Lab – Clustering #1

Data Mining, Spring 2017

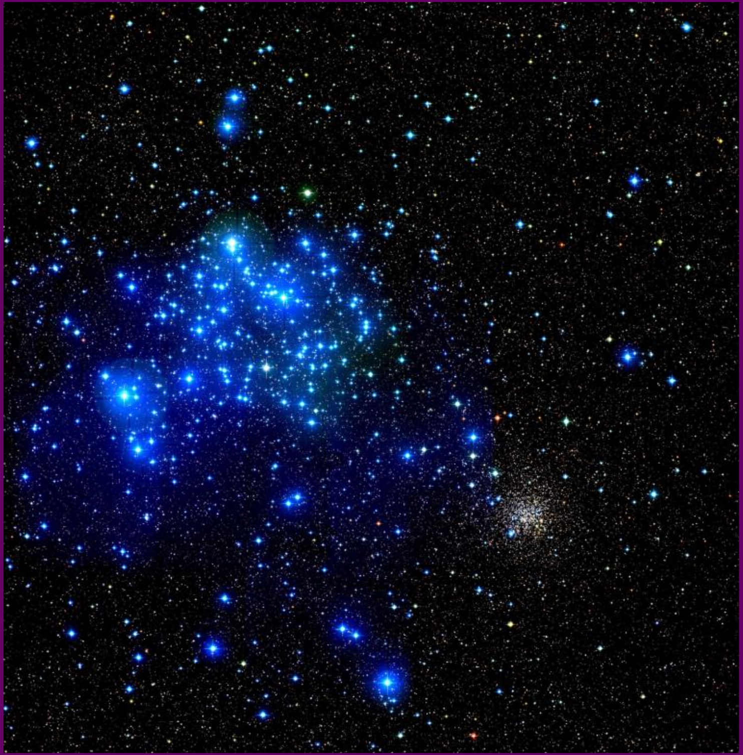
Mathias Löwe (malw@itu.dk) | Daniel Sørensen (dafr@itu.dk)

IT UNIVERSITY OF COPENHAGEN

Today's Lab: Clustering

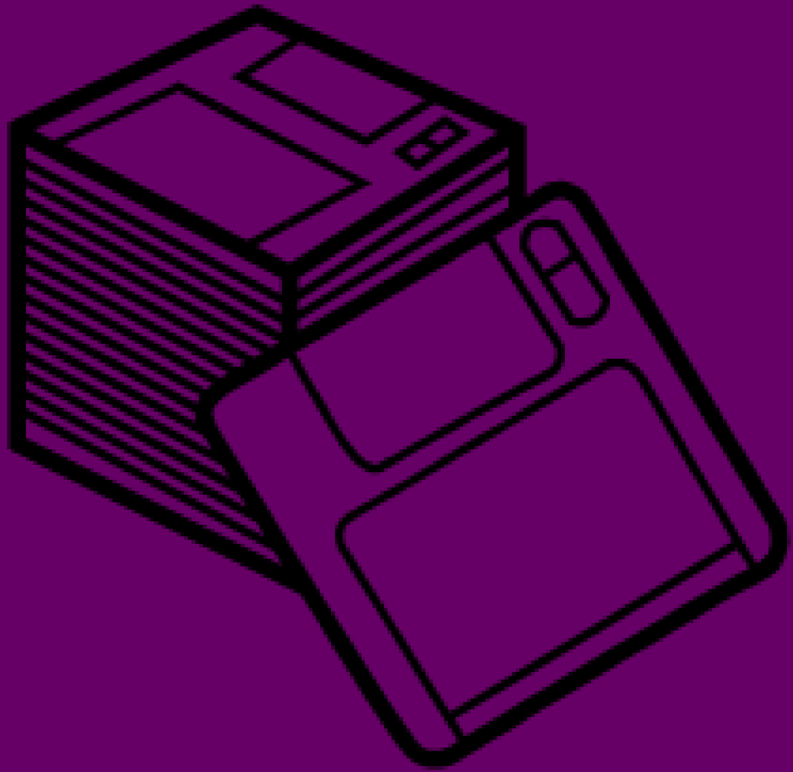
Clustering #1

- Today you will be implementing the k-means and k-medoids algorithms to cluster iris flowers.
- Code provided to help you load in the data and convert it to java-objects.
- More info:
  - K-means → Chapter 10.2.1 (pg. 451-454) in the book
  - Measuring distance between tuples → Chapter 2.4.4 (pg. 72) in the book
  - K-Medoids → Chapter 10.2.2 (pg. 454-457) in the book



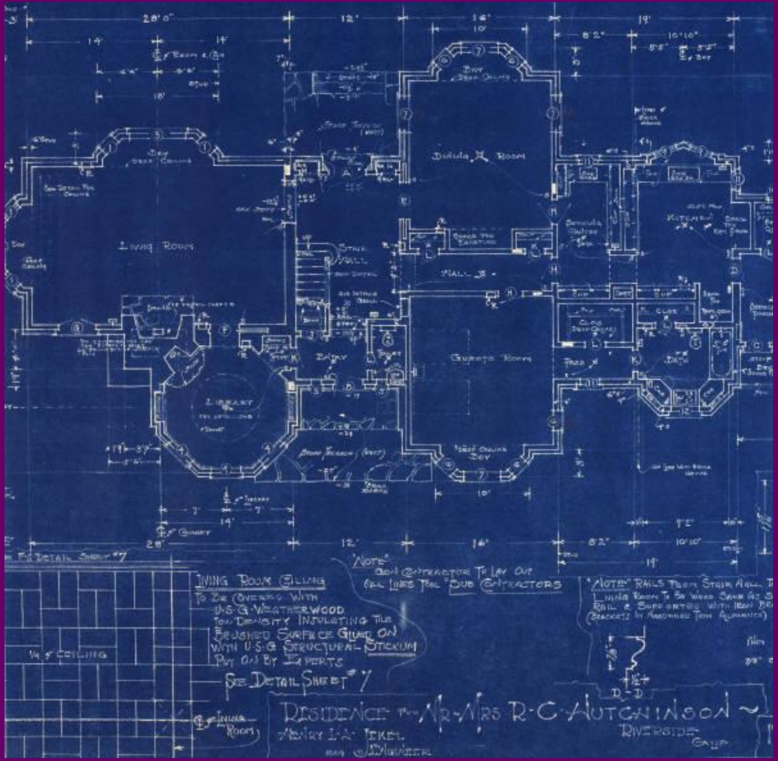
The Data

- The iris data can be found in the iris.csv file in the java-project.
- Attributes:
  - Sepal length
  - Sepal width
  - Petal length
  - Petal width
  - Class
- Possible values: Iris-setosa, Iris-versicolor and Iris-virginica



Lab Overview

- First take a look at the code provided.
- Then start working on implementing k-means/k-medoids
  - Only do clustering based on the numerical attributes.
  - Then when you have finished clustering use the nominal attribute (Class) as a focal point to see how well your clustering managed to do.
  - K = 3 (at first at least)
  - K-Means is the simplest of the two, and will require less time to implement compared to k-Medoids



Code Provided

- Iris class used to store data for each Iris flower in data.
- Data loading and conversion to Iris-objects
  - Done by the CSVFileReader and DataLoader class.
- Two Cluster classes contains some bare bone code to help you get started implementing your own clusters.
- KMeans-class has the method where you should implement k-means
- Kmedoid-class has the method where you should implement k-medoids
- Main-class contains Main-function
  - Currently it calls the LoadData method of the DataLoader which returns an ArrayList of all Iris objects loaded in from the data file.
  - It then calls the static method KMeansPartition of the Kmeans-class.
  - Finally it calls the static method KMedoidPartition of the KMedoid-class

Thanks for listening!