4. Dimension Reduction Techniques

Statistical Modelling & Machine Learning

Jaejik Kim

Department of Statistics
Sungkyunkwan University

STA3036

Why Dimension Reduction?

- Visualization of data (unsupervised learning).
- To improve prediction, the reduction of the number of dimensions of input space is important.
  - The curse of dimensionality.
  - Irrelevant input variables make data pattern unclear.
- Methods to reduce the # of input dimensions:
  - Dimension reduction techniques.
  - Filtering input variables (removing irrelevant inputs).
- Why dimension reduction?
  - Data visualization.
  - Reduction of computing time for predictive model building.
  - Unsupervised dimension reduction techniques do not guarantee the improvement of predictive models.

Dimension Reduction Techniques

Unsupervised learning:

- Principal component analysis (PCA): Linear projection (i.e., $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{A}$, where $\boldsymbol{A}$ is the $p \times p$ projection matrix).
- Principal curve / principal surface: It directly finds one or two-dimensional nonlinear principal components.
- Kernel PCA: Kernel transformation of $\boldsymbol{X}$, and then PCA.
- Non-negative matrix factorization (NMF): It works only for non-negative data (e.g., image data).
- Independent component analysis (ICA): Latent variables that are statistically independent and non-Gaussian.

Supervised learning:

- Partial least squares: Directions maximizing the covariance with Y.

Principal Component Analysis

- Let $\boldsymbol{X}$ be $n \times p$ matrix with standardized variables.
- $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{V}$: Principal components.
  - $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p]$, where $\boldsymbol{v}_j$ is the eigenvector of $\boldsymbol{X}^\top \boldsymbol{X}$ corresponding to the jth largest eigenvalue.
- The singular value decomposition (SVD) of $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top.$$

  - $\boldsymbol{U}$ is a $n \times p$ orthogonal matrix with columns which span the column space of $\boldsymbol{X}$.
  - $\boldsymbol{V}$ is a $p \times p$ orthogonal matrix with columns which span the row space of $\boldsymbol{X}$.
  - $\boldsymbol{D}$ is a $p \times p$ diagonal matrix with diagonal entries $d_1 \geq d_2 \geq \cdots \geq d_q > d_{q+1} = \ldots = d_p = 0$ where $q = rank(\boldsymbol{X})$. The entries $d_1, \ldots, d_p$ are called the *singluar values* of $\boldsymbol{X}$.

Principal Component Analysis

- Eigen decomposition of $\boldsymbol{X}^\top \boldsymbol{X}$: By SVD,

$$\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top = \boldsymbol{V}\boldsymbol{D}^2\boldsymbol{V}^\top.$$

  - $\boldsymbol{V}$: The columns of $\boldsymbol{V}$ are eigenvectors of $\boldsymbol{X}^\top \boldsymbol{X}$.
  - Diagonal elements of $\boldsymbol{D}^2$: $d_1^2, \ldots, d_p^2$ are eigenvalues of $\boldsymbol{X}^\top \boldsymbol{X}$.

- PC: $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{U}\boldsymbol{D}$.

- PCA using rank-q linear model: $f(\boldsymbol{\lambda}) = \boldsymbol{V}_q \boldsymbol{\lambda}$,
  - $\boldsymbol{V}_q$: $p \times q$ orthogonal column matrix.
  - $\boldsymbol{\lambda}$: $q \times 1$ vector of parameters.
  - Fitting $f(\boldsymbol{\lambda})$ to the data $\boldsymbol{X}$ by minimizing the reconstruction error

$$\min_{\{\boldsymbol{\lambda}_i\}, \boldsymbol{V}_q} \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{V}_q \boldsymbol{\lambda}_i)^2.$$

  - For fixed $\boldsymbol{V}_q$, $\hat{\boldsymbol{\lambda}}_i = \boldsymbol{V}_q^\top \boldsymbol{x}_i$.

Principal Component Analysis

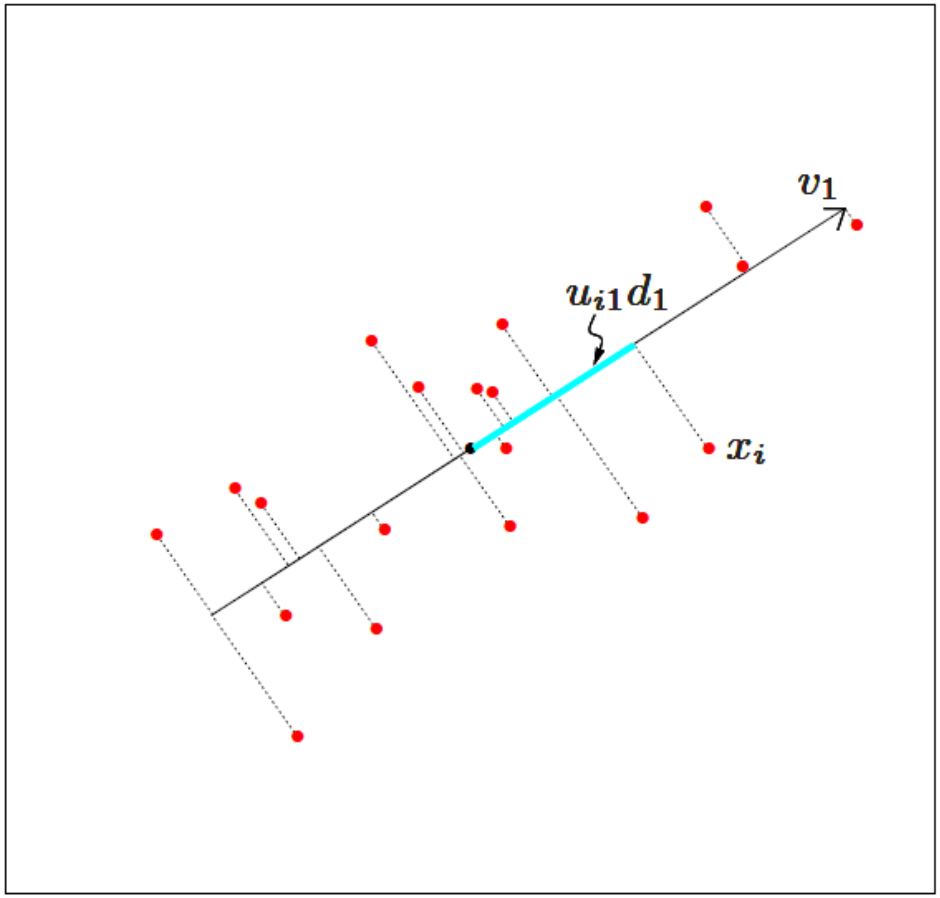
- Minimization of the reconstruction error becomes

$$\min_{\boldsymbol{V}_q} \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{V}_q \boldsymbol{V}_q^\top \boldsymbol{x}_i)^2.$$

- The solution of $\boldsymbol{V}_q$: The matrix with the first q columns of $\boldsymbol{V}$ from SVD.
- The first q PC's: $\boldsymbol{Z}_q = \boldsymbol{X}\boldsymbol{V}_q = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top \boldsymbol{V}_q = \boldsymbol{U}_q \boldsymbol{D}_q$.

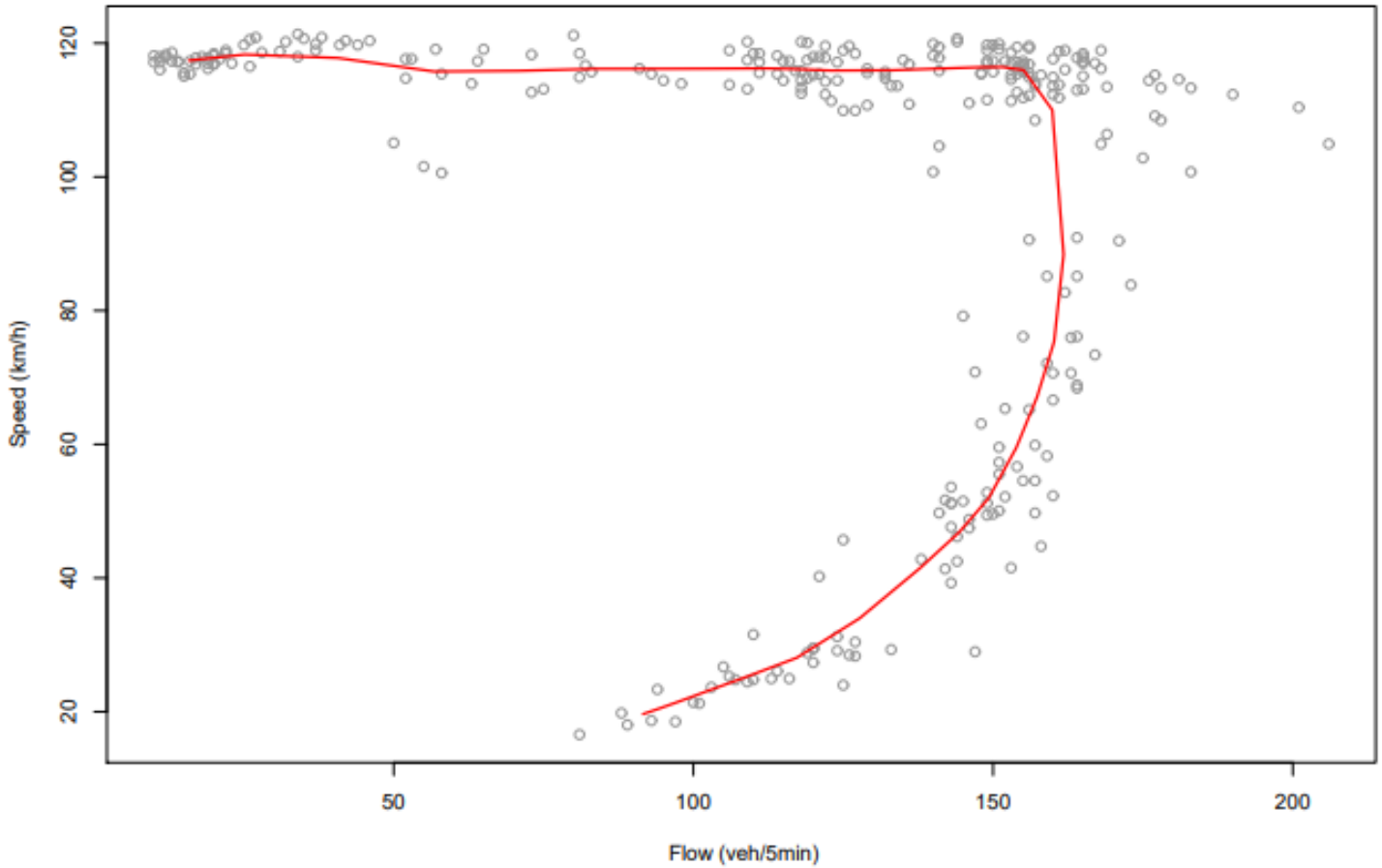
Principal Component Analysis

- The first PC line in $\boldsymbol{X} = (X_1, X_2)$ space:
  - The first PC direction $\boldsymbol{v}_1 = \boldsymbol{V}_1$ $(q = 1)$.
  - $\hat{\lambda}_i = u_{i1} d_1$: Distance along the PC line from the origin.



Principal Curves / Surfaces

Example: Speed-Flow data from a Californian 'freeway'.



Principal Curves

- Principal curves:
  - Smooth curves passing through the middle of the data cloud.
  - One-dimensional curved approximation to a set of data points in p-dimension.
  - $X_1, \ldots, X_p$ have nonlinear relationships.
  - Visualization of data.

- The principal curve for random variables $\boldsymbol{X} \in \mathbb{R}^p$: $\boldsymbol{f}(\lambda) = [f_1(\lambda), \ldots, f_p(\lambda)]$
  - A smooth curve in $\mathbb{R}^p$.
  - A vector function with p coordinates indexed by λ.
  - Each function $f_j(\lambda)$ is a coordinate function of a single parameter λ.
  - λ: The arc-length along the curve from some fixed origin.

Principal Curves

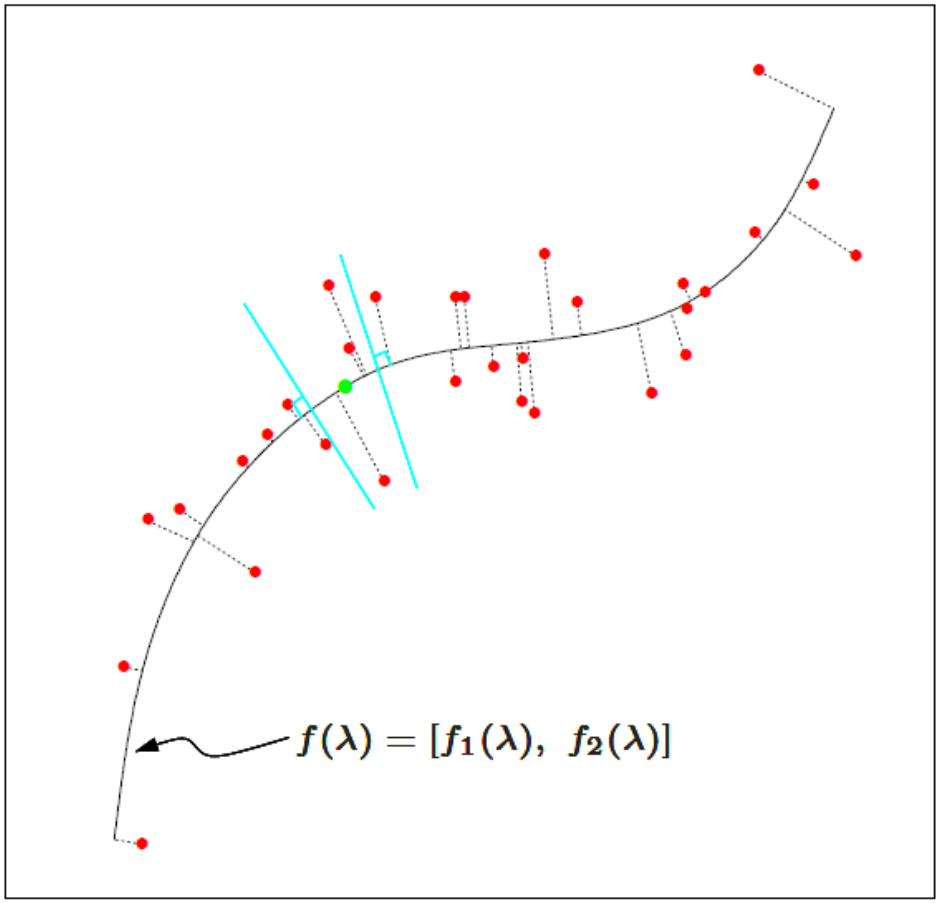


Figure: Each point on the curve is the average of all data points that project there.

Principal Curves

- Principal curve:

$$\boldsymbol{f}(\lambda) = E(\boldsymbol{X}|\lambda_{\boldsymbol{f}}(\boldsymbol{X}) = \lambda),$$

  - $\lambda_{\boldsymbol{f}}(\boldsymbol{X})$: The closest point on curve to a data point $\boldsymbol{x}$ (Euclidean).
  - $\boldsymbol{f}(\lambda)$: The average of all data points that project to $\boldsymbol{f}(\lambda)$ $\Rightarrow$ 'self-consistency' property.
  - In practice, for continuous multivariate distributed $\boldsymbol{X}$, principal curves are not unique.

Principal Curves

- To find a principal curve, we need to iteratively find $[f_1(\lambda), \ldots, f_p(\lambda)]$.

- Algorithm: For given data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$,
  (1) Initialize $f_1(\lambda), \ldots, f_p(\lambda)$ using the first principal component direction.
  (2) Initialize $\lambda_1, \ldots, \lambda_n$ using the first principal component values.
  (3) Update $\hat{f}_j(\lambda) \leftarrow \hat{E}[X_j | \hat{\lambda}_{\boldsymbol{f}}(\boldsymbol{X}) = \lambda]$ for $j = 1, \ldots, p$ using a smoother.
  (4) Update $\hat{\lambda}_{\boldsymbol{f}}(\boldsymbol{X}) \leftarrow \arg\max_\lambda \| \boldsymbol{X} - \hat{\boldsymbol{f}}(\lambda) \|$ by the projection on the curve.
  (5) Iterate (3) and (4) until convergence.

- Tuning parameter: df of smoother in step (3).

Principal Surfaces

- Let the surface be defined in $q < p$ dimensions.
- The most commonly used is 2D surface. When $q = 2$, $\boldsymbol{f}(\lambda_1, \lambda_2) = [\boldsymbol{f}_1(\lambda_1, \lambda_2), \ldots, \boldsymbol{f}_p(\lambda_1, \lambda_2)]$.
- Initial using plane defined by the first two principal component directions.
- Use 2D scatterplot smoother in algorithm to estimate $f_j(\lambda_1, \lambda_2),\ j = 1, \ldots, p$.
- Data points can be plotted by estimated nonlinear coordinates $(\hat{\lambda}_1(\boldsymbol{x}_i), \hat{\lambda}_2(\boldsymbol{x}_i))$.

Principal Surfaces

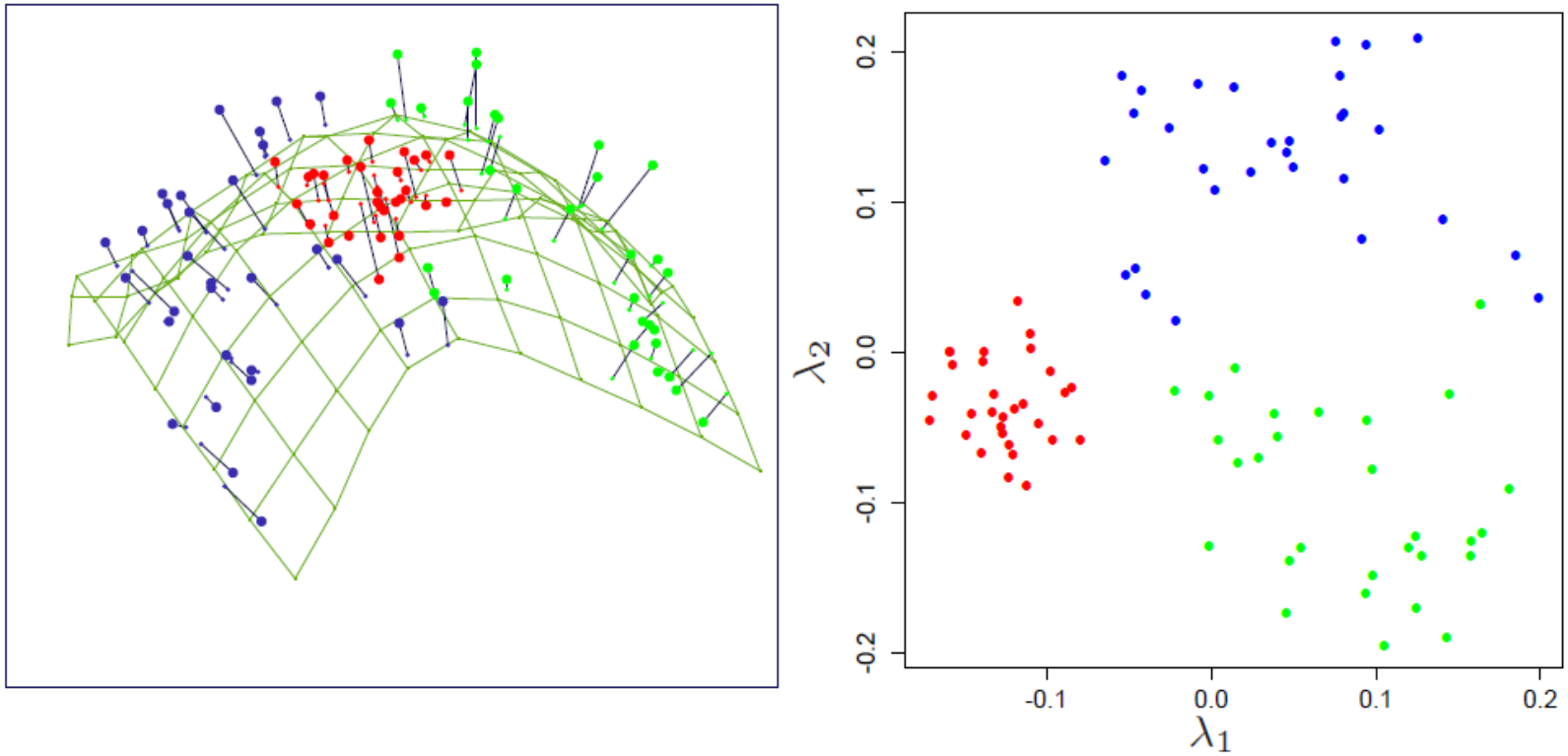


Figure: Left panel: Fitted two-dimensional surface; Right panel: Projections of data points onto the surface, resulting in coordinates $(\hat{\lambda}_1, \hat{\lambda}_2)$.

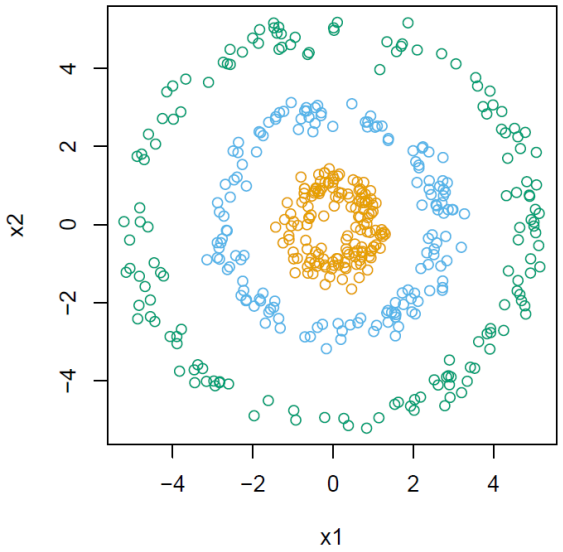
Kernel PCA

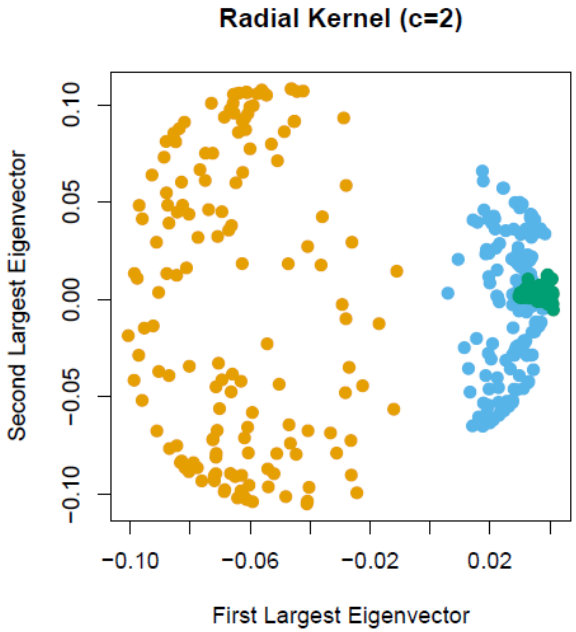
- Kernel PCA: Nonlinear version of PCA.
- Nonlinear transformation of inputs $\Rightarrow$ PCA for the transformed feature space.
- PCA:
  - Let $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^\top$, where $\boldsymbol{X}$ is the standardized input matrix, and let $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{D}$, where $\boldsymbol{Z}$ is principal component matrix.
  - $\boldsymbol{K} = \{\langle \boldsymbol{x}_i, \boldsymbol{x}_{i'} \rangle\}$: It consists of inner-product terms.
  - By SVD, $\boldsymbol{K} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\top = \boldsymbol{U}\boldsymbol{D}^2\boldsymbol{U}^\top \Rightarrow$ Eigen decomposition of $\boldsymbol{K}$.
  - This means that the principal component matrix can be obtained by the eigen decomposition of $\boldsymbol{K}$.
- Kernel PCA simply mimics this procedure of PCA.

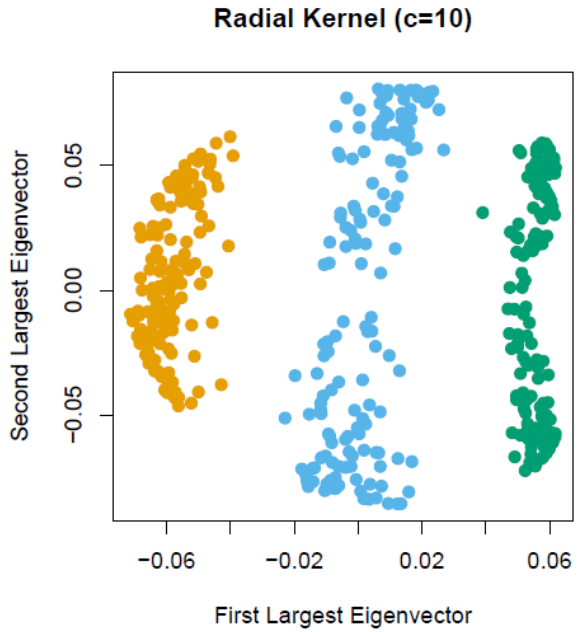
Kernel PCA

- Kernel matrix $\boldsymbol{K} = \{K(\boldsymbol{x}_i, \boldsymbol{x}_{i'})\}$: An inner-product matrix of the implicit features $\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_{i'}) \rangle$.
- From the eigen decomposition of $\boldsymbol{K}$, $\boldsymbol{U}$ and $\boldsymbol{D}$ can be obtained.
- The mth kenel PC: $\boldsymbol{z}_m =$ the mth column of $\boldsymbol{Z} = \boldsymbol{UD}$.
- $z_{im} = \sum_{j=1}^{n} \alpha_{jm} K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $\alpha_{jm} = u_{jm}/d_m$.
- We do NOT have to specify transformed feature $\phi(\boldsymbol{x})$. To construct $\boldsymbol{K}$, a kernel function should be chosen.
- Kernel functions:
  - Radial: $K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \exp(- \parallel \boldsymbol{x}_i - \boldsymbol{x}_{i'} \parallel^2 / c)$.
  - Polynomial: $K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = (1 + \boldsymbol{x}_i^\top \boldsymbol{x}_{i'})^d$.
  - Neural network: $K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \tanh(\boldsymbol{x}_i^\top \boldsymbol{x}_{i'} + \delta)$.

Kernel PCA


x2
x1
4
2
0
−2
−4
−4
−2
0
2
4


Radial Kernel (c=2)
Second Largest Eigenvector
First Largest Eigenvector
0.10
0.05
0.00
−0.05
−0.10
−0.10
−0.06
−0.02
0.02


Radial Kernel (c=10)
Second Largest Eigenvector
First Largest Eigenvector
0.05
0.00
−0.05
−0.06
−0.02
0.02
0.06

Non-negative Matrix Factorization (NMF)

- Alternative approach to PCA.
- Data and components are assumed to be non-negative (e.g., image data).
- The $n \times p$ input matrix $\boldsymbol{X}$ is approximated by

$$\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H},$$

  - $\boldsymbol{W}$: $n \times r$; $\boldsymbol{H}$: $r \times p$; $r \leq \max(n, p)$.
  - $x_{ij},\ w_{ik},\ h_{kj} \geq 0$.
- $\boldsymbol{W}$ and $\boldsymbol{H}$ can be found by maximizing

$$L(\boldsymbol{W}, \boldsymbol{H}) = \sum_{i=1}^{n} \sum_{j=1}^{p} \left[x_{ij} \log(\boldsymbol{W}\boldsymbol{H})_{ij} - (\boldsymbol{W}\boldsymbol{H})_{ij} \right].$$

  - This is the log-likelihood from a model in which $x_{ij} \sim$ Poisson with mean $(\boldsymbol{W}\boldsymbol{H})_{ij}$.

NMF

- Algorithm converging to a local maximum of $L(\boldsymbol{W}, \boldsymbol{H})$:
  (1) Initialize $\boldsymbol{W}$ & $\boldsymbol{H}$.
  (2) Update w_{ik} by

$$w_{ik} \leftarrow w_{ik} \frac{\sum_{j=1}^{p} h_{kj} x_{ij} / (\boldsymbol{WH})_{ij}}{\sum_{j=1}^{p} h_{kj}}.$$

  (3) Update h_{kj} by

$$h_{kj} \leftarrow h_{kj} \frac{\sum_{j=1}^{p} w_{ik} x_{ij} / (\boldsymbol{WH})_{ij}}{\sum_{j=1}^{p} w_{ik}}.$$

  (4) Iterate steps (2) & (3) until convergence.

- This algorithm is related to the iterative proportional scaling algorithm for log-linear models.

NMF

- NMF may not be unique. This means that the optimum value of the algorithm depends on initial values.
- The non-uniqueness could hamper the interpretability of the factorization.
- Approximation to the columns of $\boldsymbol{X} \Rightarrow$ Columns of $\boldsymbol{W}$ (primary non-negative components).
- Approximation to the rows of $\boldsymbol{X} \Rightarrow$ Rows of $\boldsymbol{H}$ (archetypal data points).

Independent Component Analysis (ICA)

- PC's are uncorrelated linear combinations of $X = (X_1, \ldots, X_p)$.
- Statistical independent components (linear combinations of X $\Rightarrow$ Independent component analysis.
- ICA problem:

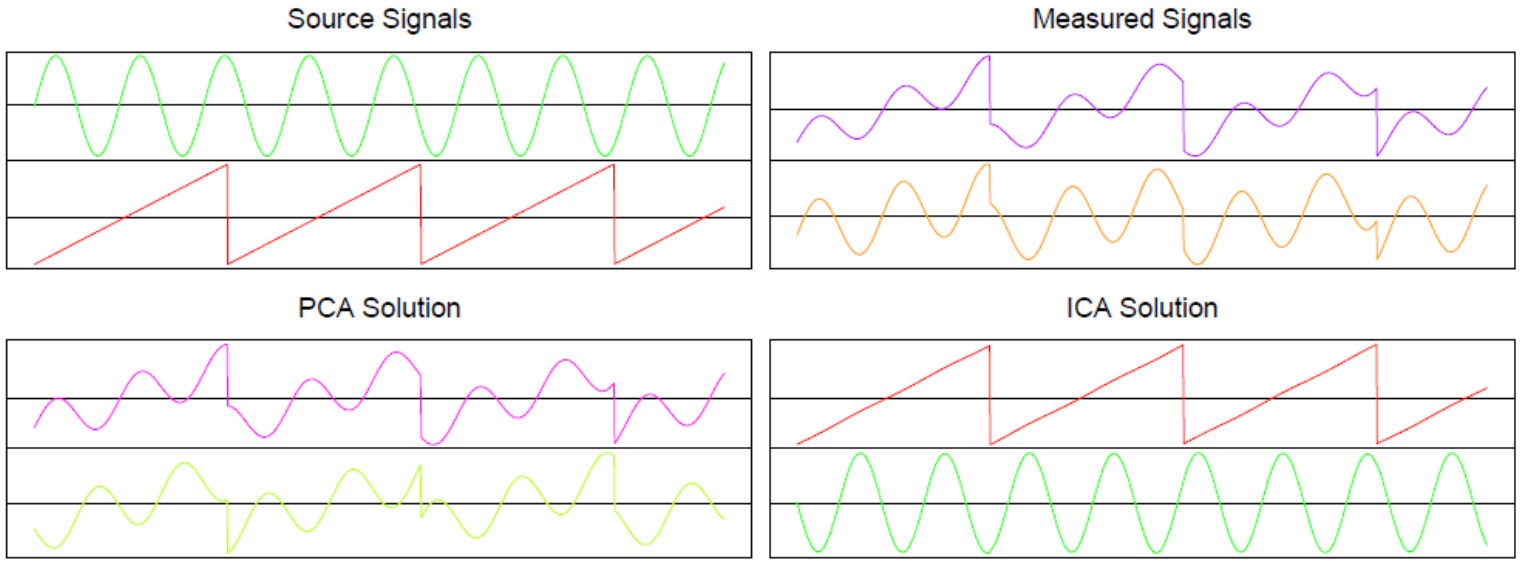
$$X = \boldsymbol{A}S,$$

  - $X = (X_1, \ldots, X_p)^\top$: A $p \times 1$ random vector representing multivariate input measurements.
  - $S = (S_1, \ldots, S_p)^\top$: A $p \times 1$ latent source vector. $S_1, \ldots, S_p$ are independent random variables.
  - $\boldsymbol{A}$: $p \times p$ mixing matrix.

ICA

- The goal of ICA:
  - Estimation of $\boldsymbol{A}$.
  - Estimation the source distributions $S_j \sim f_{S_j},\ j = 1, \ldots, p$.

- Cocktail party problem:
  - p independent sources of sound in a room.
  - p microphones placed around the room hear different mixtures.
  - Recovery of the sources.



ICA

- Assume that $E(S) = \mathbf{0}$ and $Cov(S) = \boldsymbol{I}$.

$$Cov(X) = Cov(\boldsymbol{A}S) = \boldsymbol{A}Cov(S)\boldsymbol{A}^\top = \boldsymbol{A}\boldsymbol{A}^\top.$$

- $X = \boldsymbol{A}S$: Each correlated X_j is represented as a linear expansion in the uncorrelated, unit variance variables S_j.
- For any given orthogonal $p \times p$ matrix $\boldsymbol{R}$,

$$X = \boldsymbol{A}S = \boldsymbol{A}\boldsymbol{R}^\top \boldsymbol{R}S = \boldsymbol{A}^* S^*,$$

  - $Cov(S^*) = \boldsymbol{R}Cov(S)\boldsymbol{R}^\top = \boldsymbol{I}$.
  - Non-uniqueness of $\boldsymbol{A}$ and S under uncorrelated and Gaussian assumptions.
  - Impossible to identify any particular latent source variables as unique underlying sources.

ICA

- To recover $\boldsymbol{A}$, ICA assumes statistical independence and non-Gaussianity of S.
- Under the uncorrelatedness and Gaussian assumptions, the second order moment (covariance) of random variables is required.
- Independence and non-Gaussianity require higher order moments.
- Like PC, Independent components are linear combinations of X. However, the linear combinations of X should be chosen to be as independent as possible.

ICA

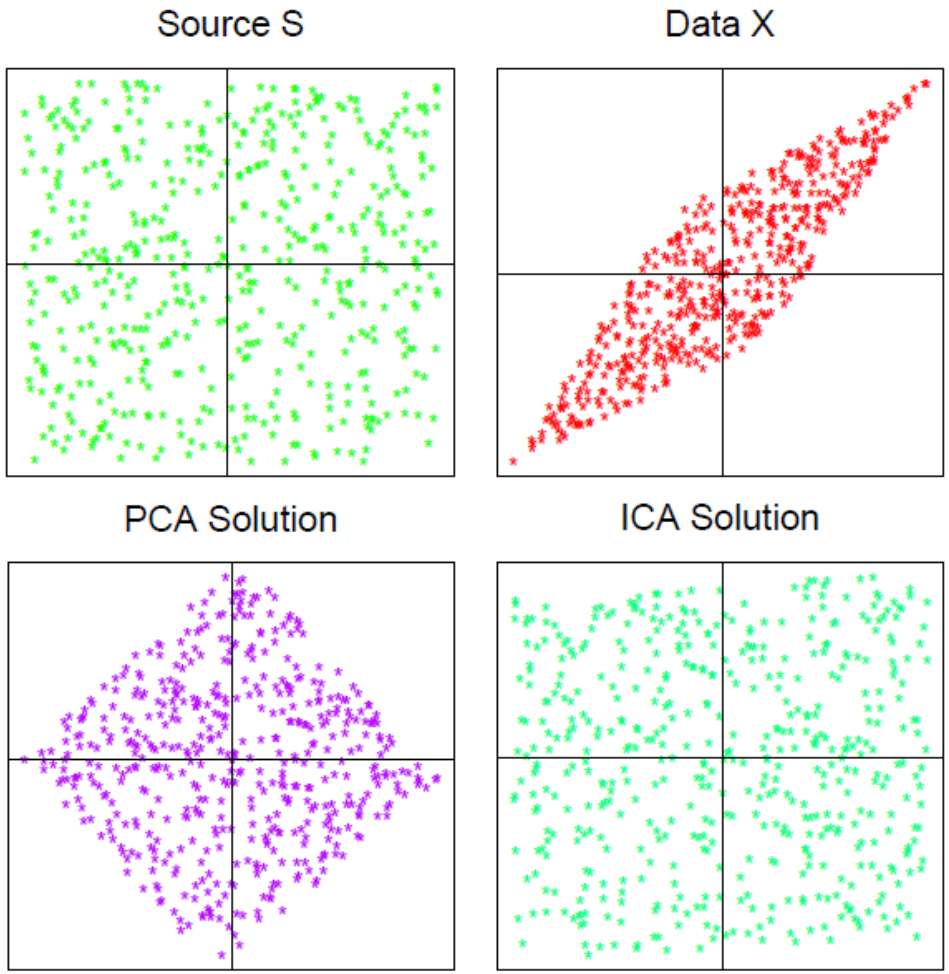


Figure: Source $S = (S_1, S_2)^\top$ is two independent uniform random variables. Data are generated by $X = \boldsymbol{A}S$.

ICA

- There are a lot of methods to estimate $\boldsymbol{A}$ & S in ICA.
- The most methods start with this:
  - Let $\boldsymbol{\Sigma} = Cov(X)$ and $X = \boldsymbol{A}S$. Then, $S = \boldsymbol{A}^{-1}X$.
  - Let $\boldsymbol{A}^{-1} = \boldsymbol{W}\boldsymbol{\Sigma}^{-\frac{1}{2}}$ for some non-singular $\boldsymbol{W}$. Then, $S = \boldsymbol{A}^{-1}X = \boldsymbol{W}\boldsymbol{\Sigma}^{-\frac{1}{2}}X$ with $Cov(S) = \boldsymbol{I}$ and orthonormal $\boldsymbol{W}$.
  - To find S, transform the data $\tilde{X} = \boldsymbol{\Sigma}^{-\frac{1}{2}}X$, and then seek an orthogonal matrix $\boldsymbol{W}$ so that the components $S = \boldsymbol{W}\tilde{X}$ are independent.

ICA

Methods to find the orthonormal matrix $\boldsymbol{W}$:

- Mutual information and entropy, maximizing non-Gaussianity.
  - FastICA, Infomax methods.
  - Kullback-Leibler divergence: A measure of dependence between two random vectors.
  - Using the K-L divergence, it looks for low-entropy or non-Gaussian projections $\boldsymbol{W}$.
- Likelihood methods:
  - ProdDenICA.
  - It directly fits the independent density model $f_S(s) = \prod_{j=1}^{p} f_j(s_j)$ to find $\boldsymbol{W}$.
- etc.