3253 Machine Learning

Data Science Fundamentals Certificate

Module 4

CLUSTERING & UNSUPERVISED LEARNING

Course Roadmap

Module / Week	Title
1	Introduction to Machine Learning
2	End to End Machine Learning Project
3	Classification
4	Clustering & Un-Supervised Learning
5	Training Models & Feature Selection
6	Support Vector Machines
7	Decision Trees, Ensemble Learning & Random Forests
8	Dimensionality Reduction
9	Introduction to TensorFlow and Neural Networks
10	Training Deep NNs
11	Distributing TensorFlow and Other Architectures
12	External Speakers and Students Presentations

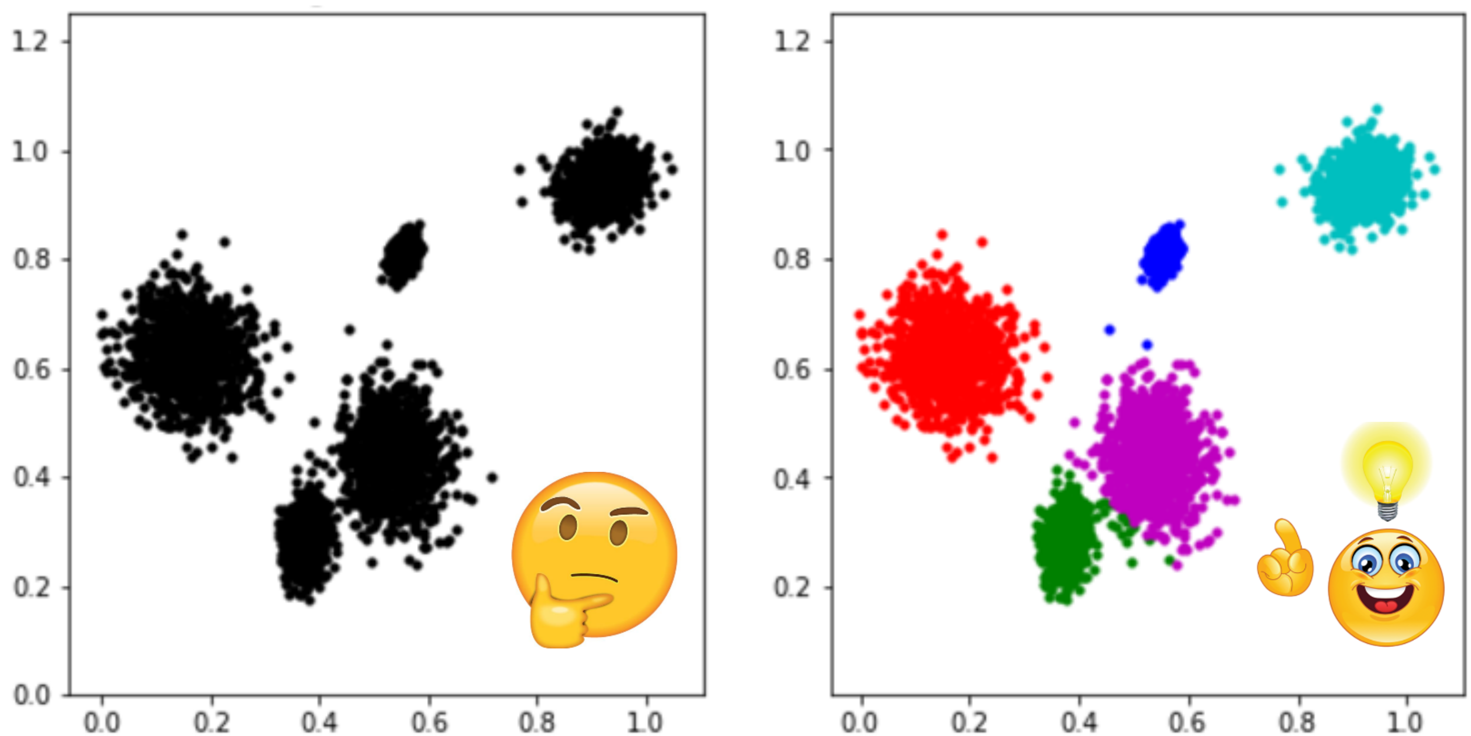
Module 4: Learning Objectives

- Define Unsupervised Learning
- Clustering: ideas and objective
- Clustering algorithms: k-means, agglomerative, DBSCAN
- Performance measures

CLUSTERING AND UNSUPERVISED LEARNING

Clustering goal

- The aim is to group points (examples) into a small number of clusters



Clustering goal

- Similar examples should go to a same cluster; while different examples should be in different clusters
- There are many different clustering methods
- The clustering algorithm also learns how to assign a cluster to an example seen later
- Applications:
  - automatic topic detection of documents
  - customer segmentation
  - variable selection

Supervised VS Unsupervised Learning

- Algorithms used to build classifiers need supervised data examples
- The input data to the learner consists of examples $(x_1, y_1), \dots (x_n, y_n)$
- An example (x_i, y_i) shows the correct response y_i to the input x_i
- In unsupervised ML the learner does not have labels, only examples $x_1, \dots, x_n$

Unsupervised Learning

- A clustering algorithm will still produce an output $C(x) = c$ given an input x
- However, there is no way to know if the output is correct or not
- The learning algorithm does not optimize a cost function based on labels
- But some classification algorithms do optimize a cost function based on the input examples $x_1, \ldots, x_n$

Unsupervised Algorithms

- Tasks to consider:
  - Reduce dimensionality
  - Find clusters
  - Model data density
  - Find hidden causes

- Key utility
  - Compress data
  - Detect outliers
  - Facilitate other learning

Unsupervised Algorithms

- Approaches in unsupervised learning fall into three classes:
  - Dimensionality reduction: represent each input case using a small number of variables (e.g., principal components analysis, factor analysis, independent components analysis)
  - Clustering: represent each input case using a prototype example (e.g.,k-means, mixture models)
  - Density estimation: estimating the probability distribution over the data space

Clustering Algorithms

- Input: n vectors, m-dimensional, represent the objects to be clustered:

- Can start with object themselves (e.g. documents), but need a vector representation

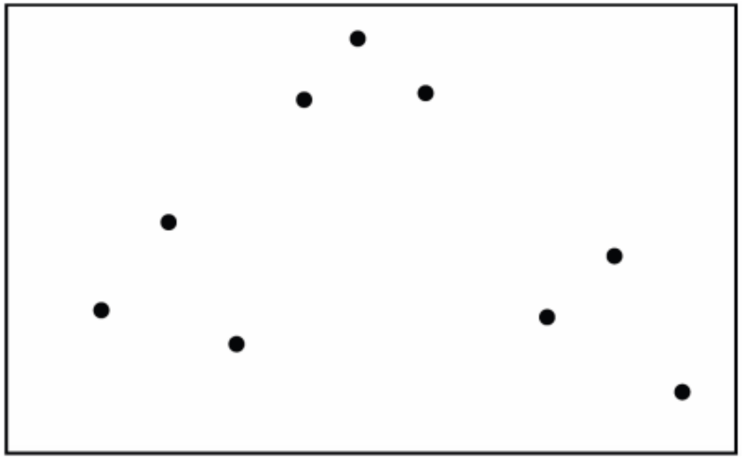
  Document → vector of word counts
- Vectors have same (fix length) but clustering can be done over sequences of different length (the matrix of distances is needed)

Clustering

- Motivation: prediction; lossy compression; outlier detection
- We assume that the data was generated from a number of different classes. The aim is to cluster data from the same class together.
  - How many classes?
  - Why not put each datapoint into a separate class?
  - What is the objective function that is optimized by sensible clustering?

Clustering

- Assume the data {x(1), . . . , x(N)} lives in a Euclidean space, $x(n) \in Rd$
- Assume the data belongs to K classes (patterns)
- How can we identify those classes (data points that belong to each class)?

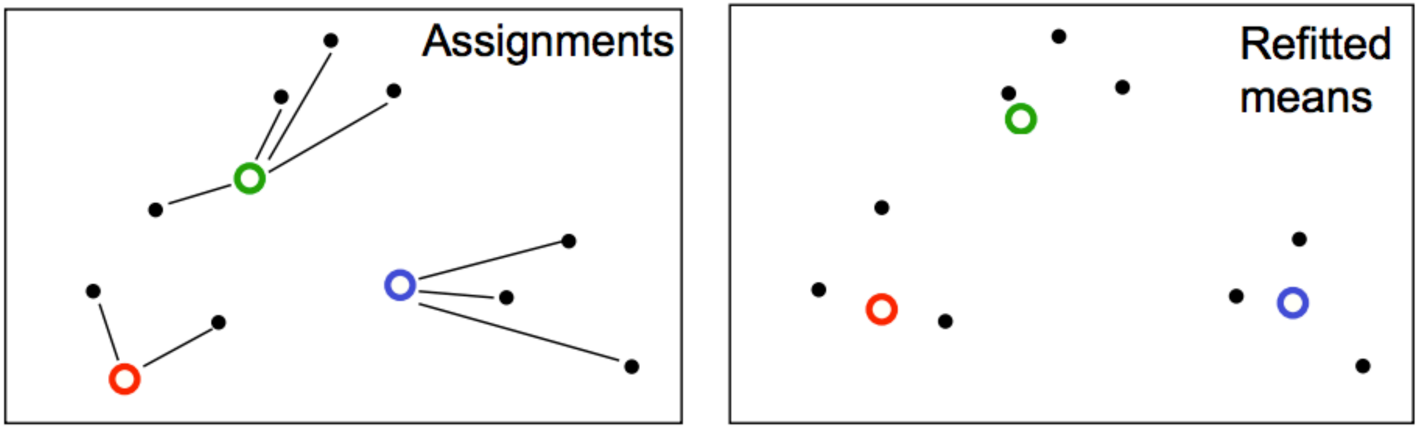


k-means algorithm

- Input: vectors $S = \{x^{(1)}, \ldots, x^{(n)}\}$
  $k =$ number of desired clusters
- Output: a partition of S into k clusters, and the clusters' average (centroid)
- Goal: $S_1, \ldots, S_k$ should minimize the square distances between each example x_i and its closest centroid $c(x_i)$: $\sum_{j=1}^{n} ||x_i - c(x_i)||^2$
- Lloyd's algorithm finds (a good enough) solution

k-means

- Initialization: randomly initialize cluster centers
- The algorithm iteratively alternates between two steps:
  - Assignment step: Assign each data point to the closest cluster
  - Refitting step: Move each cluster center to the center of gravity of the data assigned to it



K-means

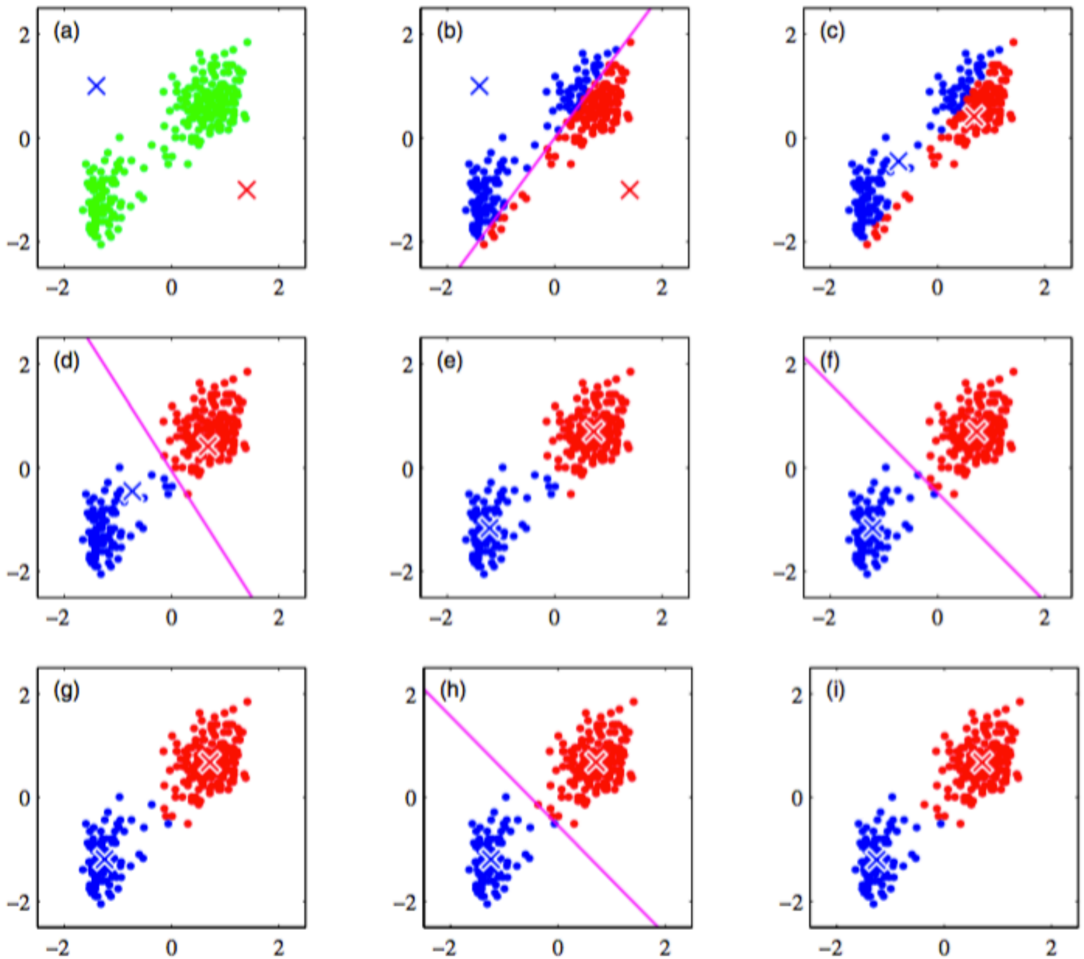


Figure 9.1 Bishop

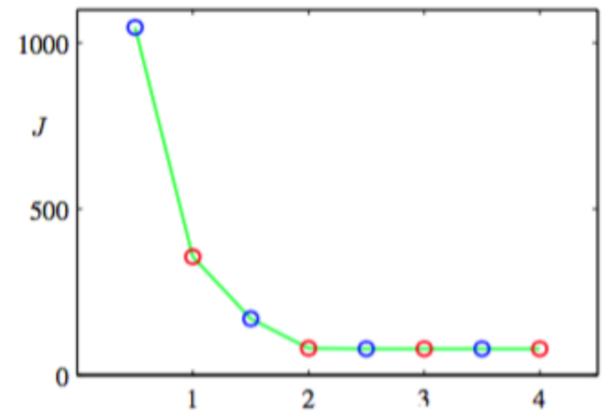


Figure 9.2 Bishop

k-means algorithm

- Steps:
  - 0) Start with a set of k centroids (random points from S)
  - 1) Assign each point to the centroid to which it is closest: this defines clusters
  - 2) Update the centroids as the mean within each cluster
  - 3) Repeat (1) and (2) until the centroids change is very small (threshold)

http://syskall.com/kmeans.js/ http://shabal.in/visuals/kmeans/2.html

k-means optimization

Find cluster centers m and assignments r to minimize the sum of squared distances of data points $\{x^{(n)}\}$ to their assigned cluster centers

$$\min_{\{\mathbf{m}\},\{\mathbf{r}\}} J(\{\mathbf{m}\},\{\mathbf{r}\}) = \min_{\{\mathbf{m}\},\{\mathbf{r}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

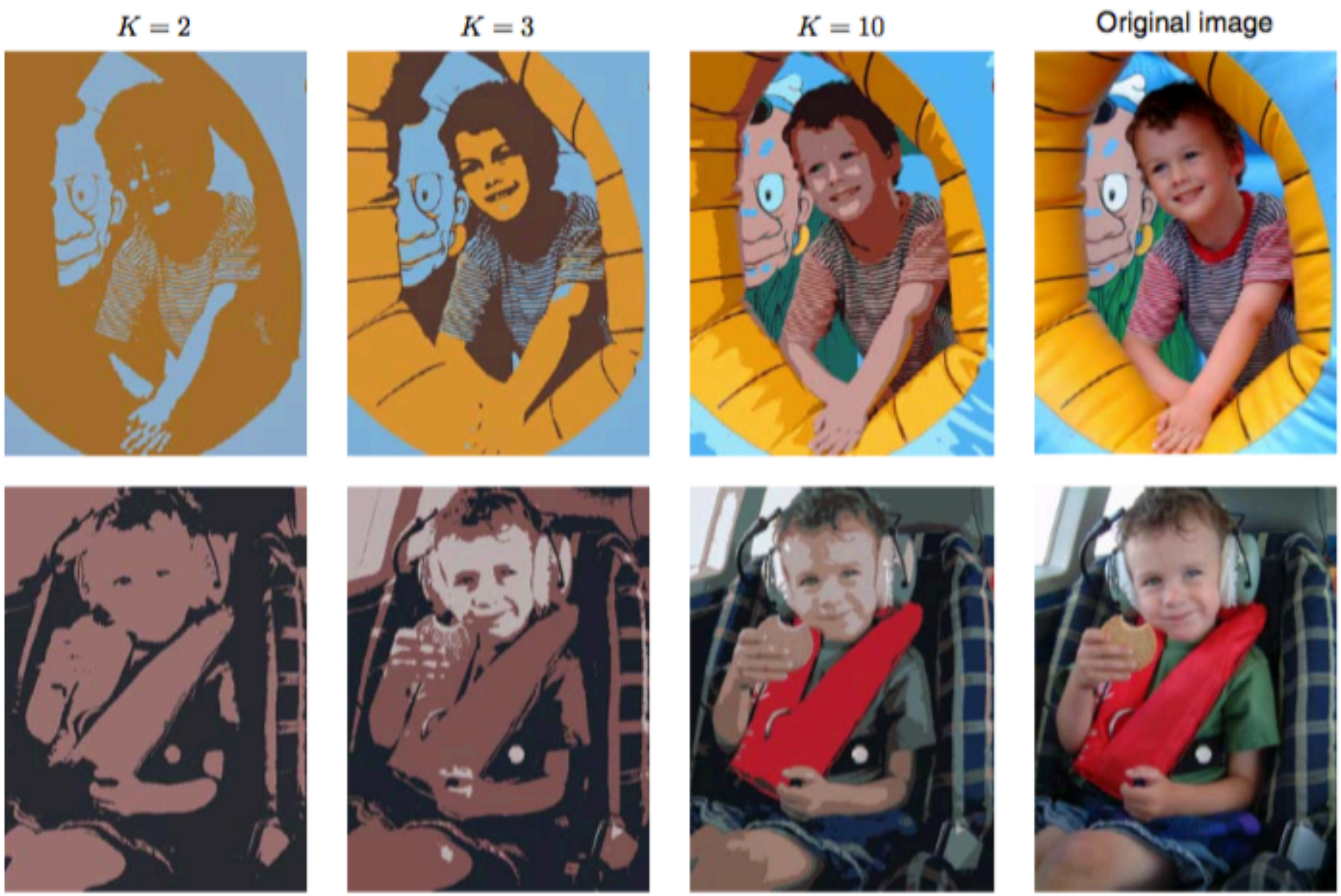
$$\text{s.t.} \sum_k r_k^{(n)} = 1, \forall n, \quad \text{where} \quad r_k^{(n)} \in \{0,1\}, \forall k, n$$

where $r_k^{(n)} = 1$ means that $x^{(n)}$ is assigned to cluster k (with center m_k)

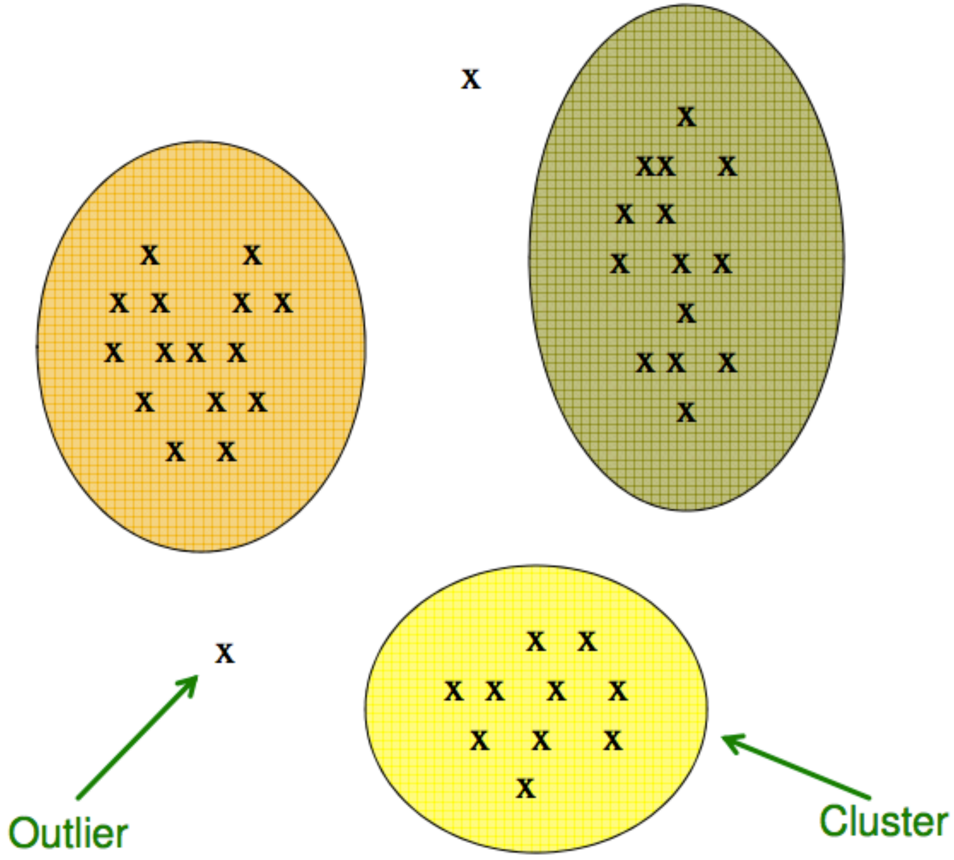
k-means algorithm

- k is a hyper-parameter: input to the algorithm. User species it
- Sometimes the value for k is known for the application (e.g., the goal is to find 5 segments)
- The value of k can be data-driven:
  - inertia:
  - inertia/inertia2
  - silhouette

k-means for image segmentation



Clustering and Outliers



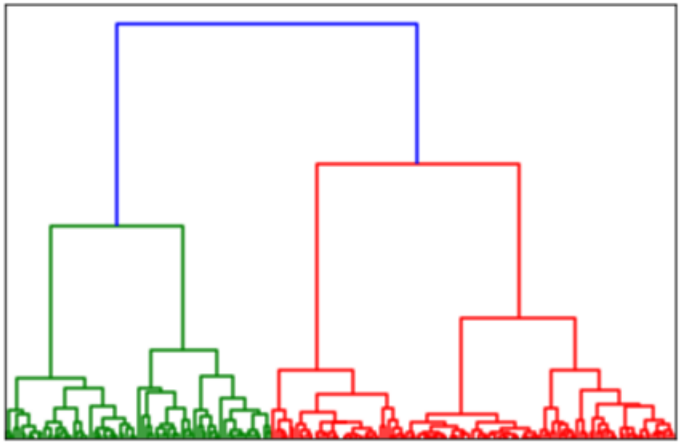
J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

k-mean challenges

- High-dimensional spaces look different:
  - Almost all pairs of points are at about the same distance
- There is nothing to prevent k-means getting stuck at local minima.

Hierarchical Clustering

- A bottom-up hierarchical clustering starts with as many clusters as points, and merge them iteratively
- Steps:
  0) Make of each data point a distinct cluster
  1) Find the two closest clusters and merge them
  2) Repeat (1) until all points below to one single clu



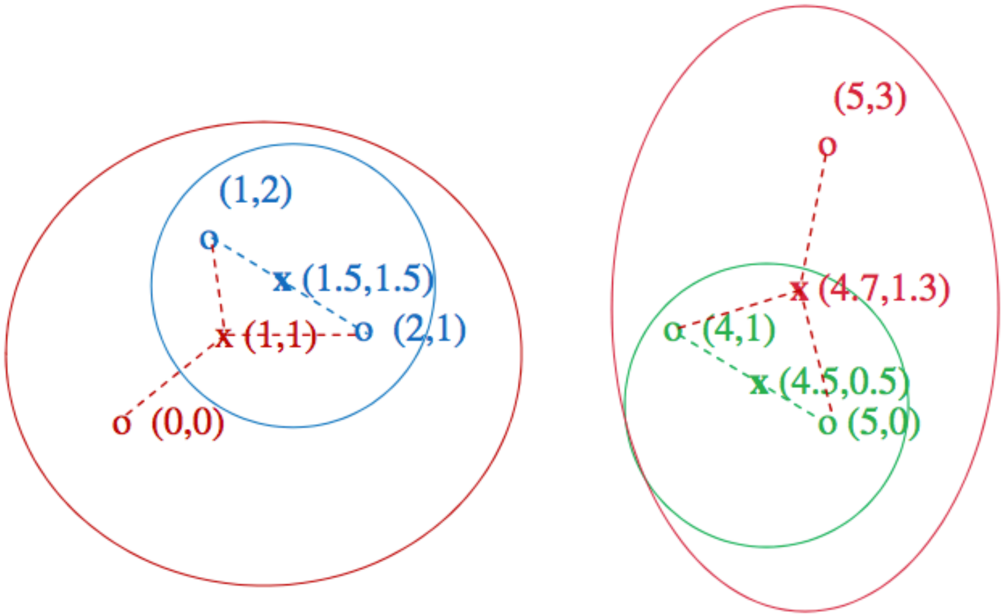
Hierarchical Clustering

- Key operation: Repeatedly combine two nearest clusters
- How to represent a cluster of many points?
  - Key problem: As you merge clusters, how do you represent the "location" of each cluster, to tell which pair of clusters is closest?
  - Euclidean case: each cluster has a centroid = average of its (data)points
- How to determine "nearness" of clusters?
  - Measure cluster distances by distances of centroids

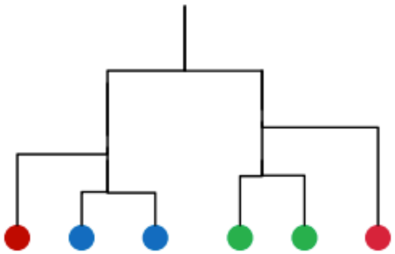
Hierarchical Clustering

- There are different ways to determine the 2 clusters that are joined in each step:
  - *ward*: minimize variance
  - average: minimize average distance between every pair of points (one in each cluster)
  - complete: minimize maximum distance between a pair of points, one in each cluster
- The user decides the number of clusters to use

Hierarchical Clustering Example



Data:
o … data point
x … centroid



Dendrogram

J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

DBSCAN clustering

- k-means clusters tend to be delimited by convex regions

- Both k-means and hierarchical clusters assign a cluster to every point

  outlier are forced to belong to a cluster

DBSCAN clustering

- DBSCAN is an algorithm that allows:
  - clusters with non-convex shapes
  - outlier detection
- Other algorithms allow non-convex shaped clusters:
  - agglomerative with ward linkage
  - spectral clustering
- Demo https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

DBSCAN clustering

- Parameters:
  min_samples (non-negative integer),
  epsilon (positive number)
- A core point is a point that has at least *min_samples* points within *epsilon* distance
- Core points are determined first
- Core points belonging to a cluster are computed iteratively:
  - take a core point
  - find all core points within *epsilon* distance
  - repeat until no more core points exists within epsilon
  - continue creating other clusters until no core points exists
- Non-core points:
  - Add to each cluster non-core points within epsilon distance from a core point
- A point that do not belong to any cluster are outliers
- Note that the number of clusters is not decided by the user

Clustering and Feature Selection

- An important part of building models is feature selection
- Many variables could be available to predict a target, but many of them could carry no information about the target
- There are many method for feature selection: univariate methods, regularization, feature importance, etc.
- Clustering the features (columns, instead of rows) is a way to reduce the dimensionality by picking a representative on each cluster
- Python Scikit-Learn provides this with FeatureAgglomeration

Resources

- http://scikit-learn.org/stable/modules/clustering.html
- Data Science from Scratch, Joel Grus
- An Introduction to Statistical Learning, James, G.; Witten, D.; Hastie, T.; Tibshirani, R

Homework

- Complete the notebook in the assignments section for this week
- Submit your solution here
  - https://goo.gl/forms/F5ytppo5KWnCqkt62
  - Rename your notebook to
    - W4_LastName_UTORid.ipynb
    - Example W4_Benitez_q212131.ipynb

Next Class

- Training Models and Features Selection
- Reading Hands-on ML (Chapter 4)