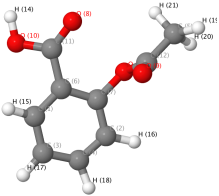
From local to global in clustering and dimension reduction

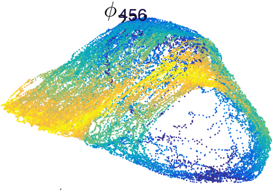
Hanyu Zhang

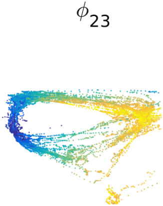
University of Washington
hanyuz6@uw.edu
with the Geometric Data Analysis Group
Marina Meila, Dominique Perrault-Joncas, James McQueen, Yu-chia Chen, Samson Koelle

From Local to Global Information Research Workshop 2/6/2020

A motivating example: embedding of MD simulation data of aspirin







- local to global in clustering and dimension reduction.
- Clustering: local similarity to find groups.
- Manifold Learning: local neighborhood to find global embedding.

Unsupervised learning for scientific data

- Understanding structure of data is typical for science.
- Unsupervised learning aims to find structure in data: clusters, low dimensionality, sparsity, causality, etc.
- Find knowledge that is non-specific to task or current query.

- Think as a scientist, answers cannot be crowdsourced:
  - In the least, should be free of artifacts
  - Ideally, should have guarantees without untestable model assumptions

Unsupervised learning for scientific data

- Understanding structure of data is typical for science.
- Unsupervised learning aims to find structure in data: clusters, low dimensionality, sparsity, causality, etc.
- Find knowledge that is non-specific to task or current query.

- Think as a scientist, answers cannot be crowdsourced:
  - In the least, should be free of artifacts
  - Ideally, should have guarantees without untestable model assumptions

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

Geometry Data Analysis (GDA) for unsupervised learning

- Unsupervised learning aims to find structure in data: clusters, low dimensionality, sparsity, causality, etc
- Convex analysis for clustering.
  - Local optimum to guarantee global optimality
- Differential geometry for Manifold Learning (ML)
  - Local metric to preserve geometry
  - Local tangent space to find global coordinates with physical meaning
- (Not dicussed) topological data analysis

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

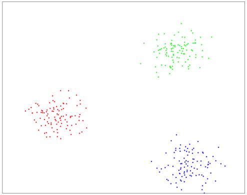
- IDEALLY WANTED: guarantee that $\mathcal{C}$ is correct/optimal
- WHAT WE CAN DO: guarantee that $\mathcal{C}$ is approximately correct/optimal

For the practitioner of clustering

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

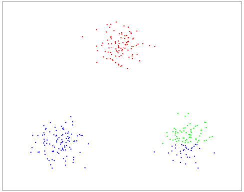
- IDEALLY WANTED: guarantee that $\mathcal{C}$ is correct/optimal
- WHAT WE CAN DO: guarantee that $\mathcal{C}$ is approximately correct/optimal
- WHEN $\mathcal{C}$ is good and stable

Good, stable



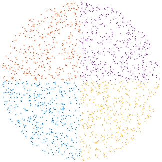
SS output: OI=$1e^{-4}$

Bad



no guarantee

Unstable



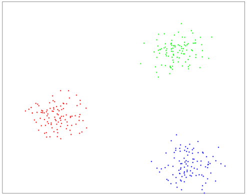
no guarantee

For the practitioner of clustering

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

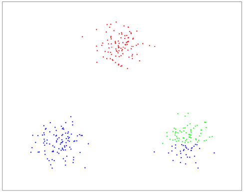
- IDEALLY WANTED: guarantee that $\mathcal{C}$ is correct/optimal
- WHAT WE CAN DO: guarantee that $\mathcal{C}$ is approximately correct/optimal
- WHEN $\mathcal{C}$ is good and stable

Good, stable



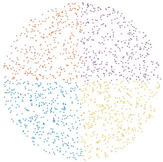
SS output: OI=$1e^{-4}$
OI = Optimality Interval

Bad



no guarantee

Unstable



no guarantee

Convex relaxations

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

$$L^* = \min_{\mathcal{C} \in \mathbf{C}_k} \text{Loss}(\mathcal{C}), \text{ with solution } \mathcal{C}^* \textit{Hard}! \tag{1}$$

Convex relaxation of problem (1).

- clustering $\mathcal{C} \to$ matrix $X(\mathcal{C}) \in \mathcal{X}$

  where $\mathcal{X}$ is convex set

  and $\text{Loss}(X)$ convex in X
- solve

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

Mapping a clustering to a matrix

$n = 5$, $\mathcal{C} = (1, 1, 1, 2, 2)$,

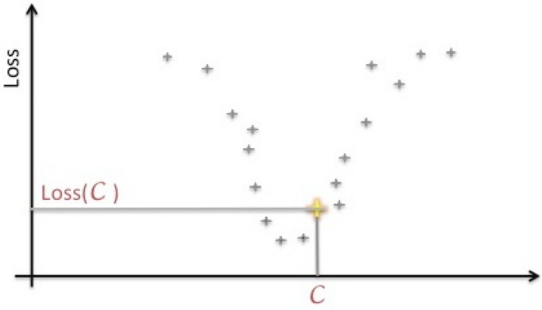
$$X(\mathcal{C}) = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

1. $X(\mathcal{C})$ is symmetric, positive definite, ≥ 0 elements
2. $X(\mathcal{C})$ has row sums equal to 1
3. $\operatorname{trace} X(\mathcal{C}) = K$

Let $\mathcal{X}$ be the space $n \times n$ of matrices with Properties 1, 2, 3 above

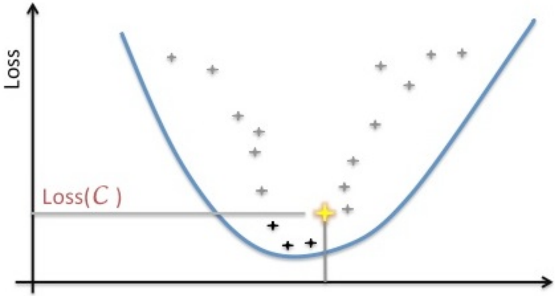
- $X(C)$ are extreme points of $\mathcal{X}$

The Sublevel Set (SS) method



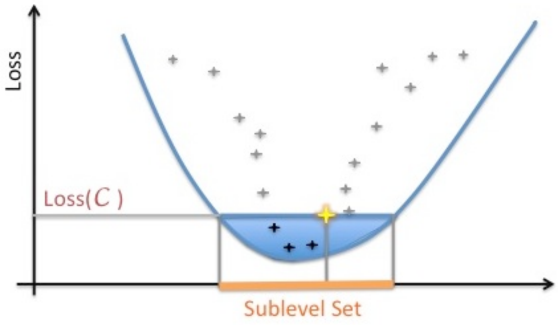
-

The Sublevel Set (SS) method



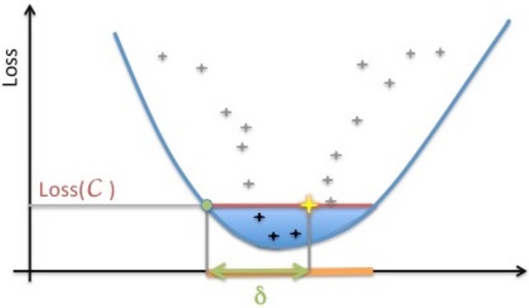
-

The Sublevel Set (SS) method



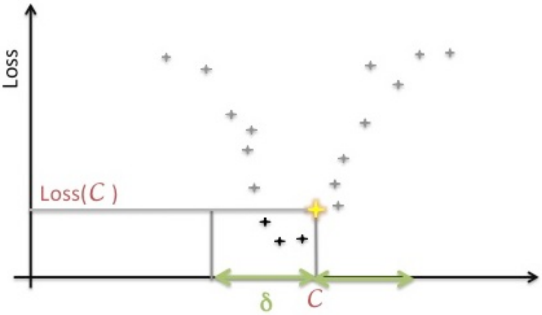
▶

The Sublevel Set (SS) method



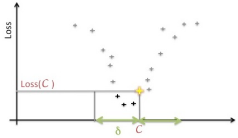
- a convex optimization problem

The Sublevel Set (SS) method



- δ is OI

The Sublevel Set (SS) method



δ is OI

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

Step 1 Define convex optimization problem

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

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 $\mathcal{C}$.

Two technical bits

1. SS is convex only if $||X' - X(\mathcal{C})||$ concave
   - Use $|| \; ||_F$ Frobenius norm. $||X(\mathcal{C})||_F^2 = K$ for any clustering.

Two technical bits

1. SS is convex only if $||X' - X(\mathcal{C})||$ concave
   - Use $||\ ||_F$ Frobenius norm. $||X(\mathcal{C})||_F^2 = K$ for any clustering.
2. Relating $\|\ \|_F$ to distance between clusterings.

$\|X(\mathcal{C}) - X(\mathcal{C})'\|_F^2 \leq \delta$ $\Rightarrow$ $d^{EM}(\mathcal{C}, \mathcal{C}') \leq \epsilon$

distance between matrices — "misclassification error" metric between clusterings

- Theorem proved in [M, Machine Learning Journal, 2012] with $\epsilon = 2\delta p_{\max}$.
- The tightest result known. Upper/lower bounds between $d^{EM}, \|\ \|_F$ and Rand
- Proofs use geometry of convex sets + refined analysis for small distances
- Example from [Wan,M NIPS16] OI by existing results [] OI by our method

K-means Sublevel Set problem

$$\text{Loss}(\mathcal{C}) \quad = \quad \langle D, X(\mathcal{C}) \rangle, \quad D = \text{squared distance matrix} \in \mathbb{R}^{n \times n}$$

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

a Semi-Definite Program (SDP).

Algorithm

Input Matrix of squared distances D, clustering $\mathcal{C}$

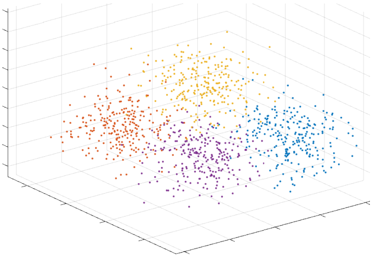
1. Solve SS_{Km}, get optimal value δ.
2. **If** $\epsilon = (K - \delta) p_{\max} \leq p_{\min}$ **then** $\mathcal{C}$ is stable

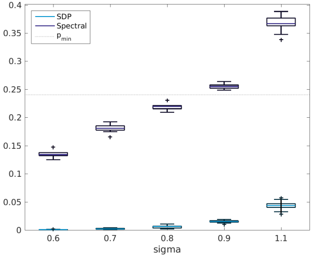
else no guarantee.

Results for K-means clusterings

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

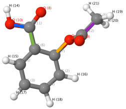
data for $\sigma = 0.9$ Values of ϵ vs cluster spread σ

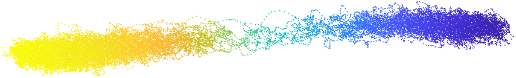




Spectral=[M ICML06], SDP=[M NeurIPS 2018]

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



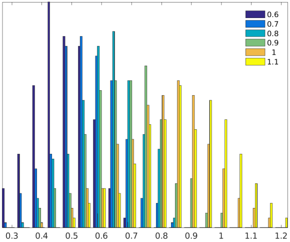


$K = 2$
$p_{\min}$=.26
$p_{\max}$=.74

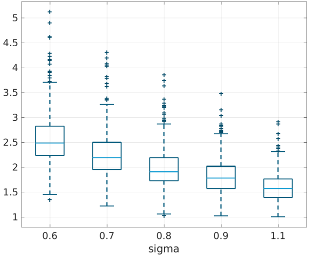
$n = 2118$ $\epsilon = 0.065$

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



For what clustering paradigms can we obtain OI's?

"All" 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
$\widetilde{\mathcal{X}}$	$\widetilde{X}(\mathcal{C})$	$\widetilde{X}_{ij} = 1$ 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 $X, \widetilde{X}, Z$, then

(1) There exists a convex SS problem

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

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

$$X : X_{ij} = 1/n_k \text{ iff } i, j \in C_k \qquad \epsilon = (K - \delta) p_{\max}$$

$$\widetilde{X} : \widetilde{X}_{ij} = 1 \text{ iff } i, j \in C_k \qquad \epsilon = \frac{\sum_{k \in [K]} n_k^2 + (n-K+1)^2 + (K-1) - 2\delta}{2 p_{\min}}$$

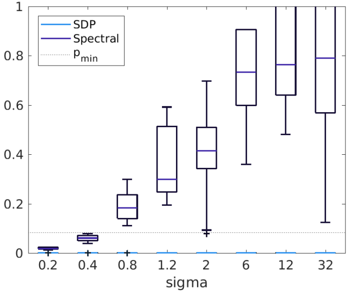
$$Z : Z_{ik} = 1/\sqrt{n_k} \text{ iff } i \in C_k \qquad \epsilon = (K - \delta^2/2) p_{\max}$$

Existence of guarantee depends only on space of convex relaxation.

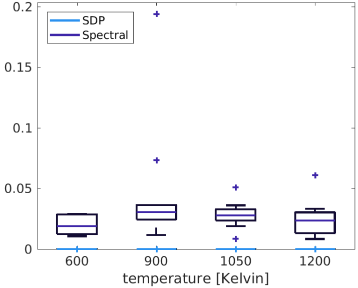
Results for Spectral Clustering by Normalized Cut

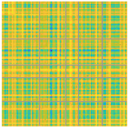
Spectral=[M AISTATS05], SDP=[M NeurIPS 2018]

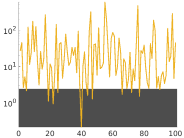
Synthetic S, $n = 100$

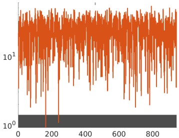


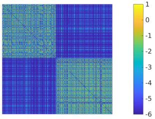
Chemical reaction data, $n \approx 1000$



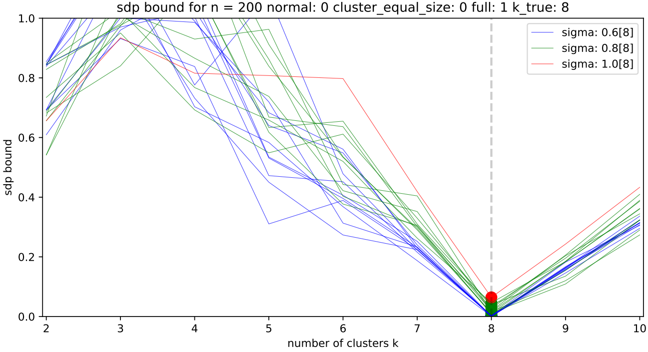








Stability and the selection of K [Cheng,M,Harchaoui (in preparation)]



Summary of SS method

1. Cluster data
2. Set up and solve SS problem
3. If solution ϵ small enough, guarantee $\mathcal{C}$ is approximately optimal and all other good clusterings are near it

- without any model assumptions, practically applicable
- not all $\mathcal{C}$ can have guarantees

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

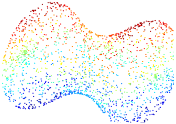
Manifold coordinates with physical meaning [M,Koelle,Zhang arXiv:1811.11891,. . .]

interpretability in the language of the problem

Brief intro to manifold learning algorithms

ALL ML Algorithms

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

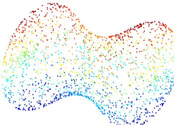


$p_1, \dots p_n \subset \mathbb{R}^D$

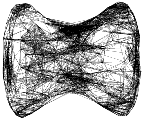
Brief intro to manifold learning algorithms

ALL ML Algorithms

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



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

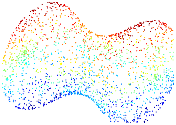


Brief intro to manifold learning algorithms

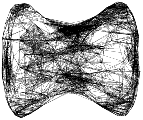
ALL ML Algorithms

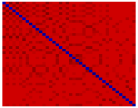
- **Input** Data $p_1, \ldots p_n$, embedding dimension m, neighborhood scale parameter ϵ
- Construct neighborhood graph p, p' neighbors iff $||p - p'||^2 \leq \epsilon$
- Construct a $n \times n$ sparse distance matrix

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



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





Brief intro to manifold learning algorithms

ALL ML Algorithms

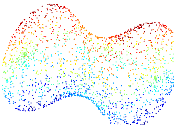
- **Input** Data $p_1, \ldots p_n$, embedding dimension m, neighborhood scale parameter ϵ
- Construct neighborhood graph p, p' neighbors iff $||p - p'||^2 \leq \epsilon$
- Construct a $n \times n$ sparse distance matrix

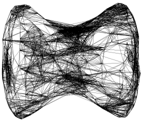
$$D = [||p - p'||]_{p,p' \text{neighbors}}$$

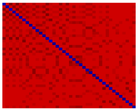
- Optional: construct kernel matrix, .e.g

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

and Laplacian matrix



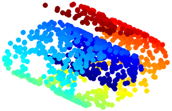




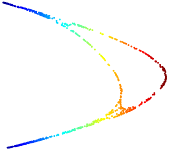
$p_1, \ldots p_n \subset \mathbb{R}^D$

Embedding in 2 dimensions by different manifold learning algorithms

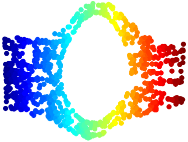
Original data
(Swiss Roll with hole)



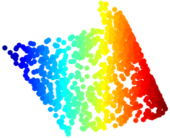
Laplacian Eigenmaps
(LE)



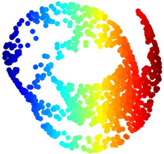
Isomap



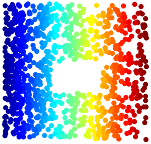
Hessian Eigenmaps (HE)



Local Linear Embedding
(LLE)



Local Tangent Space
Alignment (LTSA)



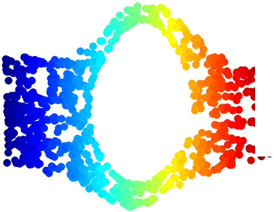
Preserving topology vs. preserving (intrinsic) geometry

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

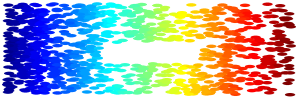
- Mapping $\mathcal{M} \longrightarrow \phi(\mathcal{M})$ is diffeomorphism
  preserves topology
  often satisfied by embedding algorithms
- Mapping ϕ preserves
  - distances along curves in $\mathcal{M}$
  - angles between curves in $\mathcal{M}$
  - areas, volumes

  ...i.e. ϕ is isometry
  For most algorithms, in most cases, ϕ is not isometry

Preserves topology



Preserves topology + intrinsic geometry



Our approach: Metric Manifold Learning

[Perrault-Joncas,M 10]

Given

- mapping ϕ that preserves topology

  true in many cases

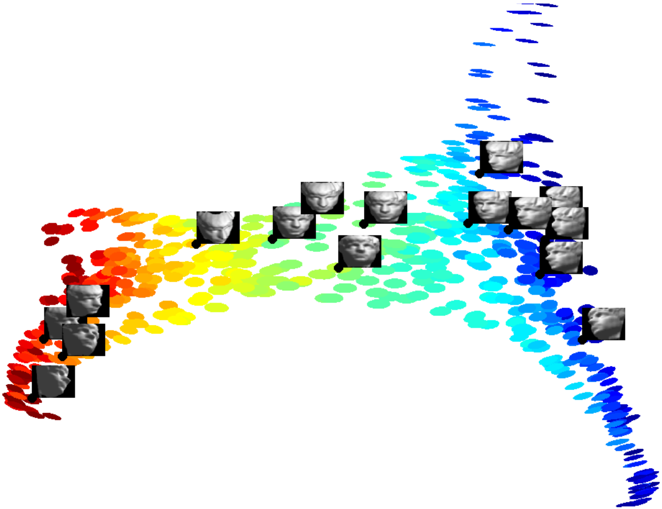
Objective

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

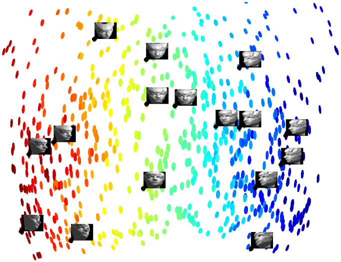
g is the Riemannian metric.

g for Sculpture Faces

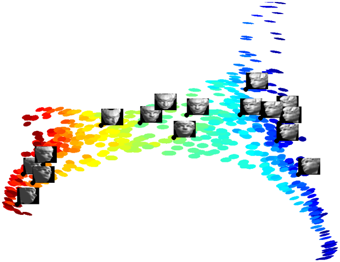
- $n = 698$ gray images of faces in $D = 64 \times 64$ dimensions
  - head moves up/down and right/left



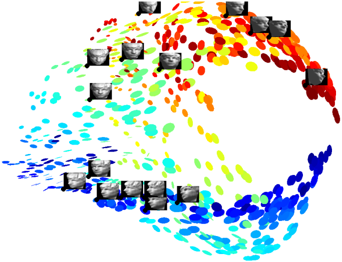
LTSA Algoritm



Isomap



LTSA



Laplacian Eigenmaps

Relation between g and Δ

- $\Delta =$ 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

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

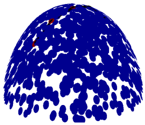
where $h = g^{-1}$ (matrix inverse) and $k, l = 1, 2, \ldots m$ are embedding dimensions

Intuition:

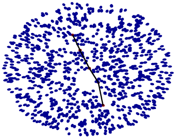
- at each point $p \in \mathcal{M}$, $G(p)$ is a $d \times d$ matrix
- apply Δ to embedding coordinate functions $\phi_1, \ldots \phi_m$
- this produces $G^{-1}(p)$ in the given coordinates
- our algorithm implements matrix version of this operator result
- consistent estimation of Δ is well studied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

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

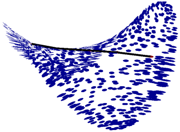
Original



Isomap



Laplacian Eigenmaps

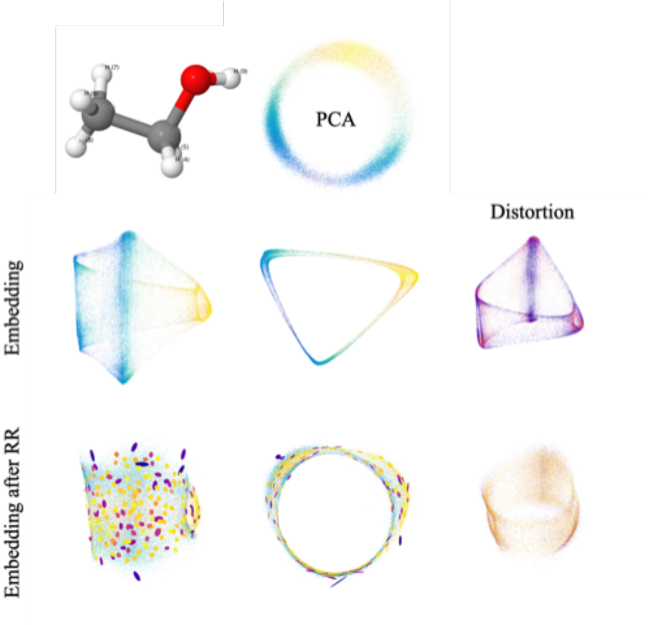


true distance $d = 1.57$

Embedding	$\|\|f(p) - f(p')\|\|$	Shortest Path d	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

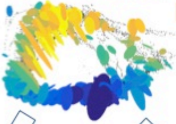
Why useful

- 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

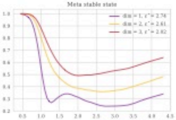
Applications

- 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]

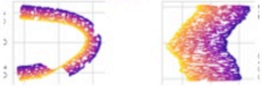
Estimate
Riemannian metric



Optimize
neighborhood size
[NIPS 2016]



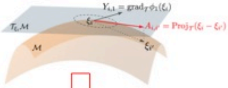
Choose independent e-vectors
[NeurIPS 2019]



Distances,
angles, areas
preserved

Riemannian relaxation
[NIPS 2015]

Vector fields
preserved



Coordinates with physical meaning

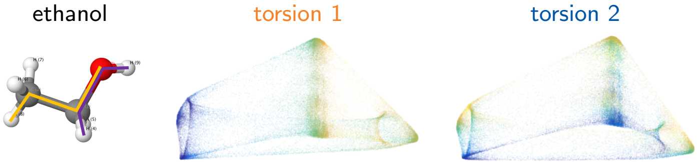
Outline

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



- 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

- 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 $\mathcal{F} = \{f_1, \ldots f_p$, with $f_j : \mathbb{R}^D \to \mathbb{R}\}$.
  - e.g. all torsions in ethanol

Problem formulation

- 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 $\mathcal{F} = \{f_1, \ldots f_p$, with $f_j : \mathbb{R}^D \to \mathbb{R}\}$.
  - e.g. all torsions in ethanol

Problem formulation

- 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 $\mathcal{F} = \{f_1, \ldots f_p$, with $f_j : \mathbb{R}^D \to \mathbb{R}\}$.
  - e.g. all torsions in ethanol
- Goal to express the embedding coordinate functions $\phi_1 \ldots \phi_m$ in terms of functions in $\mathcal{F}$.

  More precisely, we assume that

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

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

Challenges

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

- Framework: sparse regression
- 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

$$\text{e.g.} \quad \begin{array}{l} \phi_1 = f_1/\sqrt{f_2}, \\ \phi_2 = f_1 \sin(f_3^2) \end{array} \quad \text{or} \quad \begin{array}{l} \phi_1 = \sin(\tau_1) \\ \phi_2 = \cos(\tau_1)(\text{ethanol}) \\ \phi_3 = \sin(\tau_2) \end{array}$$

Challenges

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

- Framework: sparse regression
- 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

$$\text{e.g.} \quad \begin{array}{l} \phi_1 = f_1/\sqrt{f_2}, \\ \phi_2 = f_1 \sin(f_3^2) \end{array} \quad \text{or} \quad \begin{array}{l} \phi_1 = \sin(\tau_1) \\ \phi_2 = \cos(\tau_1)(\text{ethanol}) \\ \phi_3 = \sin(\tau_2) \end{array}$$

- we do not assume ϕ isometric
- what requirements on dictionary functions $f_{1:p}$ for unique recovery?

First Idea: from non-linear to linear

- If

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

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

- then

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

  - sparse linear recovery

First Idea: from non-linear to linear

- If

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

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

  $\boxed{D\phi = DhDf}$

  - 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

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

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

Theory

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

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

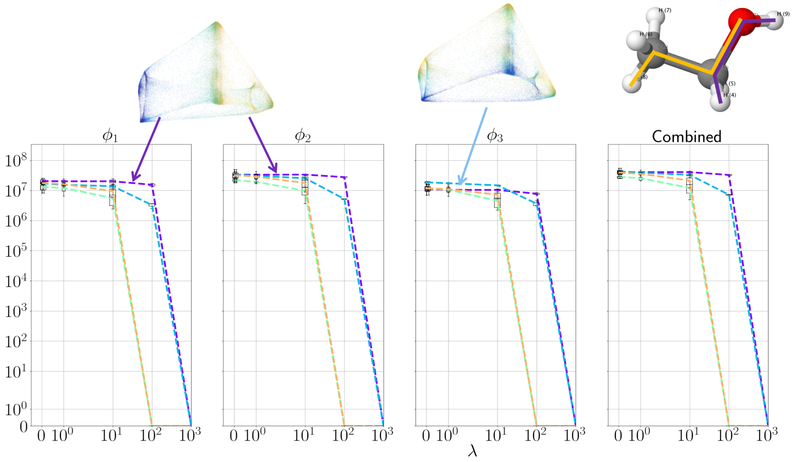
$$\text{rank}\begin{pmatrix} Dg_S \\ Dg_{S'} \end{pmatrix} = \text{rank}\, Dg_{S'} \quad \text{on } U$$

- When can FLasso recover S ?
  Incoherence conditions

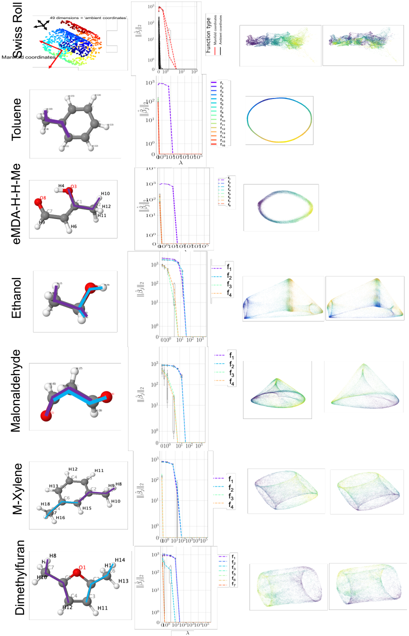
$$\mu = \max_{i=1:n, j\in S, j'\notin S} |X_{ji}^T X_{j'i}| \quad \nu = \frac{1}{\min_{i=1:n} ||X_{iS}^T X_{iS}||_2} \quad nd\sigma^2 = \sum_{i,k} \epsilon_{ik}^2$$

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

Ethanol MD simulation

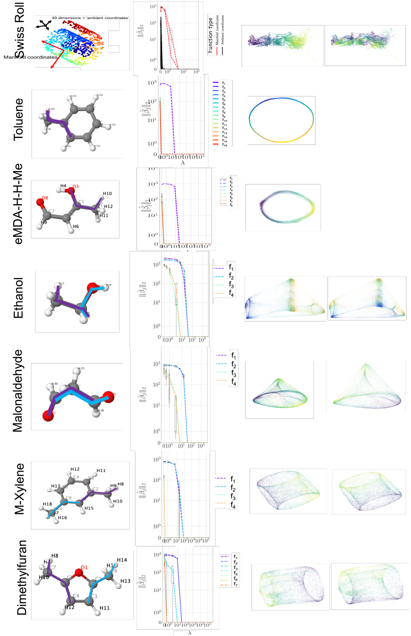


Summary of ManifoldLasso/FunctionalLasso



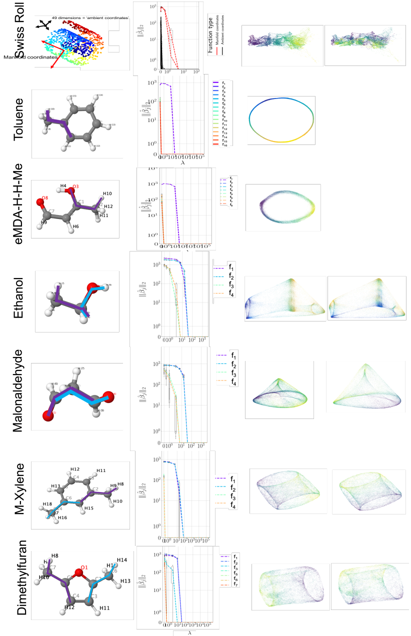
- Regress non-linearly functions $\phi_{1:m}$ on $\mathcal{F} = \{f_{1:p}\}$

Summary of ManifoldLasso/FunctionalLasso



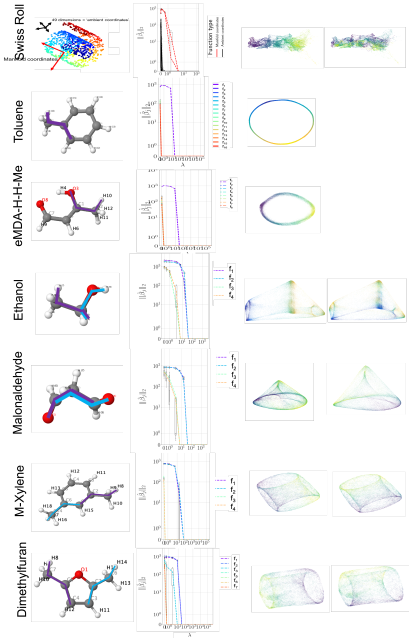
- Regress non-linearly functions $\phi_{1:m}$ on $\mathcal{F} = \{f_{1:p}\}$

Summary of ManifoldLasso/FunctionalLasso



- Regress non-linearly functions $\phi_{1:m}$ on $\mathcal{F} = \{f_{1:p}\}$
- explain learned coordinates by dictionaries of domain-relevant functions
- sparse functional regression
- rank of feature set, of neural net embedding
- set of f's that covary (e.g. protein folding), level sets (in progress)

Summary of ManifoldLasso/FunctionalLasso



- Regress non-linearly functions $\phi_{1:m}$ on $\mathcal{F} = \{f_{1:p}\}$
- explain learned coordinates by dictionaries of domain-relevant functions
- sparse functional regression
- rank of feature set, of neural net embedding
- set of f's that covary (e.g. protein folding), level sets (in progress)
- Method to push/pull vectors through mappings ϕ

$\mathcal{M}$

Summary: Towards knowledge that is transferable

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: Towards knowledge that is transferable

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: Towards knowledge that is transferable

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 artifacts

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 artifacts
- 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 artifacts
- 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 artifacts
- 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 artifacts
- 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
- Interpretability – in the language of the domain

Sam Koelle, Yu-Chia Chen, 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