Unsupervised Machine Learning with Python

Useful resources

Section 1: Introduction

Course Github site:

https://github.com/satishchandrareddy/UnsupervisedML

Wikipedia page for Machine Learning

https://en.wikipedia.org/wiki/Machine_learning

Wikipedia page for Unsupervised Learning:

https://en.wikipedia.org/wiki/Unsupervised_learning

Wikipedia page for cluster analysis:

https://en.wikipedia.org/wiki/Cluster_analysis

Website for Anaconda package which is a downloadable data science platform for Python:

https://www.anaconda.com/

Website for Anaconda documentation:

https://docs.anaconda.com/anaconda/user-guide/

Python website:

https://www.python.org/

Numpy, Matplotlib, Pandas, scikit-learn, IPython, and wordcloud package websites:

https://numpy.org/

https://matplotlib.org/

https://pandas.pydata.org/

https://scikit-learn.org/stable/

https://pypi.org/project/ipython/

https://pypi.org/project/wordcloud/

Section 2: Python Demos

Many examples and tutorials on numpy, matplotlib, pandas, and sklearn.

The following links to details about animation using matplotlib

https://matplotlib.org/stable/api/animation_api.html

Website for ffmpeg for creating mp4 files from matplotlib animations (this is optional):

https://ffmpeg.org/

This is the Youtube video I followed to install ffmpeg on my Windows 10 machine:

https://www.youtube.com/watch?v=a_KqycyErd8

scikit-learn page on datasets for clustering:

https://scikit-learn.org/stable/modules/clustering.html

Section 3: Review of Mathematical Concepts

Kaggle is a free website for Data Science Competitions. I believe that you have to register to be able to download datasets. (Registration is not required for this course. I have made necessary data available.)

https://www.kaggle.com/

University of California, Irvine, Machine Learning Repository is a free site (no registration required):

Citation: Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

https://archive.ics.uci.edu/ml/index.php

Wikipedia page for Computational Complexity:

https://en.wikipedia.org/wiki/Computational_complexity

Wikipedia page for Singular Value Decomposition:

https://en.wikipedia.org/wiki/Singular_value_decomposition

Wikipedia page for Covariance Matrices

https://en.wikipedia.org/wiki/Covariance_matrix

Section 4: Hierarchical Clustering

Wikipedia page for Hierarchical Clustering:

https://en.wikipedia.org/wiki/Hierarchical_clustering

Section 5: DBScan

Wikipedia page for DBSCAN:

https://en.wikipedia.org/wiki/DBSCAN

Wikipedia page for elbow method

https://en.wikipedia.org/wiki/Elbow_method_(clustering)

Section 6: K Means Clustering

Wikipedia page for K means clustering:

https://en.wikipedia.org/wiki/K-means_clustering

Wikipedia page for K means ++

https://en.wikipedia.org/wiki/K-means%2B%2B

Wikipedia page for elbow method

https://en.wikipedia.org/wiki/Elbow_method_(clustering)

Section 7: Gaussian Mixture Model

Wikipedia page for mixture models:

https://en.wikipedia.org/wiki/Mixture_model

I used formulas from the following document in the derivation of the Expectation Maximization algorithm for Gaussian Mixture Model in multiple dimensions

Kaare Brandt Petersen and Michael Syskind Pedersen, *The Matrix Cookbook*, Version November 15, 2012. https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf

In terms of regularization, here is a general reference:

https://en.wikipedia.org/wiki/Regularization_(mathematics)

Here is the documentation page for the sklearn Gaussian Mixture Model function. You can look at the covariance_type input for the 4 types "full", "tied", "diag", "spherical" and the reg_covar input to get a sense of what is done for regularization.

https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html

Section 8: Comparison of Methods

Silhouette coefficient:

https://en.wikipedia.org/wiki/Silhouette_(clustering)

Dunn Index:

https://en.wikipedia.org/wiki/Dunn_index

Davies-Bouldin Index:

https://en.wikipedia.org/wiki/Davies-Bouldin_index

You can see the sklearn version of comparison of methods at:

https://scikit-learn.org/stable/modules/clustering.html

Section 9: Principal Component Analysis

Wikipedia page for PCA:

https://en.wikipedia.org/wiki/Principal_component_analysis

Wikipedia page for autoencoders:

https://en.wikipedia.org/wiki/Autoencoder

The following website presents various tensorflow examples of autoencoders for dimension reduction for the MNIST digits dataset. In this example, "keras" is imported and "keras.layers.Dense" is used to set up a dense neural network layer. In the demo in Section 9.6, I import tensorflow as tf. One can set up the dense layer using "tf.keras.layer.Dense"

https://blog.keras.io/building-autoencoders-in-keras.html

Here is an example from the tensorflow website:

https://www.tensorflow.org/tutorials/generative/autoencoder

Section 10: Case Studies

For information about the Purity metric, go to the section on External Evaluation in

https://en.wikipedia.org/wiki/Cluster_analysis

Setosa, Versicolor, and Virginica figure citations and licenses:

Setosa:

https://en.wikipedia.org/wiki/Iris_flower_data_set#/media/File:Kosaciec_szczecinkowaty_Iris_setosa.jpg

CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=170298

https://commons.wikimedia.org/wiki/File:Iris_setosa01.jpg

Miya.m, CC BY-SA 3.0 <http://creativecommons.org/licenses/by-sa/3.0/>, via Wikimedia Commons

Versicolor:

https://commons.wikimedia.org/wiki/File:Iris_versicolor_3.jpg

No machine-readable author provided. Dlanglois assumed (based on copyright claims)., CC BY-SA 3.0 <http://creativecommons.org/licenses/by-sa/3.0/>, via Wikimedia Commons

Virginica:

https://commons.wikimedia.org/wiki/File:Iris_virginica.jpg

Frank Mayfield, CC BY-SA 2.0 <https://creativecommons.org/licenses/by-sa/2.0>, via Wikimedia Commons

Source for Iris Flower Dataset:

https://archive.ics.uci.edu/ml/datasets/iris

Source for MNIST Dataset:

http://yann.lecun.com/exdb/mnist/

Source for BBC Text Data:

See following link at Kaggle:

https://www.kaggle.com/yufengdev/bbc-fulltext-and-category

License: https://creativecommons.org/publicdomain/zero/1.0/

Section 11: Concluding Remarks and Thank You

Here are links to various packages:

scikit-learn package:

https://scikit-learn.org/stable/

Python package for Hierarchical Clustering:

https://pypi.org/project/fastcluster/

Python package for identifying elbow (called knee) of a curve:

https://pypi.org/project/kneed/

Python package for Gaussian Mixture Model:

https://pypi.org/project/gmr/

Python package for various clustering algorithms:

https://pypi.org/project/klusterpy/