CLUSTERING

LEARNING OBJECTIVES

- Supervised vs unsupervised algorithms
- Understand and apply k-means clustering
- Density-based clustering: DBSCAN
- Silhouette Metric

UNSUPERVISED LEARNING

There are two main categories of machine learning: **supervised learning** and **unsupervised learning**.

Unsupervised learning:

- Extracting structure from data
- Example: segment grocery store shoppers into “clusters” that exhibit similar behaviors
- Goal is “representation”

UNSUPERVISED LEARNING

- So far all the algorithms we have used are *supervised*: each observation (row of data) came with one or more *labels*, either *categorical variables* (classes) or *measurements* (regression)
- **Unsupervised learning** has a different goal: **feature discovery**
- **Clustering** is a common and fundamental example of unsupervised learning
- **Clustering** algorithms try to find meaningful groups within data

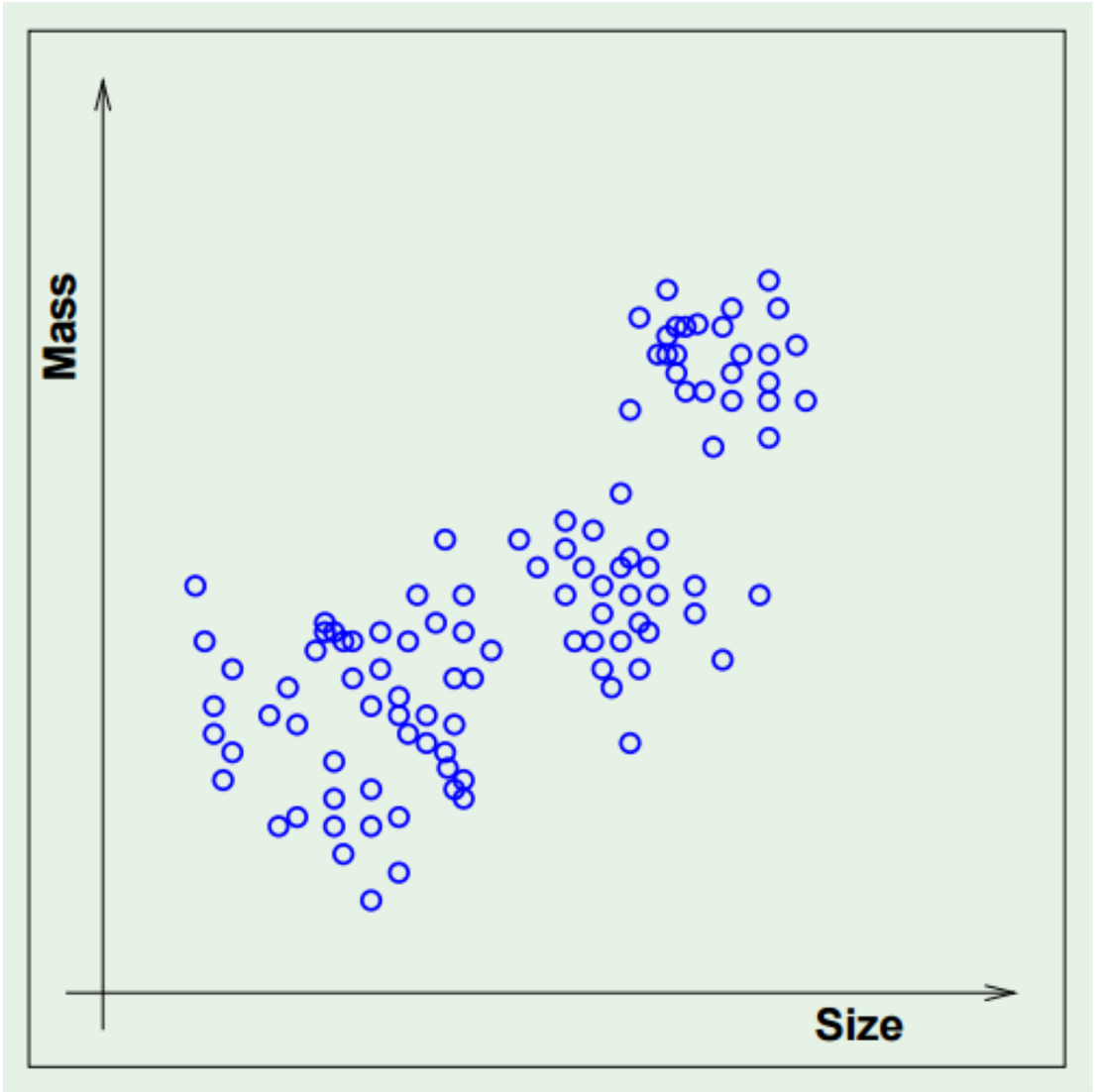
Unsupervised learning has some clear differences from supervised learning. With **unsupervised learning:**

- There is no clear objective
- There is no “right anwser” (hard to tell how well you are doing)
- There is no response variable, just observations with features
- Labeled data is not required

Unsupervised learning example: Coin clustering

- Observations: Coins
- Features: Size and mass
- Response: There isn't one (no hand-labeling required!)

1. Perform **unsupervised learning**
   - Cluster the coins based on "similarity"
   - You're done!


Mass
Size

Common types of unsupervised learning:

- **Clustering:** group “similar” data points together
- **Dimensionality Reduction:** reduce the dimensionality of a dataset by extracting features that capture most of the variance in the data
  - Decision Trees
  - Principal Component Analysis

CLUSTERING

INTRO TO CLUSTERING

Clustering is the task of dividing the population (our data) into a number of groups such that data points within each group are more similar to its own group than other groups. Simply put – the goal is to segregate groups with similar traits into clusters. We'll talk today about three different models.

Centroid Models - iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters. In these models, the no. of clusters required at the end have to be mentioned beforehand, which makes it important to have prior knowledge of the dataset. These models run iteratively to find the local optima.

Hierarchical Models - clustering algorithm that builds hierarchy of clusters. This algorithm starts with all the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.

Density Models: These models search the data space for areas of varied density of data points in the data space. It isolates various different density regions and assign the data points within these regions in the same cluster. Popular examples of density models are DBSCAN and OPTICS.

INTRO TO CLUSTERING

Clustering has a large no. of applications spread across various domains. Some of the most popular applications of clustering are:

- Recommendation engines
- Market segmentation
- Social network analysis
- Search result grouping
- Medical imaging
- Image segmentation
- Anomaly detection

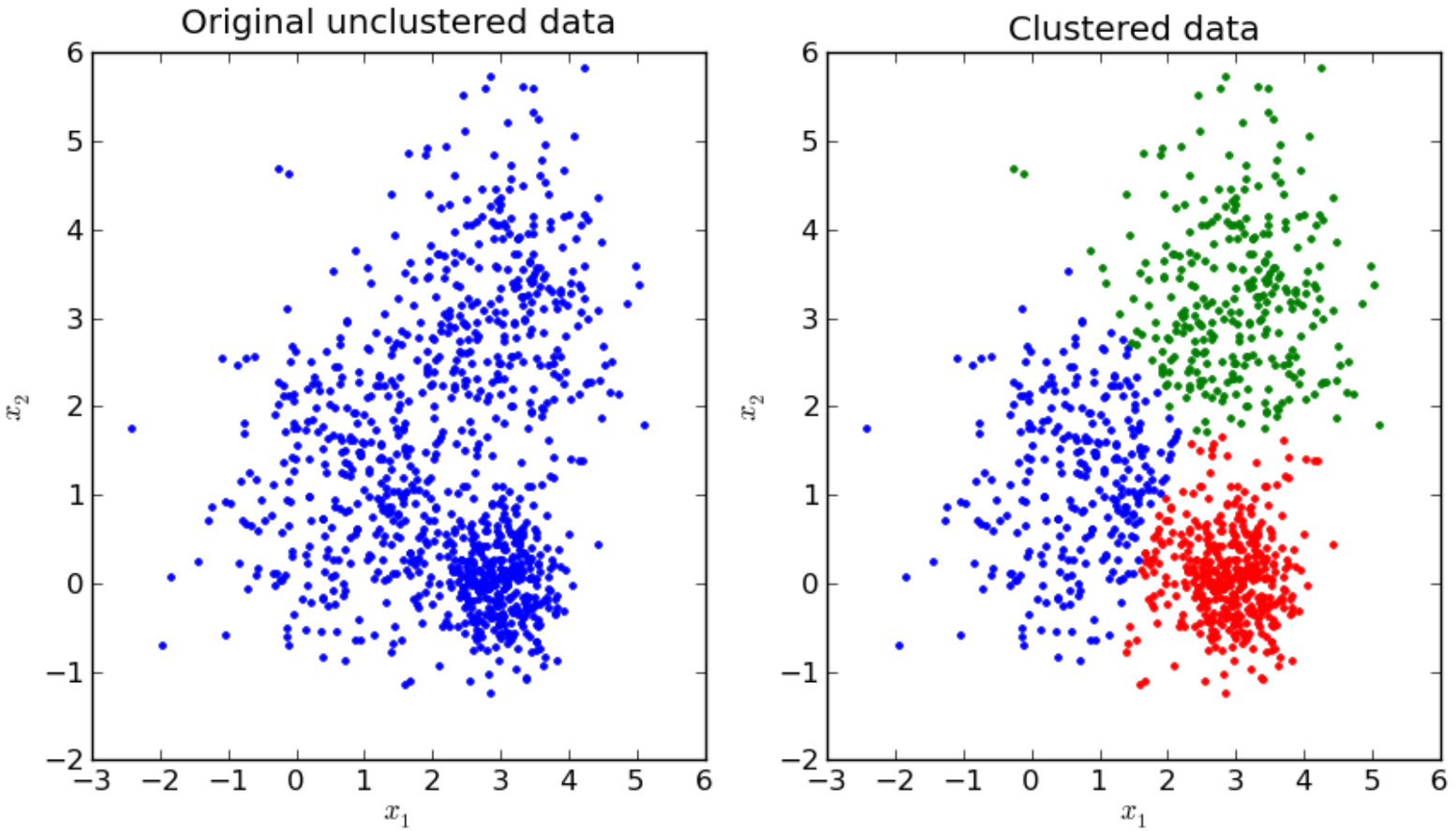
OR

When your boss comes to you and says “Help me understand our customers better so we can better sell them things”

Since there’s no specific objective we aren’t looking for specific insights into a phenomena. So instead we can look for structures within our data without being tied to a specific objective.

Clusters add new features to our data by grouping them together through similarity

CLUSTERING: Centroids



Source: http://stackoverflow.com/questions/24645068/k-means-clustering-major-understanding-issue

ACTIVITY: KNOWLEDGE CHECK

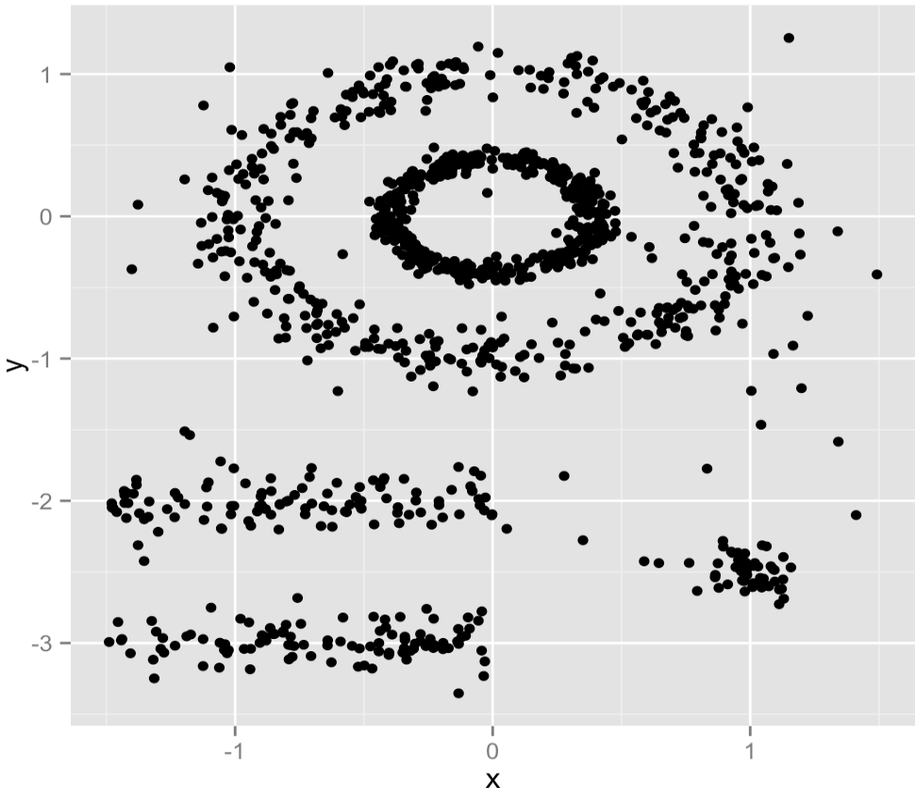
ANSWER THE FOLLOWING QUESTIONS

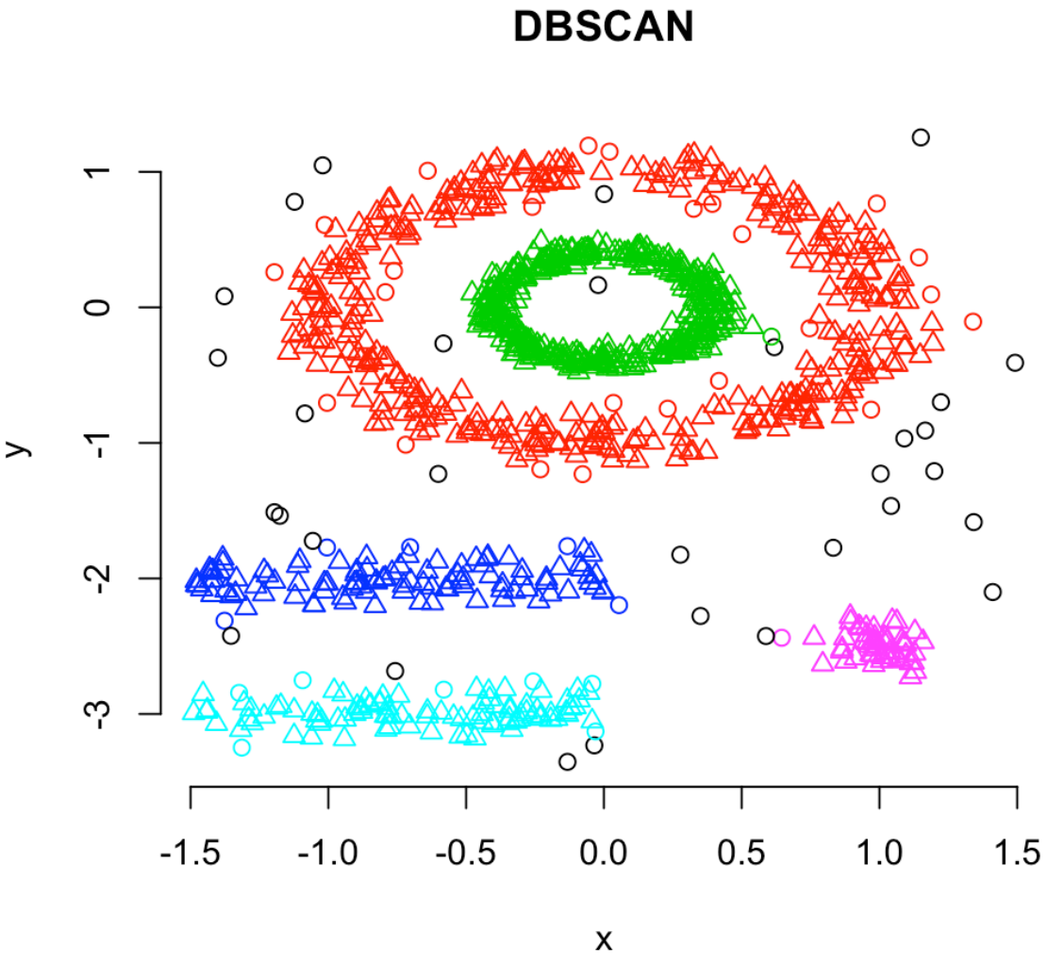
1. Why might data often appear in centered clusters?

DELIVERABLE

Answers to the above questions

CLUSTERING: Density-Based





Source: http://www.sthda.com/english/wiki/dbscan-density-based-clustering-for-discovering-clusters-in-large-datasets-with-noise-unsupervised-machine-learning

ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

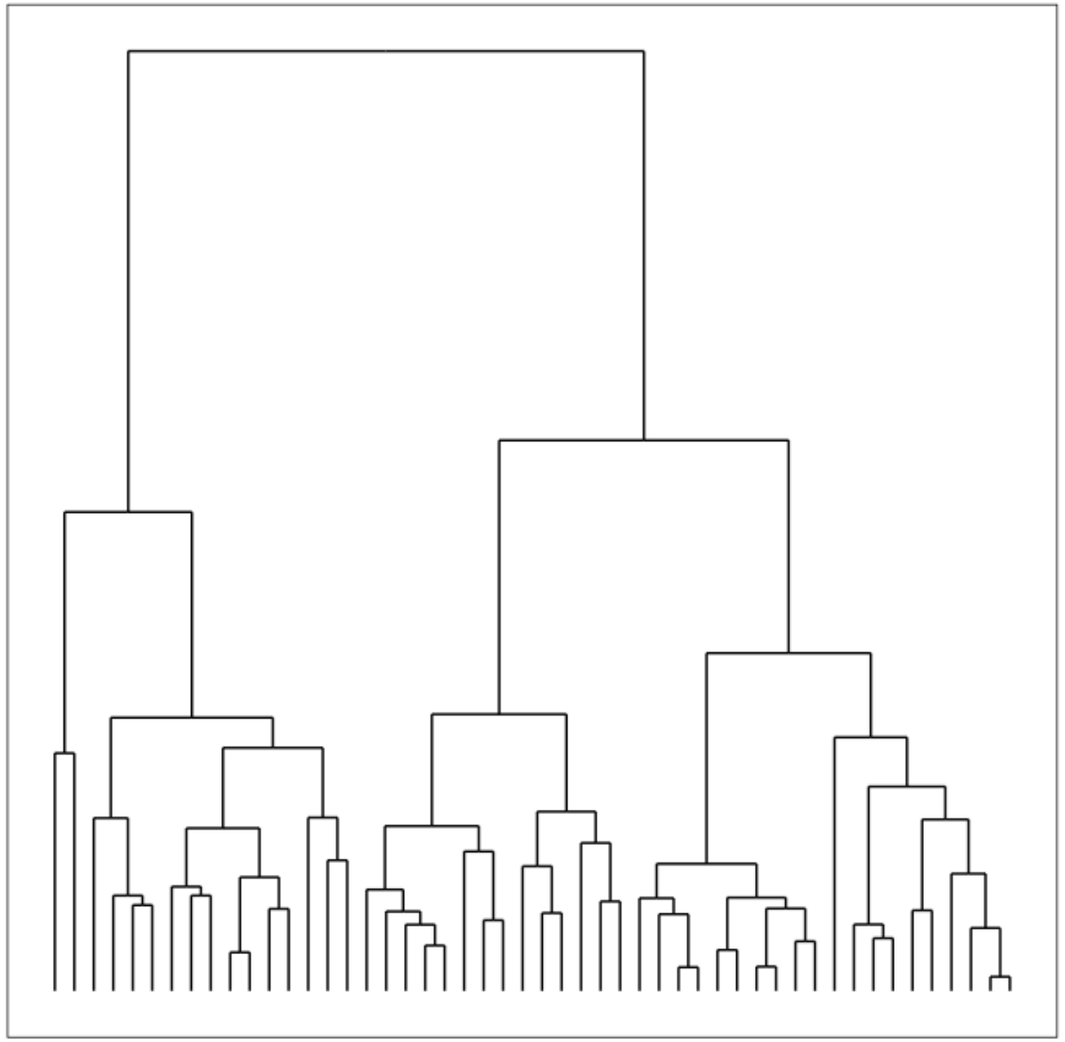
1. Why might data often appear in density-based clusters?

DELIVERABLE

Answers to the above questions

CLUSTERING: Hierarchical

- Build hierarchies that form clusters
- Based on classification trees (future lesson)



ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

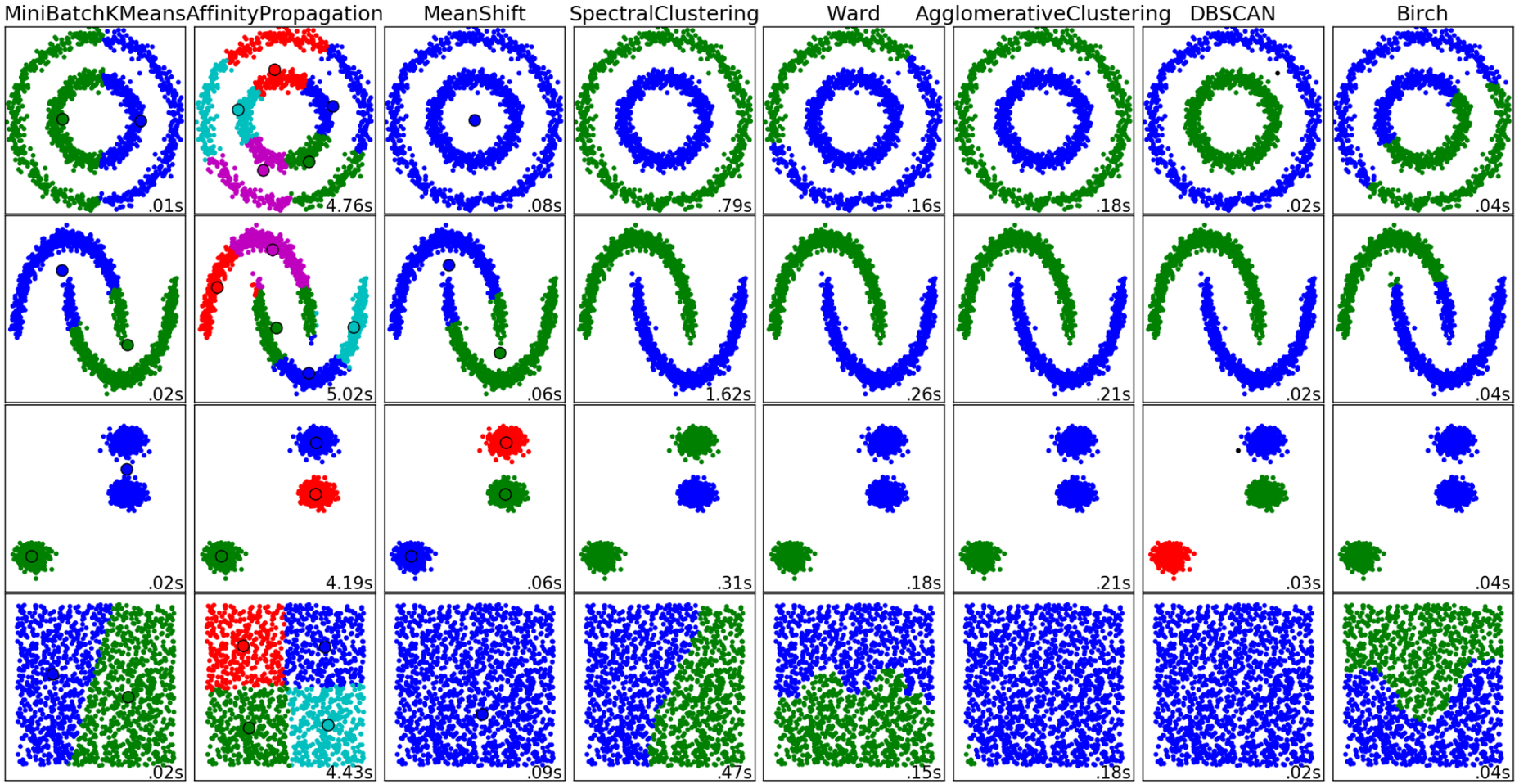
1. How is unsupervised learning different from classification?

DELIVERABLE

Answers to the above questions

CLUSTERING

- There are many clustering algorithms



ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

1. Can you think of a real-world clustering application?

DELIVERABLE

Answers to the above questions

ACTIVITY: KNOWLEDGE CHECK

ANSWERS

1. Recommendation Systems e.g. Netflix genres
2. Medical Imaging: differentiate tissues
3. Identifying market segments
4. Discover communities in social networks
5. Lots of applications for genomic sequences (homologous sequences, genotypes)
6. Earthquake epicenters
7. Fraud detection

K-MEANS: CENTRIOD CLUSTERING

K-MEANS CLUSTERING

- k-Means seeks to minimize the sum of squares about the means

- Precisely, find k subsets S_1, … S_k of the data with means mu_1, …, mu_k that minimizes:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2$$

K-MEANS CLUSTERING

- k-Means clustering is a popular centroid-based clustering algorithm
- Basic idea: find *k* clusters in the data centrally located around various mean points
- Awesome Demo

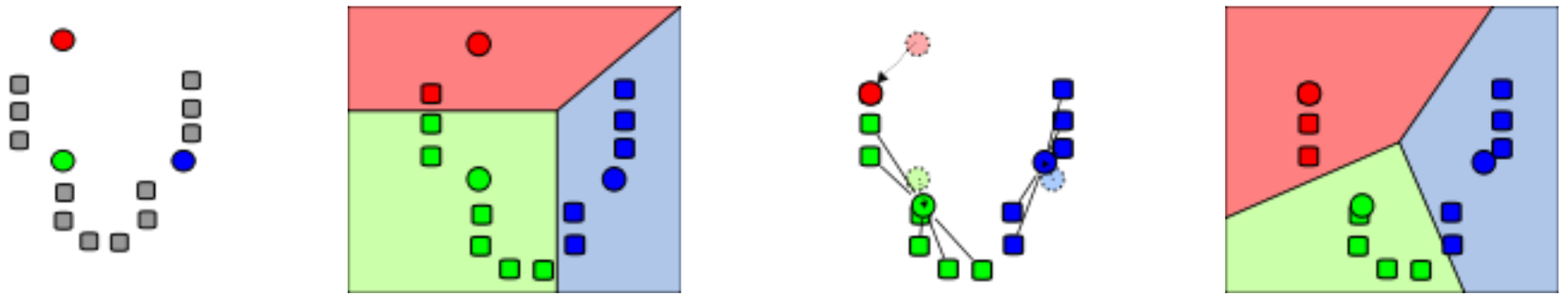
K-MEANS CLUSTERING

- This is a computationally difficult problem to solve so we rely on heuristics

- The “standard” heuristic is called “Lloyd’s Algorithm”:
  - Start with k initial mean values
  - Data points are then split up into a Voronoi diagram
    - Each point is assigned to the “closest” mean
  - Calculate new means based on centroids of points in the cluster
  - Repeat until clusters do not change

K-MEANS CLUSTERING

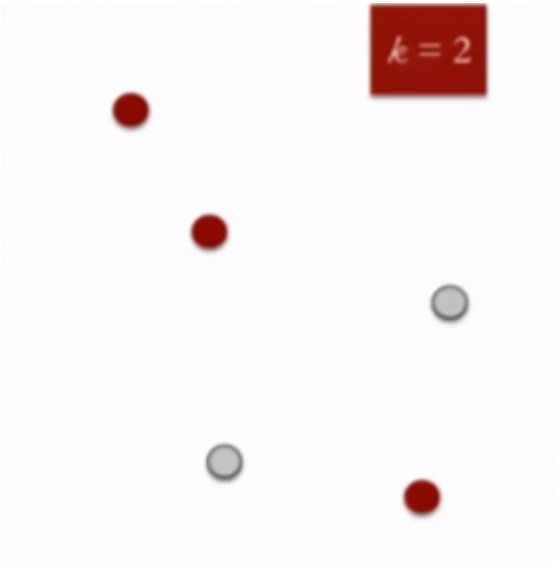
- Start with initial k mean values
- Data points are then split up into a Voronoi diagram
- Calculate new means based on centroids



K-MEANS CLUSTERING

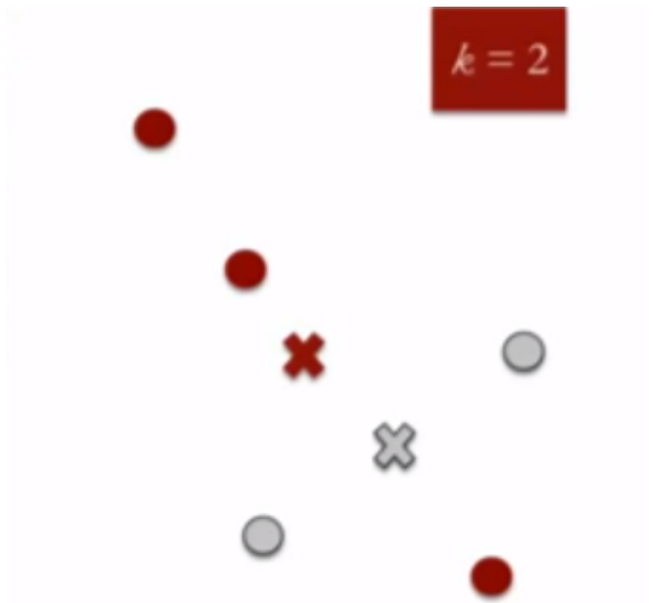
K means is an iterative clustering algorithm that aims to find local maxima in each iteration. This algorithm works in these 5 steps :

1. Specify the desired number of clusters K : Let us choose k=2 for these 5 data points in 2-D space.
2. Randomly assign each data point to a cluster : Let’s assign three points in cluster 1 shown using red color and two points in cluster 2 shown using grey color.



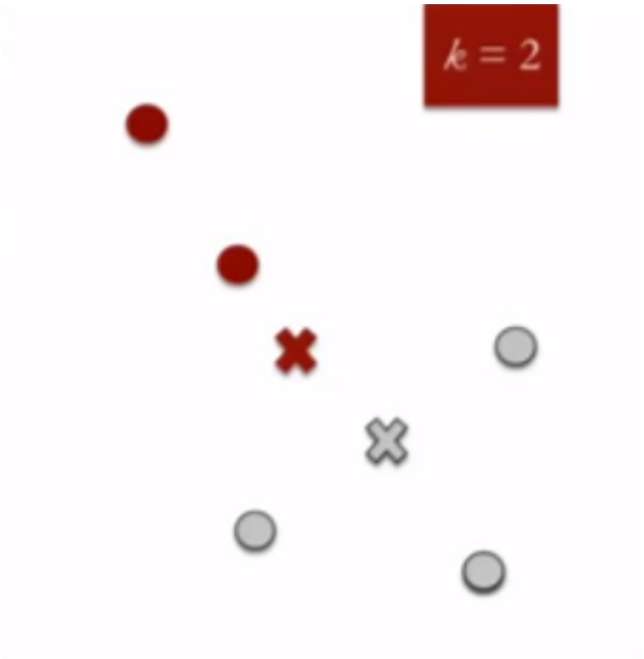
K-MEANS CLUSTERING

3. Compute cluster centroids : The centroid of data points in the red cluster is shown using red cross and those in grey cluster using grey cross.



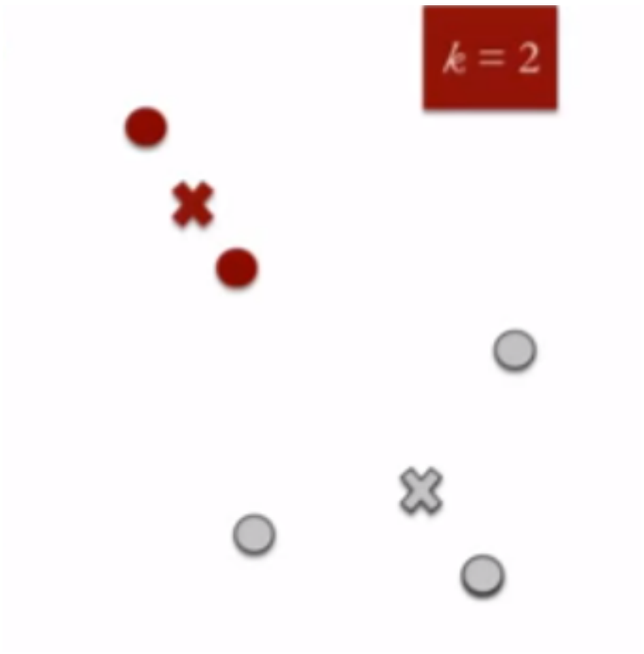
K-MEANS CLUSTERING

4. Re-assign each point to the closest cluster centroid : Note that only the data point at the bottom is assigned to the red cluster even though its closer to the centroid of grey cluster. Thus, we assign that data point into grey cluster



K-MEANS CLUSTERING

5. Re-compute cluster centroids : Now, re-computing the centroids for both the clusters.



6. Repeat steps 4 and 5 until no improvements are possible : Similarly, we'll repeat the 4th and 5th steps until we'll reach global optima. When there will be no further switching of data points between two clusters for two successive repeats. It will mark the termination of the algorithm if not explicitly mentioned.

ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

1. How do we assign meaning to the clusters we find?
2. Do clusters always have meaning?

DELIVERABLE

Answers to the above questions

K-MEANS CLUSTERING

- Assumptions are important! k-Means assumes:
  - k is the correct number of clusters
  - the data is isotropically distributed (circular/spherical distribution)
  - the variance is the same for each variable
  - clusters are roughly the same size

Nice counterexamples / cases where assumptions are not met:

- http://varianceexplained.org/r/kmeans-free-lunch/
- Scikit-Learn Examples

DBSCAN: DENSITY BASED CLUSTERING

DBSCAN CLUSTERING

- DBSCAN: Density-based spatial clustering of applications with noise (1996)

- Main idea: Group together closely-packed points by identifying
  - Core points
  - Reachable points
  - Outliers (not reachable)

DBSCAN CLUSTERING

- DBScan takes in two parameters:
  - **min_samples** the number of samples (or total weight) in a neighborhood for a point to be considered a core point. This includes the point itself.

  - Epsilon (**eps**) the local radius for expanding clusters. Think of it as a step size - DBSCAN never takes a step larger than this, but by doing multiple steps DBSCAN clusters can become much larger than eps.

Simply put
  - Eps: Maximum radius of the neighborhood.
  - MinPts: Minimum number of points in the Eps-neighborhood to make a determination

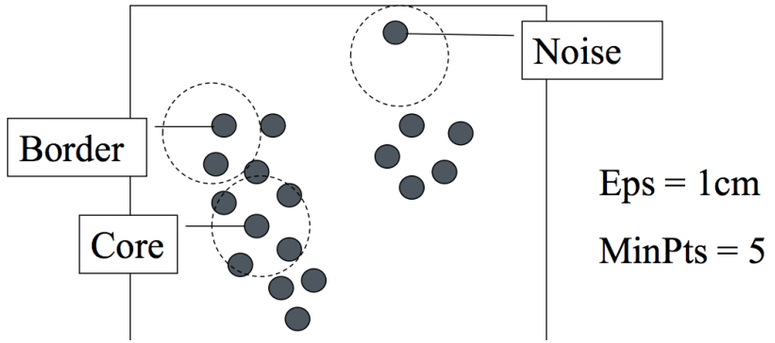
UNDERSTANDING TERMS

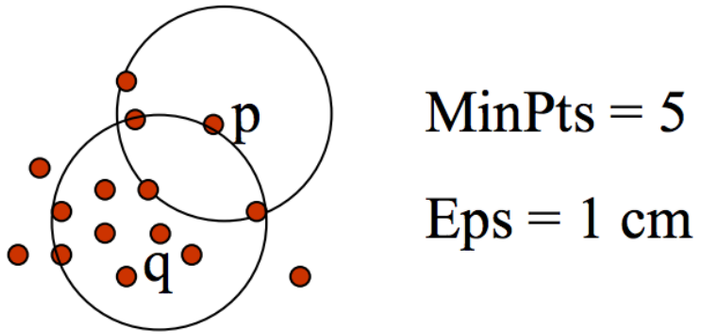
Terms

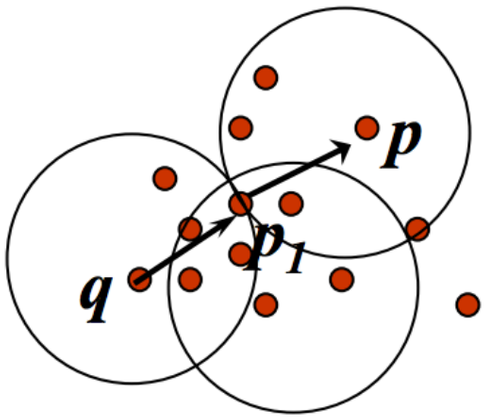
- **The Eps-neighborhood of a point**: Point inside the circle
- **Outlier**: Not in a cluster.
- **Core point**: In a dense neighborhood
- **Border point**: In cluster but neighborhood is not dense

- **Directly density-reachable**: A point **p** is directly density-reachable from a point **q**
    - **p** is in the neighborhood and **q** is a core point
    - **p** doesn't need to be a core point

- **Density-reachable:** A point **p** is density-reachable from a point **q** w.r.t. **Eps** and **MinPts** if there is a chain of points $\mathbf{p_1 \ldots p_n}$, $\mathbf{p_1 = q}$ $\mathbf{p_n = p}$ such that $\mathbf{p_{i+1}}$ is directly density-reachable from $\mathbf{p_i}$

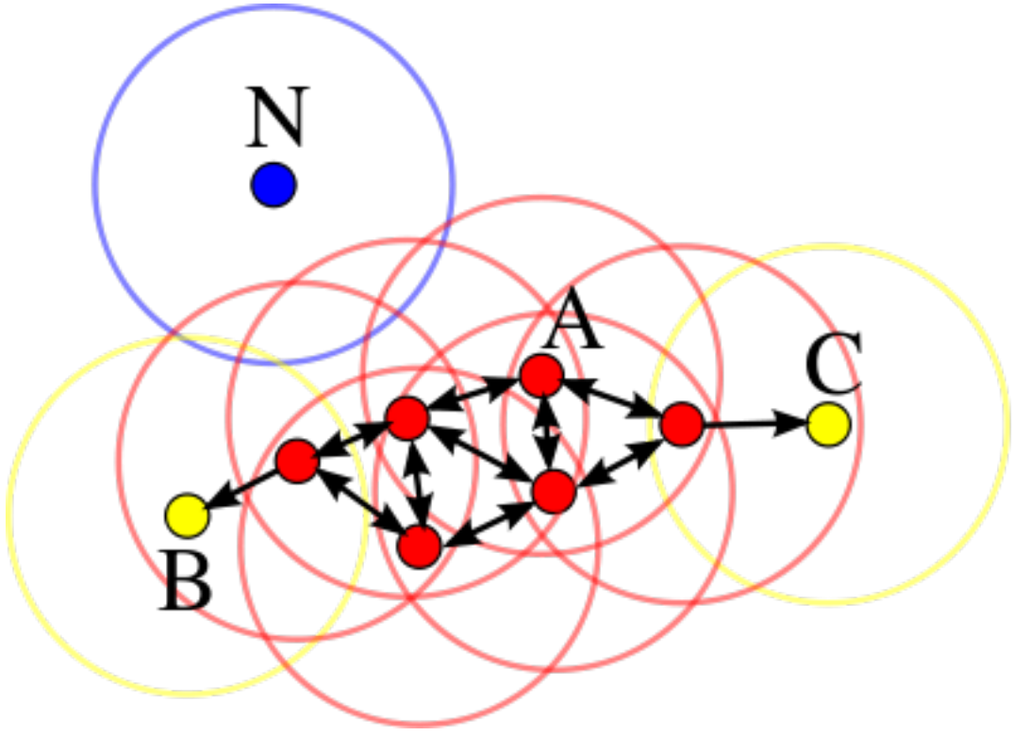






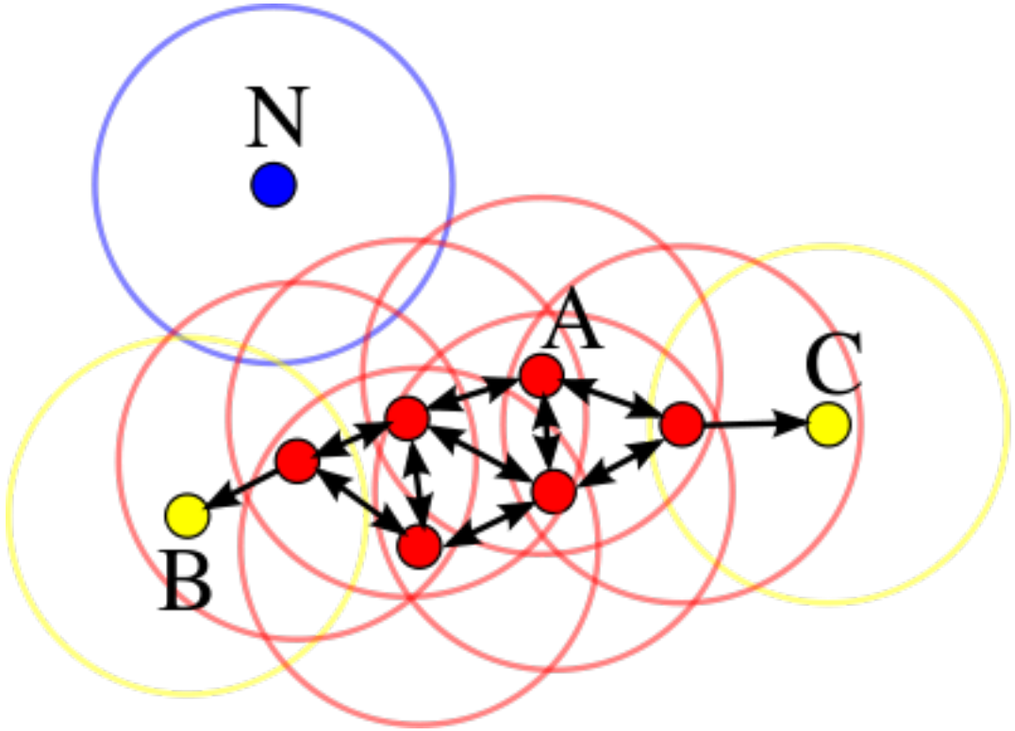
DBSCAN CLUSTERING

- Core points: at least **min_samples** points within **eps** of the core point
  - Such points are *directly reachable* from the core point
- Reachable: point *q* is reachable from *p* if there is a path of core points from *p* to *q*
- Outlier: not reachable



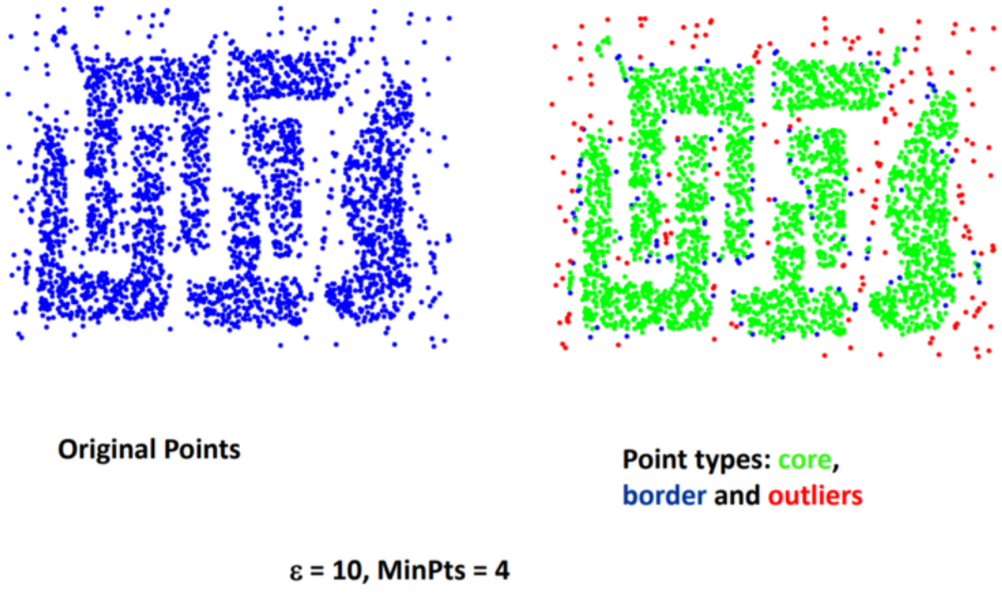
DBSCAN CLUSTERING

- A cluster is a collection of connected core and reachable points



CLUSTERING: Density-Based

- Another example:



- Awesome Demo
- Image Source

ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

1. How does DBSCAN differ from k-means?

DELIVERABLE

Answers to the above questions

DBSCAN CLUSTERING

- DBSCAN advantages:
  - Can find arbitrarily-shaped clusters
    - Can handle clusters of different shapes and sizes
  - Resistant to Noise
  - Don’t have to specify number of clusters
  - Robust to outliers

- DBSCAN disadvantages:
  - Doesn’t work well when clusters are of varying densities
    - hard to chose parameters that work for all clusters
  - Sensitive to parameter settings – Hard to determine the correct set of parameters

ACTIVITY: CLUSTERING USERS

ANSWER THE FOLLOWING QUESTIONS

1. How does DBSCAN differ from k-means?

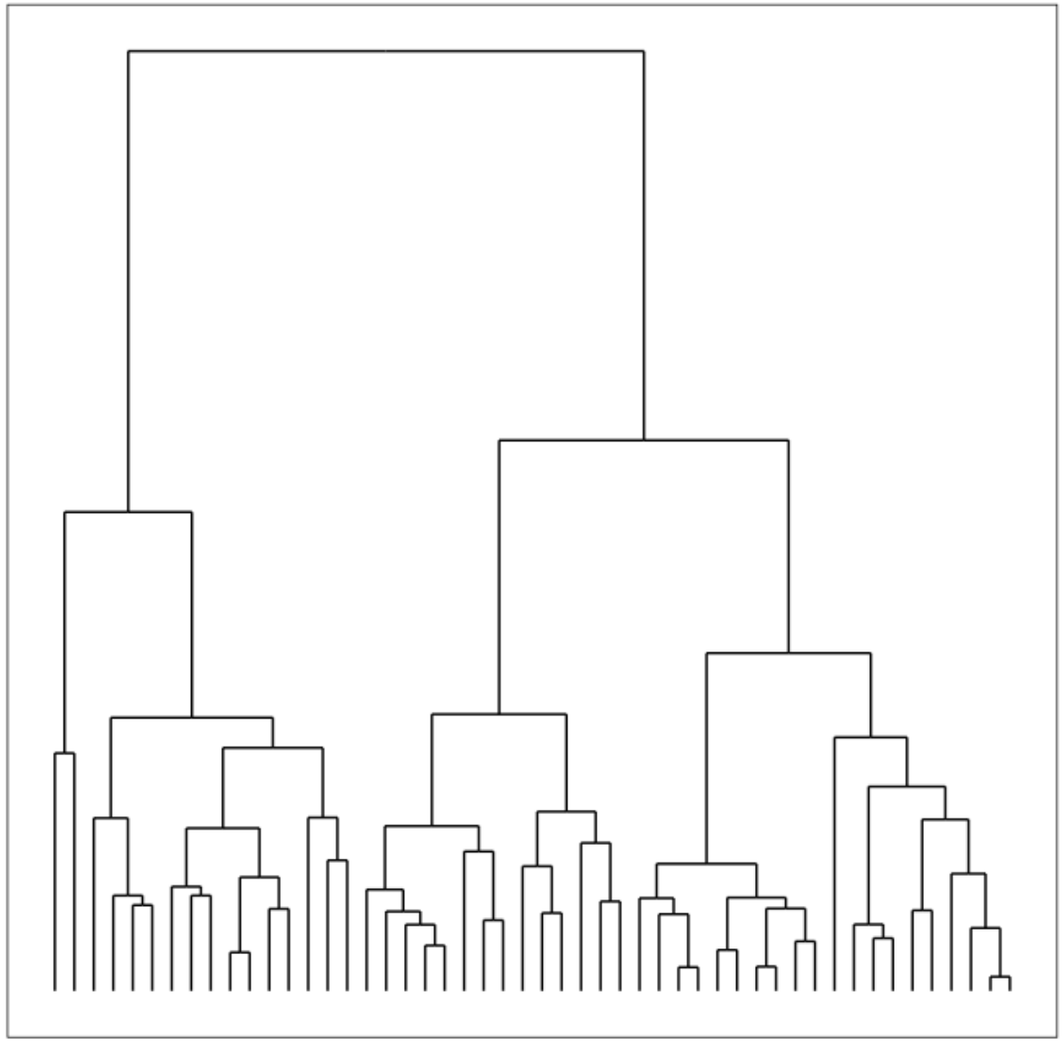
DELIVERABLE

Answers to the above questions

HIERARCHICAL CLUSTERING

CLUSTERING: Hierarchical

- Build hierarchies that form clusters
- Based on classification trees



HIERARCHICAL CLUSTERING

We'll discuss the details once we cover decision trees. For now we can black box the model and fit with sklearn

- from sklearn.cluster import AgglomerativeClustering
- est = AgglomerativeClustering(n_clusters=4)
- est.fit(X)
- labels = est.labels_

Let's try it out!

CLUSTERING METRICS

CLUSTERING METRICS

- As usual we need a metric to evaluate model fit

- For clustering we use a metric called the Silhouette Coefficient
  - **a** is the mean distance between a sample and all other points in the cluster
  - **b** is the mean distance between a sample and all other points in the *nearest* cluster

- The Silhouette Coefficient is:

$$\frac{b - a}{\max(a, b)}$$

- Ranges between 1 and -1
- Average over all points to judge the cluster algorithm

CLUSTERING METRICS

- from sklearn import metrics
- from sklearn.cluster import KMeans
- kmeans_model = KMeans(n_clusters=3, random_state=1).fit(X)
- labels = kmeans_model.labels_
- metrics.silhouette_score(X, labels, metric='euclidean')

CLUSTERING METRICS

- There are a number of other metrics based on:
  - Mutual Information
  - Homogeneity
  - Adjusted Rand Index (when you know the labels on the training data)

CLUSTERING, CLASSIFICATION, AND REGRESSION

ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

1. How might we combine clustering and classification?

DELIVERABLE

Answers to the above questions

CLUSTERING, CLASSIFICATION, AND REGRESSION

- We can use clustering to discover new features and then use those features for either classification or regression
- For classification, we could use e.g. k-NN to classify new points into the discovered clusters
- For regression, we could use a dummy variable for the clusters as a variable in our regression

TOPIC REVIEW

REVIEW AND NEXT STEPS

- Clustering is used to discover features, e.g. segment users or assign labels (such as species)

- Clustering may be the goal (user marketing) or a step in a data science pipeline

LESSON

Q & A

LESSON

EXIT TICKET

DON’T FORGET TO FILL OUT YOUR EXIT TICKET