University of Utah
School of Computing

CS 6965 Project #1 Spring 2018

Due Feburary 1, 2018 at the start of class

Project 1 is the 1st mini project, with a total of 15 possible points (15% of the final grade).

Project description: Using KeplerMapper to explore various datasets.

1 Getting Started

To start, install KeplerMapper by the following information on:
`https://github.com/MLWave/kepler-mapper`.

It is recommended to obtain the source code:
git clone https://github.com/MLWave/kepler-mapper
cd kepler-mapper

In the source code folder, go to the *examples* folder, and see if you can get all examples to run.
For example, under the *examples/cat/* folder, run:
python cat.py

You will likely get an error as follows:

```
..Projecting data using: sum

..Scaling with: MinMaxScaler(copy=True, feature_range=(0, 1))

Traceback (most recent call last):
  File "cat.py", line 15, in <module>
    coverer=km.Cover(nr_cubes=15, overlap_perc=0.2))
AttributeError: 'module' object has no attribute 'Cover'
```

KeplerMapper is under active development, so it is not perfect. To get all examples to run, replace the following code in cat.py:

```
graph = mapper.map(lens,
                   data,
                   clusterer=sklearn.cluster.DBSCAN(eps=0.1, min_samples=5),
                   coverer=km.Cover(nr_cubes=15, overlap_perc=0.2))
```

by a modified version:

```
graph = mapper.map(lens,
                   data,
                   clusterer=sklearn.cluster.DBSCAN(eps=0.1, min_samples=5),
                   nr_cubes=15, overlap_perc=0.2)
```

Similar modifications apply to almost all example code.

2 The Cat Example (2 pts)

For the cat example, go to the project folder *code/cat/* and following the steps below.

2.1 Getting an initial cat example to run

Make sure you can run the following command successfully:
python p1_cat.py

Open *cat_keplermapper_output.html* with a Browser and play with various parameters in p1_cat.py to see how the results change.

- (1 pt) Change the mapper *interval overlap* parameter from 20% to 80%, rename the file as p1_cat_test1.py and observe its corresponding result. Answer the following question **Q1**: What is the effect of increasing interval overlap parameter on the final graph in the visualization?
- (1 pt) Modifying p1_cat_test1.py further by changing the mapper parameter that deals with the number of intervals (per dimension) from 15 to 30, rename the file as p1_cat_test2.py and observe its corresponding result. Answer the following question **Q2**: What is the effect of increasing number of interval parameter on the final graph in the visualization?

3 The Bunny Example (3 pts)

For the bunny example, go to the project folder *code/bunny/* and following the steps below.

The *bunny* folder contains a data file called *bunny.csv* and nothing else at the moment.

- (1 pt) Modify the p1_cat.py and rename it as p1_bunny_test1.py so that it analyzes the bunny dataset with the same parameter setting as the cat example. Answer the following question **Q3**: Why does the resulting graph contain two connected component?
- (1 pt) Replacing the line in p1_bunny_test1.py:

```
# Default lens: sum
lens = mapper.fit_transform(data)
```

by

```
lens = mapper.fit_transform(data, projection=[1])
```

Save the file as p1_bunny_test2.py and answer the following question **Q4**: What does this modification do to my data? (Hint, what lens am I using for my data after the modification?)

- (1 pt) Modify the file p1_bunny_test2.py and save it as p1_bunny_test3.py as follows. Replacing the mapper clustering method with K-means clustering from scikit-lean, with parameter n_clusters set to 2. (See `http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html` for details.) Answer the following question **Q5**: Does this modification helps better capture the shape of the data? If yes, how?

4 The Digits Example (5 pts)

For the digits example, go to the project folder *code/bunny/* and following the steps below.

- (1 pt) Run the code p1_digits.py twice. And answer the following question **Q6**: Why are the results not necessarily identical?

- (2 pt) Modify p1_digits.py and save it as p1_digits_test1.py such that it uses Spectral Embedding as part of the projection, with parameters, $n\_components = 2$, $random\_state = 0$, and $eigen\_solver$ equal to "arpack". Observe the results and answer the following question **Q7**: What is the difference between the resulting using Spectral Embedding in comparison to the results using t-SNE?

- (2 pt) Modify the parameters for Spectral Embedding in p1_digits_test1.py and save the file as p1_digits_test2.py so that the resulting clusters (digits) are better separated. Answer the following question **Q8**: What is your modification and its effect on the data? (Hint: consider modifying the dimension of the projected subspace).

5 Your Own Dataset (5 pts)

Go to the project folder *code/mydata/* and following the steps below. The folder is empty at the moment.

The last part of the project involving applying the Mapper framework to a dataset of your own. You could work with a 3D point cloud (similar to the setting of Cat and Bunny Examples); or you could work with a high-dimensional example (similar to the breast-cancer example).

Your point cloud data should have at least 200 points (if you are not sure, please speak with the instructor).

If you would like to work with a 3D point cloud data, check out the point cloud library (`http://www.pointclouds.org/news/2013/01/07/point-cloud-data-sets/`). Or check out the Mesh data set at (`http://people.csail.mit.edu/sumner/research/deftransfer/data.html`). In this case, you would need to convert the data to the cvs format.

For high-dimensional datasets, check out public data such as `http://www.wolframalpha.com/` or `https://github.com/caesar0301/awesome-public-datasets`.

- (2 pt) Preparing your data in csv format and save it as mydata.csv. The data should be cleaned and readily usable by KeplerMapper.

- (3 pt) Apply KeplerMapper to your dataset and give an description as what insights one might obtain from the results. Your code should be named as mydata.py. Your code should be able to run properly without error and give meaningful results. (If you are unsure, ask the instructor). There is no partial credit for a program that does not run.

Submission: Project 1 is to be submitted via Canvas. Please provide the following in a ZIP file (contains a PDF and another ZIP):

- Report: A PDF that answers the questions **Q1** to **Q8**. It should also contain description and explanation associated with your own dataset.
- Source code: A ZIP file that contains the following Python programs. Each program is expected to run properly. There is no partial credit if the program does not run or does not give the expected result.
  - Folder *cat*: p1_cat_test1.py, p1_cat_test2.py.
  - Folder *bunny*: p1_bunny_test1.py, p1_bunny_test2.py, p1_bunny_test3.py.
  - Folder *digits*: p1_digits_test1.py, p1_digits_test2.py.
  - Folder *mydata*: mydata.csv, mydata.py.