Last Time

Embedding $\subseteq \mathbb{R}^3$  $\rightarrow$  Geodesic distance

Right bunny from "Geodesics in Heat" (Crane et al.)
Right bunny from “Geodesics in Heat” (Crane et al.)
Many Overlapping Tasks

- Dimensionality reduction
  - Embedding
  - Parameterization
  - Manifold learning
  ...

Given pairwise distances
extract an embedding.

Is it always possible?
Embedding into which space?
What dimensionality?
Ordered pair \((M, d)\) where \(M\) is a set and \(d: M \times M \to \mathbb{R}\) 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!*
Contrasting Example

**Proposition.** Every finite metric space embeds isometrically into $\ell_\infty(\mathbb{R}^n)$ for some $n$.

Extends to infinite-dimensional spaces!
Approximate Embedding

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

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

\[
\text{distortion}(f) := \text{expansion}(f) \times \text{contraction}(f)
\]
Fréchet Embedding

**Definition** (Fréchet embedding). Suppose $(M, d)$ is a metric space that $S_1, \ldots, S_r \subseteq M$. We define the Fréchet embedding of $M$ with respect to $\{S_1, \ldots, S_r\}$ to be the map $\phi : M \to \mathbb{R}^r$ given by

$$\phi(x) := (d(x, S_1), d(x, S_2), \ldots, d(x, S_r)),$$

where $d(x, S) := \min_{y \in S} d(x, y)$. 
Well-Known Result

**Proposition** (Bourgain’s Theorem). Suppose \((M, d)\) is a metric space consisting of \(n\) points, that is, \(|M| = n\). Then, for \(p \geq 1\), \(M\) embeds into \(\ell_p(\mathbb{R}^m)\) with \(O(\log n)\) distortion, where \(m = O(\log^2 n)\).

Matousek improved the distortion bound to \(\log^{n/p}\) [14].

\[
m := 576 \log(n)
\]

\[
\begin{align*}
\text{for } j = 1 \text{ to } \log n & \text{ do} \\
& \text{/* levels of density */} \\
& \text{for } i = 1 \text{ to } m \text{ do} \\
& \quad \text{/* repeat for high probability */} \\
& \quad \text{choose set } S_{ij} \text{ by sampling each node in } X \\
& \quad \text{ independently with probability } 2^{-j} \\
& \text{end}
\end{align*}
\]

\[
\begin{align*}
f_{ij}(x) := d(x, S_{ij})
\end{align*}
\]

\[
f(x) := \bigoplus_{j=1}^{\log n} \bigoplus_{i=1}^m f_{ij}(x)
\]

Uses Fréchet embedding
Recall: Isometry [ahy-som-i-tree]:
A map between metric spaces that preserves pairwise distances.
Euclidean Problem

Given:

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

Reconstruct:

\[ x_1, \ldots, x_n \in \mathbb{R}^m \]

Alternative notation:

\[ X \in \mathbb{R}^{m \times n} \]
Gram Matrix [gram mey-triks]:
A matrix of inner products

\[ X^T X \]
Classical Multidimensional Scaling

1. Double centering: \( G := -\frac{1}{2} J^\top P J \)
   Centering matrix \( J := I_{n \times n} - \frac{1}{n} 11^\top \)

2. Find \( m \) largest eigenvalues/eigenvectors
   \( G = Q \Lambda Q^\top \)

3. \( \bar{X} = \sqrt{\Lambda} Q^\top \)

"MDS"

Extension: Landmark MDS

Simple Example

Voting patterns

https://en.wikipedia.org/wiki/Multidimensional_scaling#/media/File:RecentVotes.svg
where $p$ contains squared distances to landmarks.

Stress Majorization

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

Nonconvex!

SMACOF:
Scaling by Majorizing a Complicated Function

SMACOF Potential Terms

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

\[
\sum_{i,j} (D_{0ij})^2 = \text{const.}
\]

\[
\sum_{i,j} \|x_i - x_j\|_2^2 = \text{tr}(XVX^T), \text{ where } V = 2nJ
\]

\[
-2 \sum_{i,j} D_{0ij} \|x_i - x_j\|_2 = -2\text{tr}(XB(X)X^T)
\]

where \( B_{ij}(X) := \begin{cases} 
-\frac{2D_{0ij}}{\|x_i - x_j\|_2} & \text{if } x_i \neq x_j, i \neq j \\
0 & \text{if } x_i = x_j, i \neq j \\
-\sum_{j \neq i} B_{ij} & \text{if } i = j
\end{cases} \)
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 \( X \propto Z \).

Proof using Cauchy-Schwarz.

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

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

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

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

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

\[ \implies X^{k+1} = X^k B(X^k) \left( I_{n \times n} - \frac{11^\top}{n} \right) \]

Objective convergence:

\[ \tau(X^{k+1}, X^{k+1}) \leq \tau(X^k, X^k) \]
SMACOF: Single Step

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

$$X^{k+1} = X^k B(X^k) \left( I_{n \times n} - \frac{11^\top}{n} \right)$$

Majorization-Minimization (MM) algorithm

Objective convergence:

$$\tau(X^{k+1}, X^{k+1}) \leq \tau(X^k, X^k)$$
Graph Embedding

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.

Image from “Data Visualization with Multidimensional Scaling” (Buja et al.)
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 reference model is applied to the other model to obtain the style of the first model. Right: shape-from-eigenvectors problem.
Cares more about preserving small distances

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

“Sammon mapping”

Only Scratching the Surface
Change in Perspective

Extrinsic embedding
All distances equally important

Intrinsic embedding
Locally distances more important
Theory: These Problems are Linked

**Theorem** (Whitney embedding theorem). Any smooth, real $k$-dimensional manifold maps smoothly into $\mathbb{R}^{2k}$.

**Theorem** (Nash–Kuiper embedding theorem, simplified). Any $k$-dimensional Riemannian manifold admits an isometric, differentiable embedding into $\mathbb{R}^{2k}$.

Image: HEVEA Project/PNAS
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

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 $v$
  dist[v][v] ← 0

for each edge $(u, v)$
  dist[u][v] ← 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 dist[i][j] > dist[i][k] + dist[k][j]
        dist[i][j] ← dist[i][k] + dist[k][j]
    end if
  end for
end for
Landmark ISOMAP

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

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

- MDS on landmarks
  - Smaller $n \times n$ problem

- Closed-form embedding formula
  - $\delta(x)$ vector of squared distances from $x$ to landmarks

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

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

- Analysis step: Compute weights \(W_{ij}\)
  \[
  \min_{\omega_1, \ldots, \omega_k} \left\| x_i - \sum_j \omega_j n_j \right\|_2^2
  \text{subject to} \quad \sum_j \omega_j = 1
  \]

- Embedding step: Minimum eigenvalue problem
  \[
  \min_Y \quad \| Y - YW^\top \|_{\text{Fro}}^2
  \text{subject to} \quad YY^\top = I_{p \times p}
  \quad Y1 = 0
  \]
## 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:* \( K(x, y) := e^{-\|x-y\|^2 / \varepsilon} \)

- **Normalize rows**

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

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

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

*(more later)*

---

Mesh Parameterization

<table>
<thead>
<tr>
<th>Name</th>
<th>$\mathcal{D}(\mathbf{J})$</th>
<th>$\mathcal{D}(\sigma)$</th>
<th>$(\nabla_S \mathcal{D}(S))_i$</th>
<th>$(S_A)_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric Dirichlet</td>
<td>$|\mathbf{J}|_F + |\mathbf{J}^{-1}|_F$</td>
<td>$\sum_{i=1}^n (\sigma_i^2 + \sigma_i^{-2})$</td>
<td>$2(\sigma_i - \sigma_i^{-3})$</td>
<td>1</td>
</tr>
<tr>
<td>Exponential Symmetric Dirichlet</td>
<td>$\exp(s(|\mathbf{J}|_F + |\mathbf{J}^{-1}|_F))$</td>
<td>$\exp(s\sum_{i=1}^n (\sigma_i^2 + \sigma_i^{-2}))$</td>
<td>$2s(\sigma_i - \sigma_i^{-3})\exp(s(\sigma_i^2 + \sigma_i^{-2}))$</td>
<td>1</td>
</tr>
<tr>
<td>Hencky strain</td>
<td>$|\log\mathbf{J}|_F$</td>
<td>$\sum_{i=1}^n (\log^2\sigma_i)$</td>
<td>$2(\log\sigma_i^2)$</td>
<td>1</td>
</tr>
<tr>
<td>AMIPS</td>
<td>$\exp(s \cdot \frac{1}{2} \frac{\text{tr}(\mathbf{J} \cdot \mathbf{J}^T)}{\det(\mathbf{J})}) + \frac{1}{2}(\det(\mathbf{J}) + \det(\mathbf{J}^{-1}))$</td>
<td>$\exp(s \cdot \frac{1}{2} (\frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1})$</td>
<td>$s \cdot \exp(s \cdot \frac{1}{2} (\frac{\sigma_1}{\sigma_2} - \frac{1}{\sigma_1 + \sigma_2})$</td>
<td>$\frac{1}{\sigma_1 + \sigma_2}$</td>
</tr>
<tr>
<td>Conformal AMIPS 2D</td>
<td>$\frac{\text{tr}(\mathbf{J} \cdot \mathbf{J}^T)}{\det(\mathbf{J})}$</td>
<td>$\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2}$</td>
<td>$\frac{1}{\sigma_1 + \sigma_2}$</td>
<td>$\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$</td>
</tr>
<tr>
<td>Conformal AMIPS 3D</td>
<td>$\frac{\text{tr}(\mathbf{J} \cdot \mathbf{J}^T)}{\det(\mathbf{J})}$</td>
<td>$\frac{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}{(\sigma_1 \sigma_2 \sigma_3)^{\frac{3}{2}}}$</td>
<td>$\frac{-2\sigma_1 \sigma_2 \sigma_3 (\sigma_1 \sigma_2 + \sigma_2 \sigma_3 + \sigma_3 \sigma_1)}{(3\sigma_1 \sigma_2 \sigma_3)^{\frac{3}{2}}}$</td>
<td>$\frac{\sqrt{\sigma_1 \sigma_2 \sigma_3}}{2}$</td>
</tr>
</tbody>
</table>

$\min_x \sum_f A_f \mathcal{D}(J_f(x))$

- Key consideration: Injectivity
- Connection to PDE

Images/table from: Rabinovich et al. “Scalable Locally Injective Mappings.”
Line search: Smith & Schaefer. “Bijective Parameterization with Free Boundaries.”
Embedding from Geodesic Distance

On reconstruction of non-rigid shapes with intrinsic regularization

Yohai S. Devir      Guy Rosman      Alexander M. Bronstein      Michael M. Bronstein
Ron Kimmel
{ydevir,rosman,bron,mbron}@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 computer graphics, computer animation, and computer vision, 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, an unknown object is needed.

Of particular interest are reconstructing non-rigid shapes. The world contains objects such as live bodies, paper, 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 model or a prior range of shapes that the unknown object may approach. We call this type of problem shape-from-X.
Relative Distance Embedding

ASIF: coupled data turns unimodal models to multimodal without training

Antonio Norelli, Marco Fumero, Valentina Maiorca, Luca Moschella, Emanuele Rodolò, Francesco Locatello

Published: 31 Feb 2023, Last Modified: 13 Feb 2023, Submitted to ICLR 2023
Keywords: Representation learning, Multimodal models, Ancology, Sparsity, Relative representations
TLDR: How to build a CLIP-like model with two pretrained encoders and a limited amount of image-text pairs without tuning a neuron.

Abstract: Aligning the visual and language spaces requires to train deep neural networks from scratch on giant multimodal datasets. CLIP trains both an image and a text encoder while LT manages to train just the latter by taking advantage of a pretrained vision network. In this paper, we show that spanner relative representations are sufficient to align text and images without training any network. Our method relies on readily available single-domain encoders trained with or without supervision and a modest (in comparison) number of image-text pairs. ASIF redefines what constitutes a multimodal model by explicitly disentangling memory from processing: here the model is defined by the embedded pairs of all the entries in the multimodal dataset, in addition to the parameters of the two encoders. Experiments on standard zero-shot visual benchmarks demonstrate the typical transfer ability of image-text models. Overall, our method represents a simple yet surprisingly strong baseline for foundation multimodal models, raising important questions on their data efficiency and on the role of retrieval in machine learning.

Anonymous URL: I certify that there is no URL in (e.g., github page) that could be used to find authors’ identity.

No Acknowledgement Section: I certify that there is no acknowledgement section in this submission for double blind review.

Code Of Ethics: I acknowledge that I and all co-authors of this work have read and commit to adhering to the ICLR Code of Ethics
Submission Guidelines: Yes

Please Choose The Closest Area That Your Submission Falls Into: Deep Learning and representational learning

ASIF recipe. Ingredients:

- Two good encoders, each mapping a single data modality to a vector space. Let \( X \) and \( Y \) be the mode domains, for instance a pixel space and a text space, we need \( E_1 : X \to \mathbb{R}^{d_1} \) and \( E_2 : Y \to \mathbb{R}^{d_2} \).

- A collection of ground truth multimodal pairs: \( D = \{(x_1, y_1), \ldots, (x_n, y_n)\} \), for instance captioned images.

Procedure to find the best caption among a set of original ones \( \hat{Y} = \{\hat{y}_1, \ldots, \hat{y}_c\} \) for a new image \( x^* \):

1. Compute and store the embeddings of the multimodal dataset \( D \) with the encoders \( E_1, E_2 \) and discard \( D \). Now in memory there should be just \( D_E = \{(E_1(x_1), E_2(y_1)), \ldots, (E_1(x_n), E_2(y_n))\} \).

2. Compute the \( n \)-dimensional relative representation for each candidate caption \( r_i(\hat{y}_i) = (\sin(E_2(\hat{y}_i)), \ldots, \sin(E_2(\hat{y}_i))) \), where \( \sin \) is a similarity function, e.g. cosine similarity. Then for each \( r_i(\hat{y}_i) \) set to zero all dimensions except for the highest \( k \), and raise them to \( p \geq 1 \). Finally normalize and store the processed \( c \) vectors \( r_i(\hat{y}_i) \). Choose \( k \) and \( p \) to taste, in our experiments \( k = 800 \) and \( p = 8 \).

3. Compute the relative representation of \( x^* \) using the other half of the embedded multimodal dataset \( D_E \) and repeat the same processing with the chosen \( k \) and \( p \).

4. We consider the relative representation of the new image \( x^* \) as if it was the relative representation of its ideal caption \( y^* \), i.e. we define \( r_i(y^*) := r_i(x^*) \). So we choose the candidate caption \( \hat{y}_i \) most similar to the ideal one, with \( i = \text{argmax}_j(\sin(r_i(y^*)), \hat{y}_j)) \).

To assign one of the captions to a different image \( x^{**} \) repeat from step 3.
Huge zoo of embedding techniques.

*Each with different theoretical properties: Try them all!*

*But what if the distance matrix is incomplete or noisy?*
More General: Metric Nearnessness

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

\[
\text{TRIANGLE\_FIXING}(D, \epsilon)
\]

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

Output: \( M = \arg\min_{X \in \mathcal{M}_{N \times N}} \| X - D \|_F^2 \)

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 \}

for each triangle \((i, j, k)\)
\[
b \leftarrow d_{ki} + d_{jk} - d_{ij}
\]
\[
\mu \leftarrow \frac{1}{3}(e_{ij} - e_{jk} - e_{ki} - b)
\]
\[
\theta \leftarrow \min\{-\mu, z_{ijk}\}
\]
\[
e_{ij} \leftarrow e_{ij} - \theta, e_{jk} \leftarrow e_{jk} + \theta, e_{ki} \leftarrow e_{ki} + \theta
\]
\[
z_{ijk} \leftarrow z_{ijk} - \theta
\] \{ Stay within half-space of constraint \}

end for each

\[
\delta \leftarrow \text{sum of changes in the } e_{ij}
\]

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

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

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

(3.2)

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

(3.3)

Euclidean Matrix Completion

\[
\min_G \| H \odot (P(G) - P_0) \|_{\text{Fro}}^2 \\
\text{s.t. } G \succeq 0
\]

Convex program

Maximum Variance Unfolding

\[
\begin{align*}
\max_G \text{tr}(G) \\
\text{s.t. } G &\succeq 0 \\
G_{ii} + G_{jj} - G_{ij} - G_{ji} &= D_{0ij}^2 \forall (i, j, D_{0ij}) \\
G1 &= 0
\end{align*}
\]

Convex program

Challenging Computational Problems

- 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.

1. 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 $\sum_{i,j} g(f(d_{ij}) - f(|x_i - x_j|))$. We assume that $f, g$ are

1. symmetric;
2. monotonically increasing in the absolute values of their arguments;
3. Lipschitz on $[0, 1]$ with constant $\lambda_L$, that is, for $x, y \in [0, 1]$, $|f(x) - f(y)| \leq \lambda_L |x - y|$; 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 $\lambda_L = 2, \lambda_L = 1$, meaning that $\|D - D^*\|_1$ and $\|D - D^*\|_2$ are both hard to minimize over one-dimensional embeddings.
Typical approaches:

- **Parameterize a distance** $d(\cdot, \cdot)$ directly
  
  Example: Mahalanobis metric $d(x, y) := \sqrt{(x - y)^TA(x - y)}, A \succeq 0$

- **Use closed-form distances on a kernel space**
  
  Example: Network embedding $x \mapsto \phi_\theta(x)$
Kernelization

\[ \phi_{\theta} : \text{Data} \rightarrow \mathbb{R}^n \]

Preserve proximity relationships
Useful for downstream tasks
\( \phi_{\theta} \) can be interpreted as a kernel

“Feature vector”
Metric Learning: Example Losses & Constraints

Bound constraints:
\[d(x_i, x_j) \leq u \quad \forall (i, j) \in S\]
\[d(x_i, x_j) \geq \ell \quad \forall (i, j) \in D\]

Hinge loss:
\[\max(0, d(x_i, x_j) - u) \quad \forall (i, j) \in S\]
\[\max(0, \ell - d(x_i, x_j)) \quad \forall (i, j) \in D\]

Triplet loss:
\[\max\left(d(x_i, x_j) - d(x_i, x_k) + \alpha, 0\right) \quad \forall (i, j) \in S, (i, k) \in D\]

From “Metric Learning: A Survey” (Kulis 2013)
Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov  
Google Inc.  
Mountain View  
mikolov@google.com

Ilya Sutskever  
Google Inc.  
Mountain View  
ilyasu@google.com

Kai Chen  
Google Inc.  
Mountain View  
kai@google.com

Greg Corrado  
Google Inc.  
Mountain View  
gcorrado@google.com

Jeffrey Dean  
Google Inc.  
Mountain View  
jeff@google.com

The recently introduced continuous bag-of-words model for learning high-quality distributed word representations has become a popular technique in natural language processing. However, there are several limitations of this model. By subsampling of the five million words in the Google News dataset, we show an efficient estimation of word representations in vector space. An inherent limitation of word representations is their inability to represent idiosyncratic words like “Canada” and “Air.”

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov  
Google Inc., Mountain View, CA  
tmikolov@google.com

Kai Chen  
Google Inc., Mountain View, CA  
kaichen@google.com

Skip-gram architecture: Predict neighborhood of a word

Download the embedding!
Two Simple Ways to Learn Individual Fairness Metrics from Data

Debarghya Mukherjee 1 *  Mikhail Yurochkin 2 *  Moulinath Banerjee 1  Yuekai Sun 1

Abstract

Individual fairness is an intuitive definition of algorithmic fairness that addresses some of the drawbacks of group fairness. Despite its benefits, it depends on a task specific fair metric that encodes our intuition of what is fair and unfair for the ML task at hand, and the lack of a widely accepted fair metric for many ML tasks is the main barrier to broader adoption of individual fairness. In this paper, we present two simple ways to learn fair metrics from a variety of data types. We show empirically that fair training with the learned metrics leads to improved fairness on three machine learning tasks susceptible to gender and racial biases. 1 We also provide theoretical guarantees on the statistical performance of both approaches.

1. Introduction

Machine learning (ML) models are an integral part of modern decision-making pipelines. They are even part of some high-stakes decision support systems in criminal justice, lending, medicine etc. Although replacing humans with ML models in the decision-making process appear to eliminate human biases, there is growing concern about ML fairness and individual fairness. Group fairness divides the feature space into (non-overlapping) protected subsets and imposes invariance of the ML model on the subsets. Most prior work focuses on group fairness because it is amenable to statistical analysis. Despite its prevalence, group fairness suffers from two critical issues. First, it is possible for an ML model that satisfies group fairness to be blatantly unfair with respect to subsets of the protected groups and individuals (Dwork et al., 2011). Second, there are fundamental incompatibilities between seemingly intuitive notions of group fairness (Kleinberg et al., 2016; Chouldechova, 2017).

In light of the issues with group fairness, we consider individual fairness in our work. Intuitively, individually fair ML models should treat similar users similarly. Dwork et al. (2011) formalize this intuition by viewing ML models as maps between input and output metric spaces and defining individual fairness as Lipschitz continuity of ML models. The metric on the input space is the crux of the definition because it encodes our intuition of which users are similar. Unfortunately, individual fairness was dismissed as impractical because there is no widely accepted similarity metric for most ML tasks. In this paper, we take a step towards operationalizing individual fairness by showing it is possible to learn good similarity metrics from data.

The rest of the paper is organized as follows. In Section 2, we introduce the different notions of individual fairness. In Section 3, we outline our methods for learning fair metrics. In Sec-
t-SNE

t-distributed stochastic neighbor embedding

1. Compute probabilities on input data $x_i$

   $p_{j|i} = \frac{\exp(-\|x_i - x_j\|_2^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|_2^2/2\sigma_i^2)}$

   Likelihood of choosing $j$ as a neighbor under Gaussian prior at $i$ ($\sigma$ is perplexity, or variance)

2. Symmetrize

   $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$

2. Optimize for an embedding

   $KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$

   $q_{ij} = \frac{(1 + \|y_i - y_j\|_2^2)^{-1}}{\sum_{k \neq i} (1 + \|y_i - y_k\|_2^2)^{-1}}$

Find low-dimensional points $y_i$ whose heavy-tailed Student $t$-distribution resembles $p$. (Gradient descent!)

[van der Maaten and Hinton 2008]
Heuristic Explanation

Normal vs Cauchy (Students-T) Distribution

Intuition: Overcome *curse of dimensionality*

https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1
Typical Result

https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1
"How to Use t-SNE Effectively" (Wattenberg et al., 2016)
https://distill.pub/2016/misread-tsne/
Another Popular Choice: UMAP

UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction (McInnes, Healy)
Comparison: https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668
Nice article: https://pair-code.github.io/understanding-umap/
Structure-Preserving Embedding

Justin Solomon

6.8410: Shape Analysis
Spring 2023