


UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

add some nice picture and/or structures, uni logos

Topological Data Analysis in Python

organized by:
Michael Bleher, Maximilian Schmahl and Daniel Spitz

Heidelberg University

26^{th} - 28^{th} of October 2020

Contents

Contents

Programme

?

scikit-tda

scikit-tda Libraries

- Ripser.py
- Kepler Mapper
- Persim
- CechMate
- TaDAsets

copy scikit-tda slide, but insert schedule and highlight & encircle the parts and when we will cover them

Contents

Topology

Topology is the field of mathematics that studies shapes.

Topology is "blind" to continuous deformations.

add obligatory pictures and examples that give intuition of homotopy equivalence (interval, circle, sphere, punctured sphere, torus, coffee cup)

Königsberg Problem

"counting holes" as a way to distinguish topological spaces
$V - E + F = 2$ introduce Graphs for later convenience

Data

Data = points in $\mathbb{R}^n$

Challenges of data science: size, complexity, curse of dimensionality

Topology in Data?

points are not continuous!

$\mathbb{R}^n$ has no interesting topology!

$\rightarrow$ where is the topology?

Topology in Data

pictures like the ones from carlsson's talk

(regression, clusters, loops, bifurcations)

Topology in Data

noisy S^1, noisy pictures from MNIST or house numbers

$\rightarrow$ would like to have ways to extract these topological properties from the data

Topological Data Analysis

- A lossy compressed mathematical representation of a data set. You can study the global structure of a dataset, down to the details of a single data point, without incurring a cognitive overload.
- Resistance to noise and missing data. TDA retains significant features of the data.
- Invariance. Only connectedness matters. The skew, size, or orientation of data does not fundamentally change that data.
- A data exploration tool. Get answers to questions you haven't even asked yet.
- A methodology to study the shape of data and manifolds. TDA has a solid theoretical foundation and inherits functoriality.

Contents

Intro

Recall pictures from "The Shape of Data"

Königsberg Problem (revisited)

sth. about traveling salesman, distances, ...

such that the setup of Königsberg corresponds to a point cloud where the mapper network is interesting?

idea: prepare people for mapper with the following analogy

Bridges = ways to go from one part of the city to another Edges = overlap

Mapper

The Mapper Algorithm (show items iteratively)

1. Project(Filter dependency!)
2. Cover (Cover dependency!)
3. Clustering (metric dependeny!)
4. Graph
   - For each **cluster** draw a **node**
   - Whenever clusters **interesect** draw an **edge**
5. Prettify and Analyze

Pictures corresponding to the steps

Examples

some more examples to get a better feeling also hint at or state exercises

Contents

Kepler Mapper

Test

```
import kmapper

data = some_way_to_load_data()

kmapper(data, filter, bins)
```

kmapper example

first slide: short look at Data

second slide top: code second slide bottom: resulting graph

maybe several instances with different filter, cover, metric ...