Noise in Data: A geometric perspective

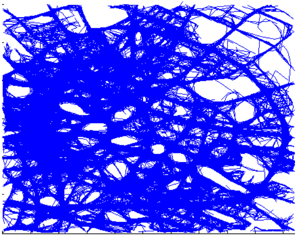
Yusu Wang

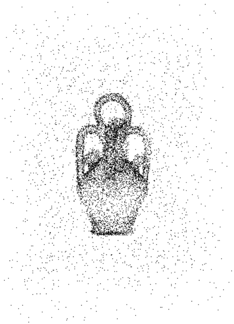
Compuer Science Dept.,
The Ohio State University

NSF-CBMS Conf. 2016

Introduction

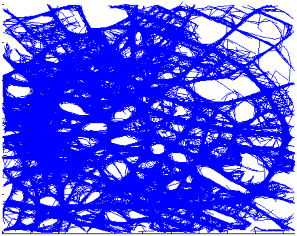
- Noise in data prevalent in various applications

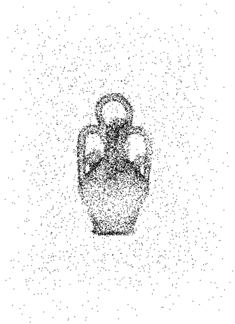




Introduction

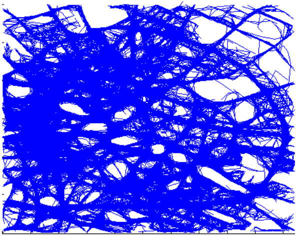
- Noise in data prevalent in various applications
- Noise present in diverse forms

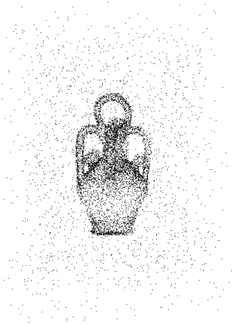




Introduction

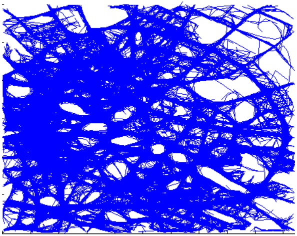
- Noise in data prevalent in various applications
- Noise present in diverse forms
- Effective handling of noise depends on how they are generated and what the target uses of data are

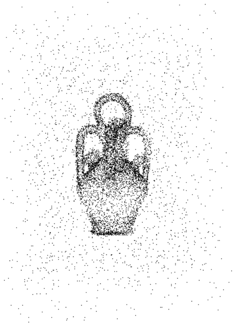




Introduction

- Noise in data prevalent in various applications
- Noise present in diverse forms
- Effective handling of noise depends on how they are generated and what the target uses of data are
- This talk:
  - Focus on noise in metric of input data



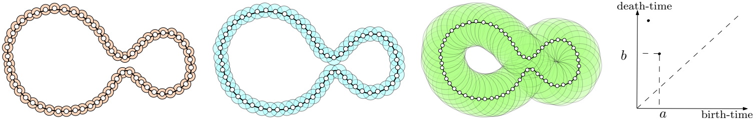


Motivation

- Many geometric / topological data analysis algorithms often assume that the input is a (discrete) metric space.
- What are natural ways to model noise in input metric, and how to process such noise with theoretical guarantees.

Motivation

- Many geometric / topological data analysis algorithms often assume that the input is a (discrete) metric space.
- What are natural ways to model noise in input metric, and how to process such noise with theoretical guarantees.



Talk Outline

In this talk, we consider three different settings to explore:

What are natural ways to model noise in input metric, and how to process such noise effeciently with theoretical guarantees.

- Quest 1: towards parameter-free denoising for embedded point cloud data (PCD)
- Quest 2: metric embedding with outliers
- Quest 3: recovering shortest path metrics from perturbed graphs

Talk Outline

In this talk, we consider three different settings to explore:

What are natural ways to model noise in input metric, and how to process such noise effeciently with theoretical guarantees.

- Quest 1: towards parameter-free denoising for embedded point cloud data (PCD)
- Quest 2: metric embedding with outliers
- Quest 3: recovering shortest path metrics from perturbed graphs

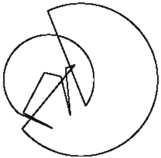
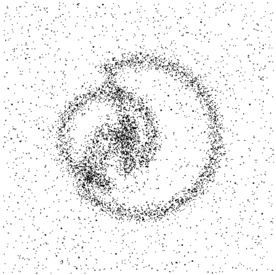
First Quest

Towards parameter-free denoising for PCD via decluttering and resampling

- *Joint work with Mickaël Buchet, Tamal Dey and Jiayuan Wang*

Problem Setup

Input: A set of points P embedded in a metric space, which is a *"noisy"* sample of a hidden ground truth
Output: A *"denoised"* set of points $Q \subset P$

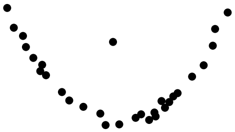




Some Existing Denoising Approaches

- Thresholding

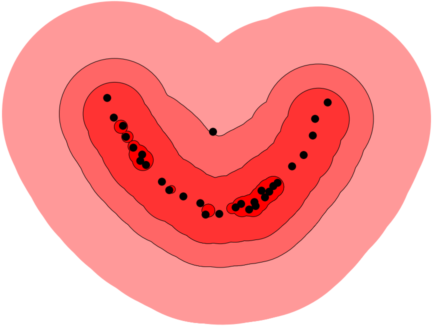
Some Existing Denoising Approaches

- Thresholding



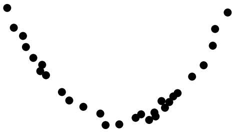
Some Existing Denoising Approaches

- Thresholding



Some Existing Denoising Approaches

- Thresholding

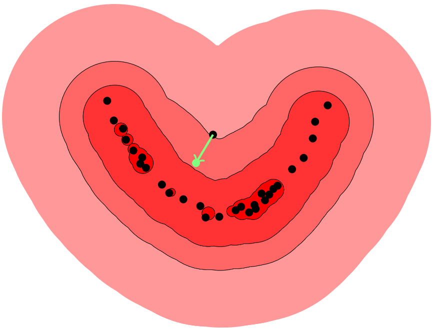


Some Existing Denoising Approaches

- Thresholding
  - choice of a density estimator, which involves parameter(s)
  - choice of a threshold

Some Existing Denoising Approaches

- Thresholding
  - choice of a density estimator, which involves parameter(s)
  - choice of a threshold
- Mean-shift type



Some Existing Denoising Approaches

- Thresholding
  - choice of a density estimator, which involves parameter(s)
  - choice of a threshold
- Mean-shift type
  - needs additional parameters: such as step size, stopping criteria.

Some Existing Denoising Approaches

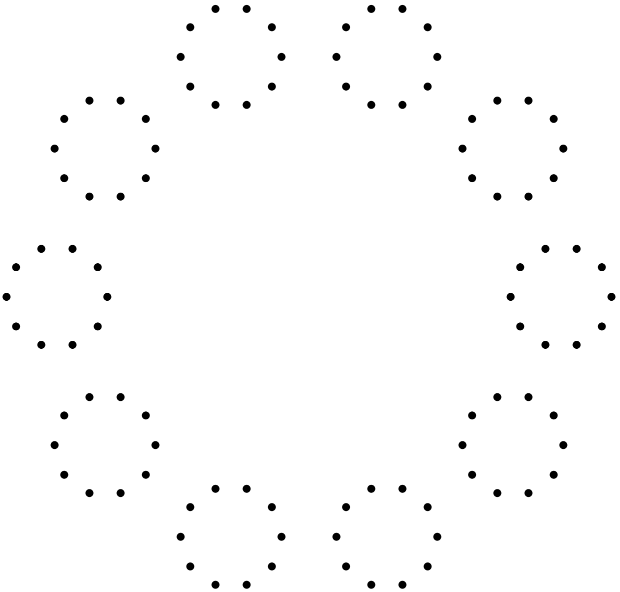
- Thresholding
  - choice of a density estimator, which involves parameter(s)
  - choice of a threshold
- Mean-shift type
  - needs additional parameters: such as step size, stopping criteria.
- "Deconvolution"
  - assuming the noise generative model, aiming to cancel it out.
  - requires a knowledge of the generative model
  - typically asymptotic guarantees

Some Existing Denoising Approaches

- Thresholding
  - choice of a density estimator, which involves parameter(s)
  - choice of a threshold
- Mean-shift type
  - needs additional parameters: such as step size, stopping criteria.
- "Deconvolution"
  - assuming the noise generative model, aiming to cancel it out.
  - requires a knowledge of the generative model
  - typically asymptotic guarantees

Require parameters and / or assumptions of noise models.

Parameters perhaps unavoidable



Goal of Quest-1

Minimize the use of parameter in denoising embedded PCD data, yet still provide theoretical guarantees and understanding

Goal of Quest-1

Minimize the use of parameter in denoising embedded PCD data, yet still provide theoretical guarantees and understanding

- Decluttering algorithm (works for any input, use one parameter)

Goal of Quest-1

Minimize the use of parameter in denoising embedded PCD data, yet still provide theoretical guarantees and understanding

- Decluttering algorithm (works for any input, use one parameter)

Parameter-free? Require stronger assumptions on noise model

Goal of Quest-1

Minimize the use of parameter in denoising embedded PCD data, yet still provide theoretical guarantees and understanding

- Decluttering algorithm (works for any input, use one parameter)

Parameter-free? Require stronger assumptions on noise model

- Declutter+Resample algorithm

Goal of Quest-1

Minimize the use of parameter in denoising embedded PCD data, yet still provide theoretical guarantees and understanding

- Decluttering algorithm (works for any input, use one parameter)

Parameter-free? Require stronger assumptions on noise model

- Declutter+Resample algorithm

k-distance

Definition (CCM'11)

Given a point set P from a metric space $(\mathcal{X}, d_{\mathcal{X}})$, and an integer k, the k-distance is defined by:

$$d_{P,k}(x) = \sqrt{\frac{1}{k}\sum_{i=1}^{k} d_{\mathcal{X}}(x, p_i(x))^2}$$

where $p_i(x)$ is the i^{th} nearest neighbor of x.

k-distance

$$\boxed{d_{P,k}(x) = \sqrt{\frac{1}{k}\sum_{i=1}^{k} d_{\mathcal{X}}(x, p_i(x))^2}}$$

- Intuitively, k-distance (average distance to k nearest neighbors) can be considered as inverse to a density estimator
  - *[Biau, Chazal, Cohen-Steiner, Devroye and Rodrigues, 2011]*

k-distance

$$\boxed{d_{P,k}(x) = \sqrt{\frac{1}{k}\sum_{i=1}^{k} d_{\mathcal{X}}(x, p_i(x))^2}}$$

- Intuitively, k-distance (average distance to k nearest neighbors) can be considered as inverse to a density estimator
  - *[Biau, Chazal, Cohen-Steiner, Devroye and Rodrigues, 2011]*
- Parameter k specifies level of noise

k-distance

$$\boxed{d_{P,k}(x) = \sqrt{\frac{1}{k}\sum_{i=1}^{k} d_{\mathcal{X}}(x, p_i(x))^2}}$$

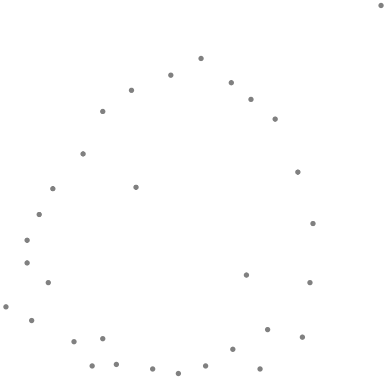
- Intuitively, k-distance (average distance to k nearest neighbors) can be considered as inverse to a density estimator
  - *[Biau, Chazal, Cohen-Steiner, Devroye and Rodrigues, 2011]*
- Parameter k specifies level of noise
- For any $p \in P$, we can consider $r_p = 2d_{P,k}(p)$ as the radius of uncertainty for point p.

Algorithm Declutter

Input: A set of points P in a metric space
Output: A *denoised and sparsified* set of points $Q \subset P$

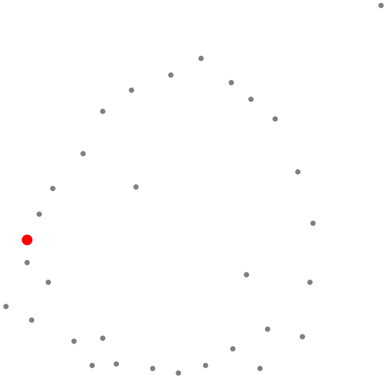
1. $Q_0 = \emptyset$
2. Sort P according to increasing k-distance.
3. For i from 1 to $n = |P|$, if $B(p_i, 2d_{P,k}(p_i)) \cap Q_{i-1} = \emptyset$:
    - then $Q_i = Q_{i-1} \cup p_i$
    - else $Q_i = Q_{i-1}$.

Example



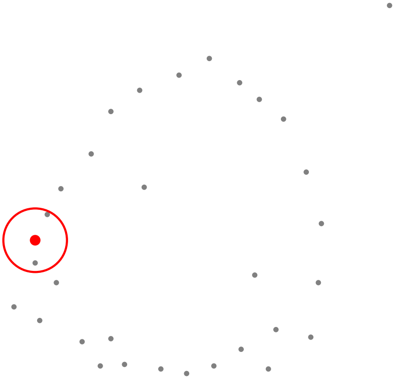
- Process points in increasing order of their k-distance (intuitively, in descreasing density).

Example



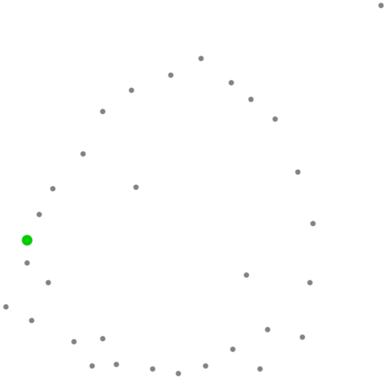
- Process points in increasing order of their k-distance (intuitively, in descreasing density).

Example



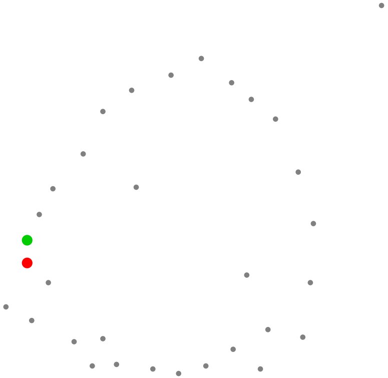
- Process points in increasing order of their k-distance (intuitively, in descreasing density).

Example



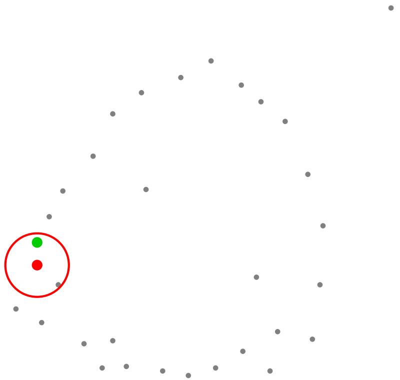
- Process points in increasing order of their k-distance (intuitively, in descreasing density).

Example



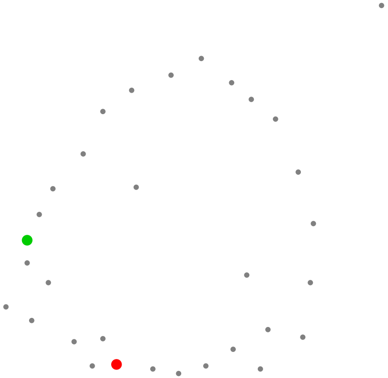
- Process points in increasing order of their k-distance (intuitively, in descreasing density).

Example



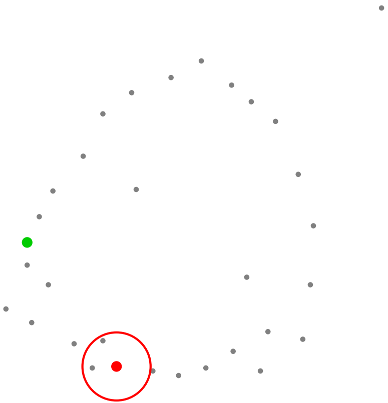
- Process points in increasing order of their k-distance (intuitively, in descreasing density).

Example



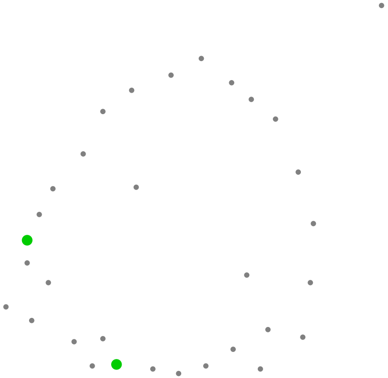
- Process points in increasing order of their k-distance (intuitively, in descreasing density).

Example



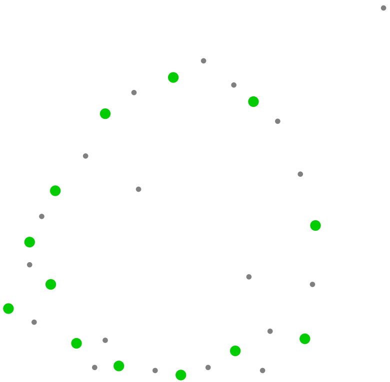
- Process points in increasing order of their k-distance (intuitively, in descreasing density).

Example



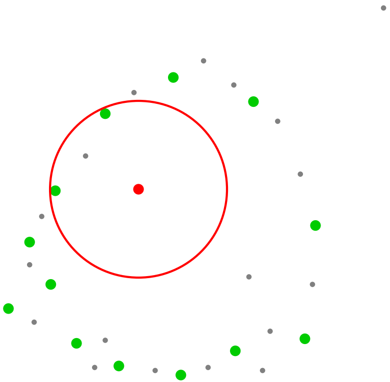
- Process points in increasing order of their k-distance (intuitively, in descreasing density).

Example



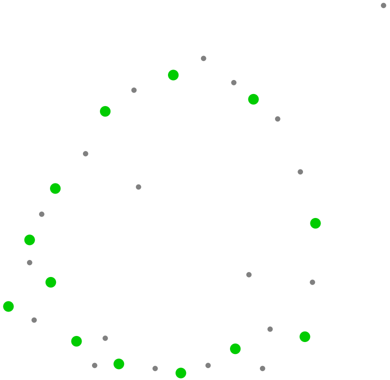
- Process points in increasing order of their k-distance (intuitively, in descreasing density).

Example



- Process points in increasing order of their k-distance (intuitively, in descreasing density).

Example

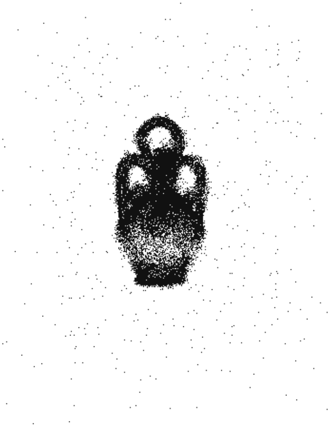


- Process points in increasing order of their k-distance (intuitively, in descreasing density).

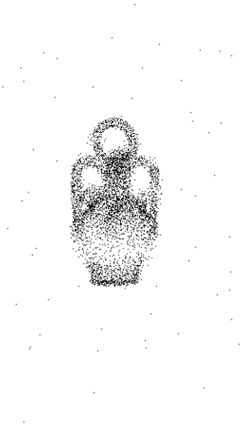
Illustration I

Input $\qquad$ $k = 9$ $\qquad$ $k = 30$

Illustration II



Input



$k = 4$

$k = 47$

Algorithm Declutter

Input: A set of points P in a metric space
Output: A denoised and sparsified set of points $Q \subset P$

1. $Q_0 = \emptyset$
2. Sort P according to increasing k-distance.
3. For i from 1 to $n = |P|$, if $B(p_i, 2d_{P,k}(p_i)) \cap Q_{i-1} = \emptyset$:
    - then $Q_i = Q_{i-1} \cup p_i$
    - else $Q_i = Q_{i-1}$.

Properties:

- Requires only one parameter
- Output is also sparsified (good? bad?)
- Have theoretical guarantee (shortly)

Algorithm Declutter

Input: A set of points P in a metric space
Output: A denoised and sparsified set of points $Q \subset P$

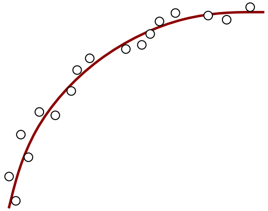
1. $Q_0 = \emptyset$
2. Sort P according to increasing k-distance.
3. For i from 1 to $n = |P|$, if $B(p_i, 2d_{P,k}(p_i)) \cap Q_{i-1} = \emptyset$:
   - then $Q_i = Q_{i-1} \cup p_i$
   - else $Q_i = Q_{i-1}$.

Properties:

- Requires only one parameter
- Output is also sparsified (good? bad?)
- Have theoretical guarantee (shortly)
  - sampling conditions

Sampling conditions

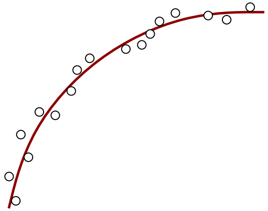
We assume that we have a point cloud P describing an underlying compact set K in a metric space $\mathcal{X}$ and we choose an integer k.



Sampling conditions

We assume that we have a point cloud P describing an underlying compact set K in a metric space $\mathcal{X}$ and we choose an integer k.

P is an ϵ_k noisy sample of K if:

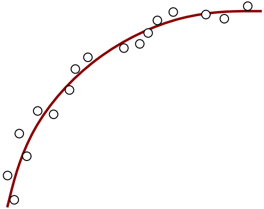


Sampling conditions

We assume that we have a point cloud P describing an underlying compact set K in a metric space $\mathcal{X}$ and we choose an integer k.

P is an ϵ_k noisy sample of K if:

1. $\forall x \in K,\ d_{P,k}(x) \leq \epsilon_k$.

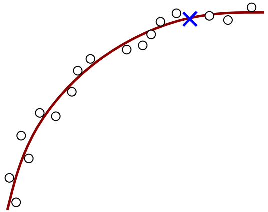


Sampling conditions

We assume that we have a point cloud P describing an underlying compact set K in a metric space $\mathcal{X}$ and we choose an integer k.

P is an ϵ_k noisy sample of K if:

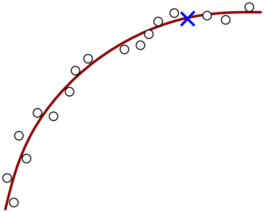
1. $\forall x \in K,\ d_{P,k}(x) \leq \epsilon_k$.



Sampling conditions

We assume that we have a point cloud P describing an underlying compact set K in a metric space $\mathcal{X}$ and we choose an integer k.

P is an ϵ_k noisy sample of K if:



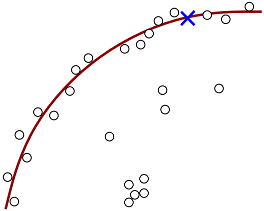
1. $\forall x \in K,\ d_{P,k}(x) \leq \epsilon_k$.
2. $\forall y \in \mathcal{X},\ d_K(y) \leq d_{P,k}(y) + \epsilon_k$.

Sampling conditions

We assume that we have a point cloud P describing an underlying compact set K in a metric space $\mathcal{X}$ and we choose an integer k.

P is an ϵ_k noisy sample of K if:

1. $\forall x \in K,\ d_{P,k}(x) \leq \epsilon_k.$
2. $\forall y \in \mathcal{X},\ d_K(y) \leq d_{P,k}(y) + \epsilon_k.$

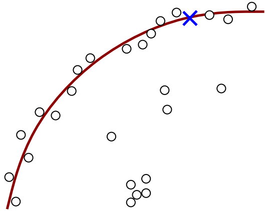


Sampling conditions

We assume that we have a point cloud P describing an underlying compact set K in a metric space $\mathcal{X}$ and we choose an integer k.

P is an ϵ_k noisy sample of K if:

1. $\forall x \in K,\ d_{P,k}(x) \leq \epsilon_k$.
2. $\forall y \in \mathcal{X},\ d_K(y) \leq d_{P,k}(y) + \epsilon_k$.



Very general, can be used to model classical noise models such as Hausdorff, Gaussian noise etc.

- *[Buchet, Topological Inference from Measures, PhD Thesis 2014]*

Theoretical guarantees for decluttering

Theorem

Given a point set P which is an ϵ_k noisy sample of a compact K, Algorithm Declutter *returns a set Q such that*

$$d_H(K, Q) \leq 7\epsilon_k.$$

Adaptivity

The conditions can also be expressed for adaptive samples. Let f be a feature size function (i.e. 1-Lipschitz non-negative function on K) and $\bar{p}$ the projection of p onto K.

Adaptivity

The conditions can also be expressed for adaptive samples. Let f be a feature size function (i.e. 1-Lipschitz non-negative function on K) and $\bar{p}$ the projection of p onto K.

P is an ϵ_k adaptive noisy sample of K if:

Adaptivity

The conditions can also be expressed for adaptive samples. Let f be a feature size function (i.e. 1-Lipschitz non-negative function on K) and $\bar{p}$ the projection of p onto K.

P is an ϵ_k adaptive noisy sample of K if:

1. $\forall x \in K,\ d_{P,k}(x) \leq \epsilon_k f(x)$.

Adaptivity

The conditions can also be expressed for adaptive samples. Let f be a feature size function (i.e. 1-Lipschitz non-negative function on K) and $\bar{p}$ the projection of p onto K.

P is an ϵ_k adaptive noisy sample of K if:

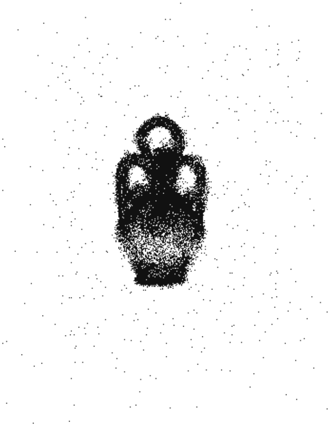
1. $\forall x \in K,\ d_{P,k}(x) \leq \epsilon_k f(x)$.
2. $\forall y \in \mathcal{X},\ d_K(y) \leq d_{P,k}(y) + \epsilon_k f(\bar{y})$.

Adaptive guarantees

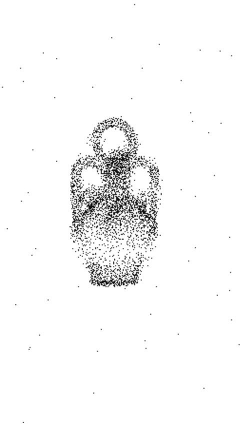
Theorem

Given an input point set P which is an ϵ_k adaptive noisy sample of a compact K with $\epsilon_k \leq \frac{1}{2}$, Algorithm Declutter *returns a sample Q of K where $\delta_H^f(Q, K) \leq 7\epsilon_k$.*

Illustration II



Input



$k = 4$

$k = 47$

Declutter Algorithm

Pros:

- Requires only one parameter
- Output is also sparsified (good? bad?)
- Provide theoretical guarantee

Declutter Algorithm

Pros:

- Requires only one parameter
- Output is also sparsified (good? bad?)
- Provide theoretical guarantee

Cons:

- Choice of the parameter k.
- Absence of a correct k.
- Sparsifying effect too pronounced.

Goal of Quest-1

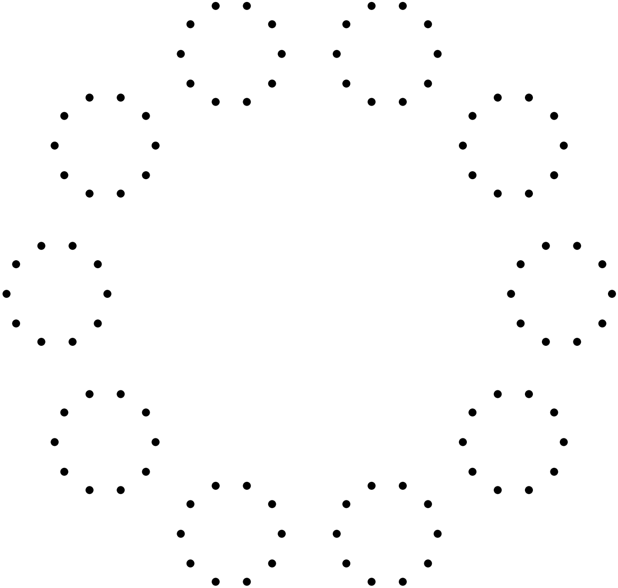
Minimize the use of parameter in denoising embedded PCD data, yet still provide theoretical guarantees and understanding

- Decluttering algorithm (works for any input, use one parameter)

Parameter-free? Require stronger assumptions on noise model

- Declutter+Resample algorithm

Need for one parameter



How to avoid parameter?

The parameter-free algorithm will assume a stronger sampling condition on input point samples.

How to avoid parameter?

The parameter-free algorithm will assume a stronger sampling condition on input point samples.

P is an c-uniform ϵ_k noisy sample of K if:

1. $\forall x \in K,\ d_{P,k}(x) \leq \epsilon_k$.
2. $\forall y \in \mathcal{X},\ d_K(y) \leq d_{P,k}(y) + \epsilon_k$.
3. $\forall p \in P,\ d_{P,k}(p) \geq \frac{\epsilon_k}{c}$.

Idea 1: Declutter+Resample

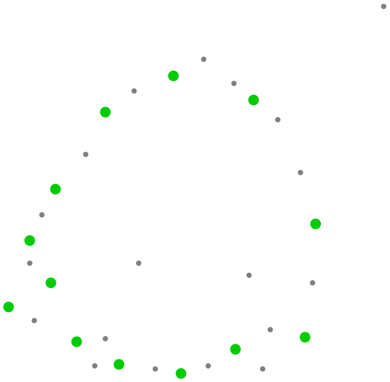
First idea:

- To address over-sparsity, bring back neighboring points of sub-samples Q!

Idea 1: Declutter+Resample

First idea:

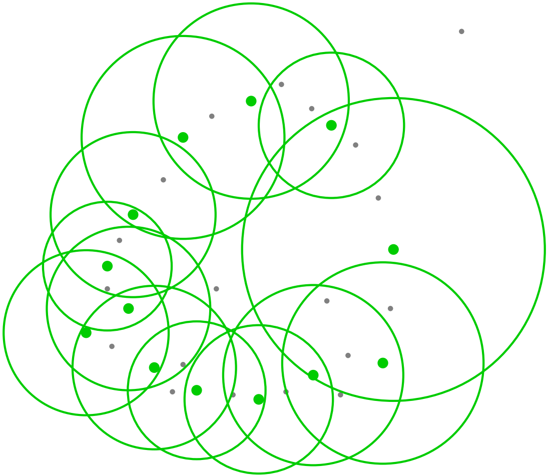
- To address over-sparsity, bring back neighboring points of sub-samples Q!



Idea 1: Declutter+Resample

First idea:

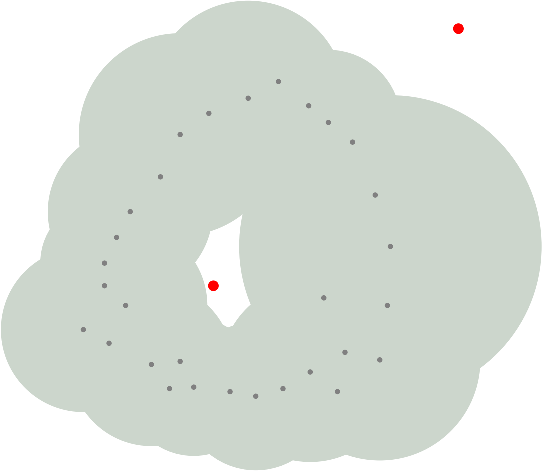
- To address over-sparsity, bring back neighboring points of sub-samples Q!



Idea 1: Declutter+Resample

First idea:

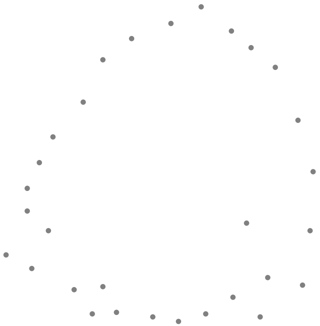
- To address over-sparsity, bring back neighboring points of sub-samples Q!



Idea 1: Declutter+Resample

First idea:

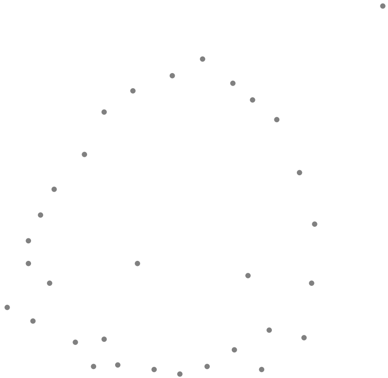
- To address over-sparsity, bring back neighboring points of sub-samples Q!



Idea 2: Gradually descrease parameter k

Second idea:

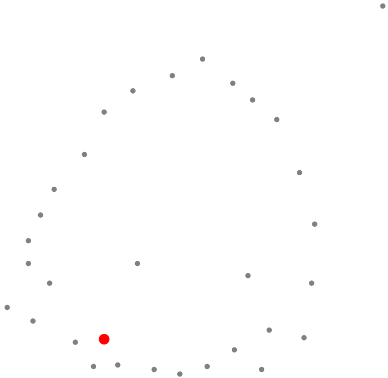
- To avoid choosing a parameter k: starting with $k = n$, gradually decrease k till $k = 1$.



Idea 2: Gradually descrease parameter k

Second idea:

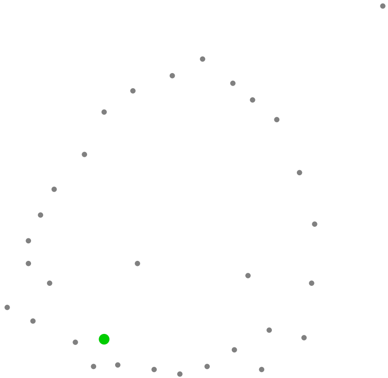
- To avoid choosing a parameter k: starting with $k = n$, gradually decrease k till $k = 1$.



Idea 2: Gradually descrease parameter k

Second idea:

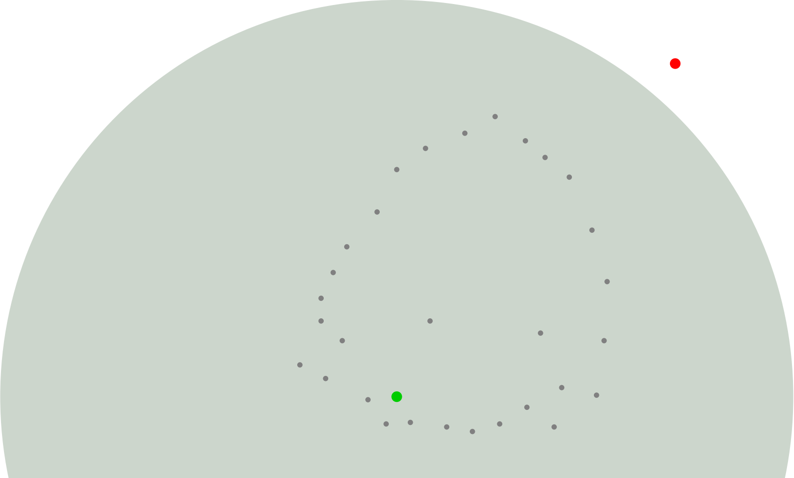
- To avoid choosing a parameter k: starting with $k = n$, gradually decrease k till $k = 1$.



Idea 2: Gradually descrease parameter k

Second idea:

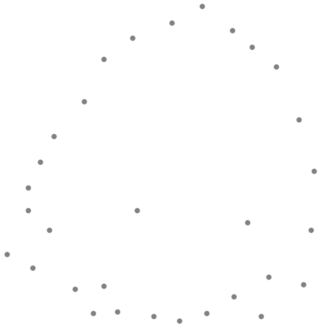
- To avoid choosing a parameter k: starting with $k = n$, gradually decrease k till $k = 1$.



Idea 2: Gradually descrease parameter k

Second idea:

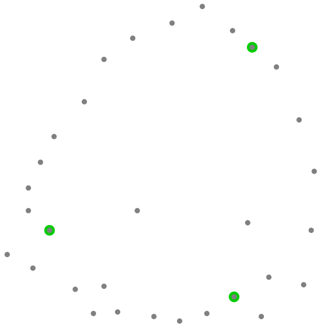
- To avoid choosing a parameter k: starting with $k = n$, gradually decrease k till $k = 1$.



Idea 2: Gradually descrease parameter k

Second idea:

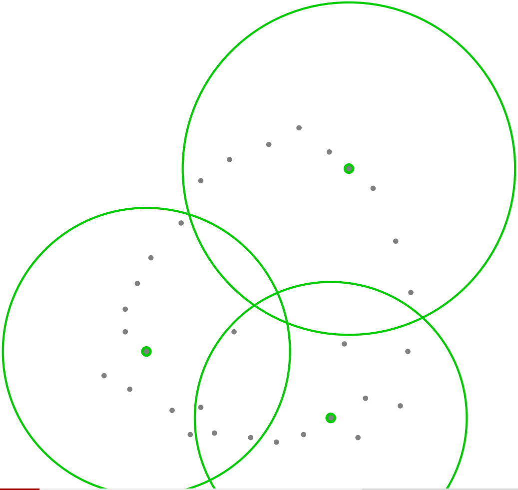
- To avoid choosing a parameter k: starting with $k = n$, gradually decrease k till $k = 1$.



Idea 2: Gradually descrease parameter k

Second idea:

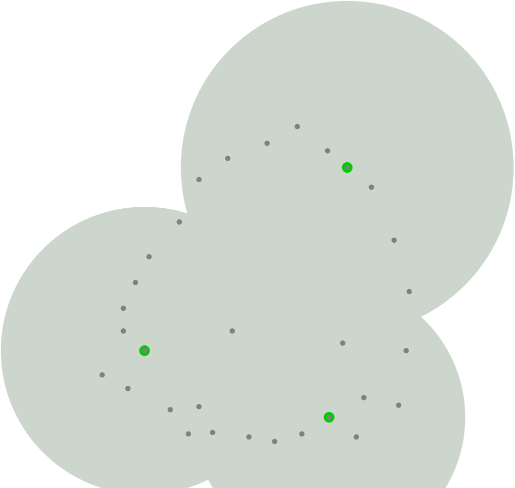
- To avoid choosing a parameter k: starting with $k = n$, gradually decrease k till $k = 1$.



Idea 2: Gradually descrease parameter k

Second idea:

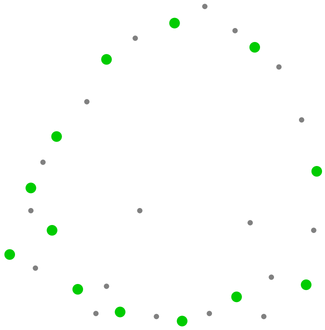
- To avoid choosing a parameter k: starting with $k = n$, gradually decrease k till $k = 1$.



Idea 2: Gradually descrease parameter k

Second idea:

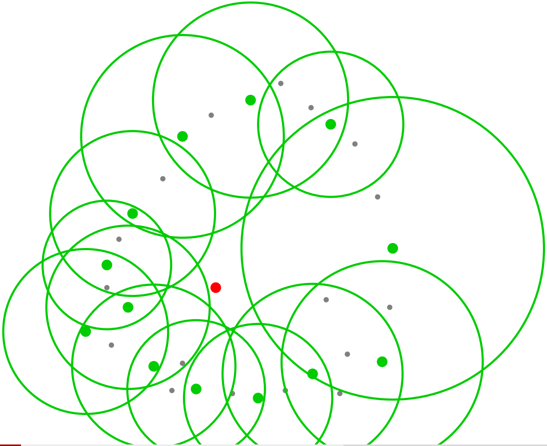
- To avoid choosing a parameter k: starting with $k = n$, gradually decrease k till $k = 1$.



Idea 2: Gradually descrease parameter k

Second idea:

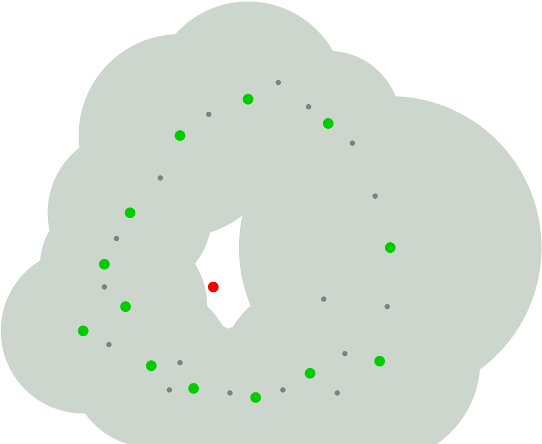
- To avoid choosing a parameter k: starting with $k = n$, gradually decrease k till $k = 1$.



Idea 2: Gradually descrease parameter k

Second idea:

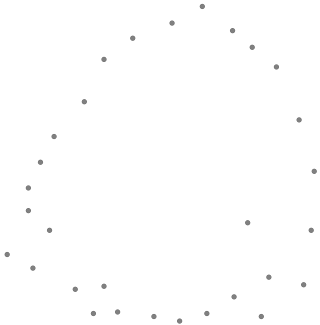
- To avoid choosing a parameter k: starting with $k = n$, gradually decrease k till $k = 1$.



Idea 2: Gradually descrease parameter k

Second idea:

- To avoid choosing a parameter k: starting with $k = n$, gradually decrease k till $k = 1$.



Algorithm ParfreeDeclutter

1. $i_M \leftarrow \log_2(|P|)$
2. $P_{i_M} \leftarrow P$
3. For $i = i_M$ to 1
   - $Q_i \leftarrow \text{Declutter}(P_i, 2^i)$
   - $P_{i-1} \leftarrow \cup_{q \in Q_i} B(q, 4d_{P_i, 2^i}(q))$
4. Return P_0

Theoretical guarantees

Theorem

Given a point set P and i_0 such that for all $i > i_0$, P is a weak uniform $(\epsilon_{2^i}, 2)$ noisy sample of K and is also an $(\epsilon_{2^{i_0}}, 2)$ noisy sample of K, Algorithm ParfreeDeclutter *returns a point set P_0 such that $d_H(P_0, K) \leq (87 + 16\sqrt{2})\epsilon_{2^{i_0}}$.*

Theoretical guarantees

Theorem

Given a point set P and i_0 such that for all $i > i_0$, P is a weak uniform $(\epsilon_{2^i}, 2)$ noisy sample of K and is also an $(\epsilon_{2^{i_0}}, 2)$ noisy sample of K, Algorithm ParfreeDeclutter *returns a point set P_0 such that $d_H(P_0, K) \leq (87 + 16\sqrt{2})\epsilon_{2^{i_0}}$.*

Consider $k_0 = 2^{i_0}$.

Theoretical guarantees

Theorem

Given a point set P and i_0 such that for all $i > i_0$, P is a weak uniform $(\epsilon_{2^i}, 2)$ noisy sample of K and is also an $(\epsilon_{2^{i_0}}, 2)$ noisy sample of K, Algorithm ParfreeDeclutter *returns a point set P_0 such that $d_H(P_0, K) \leq (87 + 16\sqrt{2})\epsilon_{2^{i_0}}$.*

Consider $k_0 = 2^{i_0}$.

- As algorithm reaches $k = k_0$, all bad points are removed.

Theoretical guarantees

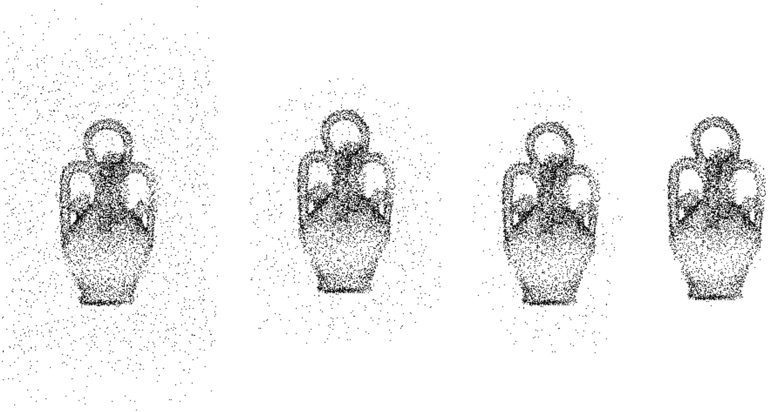
Theorem

Given a point set P and i_0 such that for all $i > i_0$, P is a weak uniform $(\epsilon_{2^i}, 2)$ noisy sample of K and is also an $(\epsilon_{2^{i_0}}, 2)$ noisy sample of K, Algorithm ParfreeDeclutter *returns a point set P_0 such that $d_H(P_0, K) \leq (87 + 16\sqrt{2})\epsilon_{2^{i_0}}$.*

Consider $k_0 = 2^{i_0}$.

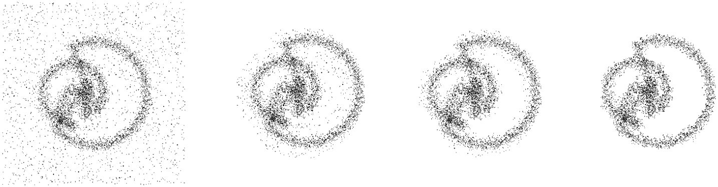
- As algorithm reaches $k = k_0$, all bad points are removed.
- As algorithm continues with $k < k_0$, no harm done!

Experimental results

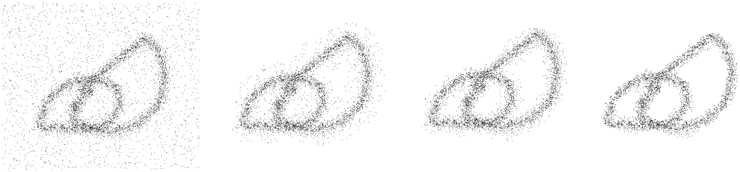


Input $\quad k = 1024 \quad k = 256 \quad k = 1$

Experimental results



Experimental results



MNIST Digits – 1 and 7

					Error(%)
Original	# Digit 1	1352	# Digit 7	1279	0.66
Swap. Noise	# Mislabel 1	270	# Mislabel 7	266	4.1
	Digit 1		Digit 7		
	# Removed	# True Noise	# Removed	# True Noise	
Denoising	*314*	*264*	*17*	*1*	*2.45*
Back. Noise	# Noisy 1	250	# Noisy 7	250	1.15
	Digit 1		Digit 7		
	# Removed	# True Noise	# Removed	# True Noise	
Denoising	*294*	*250*	*277*	*250*	*0.75*

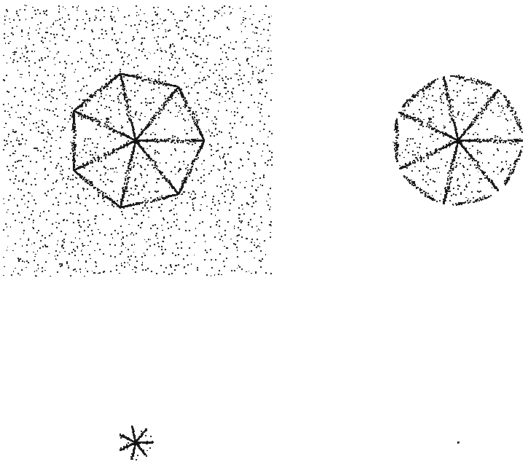
Table: Denoising on digits 1 and 7 from the MNIST. Using linear SVM as classifier.

MNIST Digits – All

$c_{declutter} = 1.5, c_{resample} = 2.2, L_1$		
Digit	# Removed	#True Noise
0	369	311
1	1703	1025
2	107	96
3	584	383
4	575	468
5	652	337
6	1011	585
7	1558	930
8	699	300
9	1179	776

Table: Denoising on all 60k MNIST. Every class has about 6000 points and about 1200 are noises.

Bad example



Topological inference using Declutter

Theorem

If P is an ϵ_k uniform sampling of $K \subset \mathbb{R}^d$, with $\epsilon_k < \frac{1}{28}\mathrm{wfs}(K)$. Then for all α, $\alpha' \in [7\epsilon_k, \mathrm{wfs}(K) - 7\epsilon_k]$ such that $\alpha' - \alpha > 14\epsilon_k$ and for all $\lambda \in (0, \mathrm{wfs}(K))$, we have

$$H_*(K^\lambda) \cong H_*(C_\alpha(Q_n) \hookrightarrow C_{\alpha'}(Q_n)).$$

Talk Outline

In this talk, we consider three different settings to explore:

What are natural ways to model noise in input metric, and how to process such noise effeciently with theoretical guarantees.

- Quest 1: towards parameter-free denoising for embedded point cloud data (PCD)
  - Declutter algorithm – one parameter
  - Declustter+Resampling – parameter free, but requires stronger sampling conditions
- Quest 2: metric embedding with outliers
- Quest 3: recovering shortest path metrics from perturbed graphs

Talk Outline

In this talk, we consider three different settings to explore:

What are natural ways to model noise in input metric, and how to process such noise effeciently with theoretical guarantees.

- Quest 1: towards parameter-free denoising for embedded point cloud data (PCD)
  - Declutter algorithm – one parameter
  - Declustter+Resampling – parameter free, but requires stronger sampling conditions
- Quest 2: metric embedding with outliers
- Quest 3: recovering shortest path metrics from perturbed graphs

Second Quest

Metric embedding with outliers

Joint work with Anastasios Sidiropoulos

Problem Setup

Input: A discrete n-point metric space $(X = \{x_1, \ldots, x_n\}, \rho)$

- (X, ρ) approximately comes from a "nice" *target metric space*
- some input points could have corrupted / erroneous distance to other points, they are "outliers"

Problem Setup

Input: A discrete n-point metric space $(X = \{x_1, \ldots, x_n\}, \rho)$

- (X, ρ) approximately comes from a "nice" *target metric space*
- some input points could have corrupted / erroneous distance to other points, they are "outliers"

Output: A "near-optimal" set of outliers $K \subset X$ together with a "low-distortion" embedding of $(X \setminus K, \rho)$ into some target metric space

- the target space could be a tree metric, ultrametric, or constant-dimensional Euclidean space.

Notations

Definition (Embedding)

Given two metric spaces $\mathcal{X} = (X, \rho_X)$ and $\mathcal{Y} = (Y, \rho_Y)$, an *embedding* of $\mathcal{X}$ into $\mathcal{Y}$ is simply a map $\phi : X \to Y$.

- ϕ is an *isometric embedding* if for any $x, x' \in X$, $\rho_X(x, x') = \phi_Y(\phi(x), \phi(x'))$.
- ϕ is an *ε-distorted embedding* if for any $x, x' \in X$, $|\rho_X(x, x') - \rho_Y(\phi(x), \phi(x')| \leq \varepsilon$. Alternatively, we say that $\mathcal{X}$ admits an embedding into $\mathcal{Y}$ with (additive) distortion ε.

Optimization Problem

Minimum outlier-embedding problem: Given a discrete n-point metric space $(X = \{x_1, \ldots, x_n\}, \rho)$, compute the *smallest set* $K^* \subset X$ such that $(X \setminus K^*, \rho)$ embeds into a target metric space either isometrically, or with distortion at most ε.

- Choices of target metric spaces: ultrametric, tree metric, constant-dimensional Euclidean space $\mathbb{R}^d$
- The set K^* is refered to as the *optimal set of outliers*

Optimization Problem

Minimum outlier-embedding problem: Given a discrete n-point metric space $(X = \{x_1, \ldots, x_n\}, \rho)$, compute the *smallest set* $K^* \subset X$ such that $(X \setminus K^*, \rho)$ embeds into a target metric space either isometrically, or with distortion at most ε.

- Choices of target metric spaces: ultrametric, tree metric, constant-dimensional Euclidean space $\mathbb{R}^d$
- The set K^* is refered to as the *optimal set of outliers*

$$
\begin{array}{cc}
 & \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} &
\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 2 & 2 \\ 1 & 2 & 0 & 2 \\ 1 & 2 & 2 & 0 \end{bmatrix}
\end{array}
$$

Input metric $\mathcal{X}$

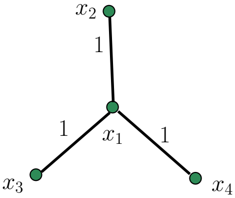
Optimization Problem

Minimum outlier-embedding problem: Given a discrete n-point metric space $(X = \{x_1, \dots, x_n\}, \rho)$, compute the *smallest set* $K^* \subset X$ such that $(X \setminus K^*, \rho)$ embeds into a target metric space either isometrically, or with distortion at most ε.

- Choices of target metric spaces: ultrametric, tree metric, constant-dimensional Euclidean space $\mathbb{R}^d$
- The set K^* is refered to as the *optimal set of outliers*

$$\begin{array}{c c} & \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 2 & 2 \\ 1 & 2 & 0 & 2 \\ 1 & 2 & 2 & 0 \end{bmatrix} \end{array}$$

Input metric $\mathcal{X}$



Isometric embedding to a tree metric

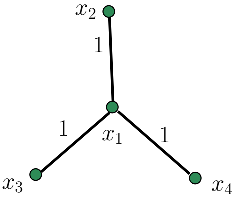
Optimization Problem

Minimum outlier-embedding problem: Given a discrete n-point metric space $(X = \{x_1, \ldots, x_n\}, \rho)$, compute the *smallest set* $K^* \subset X$ such that $(X \setminus K^*, \rho)$ embeds into a target metric space either isometrically, or with distortion at most ε.

- Choices of target metric spaces: ultrametric, tree metric, constant-dimensional Euclidean space $\mathbb{R}^d$
- The set K^* is refered to as the *optimal set of outliers*

$$\begin{array}{cc} & \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} & \begin{bmatrix} 0 & 1.1 & 0.89 & 1.05 \\ 1.1 & 0 & 2.12 & 1.95 \\ 0.89 & 2.12 & 0 & 2 \\ 1.05 & 1.95 & 2 & 0 \end{bmatrix} \end{array}$$

Input metric $\mathcal{X}'$



Isometric embedding to a tree metric

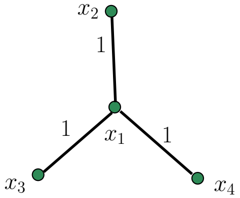
Optimization Problem

Minimum outlier-embedding problem: Given a discrete n-point metric space $(X = \{x_1, \ldots, x_n\}, \rho)$, compute the *smallest set* $K^* \subset X$ such that $(X \setminus K^*, \rho)$ embeds into a target metric space either isometrically, or with distortion at most ε.

- Choices of target metric spaces: ultrametric, tree metric, constant-dimensional Euclidean space $\mathbb{R}^d$
- The set K^* is refered to as the *optimal set of outliers*

$$\begin{array}{c c} & \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} & \begin{bmatrix} 0 & 1.1 & 0.89 & 1.05 \\ 1.1 & 0 & 2.12 & 1.95 \\ 0.89 & 2.12 & 0 & 2 \\ 1.05 & 1.95 & 2 & 0 \end{bmatrix} \end{array}$$

Input metric $\mathcal{X}'$



An embedding with additive distortion 0.2

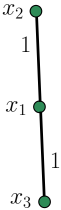
Optimization Problem

Minimum outlier-embedding problem: Given a discrete n-point metric space $(X = \{x_1, \ldots, x_n\}, \rho)$, compute the *smallest set* $K^* \subset X$ such that $(X \setminus K^*, \rho)$ embeds into a target metric space either isometrically, or with distortion at most ε.

- Choices of target metric spaces: ultrametric, tree metric, constant-dimensional Euclidean space $\mathbb{R}^d$
- The set K^* is refered to as the *optimal set of outliers*

$$\begin{array}{c c} & \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} & \begin{bmatrix} 0 & 1.1 & 0.89 & 1.05 \\ 1.1 & 0 & 2.12 & 1.95 \\ 0.89 & 2.12 & 0 & 2 \\ 1.05 & 1.95 & 2 & 0 \end{bmatrix} \end{array}$$

Input metric $\mathcal{X}'$



Outlier embedding to $\mathbb{R}^1$ with distortion 0.13

Hardness of the Outlier Embedding

Theorem

The problem of minimum outlier embedding into a tree metric, an ultrametric, or $\mathbb{R}^d$, is NP-hard.

Furthermore, assuming the Unique Games Conjecture, it is NP-hard to approximate the isometric version with a factor of $2 - \nu$ *for any* $\nu > 0$.

Hardness of the Outlier Embedding

Theorem

The problem of minimum outlier embedding into a tree metric, an ultrametric, or $\mathbb{R}^d$, is NP-hard.

Furthermore, assuming the Unique Games Conjecture, it is NP-hard to approximate the isometric version with a factor of $2 - \nu$ for any $\nu > 0$.

Our next goal

Efficient approximation algorithms for the outlier-embedding problems.

Hardness of the Outlier Embedding

Theorem

The problem of minimum outlier embedding into a tree metric, an ultrametric, or $\mathbb{R}^d$, is NP-hard.

Furthermore, assuming the Unique Games Conjecture, it is NP-hard to approximate the isometric version with a factor of $2 - \nu$ for any $\nu > 0$.

Our next goal

Efficient approximation algorithms for the outlier-embedding problems.

- We developed various approximation algorithms

Hardness of the Outlier Embedding

Theorem

The problem of minimum outlier embedding into a tree metric, an ultrametric, or $\mathbb{R}^d$, is NP-hard.

Furthermore, assuming the Unique Games Conjecture, it is NP-hard to approximate the isometric version with a factor of $2 - \nu$ for any $\nu > 0$.

Our next goal

Efficient approximation algorithms for the outlier-embedding problems.

- We developed various approximation algorithms
- Focus on special case: *isometric outlier-embedding into $\mathbb{R}^d$*

First Approximation Algorithm

Theorem (2-approximation)

Given an n-point metric space (X, ρ), there is an algorithm that can compute at most $2|K^*|$ number of points $K \subset X$, such that $(X \setminus K, \rho)$ admits an isometric embeddign into $\mathbb{R}^d$. The algorithm runs in $O(n^{d+1})$ time.

First Approximation Algorithm

Theorem (2-approximation)

Given an n-point metric space (X, ρ), there is an algorithm that can compute at most $2|K^*|$ number of points $K \subset X$, such that $(X \setminus K, \rho)$ admits an isometric embeddign into $\mathbb{R}^d$. The algorithm runs in $O(n^{d+1})$ time.

- There is a randomized algorithm that can improve the running time to $O(n^2)$ while worsening the approximation factor (w.r.to the number of outliers) to $(2 + \nu)$ for $\nu > 0$.

Approximation Algorithm

Input: An n-point metric space (X, ρ), dimension $d > 1$
Output: A set of outliers $\widehat{K} \subset X$

---

1. Initialize the set of candidate outlier sets $\mathcal{C}$ to empty set
2. For each $d+1$ distinct points $Y_d = \{y_0, \ldots, y_d\} \subset X$:
   - Initialize sets $Z = K = \emptyset$
   - If (Y_d, ρ) is *not* d-embeddable, return to step-2.
   - For each remaining point $x \in X \setminus Y_d$, check whether $(Y_d \cup \{x\}, \rho)$ is d-embeddable. If *yes*, insert x to Z; otherwise, add x to the outlier set K.
   - Construct a graph $G = (Z, E)$ where $(z, z') \in E$ iff $(Y_d \cup \{z, z'\}, \rho)$ is *not d-embeddable*.
   - Compute a 2-approximation $Z' \subset Z$ of the vertex cover of G. Set $K = K \cup Z'$.
   - Add K to the collection of candidate outlier sets $\mathcal{C}$.
3. Return $\widehat{K} \in \mathcal{C}$ with smallest cardinality as the outlier set.

Illustration of Algorithm

Input: An n-point metric space (X, ρ), dimension $d > 1$

Illustration of Algorithm

Input: An n-point metric space (X, ρ), dimension $d > 1$

$$Y_d = \{y_0, y_1, y_2\}$$

$$d(y_0, y_1) = 3$$
$$d(y_1, y_2) = 4$$
$$d(y_0, y_2) = 5$$

Illustration of Algorithm

Input: An n-point metric space (X, ρ), dimension $d > 1$

$$Y_d = \{y_0, y_1, y_2\}$$

$$d(y_0, y_1) = 3$$
$$d(y_1, y_2) = 4$$
$$d(y_0, y_2) = 5$$

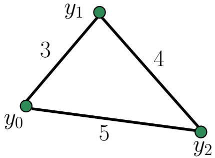
Is $Y_d = \{y_0, y_1, y_2\}$ embeddable in $\mathbb{R}^2$?

Illustration of Algorithm

Input: An n-point metric space (X, ρ), dimension $d > 1$

$$Y_d = \{y_0, y_1, y_2\}$$

$$d(y_0, y_1) = 3$$
$$d(y_1, y_2) = 4$$
$$d(y_0, y_2) = 5$$



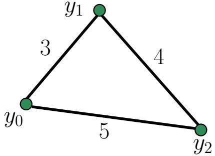
Is $Y_d = \{y_0, y_1, y_2\}$ embeddable in $\mathbb{R}^2$?

Illustration of Algorithm

Input: An n-point metric space (X, ρ), dimension $d > 1$

$$Y_d = \{y_0, y_1, y_2\}$$

$$d(y_0, y_1) = 3$$
$$d(y_1, y_2) = 4$$
$$d(y_0, y_2) = 5$$



Z: points consistent with $\{y_0, y_1, y_2\}$

K: points not consistent with $\{y_0, y_1, y_2\}$ / outliers

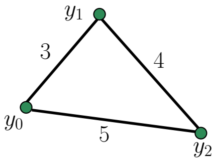
Is $Y_d = \{y_0, y_1, y_2\}$ embeddable in $\mathbb{R}^2$?

Illustration of Algorithm

Input: An n-point metric space (X, ρ), dimension $d > 1$

$$Y_d = \{y_0, y_1, y_2\}$$

$$d(y_0, y_1) = 3$$
$$d(y_1, y_2) = 4$$
$$d(y_0, y_2) = 5$$



Z: points consistent with $\{y_0, y_1, y_2\}$

K: points not consistent with $\{y_0, y_1, y_2\}$ / outliers

For each point $x \in X \setminus Y_d$, is $\{y_0, y_1, y_2, x\}$ embeddable in $\mathbb{R}^2$?

Illustration of Algorithm

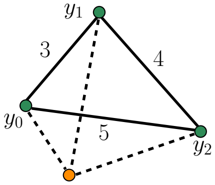
Input: An n-point metric space (X, ρ), dimension $d > 1$

$$Y_d = \{y_0, y_1, y_2\}$$

$$d(y_0, y_1) = 3$$
$$d(y_1, y_2) = 4$$
$$d(y_0, y_2) = 5$$

Z: points consistent with $\{y_0, y_1, y_2\}$

K: points not consistent with $\{y_0, y_1, y_2\}$ / outliers



If yes, add x to Z; otherwise add x to outlier set K.

Illustration of Algorithm

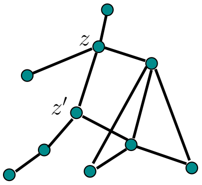
Input: An n-point metric space (X, ρ), dimension $d > 1$

$$Y_d = \{y_0, y_1, y_2\}$$

$$d(y_0, y_1) = 3$$
$$d(y_1, y_2) = 4$$
$$d(y_0, y_2) = 5$$

Z: points consistent with $\{y_0, y_1, y_2\}$

K: points not consistent with $\{y_0, y_1, y_2\}$ / outliers



Construct $G = (Z, E)$ with $(z, z') \in E$ iff $\{y_0, y_1, y_2, z, z'\}$ not d-embeddable.

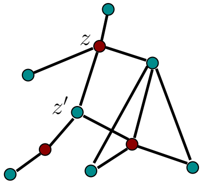
Illustration of Algorithm

Input: An n-point metric space (X, ρ), dimension $d > 1$

$$Y_d = \{y_0, y_1, y_2\}$$

$$d(y_0, y_1) = 3$$
$$d(y_1, y_2) = 4$$
$$d(y_0, y_2) = 5$$

Z: points consistent with $\{y_0, y_1, y_2\}$
K: points not consistent with $\{y_0, y_1, y_2\}$ / outliers



Take $Z' \subset Z$ vertex cover of G, add Z' to outlier set K

Approximation Algorithm

Input: An n-point metric space (X, ρ), dimension $d > 1$
Output: A set of outliers $\widehat{K} \subset X$

---

1. Initialize the set of candidate outlier sets $\mathcal{C}$ to empty set
2. For each $d + 1$ distinct points $Y_d = \{y_0, \ldots, y_d\} \subset X$:
   - Initialize sets $Z = K = \emptyset$
   - If (Y_d, ρ) is *not* d-embeddable, return to step-2.
   - For each remaining point $x \in X \setminus Y_d$, check whether $(Y_d \cup \{x\}, \rho)$ is d-embeddable. If *yes*, insert x to Z; otherwise, add x to the outlier set K.
   - Construct a graph $G = (Z, E)$ where $(z, z') \in E$ iff $(Y_d \cup \{z, z'\}, \rho)$ is *not d-embeddable*.
   - Compute a 2-approximation $Z' \subset Z$ of the vertex cover of G. Set $K = K \cup Z'$.
   - Add K to the collection of candidate outlier sets $\mathcal{C}$.
3. Return $\widehat{K} \in \mathcal{C}$ with smallest cardinality as the outlier set.

Correctness

Lemma

The $\widehat{K}$ output by previous algorithm satisfies:

- $|K^*| \leq |\widehat{K}| \leq 2|K^*|$
- $(X \setminus \widehat{K}, \rho)$ *is d-embeddable.*

Correctness

Lemma

The $\widehat{K}$ output by previous algorithm satisfies:

- $|K^*| \leq |\widehat{K}| \leq 2|K^*|$
- $(X \setminus \widehat{K}, \rho)$ *is d-embeddable.*

The correctness follows from the following classic result in distance geometry; see e.g, [Blumenthal'70].

Theorem

A metric space (Y, ρ_Y) is d-embeddable in $\mathbb{R}^d$ iff there exists $d+1$ points say Y_d such that:

(i) (Y_d, ρ_Y) *is d-embeddable; and*

(ii) *for any $y, y' \in Y \setminus Y_d$, $(Y_d \cup \{y, y'\}, \rho_Y)$ is d-embeddable.*

First Approximation Algorithm

Theorem (2-approximation)

Given an n-point metric space (X, ρ), there is an algorithm that can compute at most $2|K^*|$ number of points $K \subset X$, such that $(X \setminus K, \rho)$ admits an isometric embeddign into $\mathbb{R}^d$. The algorithm runs in $O(n^{d+1})$ time.

Improving Efficiency

Theorem (Improved 2-approximation)

Given an n-point metric space (X, ρ), there is a $O(n^2)$ time randomized algorithm that can compute at most $(2 + \nu)|K^*|$ number of points $K \subset X$, such that with constant probability, $(X \setminus K, \rho)$ admits an isometric embeddign into $\mathbb{R}^d$.

Improving Efficiency

Theorem (Improved 2-approximation)

Given an n-point metric space (X, ρ), there is a $O(n^2)$ time randomized algorithm that can compute at most $(2+\nu)|K^*|$ number of points $K \subset X$, such that with constant probability, $(X \setminus K, \rho)$ admits an isometric embeddign into $\mathbb{R}^d$.

- The big O notation hides constants depending on $(\frac{1}{\nu})^d$.
- Algorithm still simple, but analysis much more involved.

Improving Efficiency

Theorem (Improved 2-approximation)

Given an n-point metric space (X, ρ), there is a $O(n^2)$ time randomized algorithm that can compute at most $(2+\nu)|K^*|$ number of points $K \subset X$, such that with constant probability, $(X \setminus K, \rho)$ admits an isometric embeddign into $\mathbb{R}^d$.

- The big O notation hides constants depending on $(\frac{1}{\nu})^d$.
- Algorithm still simple, but analysis much more involved.
- Algorithm can be extended to embedding with low-distortion.

Talk Outline

In this talk, we consider three different settings to explore:

What are natural ways to model noise in input metric, and how to process such noise effeciently with theoretical guarantees.

- Quest 1: towards parameter-free denoising for embedded point cloud data (PCD)
- Quest 2: metric embedding with outliers
  - identifying near optimal number of outliers so that the remaining points can be embedded into a target metric space isometrically or with low additive distortion.
- Quest 3: recovering shortest path metric from perturbed graphs

Talk Outline

In this talk, we consider three different settings to explore:

What are natural ways to model noise in input metric, and how to process such noise effeciently with theoretical guarantees.

- Quest 1: towards parameter-free denoising for embedded point cloud data (PCD)
- Quest 2: metric embedding with outliers
  - identifying near optimal number of outliers so that the remaining points can be embedded into a target metric space isometrically or with low additive distortion.
- Quest 3: recovering shortest path metric from perturbed graphs

Last Quest

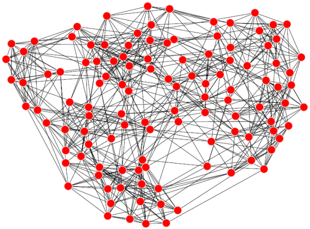
Recovering shortest-path metric from perturbed graphs

Joint work with Minghao Tian, Srinivasan Parthasarathy, and David Sivakoff

Problem Setup

Input: An observed graph $G = (V, E)$

- G is a "noisy" observation of a true graph G^*
- the metric of interest is the shortest path metric d_{G^*}

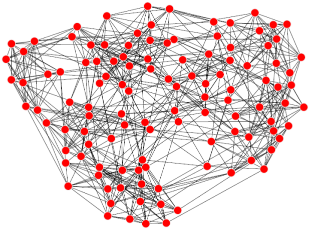


Problem Setup

Input: An observed graph $G = (V, E)$

- G is a "noisy" observation of a true graph G^*
- the metric of interest is the shortest path metric d_{G^*}

Output: Recover (approximately) the "true" shortest path metric d_{G^*} from G

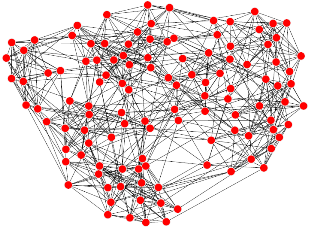


Problem Setup

Input: An observed graph $G = (V, E)$

- G is a "noisy" observation of a *true graph* G^*
- the metric of interest is the shortest path metric d_{G^*}

Output: Recover (approximately) the "true" shortest path metric d_{G^*} from G

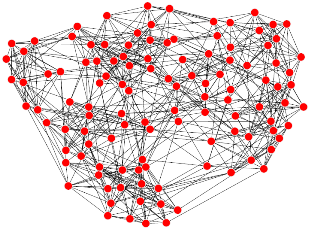


Problem Setup

Input: An observed graph $G = (V, E)$

- G is a *"noisy"* observation of a *true graph* G^*
- the metric of interest is the shortest path metric d_{G^*}

Output: Recover (approximately) the "true" shortest path metric d_{G^*} from G



The model

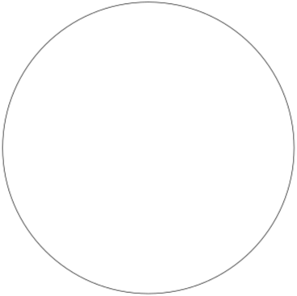
The true graph $G^* = (V, E^*)$

- V sampled i.i.d from a L-doubling measure $\mu : M \to \mathbb{R}^+$ on a compact geodesic metric space (M, d_M)
- $E^* = E^*_r = \{(u, v) \mid d_M(u, v) \leq r, u, v \in V\}$ is the r-neighborhood graph for some parameter $r > 0$

The observed graph $G = (V, E)$: A (p, q)-perturbation of G^* where

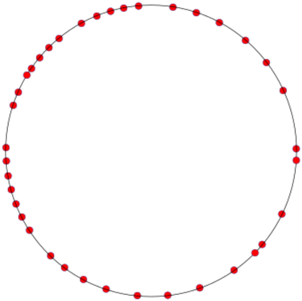
- (p-deletion): For each edge $e = (u, v) \in E^*$, we have $e \in E$ with probability $1 - p$
- (q-insertion): For any pair of nodes $u, v \in V$ s.t. $(u, v) \notin E^*$, we have $(u, v) \in E$ with probability q

Illustration



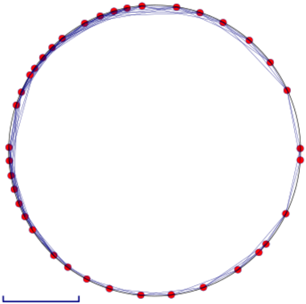
Hidden domain M

Illustration



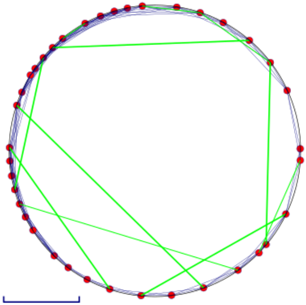
Graph Nodes V

Illustration



True graph G^*

Illustration



Random perturbation G

Motivation

- In many graphs, e.g social networks, edges encode proximity between graph nodes in certain feature space.

Motivation

- In many graphs, e.g social networks, edges encode proximity between graph nodes in certain feature space.
- Sampling from a measure allows varing degree distribution

Motivation

- In many graphs, e.g social networks, edges encode proximity between graph nodes in certain feature space.
- Sampling from a measure allows varing degree distribution
- Random Erdös-Rényi type perturbation allows exceptions / noise

Motivation

- In many graphs, e.g social networks, edges encode proximity between graph nodes in certain feature space.
- Sampling from a measure allows varing degree distribution
- Random Erdös-Rényi type perturbation allows exceptions / noise
- The model related to superposing a "structured subgraph" and a "random subgraph"
  - e.g, [Bollobás and Chung, 1988], [Watts and Strogatz, 1998], [Kleinberg 2000] (the small-world phenomenon), ...

Motivation

- In many graphs, e.g social networks, edges encode proximity between graph nodes in certain feature space.
- Sampling from a measure allows varing degree distribution
- Random Erdös-Rényi type perturbation allows exceptions / noise
- The model related to superposing a "structured subgraph" and a "random subgraph"
    - e.g, [Bollobás and Chung, 1988], [Watts and Strogatz, 1998], [Kleinberg 2000] (the small-world phenomenon), ...
- Shortest path metric natural choice in many situations (especially for sparse graphs)

Motivation

- In many graphs, e.g social networks, edges encode proximity between graph nodes in certain feature space.
- Sampling from a measure allows varing degree distribution
- Random Erdös-Rényi type perturbation allows exceptions / noise
- The model related to superposing a "structured subgraph" and a "random subgraph"
  - e.g, [Bollobás and Chung, 1988], [Watts and Strogatz, 1998], [Kleinberg 2000] (the small-world phenomenon), ...
- Shortest path metric natural choice in many situations (especially for sparse graphs)
- However, shortest path metric sensitive to random perturbations (especially "short-cuts")

Our Goal

The true graph $G^* = (V, E^*)$

- V sampled i.i.d from a L-doubling measure $\mu : M \to \mathbb{R}^+$ on a compact geodesic metric space (M, d_M)
- $E^* = E_r^* = \{(u, v) \mid d_M(u, v) \leq r, u, v \in V\}$ is the r-neighborhood graph for some parameter $r > 0$

The observed graph $G = (V, E)$: A (p, q)-perturbation of G^*

Our goal

Recover the shortest path metric d_{G^*} from G with approximation guarantee.

Assumptions

Definition (Doubling measure)

A measure $\mu : X \to \mathbb{R}^+$ on a metric space (X, d) is said to be *L-doubling* if all metric balls have finite and positive measure and that there is a constant L such that for all $x \in X$ and $r > 0$,

$$\mu(B(x, 2r)) \leq L \cdot \mu(B(x, r)).$$

We call L the *doubling constant* and $\ell = \log_2 L$ the *doubling dimension of* μ.

Assumptions

Definition (Doubling measure)

A measure $\mu : X \to \mathbb{R}^+$ on a metric space (X, d) is said to be *L-doubling* if all metric balls have finite and positive measure and that there is a constant L such that for all $x \in X$ and $r > 0$,

$$\mu(B(x, 2r)) \leq L \cdot \mu(B(x, r)).$$

We call L the *doubling constant* and $\ell = \log_2 L$ the *doubling dimension of* μ.

Assumption-R: The parameter r is large enough that $\mu(B(x, \frac{r}{2})) \geq s \geq \frac{12 \ln n}{n}$ for any $x \in M$.

Effect of Deletion

Theorem (Deletion only)

Let G^* be the true graph generated as described, and G a graph obtained by deleting each edge in G^* with probability p. Assuming Assumption-R, *then for $p < \frac{1}{2}e^{-\frac{2\ln n}{sn}}$ with probability at least $1 - \frac{1}{n^{\Omega(1)}}$, the shortest path metric d_G in the observed graph is a 2-approximation of the shortest path metric d_{G^*} in the true graph; that is,*

$$\frac{1}{2}d_G(u,v) \leq d_{G^*}(u,v) \leq 2d_G(u,v).$$

Effect of Deletion

Theorem (Deletion only)

Let G^* be the true graph generated as described, and G a graph obtained by deleting each edge in G^* with probability p. Assuming Assumption-R, *then for $p < \frac{1}{2}e^{-\frac{2\ln n}{sn}}$ with probability at least $1 - \frac{1}{n^{\Omega(1)}}$, the shortest path metric d_G in the observed graph is a 2-approximation of the shortest path metric d_{G^*} in the true graph; that is,*

$$\frac{1}{2}d_G(u,v) \leq d_{G^*}(u,v) \leq 2d_G(u,v).$$

Suppose an edge $(u,v) \in E^*$ is deleted in the observed graph G.

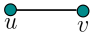
Effect of Deletion

Theorem (Deletion only)

Let G^* be the true graph generated as described, and G a graph obtained by deleting each edge in G^* with probability p. Assuming Assumption-R, *then for $p < \frac{1}{2}e^{-\frac{2\ln n}{sn}}$ with probability at least $1 - \frac{1}{n^{\Omega(1)}}$, the shortest path metric d_G in the observed graph is a 2-approximation of the shortest path metric d_{G^*} in the true graph; that is,*

$$\frac{1}{2}d_G(u,v) \leq d_{G^*}(u,v) \leq 2d_G(u,v).$$

Suppose an edge $(u,v) \in E^*$ is deleted in the observed graph G.



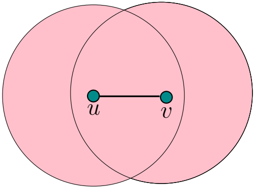
Effect of Deletion

Theorem (Deletion only)

Let G^* be the true graph generated as described, and G a graph obtained by deleting each edge in G^* with probability p. Assuming Assumption-R, *then for $p < \frac{1}{2}e^{-\frac{2\ln n}{sn}}$ with probability at least $1 - \frac{1}{n^{\Omega(1)}}$, the shortest path metric d_G in the observed graph is a 2-approximation of the shortest path metric d_{G^*} in the true graph; that is,*

$$\frac{1}{2}d_G(u,v) \leq d_{G^*}(u,v) \leq 2d_G(u,v).$$

Suppose an edge $(u, v) \in E^*$ is deleted in the observed graph G.



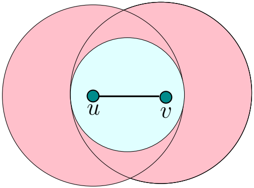
Effect of Deletion

Theorem (Deletion only)

Let G^* be the true graph generated as described, and G a graph obtained by deleting each edge in G^* with probability p. Assuming Assumption-R, *then for $p < \frac{1}{2}e^{-\frac{2\ln n}{sn}}$ with probability at least $1 - \frac{1}{n^{\Omega(1)}}$, the shortest path metric d_G in the observed graph is a 2-approximation of the shortest path metric d_{G^*} in the true graph; that is,*

$$\frac{1}{2}d_G(u,v) \leq d_{G^*}(u,v) \leq 2d_G(u,v).$$

Suppose an edge $(u, v) \in E^*$ is deleted in the observed graph G.



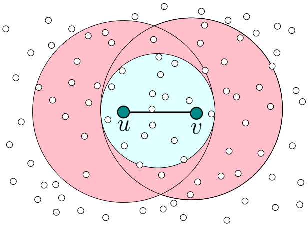
Effect of Deletion

Theorem (Deletion only)

Let G^* be the true graph generated as described, and G a graph obtained by deleting each edge in G^* with probability p. Assuming Assumption-R, *then for $p < \frac{1}{2}e^{-\frac{2\ln n}{sn}}$ with probability at least $1 - \frac{1}{n^{\Omega(1)}}$, the shortest path metric d_G in the observed graph is a 2-approximation of the shortest path metric d_{G^*} in the true graph; that is,*

$$\frac{1}{2}d_G(u,v) \leq d_{G^*}(u,v) \leq 2d_G(u,v).$$

Suppose an edge $(u,v) \in E^*$ is deleted in the observed graph G.



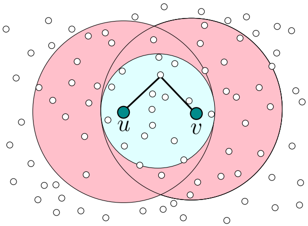
Effect of Deletion

Theorem (Deletion only)

Let G^* be the true graph generated as described, and G a graph obtained by deleting each edge in G^* with probability p. Assuming Assumption-R, *then for $p < \frac{1}{2}e^{-\frac{2\ln n}{sn}}$ with probability at least $1 - \frac{1}{n^{\Omega(1)}}$, the shortest path metric d_G in the observed graph is a 2-approximation of the shortest path metric d_{G^*} in the true graph; that is,*

$$\frac{1}{2}d_G(u,v) \leq d_{G^*}(u,v) \leq 2d_G(u,v).$$

Suppose an edge $(u, v) \in E^*$ is deleted in the observed graph G.

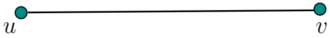


Effect of Insertion

Consider a *very-bad* inserted edge $(u, v) \in E$, meaning that $d_{G^*}(u, v) > 2$.

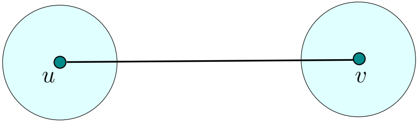
Effect of Insertion

Consider a *very-bad* inserted edge $(u, v) \in E$, meaning that $d_{G^*}(u, v) > 2$.



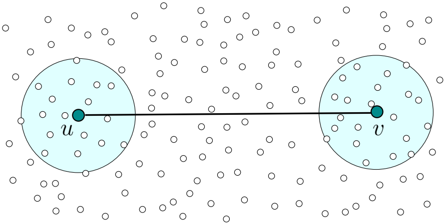
Effect of Insertion

Consider a *very-bad* inserted edge $(u, v) \in E$, meaning that $d_{G^*}(u, v) > 2$.



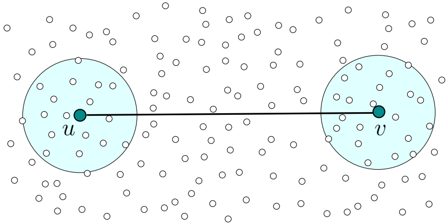
Effect of Insertion

Consider a *very-bad* inserted edge $(u, v) \in E$, meaning that $d_{G^*}(u, v) > 2$.



Effect of Insertion

Consider a *very-bad* inserted edge $(u, v) \in E$, meaning that $d_{G^*}(u, v) > 2$.



τ-Jaccard-Cleanup: Given graph G, for each edge $(u, v) \in G$, we keep the edge in a filtered graph $\widehat{G}$ iff

$$\rho_{u,v}(G) = \frac{|N_u^G \cap N_v^G|}{|N_u^G \cup N_v^G|} \geq \tau.$$

Main Result

Theorem

Given an observed graph G as a perturbed version of G^* as decribed before. Suppose Assumption-R holds, $sn = \omega(\ln n)$, the deletion probabily $p < \min\{1 - \frac{\sqrt{3}}{2}, \frac{1}{2}e^{-\frac{9 \ln n}{sn}}\}$, and that the insertion probability $q \leq cs$. Let $\widehat{G}_\tau$ denote the graph after τ-Jaccard-cleanup of G with $\tau \in (\frac{c}{1-p}q + o(1), \frac{2(1-p)^2}{15L^2(1+2c)})$. Then the shortest path distance metric $d_{\widehat{G}_\tau}$ from $\widehat{G}_\tau$ is a 2-approximation of the shortest path metric d_{G^*} of the true graph G^* with high probability.

Summary

In this talk:

- Quest 1: towards parameter-free denoising for embedded point cloud data (PCD)
- Quest 2: metric embedding with outliers
- Quest 3: recovering shortest path metrics from perturbed graphs

Summary

In this talk:

- Quest 1: towards parameter-free denoising for embedded point cloud data (PCD)
- Quest 2: metric embedding with outliers
- Quest 3: recovering shortest path metrics from perturbed graphs

One small step towards understanding / modeling noise in data, and how to process them with theoretical guarantees

Summary

In this talk:

- Quest 1: towards parameter-free denoising for embedded point cloud data (PCD)
- Quest 2: metric embedding with outliers
- Quest 3: recovering shortest path metrics from perturbed graphs

One small step towards understanding / modeling noise in data, and how to process them with theoretical guarantees

- Adaptive noise for PCD denoising with guarantees

Summary

In this talk:

- Quest 1: towards parameter-free denoising for embedded point cloud data (PCD)
- Quest 2: metric embedding with outliers
- Quest 3: recovering shortest path metrics from perturbed graphs

One small step towards understanding / modeling noise in data, and how to process them with theoretical guarantees

- Adaptive noise for PCD denoising with guarantees
- Outlier distance entries (instead of outlier points) in metric embedding

Summary

In this talk:

- Quest 1: towards parameter-free denoising for embedded point cloud data (PCD)
- Quest 2: metric embedding with outliers
- Quest 3: recovering shortest path metrics from perturbed graphs

One small step towards understanding / modeling noise in data, and how to process them with theoretical guarantees

- Adaptive noise for PCD denoising with guarantees
- Outlier distance entries (instead of outlier points) in metric embedding
- More general graph perturbation models / diffusion distance metric instead of shortest path metrics?