Manifold Learning

Ramesh Srinivasan

November 21, 2024

Topological Data Analysis

- Geometric and topological relationships are fundamental to essentially every data analysis and machine learning task, as we use geometry to identify similarities and distinctive characteristics in the data.
- In classification or clusteirng,data points that are similar (close to each other) are assigned to the same classes and data points that are significantly different (i.e. far apart in one or many of the feature dimensions) are assigned to different labels.
- We usually consider data as a finite set $X = \{x_1, x_2, \cdots, x_N\} \subset \mathbb{R}^D$ sometimes called a point cloud.
- It is often simply assumed that our data lies in the standard Euclidean space with the typical Euclidean distance metric
- The point of todays class is that many interesting and important structures that arise are actually non-Euclidean, and by fitting a continuous shape (such as a manifold) to the data, we can translate our data analysis task from an external, global coordinate system (possibly having very high dimensionality) into the intrinsic coordinate system defined by the assumed manifold structure itself.
- The goal is for the underlying manifold to capture the latent structure in our dataset.

Manifold Learning

- If the can be approximated by a manifold - topological properties remain constant with continuous deformations.
- We can often apply a dimensionality reduction transformation to map the data into lower dimensional spaces, while preserving important topological properties (not only distance but also connectedness and compactness).
- The **manifold hypothesis**, or **manifold assumption**, is that many real-world datasets can be approximately represented as lower dimensional manifolds that are embedded in a higher dimensional space.
- **Manifold learning** is a class of methods developed to learn this lower dimensional representation, sometimes referred to as the intrinsic dimensionality, of the data.
- In manifold learning, the data points $x_1, x_2, ..., x_N$ are assumed to be sampled from the (often uniform) distribution defined by the underlying d-dimensional manifold $M \subset \mathbb{R}^D$. We then attempt to learn the intrinsic structure of this underlying manifold.

Multidimensional Scaling

- Multidimensional scaling uses pairwise similarity measures (which can be similarity measures of any kind, even qualitative ratings of similarity) to construct a spatial representation that keeps similar objects close together and dissimilar objects further apart. In this way, the goal is to capture the structure of the higher dimensional data in a lower dimensional representation.
- The input to MDS is only the dissimilarity matrix, instead of the actual position vectors of the data.
- Given a pairwise dissimilarity matrix D with entries d_{ij} for the distance/dissimilarity between observations i and j, we find $x_1, \ldots, x_n \in \mathbb{R}^k$ such that:

$$\underbrace{d_{ij}^2}_{\text{original distances}} \approx \underbrace{||x_i - x_j||^2}_{\text{output configuration}}$$

- We find a configuration (typically a lower dimensional configuration in $\mathbb{R}^2$) that keeps the Euclidean distances in $\mathbb{R}^k$ as close as possible to our original distances/similarities.
- MDS considers global similarities by attempting to preserve all the pairwise distances (instead of preserving local neighborhood similarities like many other manifold learning algorithms), and this often limits the ability of MDS to produce non-linear embeddings.

Metric MDS

- Dissimilarities are quantitative but not necessarily Euclidean
- For metric MDS, we have loss function defined as:

$$L_D(x_1, x_2, \ldots, x_N) = \left(\sum_{i \neq j = 1, \ldots, N} \left(d_{ij} - \left\| x_i - x_j \right\| \right)^2 \right)^{1/2}$$

- We attempt to find the configuration $x_1, \ldots, x_n$ with Euclidean distances to minimize this error given our input matrix D

Nonmetric MDS

- Non-metric MDS - dissimilarities are qualitative (similarity measures are interpreted more as rankings or ordinal classifications)
- In the non-metric case, the dissimilarity matrix we are given is only important in its relative ranking of the observations, not the quantitative degree to which observations differ.
- For non-metric MDS, we have a monotonic function f and we find a configuration that only preserves relative ranking of the data.
- If $d_{ij} < d_{ik}, \Rightarrow f(d_{ij}) \leq f(d_{ik})$
- Non-metric MDS can also be stated as the problem of finding the optimal configuration $x_1, ..., x_n \in \mathbb{R}^k$ that minimizes the following loss function:

$$L_D\left(x_1, x_2, \ldots, x_N\right) = \sqrt{\frac{\sum_{i \neq j = 1, \ldots, N} (f(d_{ij}) - \left\|x_i - x_j\right\|)^2}{\sum i < j \left\|x_i - x_j\right\|^2}}$$

ISOMAP

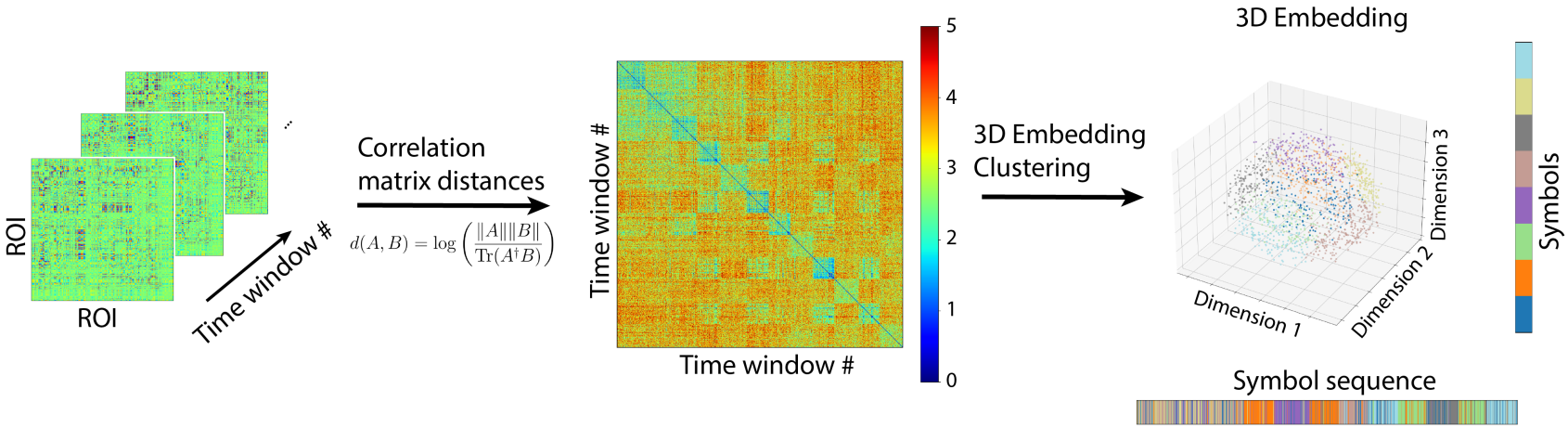
- **Isomap** introduces learning global geometric structure based on representing high dimensional datasets with nearest neighbor graphs. The idea is that the distances you travel along the edges of the graph approximate distances along the manifold.
- By learning the intrinsic metric defined along the manifold instead of the metric defined by an external coordinate system, isomap can learn highly complex, non-linear geometric relationships. Additionally, isomap then provides a lower dimensional embedding transformation that preserves the higher dimensional structure.
- Before going into the details of the algorithm, we define two important concepts. Given two metric spaces X and Y with metrics d_x and d_y, an **Isometry** is a bijective map $f : X \to Y$ between the two metric spaces that preserves distances:

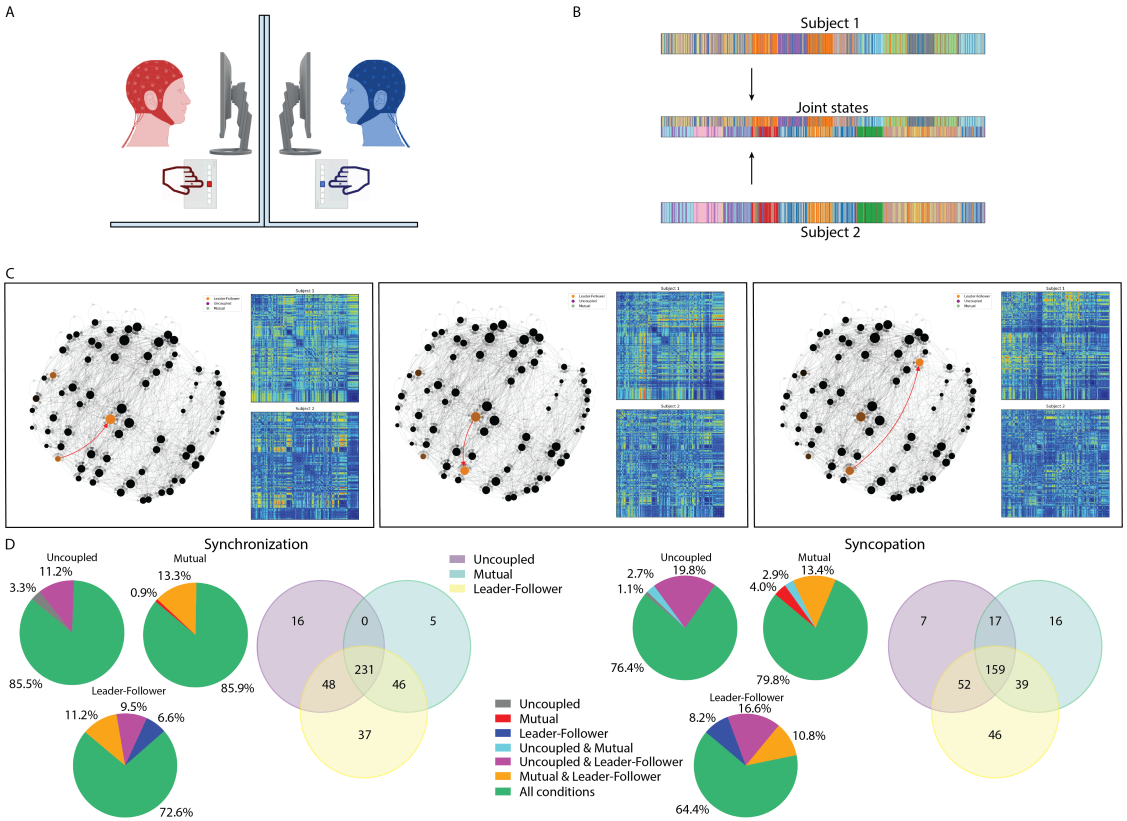
$$d_Y(f(a), f(b)) = d_X(a, b)$$

- To find an isometric mapping from the manifold where the distance between data points x_i and x_j along the manifold (as the graph geodesic distance) is equal to the Euclidean distance between the corresponding lower dimensional output vectors y_i and y_j.

Graph Geodesic Distance

- For any general surface, a geodesic is a locally length minimizing path. Geodesics generalize the notion of straight-line shortest paths (in the Euclidean plane) to intrinsically defined surface geometries.
- In the isomap algorithm, we approximate the shortest path along the surface of the manifold by the **graph geodesic** - the minimally weighted path between two nodes.
- Data vectors $\mathbf{x_1}, ..., \mathbf{x_n}$ of d dimension along with parameters k (nearest neighbors) and m (embedding dimension) are mapped to Lower dimensional embedding vectors $\mathbf{y_i}, ..., \mathbf{y_n}$ with $m < d$
- Algorithm:
  1. Find the k nearest neighbors for each point and create nearest neighbors graph G with data points $\mathbf{x_i}$ as the nodes and edges connecting nearest neighbors. The edge weights are set to the distance.
  2. Compute geodesic distance matrix D with pairwise shortest-path distances along the weighted graph
  3. Apply metric Multidimensional Scaling to matrix D to form the lower dimensional embedding to estimate $\mathbf{y_i}$


ROI
ROI
Time window #
Correlation
matrix distances
$d(A,B) = \log\left(\frac{\|A\|\|B\|}{\mathrm{Tr}(A^\dagger B)}\right)$
Time window #
Time window #
5
4
3
2
1
0
3D Embedding
Clustering
3D Embedding
Dimension 1
Dimension 2
Dimension 3
Symbols
Symbol sequence


A
B
Subject 1
Joint states
Subject 2
C
D
Synchronization
Uncoupled
11.2%
3.3%
85.5%
Mutual
13.3%
0.9%
85.9%
Leader-Follower
11.2%
9.5%
6.6%
72.6%
16
0
5
231
48
46
37
Uncoupled
Mutual
Leader-Follower
Uncoupled
Mutual
Leader-Follower
Uncoupled & Mutual
Uncoupled & Leader-Follower
Mutual & Leader-Follower
All conditions
Syncopation
Uncoupled
2.7%
1.1%
76.4%
Mutual
2.9%
4.0%
79.8%
Leader-Follower
16.6%
8.2%
10.8%
64.4%
7
17
16
159
52
39
46