Unsupervised Validation for Unsupervised Learning

Marina Meilă

University of Washington
mmp@stat.washington.edu
with Dominique Perrault-Joncas, James McQueen, Yu-chia Chen, Samson Koelle, Hanyu Zhang

09/16/2019

Scientific discovery by machine learning and the mythical human "expert"

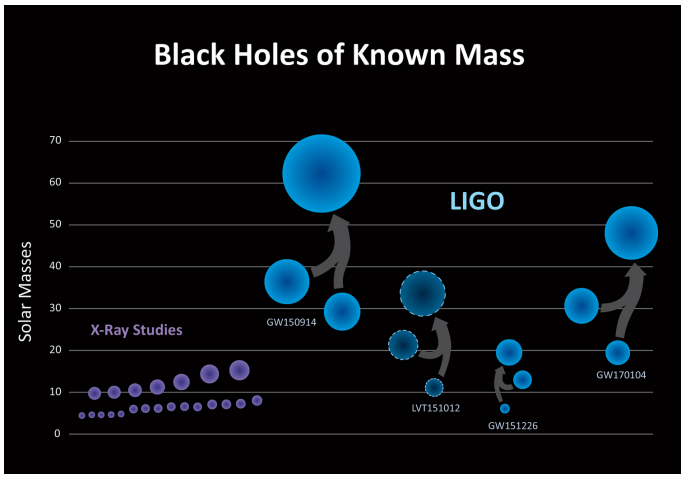
- Big data
  - Allows us to ask more detailed questions (e.g "personalized medicine")
  - Big data contains more complex patterns

Scientific discovery by machine learning and the mythical human "expert"

- Big data
  - Allows us to ask more detailed questions (e.g "personalized medicine")
  - Big data contains more complex patterns
  - Machine Learning discovers patterns fast
- Typically – validation by "domain experts"

Scientific discovery by machine learning and the mythical human "expert"

- Big data
  - Allows us to ask more detailed questions (e.g "personalized medicine")
  - Big data contains more complex patterns
  - Machine Learning discovers patterns fast
- Typically – validation by "domain experts"
- Often Hypotheses are cheap, experiments are expensive



Drowning in hypotheses...

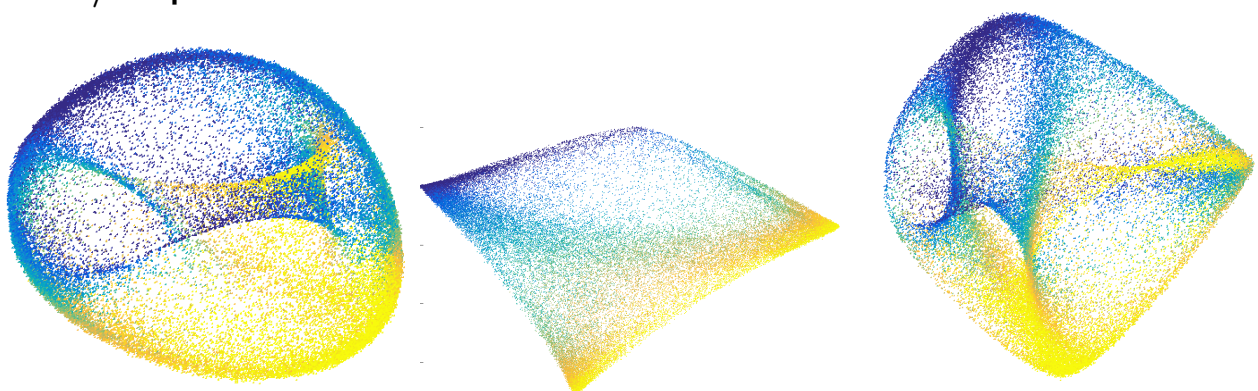
Validation is the bottleneck

- Validation by visualization
- is qualitative not quantitative

Drowning in hypotheses...

Validation is the bottleneck

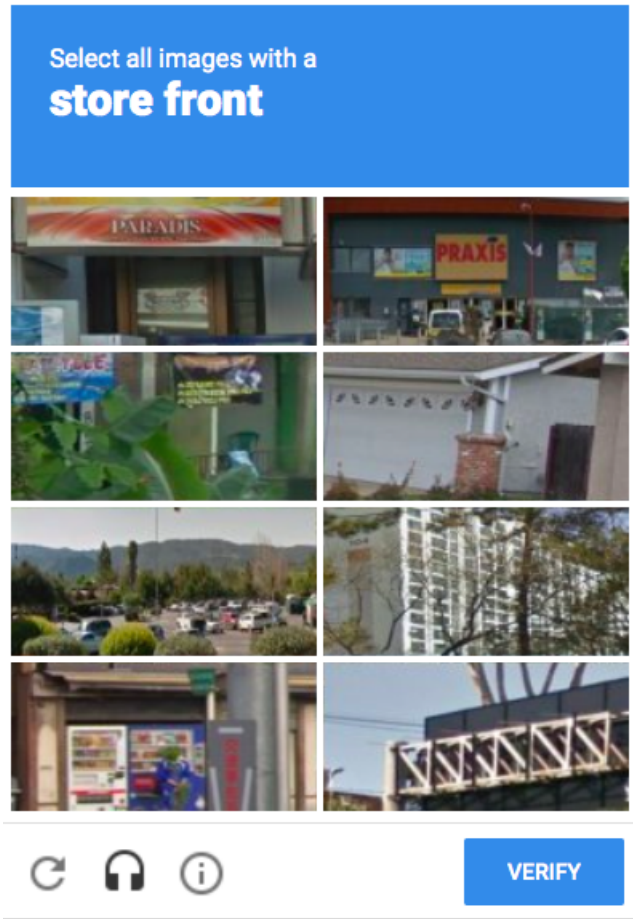
- Validation by visualization
- is qualitative not quantitative
- hard/impossible in dimension > 3



Drowning in hypotheses...

Validation is the bottleneck

- Validation by visualization
- is qualitative not quantitative
- hard/impossible in dimension > 3

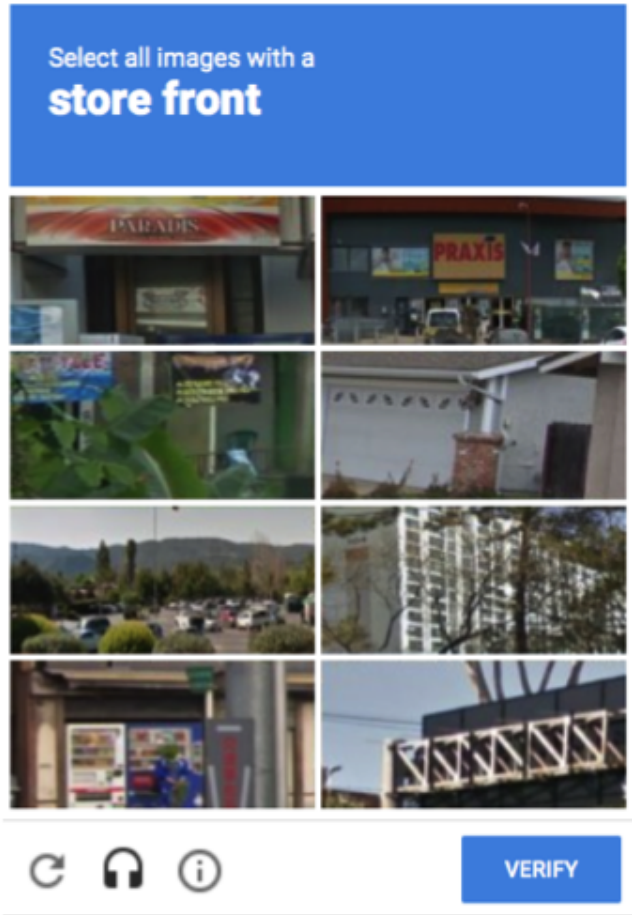


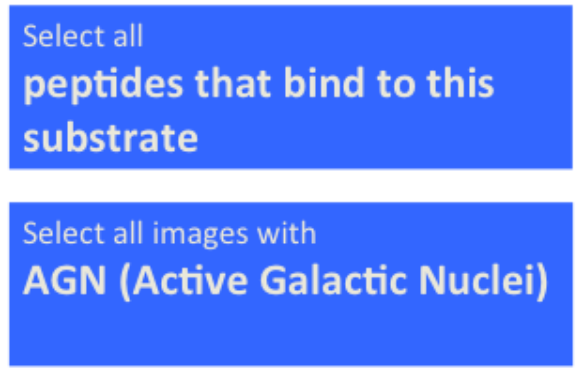
- can't be crowdsourced

Drowning in hypotheses. . .

Validation is the bottleneck

- Validation by visualization
- is qualitative not quantitative
- hard/impossible in dimension > 3
- can't be crowdsourced





Drowning in hypotheses...

Validation is the bottleneck

- Validation by visualization
- is qualitative not quantitative
- hard/impossible in dimension > 3
- can't be crowdsourced
- discovering what is known?

Validation should be part of learning

- Unsupervised validation = validation by machine not by human expert

- THIS TALK
- Data driven methods to make unsupervised learning more reproducible, trustworthy and free of artefacts
  - want stability
  - through geometry

Stability guarantees for clustering [M NeurIPS 2018]

provable "correctness" for the practitioner

Metric manifold learning [Perrault-Joncas,M arXiv:1305.7255]

"coordinate independent" geometric recovery

Manifold coordinates with physical meaning [M,Koelle,Zhang arXiv:1811.11891]

interpretability in the language of the problem

Outline

Stability guarantees for clustering [M NeurIPS 2018]
provable "correctness" for the practitioner

Metric manifold learning [Perrault-Joncas,M arXiv:1305.7255]
"coordinate independent" geometric recovery

Manifold coordinates with physical meaning [M,Koelle,Zhang arXiv:1811.11891]
interpretability in the language of the problem

For the practitioner of clustering

- Clustering algorithm e.g. K-means, Spectral clustering produces clustering $\mathcal{C}$ with K clusters

For the practitioner of clustering

- Clustering algorithm e.g. K-means, Spectral clustering produces clustering $\mathcal{C}$ with K clusters

- IDEALLY WANTED: guarantee that $\mathcal{C}$ is correct/optimal
- WHAT WE CAN DO: guarantee that $\mathcal{C}$ is approximately correct/optimal

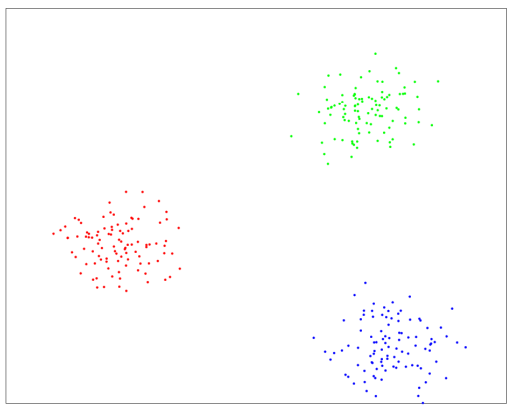
For the practitioner of clustering

- Clustering algorithm e.g. K-means, Spectral clustering produces clustering $\mathcal{C}$ with K clusters

- IDEALLY WANTED: guarantee that $\mathcal{C}$ is correct/optimal
- WHAT WE CAN DO: guarantee that $\mathcal{C}$ is approximately correct/optimal

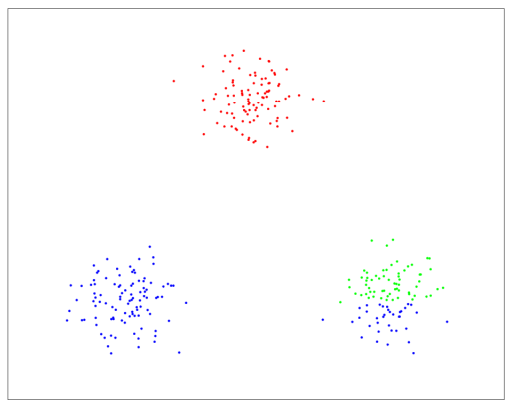
- WHEN $\mathcal{C}$ is good and stable

Good, stable



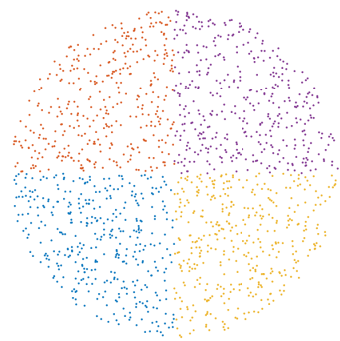
SS output: OI$=1e^{-4}$

Bad



no guarantee

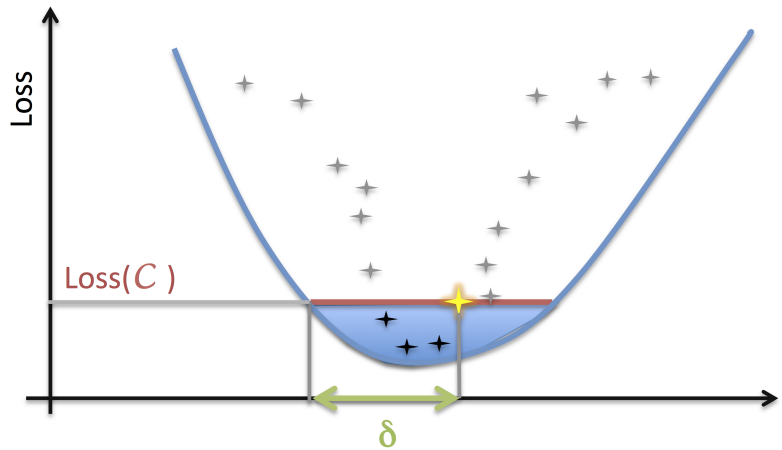
Unstable



no guarantee

What is an Optimality Interval (OI)?

- Let $B_\varepsilon = \{\mathcal{C}' \mid d^{EM}(\mathcal{C}', \mathcal{C}) \leq \varepsilon\}$.
- $\mathcal{C}'$ is good if $\text{Loss}(\mathcal{C}') \leq \text{Loss}(\mathcal{C}) + \alpha$.
- OI=ε iff any good $\mathcal{C}' \in B_\varepsilon$ in particular, $\mathcal{C}^{\text{opt}} \in B_\varepsilon$
- If OI exists, we say $\mathcal{C}$ is stable



Idea: Use convex relaxations

Clustering problem Given data, K, loss function $\text{Loss}(\mathcal{C})$

$$L^{\text{opt}} = \min_{\mathcal{C} \in \mathbf{C}_K} \text{Loss}(\mathcal{C}), \text{ with solution } \mathcal{C}^{\text{opt}} \text{Hard} \tag{1}$$

Convex relaxation of problem (1).

$$L^* = \min_{X \in \mathcal{X}} \text{Loss}(X), \quad \text{with solution } X^* \tag{2}$$

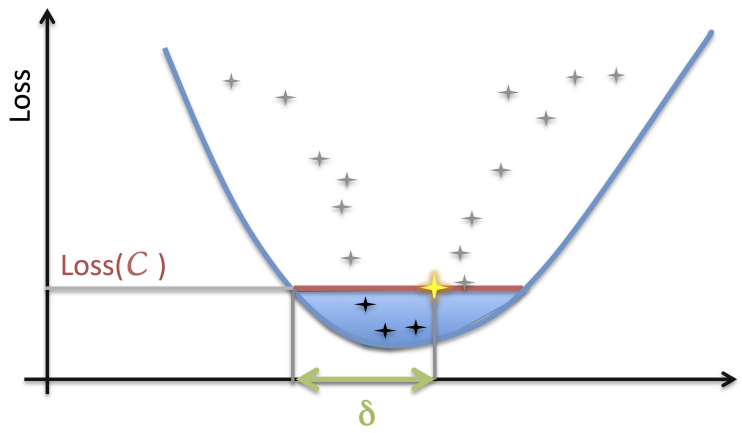
where $\mathcal{X}$ is convex set of matrices, $\mathcal{X} \supset \{X(\mathcal{C}), \mathcal{C} \in \mathbf{C}_K\}$
$\text{Loss}(X)$ convex in X and $\text{Loss}(\mathcal{C}) \equiv \text{Loss}(X(\mathcal{C}))$.

The Sublevel Set (SS) method

Framework Given clustering problem defined by Loss and data and convex relaxation with space $\mathcal{X}$.

Step 0 Cluster data, obtain a clustering $\mathcal{C}$.

$\mathcal{X}_{\leq c} = \{X \in \mathcal{X}, \text{Loss}(X) \leq c\}$ is a sublevel set of Loss



The Sublevel Set (SS) method

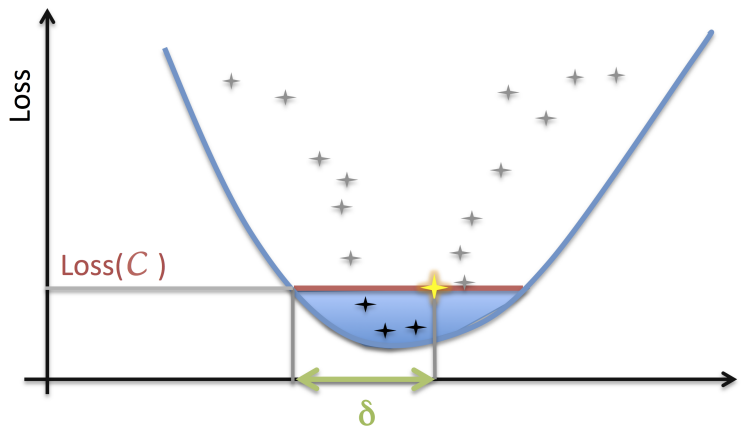
Framework Given clustering problem defined by Loss and data and convex relaxation with space $\mathcal{X}$.

Step 0 Cluster data, obtain a clustering $\mathcal{C}$.

Step 1 Use convex relaxation to define new optimization problem

$$\text{SS} \quad \delta = \max_{X' \in \mathcal{X}} \|X(\mathcal{C}) - X'\|_F, \quad \text{s.t. } \text{Loss}(X') \leq \text{Loss}(\mathcal{C}).$$

$\mathcal{X}_{\leq c} = \{X \in \mathcal{X}, \text{Loss}(X) \leq c\}$ is a sublevel set of



The Sublevel Set (SS) method

Framework Given clustering problem defined by Loss and data and convex relaxation with space $\mathcal{X}$.

Step 0 Cluster data, obtain a clustering $\mathcal{C}$.

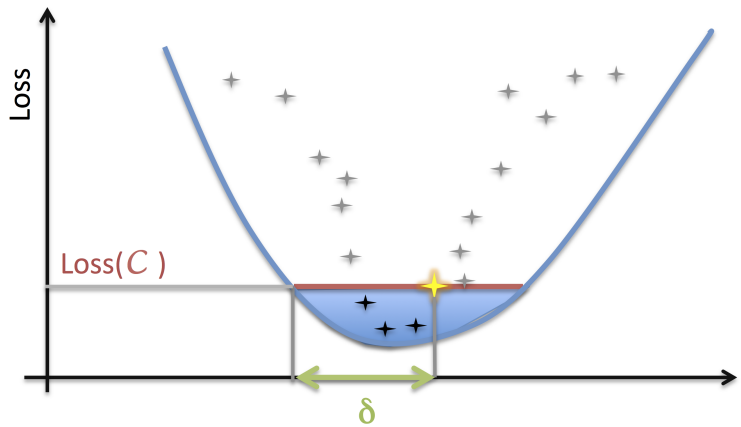
Step 1 Use convex relaxation to define new optimization problem

$$\text{SS} \quad \delta = \max_{X' \in \mathcal{X}} \|X(\mathcal{C}) - X'\|_F, \quad \text{s.t. } \text{Loss}(X') \leq \text{Loss}(\mathcal{C}).$$

Step 2 Prove that $\|X(\mathcal{C}) - X(\mathcal{C})'\|_F \leq \delta \Rightarrow d^{EM}(\mathcal{C}, \mathcal{C}') \leq \epsilon$ E.g. by [M, MLJ 2012]

Done: ϵ is a Optimality Interval (OI) for $\mathcal{C}$.

$\mathcal{X}_{\leq c} = \{X \in \mathcal{X}, \text{Loss}(X) \leq c\}$ is a sublevel set of



Loss

$||X(\mathcal{C}) - X'||$ generally *not concave*.

Relation with other work

- Previous ideas on OI
  - Spectral bounds for Spectral Clustering [M,Shortreed,Xu AISTATS05]
  - Spectral bounds for K-means, NCut and other quadratic costs [M, ICML06 and JMVA 2018]
  - Spectral bounds for networks model based clustering: Stochastic Block Model and Preference Frame Model [Wan,M NIPS16]
- Previous work we build on
  - Convex relaxations for clustering MANY! here we use SDP for K-means [Peng, Wei 2007]
  - Transforming bound on $||X - X'||_F$ into bound on d^{EM} [M MLJ 2012]
- Contrast with work on Clusterability and resilience, e.g. [Ben-David, 2015],[Bilu,Linial 2009]
  - clusterable data, resilient clustering $\approx$ stable $\mathcal{C}$
  - This work: given $\mathcal{C}$, prove it is stable
  - "Their" work: assume $\exists$ stable $\mathcal{C}$, prove it can be found efficiently

For what clustering paradigms can we obtain OI's?

Ways to map $\mathcal{C}$ to a matrix

space	matrix	definition	size
$\mathcal{X}$	$X(\mathcal{C})$	$X_{ij} = 1/n_k$ iff $i,j \in C_k$	$n \times n$, block-diagonal
$\tilde{\mathcal{X}}$	$\tilde{X}(\mathcal{C})$	$\tilde{X}_{ij} = 1$ iff $i,j \in C_k$	$n \times n$, block-diagonal
$\mathcal{Z}$	$Z(\mathcal{C})$	$Z_{ik} = 1/\sqrt{n_k}$ iff $i \in C_k$	$n \times K$, orthogonal

Theorem

[M NeurIPS 2018] If Loss has a convex relaxation involving one of $X, \tilde{X}, Z$, then

(1) There exists a convex SS problem

$$\text{(SS)} \quad \delta = \min_{X' \in \mathcal{X}_{\leq c}} \langle X(\mathcal{C}), X' \rangle \quad \text{(similarly for } \tilde{X}, Z\text{)}.$$

(2) From optimal δ an OI ε can be obtained, valid when $\varepsilon \leq p_{\min}$.

$$X : X_{ij} = 1/n_k \text{ iff } i,j \in C_k \qquad \varepsilon = (K - \delta)p_{\max}$$
$$\tilde{X} : \tilde{X}_{ij} = 1 \text{ iff } i,j \in C_k \qquad \varepsilon = \frac{\sum_{k \in [K]} n_k^2 + (n-K+1)^2 + (K-1) - 2\delta}{2p_{\min}}$$
$$Z : Z_{ik} = 1/\sqrt{n_k} \text{ iff } i \in C_k \qquad \varepsilon = (K - \delta^2/2)p_{\max}$$

Existence of guarantee depends only on space of convex relaxation.

K-means Sublevel Set problem and Optimality Interval

$$\begin{aligned} \text{Loss}(\mathcal{C}) &= \langle D, X(\mathcal{C})\rangle, \quad D = \text{squared distance matrix} \in \mathbb{R}^{n\times n} \\ X(\mathcal{C}) &= [X_{ij} = 1/n_k \text{ iff } i,j \in C_k] \in \mathbb{R}^{n\times n}, \text{block-diagonal} \\ \mathcal{X} &= \{\in \mathbb{R}^{n\times n},\ X \succeq 0, X_{ij} \geq 0,\ \text{trace}\, X = K,\ X\mathbf{1} = \mathbf{1}\} \end{aligned}$$

Sublevel Set problem

$$(\text{SS}_{\text{Km}}) \quad \delta = \min_{X'\in\mathcal{X}} \langle X(\mathcal{C}), X'\rangle \quad \text{s.t.} \langle D, X'\rangle \leq \text{Loss}(\mathcal{C})$$

This is a SDP.

K-means Sublevel Set problem and Optimality Interval

$$\begin{aligned} \text{Loss}(\mathcal{C}) &= \langle D, X(\mathcal{C})\rangle, \quad D = \text{squared distance matrix} \in \mathbb{R}^{n\times n} \\ X(\mathcal{C}) &= [X_{ij} = 1/n_k \text{ iff } i,j \in C_k] \in \mathbb{R}^{n\times n}, \text{block-diagonal} \\ \mathcal{X} &= \{\in \mathbb{R}^{n\times n},\ X \succeq 0, X_{ij} \geq 0,\ \text{trace}\, X = K,\ X\mathbf{1} = \mathbf{1}\} \end{aligned}$$

Sublevel Set problem

$$(\text{SS}_{\text{Km}}) \quad \delta = \min_{X'\in\mathcal{X}} \langle X(\mathcal{C}), X'\rangle \quad \text{s.t.} \langle D, X'\rangle \leq \text{Loss}(\mathcal{C})$$

This is a SDP.

Algorithm

Input Squared distance matrix D, clustering $\mathcal{C}$

1. Solve (SS_{Km}) let δ be the optimal value obtained.
2. **If** $\epsilon = (K-\delta)p_{\max} \leq p_{\min}$ **then** $\mathcal{C}$ is stable

   else no guarantee.

K-means Sublevel Set problem and Optimality Interval

$$\begin{aligned} \text{Loss}(\mathcal{C}) &= \langle D, X(\mathcal{C})\rangle, \quad D = \text{squared distance matrix} \in \mathbb{R}^{n\times n} \\ X(\mathcal{C}) &= [X_{ij} = 1/n_k \text{ iff } i,j \in C_k] \in \mathbb{R}^{n\times n}, \text{block-diagonal} \\ \mathcal{X} &= \{\in \mathbb{R}^{n\times n},\ X \succeq 0, X_{ij} \geq 0,\ \text{trace}\, X = K,\ X\mathbf{1} = \mathbf{1}\} \end{aligned}$$

Sublevel Set problem

$$(\text{SS}_{\text{Km}}) \quad \delta = \min_{X'\in\mathcal{X}} \langle X(\mathcal{C}), X'\rangle \quad \text{s.t.} \langle D, X'\rangle \leq \text{Loss}(\mathcal{C})$$

This is a SDP.

Algorithm

Input Squared distance matrix D, clustering $\mathcal{C}$

1. Solve (SS_{Km}) let δ be the optimal value obtained.
2. **If** $\epsilon = (K-\delta)p_{\max} \leq p_{\min}$ **then** $\mathcal{C}$ is stable

   else no guarantee.

Theorem 1

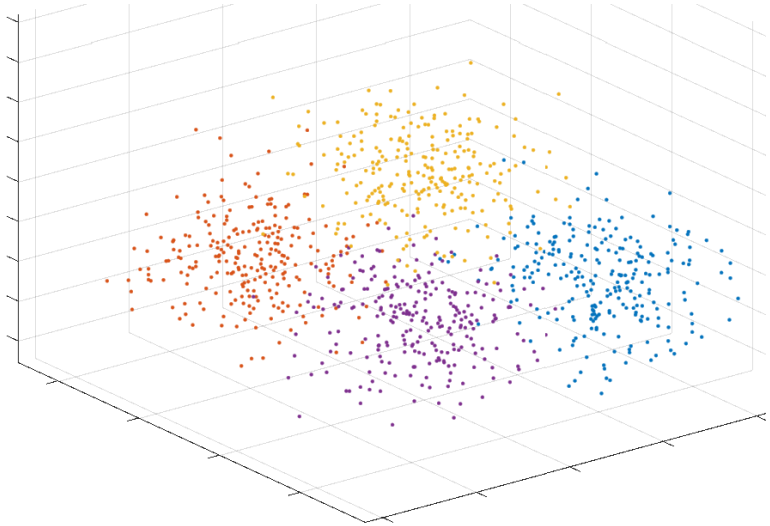
If $\varepsilon \leq p_{\min}$,

then $d^{EM}(\mathcal{C}, \mathcal{C}') \leq \varepsilon$ for any $\mathcal{C}'$ with $\text{Loss}(\mathcal{C}') \leq \text{Loss}(\mathcal{C})$.

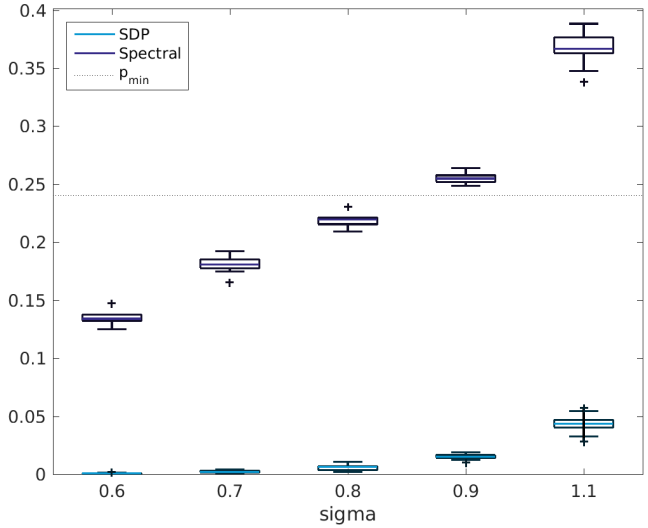
Results for K-means clusterings

$K = 4$ equal Gaussian clusters, $n = 1024$, $||\mu_k - \mu_l|| = 4\sqrt{2} \approx 5.67$

data for $\sigma = 0.9$

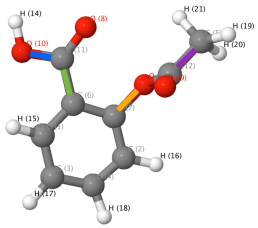


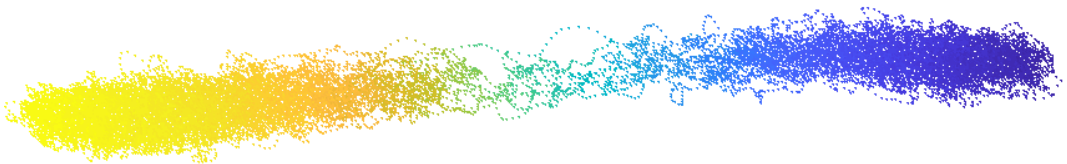
Values of ϵ vs cluster spread σ



Spectral=[M ICML06], SDP=[M NeurIPS 2018]

Aspirin ($C_9O_4H_8$) molecular simulation data [Chmiela et al. 2017]





$K = 2$

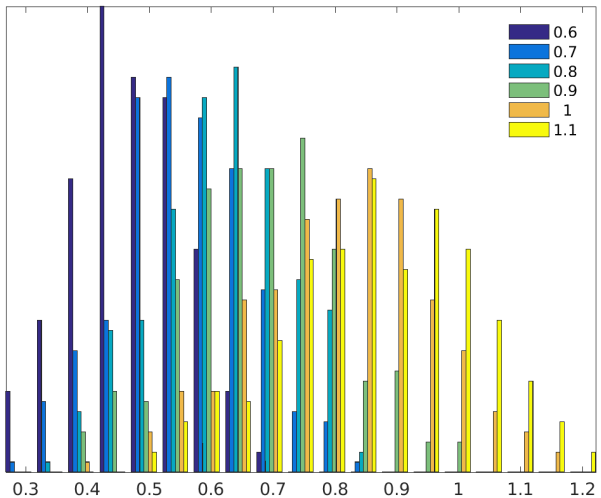
$p_{\min} = .26$

$p_{\max} = .74$

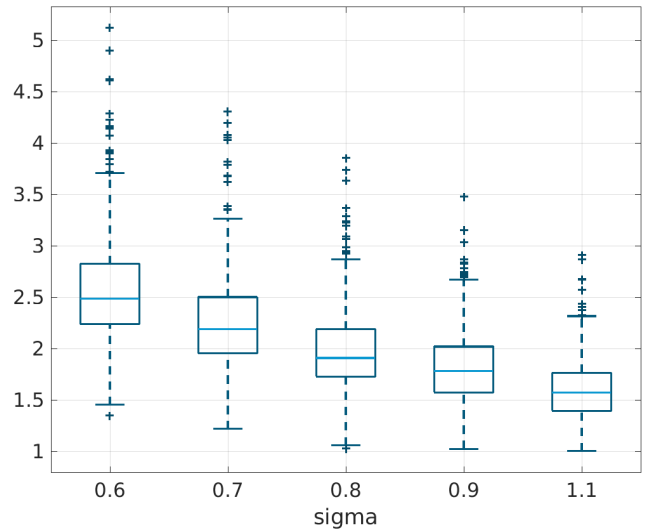
$n = 2118$ $\varepsilon = 0.065$

Separation statistics

distance to own center over min center separation, colored by σ.



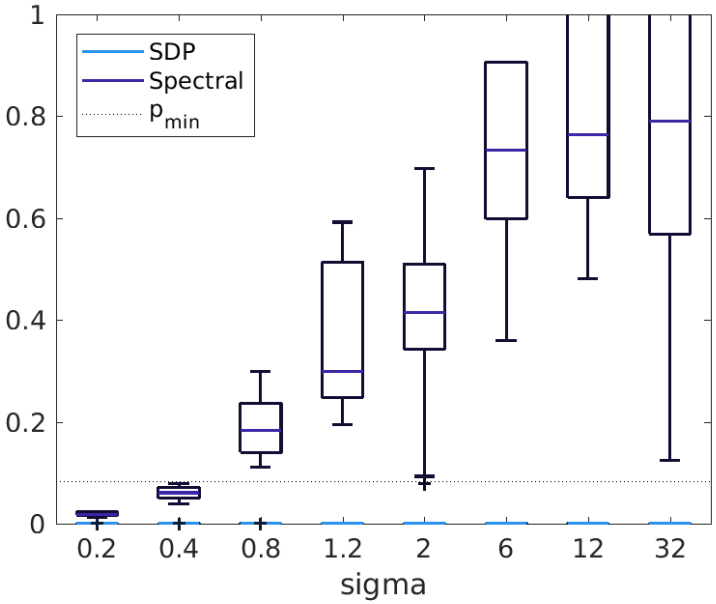
distance to second closest center over distance to own center, versus σ

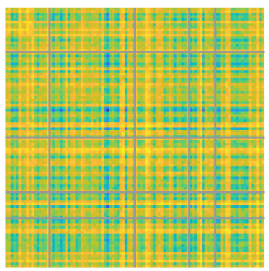


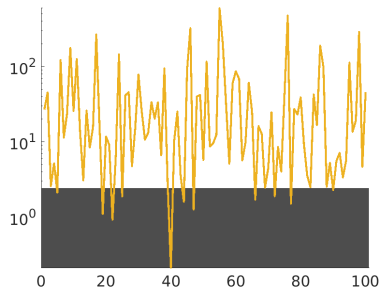
Results for Spectral Clustering by Normalized Cut

Spectral=[M AISTATS05], SDP=[M NeurIPS 2018]

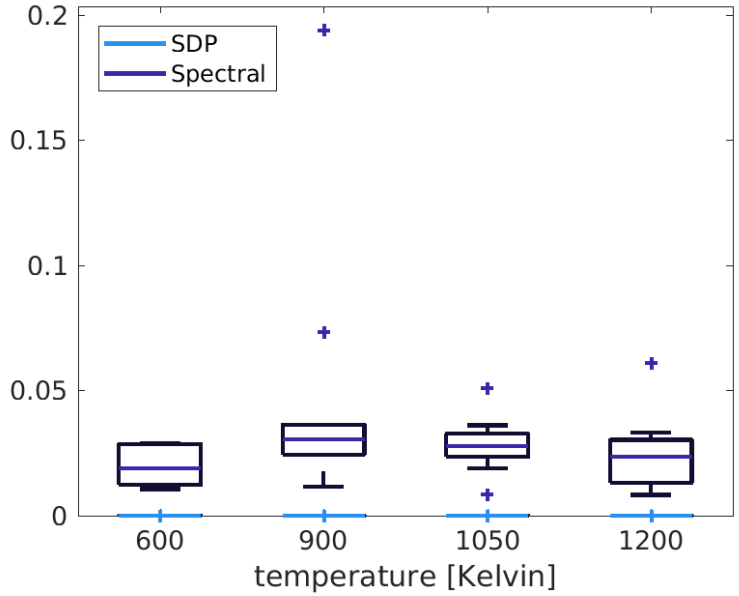
Synthetic S, $n = 100$

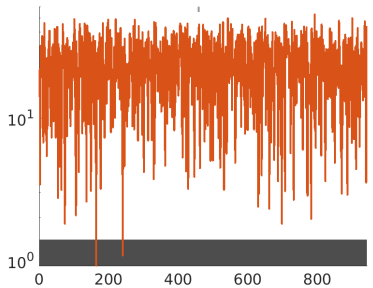


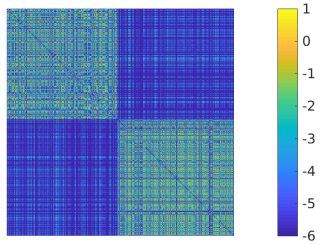




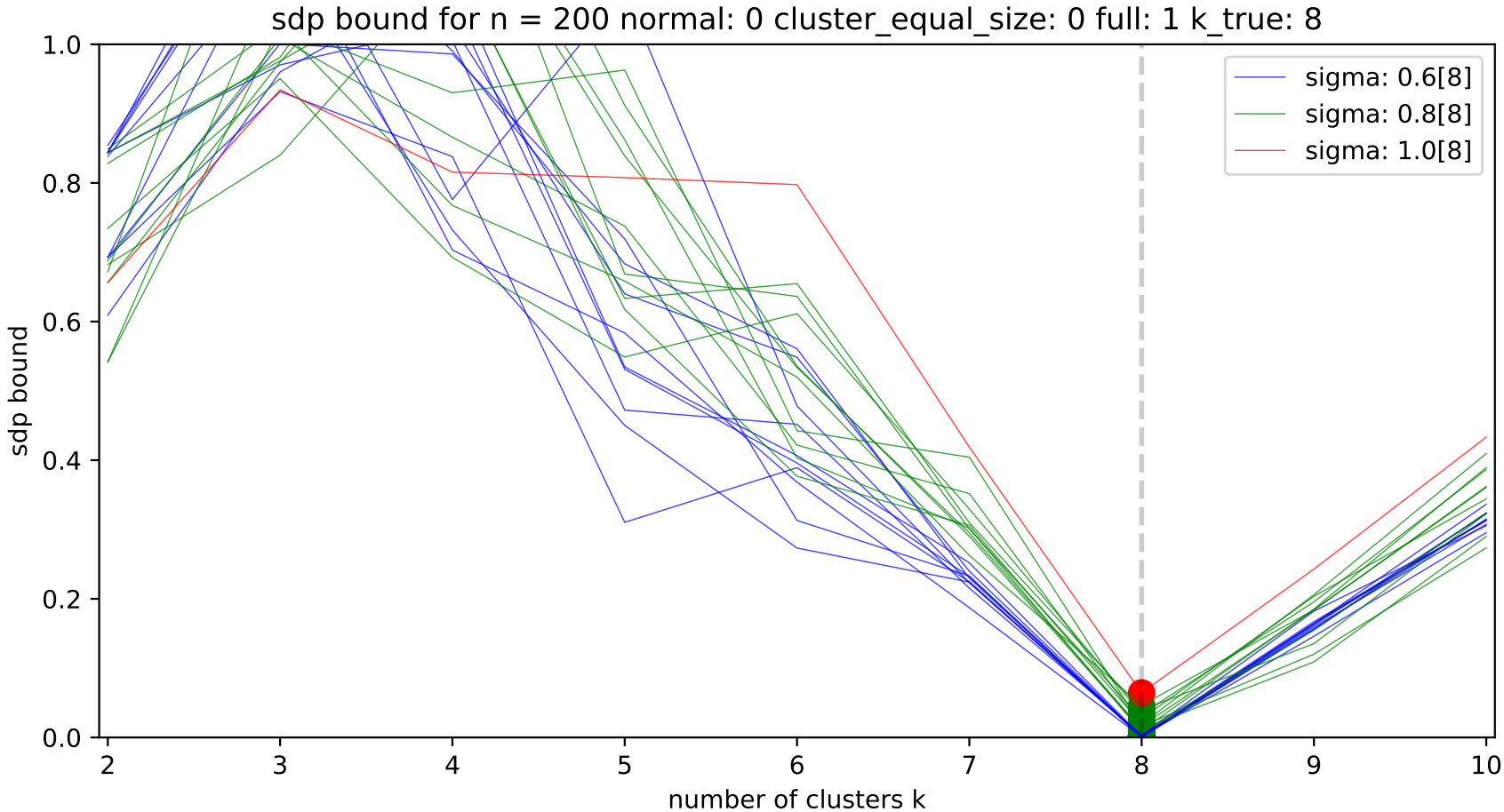
Chemical reaction data, $n \approx 1000$







Stability and the selection of K (in preparation)



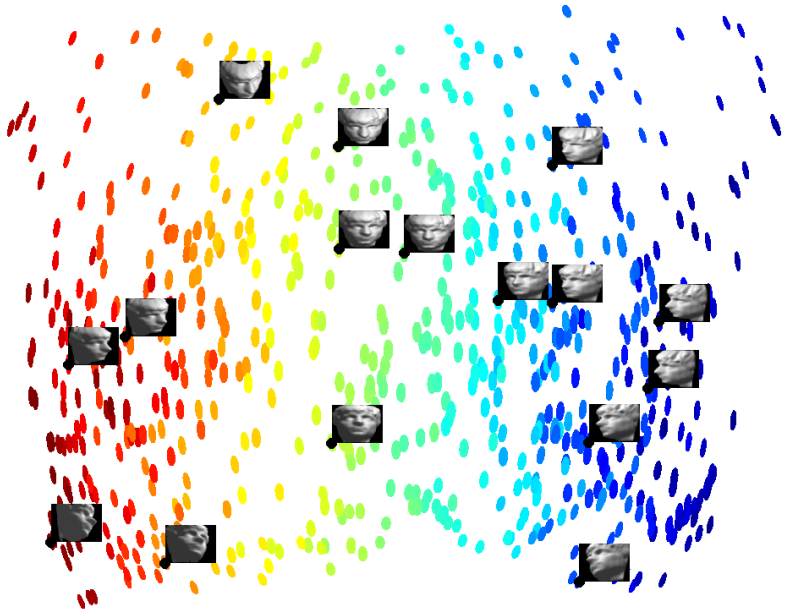
Outline

Stability guarantees for clustering [M NeurIPS 2018]
provable "correctness" for the practitioner

Metric manifold learning [Perrault-Joncas,M arXiv:1305.7255]
"coordinate independent" geometric recovery

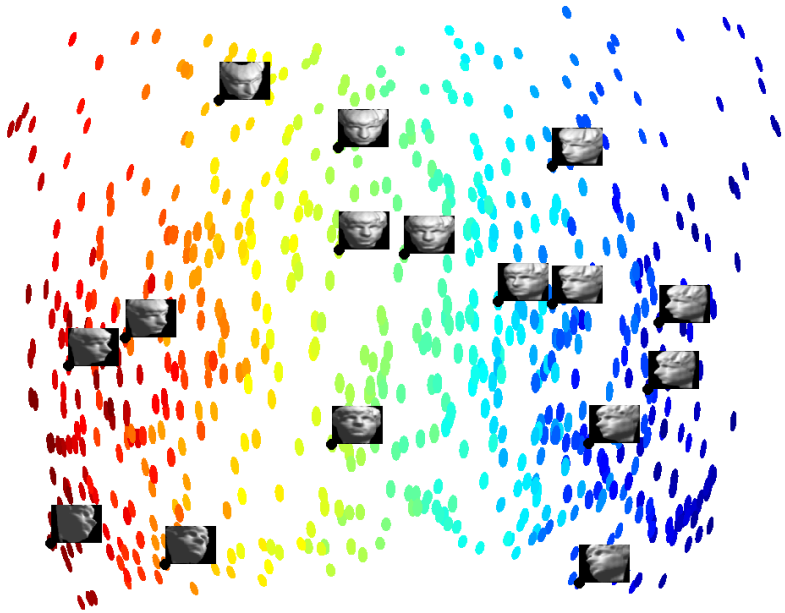
Manifold coordinates with physical meaning [M,Koelle,Zhang arXiv:1811.11891]
interpretability in the language of the problem

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

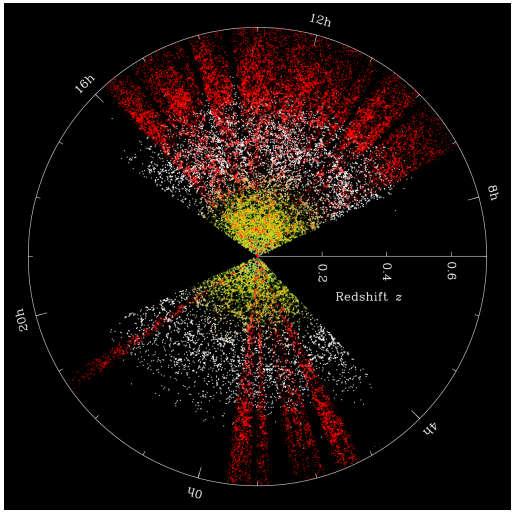


Why?

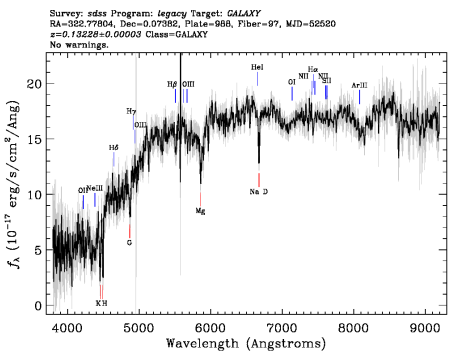
- To save space and computation
  - $n \times D$ data matrix $\rightarrow$ $n \times s$, $s \ll D$
- To use it afterwards in (prediction) tasks
- To understand the data better
  - preserve large scale features, suppress fine scale features

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

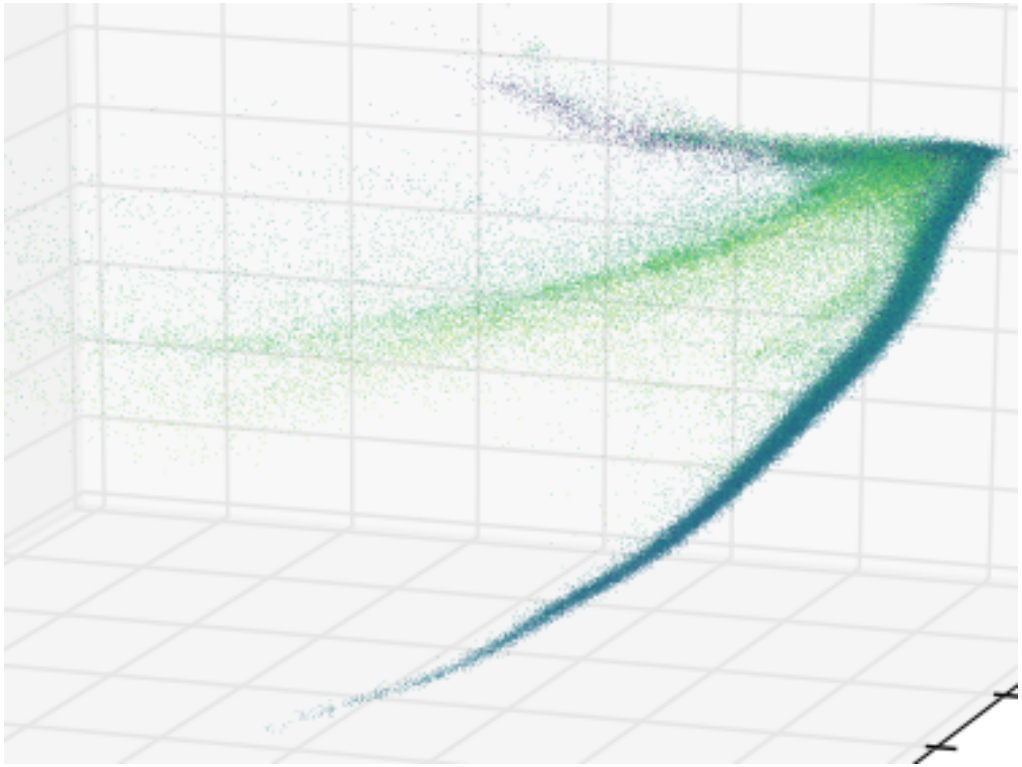
www.sdss.org



www.sdss.org

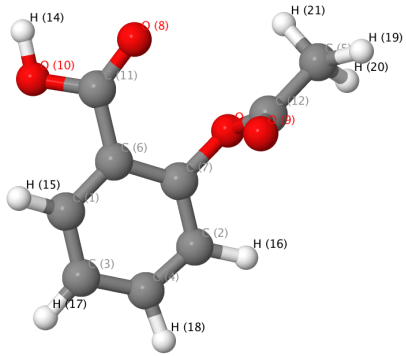


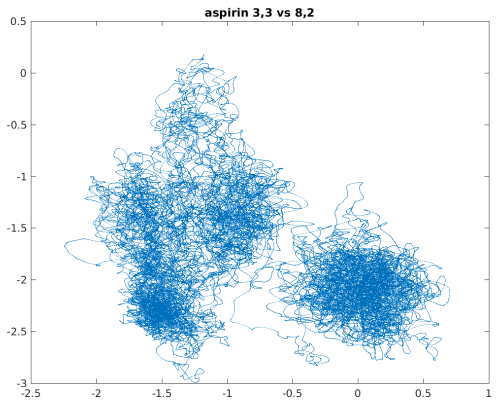
- Preprocessed by Jacob VanderPlas and Grace Telford
- $n = 675,000$ spectra $\times$ $D = 3750$ dimensions



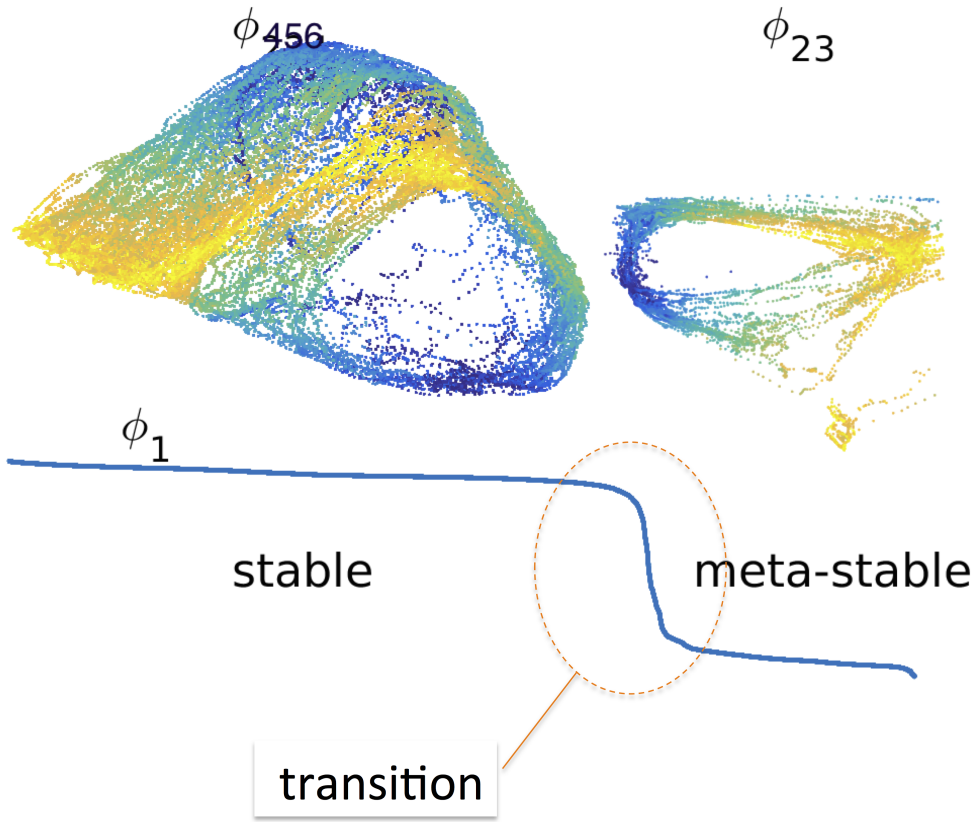
Molecular configurations

aspirin molecule





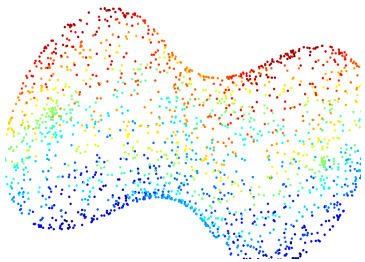
- Data from Molecular Dynamics (MD) simulations of small molecules by [Chmiela et al. 2016]
- $n \approx 200,000$ configurations $\times$ $D \sim 20 - 60$ dimensions



Brief intro to manifold learning algorithms

ALL ML Algorithms

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

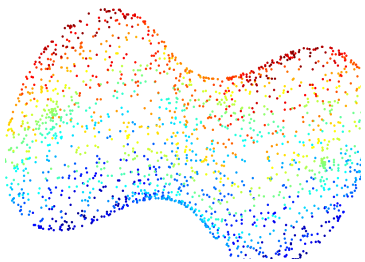


$p_1, \ldots p_n \subset \mathbb{R}^D$

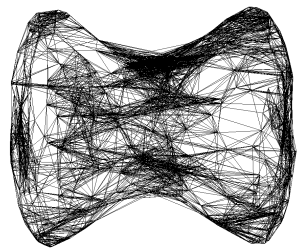
Brief intro to manifold learning algorithms

ALL ML Algorithms

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



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



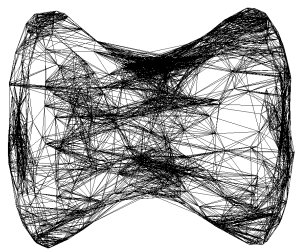
Brief intro to manifold learning algorithms

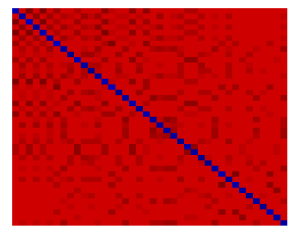
ALL ML Algorithms

- **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 $n \times n$ sparse distance matrix

$$D = [||p - p'||]_{p,p' \text{neighbors}}$$

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





Brief intro to manifold learning algorithms

ALL ML 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 $n \times n$ sparse distance matrix

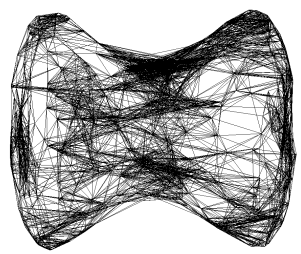
$$D = [||p - p'||]_{p,p' \text{neighbors}}$$

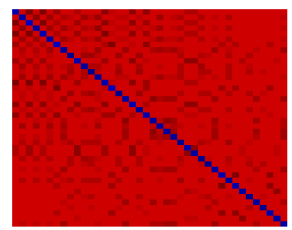
- Optional: construct kernel matrix, .e.g

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

and Laplacian matrix

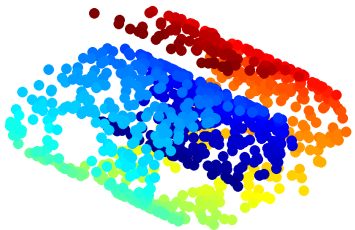
$p_1, \ldots p_n \subset \mathbb{R}^D$



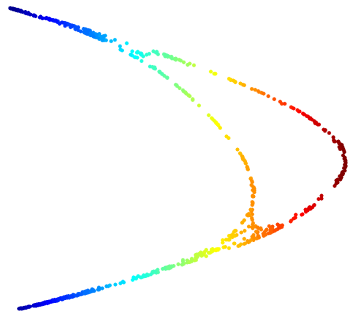


Embedding in 2 dimensions by different manifold learning algorithms

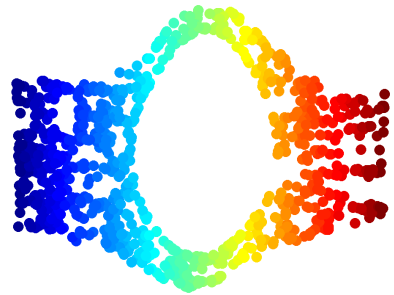
Original data
(Swiss Roll with hole)



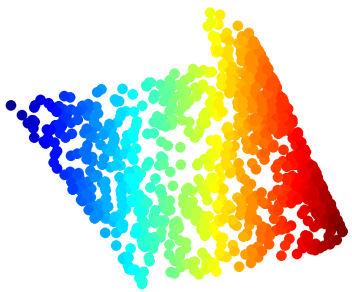
Laplacian Eigenmaps
(LE)



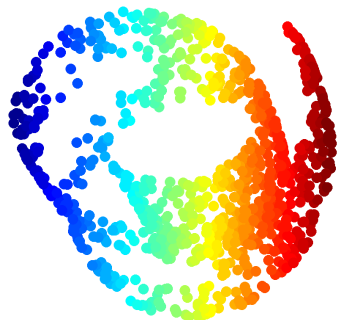
Isomap



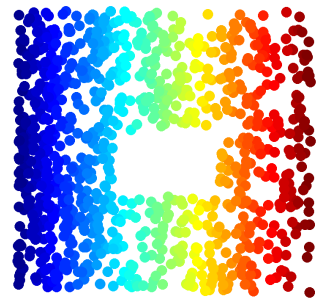
Hessian Eigenmaps (HE)



Local Linear Embedding
(LLE)



Local Tangent Space
Alignment (LTSA)

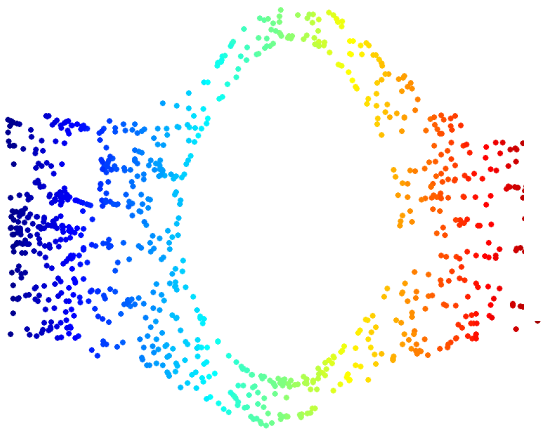


Preserving topology vs. preserving (intrinsic) geometry

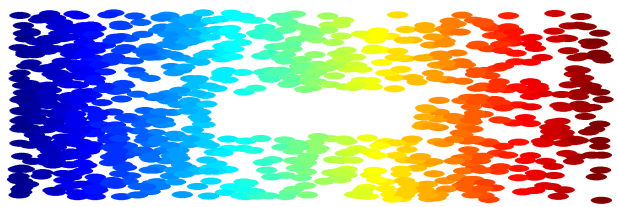
- Algorithm maps data $p \in \mathbb{R}^D \longrightarrow \phi(p) = x \in \mathbb{R}^m$

- 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

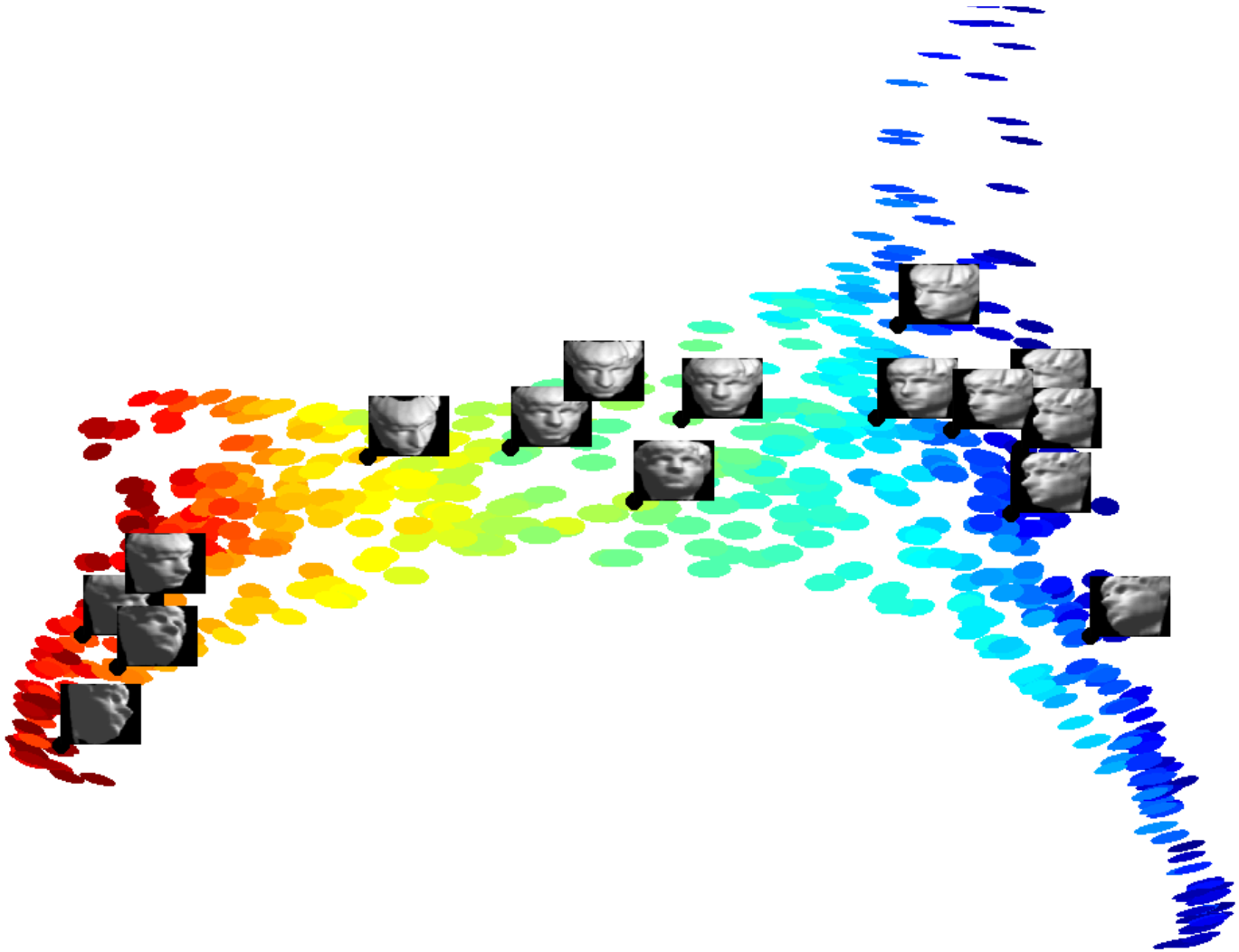
- augment ϕ with geometric information g so that (ϕ, g) preserves the geometry

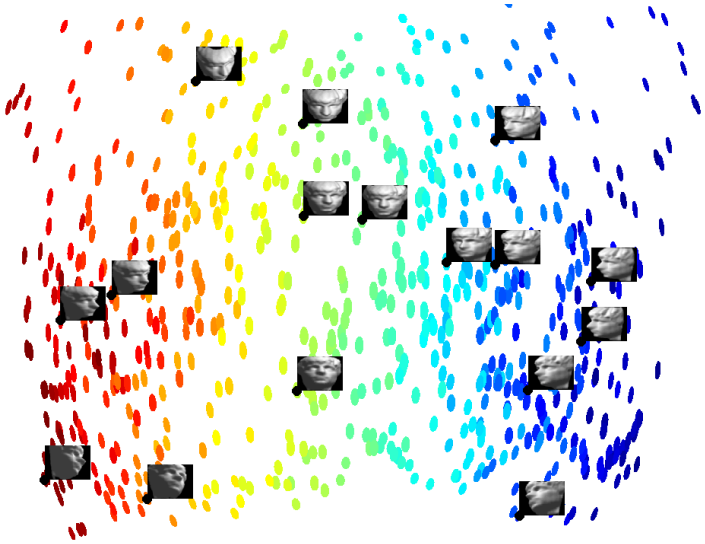
g is the Riemannian metric.

Dominique Perrault-Joncas

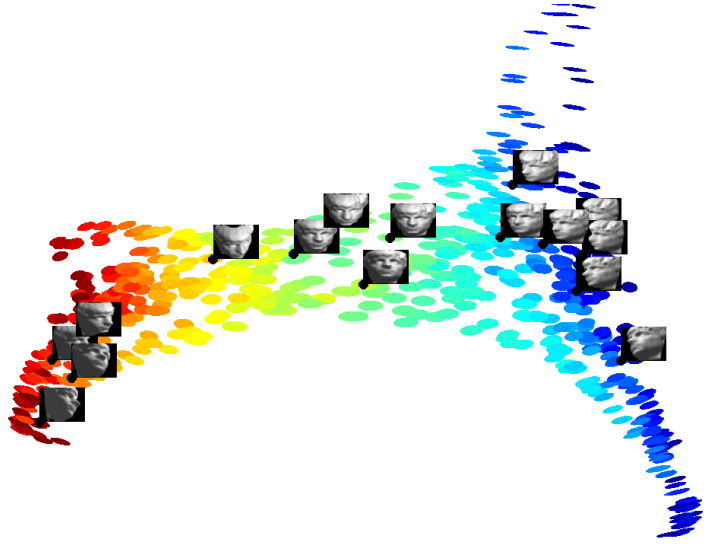
g for Sculpture Faces

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

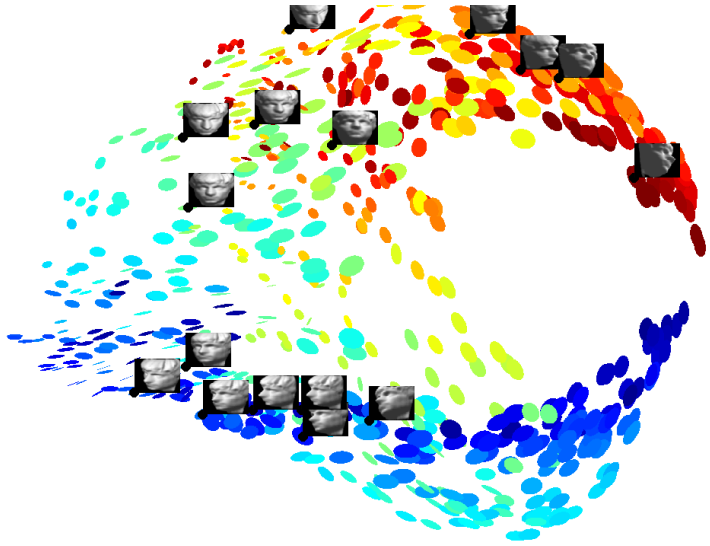




Isomap



LTSA



Laplacian Eigenmaps

Relation between g and Δ

- $\Delta =$ 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

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

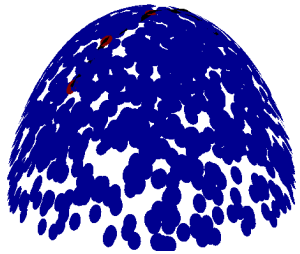
where $h = g^{-1}$ (matrix inverse) and $k, l = 1, 2, \ldots m$ are embedding dimensions

Intuition:

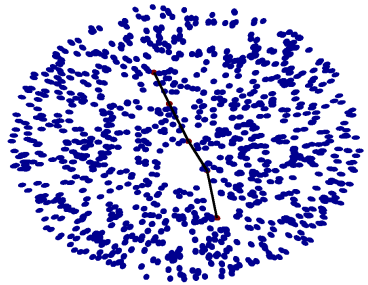
- at each point $p \in \mathcal{M}$, $G(p)$ is a $d \times d$ matrix
- apply Δ to embedding coordinate functions $\phi_1, \ldots \phi_m$
- this produces $G^{-1}(p)$ in the given coordinates
- our algorithm implements matrix version of this operator result
- consistent estimation of Δ is well studied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

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

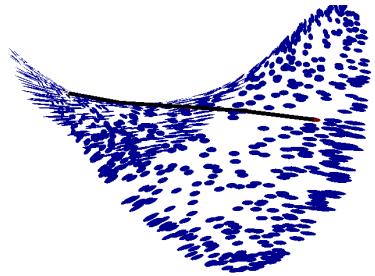
Original



Isomap



Laplacian Eigenmaps

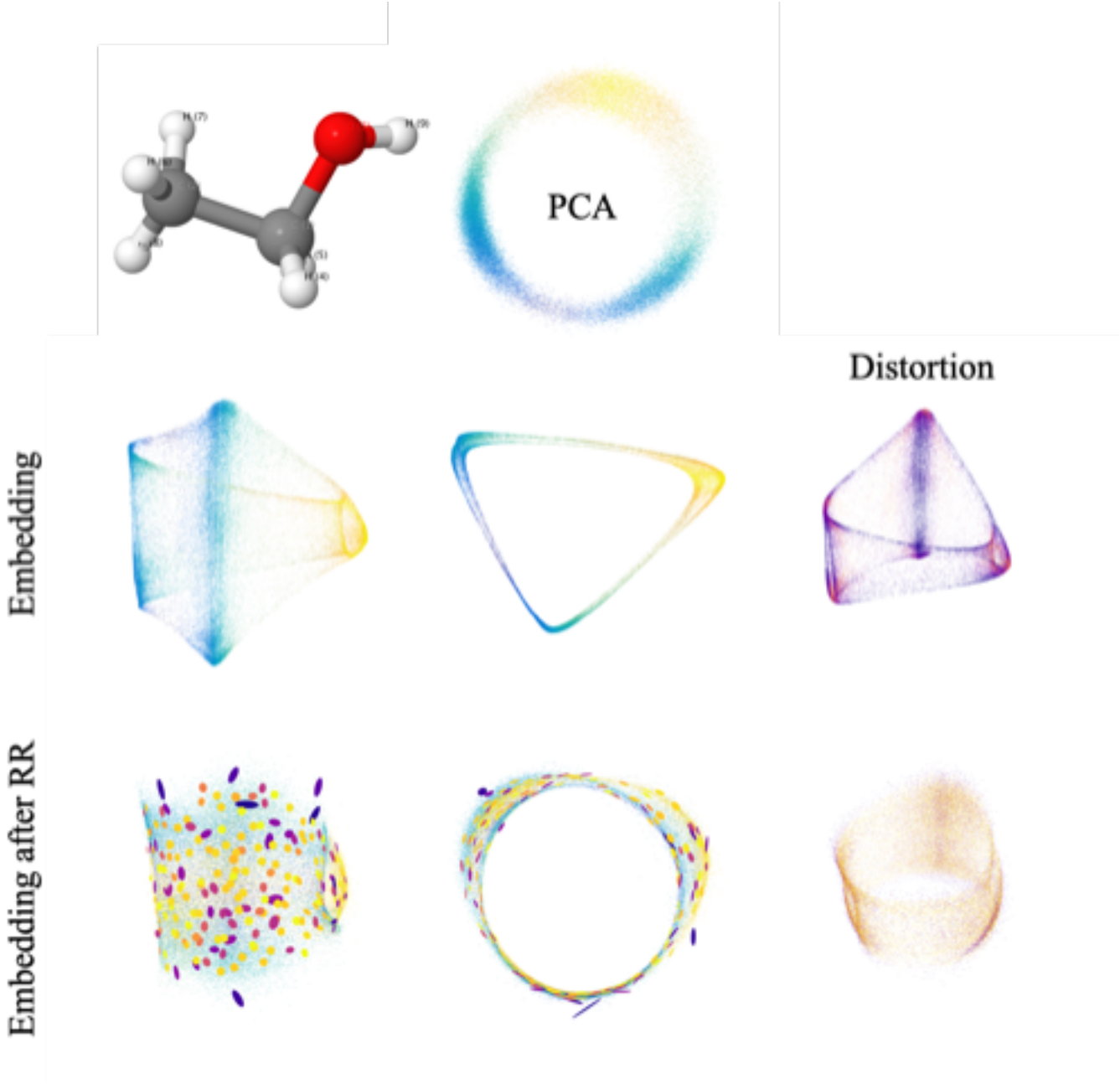


true distance $d = 1.57$

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%

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

Riemannian Relaxation for Ethanol molecular configurations



Metric Manifold Learning summary

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

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

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

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

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

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 (in progress), selecting eigencoordinates [Chen, M NeurIPS19]

Outline

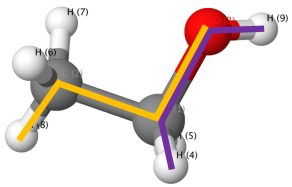
Stability guarantees for clustering [M NeurIPS 2018]
provable "correctness" for the practitioner

Metric manifold learning [Perrault-Joncas,M arXiv:1305.7255]
"coordinate independent" geometric recovery

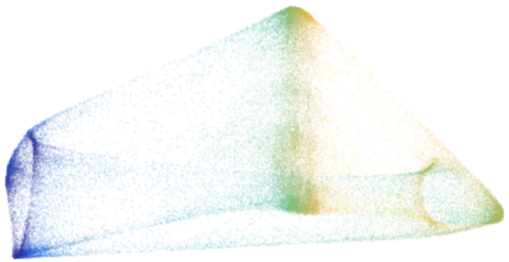
Manifold coordinates with physical meaning [M,Koelle,Zhang arXiv:1811.11891]
interpretability in the language of the problem

Motivation

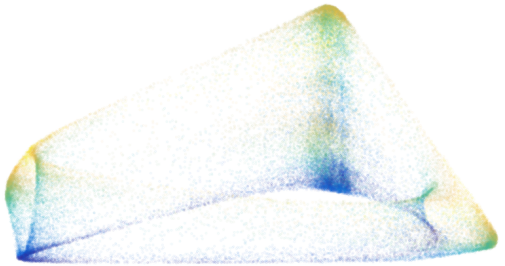
ethanol



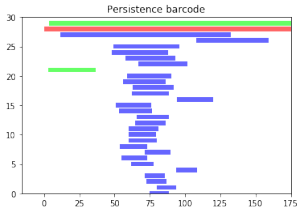
torsion 1



torsion 2



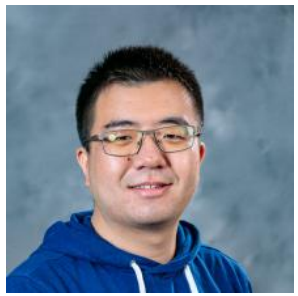
persistence



- 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?

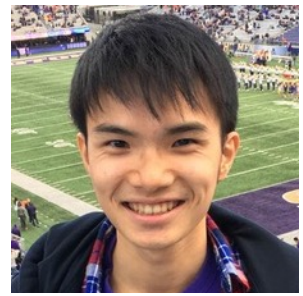
Problem formulation

Hanyu Zhang



Sam Koelle

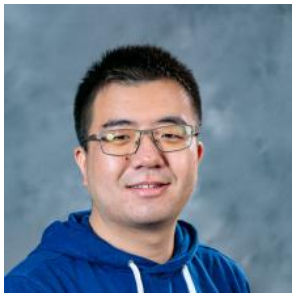
Yu-chia Chen



- 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

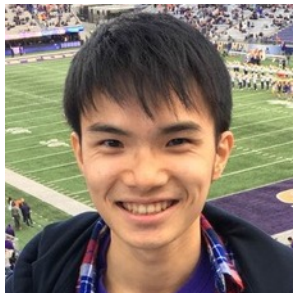
Problem formulation

Hanyu Zhang



Sam Koelle

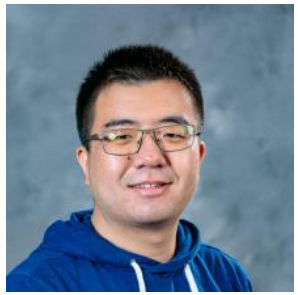
Yu-chia Chen



- Given
  - data $\xi_i \in \mathbb{R}^D$, $i \in 1 \dots n$
  - embedding of data $\phi(\xi_{1:n})$ in $\mathbb{R}^m$
- dictionary of domain-related smooth functions $\mathcal{G} = \{g_1, \dots g_p, \text{ with } g_j : \mathbb{R}^D \to \mathbb{R}\}$.
  - 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

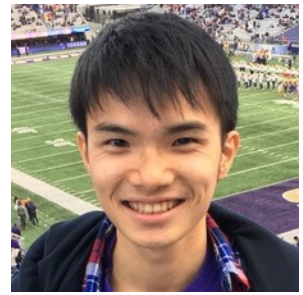
Problem formulation

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 $\mathbb{R}^m$
- dictionary of domain-related smooth functions $\mathcal{G} = \{g_1, \dots g_p, \text{ with } g_j : \mathbb{R}^D \to \mathbb{R}\}$.
  - 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 $\phi_1 \dots \phi_m$ in terms of functions in $\mathcal{G}$.
  More precisely, we assume that

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

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

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

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

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

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

- 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

$$\boxed{\phi = h \circ g}$$

  - (sparse non-linear, non-parametric recovery)
- then

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

  - sparse linear recovery

First Idea: from non-linear to linear

- If

  $\boxed{\phi = h \circ g}$

  - (sparse non-linear, non-parametric recovery)
- then

  $\boxed{D\phi = DhDg}$

  - 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

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

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

Multidimensional FLasso

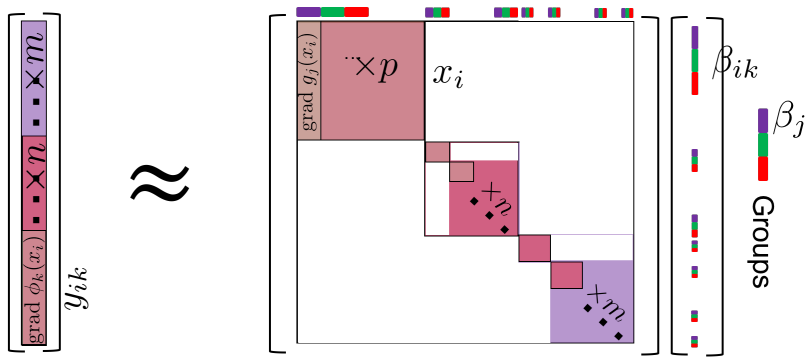
- Assume

$$y_{ik} = \nabla f_k(\xi_i) \quad X_i = \nabla g_{1:p}(\xi) \quad \beta_{\beta_{ijk}} = \frac{\partial h}{\partial g_j}(\xi_i) \tag{3}$$

and

$$\beta_j = \mathrm{vec}(\beta_{ijk},\ i = 1:n, k = 1:m) \in \mathbb{R}^{mn}, \quad \beta_{ik} = \mathrm{vec}(\beta_{ijk},\ j = 1:p) \in \mathbb{R}^{p}. \tag{4}$$

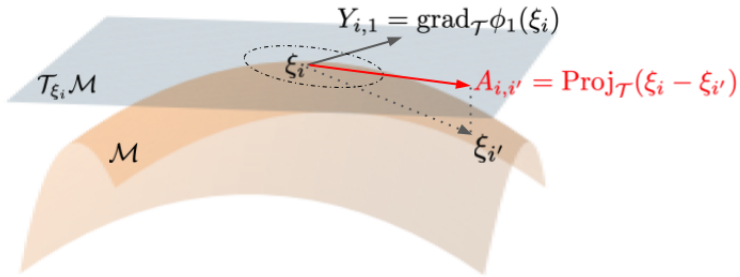
$$J_\lambda(\beta) = \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{m} ||y_{ik} - X_i\beta_{ik}||^2 + \frac{\lambda}{\sqrt{mn}}\sum_{j=1}^{p}||\beta_j||. \tag{5}$$



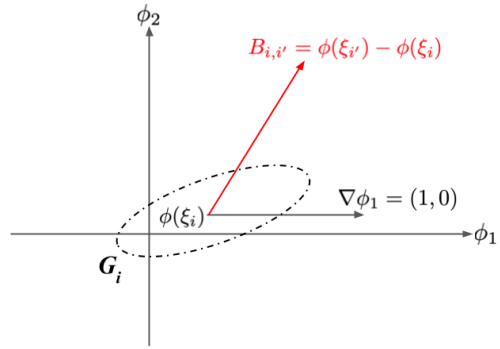
FLasso in manifold setting

- gradients $\nabla \rightarrow$ 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 $\mathcal{T}_{\xi_i}\mathcal{M}$

$\mathcal{M}$



$\phi(\mathcal{M})$



Theory

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

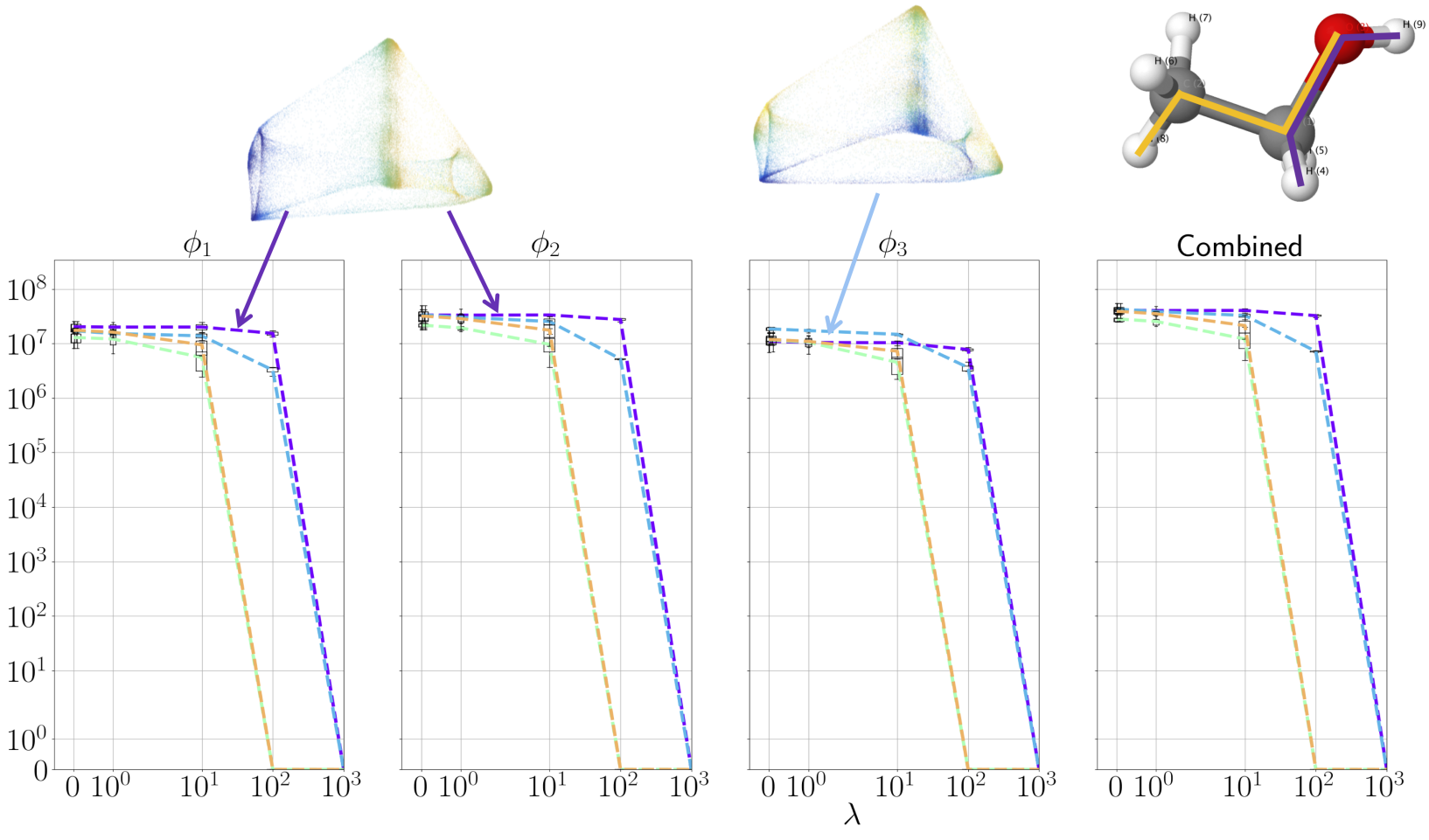
$$\text{rank} \begin{pmatrix} Dg_S \\ Dg_{S'} \end{pmatrix} = \text{rank}\, Dg_{S'} \quad \text{on } U$$

- When can FLasso recover S ?
  Incoherence conditions

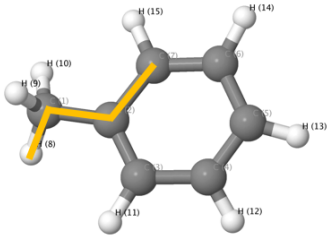
$$\mu = \max_{i=1:n, j \in S, j' \notin S} |X_{ji}^T X_{j'i}| \quad \nu = \frac{1}{\min_{i=1:n} ||X_{iS}^T X_{iS}||_2} \quad nd\sigma^2 = \sum_{i,k} \epsilon_{ik}^2$$

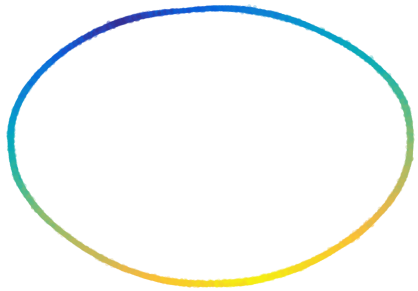
  Theorem If $\mu\nu\sqrt{s} + \frac{\sigma\sqrt{nd}}{\lambda} < 1$ then $\beta_j = 0$ for $j \notin S$.

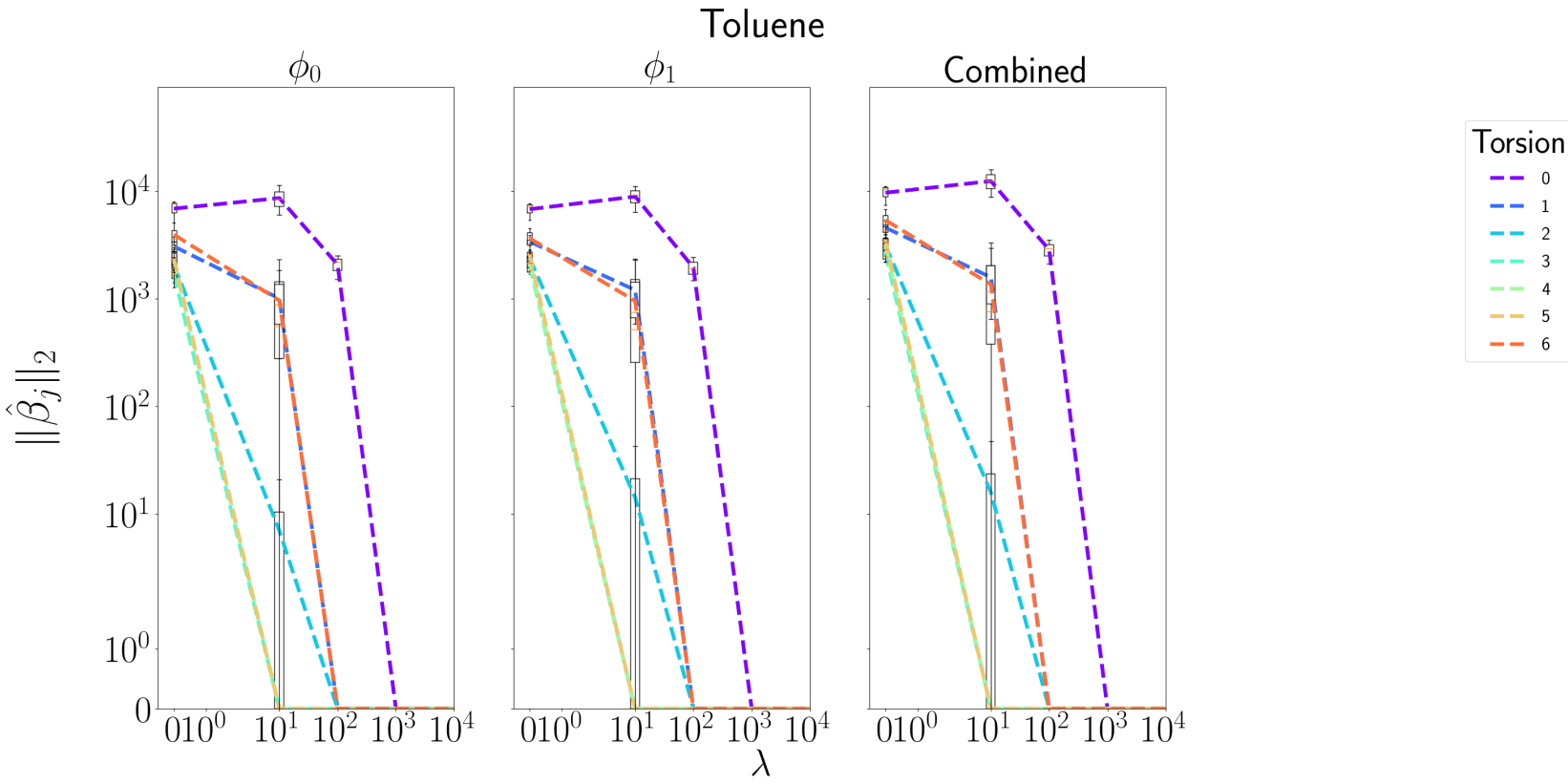
Ethanol MD simulation



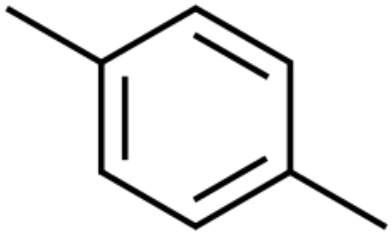
Toluene MD simulation

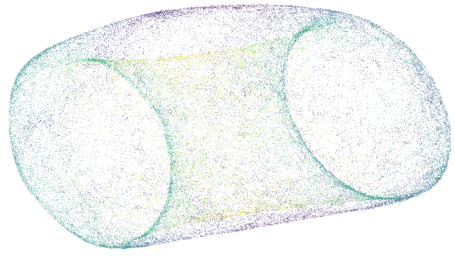


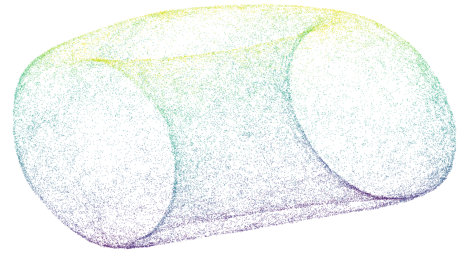


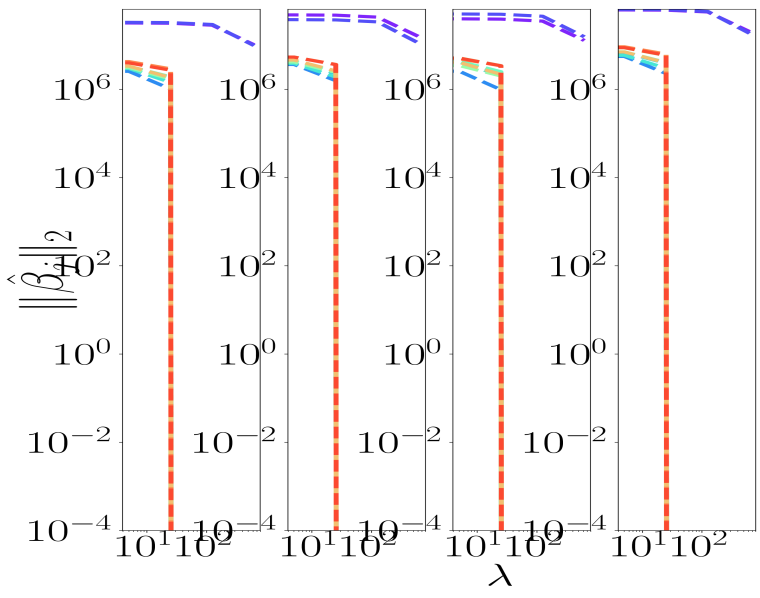


Para-xilene MD simulation

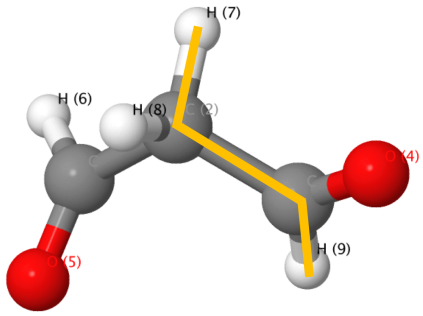


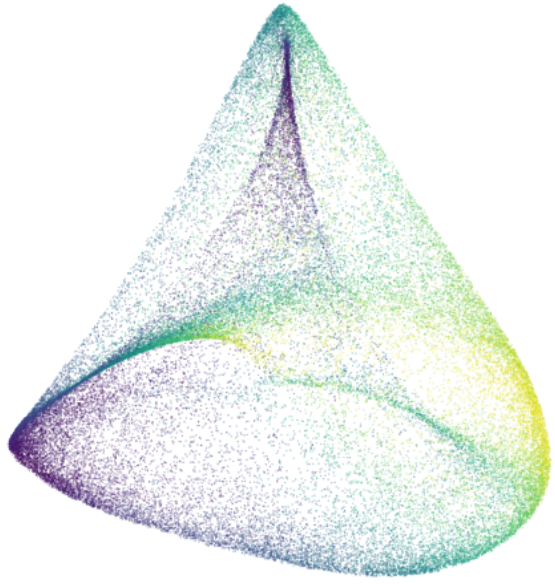


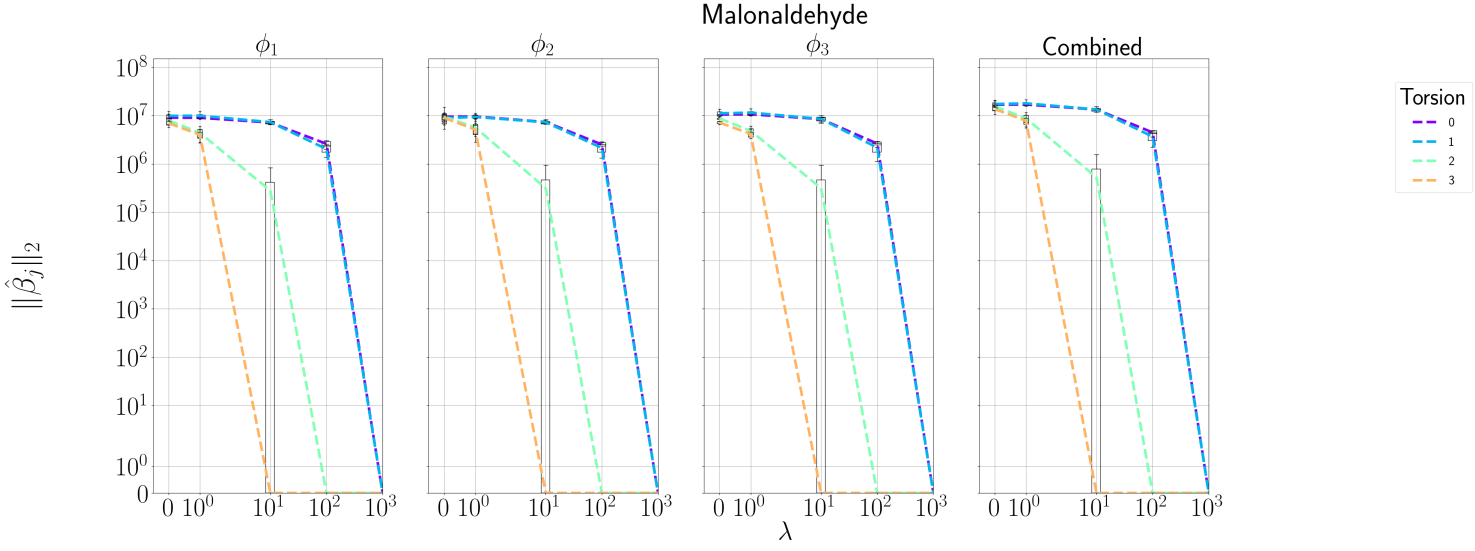




Malondialdehyde MD simulation







Summary Cluster validation without model assumptions [M NeurIPS 2018]

- A general method that can be applied to any clustering cost that has a convex relaxation

Summary Cluster validation without model assumptions [M NeurIPS 2018]

- A general method that can be applied to any clustering cost that has a convex relaxation

Metric Manifold learning

- Before embedding: choice of kernel width ϵ [Perrault-Joncas,McQueen,M 17], choice of intrinsic dimension d
- Simultaneously with embedding: Gaussian process prediction, estimating vector fields [Perrault-Joncas,M 10], eigenfunctions vs. embedding coordinates [M,Chen NeurIPS19]
- After embedding: estimate distortion by H and correct it by Riemannian Relaxation [Perrault-Joncas,M 10, McQueen,Perrault-Joncas,M 16]

Summary Cluster validation without model assumptions [M NeurIPS 2018]

- A general method that can be applied to any clustering cost that has a convex relaxation

Metric Manifold learning

- Before embedding: choice of kernel width ϵ [Perrault-Joncas,McQueen,M 17], choice of intrinsic dimension d
- Simultaneously with embedding: Gaussian process prediction, estimating vector fields [Perrault-Joncas,M 10], eigenfunctions vs. embedding coordinates [M,Chen NeurIPS19]
- After embedding: estimate distortion by H and correct it by Riemannian Relaxation [Perrault-Joncas,M 10, McQueen,Perrault-Joncas,M 16]

Manifold coordinates with pysical meaning [arXiv:1811.11891]

- Interpretation in the language of the domain
- From non-parametric to parametric

Python package `github.com/mmp2/megaman`

- tractable for millions of points
- manifold learning and clustering
- incorporates state of the art results

Towards unsupervised validation for unsupervised learning

- In Machine Learning: Unsupervised Learning is the next big challenge
- In the sciences: Unsupervised Learning is about explanation and understanding

Towards unsupervised validation for unsupervised learning

- In Machine Learning: Unsupervised Learning is the next big challenge
- In the sciences: Unsupervised Learning is about explanation and understanding
- Automated discoveries require automated validation
  - With domain knowledge
  - On purely mathematical/statistical grounds

Towards unsupervised validation for unsupervised learning

- In Machine Learning: Unsupervised Learning is the next big challenge
- In the sciences: Unsupervised Learning is about explanation and understanding
- Automated discoveries require automated validation
  - With domain knowledge
  - On purely mathematical/statistical grounds
- Remove algorithmic artefacts

Towards unsupervised validation for unsupervised learning

- In Machine Learning: Unsupervised Learning is the next big challenge
- In the sciences: Unsupervised Learning is about explanation and understanding
- Automated discoveries require automated validation
  - With domain knowledge
  - On purely mathematical/statistical grounds
- Remove algorithmic artefacts
- Quantitative measures of "correctness" / robustness / uncertainty

Towards unsupervised validation for unsupervised learning

- In Machine Learning: Unsupervised Learning is the next big challenge
- In the sciences: Unsupervised Learning is about explanation and understanding
- Automated discoveries require automated validation
  - With domain knowledge
  - On purely mathematical/statistical grounds
- Remove algorithmic artefacts
- Quantitative measures of "correctness" / robustness / uncertainty
- Is explanation unique?

Towards unsupervised validation for unsupervised learning

- In Machine Learning: Unsupervised Learning is the next big challenge
- In the sciences: Unsupervised Learning is about explanation and understanding
- Automated discoveries require automated validation
  - With domain knowledge
  - On purely mathematical/statistical grounds
- Remove algorithmic artefacts
- Quantitative measures of "correctness" / robustness / uncertainty
- Is explanation unique?
- Statistical guarantees – without untestable assumptions

Towards unsupervised validation for unsupervised learning

- In Machine Learning: Unsupervised Learning is the next big challenge
- In the sciences: Unsupervised Learning is about explanation and understanding
- Automated discoveries require automated validation
  - With domain knowledge
  - On purely mathematical/statistical grounds
- Remove algorithmic artefacts
- Quantitative measures of "correctness" / robustness / uncertainty
- Is explanation unique?
- Statistical guarantees – without untestable assumptions
- Good community practices – all machine learning algorithms should come with validation procedures

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

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

Thank you