Metric Learning of Manifolds

Dominique Perrault-Joncas and Marina Meilă

Department of Statistics
University of Washington
{mmp,dcpj}@stat.washington.edu

Outline

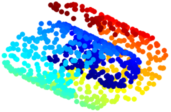
Outline

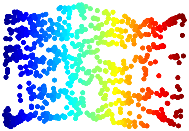
Manifold learning (ML): Results depend on data

Success

Original

(Swiss Roll)



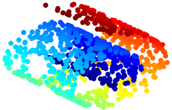


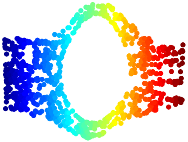
Isomap

Failure

Original

(Swiss Roll with hole)

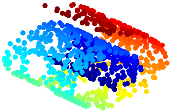




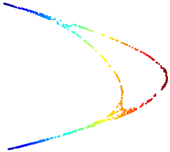
Isomap

Results depend on algorithm

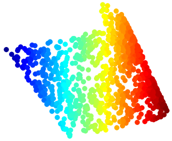
Original data
(Swiss Roll with hole)



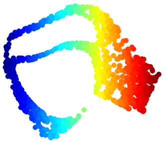
Laplacian Eigenmaps
(LE)



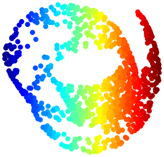
Hessian Eigenmaps
(HE)



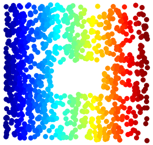
Isomap



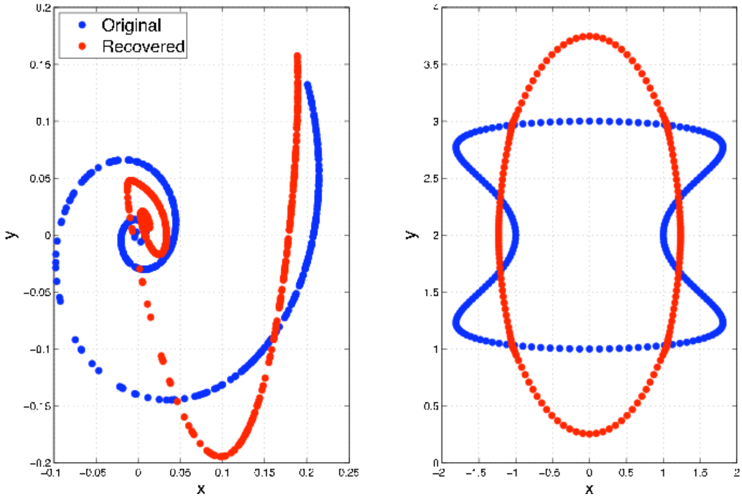
Local Linear
Embedding (LLE)



Local Tangent Space
Alignment (LTSA)



Distortion occurs even for the simplest examples



"Which ML method better?" vs "Can we make them all better?"

- A great number of ML algorithms exist
  - Isomap, Laplacian Eigenmaps (LE), Diffusion Maps (DM), Hessian Eigenmaps (HE), Local Linear Embedding (LLE), Latent Tangent Space Alignment (LTSA)
- Each of them "work well" in special cases, "fail" in other cases
- Current paradigm: Design a ML method that "works better" (i.e "succeeds" when current ones "fail")

- Our goal/New paradigm: make existing ML methods (and future ones) "successful"

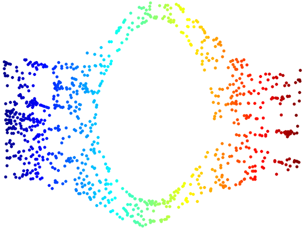
  i.e., given a ML method that "fails" on a data set from a manifold, we will augment it in a way that can make it "succeed"
- for rigurous, general definition of "success"/"failure"

Outline

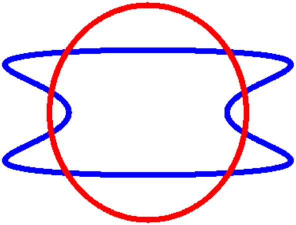
Basic notation

- D = original dimension of the data (high in real examples)
- d = intrinsic dimension of the manifold $d << D$
- m = embedding dimension $m \geq d$ (chosen by user)

$m = d = 2$



$m = 2 > d = 1$



Preserving topology vs. preserving (intrinsic) geometry

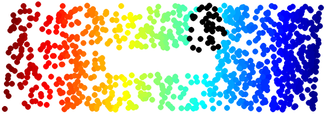
- ML Algorithm maps data $x \in \mathbb{R}^D \longrightarrow \phi(x) \in \mathbb{R}^m$
- Mapping $\mathcal{M} \longrightarrow \phi(\mathcal{M})$ is diffeomorphism
  preserves topology
  often satisfied by embedding algorithms
- Mapping ϕ preserves
  - distances along curves in $\mathcal{M}$
  - angles between curves in $\mathcal{M}$
  - areas, volumes

  ...i.e. ϕ is **isometry**

  For most algorithms, in most cases, ϕ is not isometry

Preserves topology

Preserves topology + intrinsic geometry



Previous known results in geometric recovery

Positive results

- Consistency results for Laplacian and eigenvectors
  - [Hein & al 07,Coifman & Lafon 06, Ting & al 10, Gine & Koltchinskii 06]
  - implies isometric recovery for LE, DM in special situations
- Isomap recovers (only) **flat** manifolds isometrically

Negative results

- obvious negative examples
- No affine recovery for normalized Laplacian algorithms [Goldberg&al 08]
- Sampling density distorts the geometry for LE [Coifman& Lafon 06]

Consistency is not sufficient

Necessary conditions for consistent geometric recovery

$\phi(\mathcal{D})$ isometric with $\mathcal{M}$ in the limit

- $n \to \infty$ sufficient data
- $\epsilon \to 0$ with suitable rate
  - consistent tangent plane estimation
- cancel effects of (non-uniform) sampling density [Coifman & Lafon 06]

- These conditions are not sufficient
- In particular, consistency of ϕ is not sufficient

Our approach, restated

Given

- mapping ϕ that preserves topology

  true in many cases

Objective

- augment ϕ with geometric information g so that (ϕ, g) preserves the geometry

Our approach, restated

Given

- mapping ϕ that preserves topology

  true in many cases

Objective

- augment ϕ with geometric information g so that (ϕ, g) preserves the geometry

  g is the **Riemannian metric**.

The Riemannian metric g

- $\mathcal{M}$ = manifold
- p point on $\mathcal{M}$
- $T_p\mathcal{M}$ = **tangent plane** at p
- g = **Riemannian metric** on $\mathcal{M}$

  g defines inner product on $T_p\mathcal{M}$

$$< v, w > = v^T g(p) w \quad \text{for } v, w \in T_p\mathcal{M} \text{ and for } p \in \mathcal{M}$$

  - g is symmetric and positive definite tensor field
  - g also called first differential form
  - $(\mathcal{M}, g)$ is a **Riemannian manifold**

All geometric quantities on $\mathcal{M}$ involve g

- Length of curve c

$$l(c) = \int_a^b \sqrt{\sum_{ij} g_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt}} dt,$$

- Volume of $W \subset \mathcal{M}$

$$Vol(W) = \int_W \sqrt{\det(g)} dx^1 \dots dx^d \, .$$

- Angle $\cos(v, w) = \frac{<v,w>}{\sqrt{<v,v><w,w>}}$

- Under a change of parametrization, g changes in a way that leaves geometric quantities invariant

- Current algorithms: estimate $\mathcal{M}$
- This talk: estimate g along with $\mathcal{M}$
  (and in the same coordinates)

Outline

Problem formulation

- Given:
  - data set $\mathcal{D} = \{p_1, \ldots p_n\}$
    sampled from manifold $\mathcal{M} \subset \mathbb{R}^D$
  - embedding $\{\, \phi(p),\ p \in \mathcal{D}\,\}$
    by e.g LLE, Isomap, LE, ...
- Estimate $g_p \in \mathbb{R}^{m \times m}$ the Riemannian metric for $p \in \mathcal{D}$ in the embedding coordinates ϕ

- The embedding (ϕ, g) will preserve the geometry of the original data manifold

Relation between g and Δ

- Δ = Laplace-Beltrami operator on $\mathcal{M}$
  - $\Delta = \mathrm{div} \cdot \mathrm{grad}$
  - on C^2, $\Delta f = \sum_j \frac{\partial^2 f}{\partial x_j^2}$
  - on weighted graph with similarity matrix S, and $t_p = \sum_{pp'} S_{pp'}$, $\Delta = \mathrm{diag}\{t_p\} - S$

Proposition 1 (Differential geometric fact)

$$\Delta f = \frac{1}{\sqrt{\det(g)}} \sum_l \frac{\partial}{\partial x^l} \left(\sqrt{\det(g)} \sum_k g^{lk} \frac{\partial}{\partial x^k} f \right),$$

where $[g^{lk}] = g^{-1}$

Estimation of g

Proposition 2 (Main Result 1)

$$g^{ij} = \frac{1}{2}\Delta(\phi_i - \phi_i(p))\,(\phi_j - \phi_j(p))|_{\phi_i(p),\phi_j(p)}$$

where $[g^{lk}] = g^{-1}$ (matrix inverse)

Intuition:

- at each point $p \in \mathcal{M}$, $g(p)$ is a $d \times d$ matrix
- apply Δ to embedding coordinate functions $\phi_1, \ldots \phi_m$
- this produces $g^{-1}(p)$ in the given coordinates
- our algorithm implements matrix version of this operator result
- consistent estimation of Δ is solved [Coifman&Lafon 06,Hein&al 07]

The case $m > d$

- Technical point: if $m > d$ then "g^{-1}" not full rank
- Denote
  - $\phi : \mathcal{M} \longrightarrow \phi(\mathcal{M})$ embedding
  - $d\phi$ Jacobian of ϕ

$$< v, w >_{g_p} \text{ in } T_p\mathcal{M} \quad \longrightarrow \quad < d\phi_p v, d\phi_p w >_{h_p} \text{ in } T_{\phi(p)}\phi(\mathcal{M})$$

Proposition 3

$h_p = \tilde{h}_p^{\dagger}$, where

$$\tilde{h}_p \;=\; \frac{1}{2}\Delta(\phi_i - \phi_i(p))\,(\phi_j - \phi_j(p))|_{\phi_i(p)\phi_j(p)}$$

- h is the **push-forward** of g on $\phi(\mathcal{M})$
- $h_p = d\phi_p g_p d\phi_p^T$ or in matrix notation $H = JGJ^T$
- rank of h_p is $d = \dim \mathcal{M} \; < \; m$

Algorithm to Estimate Riemann metric g (Main Result 2)

1. Preprocessing (construct neighborhood graph, ...)
2. Estimate discretized Laplace-Beltrami operator Δ
3. Find an embedding ϕ of $\mathcal{D}$ into $\mathbb{R}^m$
4. Estimate $h_p^\dagger$ and h_p for all p

Output (ϕ_p, h_p) for all p

Algorithm to Estimate Riemann metric g (Main Result 2)

1. Preprocessing (construct neighborhood graph, ...)
2. Estimate discretized Laplace-Beltrami operator Δ
3. Find an embedding ϕ of $\mathcal{D}$ into $\mathbb{R}^m$
4. Estimate $h_p^\dagger$ and h_p for all p

Output (ϕ_p, h_p) for all p

Algorithm RIEMANNIANEMBEDDING

Input data $\mathcal{D}$, m embedding dimension, ϵ resolution

1. Construct neighborhood graph p, p' neighbors iff $||p - p'||^2 \leq \epsilon$
2. Construct similary matrix
   $S_{pp'} = e^{-\frac{1}{\epsilon}||p-p'||^2}$ iff p, p' neighbors, $S = [S_{pp'}]_{p,p' \in \mathcal{D}}$
3. Construct (renormalized) Laplacian matrix [Coifman & Lafon 06]
   - 3.1 $t_p = \sum_{p' \in \mathcal{D}} S_{pp'}$, $T = \mathrm{diag}\, t_p$, $p \in \mathcal{D}$
   - 3.2 $\tilde{S} = I - T^{-1} S T^{-1}$
   - 3.3 $\tilde{t}_p = \sum_{p' \in \mathcal{D}} \tilde{S}_{pp'}$, $\tilde{T} = \mathrm{diag}\, \tilde{t}_p, p \in \mathcal{D}$
   - 3.4 $P = \tilde{T}^{-1} \tilde{S}$.
4. Embedding $[\phi_p]_{p \in \mathcal{D}} = $ GENERICEMBEDDING($\mathcal{D}$, m)
5. Estimate embedding metric H_p at each point
   denote $Z = X * Y$, $X, Y \in \mathbb{R}^N$ iff $Z_i = X_i Y_i$ for all i
   - 5.1 For $i, j = 1 : m$,
     $H^{ij} = \frac{1}{2}[P(\phi_i * \phi_j) - \phi_i * (P\phi_j) - \phi_j * (P\phi_i)]$ (column vector)
   - 5.2 For $p \in \mathcal{D}$, $\tilde{H}_p = [H_p^{ij}]_{ij}$ and $H_p = \tilde{H}_p^{\dagger}$

Ouput $(\phi_p, H_p)_{p \in \mathcal{D}}$

Computational cost

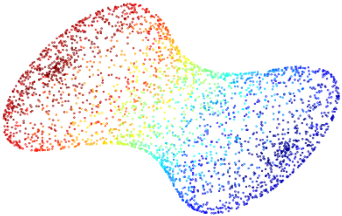
$n = |\mathcal{D}|$, $D =$ data dimension, $m=$ embedding dimension

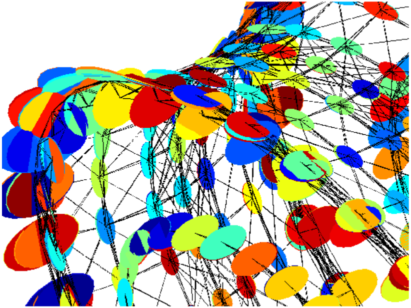
1. Neighborhood graph $+$
2. Similarity matrix $\mathcal{O}(n^2 D)$ (or less)
3. Laplacian $\mathcal{O}(n^2)$
4. GenericEmbedding e.g. $\mathcal{O}(mN)$ (eigenvector calculations)
5. Embedding metric
   - $\mathcal{O}(nm^2)$ obtain g^{-1} or $h^{\dagger}$
   - $\mathcal{O}(nm^3)$ obtain g or h

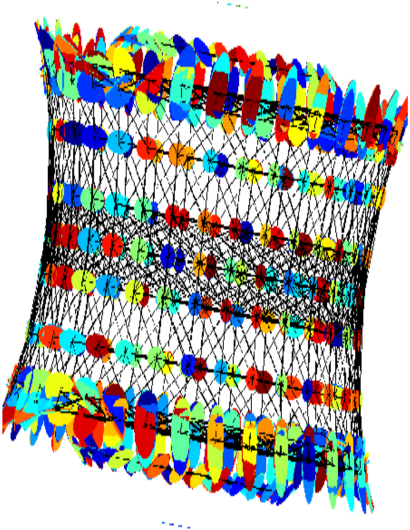
- Steps 1–3 are part of many embedding algorithms
- Steps 2–3 independent of ambient dimension D
- Matrix inversion/pseudoinverse can be performed only when needed

Outline

g shows embedding distortion



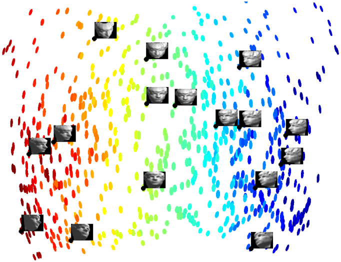




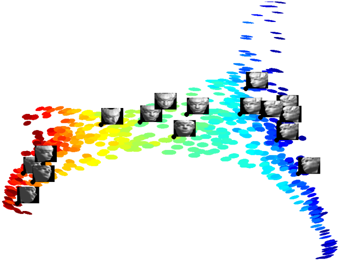
g for Sculpture Faces

- $n = 698$ with 64×64 gray images of faces
  - head moves up/down and right/left

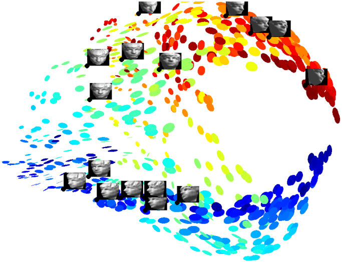
LTSA



Isomap



LTSA



Laplacian Eigenmaps

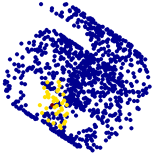
Visualization

- Visualization = isometric embedding in 2D or 3D
- Not possible globally for all manifolds

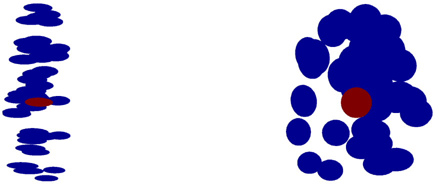
  Example: the sphere cannot be mapped onto a plane
- But possible *locally*

Locally Normalized Visualization

Given: (ϕ, g) Riemannian Embedding of $\mathcal{D}$



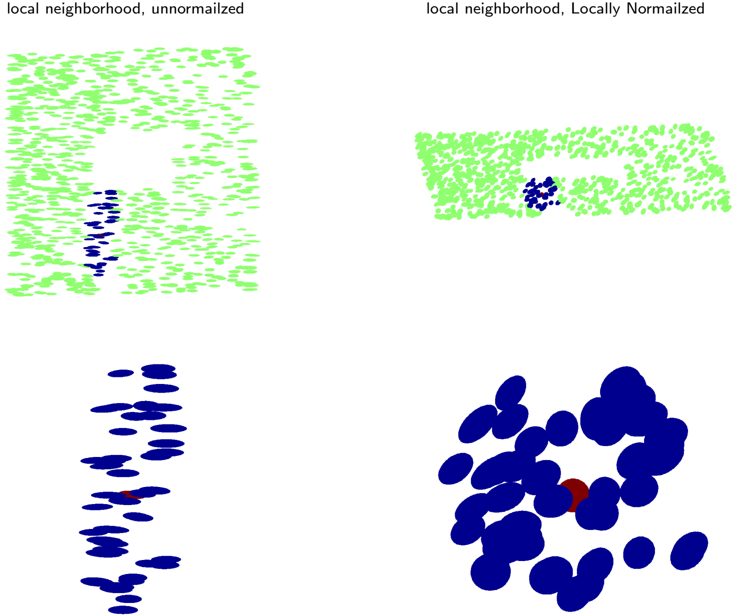
1. Select a point p on the manifold
2. Transform coordinates $\tilde{\phi}_{p'} \leftarrow g_p{}^{-1/2}\phi_{p'}$ for $p' \in \mathcal{D}$
   This assures that $\tilde{g}_p = I_m$ the unit matrix
   $\Rightarrow \tilde{\phi}$ are **normal coordinates** around p



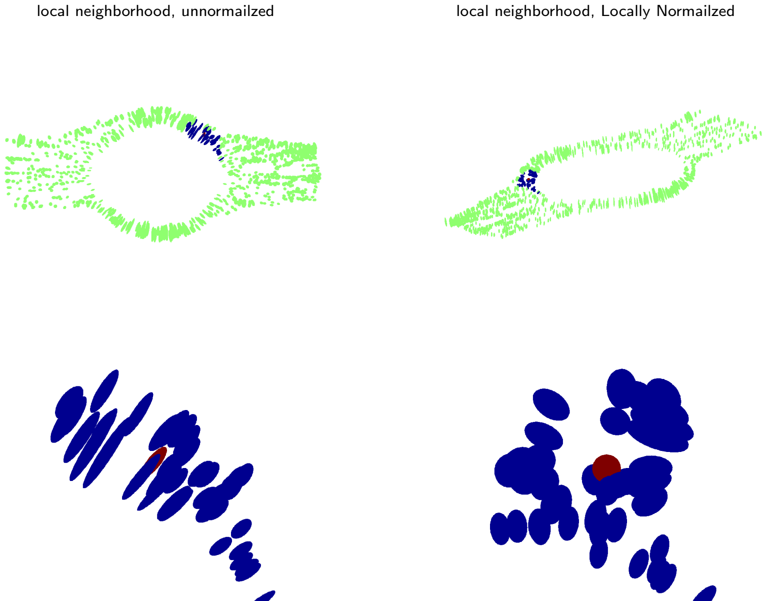
g (before) $\tilde{g}$ (after)

- Now we have a Locally Normalized view of $\mathcal{M}$ around p

Swiss Roll with hole (LTSA)

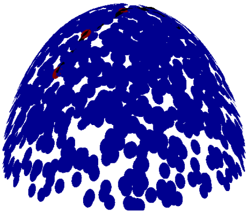


Swiss Roll with hole (Isomap)

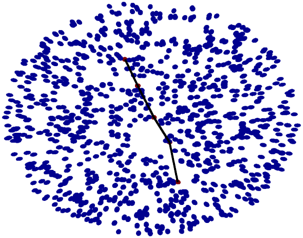


Calculating distances in the manifold $\mathcal{M}$

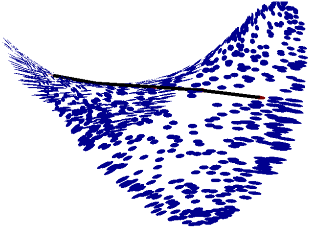
- Geodesic distance = shortest path on $\mathcal{M}$
- should be invariant to coordinate changes



Original



Isomap



Laplacian Eigenmaps

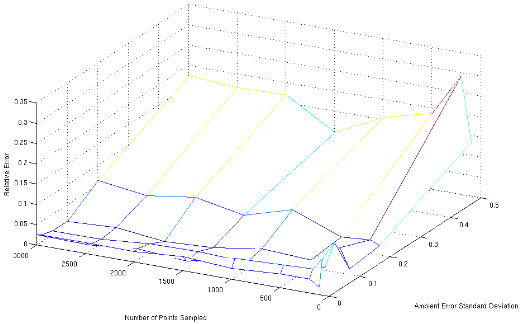
Calculating distances in the manifold $\mathcal{M}$

Embedding	$\|\|f(p) - f(p')\|\|$	Shortest Path $d_{\mathcal{G}}$	Metric $\hat{d}$	$\hat{d}$ R. Err.
Original data	1.41	1.57	1.62	3.0%
Isomap $s = 2$	1.66	1.75	1.63	3.7%
LTSA $s = 2$	0.07	0.08	1.65	4.8%
LE $s = 3$	0.08	0.08	1.62	3.1%

Table: The errors in the last column are with respect to the true distance $d = \pi/2 \simeq 1.5708$.

Convergence of the distance estimates

% error in geodesic distance vs sample size *n*, noise level



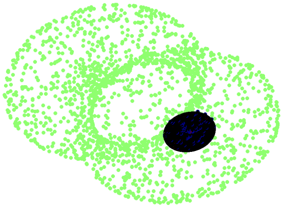
- insensitive to small noise levels, then degrades gradually
- slow convergence with *n*

Computing Area/Volume

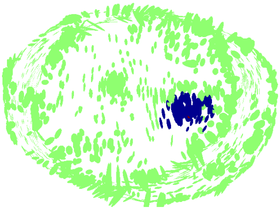
By performing a Voronoi tessellation of a coordinate chart (U, x), we can obtain the estimator $\triangle x^1 \ldots \triangle x^d$ around p and multiply it by $\sqrt{\det(h)}$ to obtain $\triangle \mathrm{Vol} \simeq d\mathrm{Vol}$. Summing over all points in a set $W \subset \mathcal{M}$ gives the estimator:

$$\hat{\mathrm{Vol}}(W) = \sum_{p \in W} \sqrt{\det(h_p)} \triangle x^1(p) \ldots \triangle x^d(p) .$$

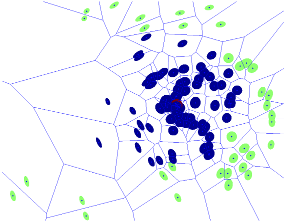
Hourglass Area



Original



Laplacian Eigenmaps



Voronoi Tessellation

Hourglass Area Results

Embedding	Naive Area of W	$\hat{\text{Vol}}(W)$	$\hat{\text{Vol}}(W)$ R. Err.
Original data	0.85 (0.03)†	0.93 (0.03)	11 %
Isomap	2.7†	0.93 (0.03)	11%
LTSA	1e-03 (5e-5)	0.93 (0.03)	11%
LE	1e-05 (4e-4)†	0.82 (0.03)	2.6%

Table: † The naive area estimator is obtained by projecting the manifold or embedding on $T_p\mathcal{M}$ and $T_{f(p)}f(\mathcal{M})$, respectively. This requires manually specifying the correct tangent planes, except for LTSA, which already estimates $T_{f(p)}f(\mathcal{M})$. The true area is $\simeq$ 0.8409.

An Application: Gaussian Processes on Manifolds

- Gaussian Processes (GP) can be extended to manifolds via SPDE's (Lindberg, Rue, and Lindstrom, 2011)
- Let $\left(\kappa^2 - \Delta_{\mathbb{R}^d}\right)^{\alpha/2} u(x) = \xi$ with ξ Gaussian with noise, then $u(x)$ is a Matérn GP
- Subsituting $\Delta_{\mathcal{M}}$ allows us to define a Matérn GP on $\mathcal{M}$
- Semi-supervised learning: unlabelled points can be learned by Kriging using the Covariance matrix Σ of $u(x)$

Solution by Finite Element Method

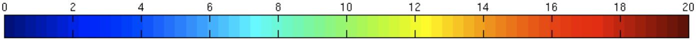
- Σ^{-1}, the precision matrix, can be derived by finite element method for α integer
- E.g. given finite element basis (ψ_i, $i = 1, \ldots, m$), Σ^{-1} for $\alpha = 1$ is given by

$$\begin{aligned}
\Sigma^{-1}_{i,j}(\kappa^2) &= \kappa^2 C_{i,j} + W_{i,j} \\
C_{i,j} &= < \psi_i, \psi_j > \\
W_{i,j} &= < \nabla\psi_i, \nabla\psi_j >
\end{aligned}$$

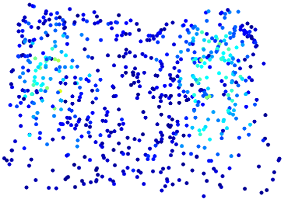
  where $< \psi_i, \psi_j >= \int_{\mathcal{M}} \psi_i, \psi_j \sqrt{\det(g)} d\mu$
- This means we can compute Σ using an embedding of the data provided we know the pushforward metric of g

Semisupervised learning

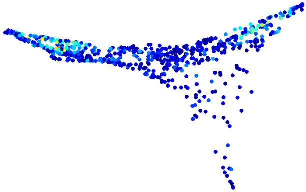
Sculpture Faces: Predicting Head Rotation



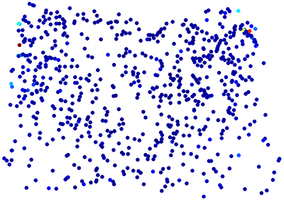
Absolute Error (AE) as percentage of range



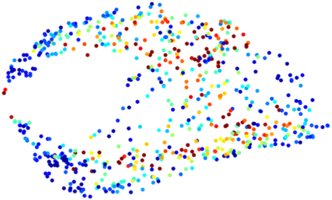
Isomap (Total AE = 3078)



LTSA (Total AE = 3020)



Metric (Total AE = 660)



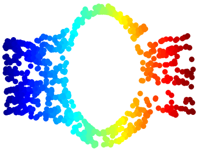
Laplacian Eigenmaps (Total AE = 3078)

Outline

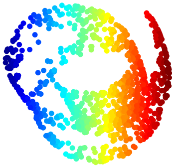
Consistency. Necessary condition

- The embedding ϕ must be diffeomorphic, consistent, Laplacian-consistent

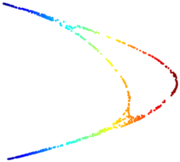
Yes



No



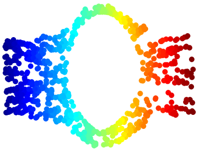
No



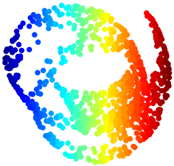
Consistency. Necessary condition

- The embedding ϕ must be diffeomorphic, consistent, Laplacian-consistent

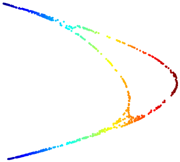
Yes



No



No



- Algorithms based on Laplacian eigenvectors (e.g LE, DM): **YES**, with modification regarding choice of eigenvectors
- LTSA **NO** unless $m = d$
- Isomap **?**

Consistency theorems

Proposition 4 (Main Result 3)

A If $\phi : \mathcal{M} \rightarrow \phi(\mathcal{M})$ diffeomorphic and consistent (i.e. $\phi(\mathcal{D}_n) \stackrel{n\rightarrow\infty}{\longrightarrow} \phi(\mathcal{M})$)
then $(\phi(\mathcal{D}_n), h_n) \stackrel{n\rightarrow\infty}{\longrightarrow} (\phi(\mathcal{M}), h)$

B Laplacian Eigenmaps and Diffusion Map satisfy conditions of **A** if $\mathcal{M}$ compact

Technical contributions

- We offer a natural solution to the geometry preserving embedding problem
- We introduce a method for estimating the Riemannian metric
  - theoretical solution (Propositions 2, 3)
  - practical solution (Algorithm RiemannEmbed)
  - Statistical analysis: consistency result (Proposition 4)

Significance

- Augmentation of manifold learning algorithms
  - For a given algorithm, all geometrical quantities are preserved simultaneously, by recovering g = geometry preserving embedding
  - We can obtain geometry preserving embeddings with any reasonable algorithm
- Unification of algorithms
  - Now, all "reasonable" algorithms/embeddings are asymptotically equivalent from the geometry point of view
  - We can focus on comparing algorithms based on other criteria speed, rate of convergence, numerical stability
  - g offers a way to compare the algorithms' outputs
    - Each algorithm has own ϕ
    - Hence outputs of different algorithms are incomparable
    - But (ϕ^A, g^A), (ϕ^B, g^B) should be comparable because they aim to represent intrinsic/geometric quantities