Data Analysis and Manifold Learning
Lecture 12: Some Examples of Manifold Learning Applications

Radu Horaud
INRIA Grenoble Rhone-Alpes, France
Radu.Horaud@inrialpes.fr
http://perception.inrialpes.fr/

Outline of Lecture 12

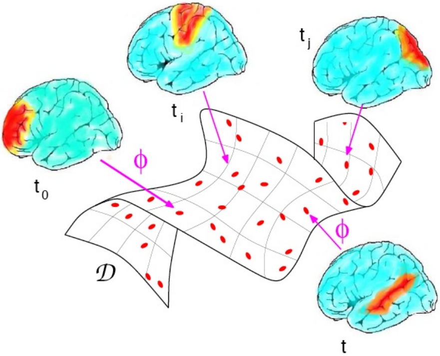
- Spectral embedding of a graph applied to fMRI data analysis
- The Google PageRank algorithm

Material for this lecture

- X. Shen and F. Meyer. Low-Dimensional Embedding of fMRI Datasets. NeuroImage 41 (2008). http://ecee.colorado.edu/~fmeyer/Pub/neuroimage08.pdf
- T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. Springer (2009). The Google PageRank algorithm is briefly described 576–578.
- Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report. Stanford University (1999). http://ilpubs.stanford.edu:8090/422/

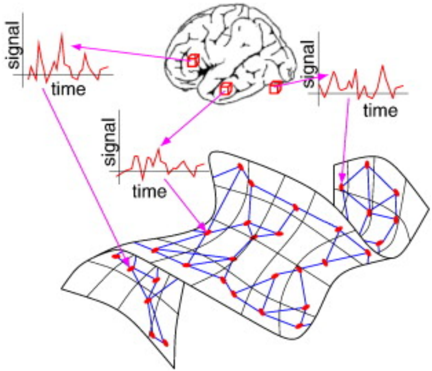
Low-dimensional embedding of fMRI

- fMRI provides a large scale measurement of neuronal activity



fMRI data

- Each voxel v_i generates a time series $\boldsymbol{x}_i = \begin{pmatrix} x_i(1) & \dots & x_i(T) \end{pmatrix}^\top \in \mathbb{R}^T$



A network of functionally correlated voxels

- A connectivity graph is formed with the standard approach:

$$W_{ij} = \begin{cases} \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/\sigma^2\right) & \text{if } v_i \sim v_j \\ 0 & \text{otherwise} \end{cases}$$

- The Euclidean distance between time series:

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \sum_{t=1}^{T} (x_i(t) - x_j(t))^2$$

- Choice for σ:

$$\sigma = 2 \min_{i<j} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$$

- Choice for n_n (nearest neighbor): user defined and varies from 7 to ... 100.

Matrices

- Diagonal degree matrix: $\mathbf{D}(i,i) = \sum_j W_{i,j}$
- Transition matrix: $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$
- Transition probabilities:

$$P_{i,j} = \mathbf{P}(i,j) = \frac{W_{i,j}}{\sum_j W_{i,j}}$$

- It is a row-stochastic matrix: $\sum_j P_{i,j} = 1$

Random walk and commute-time

- Consider a random walk on the graph denoted by Z_n: if the walk is at v_i it jumps to one of its neighbors v_j with probability $P_{i,j}$.
- if v_i and v_j are in the same functional area and v_i and v_k are in different functional areas, we expect that $P_{i,j} \gg P_{i,k}$.
- The *average hitting time* measures the number of steps that it takes for a random walk starting in v_i to hit v_j for the first time:

$$H(v_i, v_j) = E_i[T_j] \text{ with } T_j = \min\{n \geq 0; Z_n = j\}$$

- The hitting time is not symmetric, use the commute time instead:

$$\kappa(v_i, v_j) = H(v_i, v_j) + H(v_j, v_i) = E_i[T_j] + E_j[T_i]$$

The commute time distance in the spectral domain

- Let $(\lambda_k, \boldsymbol{\phi}_k)_{k=1}^N$ be the eigenvalue-eigenvector pairs of matrix $\mathbf{P}$ that can be easily computed because $\mathbf{D}^{1/2}\mathbf{P}\mathbf{D}^{-1/2}$ is a real symmetric matrix. Moreover (see Lecture #3) we have:

$$-1 \leq \lambda_N \leq \ldots \leq \lambda_k \leq \ldots \leq \lambda_1 = 1$$

- The commute time distance is:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = \sum_{k=2}^{K+1} \frac{1}{1-\lambda_k} \left(\frac{\boldsymbol{\phi}_k(i)}{\sqrt{\pi_i}} - \frac{\boldsymbol{\phi}_k(j)}{\sqrt{\pi_j}} \right)$$

- $\boldsymbol{\pi} = (\pi_1 \ldots \pi_i \ldots \pi_N)^\top$ is the eigenvector $\mathbf{P}^\top \boldsymbol{\pi} = \boldsymbol{\pi}$ with:

$$\pi_i = \frac{d_i}{\sum_{i,j} W_{i,j}}$$

Embedding

- The initial time series can now be embedded using the mapping: $\boldsymbol{x}_i \longrightarrow \Psi(\boldsymbol{x}_i)$, i.e., $\mathbb{R}^T \rightarrow \mathbb{R}^K$:

$$\Psi(\boldsymbol{x}_i) = \frac{1}{\sqrt{\pi_i}} \left(\frac{\phi_k(i)}{\sqrt{1-\lambda_2}} \dots \frac{\phi_k(i)}{\sqrt{1-\lambda_k}} \dots \frac{\phi_k(i)}{\sqrt{1-\lambda_{K+1}}} \right)^\top$$

- This is strictly equivalent to the commute-time embedding based on the normalized graph Laplacian (see Lecture #3).

Choosing the dimension

- Remind that each voxel in the brain corresponds to a graph node and there is a time series at each voxel:

$$\mathbf{X} = \begin{bmatrix} x_1(1) & \dots & x_1(t) & \dots & x_1(T) \\ \vdots & & \vdots & & \vdots \\ x_i(1) & \dots & x_i(t) & \dots & x_i(T) \\ \vdots & & \vdots & & \vdots \\ x_N(1) & \dots & x_N(t) & \dots & x_N(T) \end{bmatrix}$$

- Each column $\boldsymbol{x}(t)$ in this matrix is a scalar function defined over the graphs' vertices. It can be decomposed using the eigenvectors:

$$\boldsymbol{x}(t) = \sum_{k=2}^{K+1} \langle \boldsymbol{x}(t), \boldsymbol{\phi}_k \rangle \boldsymbol{\phi}_k + \boldsymbol{r}(t)$$

Choosing the dimension

- This can be written as:

$$\widehat{\boldsymbol{x}}(t) = \sum_{k=2}^{K+1} \langle \boldsymbol{x}(t), \boldsymbol{\phi}_k \rangle \boldsymbol{\phi}_k = \sum_{k=2}^{K+1} x_k(t) \boldsymbol{\phi}_k$$

  Therefore, each entry i (at each brain location or graph vertex) of this approximated vector is:

$$\widehat{\boldsymbol{x}}_i(t) = \sum_{k=2}^{K+1} x_k(t) \phi_k(i)$$

- The discrepancy between the initial observations and their approximate representation is:

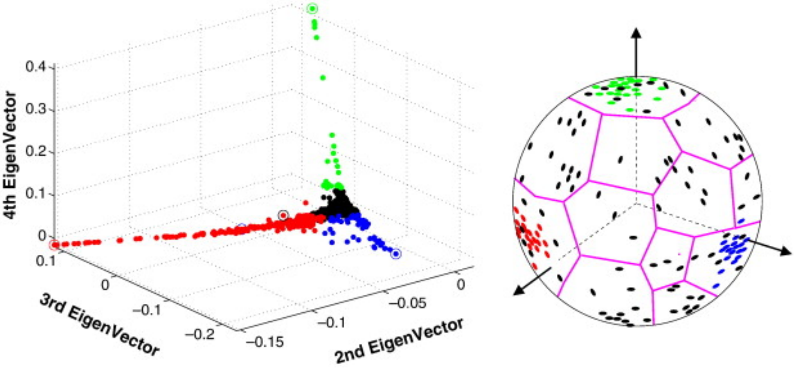
$$\varepsilon_i(K) = \frac{\sum_{t=1}^{T} (\boldsymbol{x}_i(t) - \widehat{\boldsymbol{x}}_i(t))^2}{\sum_{t=1}^{T} \boldsymbol{x}_i^2(t)}$$

Choosing the dimension

- The authors suggest to average $\varepsilon_i(K)$ over the voxels lying in a "functional" region, and to take the max over all these average values.
- It is not clear what is a functional region, since this is what it is searched for and why the average is selected.

Segmentation using K-means

- The authors notice that the embedding has a star-like shape.



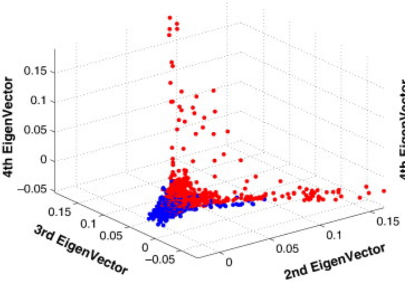
Segmentation using K-means

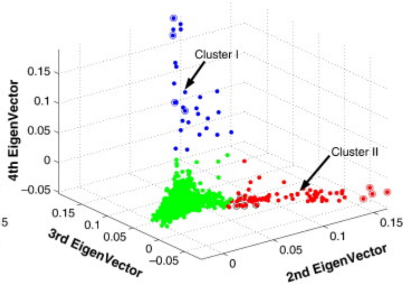
- The "arms" correspond to:
  - activated time series or
  - strong physiological artifacts
- The center blob corresponds to "background activity"
- The embedded data are projected on a hyper-sphere of dimension K.
- The background is spread over the sphere.
- The K-means algorithm is applied to the spherical data

Event related dataset

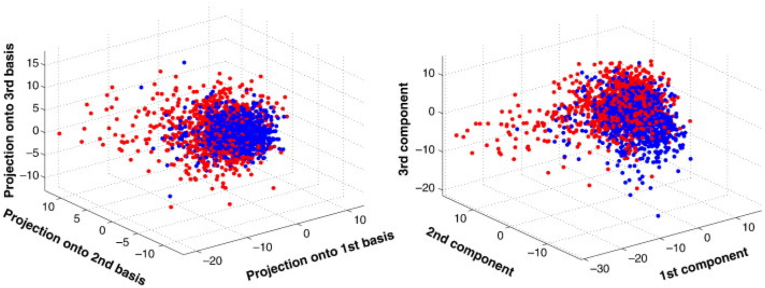
- Study of age-related changes in functional anatomy
- 2050 time series.

Spectral embedding/clustering results

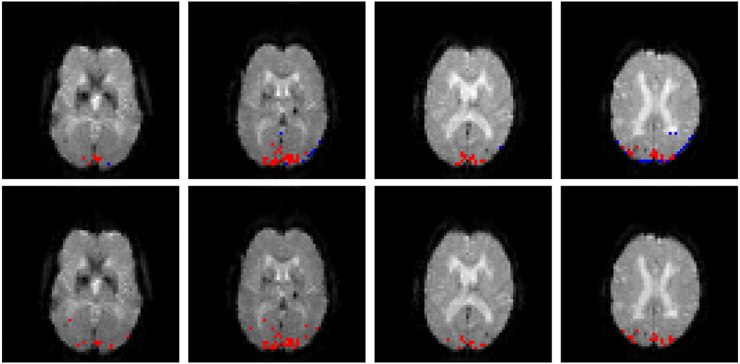




Results obtained with PCA and ISOMAP



Activation maps



The Google PageRank Algorithm

- A web page is important if many other pages point to it.
- $L_{ij} = 1$ if page j points to page i and zero otherwise
- $c_j = \sum_i L_{ij}$ number of pages pointing to by page j (number of outlinks)
- PageRanks p_i are defined by the recursive relationship:

$$p_i = (1 - d) + d \sum_j \left(\frac{L_{ij}}{c_j} \right) p_j$$

- $d = 0.85$

In matrix form

$$\mathbf{A} = \frac{1-d}{N}\boldsymbol{e}\boldsymbol{e}^\top + d\mathbf{L}\mathbf{D}_c^{-1}$$

- $\mathbf{A}$ has positive entries and each column sums to one - a column stochastic matrix.
- The algorithm becomes:

$$\boldsymbol{p} = \mathbf{A}\boldsymbol{p}$$

- $\boldsymbol{e} = (1 \ \ldots \ 1)^\top$, $\mathbf{D}_c = \mathsf{Diag}(c_i)$

Markov chain intepretation

- PageRank as a model of user behavior: a random web surfer clicks on links randomly without regard of the content.
- The surfer does a randome walk on the web, choosing among available outgoing links at random
- The maximal (equal to 1) eigenvalue-eigenvector pair of $\mathbf{A}$ corresponds to the stationary distribution of an irreducible aperiod Markov chain over the N web pages.