COSI 126A: Homework 1 - Skeleton

Substitution

For DBSCAN, please use the dataset 'rockets.csv' instead of 'anthill.csv'. This has more clearly defined shapes, as well introducing noise points. For Kernel/Spectral clustering, please use 'eye_dense.csv' instead of 'eye.csv'.

ClusterUtils

Provided for you is the basic framework for HW1. Your responsibility is to implement the execution of algorithms in the following:

1. KMeans.py
2. DBScan.py
3. ExternalValidator.py
4. InternalValidator.py
5. Spectral.py
6. KernelKM.py

There are two additional utility classes, SuperCluster.py, and ClusterPlotter.py, that are completed for you. You may benefit from taking a look at the superclass SuperCluster.py to better understand the structure, provided methods, and parameters.
Implemented for you are methods for reading data from .csv files, initializing the fitting of data, and saving processed data. The easiest way to complete the assignment is with the following pattern:

```
km = KMeans(init='random', n_clusters=3, csv_path='three_globs.csv')
```

This initializes the KMeans object with the given parameters. There are additional parameters that can be set that you can find in the code.

```
km.fit_from_csv()
```

This first reads data from the .csv, converting it to a pandas DataFrame, which is attached to the object as km.DF. An np.darray is attached to object as km.X It then executes your algorithm, updating the samples in DF and X with your labels (and centroids, if any).

```
df = km.fit_predict_from_csv()
```

This does the same as above, but also returns a pandas DataFrame.

```
km.show_plot()
```

This creates and displays a matplot plot of your processed data. Centroids are displayed and points are colored by cluster. It reduces high dimensions into 2D space if needed using PCA.

```
km.save_csv()
```

This saves km.DF to a .csv file in the directory of program execution. See 'sample_driver.py' for more example usage.

Internal Validation

Internal validation requires computing a list of clustering results. This can be plotted in 3D for CVNN, and with a line plot for the Silhouette Index. This class has its own utility methods for saving .csv results and plots. You may benefit from practicing with 'well_separated.csv' to test your code for internal validation.

Packaging

Please include your name and email in 'setup.py'. When you are finished, you may install and use the package in your system by running:

```
python setup.py install
```

Issues/GitHub

If you discover any issues with the code, please reach out to twillkens@brandeis.edu. The codebase will be maintained at https://github.com/twillkens/ClusterUtils. Updates will be posted if necessary.