taylor & Francis Group

[MM01] Martens, H. and Martens, M. (2001). Multivariate Analysis of Quality: An Introduction. J.Wiley & Sons

[VI16] Vidal, R. et al. (2016), Generalized Principal Component Analysis, Interdisciplinary Applied Mathematics 40, DOI 10.1007/978-0-387-87811-9_1, New York: Springer-Verlag

[WR17] Wright, K. (2017), The NIPALS algorithm, available at: https://cran.r-project.org/web/packages/nipals/vignettes/nipals_algorithm.html (Accessed 02/03/21)