Distance Metrics and Embeddings

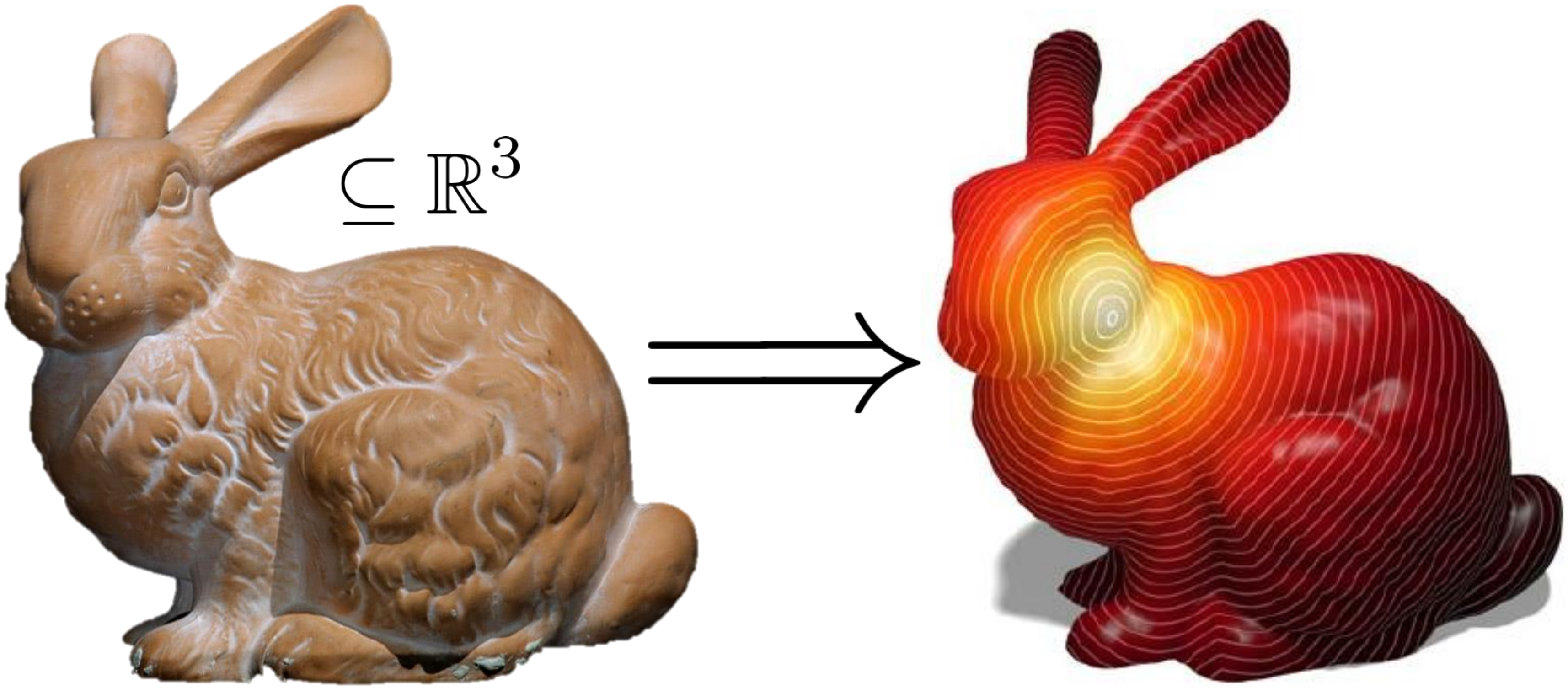
Justin Solomon

6.838: Shape Analysis

Spring 2021

MIT EECS

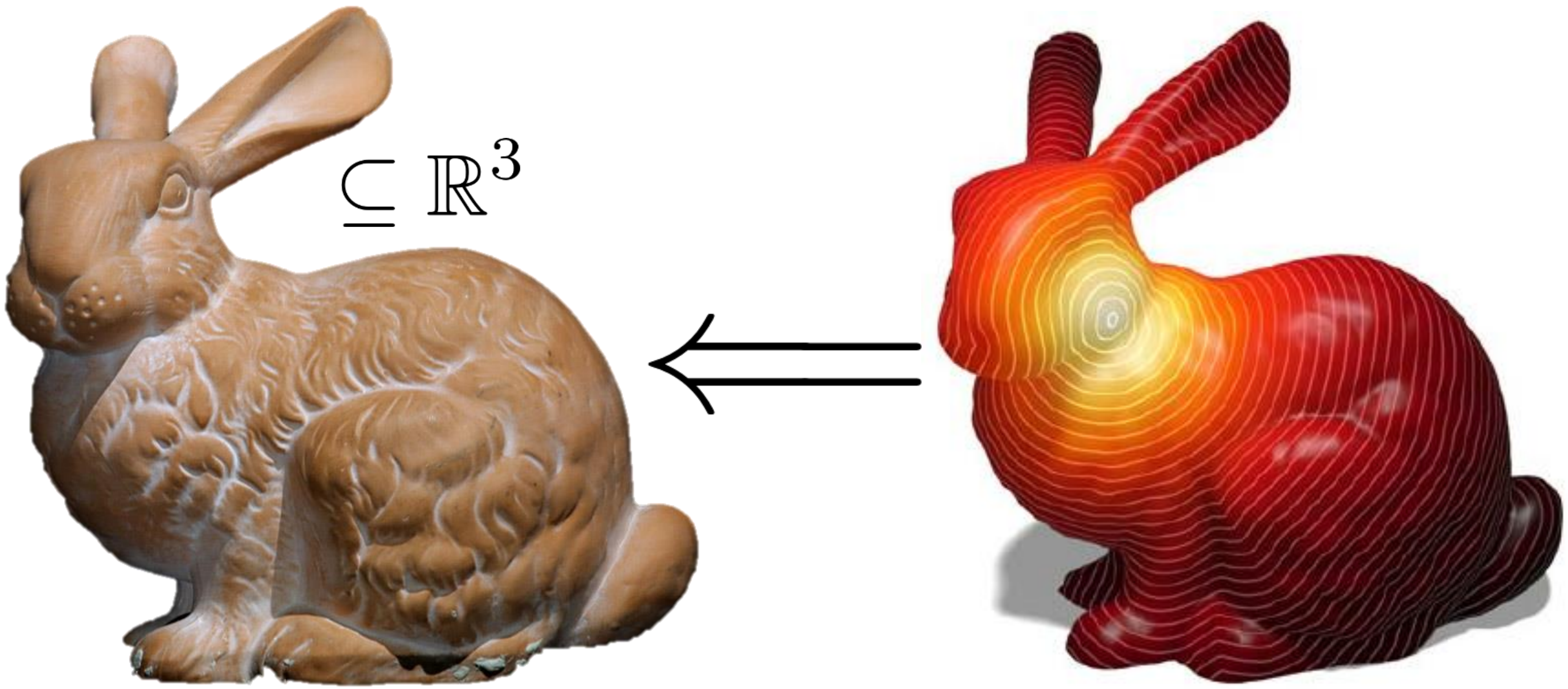
Last Time



Embedding **Geodesic distance**

Right bunny from "Geodesics in Heat" (Crane et al.)

Today



Embedding

Geodesic distance

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

Many Overlapping Tasks

- Dimensionality reduction
- Embedding
- Parameterization
- Manifold learning

…

Basic Task

Given pairwise distances
extract an embedding.

Is it always possible?
Embedding into which space?
What dimensionality?

Metric Space

Ordered pair (M, d) where M is a set and $d: M \times M \to \mathbb{R}$ satisfies

$$d(x,y) \geq 0$$

$$d(x,y) = 0 \iff x = y$$

$$d(x,y) = d(y,x)$$

$$d(x,z) \leq d(x,y) + d(y,z)$$

$$\forall x, y, z \in M$$

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

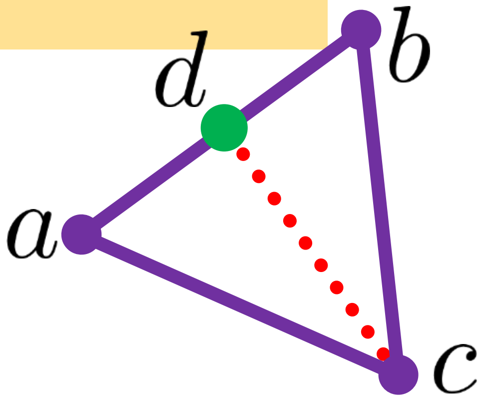
$$X := \{a, b, c, d\}$$

$$d(a, d) = d(b, d) = 1$$

$$d(a, b) = d(a, c) = d(b, c) = 2$$

$$d(c, d) = 1.5$$

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/

Contrasting Example

$$\ell_\infty(\mathbb{R}^n) := (\mathbb{R}^n, \|\cdot\|_\infty)$$
$$\|\mathbf{x}\|_\infty := \max_k |\mathbf{x}_k|$$

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

Extends to infinite-dimensional spaces!

$$\ell_\infty(\mathbb{R}^n) := (\mathbb{R}^n, \|\cdot\|_\infty)$$

$$\|\mathbf{x}\|_\infty := \max_k |\mathbf{x}_k|$$

Approximate Embedding

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

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

$$\text{distortion}(f) := \text{expansion}(f) \times \text{contraction}(f)$$

http://www.cs.toronto.edu/~avner/teaching/S6-2414/LN1.pdf

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].

$m := 576 \log(n)$
for $j = 1$ **to** $\log n$ **do** /* levels of density */
  for $i = 1$ **to** m **do** /* repeat for high probability */
    choose set S_{ij} by sampling each node in X
    independently with probability 2^{-j}
  end
end
$f_{ij}(x) := d(x, S_{ij})$
$f(x) := \bigoplus_{j=1}^{\log n} \bigoplus_{i=1}^{m} f_{ij}(x)$

Uses Fréchet embedding

http://ttic.uchicago.edu/~harry/teaching/pdf/lecture3.pdf

Distance Metrics and Embeddings

Justin Solomon

6.838: Shape Analysis

Spring 2021

Embedding Metrics into Euclidean Space

Justin Solomon

6.838: Shape Analysis

Spring 2021

Recall:

Isometry [ahy-**som**-i-tree]**:**

A map between metric spaces that preserves pairwise distances.

Euclidean Problem

Given:

$$P_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2, P \in \mathbb{R}^{n \times n}$$

Reconstruct:

$$\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^m$$

Alternative notation:

$$X \in \mathbb{R}^{m \times n}$$

Gram Matrix [gram **mey**-triks]**:**

A matrix of inner products

$$X^\top X$$

$$P_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$$

$$G = -\frac{1}{2} J^\top P J$$

Classical Multidimensional Scaling

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

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

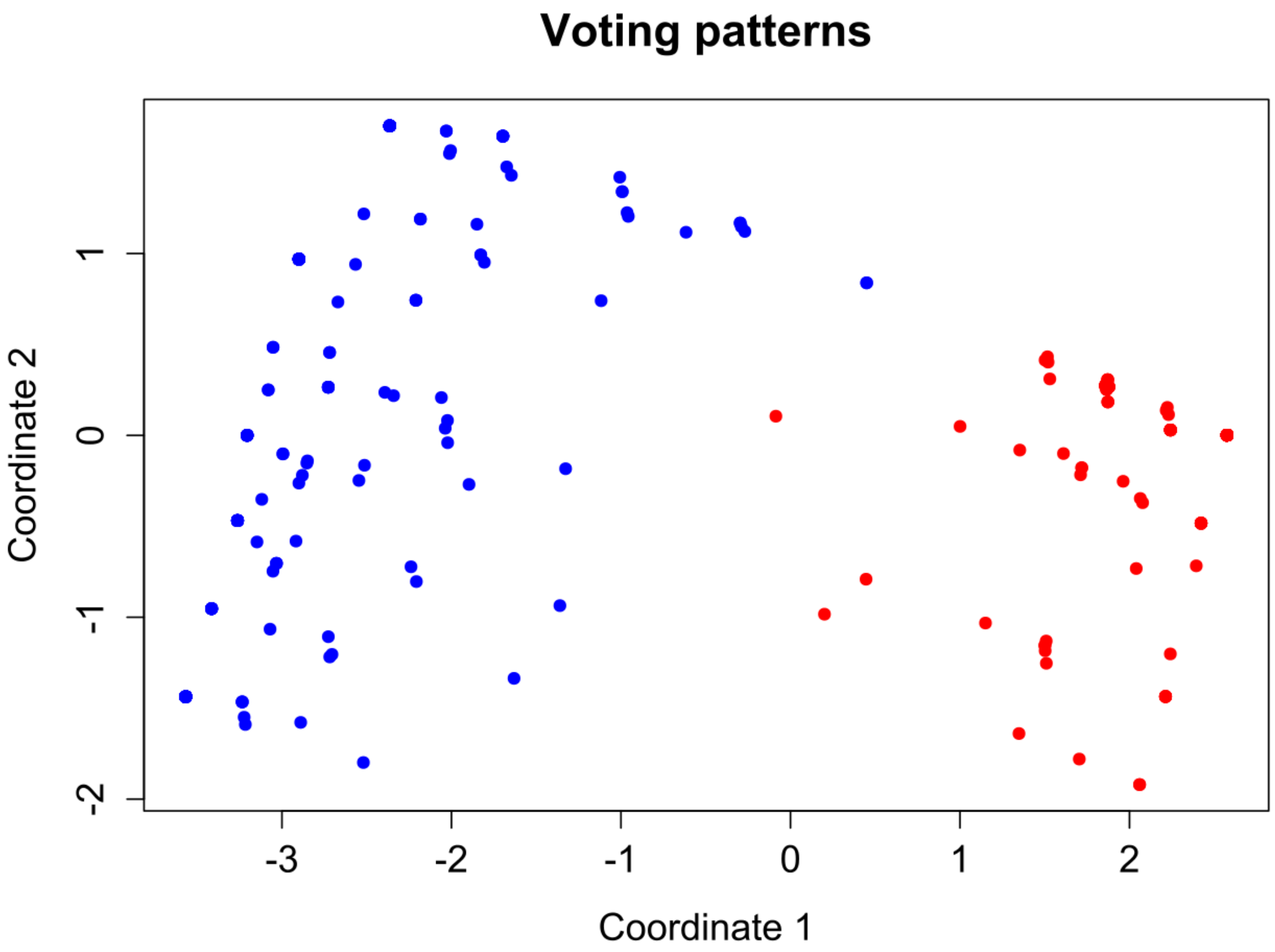
3. $\overline{X} = \sqrt{\Lambda}Q^\top$

“MDS”

Extension: Landmark MDS

Torgerson, Warren S. (1958). *Theory & Methods of Scaling*.

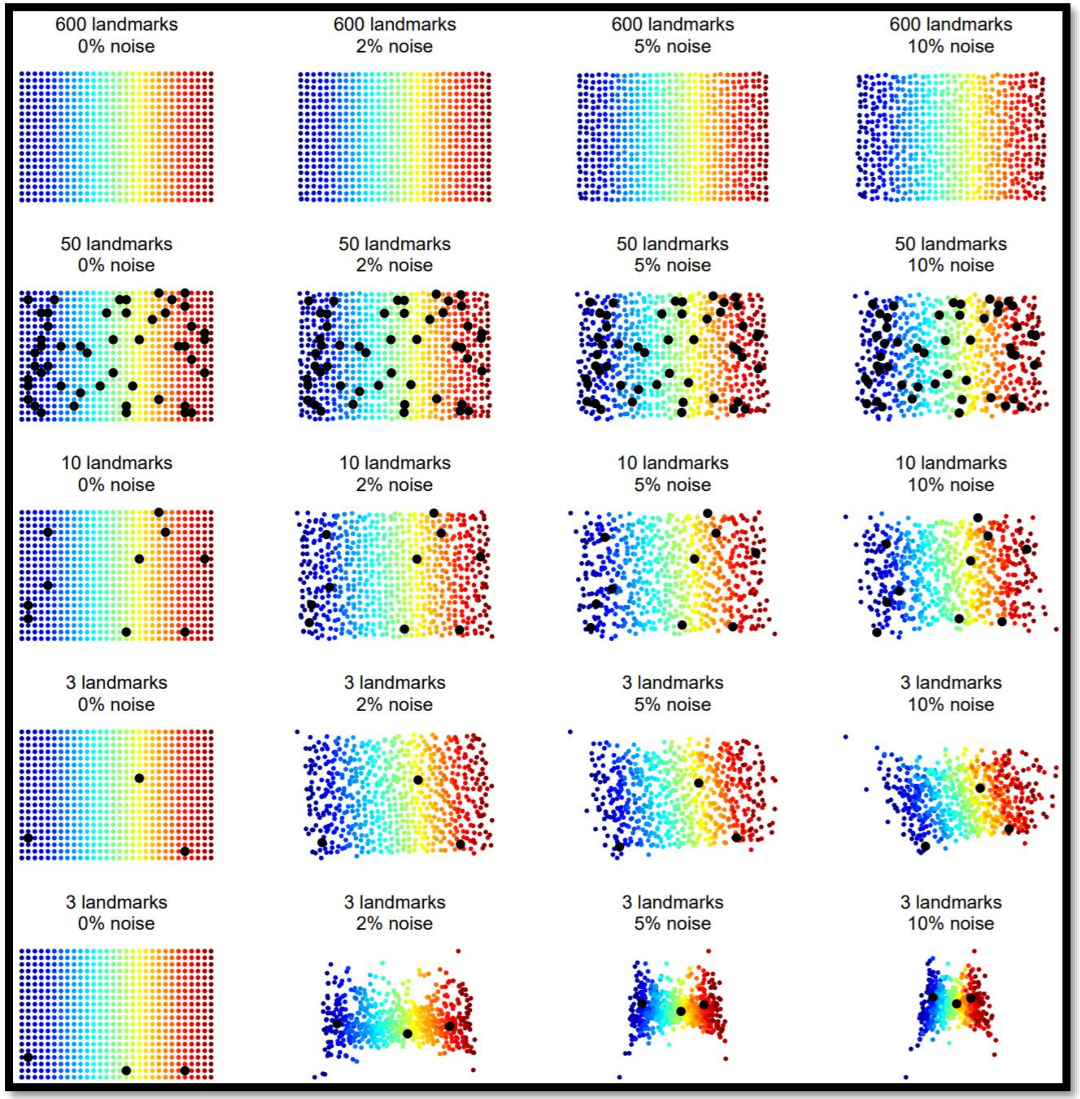
Simple Example



https://en.wikipedia.org/wiki/Multidimensional_scaling#/media/File:RecentVotes.svg

$$\overline{\mathbf{x}}_i = \frac{1}{2}\Lambda^{-1}\overline{X}(\mathbf{p}_i - \mathbf{g})$$

Landmark MDS



$$\overline{\mathbf{x}} = \frac{1}{2}\Lambda^{-1}\overline{X}(\mathbf{p} - \mathbf{g})$$

where p contains squared distances to landmarks.

de Silva and Tenenbaum. (2004). "Sparse Multidimensional Scaling Using Landmark Points." Technical Report, Stanford University, 41.

Stress Majorization

$$\min_X \sum_{ij} \left(D_{0ij} - \|\mathbf{x}_i - \mathbf{x}_j\|_2\right)^2$$

Nonconvex!

SMACOF:

Scaling by Majorizing a Complicated Function

de Leeuw, J. (1977), "Applications of convex analysis to multidimensional scaling" *Recent developments in statistics*, 133–145.

$$\min_X \sum_{ij} \left(D_{0ij} - \|\mathbf{x}_i - \mathbf{x}_j\|_2\right)^2$$

SMACOF Lemma

$$\sum_{ij} (D_{0ij})^2 = \text{const.}$$

$$\sum_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = \text{tr}(XVX^\top), \text{ where } V = 2nJ$$

$$-2\sum_{ij} D_{0ij}\|\mathbf{x}_i - \mathbf{x}_j\|_2 = -2\text{tr}(XB(X)X^\top)$$

$$\text{where } B_{ij}(X) := \begin{cases} -\frac{2D_{0ij}}{\|\mathbf{x}_i - \mathbf{x}_j\|_2} & \text{if } \mathbf{x}_i \neq \mathbf{x}_j, i \neq j \\ 0 & \text{if } \mathbf{x}_i = \mathbf{x}_j, i \neq j \\ -\sum_{j \neq i} B_{ij} & \text{if } i = j \end{cases}$$

Lemma. Define

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

Then,

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

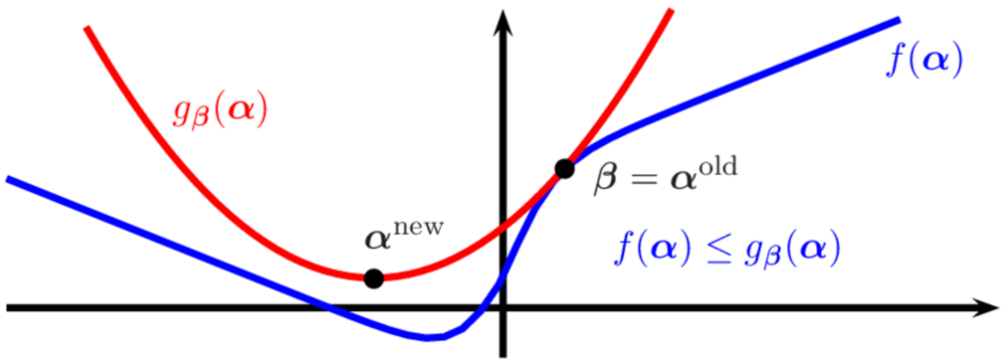
with equality exactly when $X \propto Z$.

Proof using Cauchy-Schwarz.

See *Modern Multidimensional Scaling* (Borg, Groenen)

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

SMACOF: Single Step



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

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

Majorization-Minimization (MM) algorithm

Objective convergence:
$\tau(X^{k+1}, X^{k+1}) \leq \tau(X^k, X^k)$

Image from "Sparse Modeling for Image and Vision Processing" (Mairal, Bach, and Ponce)

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

Graph Embedding

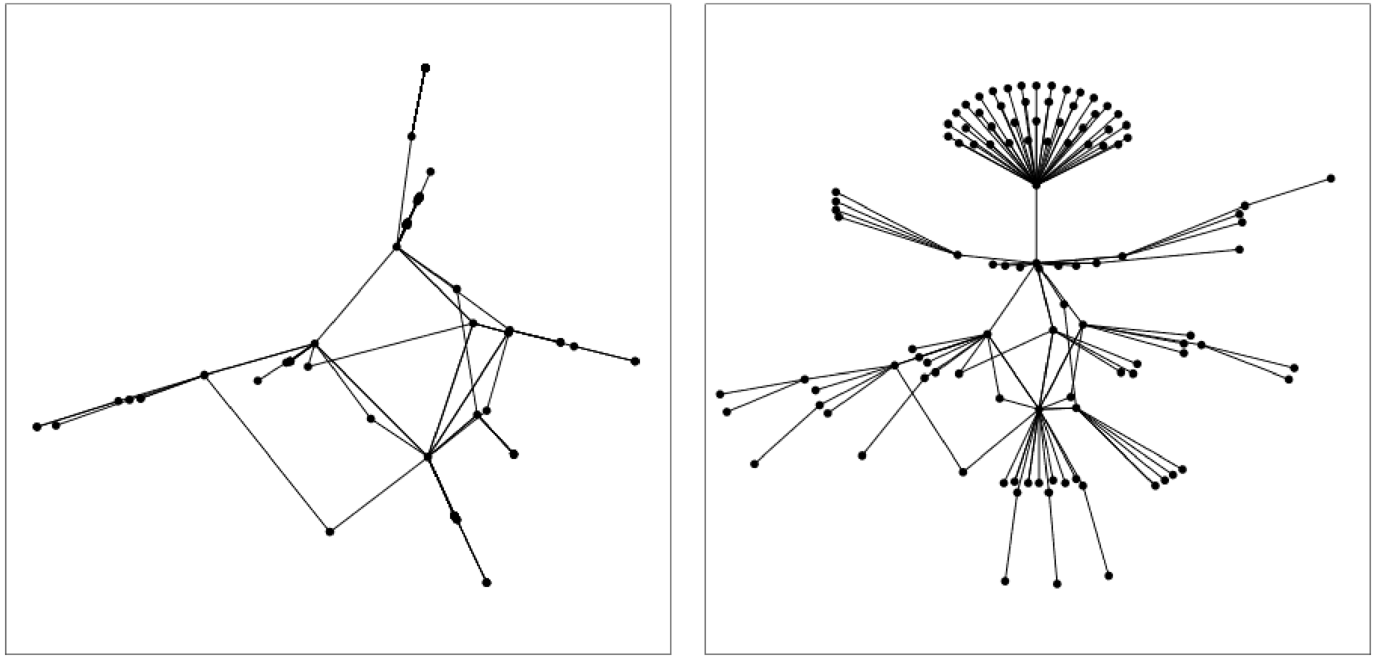


Figure 9: *A Telephone Call Graph, Layed Out in 2-D. Left: classical scaling (Stress=0.34); right: distance scaling (Stress=0.23). The nodes represent telephone numbers, the edges represent the existence of a call between two telephone numbers in a given time period.*

Image from “Data Visualization with Multidimensional Scaling” (Buja et al.)

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

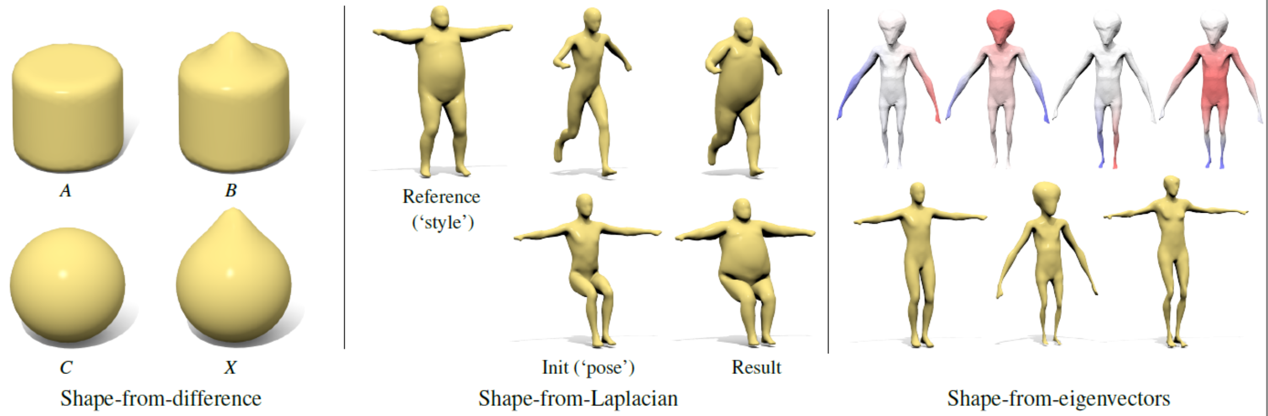
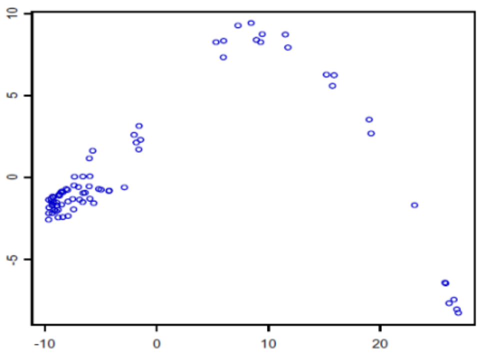


Figure 1: *Examples of three different shape-from-operator problems considered in the paper. Left: shape analogy synthesis as shape-from-difference operator problem (shape X is synthesized such that the intrinsic difference operator between C,X is as close as possible to the difference between A,B).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

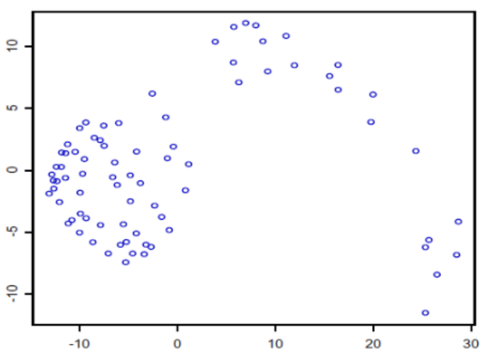
Related Method

$$\min_X \sum_{ij} \frac{\left(D_{0ij} - \|\mathbf{x}_i - \mathbf{x}_j\|_2\right)^2}{D_{0ij}}$$

Cares more about preserving small distances



Classical MDS



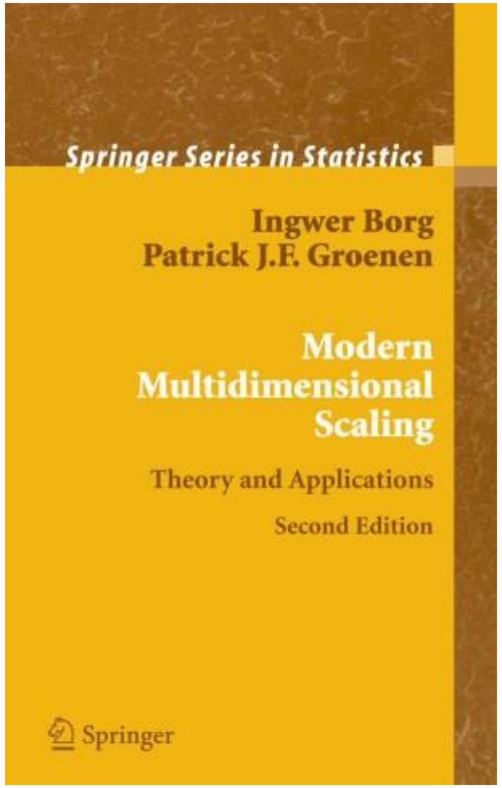
Sammon

"Sammon mapping"

Sammon (1969). "A nonlinear mapping for data structure analysis." IEEE Transactions on Computers 18.

http://www.stat.pitt.edu/sungkyu/course/2221Fall13/lec8_mds_combined.pdf

Only Scratching the Surface



Embedding Metrics into Euclidean Space

Justin Solomon

6.838: Shape Analysis

Spring 2021

Structure-Preserving Embedding

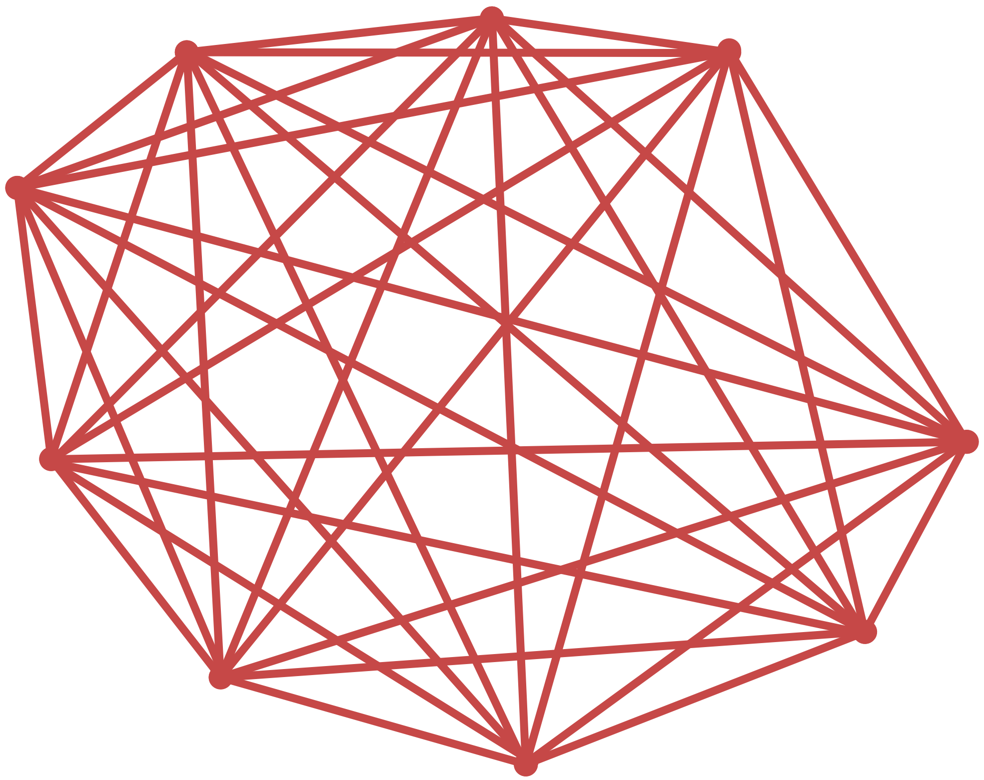
Justin Solomon

6.838: Shape Analysis

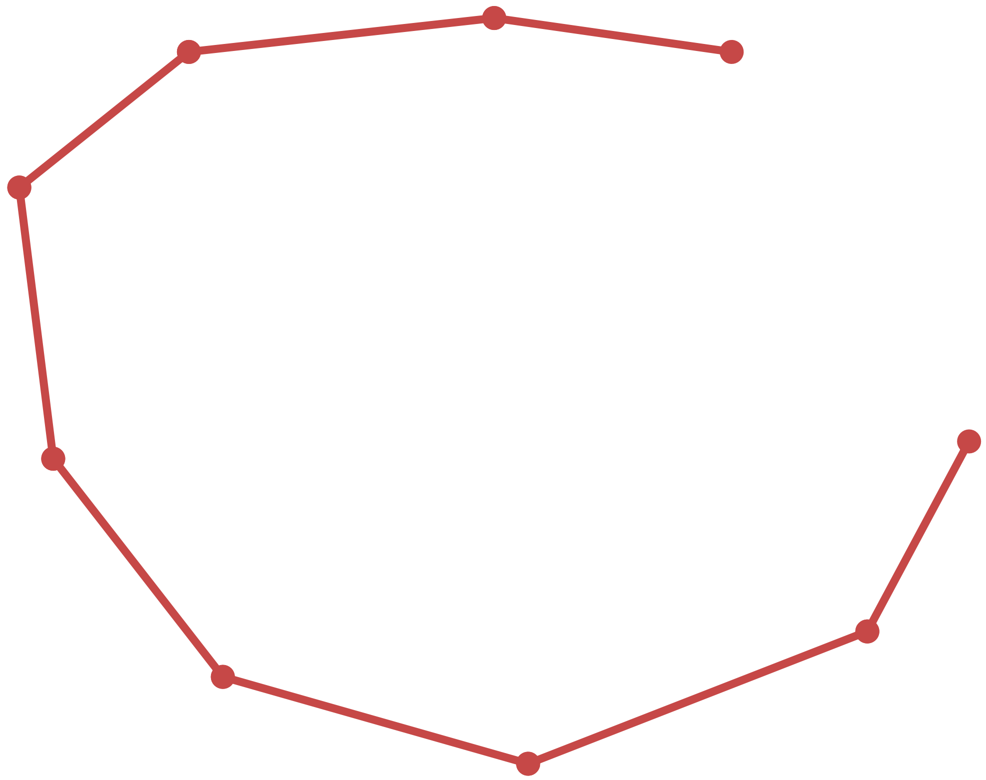
Spring 2021

MIT EECS

Change in Perspective



Extrinsic embedding
All distances equally important

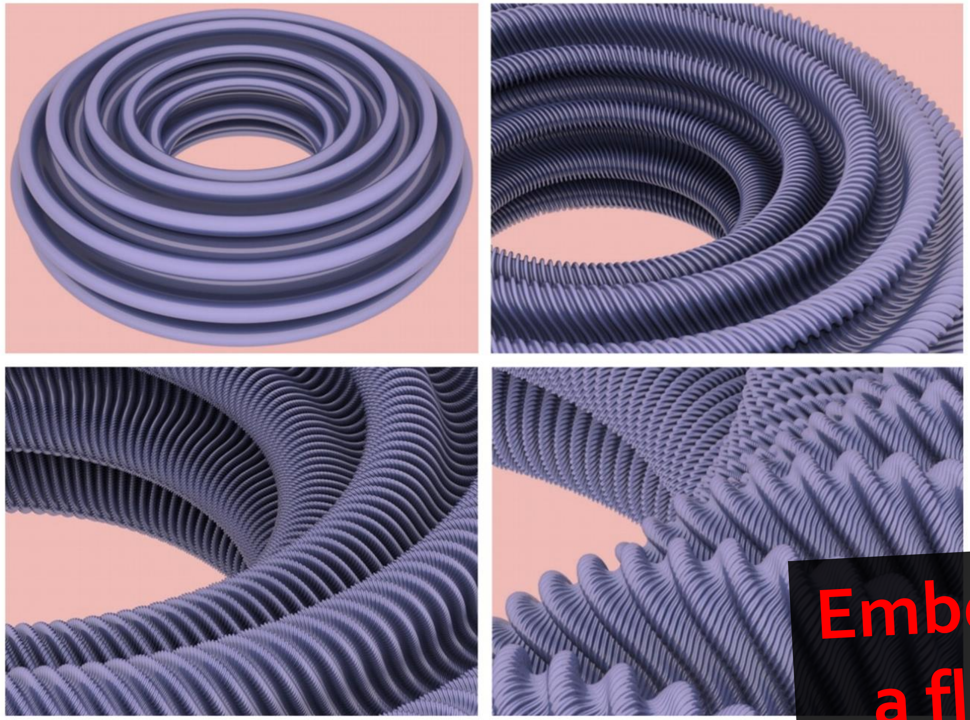


Intrinsic embedding
Locally distances more important

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* $\mathbb{R}^{2k}$.



Embedding of a flat torus

Image: HEVEA Project/PNAS

Intrinsic-to-Extrinsic: ISOMAP

- **Construct neighborhood graph**

  k-nearest neighbor graph or ε-neighborhood graph

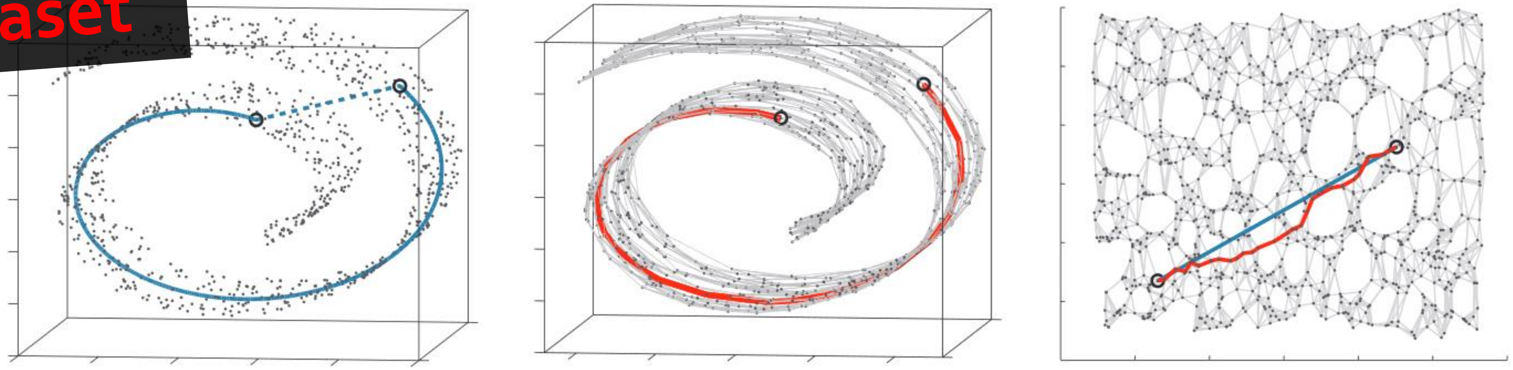
- **Compute shortest-path distances**

  Floyd-Warshall algorithm or Dijkstra

- **Classical MDS**

  Eigenvalue problem

Swiss roll dataset



Tenenbaum, de Silva, Langford.
"A Global Geometric Framework for Nonlinear Dimensionality Reduction." Science (2000).

Floyd-Warshall Algorithm

```
let dist be a |V| × |V| array of minimum distances initialized to ∞ (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
```

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

Landmark ISOMAP

- **Construct neighborhood graph**

  k-nearest neighbor graph or ε-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**

  $\delta(x)$ vector of squared distances from x to landmarks

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

Landmark MDS

Locally Linear Embedding (LLE)

- **Construct neighborhood graph**

  k-nearest neighbor graph or ε-neighborhood graph

- **Analysis step: Compute weights W_{ij}**

$$\begin{aligned} \min_{\omega^1,\ldots,\omega^k} \quad & \left\| \mathbf{x}_i - \sum_j \omega^j \mathbf{n}_j \right\|_2 \\ \text{subject to} \quad & \sum_j \omega^j = 1 \end{aligned}$$

- **Embedding step: Minimum eigenvalue problem**

$$\begin{aligned} \min_Y \quad & \|Y - YW^\top\|_{\text{Fro}}^2 \\ \text{subject to} \quad & YY^\top = I_{p\times p} \\ & Y\mathbf{1} = \mathbf{0} \end{aligned}$$

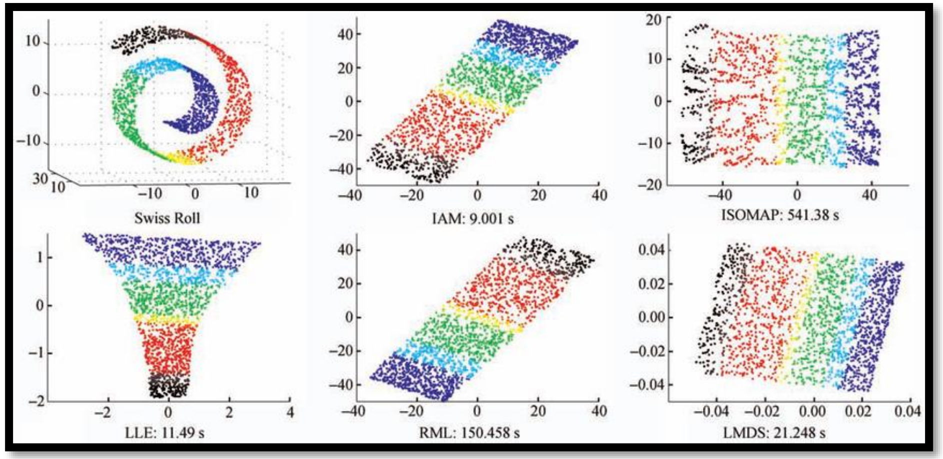
$$\min_{\omega^1,\ldots,\omega^k} \left\| \mathbf{x}_i - \sum_j \omega^j \mathbf{n}_j \right\|_2$$

$$\text{subject to} \quad \sum_j \omega^j = 1$$

$$
\begin{aligned}
\min_Y \quad & \|Y - YW^\top\|_{\text{Fro}}^2 \\
\text{subject to} \quad & YY^\top = I_{p\times p} \\
& Y\mathbf{1} = \mathbf{0}
\end{aligned}
$$

Comparison: ISOMAP vs. LLE

ISOMAP	LLE
Global distances	Local averaging
k-NN graph distances	*k*-NN graph weighting
Largest eigenvectors	Smallest eigenvectors
Dense matrix	Sparse matrix



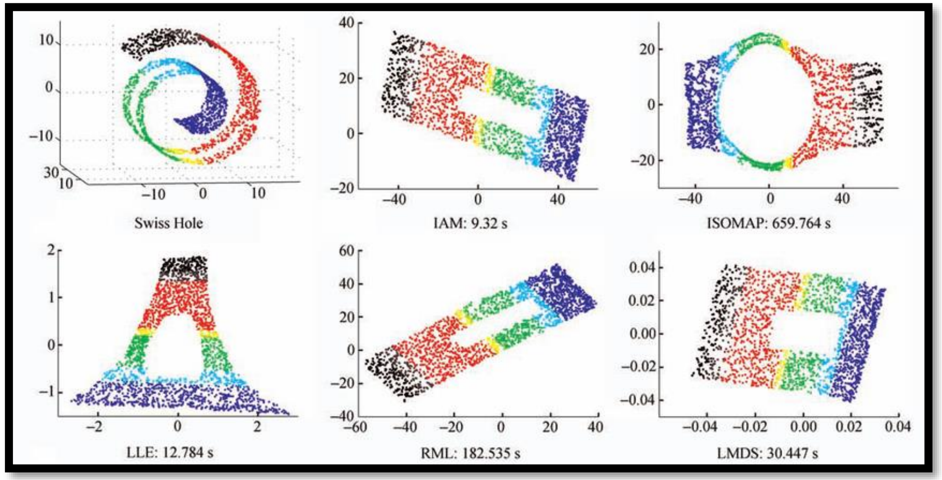


Image from "Incremental Alignment Manifold Learning." Han et al. JCST 26.1 (2011).

Other option:

Diffusion Maps

- **Construct similarity matrix**

  Example: $K(x, y) := e^{-\|x-y\|^2/\varepsilon}$

- **Normalize rows**

  $$M := D^{-1}K$$

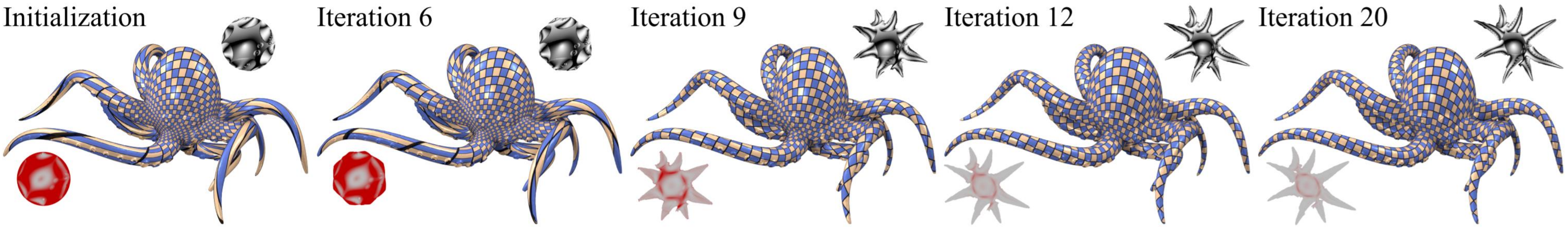
- **Embed from *k* largest eigenvectors**

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

(more later)

Coifman, R.R.; S. Lafon. (2006). "Diffusion maps." *Applied and Computational Harmonic Analysis*. 21: 5–30.

Mesh Parameterization



Name	$\mathcal{D}(\mathbf{J})$	$\mathcal{D}(\sigma)$	$(\nabla_{\mathbf{S}}\mathcal{D}(\mathbf{S}))_i$	$(\mathbf{S}_\Lambda)_i$
Symmetric Dirichlet	$\|\mathbf{J}\|_F^2 + \|\mathbf{J}^{-1}\|_F^2$	$\sum_{i=1}^n(\sigma_i^2+\sigma_i^{-2})$	$2(\sigma_i - \sigma_i^{-3})$	1
Exponential Symmetric Dirichlet	$\exp(s(\|\mathbf{J}\|_F^2 + \|\mathbf{J}^{-1}\|_F^2))$	$\exp(s\sum_{i=1}^n(\sigma_i^2+\sigma_i^{-2}))$	$2s(\sigma_i-\sigma_i^{-3})\exp(s(\sigma_i^2+\sigma_i^{-2}))$	1
Hencky strain	$\|\log \mathbf{J}^\top\mathbf{J}\|_F^2$	$\sum_{i=1}^n(log^2\sigma_i)$	$2(\frac{\log\sigma_i}{\sigma_i})$	1
AMIPS	$\exp(s\cdot\frac{1}{2}(\frac{\mathrm{tr}(\mathbf{J}^\top\mathbf{J})}{\det(\mathbf{J})} + \frac{1}{2}(\det(\mathbf{J})+\det(\mathbf{J}^{-1})))$	$\exp(s(\frac{1}{2}(\frac{\sigma_1}{\sigma_2}+\frac{\sigma_2}{\sigma_1}) + \frac{1}{4}(\sigma_1\sigma_2+\frac{1}{\sigma_1\sigma_2}))$	$s\cdot\exp(s\cdot(\frac{1}{4}(\sigma_{i+1}-\frac{1}{\sigma_{i+1}\sigma_i^2}) + \frac{1}{2}(\frac{1}{\sigma_{i+1}}-\frac{\sigma_{i+1}}{\sigma_i^2}))$	$\sqrt{\frac{2\sigma_{i+1}^2+1}{\sigma_{i+1}^2+2}}$
Conformal AMIPS 2D	$\frac{\mathrm{tr}(\mathbf{J}^\top\mathbf{J})}{\det(\mathbf{J})}$	$\frac{\sigma_1^2+\sigma_2^2}{\sigma_1\sigma_2}$	$\frac{1}{\sigma_{i+1}}-\frac{\sigma_{i+1}}{\sigma_i^2}$	$\sqrt{\sigma_1\sigma_2}$
Conformal AMIPS 3D	$\frac{\mathrm{tr}(\mathbf{J}^\top\mathbf{J})}{\det(\mathbf{J})^{\frac{2}{3}}}$	$\frac{\sigma_1^2+\sigma_2^2+\sigma_3^2}{(\sigma_1\sigma_2\sigma_3)^{\frac{2}{3}}}$	$\frac{-2\sigma_{i+1}\sigma_{i+2}(\sigma_{i+1}^2+\sigma_{i+2}^2-2\sigma_i^2)}{(3\sigma_i\sigma_{i+1}\sigma_{i+2})^{\frac{5}{3}}}$	$\sqrt{\frac{\sigma_1^2+\sigma_3^2}{2}}$

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

- 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
{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

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 underdetermined, 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 p… unknown object is needed.

Of particular interest are recon… ing non-rigid shapes. The world … objects such as live bodies, pape… 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

The numerical core of the proposed method is based on differentiating the fast marching update step for geodesic distance computation.

Take-Away

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 Nearness

$$\min_{X \in \mathcal{M}_{N \times N}} \|X - D\|_{\text{Fro}}^2$$

TRIANGLE_FIXING(D, ϵ)
Input: Input dissimilarity matrix D, tolerance ϵ
Output: $M = \text{argmin}_{X \in \mathscr{M}_N} \|X - D\|_2$.
for $1 \le i < j < k \le n$
  $(z_{ijk}, z_{jki}, z_{kij}) \leftarrow 0$
for $1 \le i < j \le n$
  $e_{ij} \leftarrow 0$
$\delta \leftarrow 1 + \epsilon$
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}
    $e_{ij} \leftarrow e_{ij} - \theta$, $e_{jk} \leftarrow e_{jk} + \theta$, $e_{ki} \leftarrow e_{ki} + \theta$
    $z_{ijk} \leftarrow z_{ijk} - \theta$ {Update correction term}
  end *foreach*
  $\delta \leftarrow$ sum of changes in the e
end *while*
return $M = D + E$

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

$$\begin{aligned} &\min_{e'} \tfrac{1}{2}[(e'_{ij} - e_{ij})^2 + (e'_{jk} - e_{jk})^2 + (e'_{ki} - e_{ki})^2)], \\ &\text{subject to} \quad e'_{ij} - e'_{jk} - e'_{ki} = b_{ijk}. \end{aligned} \tag{3.2}$$

It is easy to check that the solution is given by

$$e'_{ij} \leftarrow e_{ij} - \mu_{ijk}, \quad e'_{jk} \leftarrow e_{jk} + \mu_{ijk}, \quad \text{and} \quad e'_{ki} \leftarrow e_{ki} + \mu_{ijk}, \tag{3.3}$$

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

Iterative projection

Dhillon, Sra, Tropp. "Triangle Fixing Algorithms for the Metric Nearness Problem." NIPS 2004.

Euclidean Matrix Completion

$$\min_{G} \|H \circ (P(G) - P_0)\|_{\text{Fro}}^2$$

$$\text{s.t. } G \succeq 0$$

Convex program

Alfakih, Khandani, and Wolkowicz. "Solving Euclidean distance matrix completion problems via semidefinite programming." Comput. Optim. Appl., 12 (1999).

Maximum Variance Unfolding

$$\max_G \ \mathrm{tr}(G)$$

$$\text{s.t. } G \succeq 0$$

$$G_{ii} + G_{jj} - G_{ij} - G_{ji} = D_{0ij}^2 \ \forall (i, j, D_{0ij})$$

$$G\mathbf{1} = \mathbf{0}$$

Convex program

Alfakih, Khandani, and Wolkowicz. "Solving Euclidean distance matrix completion problems via semidefinite programming." Comput. Optim. Appl., 12 (1999).

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?

NP-Hardness Result

Robust Euclidean Embedding

Lawrence Cayton LCAYTON@CS.UCSD.EDU
Sanjoy Dasgupta DASGUPTA@CS.UCSD.EDU
Department of Computer Science and Engineering, University of California, San Diego
9500 Gilman Dr. La Jolla, CA 92093

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.

choice for embedding seems to be c
sional scaling (cMDS). Its popula
ing relatively fast, parameter-free
and optimal for its cost function. In this work, we
look carefully at the algorithm and a
has some problematic features as we
we argue that the cost function is n
conceptually awkward.

We propose a robust alternative to cM
clidean embedding (REE), that reta
desirable features of cMDS, but av
pitfalls. We show that the global
REE cost function can be found u
nite program (SDP). Though this is
dard SDP-solvers can only manage the
gram for around 100 points. So th
used on more reasonably sized data
a subgradient-based implementation

Dimensionality reduction is an important application

ℓ_1 EUCLIDEAN EMBEDDING
Input: A dissimilarity matrix $D = (d_{ij})$.
Output: An embedding into the line: $x_1, x_2, \ldots \in \mathbf{R}$
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 $\sum_{i,j} g\big(f(d_{ij}) - f(|x_i - x_j|)\big)$ 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 λ_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.

Metric Learning

Typical approaches:

- **Parameterize a distance $d(\cdot,\cdot)$ directly**

  Example: Mahalanobis metric $d(x,y) \coloneqq \sqrt{(x-y)^\top A(x-y)}, A \succcurlyeq 0$

- **Use closed-form distances on a kernel space**

  Example: Network embedding $x \mapsto \phi_\theta(x)$

Kernelization

$$\phi_\theta : \text{Data} \to \mathbb{R}^n$$

Preserve proximity relationships
Useful for downstream tasks
ϕ_θ can be interpreted as a kernel

"Feature vector"

Metric Learning: Example Losses & Constraints

Bound constraints:

$$d(\mathbf{x}_i, \mathbf{x}_j) \leq u \quad \forall (i,j) \in \mathcal{S}$$
$$d(\mathbf{x}_i, \mathbf{x}_j) \geq \ell \quad \forall (i,j) \in \mathcal{D}$$

Hinge loss:

$$\max(0, d(\mathbf{x}_i, \mathbf{x}_j) - u) \quad \forall (i,j) \in \mathcal{S}$$
$$\max(0, \ell - d(\mathbf{x}_i, \mathbf{x}_j)) \quad \forall (i,j) \in \mathcal{D}$$

Triplet loss:

$$\max\left(d(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{x}_i, \mathbf{x}_k) + \alpha, 0\right)$$
$$\forall (i,j) \in \mathcal{S}, (i,k) \in \mathcal{D}$$

From "Metric Learning: A Survey" (Kulis 2013)

Well-Known Example: Word2Vec

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

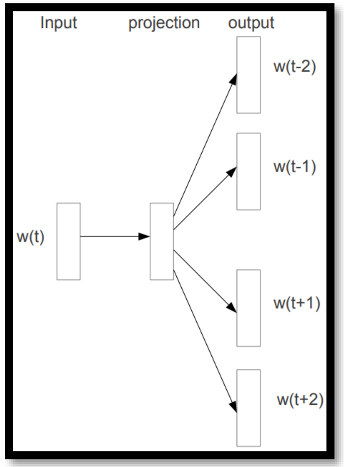
Jeffrey Dean
Google Inc.
Mountain View
jeff@google.com

The recently introduced continuo
learning high-quality distributed v
ber of precise syntactic and seman
several extensions that improve b
speed. By subsampling of the fre
also learn more regular word repr
tive to the hierarchical softmax cal
An inherent limitation of word rep
and their inability to represent idi
"Canada" and "Air" cannot be easi

Efficient Estimation of Word Representations in Vector Space

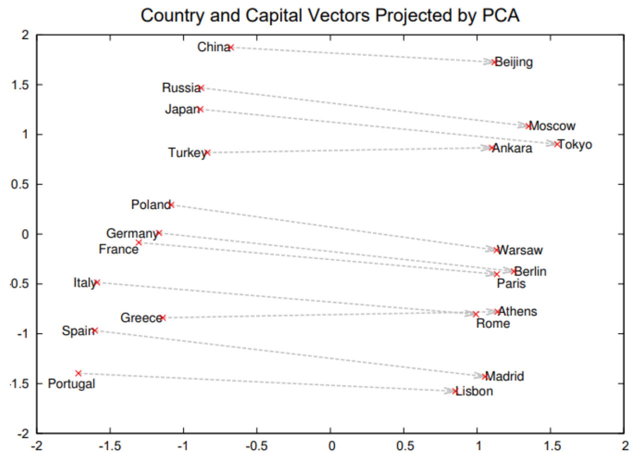
Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

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



Download the embedding!

Skip-gram architecture: Predict neighborhood of a word



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* (σ is **perplexity**, or variance)

2. Symmetrize

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

2. Optimize for an embedding

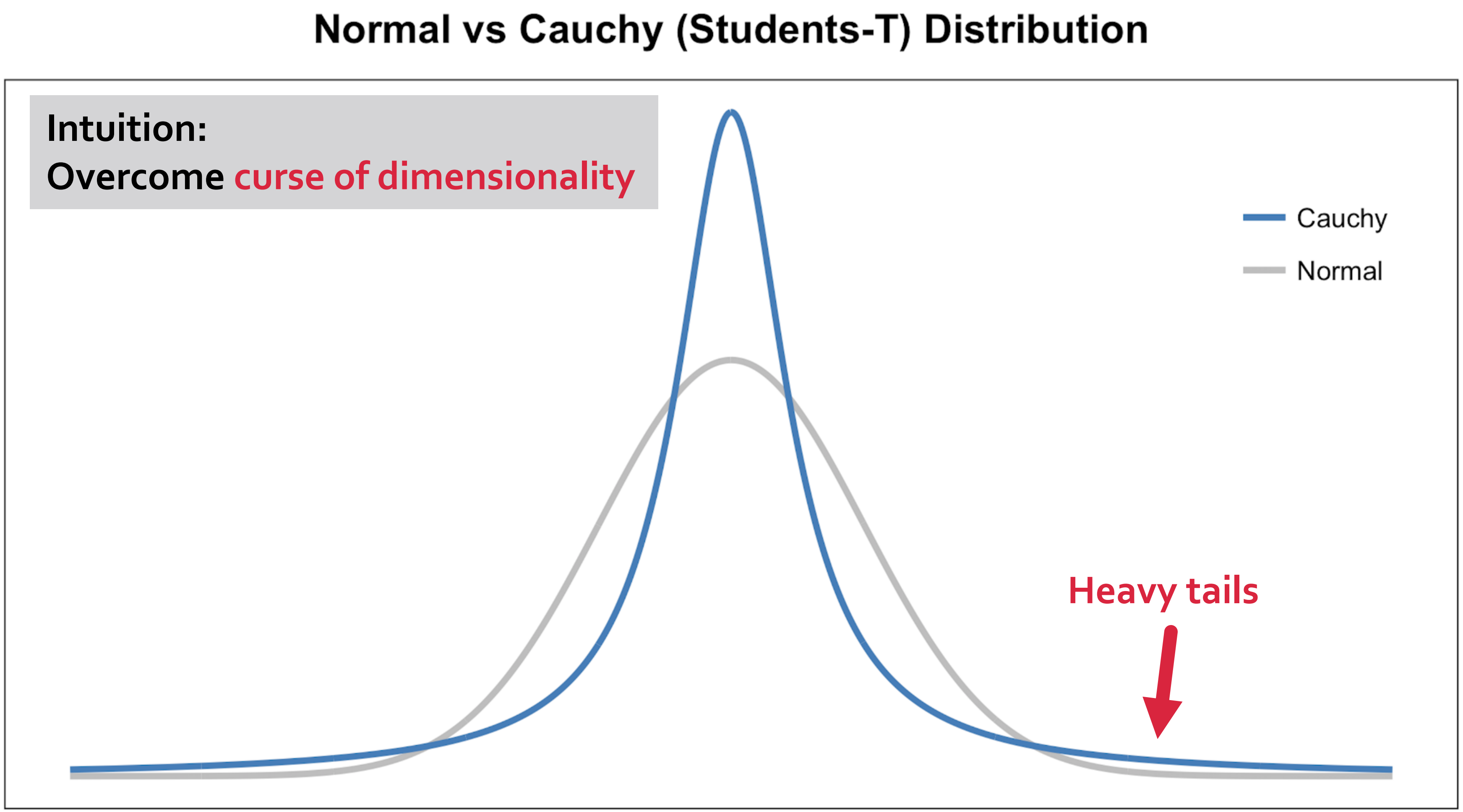
$$\mathrm{KL}(P\|Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

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

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

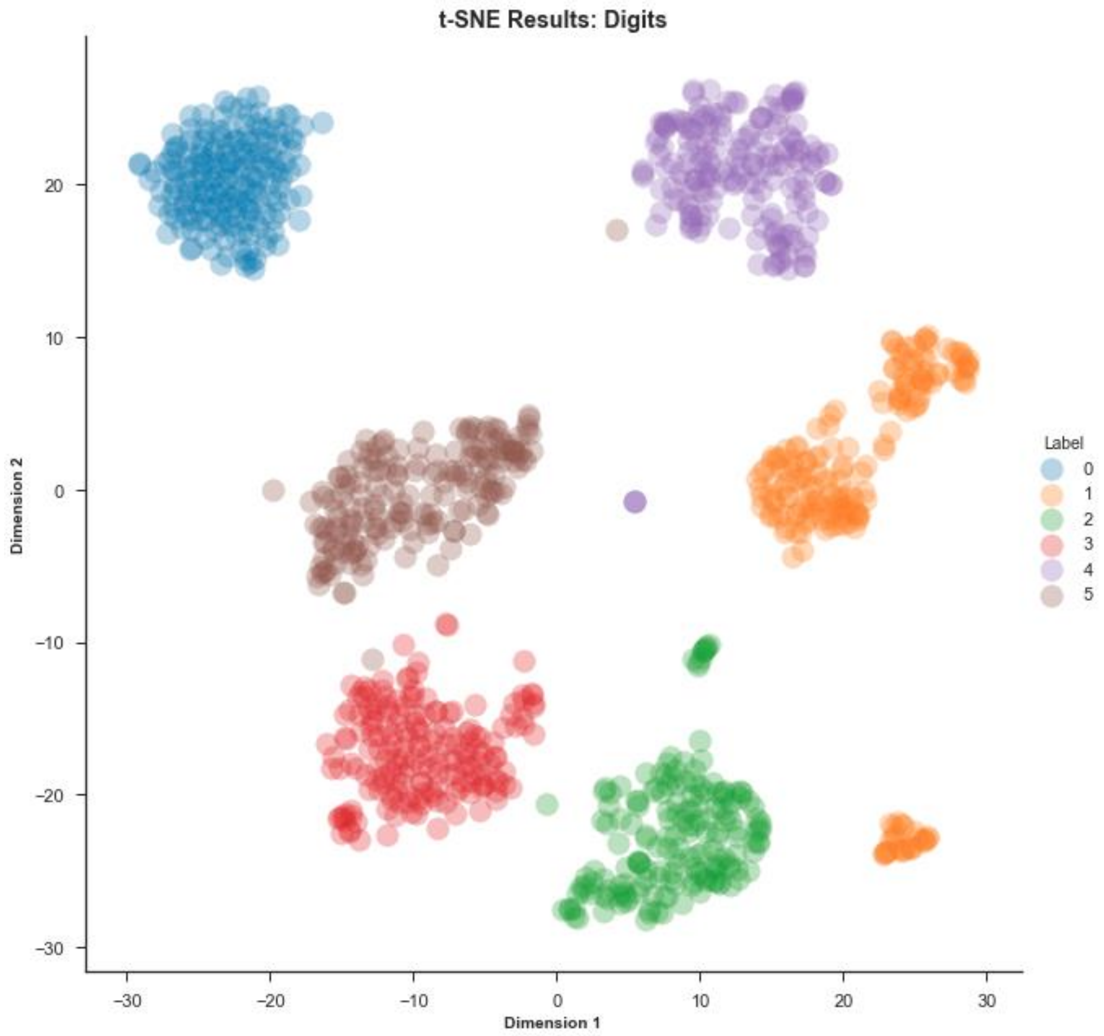
[van der Maaten and Hinton 2008]

Heuristic Explanation



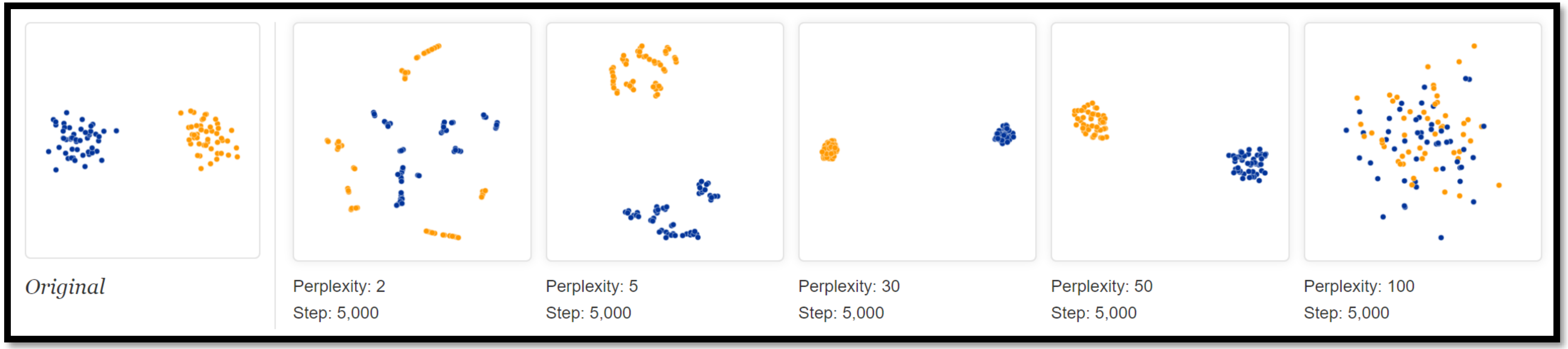
https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1

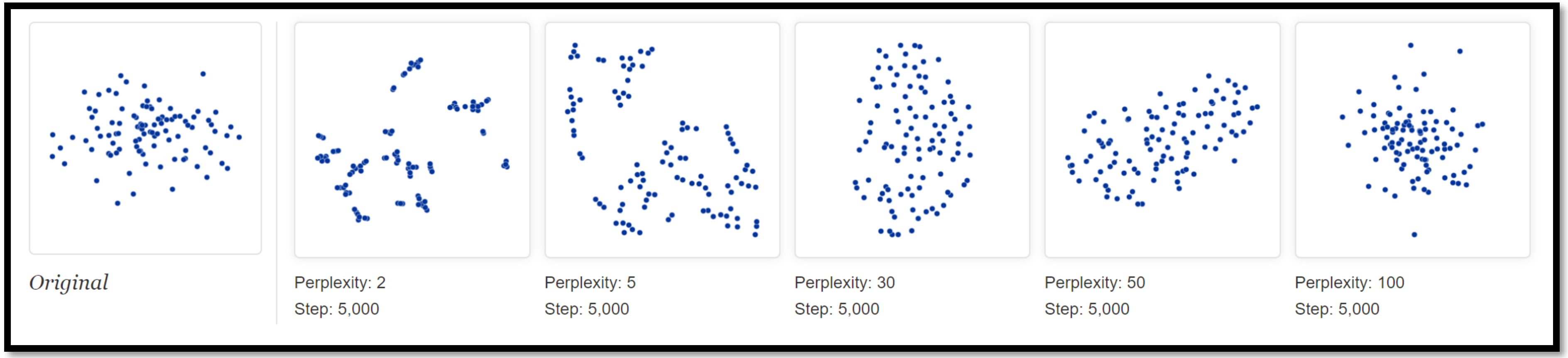
Typical Result



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

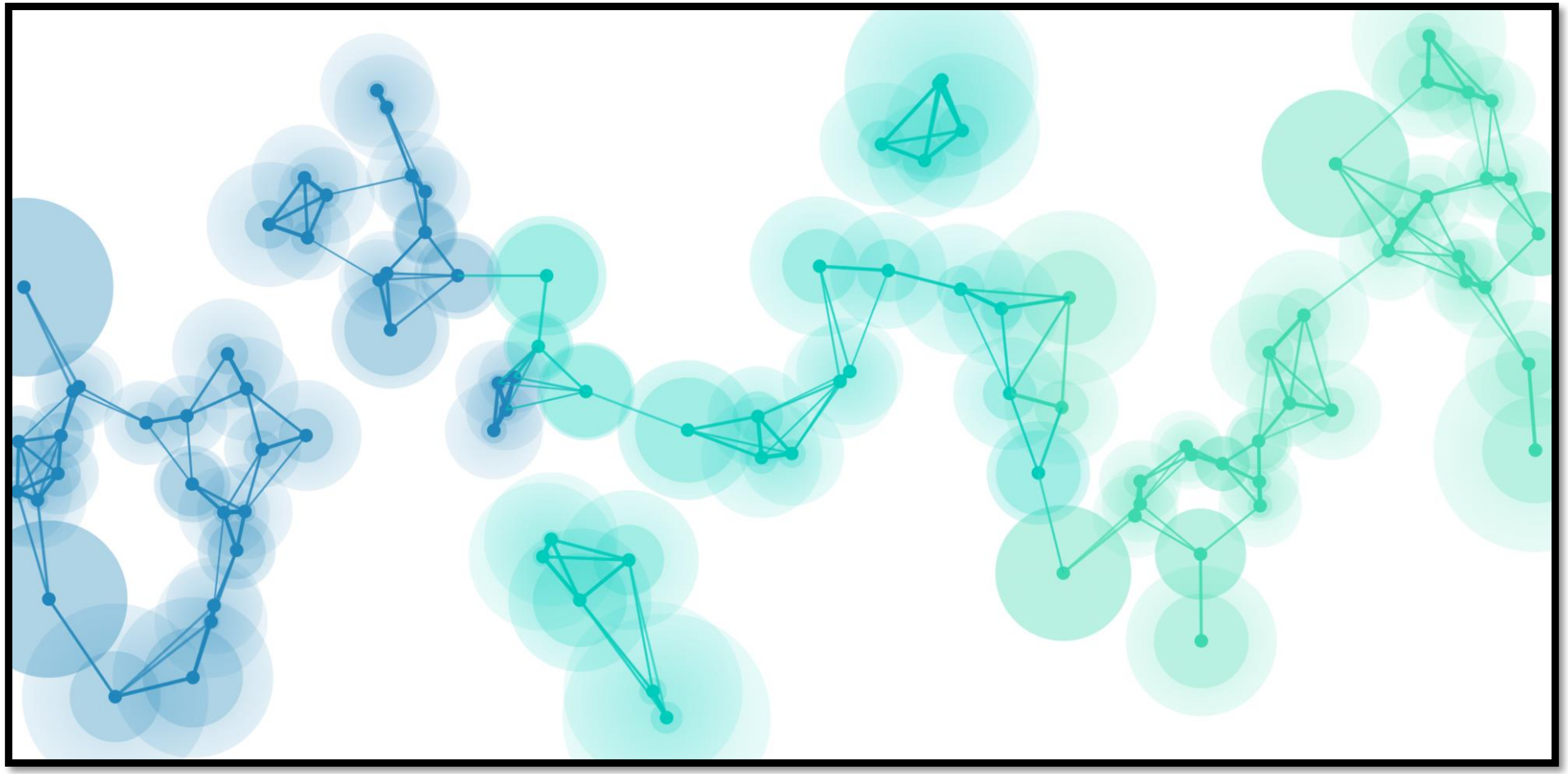
Required Reading





"How to Use t-SNE Effectively" (Wattenberg et al., 2016)
https://distill.pub/2016/misread-tsne/

Another Popular Choice: UMAP



Embeds a "fuzzy simplicial complex"

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

6.838: Shape Analysis

Spring 2021