CS 6170: Computational Topology, Spring 2019
Project 2
Topological Data Analysis for Data Scientists

Dr. Bei Wang

School of Computing
Scientific Computing and Imaging Institute (SCI)
University of Utah
www.sci.utah.edu/~beiwang
beiwang@sci.utah.edu

Feb 24, 2019

Project 2: Overview

Compare barcodes, TDA plus ML

- Posting date: 2/14/2019. Due date: 3/21/2019.
- Part 1: Compute and compare barcodes of 80 synthetic datasets using Ripser and Hera/TDA-R (10 points)
  - Specifically, choose 10 images each from 8 distinct classes
  - `https://github.com/Ripser/ripser`
  - `https://bitbucket.org/grey_narn/hera`
- Part 2: Using SVM and/or kernel-SVM in classifying the images into 4 and 8 classes respectively (10 points)
  - You may choose to use *sklearn* or any other ML packages
  - `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`
- Bonus Part: Using deep learning in classifying the images into 4 and 8 classes respectively (10 points)
  - Hofer et al. (2017): *Deep Learning with Topological Signatures*.
  - Read both the paper and its code repo `https://github.com/c-hofer/nips2017`.

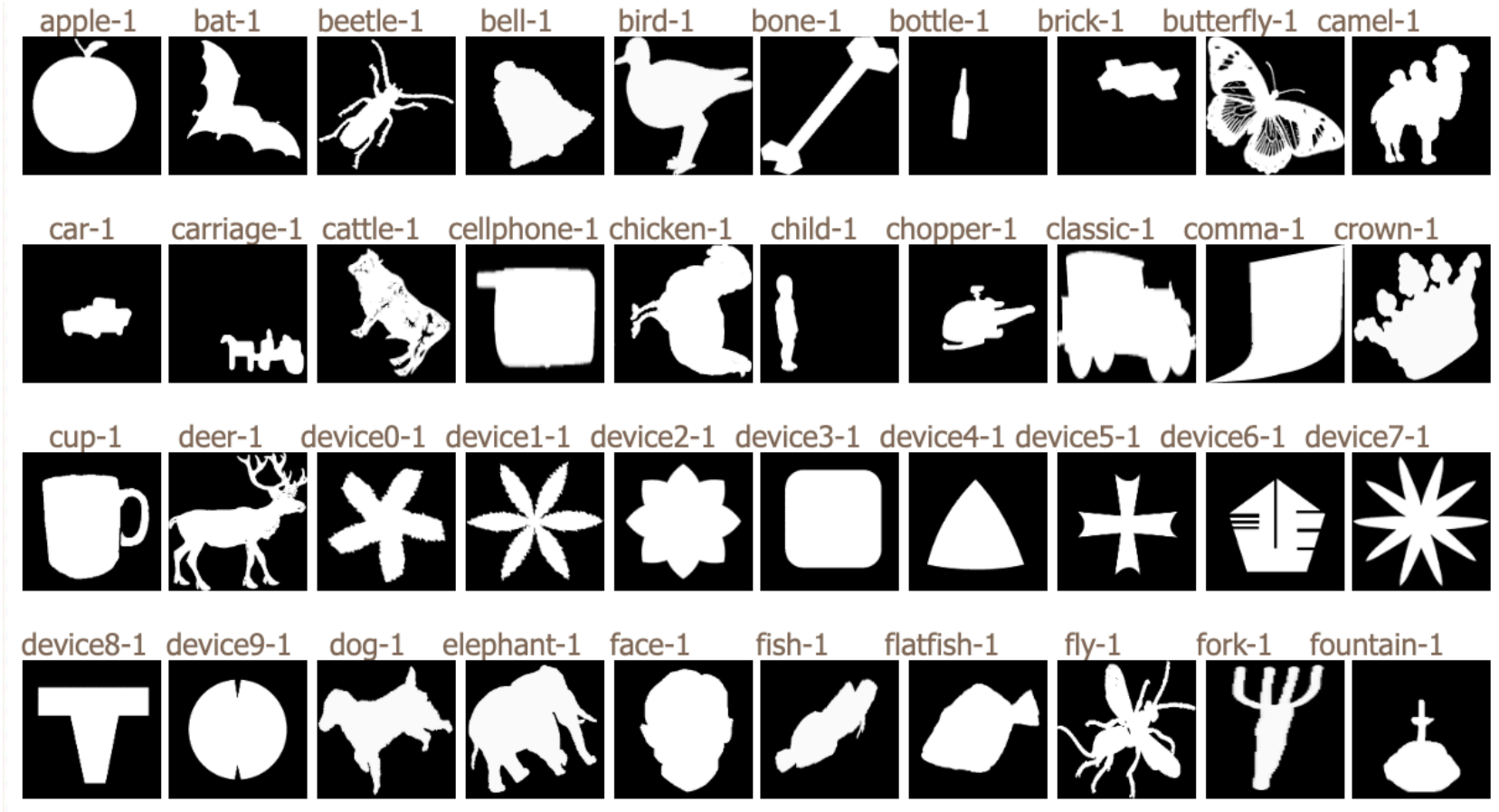
Submission

- Minimal guidelines are provided to simulate a real-world situation.
- The point cloud for each image should be recorded in a TXT file.
- The barcodes should be recorded in the form of TXT file, each line contains [birth, death) pair.
- Screen shots (PDF) and source code should be included as part of the submission as well.
- 2D Point clouds from the images (TXT), Barcodes (TXT), Screen shots (PDF), report (PDF) and source code should be submitted in a single ZIP file.

Project 2: Data

Image datasets

- MPEG-7 image data set
  `http://www.dabi.temple.edu/~shape/MPEG7/dataset.html`



Project 2 Part 1

Compute barcodes for MPEG-7 datasets: pre-processing

1. Take 10 sample images from 8 *distinct* classes of images from the MPEG-7 image dataset
2. Convert each image to a boundary point cloud, that is, extract points from the boundary of the images (you may choose to include/exclude interior points)
3. For each image, compute barcodes in dimension 0 and dimension 1, and return the barcodes as two separate TXT files, one for dimension 0, one for dimension 1 (2 points)

Compute and compare barcodes for MPEG-7 datasets

Step 1 and Step 2 each has a 0-dimensional and a 1-dimensional version.

1. Compute bottleneck distances between all pairs of barcodes, use MDS and t-SNE to project the space of barcodes onto the 2D plane, where each point in the projection represents the barcode (from a particular image). Color the points in the projects by the image class (3 points).
2. Compute Wasserstein distances between all pairs of barcodes, use MDS and t-SNE to project the space of barcodes onto the 2D plane, where each point in the projection represents the barcode (from a particular image). Color the points in the projects by the image class (3 points).
3. Using the raw images (or slightly processed images) as initial input, use MDS and t-SNE to project these images onto the 2D plane, where each point in the projection represents a particular image. Color the points in the projects by the image class. Compare the projection results with that of item 1 and item 2 (2 points).

Hints

1. `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html`
2. `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html`
3. Consider "precomputed" options.

Project 1 Part 2

Kernel-SVM for MPEG-7 image dataset

1. Use SVM or kernel-SVM in classifying the 80 images from Part 1 into 4 and 8 classes respectively.
2. Use persistence scale-space kernels for classification (5 points)
3. Use persistence images for classification (5 points)

Hints

1. You may choose to use *sklearn* or any other ML packages
2. `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`
3. Consider "precomputed" options.
4. `https://github.com/scikit-tda/scikit-tda`
5. `https://github.com/MathieuCarriere/sklearn_tda`

References I

Hofer, C., Kwitt, R., Niethammer, M., and Uhl, A. (2017). Deep learning with topological signatures. *Neural Information Processing Systems Conference (NIPS)*.