ROBUST TOPOLOGICAL INFERENCE

TOM NORMAN

1 Introduction

The goal of geometric inference on point cloud data is to recover the geometry of the underlying shape (i.e. reconstruction) or the topological deatures (i.e. Betti numbers). One way to do so is using distance functions, however one major drawback is lack of robustness to noist and outliers. In this paper the authors introduce smooth distance-like function (DTM) based on probability measure that has topological guarantees even in the presence of ouliers.
Moreover when empiriacl measures are chosen over the data these functions can be easily and intuetuvely evaluated. They also show how to use bootstrap to test the significance of the topological features and also choose the functions's hyperparameter based on it.
The code for everything in this summary is in [4].

2 Background

We'll define some key-concepts as distance function, sublevel sets, etc in order to understand the motives to using DTM.

2.1 Distance Function

Let $S \subset \mathbb{R}^d$ be compact set. The distance function on S is defind as

$$\Delta_S(x) = \inf_{y \in s} \|y - x\|, \forall x \in S$$

2.2 Sublevel Sets

Sublevel set of S defined as

$$L_t = \{x : \Delta_S(x) \leq t\}, t : 0 \to \infty$$

We'll refer to t as "time".
As t gets larger the topology of L_t changes: connected components, holes, etc appear and merge,
so for $s < t : L_s \subseteq L_t$.
Note that for $t \to \infty$ we get one big connected component.

2.3 Persistent Homology

The methos to compute the topological features that appear and merge as t gets larger.
When feature i appears it's assigned a birth time b_i (t for which it started to exist) and when it merges an older feature it's assigned a death time d_i.
Lifetime of feature i is defined as $|d_i - b_i|$. Intuitively the bigger the lifetime fo a feature the more relevant/persistent it is.

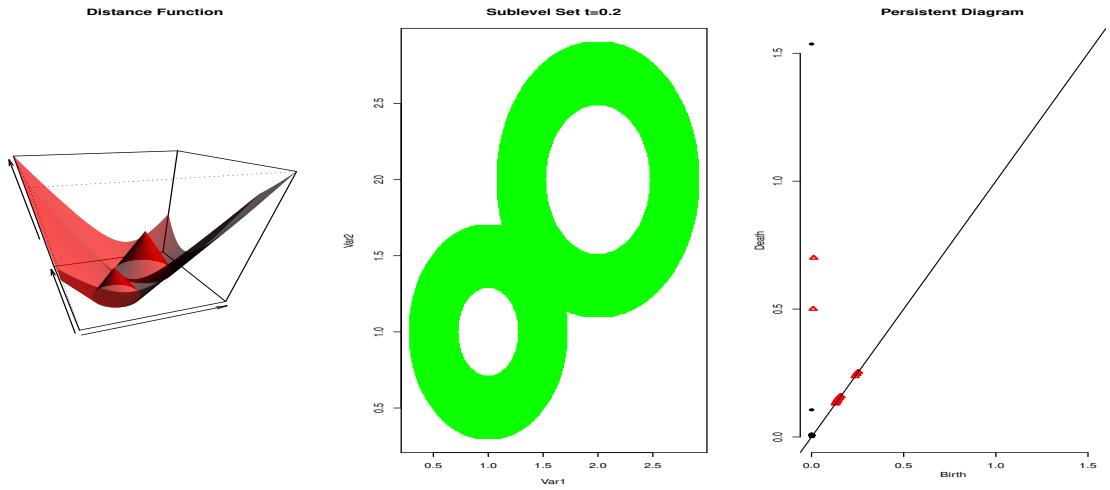

Figure 1: Plots of distance function, sublevel set and persistent diagram of two circles with different radii.

2.4 Persistent Diagram

A lifetime of a feature can be represented in the 2D plane as the coordinate (b_i, d_i).
The set of those coordinates for given set S is called the persistence digram of Δ_S. Note that birth of a feature comes before his death so the diagram is contained above the diagonal.
As we discussed before, the more persistent a feature is, the bigger its lifetime thus further away it's from the diagonal (where birth equals death). Features close to the diagonal can be seen as topological noise.
Note that in all persistence diagrams there exists a 0-dimension feature (connected component) at coordinate $(0, \infty)$.

2.5 Bottleneck Distance

Given 2 compact sets S_1, S_2 with corresponding distance functions Δ_1, Δ_2 and diagrams D_1, D_2 (including the diagonal), define the bottleneck distance between them as

$$W_\infty(D_1, D_2) = \min_{g: D_1 \to D_2} \sup_{z \in D_1} \|z - g(z)\|_\infty$$

g is a bijection.
In words: the bottleneck distance is the maximum L_∞ distance between the point os D_1, D_2 after minimzing over all pairing of the points (including the diagonals).
A property of persistence diagram is stability; according to the Persistence Stability Theorm (Cohen-Steiner et al. (2005); Chazal et al. (2012))

$$W_\infty(D_1, D_2) \leq \|\Delta_1 - \Delta_2\|_\infty$$

we'll use it later when discussing significance.

3 Motivation

Given sample $X_1, ..., X_2 \sim P$ the empirical distance function is defined as

$$\hat{\Delta}(x) = \min_{X_i} \|x - X_i\|$$

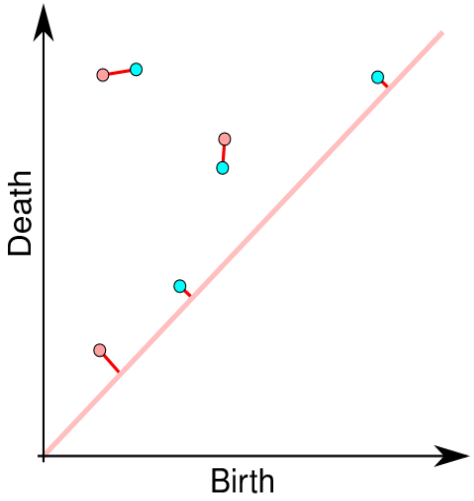

Figure 2: The bottleneck distance is the longest red edge.

Lemma 3.1. *Suppose that P is supported on S and has a density bounded away from 0 and ∞, then*

$$\sup_x |\hat{\Delta}(x) - \Delta_S(x)| \xrightarrow{P} 0$$

This lemma justifies using $\hat{\Delta}$ to estimate persistence homology of sublevel sets of Δ_S. In fact the sublevelsets of $\hat{\Delta}$ are balls around the data points with raduis t:

$$L_t = \left\{x : \hat{\Delta}(x) \leq t\right\} = \bigcup_{i=1}^{n} B_{x_i}(t)$$

which is the same as Cech complex, the persistent homology will then be extracted using filtration and linear algebra.
However, if the data includes noise or ouliers the emperical distance function $\hat{\Delta}$ no longer represents Δ_S truly (see Figure 3). The paper introduce a new notion of distance- distance to (probability) measure to overcome the noise/ouliers problem.

4 Distance to (Probability) Measure

Given probability measure P, for $m \in (0, 1)$ define DTM as

$$DTM^2_{P,m}(x) = \frac{1}{m}\int_0^m \left(G_x^{-1}(u)\right)^2 du$$

where $G_x(t) = \mathbb{P}\left(\|X - x\| \leq t\right)$.
When considering P as the emperical distribuition $\hat{P}$ (i.e. for n data points give each $\frac{1}{n}$ probability mass), we get

$$G_x(t) = \frac{\#\left\{X_i : X_i \in B_x(t)\right\}}{n}$$

$\Rightarrow G_x^{-1}(u)$ will give us the distance of the farthest X_i from x s.t. $\mathbb{P}\left(\|X - x\| \leq \|X_i - x\|\right) = u$

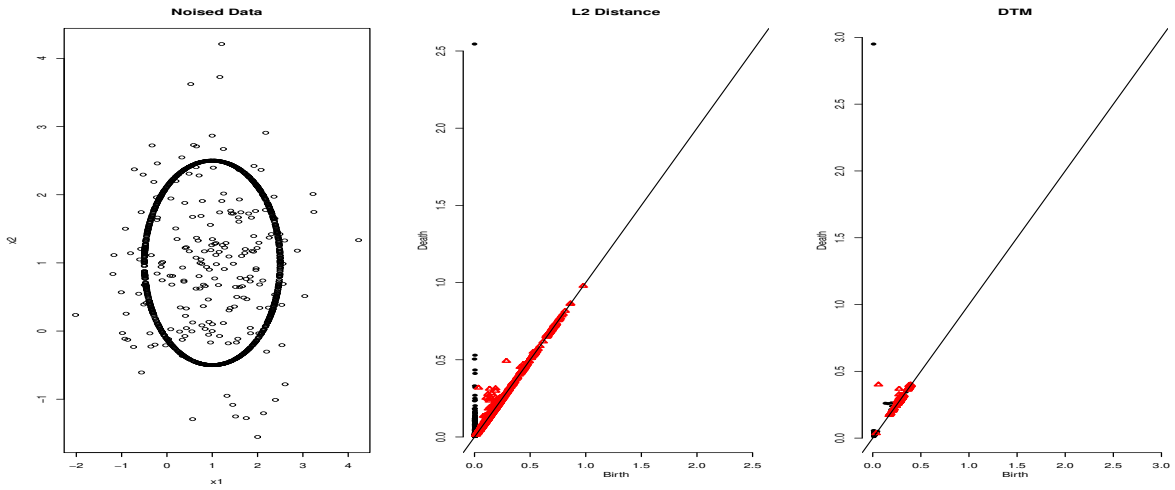

Figure 3: 1000 points samples from a circle + 200 points sampled from gaussian RV. We can see that the normal distance function creates lots of features while DTM removes most of them and leaves one that looks significant

Define $k = \lceil u * n \rceil$. Because $G_x^{-1}(u)$ is constant in the range $u \in (\frac{k}{n}, \frac{k+1}{n}]$, then $G_x^{-1}(u)$ becomes a step function with step width $\frac{1}{n}$ and the value of each step is the distance between x and its k-th nearest neighbor, which changes the integral to discrete sum of k nearest neighbors:

$$DTM^2_{X,k}(x) = \frac{1}{k} \sum_{X_i \in kNN(x)} \|X_i - x\|^2$$

Note that for k=1 we get the regular distance function.

5 Significance of Topological Features

5.1 Bootstrapping

We would like to get statistical estimations about our data like mean, variance and confidence intervals. Becuase the population (in our case the underlying shape) is unknown we will instead compute those estimation on our sampled data (the point cloud) by resampling with replacment from the sampled data number of times and creating an empirical distribution. This distribution is asymptotically consistent with the population distribuition.

Algorithm 1: Bootstrapping

Input: $X = \{x_i\}_{i=1}^n$ - point cloud data
$k \in [1, n-1]$ - smoothing parameter
B - number of resamples
α - significance level

Compute $DTM_{X,k}(x), \forall x \in S$

for *j = 1:B* **do**
  Draw n samples from X with replacement, denoted X_j
  Compute $\theta_j = \sqrt{n}\|DTM_{X,k}(x) - DTM_{X_j,k}(x)\|_\infty$

end

Compute $\hat{t}_\alpha = \inf_t \left(\sum_{j=1}^B \mathbf{1}_{\{\theta_j \geq t\}} \leq \alpha B\right)$

Result: $\hat{t}_\alpha$ - confidence band

In theory, a feature with birth and death time (b_i, d_i) is said to be significant if $|d_i - b_i| > 2\frac{t_\alpha}{\sqrt{n}}$ where t_α is defined by

$$\mathbb{P}\left(\sqrt{n}\|DTM_{X,k}(x) - DTM_{P,m}(x)\|_\infty > t_\alpha\right) = \alpha$$

When using empirical DTM, t_α can be estimated by bootstrapping to get $\hat{t}_\alpha$ which is defined by

$$\mathbb{P}\left(\sqrt{n}\|DTM_{X,k}(x) - DTM_{X_j,k}(x)\|_\infty > \hat{t}_\alpha | x_1, x_2, ..., x_n\right) = \alpha$$

To see why this makes sense let $\mathcal{D}$ be the set of persistence diagrams. Let D be the true diagram of the underlying shape and let $\hat{D}$ be the estimated diagram. Let

$$\mathcal{L}_n = \left\{E \in \mathcal{D} : W_\infty(\hat{D}, E) \leq \frac{\hat{t}_\alpha}{\sqrt{n}}\right\}$$

from persistence diagram stability we get Theorem 3.5 (in the paper)

$$W_\infty(D_P, D_Q) \geq \|DTM_{P,m}(x) - DTM_{Q,m}(x)\|_\infty$$

Taking Q as the empirical distribution of X we get

$$\mathbb{P}\left(D \in \mathcal{L}_n\right) = \mathbb{P}\left(W_\infty(D, \hat{D}) \leq \frac{\hat{t}_\alpha}{\sqrt{n}}\right) \geq \mathbb{P}\left(\sqrt{n}\|DTM_{P,m}(x) - DTM_{X,k}(x)\|_\infty \leq \hat{t}_\alpha\right) \underset{n\to\infty}{\to} 1 - \alpha$$

So $|d_i - b_i| > 2\frac{\hat{t}_\alpha}{\sqrt{n}} \Leftrightarrow$ the feature cannot be matched to the diagonal for any diagram in $\mathcal{L}_n$.
The diagonal represents features with 0 lifetime, so for $|d_i - b_i| \leq 2\frac{\hat{t}_\alpha}{\sqrt{n}}$ the feature isn't different from 0 lifetime feature, with $1 - \alpha$ probability.

6 Choosing Smoothing Parameter

Using a smoothing parameter rises the need for hyper-parameter tuning. So what is the metric for which a smoothing parameter considered good?
The paper suggets two quantities to measure the amount of siginificance information.
Define $\ell_i(m)$ as the lifetime of feature i with parameter m.

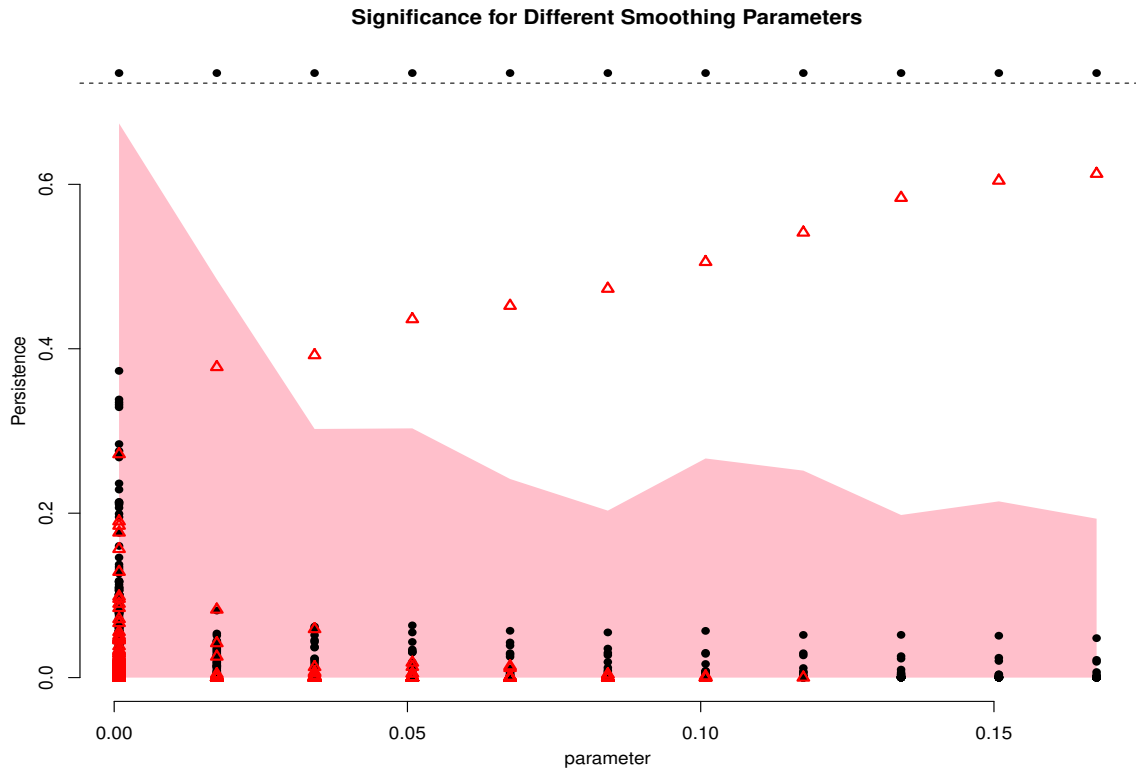

Figure 4: Using the same noised data from Figure 3, $k = [1 : 20 : 201], B = 30, \alpha = 0.05$. $m \geq 41/1200$ will maximize N and $m = 201/1200$ will maximize S.

Number of features that are α significant:

$$N(m) = \#\left\{i : \ell_i(m) > 2\frac{\hat{t}_\alpha}{\sqrt{n}}\right\}$$

Total α significant persistence:

$$S(m) = \sum_i \left[\ell_i(m) - 2\frac{\hat{t}_\alpha}{\sqrt{n}}\right]_+$$

The m that maximizes $N(m)$ or $S(m)$ is choosen (see Figure 4).

7 Criticism

The paper is very comprehensive with respect to the analysis of the proposed method. Most of the theorems and conclusions made for the general DTM (with general distribution P, not just the empirical $\hat{P}$). The empirical DTM is very intuitive- take the distance to a cluster of data points to smooth the noise, also the way to choose smoothing parameters is easy to understand.
In the paper they gave mainly syntethic examples which doesn't show if the method is really applicative, however they implemented almost everything in R, and made it open-source so we can try on real data. As a small bonus, in the end, they provide some points for further research.

8 Experiments

There is R package called 'TDA' that contains almost all of the functions from the paper. My first step was to install everything on a docker then learn the basics of R using the tutorial [3]. After I got the basics I started experimenting how robust DTM is, and finally I tried to classify numbers on MNIST using the package.

8.1 Unit Circle

In this sections I used 1000 samples from the unit circle and $k = [1, 21, 41]$.
(The following figures are of the same idea as Figure 4, showing the persistence diagram given k for multiple ks)
Figure 5 shows the unit circle with uniform noise as the number of samples varies.
Figure 6 shows the unit circle with gaussian noise as the number of samples varies.
Figure 7 shows the unit circle with 10 samples from a gaussian noise as the standard deviation varies.
As we can see DTM handles well when the number of samples change but their distribution is close to the real shape (the unit circle), but when they get further from it, even with small number of samples, DTM breaks.

8.2 Multiple Shapes

I combined several disjointed shapes (a torus, a circle inside a circle and a sphere) and computed the persistent diagram, then added gaussian noise and computed the persistent digaram, Figure 8.
Next I merged the shapes and recomputed the persistent diagrams, Figure 9.
(There are gifs on github to show those shapes [4]).
With both cases (disjoint and joint) DTM gave nice results, it removed most of the topological noise (features close to the diagonal) and kept the significant features (2 voids and 4 holes, and in the disjoint case an additional 3 connected components).

8.3 Shape Reconstruction

Another option for using DTM is its ability to smooth noise from point cloud of a shape using a grid and some threshold on the DTM value for each point on the grid. I used PointCleanNet dataset [5] to reconstruct Yoda's face from its noisy version, Figure 10, but it's possible for every shape in this dataset.

8.4 MNIST Point Cloud

I'll try to classify 0 vs 6 using DTM.

1. Take only the 0 and 6 examples and split to train/test
2. For each number iterate over all given examples, for each example run DTM with k=[1,2,5,10] and extract the hole with the maximum lifetime.
   Compute the mean of those lifetimes.
3. Iterate over all given examples in the test data, if the lifetime of the most significant hole is closet to the mean of 0, classify as 0, else as 6.
   If no hole or the lifetime is equaly close, don't classify.

I made this experiment also for 0 vs 9 and 6 vs 9. The results: (page 14)

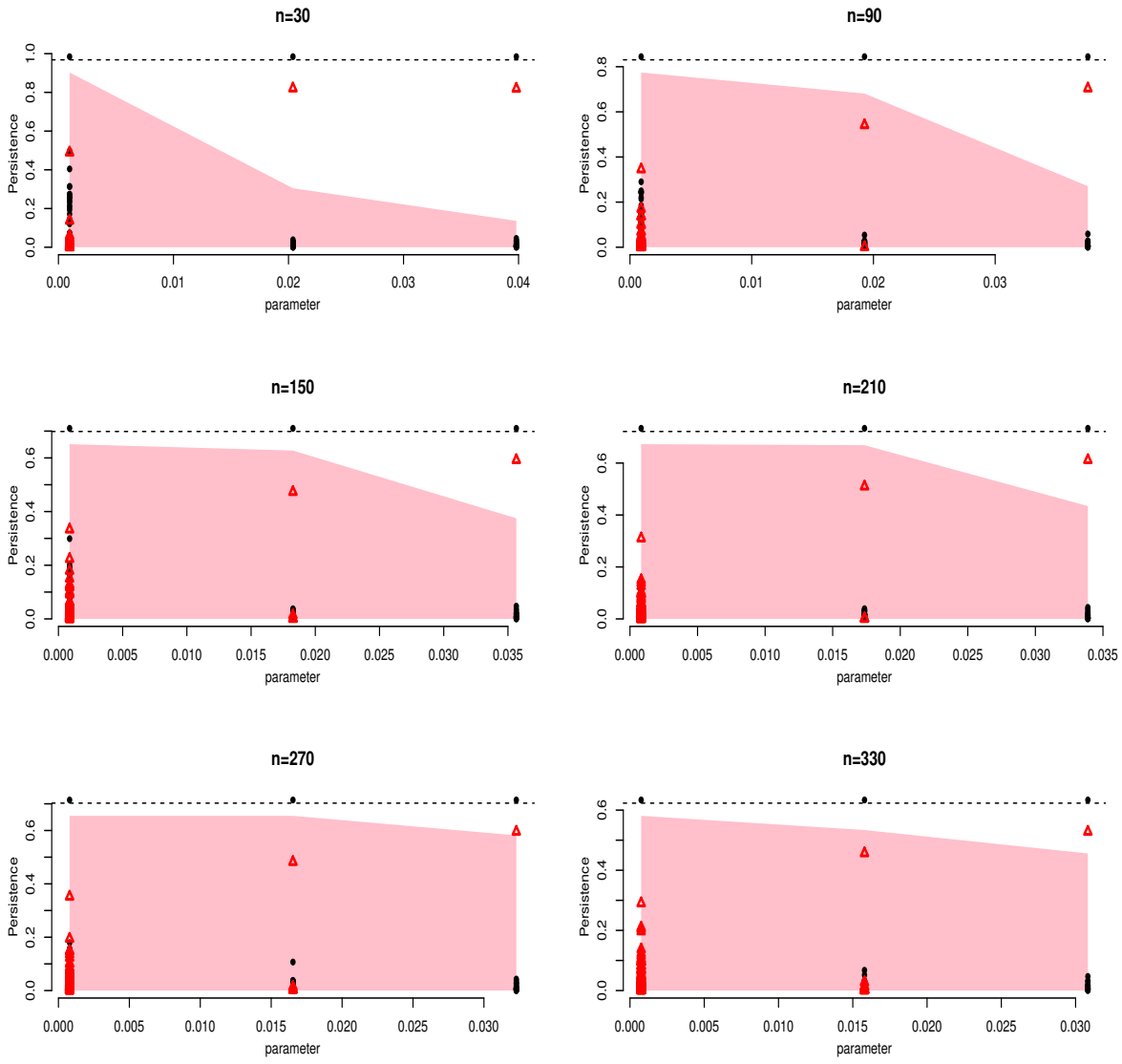

Figure 5: Adding $[30 : 30 : 330]$ samples from $U[-2, 2]^3$

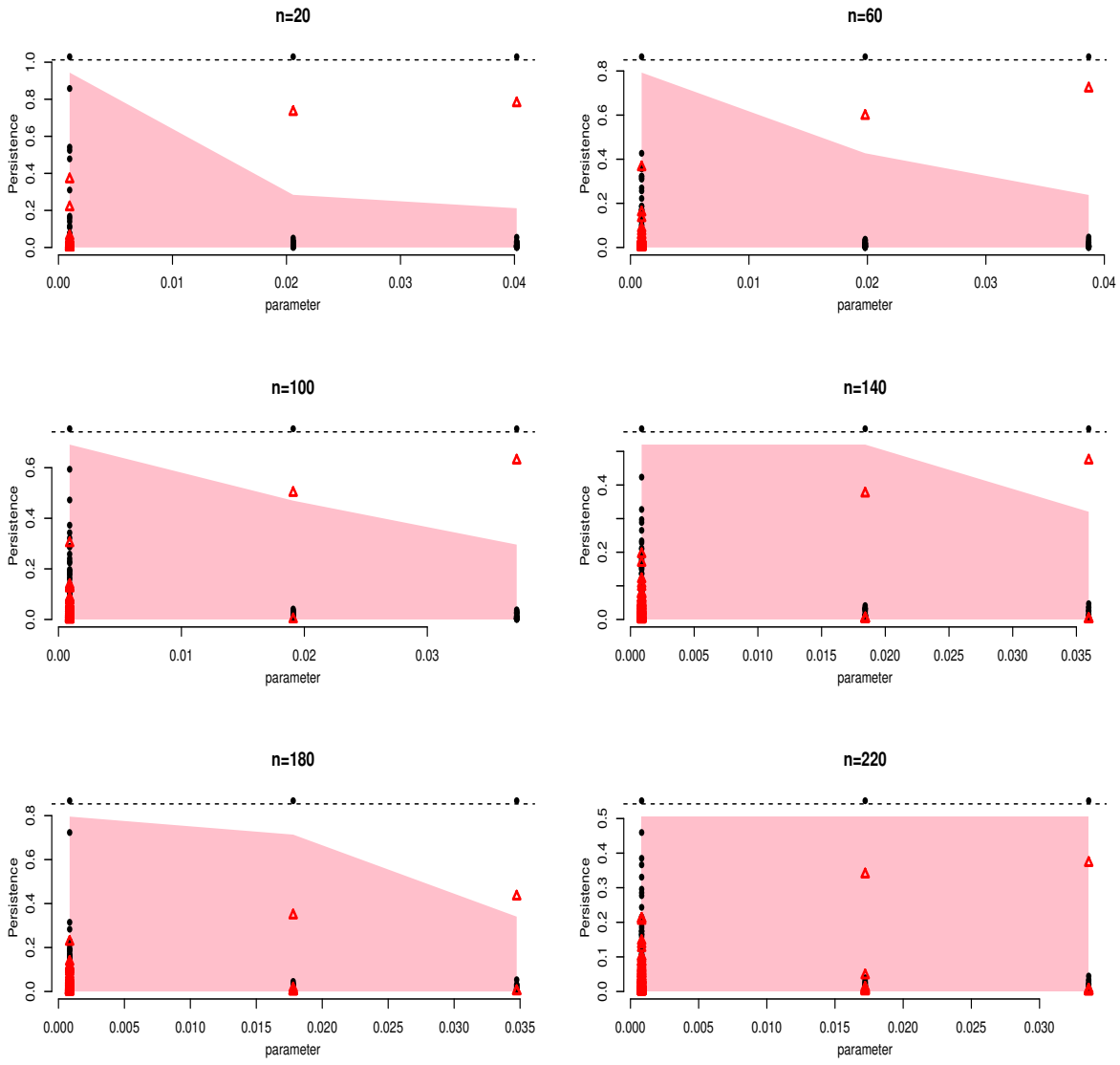

Figure 6: Adding $[50 : 100 : 550]$ samples from $\mathcal{N}(0, 1)$

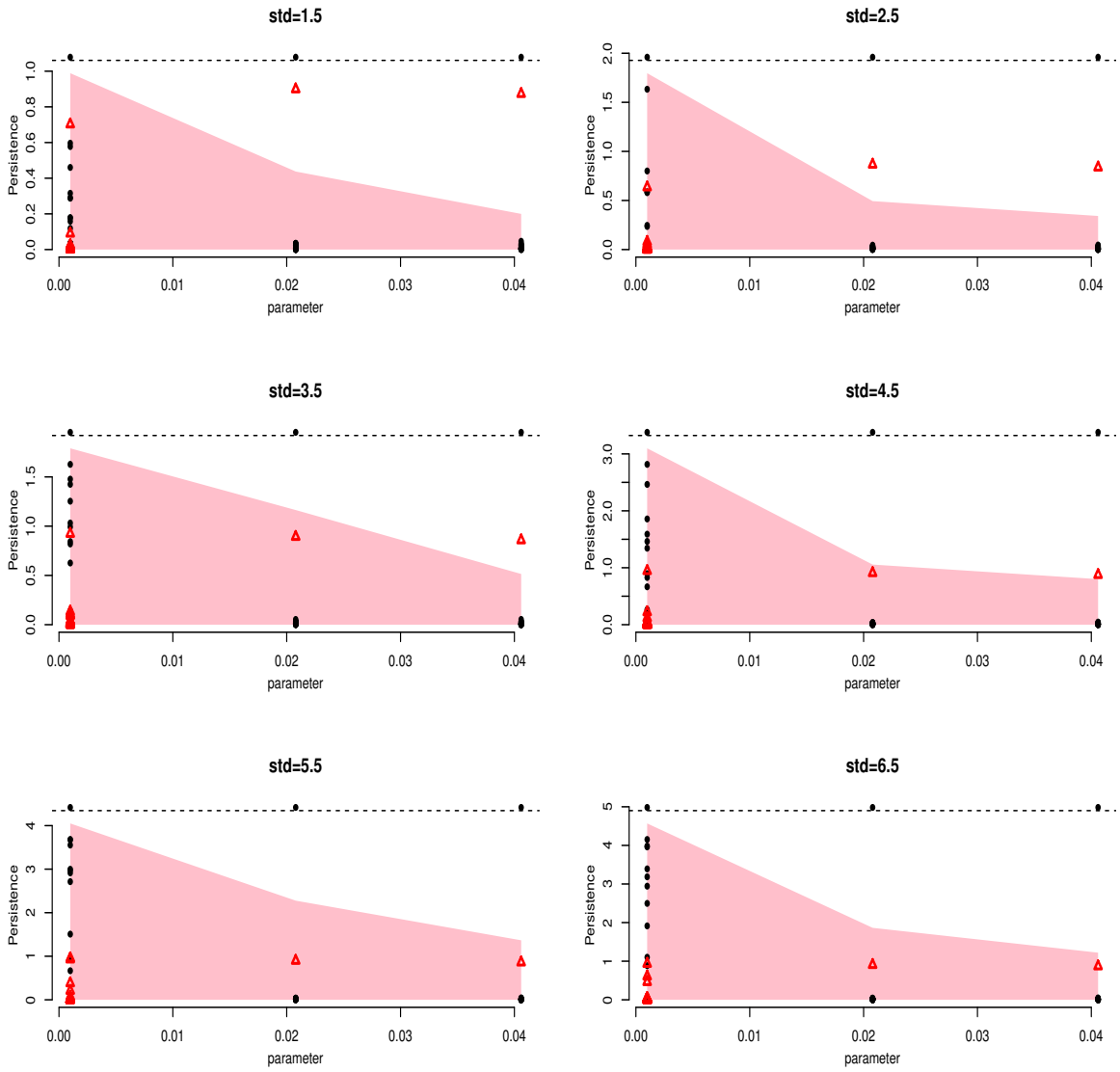

Figure 7: Adding 10 samples from $\mathcal{N}(0, [1.5 : 1 : 6.5])$

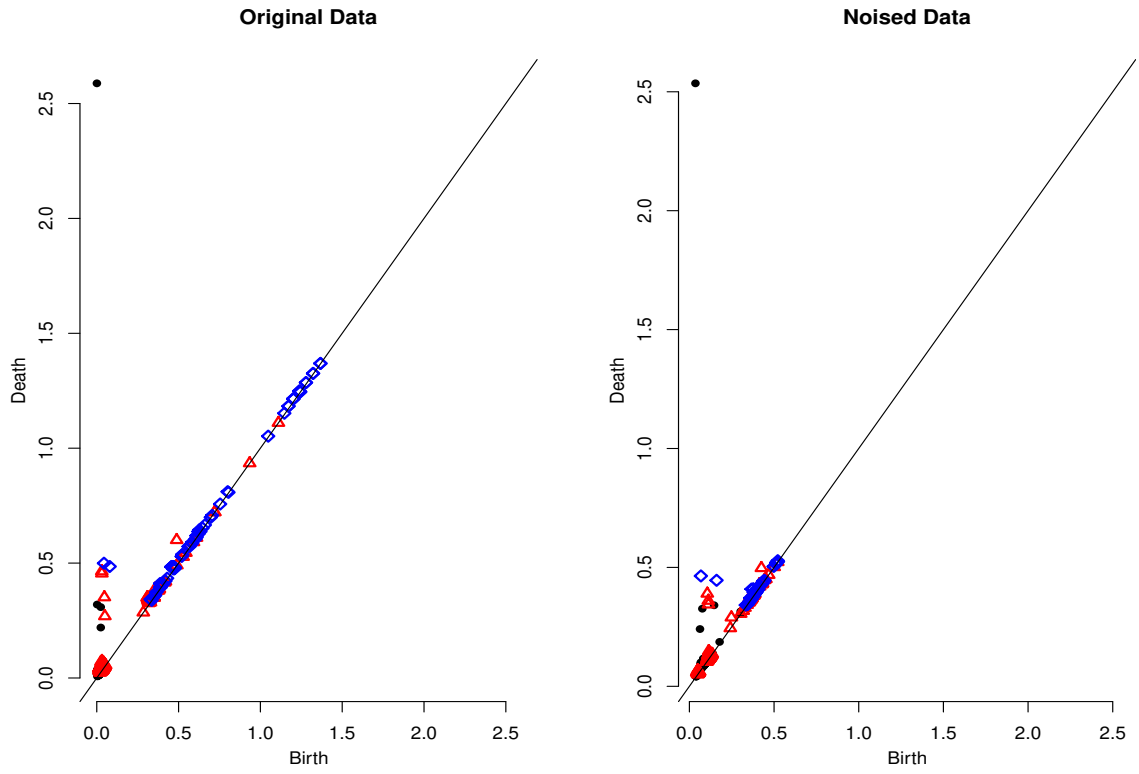

Figure 8: Disjoint shapes

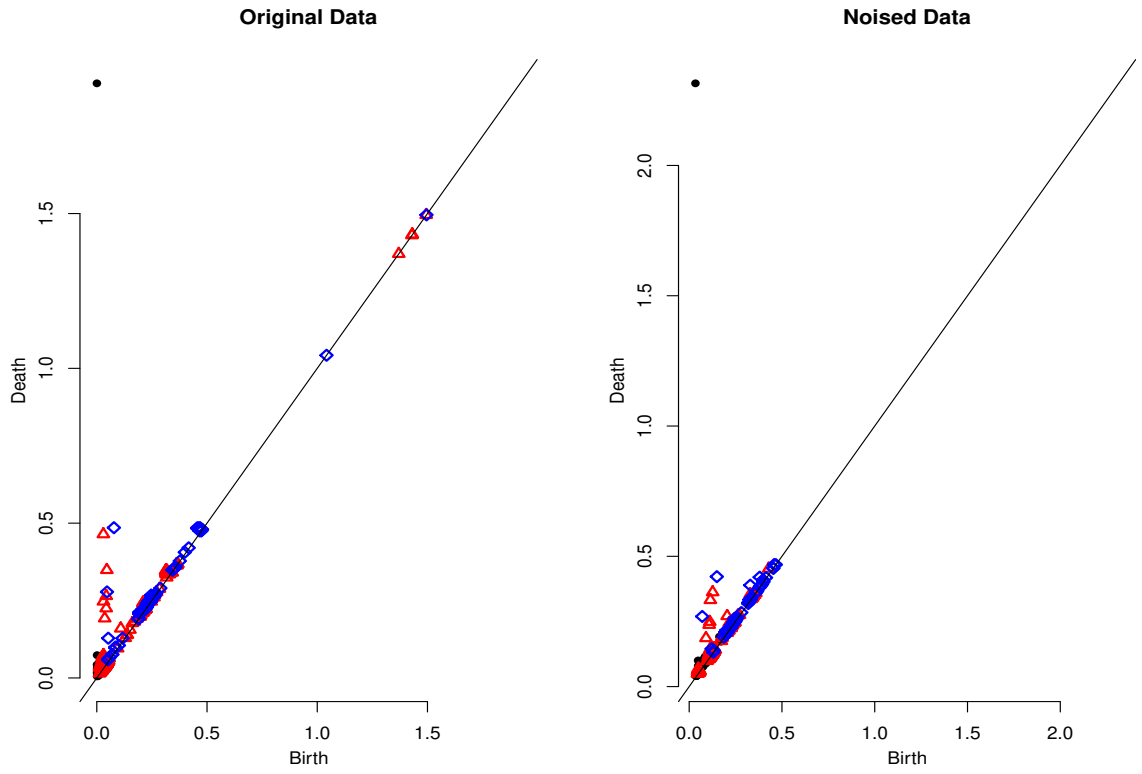

Figure 9: Joint shapes

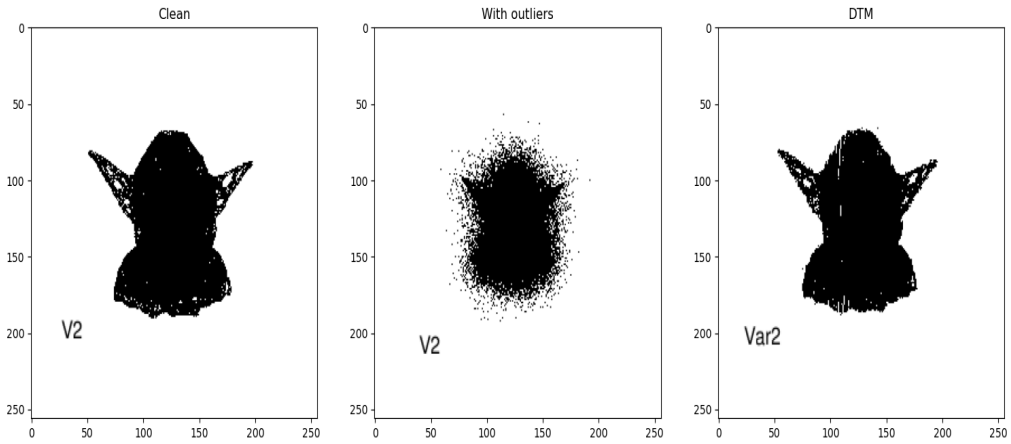

Figure 10: Reconstruction of Yoda from noisy version

- 0 vs 6
  Trained on 11841 exmaples
  Tested on 1938 examples
  k = 1
  0 mean = 3.65642875028104 — 6 mean = 1.29528674239364
  Classified correctly 1734 examples
  Not classified correctly 179 examples
  Not classified 25 examples
  k = 2
  0 mean = 3.25105078064842 — 6 mean = 0.924533413511397
  Classified correctly 1761 examples
  Not classified correctly 159 examples
  Not classified 18 examples
  k = 5
  0 mean = 3.19517496139027 — 6 mean = 0.87762722292563
  Classified correctly 1751 examples
  Not classified correctly 161 examples
  Not classified 26 examples
  k = 10
  0 mean = 3.01921602751145 — 6 mean = 0.741459643826337
  Classified correctly 1735 examples
  Not classified correctly 151 examples
  Not classified 52 examples

- 0 vs 9
  Trained on 11872 exmaples
  Tested on 1989 examples
  k = 1
  0 mean = 3.65642875028104 — 9 mean = 1.71784789743753
  Classified correctly 1700 examples
  Not classified correctly 282 examples
  Not classified 7 examples
  k = 2
  0 mean = 3.25105078064842 — 9 mean = 1.37118797421732
  Classified correctly 1720 examples
  Not classified correctly 263 examples
  Not classified 6 examples
  k = 5
  0 mean = 3.19517496139027 — 9 mean = 1.32001191988419
  Classified correctly 1718 examples
  Not classified correctly 264 examples
  Not classified 7 examples
  k = 10
  0 mean = 3.01921602751145 — 9 mean = 1.13703358223009
  Classified correctly 1738 examples
  Not classified correctly 236 examples
  Not classified 15 examples

- 6 vs 9
  Trained on 11867 exmaples

Tested on 1967 examples
k = 1
6 mean = 1.29528674239364 — 9 mean = 1.71784789743753
Classified correctly 1190 examples
Not classified correctly 745 examples
Not classified 32 examples
k = 2
6 mean = 0.924533413511397 — 9 mean = 1.37118797421732
Classified correctly 1241 examples
Not classified correctly 702 examples
Not classified 24 examples
k = 5
6 mean = 0.87762722292563 — 9 mean = 1.32001191988419
Classified correctly 1228 examples
Not classified correctly 706 examples
Not classified 33 examples
k = 10
6 mean = 0.741459643826337 — 9 mean = 1.13703358223009
Classified correctly 1202 examples
Not classified correctly 698 examples
Not classified 67 examples

As we can infer from the results DTM with k=5 works best on MNIST, while with k=10 we get poor results. For 0 vs 6/9 we get good results because the hole in zero is more significant (as we can see from the difference in the means).
Overall DTM with resonable amount of tuning outperforms the regular distance function.

Dataset taken from [6].

References

[1] Chazal, Frédéric & Cohen-Steiner, David & Mérigot, Quentin. (2011). Geometric Inference for Probability Measures. Foundations of Computational Mathematics. 11. 733-751. 10.1007/s10208-011-9098-0.

[2] Chazal, Frédéric & Fasy, Brittany & Lecci, Fabrizio & Michel, Bertrand & Rinaldo, Alessandro & Wasserman, Larry. (2014). Robust Topological Inference: Distance To a Measure and Kernel Distance. Journal of Machine Learning Research. 18.

[3] Fasy, Brittany Terese, Jisu Kim, Fabrizio Lecci and Clément Maria. Introduction to the R package TDA. ArXiv abs/1411.1830 (2014): n. pag.

[4] https://github.com/tomnorman/dtm

[5] Rakotosaona, Marie-Julie and La Barbera, Vittorio and Guerrero, Paul and Mitra, Niloy J and Ovsjanikov, Maks. PointCleanNet: Learning to Denoise and Remove Outliers from Dense Point Clouds. Computer Graphics Forum. 2019.

[6] https://www.kaggle.com/cristiangarcia/pointcloudmnist2d