Inverse Distance Problems

Justin Solomon
MIT, Spring 2017
Last Time

Embedding \( \subseteq \mathbb{R}^3 \)  

Geodesic distance

Right bunny from “Geodesics in Heat” (Crane et al.)
Today

Embedding \( \subseteq \mathbb{R}^3 \)  

Geodesic distance

Right bunny from “Geodesics in Heat” (Crane et al.)
Many Names

- Dimensionality reduction
- Embedding
- Multidimensional scaling
- Manifold learning
  ...
Basic Task

Given pairwise distances, extract an embedding.

Is it always possible?
What dimensionality?
Ordered pair \((M, d)\) where \(M\) is a set and \(d: M \times M \rightarrow \mathbb{R}\) satisfies

\[
\begin{align*}
    d(x, y) &\geq 0 \\
    d(x, y) = 0 &\iff x = y \\
    d(x, y) &= d(y, x) \\
    d(x, z) &\leq d(x, y) + d(y, z)
\end{align*}
\]

\(\forall x, y, z \in M\)
Many Examples of Metric Spaces

\( \mathbb{R}^n, d(x, y) := \|x - y\|_p \)

\( S \subset \mathbb{R}^3, d(x, y) := \text{geodesic} \)

\( C^\infty(\mathbb{R}), d(f, g)^2 := \int_{\mathbb{R}} (f(x) - g(x))^2 \, dx \)
Isometry [ahy-som-i-tree]:
A map between metric spaces that preserves pairwise distances.
Q: Can you always embed a metric space isometrically in \( \mathbb{R}^n \)?
Q: Can you always embed a finite metric space isometrically in $\mathbb{R}^n$?
Disappointing Example

\[ X := \{a, b, c, d\} \]
\[ d(a, d) = d(b, d) = 1 \]
\[ d(a, b) = d(a, c) = d(b, c) = 2 \]
\[ d(c, d) = 1.5 \]

Cannot be embedded in Euclidean space!

https://chiasme.wordpress.com/2013/10/07/when-does-a-finite-metric-space-embed-isometrically-into-an-euclidean-space/
Approximate Embedding

\[
\text{expansion}(f) := \max_{x,y} \frac{\mu(f(x), f(y))}{\rho(x, y)} \\
\text{contraction}(f) := \max_{x,y} \frac{\rho(x, y)}{\mu(f(x), f(y))} \\
\text{distortion}(f) := \text{expansion}(f) \times \text{contraction}(f)
\]

Theorem (Bourgain, 1985). Let \((X, d)\) be a metric space on \(n\) points. Then,

\[
(X, d) \xrightarrow{O(\log n)} \ell_p O(\log^2 n)
\]
**Euclidean Case**

\[
D_{ij} = \left\| x_i - x_j \right\|_2^2, \quad D \in \mathbb{R}^{n \times n}
\]

**Proposition.** \( \text{Rank}(D) \leq \min(n, m + 2) \).

**Proof:**

\[
D = -2X^\top X + \text{diag}(X^\top X)1^\top + 1\text{diag}(X^\top X)^\top
\]

Embedding via eigenvalue problem (take \( x_1 = 0 \)):

\[
\left\| x_i - x_j \right\|_2^2 = \left\| x_i \right\|_2^2 + \left\| x_j \right\|_2^2 - 2x_i \cdot x_j
\]

\[
\Rightarrow x_i \cdot x_j = \frac{1}{2} \left[ \left\| x_i \right\|_2^2 + \left\| x_j \right\|_2^2 - \left\| x_i - x_j \right\|_2^2 \right]
\]
Gram Matrix [gram mey-triks]:
A matrix of inner products
\[ X^T X \]
1. Double centering: \( B := -\frac{1}{2} J D J \)
   Centering matrix \( J := I - \frac{1}{n} 1 1^\top \)

2. Find \( m \) largest eigenvalues/eigenvectors

3. \( X = E_m \Lambda_m^{1/2} \)


“MDS”
Stress Majorization

\[ \min_X \sum_{i,j} \left( d_{ij}^0 - \| x_i - x_j \|_2 \right)^2 \]

Nonconvex!

**SMACOF**: Scaling by **Majorizing a Complicated Function**

SMACOF Potential Terms

\[
\sum_{ij} (d_{ij}^0)^2 = \text{const}.
\]

\[
\sum_{ij} \| x_i - x_j \|_2^2 = \text{tr}(XVX^\top), \text{ where } V = 2nI - 211^\top
\]

\[
\sum_{ij} d_{ij}^0 \| x_i - x_j \|_2 = \text{tr}(XB(X)X^\top)
\]

where \( b_{ij}(X) := \begin{cases} 
-\frac{2d_{ij}^0}{\| x_i - x_j \|_2} & \text{if } x_i \neq x_j, i \neq j \\
0 & \text{if } x_i = x_j, i \neq j \\
-\sum_{j \neq i} b_{ij} & \text{if } i = j
\end{cases} \)
Lemma. Define

$$\tau(X, Z) := \text{const.} + \text{tr}(XVX^\top) - 2\text{tr}(XB(Z)Z^\top)$$

Then,

$$\tau(X, X) \leq \tau(X, Z) \quad \forall Z$$

with equality exactly when $X=Z$.

Proof on board using Cauchy-Schwarz.

See Modern Multidimensional Scaling (Borg, Groenen)
SMACOF: Single Step

\[ X^{k+1} \leftarrow \min_X \tau(X, X^k) \]

\[ \tau(X, Z) := \text{const.} + \text{tr}(XVX^\top) - 2\text{tr}(XB(Z)Z^\top) \]

\[ \implies 0 = \nabla_X [\tau(X, X^k)] \]

\[ = 2XV - 2X^k B(X^k) \]

\[ \implies X^{k+1} = X^k B(X^k)V^+ \]

\[ V^+ = (2nI - 211^\top)^+ \]

\[ = \frac{1}{2n} \left( I - \frac{11^\top}{n} \right)^+ \]

\[ = \frac{1}{2n} \left( I - \frac{11^\top}{n} \right) \]

Majorization-Minimization algorithm

Objective convergence:
\[ \tau(X^{k+1}, X^{k+1}) \leq \tau(X^k, X^k) \]
Recent SMACOF Application

DOI: 10.1111/cgf.12558
EUROGRAPHICS 2015 / O. Sorkine-Hornung and M. Wimmer
(Guest Editors)

Volume 34 (2015), Number 2

Shape-from-Operator: Recovering Shapes from Intrinsic Operators

Davide Boscaini, Davide Eynard, Drosos Kourounis, and Michael M. Bronstein

Università della Svizzera Italiana (USI), Lugano, Switzerland

Figure 1: Examples of three different shape-from-operator problems considered in the paper. Left: shape analogy synthesis as shape-from-difference operator problem (shape X is synthesized such that the intrinsic difference operator between C, X is as close as possible to the difference between A, B). Center: style transfer as shape-from-Laplacian problem. The Laplacian of the shapes are matched. Right: eigenshape transfer as shape-from-eigenvectors problem.
Related Method

\[ \min_X \sum_{ij} \left( \frac{d_{ij}^0 - \|x_i - x_j\|_2}{d_{ij}^0} \right)^2 \]

Cares more about preserving small distances

"Sammon mapping"


http://www.stat.pitt.edu/sungkyu/course/2221Fall13/lec8_mds_combined.pdf
Intrinsic-to-Extrinsic: ISOMAP

- Construct neighborhood graph
  - $k$-nearest neighbor graph or $\varepsilon$-neighborhood graph
- Compute shortest-path distances
  - Floyd-Warshall algorithm or Dijkstra
- Classical MDS
  - Eigenvalue problem

Swiss roll dataset

Tenenbaum, de Silva, Langford.
Floyd-Warshall Algorithm

let dist be a \(|V| \times |V|\) array of minimum distances initialized to \(\infty\) (infinity)

for each vertex \(v\)
    \(\text{dist}[v][v] \leftarrow 0\)

for each edge \((u, v)\)
    \(\text{dist}[u][v] \leftarrow w(u, v)\) // the weight of the edge \((u, v)\)

for \(k\) from 1 to \(|V|\)
    for \(i\) from 1 to \(|V|\)
        for \(j\) from 1 to \(|V|\)
            if \(\text{dist}[i][j] > \text{dist}[i][k] + \text{dist}[k][j]\)
                \(\text{dist}[i][j] \leftarrow \text{dist}[i][k] + \text{dist}[k][j]\)
            end if

https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm
Landmark ISOMAP

- Construct neighborhood graph
  \( k \)-nearest neighbor graph or \( \varepsilon \)-neighborhood graph

- Compute some shortest-path distances
  Dijkstra: \( O(kn \, N \log N) \), \( n \) landmarks, \( N \) points

- MDS on landmarks
  Smaller \( n \times n \) problem

- Closed-form embedding formula

\[
\delta(x) \text{ vector of squared distances from } x \text{ to landmarks}
\]

\[
\text{Embedding}(x)_i = -\frac{1}{2 \sqrt{\lambda_i}} \begin{bmatrix} v_i \end{bmatrix} \left( \delta(x) - \delta_{\text{average}} \right)
\]
Locally Linear Embedding (LLE)

- Construct neighborhood graph
  $k$-nearest neighbor graph or $\varepsilon$-neighborhood graph

- Compute weights $W_{ij}$

$$
\min_{1^T W_i = 1} \left\| x_i - \sum_{x_j \in \mathcal{N}(x_i)} W_{ij} x_j \right\|_2^2
$$

- Minimum eigenvalue problem

$$
\min_{YY^T = I} \left\| y_i - \sum_{x_j \in \mathcal{N}(x_i)} W_{ij} y_j \right\|_2^2
$$

Derive on board
# Comparison: ISOMAP vs. LLE

<table>
<thead>
<tr>
<th>ISOMAP</th>
<th>LLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global distances</td>
<td>Local averaging</td>
</tr>
<tr>
<td>(k)-NN graph distances</td>
<td>(k)-NN graph weighting</td>
</tr>
<tr>
<td>Largest eigenvectors</td>
<td>Smallest eigenvectors</td>
</tr>
<tr>
<td>Dense matrix</td>
<td>Sparse matrix</td>
</tr>
</tbody>
</table>

Other option:

**Diffusion Maps**

- **Construct similarity matrix**
  
  **Example:** \( K(x, y) := e^{-\frac{||x-y||^2}{\varepsilon}} \)

- **Normalize rows**

  \[
  M := D^{-1} K
  \]

- **Embed from \( k \) largest eigenvectors**

  \((\lambda_1 \psi_1, \lambda_2 \psi_2, \ldots, \lambda_k \psi_k)\)

---

On reconstruction of non-rigid shapes with intrinsic regularization

Yohai S. Devir  Guy Rosman  Alexander M. Bronstein  Michael M. Bronstein  Ron Kimmel
{yd|rosman|bron|mbron|ron}@cs.technion.ac.il
Department of Computer Science
Technion - Israel Institute of Technology

Abstract

Shape-from-X is a generic type of inverse problems in computer vision, in which a shape is reconstructed from some measurements. A specially challenging setting of this problem is the case in which the reconstructed shapes are non-rigid. In this paper, we propose a framework for intrinsic regularization of such problems. The assumption is that we have the geometric structure of a shape which is intrinsically (up to bending) similar to the one we would like to reconstruct. For that goal, we formulate a variation with respect to vertex coordinates of a triangulated mesh approximating the continuous shape. The numerical core of the proposed method is based on differentiating the fast marching update step for geodesic distance computation.

1. Introduction

In many tasks, both in human and computer vision, one tries to deduce the shape of an object given an observation. Many other problems, in which an object is reconstructed based on some measurement, are known as shape reconstruction problems. They are a subset of what is called inverse problems. Most such inverse problems are under-determined, in the sense that measuring different objects may yield similar measurements. Thus, in the above illustration, the essence of the shadow theater is that it is hard to distinguish between shadows cast by an animal and shadows cast by hands. Therefore prior knowledge about the unknown object is needed.

Of particular interest are reconstruction problems involving non-rigid shapes. The world surrounding us is full with objects such as live bodies, paper products, plants, clothes etc., which may be deformed to different postures. These objects may be deformed to an infinite number of different postures. While bending, though, objects tends to preserve their internal geometric structure. Two objects differing by a bending are said to be intrinsically similar. In many cases, while we do not know the measured object, we have a prior on its intrinsic geometry. For example, in the shadow theater, though we do not know which exact posture of the hand
Huge zoo of embedding techniques.

Each with different theoretical properties: Try them all!

But what if the distance matrix is incomplete or noisy?
Euclidean Matrix Completion

\[
\min_G \| H \circ (\mathcal{D}(G) - D_{\text{input}}) \|_{\text{Fro}}^2
\]

s.t. \( G \succeq 0 \)

Related method: “Maximum variance unfolding”

**More General: Metric Nearness**

\[
\min_{X \in \mathcal{M}_{N \times N}} \| X - D \|_F^2
\]

**Triangle Fixing**

**Input:** Input dissimilarity matrix \( D \), tolerance \( \epsilon \).

**Output:** \( M = \arg\min_{X \in \mathcal{M}_N} \| X - D \|_2 \).

for \( 1 \leq i < j < k \leq n \)

\[
(z_{ijk}, z_{jki}, z_{kij}) \leftarrow 0
\]

for \( 1 \leq i < j \leq n \)

\[
e_{ij} \leftarrow 0
\]

\[
\delta \leftarrow 1 + \epsilon
\]

while (\( \delta > \epsilon \))

\{convergence test\}

foreach triangle \( (i, j, k) \)

\[
b \leftarrow d_{ki} + d_{jk} - d_{ij}
\]

\[
\mu \leftarrow \frac{1}{3}(e_{ij} - e_{jk} - e_{ki} - b)
\]

\[
\theta \leftarrow \min\{-\mu, z_{ijk}\}
\]

\{Stay within half-space of constraint\}

\[
e_{ij} \leftarrow e_{ij} - \theta, e_{jk} \leftarrow e_{jk} + \theta, e_{ki} \leftarrow e_{ki} - \theta
\]

\{Update correction term\}

\[
z_{ijk} \leftarrow z_{ijk} - \theta
\]

end foreach

\[
\delta \leftarrow \text{sum of changes in the } e_{i,j}
\]

end while

return \( M = D + E \)

In other words, the vector \( e \) is projected orthogonally onto the constraint set \( \{ e' : e'_{ij} - e'_{jk} - e'_{ki} \leq b_{ijk} \} \). This is tantamount to solving

\[
\min_{e'} \frac{1}{2} [(e'_{ij} - e_{ij})^2 + (e'_{jk} - e_{jk})^2 + (e'_{ki} - e_{ki})^2],
\]

subject to \( e'_{ij} - e'_{jk} - e'_{ki} = b_{ijk} \).

It is easy to check that the solution is given by

\[
e'_{ij} \leftarrow e_{ij} - \mu_{ijk}, \quad e'_{jk} \leftarrow e_{jk} + \mu_{ijk}, \quad \text{and} \quad e'_{ki} \leftarrow e_{ki} + \mu_{ijk},
\]

where \( \mu_{ijk} = \frac{1}{3}(e_{ij} - e_{jk} - e_{ki} - b_{ijk}) > 0 \).

---

- Is my data **embeddable**?
- Can you compute intrinsic **dimensionality**?
- Are two metric spaces **isometric**?
- How **similar** are two metric spaces?
- What is the **average** of two metric spaces?
- Can I embed into **non-Euclidean** spaces?
Robust Euclidean Embedding

Lawrence Cayton  
Sanjoy Dasgupta  
Department of Computer Science and Engineering, University of California, San Diego  
9500 Gilman Dr. La Jolla, CA 92037

Abstract

We derive a robust Euclidean embedding procedure based on semidefinite programming that may be used in place of the popular classical multidimensional scaling (cMDS) algorithm. We motivate this algorithm by arguing that cMDS is not particularly robust and has several other deficiencies. General-purpose semidefinite programming solvers are too memory intensive for medium to large sized applications, so we also describe a fast subgradient-based implementation of the robust algorithm. Additionally, since cMDS is often used for dimensionality reduction, we provide an in-depth look at reducing dimensionality with embedding procedures. In particular, we show that it is NP-hard to find optimal low-dimensional embeddings under a variety of cost functions.

ɛ1 Euclidean Embedding

Input: A dissimilarity matrix \( D = (d_{ij}) \).

Output: An embedding into the line: \( x_1, x_2, \ldots \in \mathbb{R} \)

Goal: Minimize \( \sum_{i,j} |d_{ij} - ||x_i - x_j||| \).

We show that this problem is NP-hard by reducing from a variant of not-all-equal 3SAT.

The hardness result can be extended to distortion functions of the form \( \sum_{i,j} g(f(d_{ij}) - f(||x_i - x_j||)) \) assuming that \( f, g \) are

1. symmetric;
2. monotonically increasing in the absolute values of their arguments;
3. Lipschitz on \([0,1]\) with constant \( \lambda_U \), that is, for \( x, y \in [0,1], |f(x) - f(y)| \leq \lambda_U |x - y| \); and
4. similarly lower-bounded: for some \( \lambda_L > 0 \), for any \( x, y \in [0,1], |f(x) - f(y)| \geq \lambda_L |x - y| \max\{x, y\} \).

Notice that \( f(x), g(x) \in \{x, x^2\} \) satisfy these conditions with \( \lambda_U = 2, \lambda_L = 1 \), meaning that \( \|D - D^*\|_1 \) and \( \|D - D^*\|_2 \) are both hard to minimize over one-dimensional embeddings.
Dependence on Curvature

positively curved space sphere

negatively curved space saddle

https://www.learner.org/courses/physics/unit/text.html?unit=3&secNum=6
Q: What are some applications of this machinery?
Applications

- Reduce algorithmic runtime
  - Compression
  - Visualize data
  - Interpolate
    - Sample
    ...

Figure 10: Nanotube Embedding. One of Asimov’s graphs for a nanotube is rendered with MDS in 3-D (Stress=0.06). The nodes represent carbon atoms, the lines represent chemical bonds. The right hand frame shows the cap of the tube only. The highlighted points show some of the pentagons that are necessary for forming the cap.
Figure 9: A Telephone Call Graph, Layed Out in 2-D. Left: classical scaling (Stress=0.34); right: distance scaling (Stress=0.23). The nodes represent telephone numbers, the edges represent the existence of a call between two telephone numbers in a given time period.

Figure 7: Maps of Computer Commands for Two Individuals. 
Left: a member of technical staff who programs and manipulates data (Stress=0.29). 
Right: an administrative assistant who does e-mail and word processing (Stress=0.34). 
The labels show selected operating system commands used by these individuals.
Inverse Distance Problems

Justin Solomon
MIT, Spring 2017