Graph-Based Pattern Recognition summary

A. Iosifidis, O. Platias, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 1.1

Artificial Intelligence &
Information Analysis Lab

Graph-Based Dimensionality Reduction

- Graph-based Clustering
- Locality Preserving Projections
- Locally Linear Embedding
- ISOMAP
- Laplacian Embedding
- Linear Discriminant Analysis
- Marginal Fisher Analysis
- Local Fisher Discriminant Analysis
- Semi-supervised Discriminant Analysis
- Laplacian Support Vector Machines

Dimensionality Reduction

Problem:

- Let n be the data (feature vector) dimensionality: $\mathbf{x} \in \mathbb{R}^n$.
- If it is high,
  - there are performance problems in data classification/clustering.
  - there are high computational costs in data classification/clustering
- Solution:
  - Feature vector ***Dimensionality Reduction (DR)*** to $d \ll n$.
  - DR must capture/retain the discriminative information of the data.

Dimensionality Reduction

- Applications:
  - Removal of irrelevant and noisy features.
  - Extraction of the most important features.
  - Data Visualization.
  - Data search and retrieval.
  - Coupled use with various ML techniques:
    - Data classification
    - Data clustering

Dimensionality Reduction

- Given a sample $\mathbf{x} \in \mathbb{R}^n$, the ML model computes a new sample representation $\hat{\mathbf{x}} = \boldsymbol{\phi}(\mathbf{x}; \boldsymbol{\theta})$.
- $\boldsymbol{\phi}: \mathbb{R}^n \rightarrow \mathbb{R}^d$ is a function, mapping $\mathbf{x}$ to a lower dimensionality space d, $d \ll n$,
- $\boldsymbol{\theta}$ are the learnable parameters of the model.
- The representation $\hat{\mathbf{x}}$ is meant:
  - to capture relevant high level information from the initial sample $\mathbf{x}$;
  - provide abstraction from detail
  - increase robustness to noise.

Artificial Intelligence & Information Analysis Lab

Dimensionality Reduction

- Unsupervised Methods
- Supervised Methods
- Semi-Supervised Methods

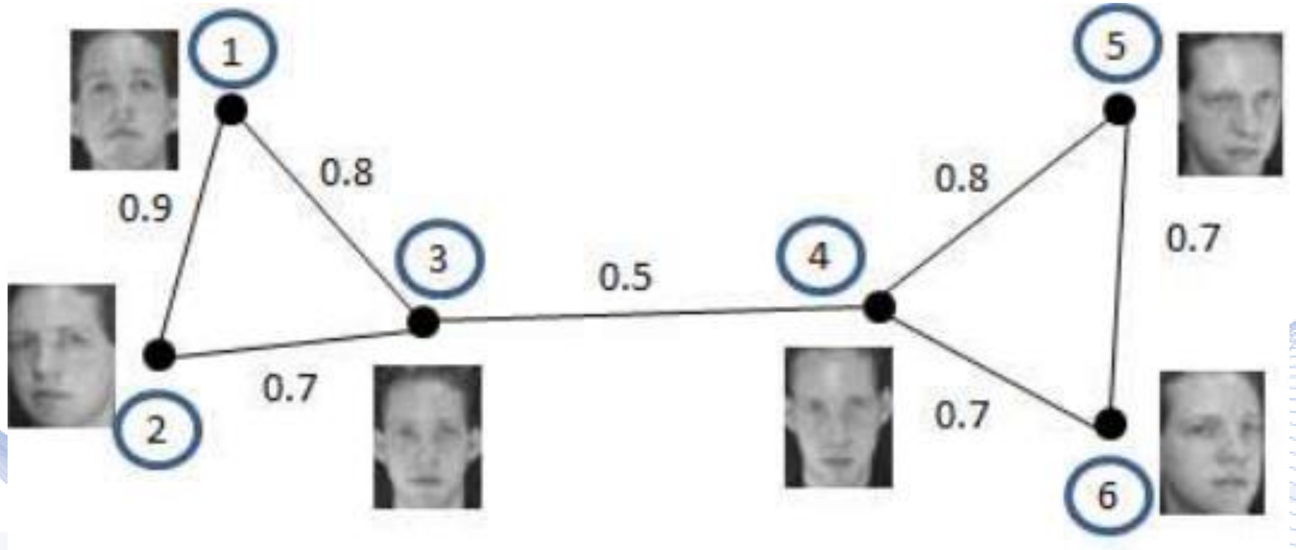
Graph-based Clustering

Similarity graph, Adjacency/Similarity matrix:

- Let $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ be the data set where $\mathbf{x}_i \in \mathbb{R}^{\boldsymbol{n}}$.
- Construct a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where each graph vertex corresponds to a point $\mathbf{x}_i$, $i = 1, \dots, N$.
- Graph is weighted connected and undirected.
- Graph $N \times N$ ***adjacency matrix*** $\mathbf{A} \in \{0,1\}^{N \times N}$.
- Similarity (weight) matrix $\mathbf{W} = [W_{ij}] \in \mathbb{R}^{N \times N}$.

Artificial Intelligence & Information Analysis Lab

Graph-based Clustering



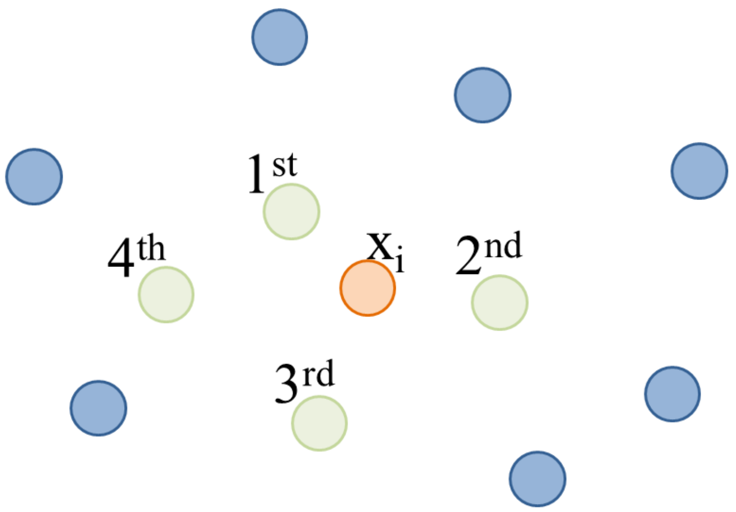
1	0.9	0.8	0	0	0
0.9	1	0.7	0	0	0
0.8	0.7	1	0.5	0	0
0	0	0.5	1	0.8	0.7
0	0	0	0.8	1	0.7
0	0	0	0.7	0.7	1

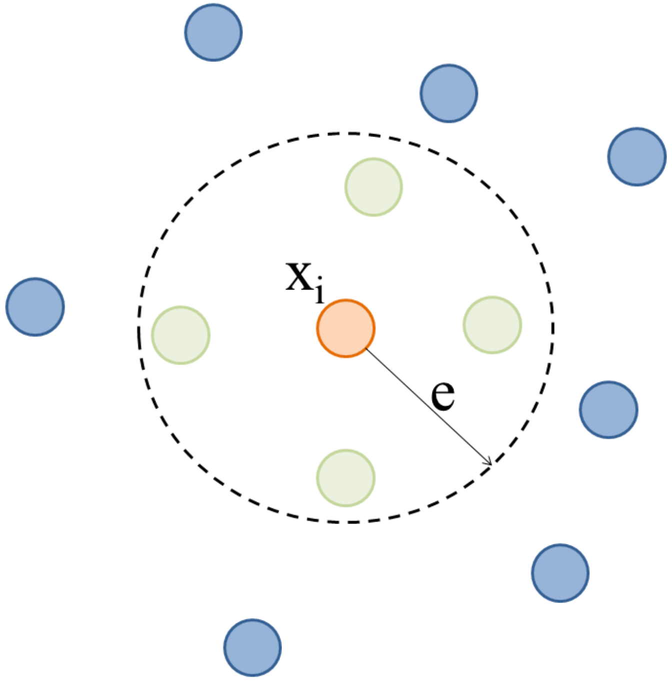
a) Similarity graph; b) Similarity matrix.

Artificial Intelligence & Information Analysis Lab

Graph-based Clustering

Nearest neighbor graphs





a) k-nearest neighbors graph; b) e-neighborhood graph.

Artificial Intelligence & Information Analysis Lab

Unsupervised Learning

- no class labels only geometric data relationship
- partition the graph vertex set into smaller clusters (graph clustering)
- embed graph vertices in a low-dimensional feature space, while preserving geometrical data properties

Semi-supervised Learning

- Some data have class - labels, the rest do not.

Classification approaches:

- **Transductive:** Use the geometric data relationships and the labels to assign labels to the unlabeled data items
- **Inductive:** Use the geometric data relationships and the labels to learn a function that maps new items or unlabeled data to classes.

Supervised Learning

- All data have class – labels.
- Each graph vertex $\mathbf{x}_i$ is accompanied by a class label $\mathcal{C}_i \epsilon \mathcal{C}$ where $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_m\}$.
  - Learn a mapping $f(\mathbf{x})$: $\mathbb{R}^n \longrightarrow \mathcal{C}$.
  - Once learned , this mapping can be used to map a new test sample $\mathbf{x}$ (not belonging to the training set $\mathbf{V}$) to one of the classes in $\mathcal{C}$.

Locality Preserving Projections

- Locality Preserving Projections (LPP) finds a low-dimensional embedding of the original data $\mathbf{x}_i \in \mathbb{R}^n$, so that nearby samples in the high-dimensional space $\mathbb{R}^n$ **remain placed nearby** in the low dimensional space $\mathbb{R}^d (d << n)$.
- It finds K nearest neighbors of each sample $\mathbf{x}_i$ based on Euclidean distances
- Constructs a neighborhood graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and the graph weight matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$:

$$W_{ij} = \begin{cases} 1, & i \in \mathcal{N}_j \text{ or } j \in \mathcal{N}_i \\ 0, & \text{otherwise.} \end{cases}$$

- $\mathcal{N}_i$: neighborhood of $\mathbf{x}_i$.

Locally Linear Embedding

- Difference from LPP is that LLE employs a **weighted** graph, while LPP employs a **unweighted** graph.

- A local fitting step is performed. Each sample $\mathbf{x}_i$ is approximated by its neighbour $\mathbf{x}_j, j \in \mathcal{N}_i$ according to fitting weights w_{ij} by solving:

$$\min_{\Sigma_j \in \mathcal{N}_i} \left\| \mathbf{x}_i - \sum_{j \in \mathcal{N}_i} W_{ij} (\mathbf{x}_j - \mathbf{x}_i) \right\|_2^2 .$$

ISOMAP

ISOMAP determines a low – dimensional embedding of the original data $\mathbf{x}_i$ so that the pairwise **geodesic** distances between the data are preserved in the low dimensional space. ISOMAP constructs a neighborhood graph vertices. Then the elements of the graph weight $\mathbf{W}$ are set to:

$$w_{ij} = \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2.$$

The shortest path distances calculate the distance matrix $\mathbf{D}$:

$$\mathbf{D}_{ij} = min\left(\left\| \mathbf{x}_j - \mathbf{x}_{t_1} \right\|_2 + \cdots + \left\| \mathbf{x}_{t_{k-1}} - \mathbf{x}_j \right\|_2 \right)$$

Laplacian Embedding

- LE compute a low – dimensional embedding of the original data $\mathbf{x}_i$ with the property that nearby samples in the high-dimensional space $\mathbb{R}^d$ remain placed nearby in the low dimensional space.
- After constructing the graph , the graph weight matrix $\mathbf{W}$ is constructed as:

$$\mathbf{W}_{ij} = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2}{2\sigma}\right).$$

- After eigenanalysis, this method can exploit **both local and global** geometric information, depending on the value of the parameter σ.
- This is an advantage in the cases where a smooth low-dimensional embedding is searched for.

Diffusion Maps

- This method is focused on the analysis of the geometry of general datasets based on the definition of **Markov chains**. For a fixed value ε,the isotropic diffusion kernel can be defined as:

$$k_{\varepsilon}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{4\varepsilon}\right).$$

Artificial Intelligence & Information Analysis Lab

Linear Discriminant Analysis

- Minimize within-class data scatter :

$$\mathbf{S}_w = \sum_{k=1}^{m} \sum_{\mathbf{x}_i \in \mathcal{C}_k} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$$

- Maximize the between-class data scatter :

$$\mathbf{S}_\mathrm{b} = \sum_{k=1}^{m} N_m(\mathbf{x}_i - \boldsymbol{\mu}_m)(\mathbf{x}_i - \boldsymbol{\mu}_m)^T$$

Marginal Fisher Analysis

- Similar as Marginal Analysis but here LFDA focuses more on local relationships.

$$W_{ij} = \begin{cases} 1\,, & \mathcal{C}_i = \mathcal{C}_j \text{ and } j \in \mathcal{N}_i \\ 1\,, & \mathcal{C}_i = \mathcal{C}_j \text{ and } i \in \mathcal{N}_j \\ 0, & \text{otherwise.} \end{cases}$$

- Penalty **W** matrix :

$$W_{ij}^{(p)} = \begin{cases} 1\,, & \mathcal{C}_i \neq \mathcal{C}_j \text{ and } j \in \mathcal{N}_i \\ 1\,, & \mathcal{C}_i \neq \mathcal{C}_j \text{ and } i \in \mathcal{N}_j \\ 0, & \text{otherwise.} \end{cases}$$

Local Fisher Discriminant Analysis

LDFA defines the within-class and between-class relationships by using graph relationships. There is an intrinsic graph and a penalty graph.

The matrix $\mathbf{W}_{ij}^{(w)}$ expresses local relationships between data belonging to the same class.

$$\mathbf{W}_{ij}^{(w)} = \begin{cases} \dfrac{S_{ij}}{N_{\mathcal{C}_i}}, & \mathcal{C}_i = \mathcal{C}_j \\ 0\,, & \text{otherwise} \end{cases}$$

Artificial Intelligence & Information Analysis Lab

Local Fisher Discriminant Analysis

The matrix $\mathbf{W}_{ij}^{(b)}$ expresses local relationships between data placed at the borders of different classes

$$W_{ij}^{(b)} = \begin{cases} S_{ij}\left(\frac{1}{N} - \frac{1}{N_{\mathcal{C}_i}}\right), & \mathcal{C}_i = \mathcal{C}_j \\ 0\,, & \text{otherwise.} \end{cases}$$

Artificial Intelligence & Information Analysis Lab

Graph Embedding

- Here the graph is undirected weighted where we assume that the training samples $\mathbf{x}_i$ reside on graph vertices and $\mathbf{W}$ is the corresponding graph weight matrix. Let us denote L_x the graph Laplacian matrix describing a certain criterion X.

$$\boldsymbol{S}_X = \mathbf{X}\mathbf{L}_T\mathbf{X}^T$$

Semi-supervised Discriminant Analysis

This method uses :

- discriminant information inferred from the labelled data
- Local geometrical information from both the labelled and the unlabelled data.

Weight matrix $\mathbf{W}$ expresses the local relationship between unlabelled and labelled data :

$$\mathbf{W}_{ij} = \begin{cases} w_{ij} \,, & i \in \mathcal{N}_j \text{ and } j \in \mathcal{N}_i \\ 0 \,, & \text{otherwise} \end{cases}$$

Laplacian Support Vector Machines

The K – nearest neighbour based on the heat kernel is usually employed. The following regualizer is incorporated in the SVM formulation.

That leads to the following optimization problem:

$$\min_{\mathrm{w,b}} \|\mathbf{w}\|_2^2 + \mathrm{c}_1 \sum_{i=1}^{\mathrm{N}} \xi_i + \frac{\mathrm{c}_2}{\mathrm{N}^2} \mathbf{w}^T (\mathbf{XLX}^T)$$

Where $\mathbf{L}$ is the Laplacian matrix calculated by using both the labelled and the unlabelled data.

Artificial Intelligence & Information Analysis Lab

Bibliography

[IOS2016] A.Iosifidis and I.Pitas, "Graph-Based Pattern Classification and Dimensionality Reduction" in Graph-Based Social Media Analysis, edited by: I. Pitas, pp. 163-186, CRC Press, 2016.

[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.

[PIT2017] I. Pitas, “Digital video processing and analysis ” , China Machine Press, 2017 (in Chinese).

[PIT2013] I. Pitas, “Digital Video and Television ” , Createspace/Amazon, 2013.

[NIK2000] N. Nikolaidis and I. Pitas, 3D Image Processing Algorithms, J. Wiley, 2000.

[PIT2000] I. Pitas, “Digital Image Processing Algorithms and Applications”, J. Wiley, 2000.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr