Manifold Learning for the Sciences

Marina Meilă

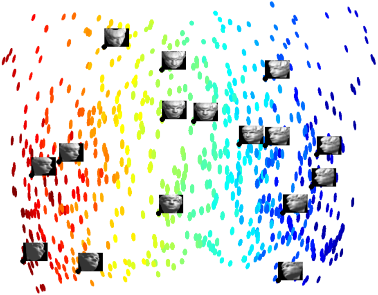
University of Washington
mmp@stat.washington.edu

ipam

Geometry of Big Data2019 Workshop

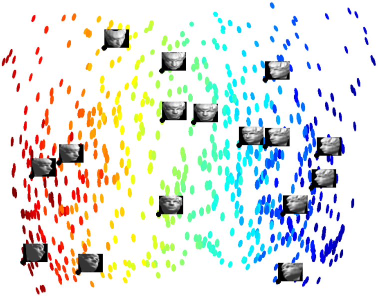
Outline

When to do (non-linear) dimension reduction



- high-dimensional data $p \in \mathbb{R}^D$, $D = 64 \times 64$
- can be described by a small number d of continuous parameters
- Usually, large sample size n

When to do (non-linear) dimension reduction

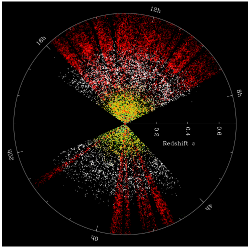


Why?

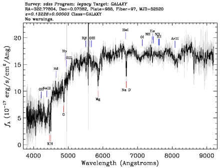
- To save space and computation
  - $n \times D$ data matrix $\to$ $n \times s$, $s \ll D$
- To use it afterwards in (prediction) tasks
- To understand the data better
  - preserve large scale features, suppress fine scale features

Spectra of galaxies measured by the Sloan Digital Sky Survey (SDSS)

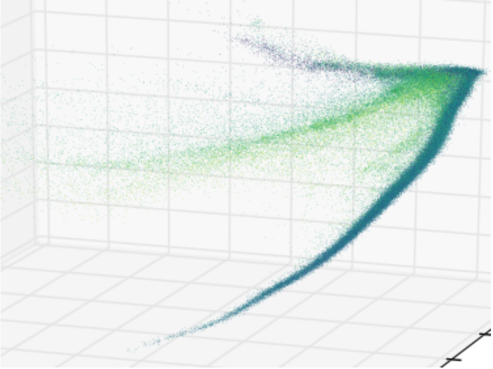
www.sdss.org



www.sdss.org

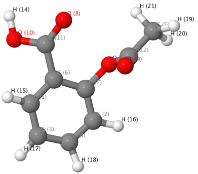


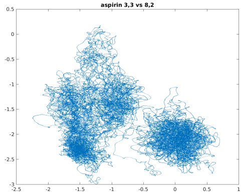
- Preprocessed by Jacob VanderPlas and Grace Telford
- $n = 675,000$ spectra $\times$ $D = 3750$ dimensions



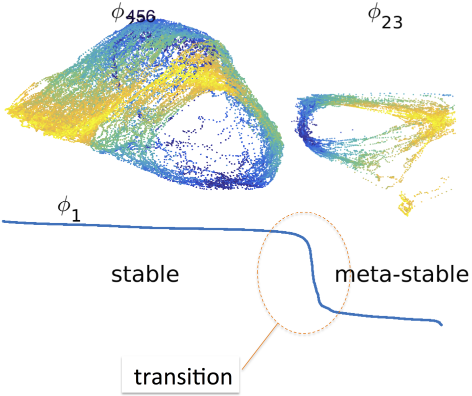
Molecular configurations

aspirin molecule





- Data from Molecular Dynamics (MD) simulations of small molecules by [Chmiela et al. 2016]
- $n \approx 200,000$ configurations $\times D \sim 20 - 60$ dimensions

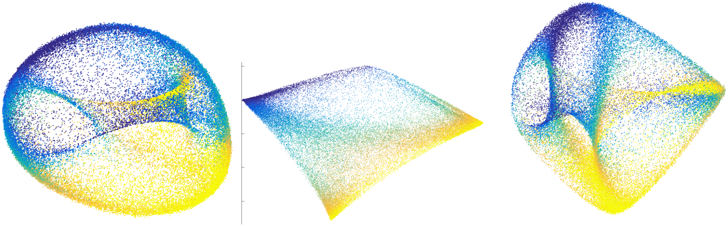


Geometric Learning for the sciences

- Big data
  - Necessary in non-parametric estimation
  - Big data contains more complex patterns
- Beyond "validation by visualization"
  - results/correctness should be quantified

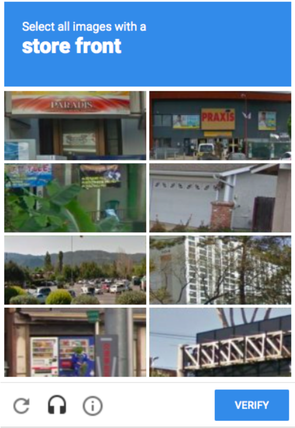
Geometric Learning for the sciences

- Big data
  - Necessary in non-parametric estimation
  - Big data contains more complex patterns
- Beyond "validation by visualization"
  - results/correctness should be quantified
  - hard/impossible if $d > 3$



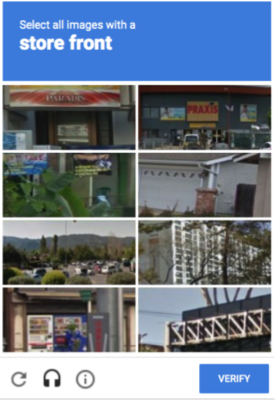
Geometric Learning for the sciences

- Big data
  - Necessary in non-parametric estimation
  - Big data contains more complex patterns
- Beyond "validation by visualization"
  - results/correctness should be quantified
  - hard/impossible if $d > 3$
  - demanding on expert time



Geometric Learning for the sciences

- Big data
  - Necessary in non-parametric estimation
  - Big data contains more complex patterns
- Beyond "validation by visualization"
  - results/correctness should be quantified
  - hard/impossible if $d > 3$
  - demanding on expert time



Select all
peptides that bind to this substrate

Select all images with
AGN (Active Galactic Nuclei)

Geometric Learning for the sciences

- Big data
  - Necessary in non-parametric estimation
  - Big data contains more complex patterns
- Beyond "validation by visualization"
  - results/correctness should be quantified
  - hard/impossible if $d > 3$
  - demanding on expert time
  - discovering what known?

Geometric Learning for the sciences

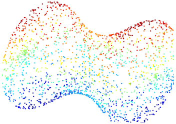
- Big data
  - Necessary in non-parametric estimation
  - Big data contains more complex patterns
- Beyond "validation by visualization"
  - results/correctness should be quantified
  - hard/impossible if $d > 3$
  - demanding on expert time
  - discovering what known?

This talk

- Metric Manifold Learning arxiv:1305.7255
  - estimate/correct the geometric distortion
  - "effectively" isometric embedding
- physical meaning of manifold coordinates arxiv 1811.11891

Brief intro to manifold learning algorithms

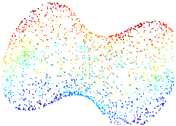
- **Input** Data $p_1, \dots p_n$, embedding dimension m, neighborhood scale parameter ϵ



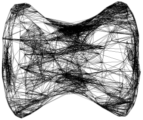
$p_1, \dots p_n \subset \mathbb{R}^D$

Brief intro to manifold learning algorithms

- **Input** Data $p_1, \ldots p_n$, embedding dimension m, neighborhood scale parameter ϵ
- Construct neighborhood graph p, p' neighbors iff $||p - p'||^2 \leq \epsilon$

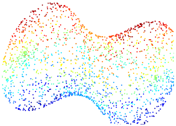


$p_1, \ldots p_n \subset \mathbb{R}^D$

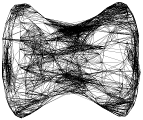


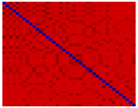
Brief intro to manifold learning algorithms

- **Input** Data $p_1, \ldots p_n$, embedding dimension m, neighborhood scale parameter ϵ
- Construct neighborhood graph p, p' neighbors iff $||p - p'||^2 \leq \epsilon$
- Construct a $n \times n$ matrix: its leading eigenvectors are the coordinates $\phi(p_{1:n})$



$p_1, \ldots p_n \subset \mathbb{R}^D$





Brief intro to manifold learning algorithms

- **Input** Data $p_1, \ldots p_n$, embedding dimension m, neighborhood scale parameter ϵ
- Construct neighborhood graph p, p' neighbors iff $||p - p'||^2 \leq \epsilon$
- Construct a $n \times n$ matrix: its leading eigenvectors are the coordinates $\phi(p_{1:n})$

Laplacian Eigenmaps/Diffusion Maps [Belkin,Niyogi 02,Nadler et al 05]

- Construct similarity matrix

$$S = [S_{pp'}]_{p,p' \in \mathcal{D}} \quad \text{with} \quad S_{pp'} = e^{-\frac{1}{\epsilon}||p-p'||^2} \quad \text{iff } p, p' \text{ neighbors}$$

- Construct Laplacian matrix $L = I - T^{-1}S$ with $T = \mathrm{diag}(S\mathbf{1})$
- Calculate $\phi^{1\ldots m}$ = eigenvectors of L (smallest eigenvalues)
- coordinates of $p \in \mathcal{D}$ are $(\phi^1(p), \ldots \phi^m(p))$

Brief intro to manifold learning algorithms

- **Input** Data $p_1, \ldots p_n$, embedding dimension m, neighborhood scale parameter ϵ
- Construct neighborhood graph p, p' neighbors iff $||p - p'||^2 \leq \epsilon$
- Construct a $n \times n$ matrix: its leading eigenvectors are the coordinates $\phi(p_{1:n})$

ISOMAP [Tennenbaum, deSilva & Langford 00]

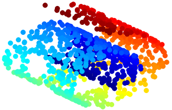
- Find all shortest paths in neighborhood graph, construct matrix of distances

$$M = [\text{distance}^2_{pp'}]$$

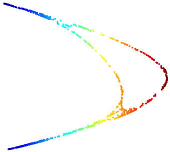
- use M and Multi-Dimensional Scaling (MDS) to obtain m dimensional coordinates for $p \in \mathcal{D}$

Embedding in 2 dimensions by different manifold learning algorithms

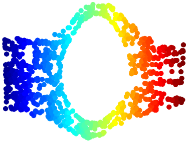
Original data
(Swiss Roll with hole)



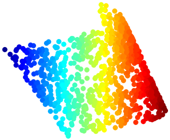
Laplacian Eigenmaps
(LE)



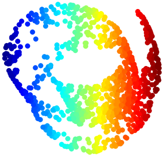
Isomap



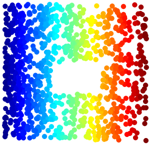
Hessian Eigenmaps (HE)



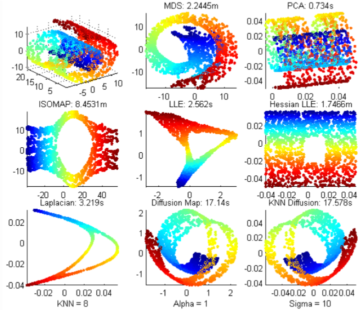
Local Linear Embedding
(LLE)



Local Tangent Space
Alignment (LTSA)



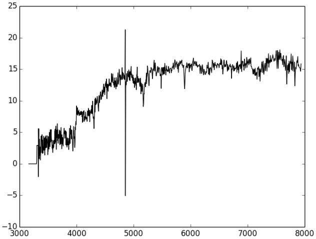
How to evaluate the results objectively?



- which of these embedding are "correct"?
- if several "correct", how do we reconcile them?
- if not "correct", what failed?

Algorithms Multidimensional Scaling (MDS), Principal Components (PCA), Isomap, Locally Linear Embedding (LLE), Hessian Eigenmaps (HE), Laplacian Eigenmaps (LE), Diffusion Maps (DM)

How to evaluate the results objectively?



Spectrum of a galaxy. Source SDSS, Jake VanderPlas

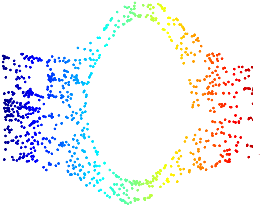
- which of these embedding are "correct"?
- if several "correct", how do we reconcile them?
- if not "correct", what failed?
- what if I have real data?

Preserving topology vs. preserving (intrinsic) geometry

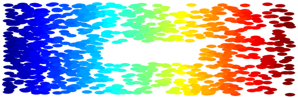
- Algorithm maps data $p \in \mathbb{R}^D \longrightarrow \phi(p) = x \in \mathbb{R}^m$
- Mapping $\mathcal{M} \longrightarrow \phi(\mathcal{M})$ is diffeomorphism
  preserves topology
  often satisfied by embedding algorithms
- Mapping ϕ preserves
  - distances along curves in $\mathcal{M}$
  - angles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

- Nash's Theorem: Isometric embedding is possible.
- Diffusion Maps embedding is isometric in the limit [Berard,Besson,Gallot 94]
- algorithm based on Nash's theorem (isometric embedding for very low d) [Verma 11]
- Isomap [Tennenbaum,]recovers flat manifolds isometrically
- Consistency results for Laplacian and eigenvectors
  - [Hein & al 07,Coifman & Lafon 06, Singer 06, Ting & al 10, Gine & Koltchinskii 06]
  - imply isometric recovery for LE, DM in special situations

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]

Our approach: Metric Manifold Learning

[Perrault-Joncas,M 10]

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.

Dominique Perrault-Joncas

The Riemannian metric g

Mathematically

- $\mathcal{M}$ = (smooth) manifold
- p point on $\mathcal{M}$
- $T_p\mathcal{M}$ = **tangent subspace** 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 fundamental form
  - $(\mathcal{M}, g)$ is a Riemannian manifold

In coordinates at each point $p \in \mathcal{M}$, G_p is a positive definite matrix of rank d

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

- Volume element on manifold

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

- Length of curve c

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

- 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)

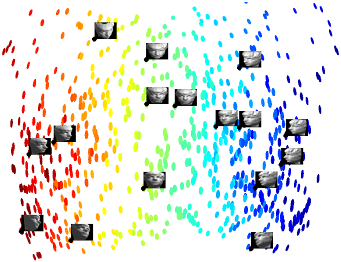
Problem formulation

- Given:
  - data set $\mathcal{D} = \{p_1, \dots p_n\}$ sampled from manifold $\mathcal{M} \subset \mathbb{R}^D$
  - embedding $\{ x_i = \phi(p_i),\ p_i \in \mathcal{D} \}$
    by e.g LLE, Isomap, LE, ...
- Estimate $G_i \in \mathbb{R}^{m \times m}$ the (pushforward) Riemannian metric for $p_i \in \mathcal{D}$ in the embedding coordinates ϕ
- The embedding $\{x_{1:n}, G_{1:n}\}$ will preserve the geometry of the original data

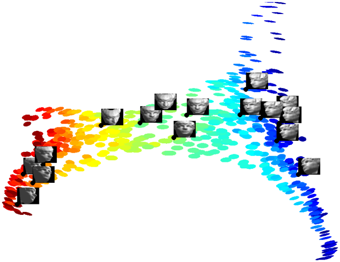
g for Sculpture Faces

- $n = 698$ gray images of faces in $D = 64 \times 64$ dimensions
  - head moves up/down and right/left

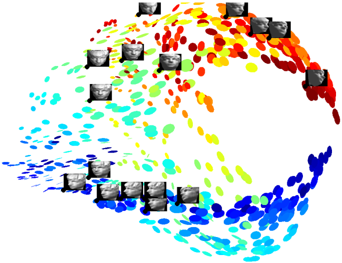
LTSA Algoritm



Isomap



LTSA



Laplacian Eigenmaps

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)

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

Estimation of g

Proposition

Let Δ be the Laplace-Beltrami operator on $\mathcal{M}$. Then

$$h_{kl}(p) = \frac{1}{2}\Delta(\phi_k - \phi_k(p))(\phi_l - \phi_l(p))|_{\phi_k(p),\phi_l(p)}$$

where $h = g^{-1}$ (matrix inverse) and $k, l = 1, 2, \ldots m$ are embedding dimensions

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 well studied [Coifman&Lafon 06,Hein&al 07]

Algorithm to Estimate Riemann metric g

Given dataset $\mathcal{D}$

1. Preprocessing (construct neighborhood graph, ...)
2. Find an embedding ϕ of $\mathcal{D}$ into $\mathbb{R}^m$
3. Estimate discretized Laplace-Beltrami operator L
4. Estimate H_p and $G_p = H_p^\dagger$ for all p
   4.1 For $i, j = 1 : m$,
   $H^{ij} = \frac{1}{2}\left[L(\phi_i * \phi_j) - \phi_i * (L\phi_j) - \phi_j * (L\phi_i)\right]$
   where $X * Y$ denotes elementwise product of two vectors $X, Y \in \mathbb{R}^N$
   4.2 For $p \in \mathcal{D}$, $H_p = [H_p^{ij}]_{ij}$ and $G_p = H_p^\dagger$

Output (ϕ_p, G_p) for all p

Algorithm MetricEmbedding

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 $t_p = \sum_{p' \in \mathcal{D}} S_{pp'}$, $T = \operatorname{diag} t_p$, $p \in \mathcal{D}$
   3.2 $\tilde{S} = I - T^{-1} S T^{-1}$
   3.3 $\tilde{t}_p = \sum_{p' \in \mathcal{D}} \tilde{S}_{pp'}$, $\tilde{T} = \operatorname{diag} \tilde{t}_p, p \in \mathcal{D}$
   3.4 $P = \tilde{T}^{-1} \tilde{S}$.
4. Embedding $[\phi_p]_{p \in \mathcal{D}} = \text{GenericEmbedding}(\mathcal{D}, m)$
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 $i, j = 1 : m$, $H^{ij} = \frac{1}{2}\left[P(\phi_i * \phi_j) - \phi_i * (P\phi_j) - \phi_j * (P\phi_i)\right]$ (column vector)
   5.2 For $p \in \mathcal{D}$, $\tilde{H}_p = [H_p^{ij}]_{ij}$ and $H_p = \tilde{H}_p^{\dagger}$

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

Metric Manifold Learning summary

Metric Manifold Learning = estimating (pushforward) Riemannian metric G_i along with embedding coordinates

Why useful

- Measures local distortion induced by any embedding algorithm $G_i = I_d$ when no distortion at p_i
- Algorithm independent geometry preserving method
- Outputs of different algorithms on the same data are comparable
- Models built from compressed data are more interpretable

Applications

- Estimating distortion
- Correcting distortion
  - Integrating with the local volume/length units based on G_i
  - Riemannian Relaxation [McQueen, M, Perrault-Joncas NIPS16]
- Estimation of neighborhood radius [Perrault-Joncas,M,McQueen NIPS17] and of intrinsic dimension d (variant of [Chen,Little,Maggioni,Rosasco])
- Accelerating Topological Data Analysis, selecting eigencoordinates,. . . (in progress)

Consistency of the Riemannian metric estimator

Proposition

- If the embedding $\phi : \mathcal{M} \rightarrow \phi(\mathcal{M})$ is
  - **A** diffeomorphic
  - **B** consistent $\phi(\mathcal{D}_n) \stackrel{n\rightarrow\infty}{\longrightarrow} \phi(\mathcal{M})$
  - **C** Laplacian consistent $L_n\phi(\mathcal{D}_n) \stackrel{n\rightarrow\infty}{\longrightarrow} \Delta\phi(\mathcal{M})$

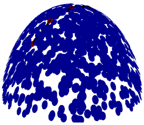
  then the dual Riemannian metric estimator h is consistent

$$(\phi(\mathcal{D}_n), h_n) \stackrel{n\rightarrow\infty}{\longrightarrow} (\phi(\mathcal{M}), h)$$

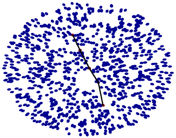
- Laplacian Eigenmaps and Diffusion Map satisfy **A, B** if $\mathcal{M}$ compact

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

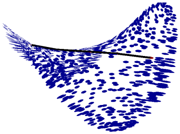
Original



Isomap



Laplacian Eigenmaps



true distance $d = 1.57$

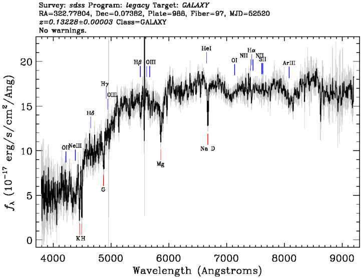
Embedding	$\|\|f(p) - f(p')\|\|$	Shortest Path $d_{\mathcal{G}}$	Metric $\hat{d}$	Rel. error
Original data	1.41	1.57	1.62	3.0%
Isomap $s = 2$	1.66	1.75	1.63	3.7%
LTSA $s = 2$	0.07	0.08	1.65	4.8%
LE $s = 2$	0.08	0.08	1.62	3.1%

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

Manifold learning for SDSS Spectra of Galaxies

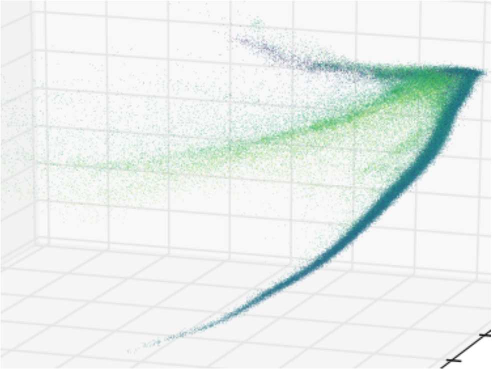
Main sample of galaxy spectra from the Sloan Digital Sky Survey (675,000 spectra originally in 3750 dimensions).

- n =675,000 spectra in $D = 3750$ dimensions



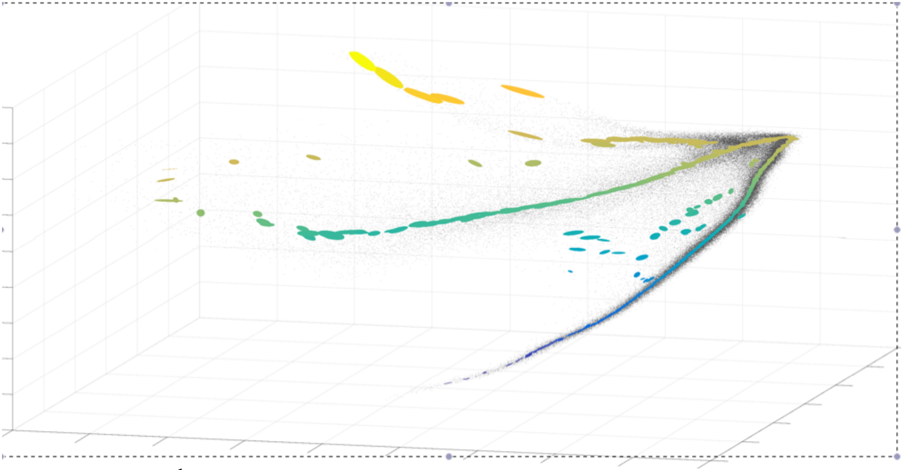
- data curated by Grace Telford,
- "noise removal" by Jake VanderPlas

Embedding into 3 dimensions



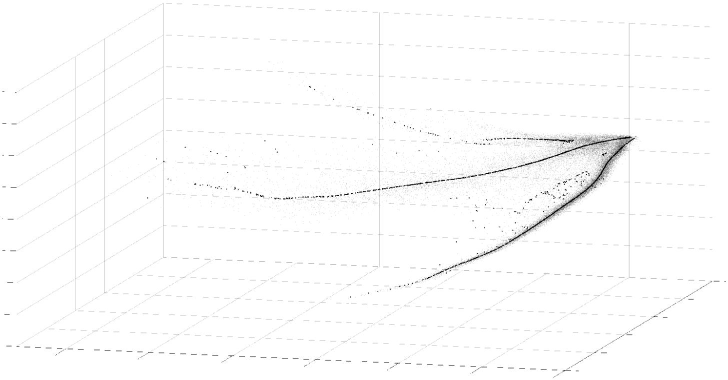
embedding by

How distorted is this embedding?



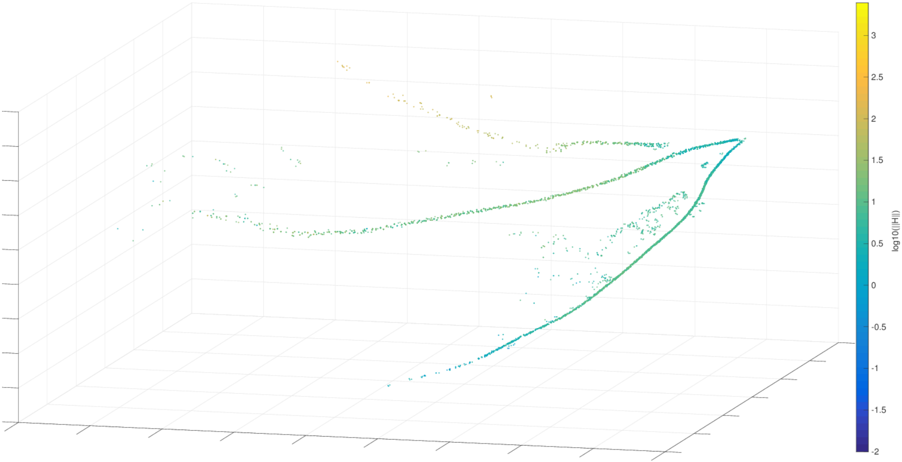
(ellipses represent G_i^{-1})

Riemannian Relaxation along principal curves



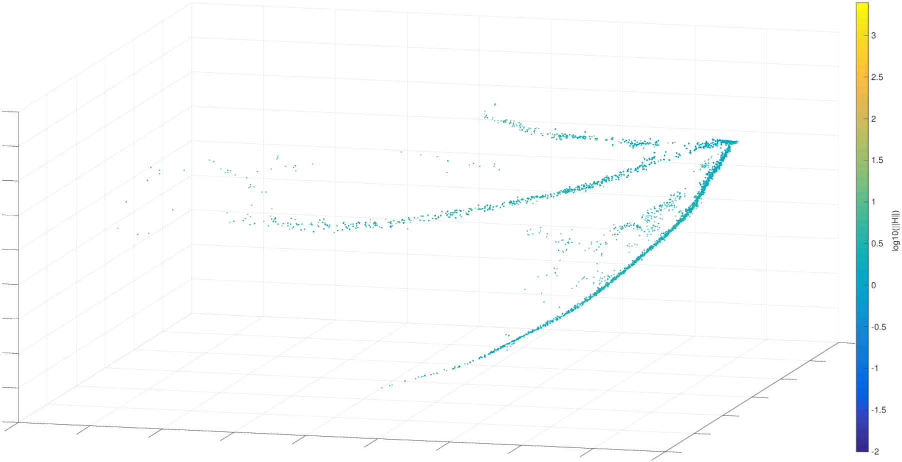
Find principal curves

Riemannian Relaxation along principal curves



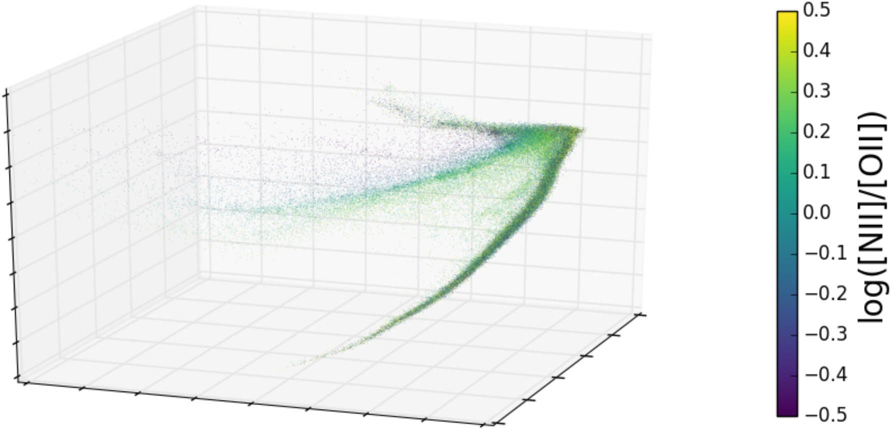
Points near principal curves, colored by $\log_{10}(G_i)$ (0 means no distortion)

Riemannian Relaxation along principal curves



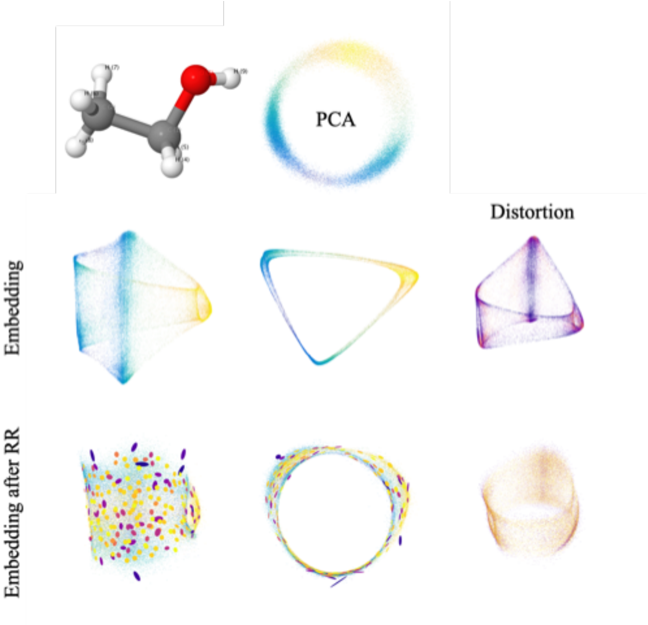
Points near principal curves, colored by $\log_{10}(G_i)$, after Riemannian Relaxation
(0 means no distortion)

Riemannian Relaxation along principal curves



All data after Riemannian Relaxation

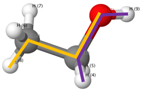
Embedding and Riemannian Relaxation for Ethanol molecular configurations



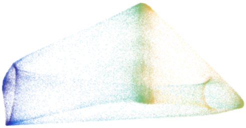
Outline

Motivation

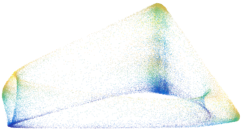
ethanol



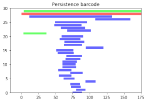
torsion 1



torsion 2



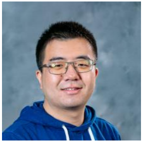
persistence



- 2 rotation angles parametrize this manifold
- Can we discover these features automatically? Can we select these angles from a larger set of features with physical meaning?

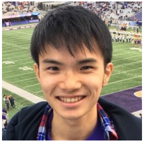
Problem formulation

Hanyu Zhang



Sam Koelle

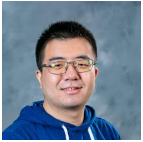
Yu-chia Chen



- Given
  - data $\xi_i \in \mathbb{R}^D$, $i \in 1 \dots n$
  - embedding of data $\phi(\xi_{1:n})$ in $\mathbb{R}^m$
- Assume
  - data sampled from smooth manifold $\mathcal{M}$
  - $\mathcal{M}$ Riemannian with metric inherited from $\mathbb{R}^D$
  - embedding algorithm $\phi : \mathcal{M} \to \phi(\mathcal{M})$ is smooth embedding

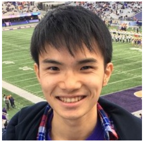
Problem formulation

Hanyu Zhang



Sam Koelle

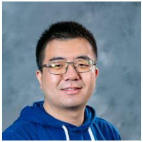
Yu-chia Chen



- Given
  - data $\xi_i \in \mathbb{R}^D$, $i \in 1 \ldots n$
  - embedding of data $\phi(\xi_{1:n})$ in $\mathbb{R}^m$
- dictionary of domain-related smooth functions $\mathcal{G} = \{g_1, \ldots g_p$, with $g_j : \mathbb{R}^D \rightarrow \mathbb{R}\}$.
  - e.g. all torsions in ethanol
- Assume
  - data sampled from smooth manifold $\mathcal{M}$
  - $\mathcal{M}$ Riemannian with metric inherited from $\mathbb{R}^D$
  - embedding algorithm $\phi : \mathcal{M} \rightarrow \phi(\mathcal{M})$ is smooth embedding

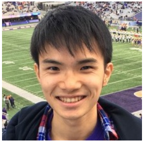
Problem formulation

Hanyu Zhang



Sam Koelle

Yu-chia Chen



- Given
  - data $\xi_i \in \mathbb{R}^D$, $i \in 1 \ldots n$
  - embedding of data $\phi(\xi_{1:n})$ in $\mathbb{R}^m$
- dictionary of domain-related smooth functions $\mathcal{G} = \{g_1, \ldots g_p$, with $g_j : \mathbb{R}^D \to \mathbb{R}\}$.
  - e.g. all torsions in ethanol
- Assume
  - data sampled from smooth manifold $\mathcal{M}$
  - $\mathcal{M}$ Riemannian with metric inherited from $\mathbb{R}^D$
  - embedding algorithm $\phi : \mathcal{M} \to \phi(\mathcal{M})$ is smooth embedding
- Goal to express the embedding coordinate functions $\phi_1 \ldots \phi_m$ in terms of functions in $\mathcal{G}$.
  More precisely, we assume that

$$\phi(x) = h(g_{j_1}(x), \ldots g_{j_s}(x)) \quad \text{with } g_{j_1,\ldots j_s} \subset \mathcal{G}.$$

Problem: find $S = \{j_1, \ldots j_s\}$

Challenges

$$\phi(x) = h(g_{j_1}(x), \ldots g_{j_s}(x)) \quad \text{with } g_{j_1,\ldots j_s} \subset \mathcal{G}.$$

- Framework: sparse recovery

- Challenges
- h non-linear (but smooth)
- ϕ defined up to diffeomorphism
  - hence, h cannot assume a parametric form
  - will not assume one-to-one correspondence between ϕ_k coordinates and g_j in dictionary

$$\text{e.g.} \quad \begin{array}{l} \phi_1 = g_1 g_2, \\ \phi_2 = g_1 \sin(g_3^2) \end{array} \quad \text{or} \quad \begin{array}{l} \phi_1 = \sin(\tau_1) \\ \phi_2 = \cos(\tau_1)\text{(ethanol)} \\ \phi_3 = \sin(\tau_2) \end{array}$$

Challenges

$$\phi(x) = h(g_{j_1}(x), \ldots g_{j_s}(x)) \quad \text{with } g_{j_1,\ldots j_s} \subset \mathcal{G}.$$

- Framework: sparse recovery

- Challenges
- h non-linear (but smooth)
- ϕ defined up to diffeomorphism
  - hence, h cannot assume a parametric form
  - will not assume one-to-one correspondence between ϕ_k coordinates and g_j in dictionary

$$\text{e.g.} \quad \begin{array}{l} \phi_1 = g_1 g_2, \\ \phi_2 = g_1 \sin(g_3^2) \end{array} \quad \text{or} \quad \begin{array}{l} \phi_1 = \sin(\tau_1) \\ \phi_2 = \cos(\tau_1)\text{(ethanol)} \\ \phi_3 = \sin(\tau_2) \end{array}$$

- we do not assume ϕ isometric (but smooth)
- what requirements on dictionary functions $g_{1:p}$ for unique recovery?

First Idea: from non-linear to linear

- If $\phi = h \circ g$, then

$$\boxed{\mathrm{D}\phi = DhDg}$$

- Sparse non-linear, non-parametric recovery $\rightarrow$ Sparse linear recovery

- A sparse linear system for every data point i
- Require subset S is same for all i
  - group Lasso problem

- Functional Lasso
  - optimize

$$(\text{FLasso}) \quad \min_{\beta} J_\lambda(\beta) = \frac{1}{2}\sum_{i=1}^{n} ||y_i - \mathbf{X}_i \boldsymbol{\beta}_i||_2^2 + \lambda/\sqrt{n}\sum_j ||\beta_j||,$$

  - with $y_i = \nabla\phi(\xi_i)$, $\mathbf{X}_i = \nabla g_{1:p}(\xi)$, $\beta_{ij} = \frac{\partial h}{\partial g_j}(\xi_i)$
  - support S of β selects $g_{j_1,\ldots j_s}$ from $\mathcal{G}$

Multidimensional FLASSO

- Assume

$$y_{ik} = \nabla f_k(\xi_i) \quad \mathbf{X}_i = \nabla g_{1:p}(\xi) \quad \beta_{ijk} = \frac{\partial h_k}{\partial g_j}(\xi_i) \tag{1}$$

and

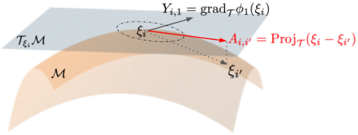
$$\beta_j = \text{vec}(\beta_{ijk},\ i = 1:n, k = 1:m) \in \mathbb{R}^{mn}, \quad \beta_{ik} = \text{vec}(\beta_{ijk},\ j = 1:p) \in \mathbb{R}^{p}. \tag{2}$$

$$J_\lambda(\beta) = \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{m}||y_{ik} - \mathbf{X}_i\beta_{ik}||^2 + \frac{\lambda}{\sqrt{mn}}\sum_{j=1}^{p}||\beta_j||. \tag{3}$$

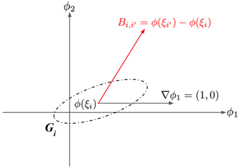
FLasso in manifold setting

- gradients ∇ $\rightarrow$ manifold gradients grad
- grad g_j is in $\mathcal{T}_{\xi_i}\mathcal{M}$
  - ∇g_j known analytically
- grad ϕ_k is in $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$
  - must be estimated
- must pull-back grad $\phi_k(\phi(\xi_i))$ to $\mathcal{T}_{\xi_i}\mathcal{M}$

$\mathcal{M}$



$\phi(\mathcal{M})$



Second Idea: pulling back gradients

- Estimating grad g_j
  1. Estimate tangent subspace at ξ_i by (weighted) PCA
  2. Project ∇g_j on tangent subspace

- Pulling back gradients of $\phi_{1:k}$
- Will use (push-forward) Riemannian metric G_i
- $\nabla \phi_k$ = unit vector in $\mathbb{R}^m$
- y_k = grad ϕ_k is projection of $\nabla \phi_k$ on $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$

$$Y_i = \text{grad}_{\mathcal{T}}\, \phi(\xi_i) \in \mathbb{R}^{m\times d}$$

- Idea Use G_i
  - Create neighbor matrices for ξ_i and $\phi(\xi_i)$.

$$A_i = \left[\text{Proj}_{\mathcal{T}_{\xi_i}\mathcal{M}}(\xi_{i'} - \xi_i)\right]_{i'\in\mathcal{N}_i} \qquad B_i = [\phi(\xi_{i'}) - \phi(\xi_i)]_{i'\in\mathcal{N}_i},$$

  - Remember $(\phi(\mathcal{M}), g)$ isometric to $(\mathcal{M}, id)$.
  - Solve linear system

$$\langle A_i, Y_i\rangle \approx \langle B_i, I\rangle_{G_i} \qquad A_i^T Y_i \approx B_i^T G_i I$$

- column span of G_i is $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$
- Proj on $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$ is implicit in G_i

Theory

- When is S unique? / When can $\mathcal{M}$ be uniquely parametrized by $\mathcal{G}$?
  Functional independence conditions on dictionary $\mathcal{G}$ and subset $g_{j_1,\ldots j_s}$
- Basic result

  $g_S = h \circ g_{S'}$ on U iff

$$\operatorname{rank}\begin{pmatrix} Dg_S \\ Dg_{S'} \end{pmatrix} = \operatorname{rank} Dg_{S'} \quad \text{on } U$$

Theory

- When is S unique? / When can $\mathcal{M}$ be uniquely parametrized by $\mathcal{G}$?
  Functional independence conditions on dictionary $\mathcal{G}$ and subset $g_{j_1,\ldots j_s}$
- Basic result

  $g_S = h \circ g_{S'}$ on U iff

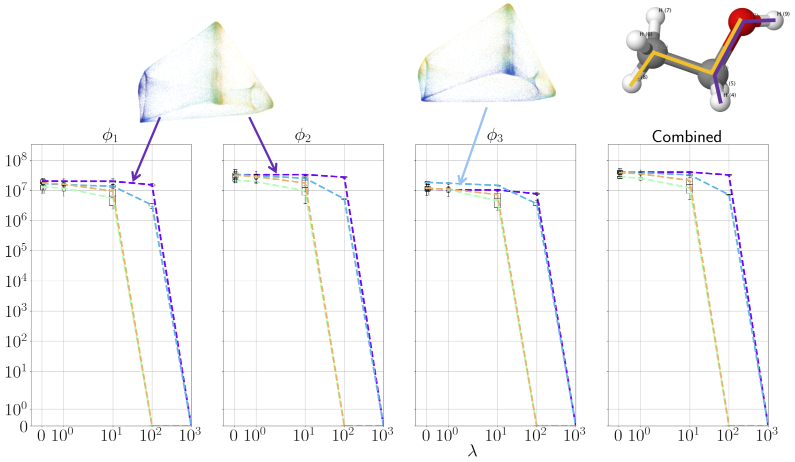
$$\text{rank}\begin{pmatrix} Dg_S \\ Dg_{S'} \end{pmatrix} = \text{rank}\, Dg_{S'} \quad \text{on } U$$

- When can FLasso recover S ?
  Incoherence conditions

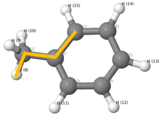
$$\mu = \max_{i=1:n, j\in S, j'\notin S} |\mathbf{X}_{ji}^T \mathbf{X}_{j'i}| \quad \nu = \frac{1}{\min_{i=1:n} ||\mathbf{X}_{iS}^T \mathbf{X}_{iS}||_2} \quad nd\sigma^2 = \sum_{i,k} \epsilon_{ik}^2$$

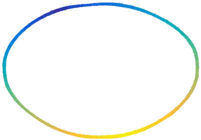
Theorem If $\mu\nu\sqrt{s} + \frac{\sigma\sqrt{nd}}{\lambda} < 1$ then $\beta_j = 0$ for $j \notin S$.

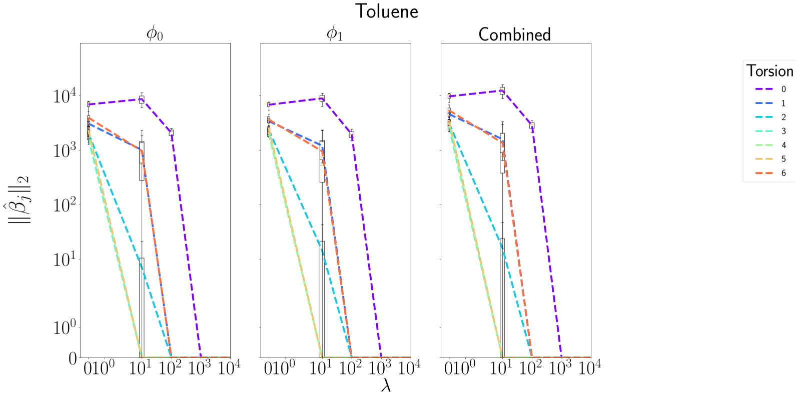
Ethanol MD simulation



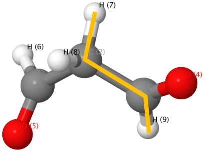
Toluene MD simulation

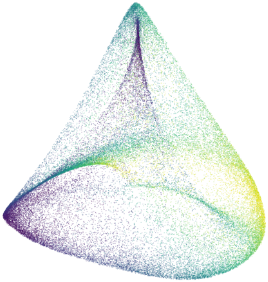


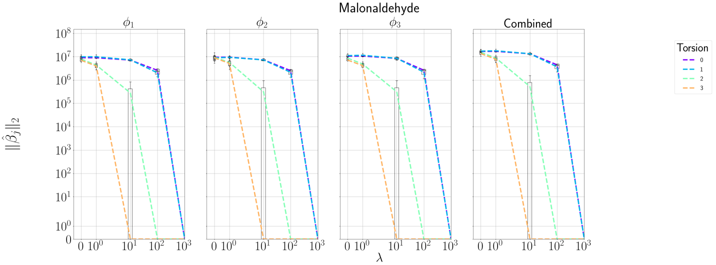




Malondialdehyde MD simulation







Manifold learning for sciences and engineering

Manifold learning should be like PCA

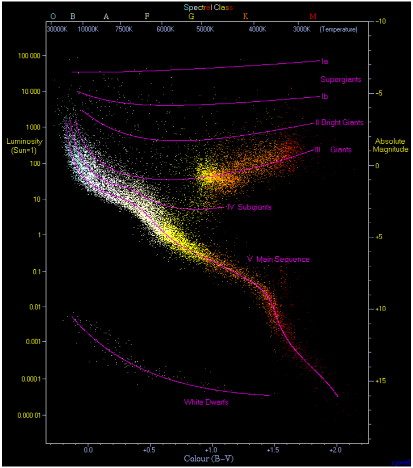
- tractable/scalable
- “automatic” – minimal burden on human
- first step in data processing pipe-line

  should not introduce artefacts

More than PCA

- estimate richer geometric/topological information
- dimension
- borders, stratification
- clusters
- Morse complex
- meaning of coordinates/continuous parametrization

Manifold Learning for engineering and the sciences



- “physical laws through machine learning”
- scientific discovery by quantitative/statistical data analysis
- manifold learning as preprocessing for other tasks

Samson Koelle, Yu-Chia Chen, Hanyu Zhang, Alon Milchgrub
Dominique-Perrault Joncas (Google), James McQueen (Amazon)

Jacob VanderPlas, Grace Telford (UW Astronomy)
Jim Pfaendtner (UW), Chris Fu (UW)
A. Tkatchenko (Luxembourg), S. Chmiela (TU Berlin), A. Vasquez-Mayagoitia (ALCF)

Thank you