Chapter 3+10 review

CS 412 Fall 2015

12/08/2015

General notes

- Coverage: all chapters
- What we will present here are the topics we want to remind you. Don’t exclude anything that do not appear here.
- For conceptual/explanatory questions:
  - Answers should be concise and to-the-point.
  - Example: If it asks you to suggest a method to do clustering in a particular scenario, you can say using “k-Means” (without describing what is k-Means) and show how it is suitable with the scenario.
- For application questions: Read the requirements carefully and look for a quick way to solve it.

Data Preprocessing

If possible, learn concepts in a group, compare & contrast them

- Correlation vs. Covariance
- Chi-square vs. Pearson correlation coefficient
- Sampling with vs. without replacement
- Uniform sampling vs. stratified sampling
- Min-Max normalization vs. Z-score normalization
- …

PCA

- Understand the algorithmic procedure
  - Don’t miss any step like zero-mean normalization
- Understand PCA’s purpose (dimension reduction) and how PCA can achieve the purpose (select top k dimensions)
- Don’t need to know how to do eigenvector decomposition
- Don’t expect that the only way to ask about PCA is running PCA on a particular dataset

Summary:
PCA by Eigenvector Decomposition

- Given $\mathbf{X}$ mean-normalized data matrix with m dims x n points.
- Find covariance matrix $\mathbf{C_X} = \frac{1}{n}\mathbf{X}\mathbf{X}^{\mathbf{T}}$, an m x m symmetric matrix.
- For $\mathbf{C_X}$: $\mathbf{E} = [\mathbf{e}_1, \dots, \mathbf{e}_m]$ (eigenvectors) and $\mathbf{D}$ ($\mathbf{C_Y} = \mathbf{D}$, new covar), sorted by variance each -- Use eigenvector decomposition or SVD.
- Find $\mathbf{P} = \begin{bmatrix} \mathbf{e}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{e}_m^{\mathbf{T}} \end{bmatrix}$ as principle components.
- Select $\mathbf{P}_k$ as first k principle components.
- New data $\mathbf{Y} = \mathbf{P}_k\mathbf{X}$.

- What k to choose? Large enough to preserve sufficient covariance.

Cluster Analysis

Learn all algorithms together, compare and contrast them

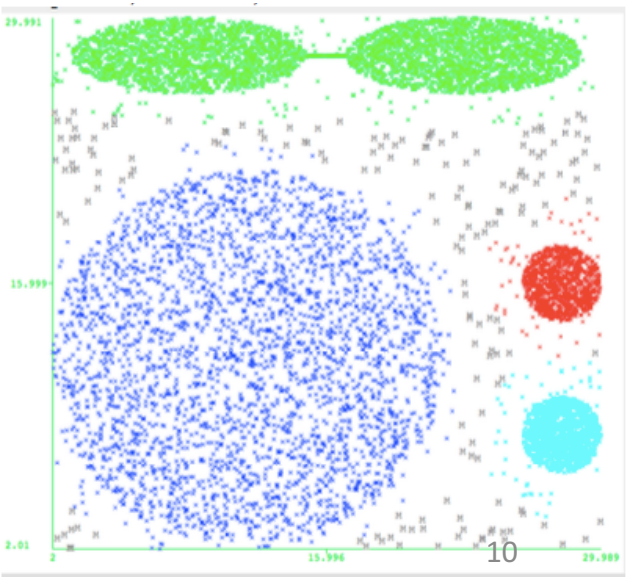
- K-Means vs. K-Medoids
- K-Means vs. DBSCAN vs. AGNES
- Single-link vs. Complete link
- …

Tips for algorithm application problems

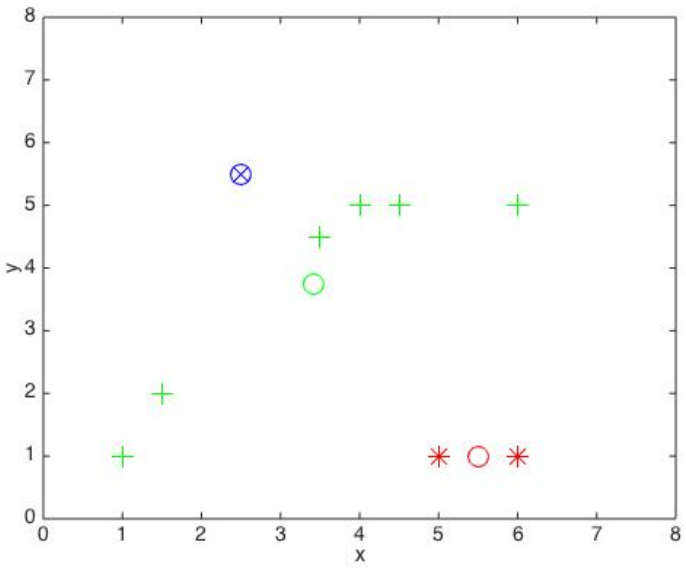
- If you used Excel/Java/Matlab… to run algorithms in your homework, try to think what you would do with a pencil.
  - Data will be so special that calculation can be done simply, e.g., data points lie on a grid.
- Sometimes, you can guess or quikly come up with the output without actually running the algorithm. If we need you to show only the final output, you can guess!
  - Be careful, different algorithms may give different outputs for the same data. So you must understand the algorithm to guess its output with regard to the provided data.

K-Means algorithm

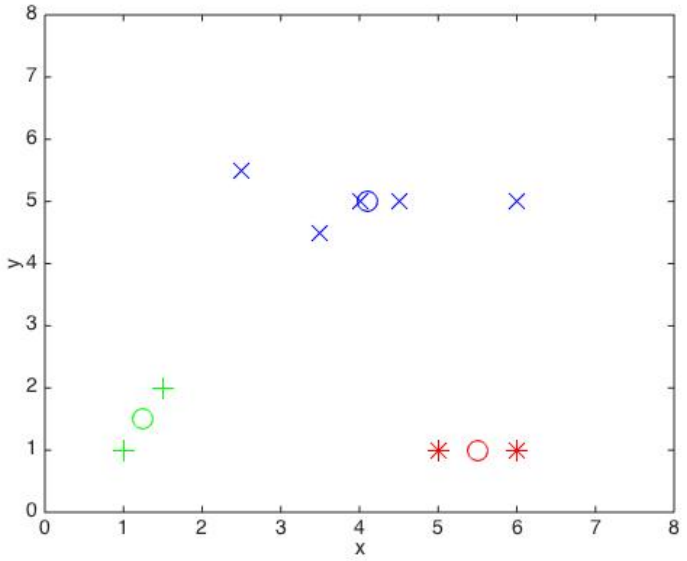
- Efficiency: O(tKn) where n: # of objects, K: # of clusters, and t: # of iterations
  - Normally, K, t << n; thus, nearly linear -> an efficient method
- Need to specify K, the number of clusters, in advance
  - There are ways to automatically determine the “best” K
  - In practice, one often runs a range of values and selected the “best” K value
- Restricted usage:
  - Applicable only to objects in a continuous n-dimensional space
    - Using K-modes for categorical data
  - Not suitable to discover clusters with non-convex shapes
    - Using density-based clustering, kernel K-means, etc. instead
- Other than that: still have at least two issues



Issue 1: K-means often terminates at a local optimal: initialization can be important to find high-quality clusters

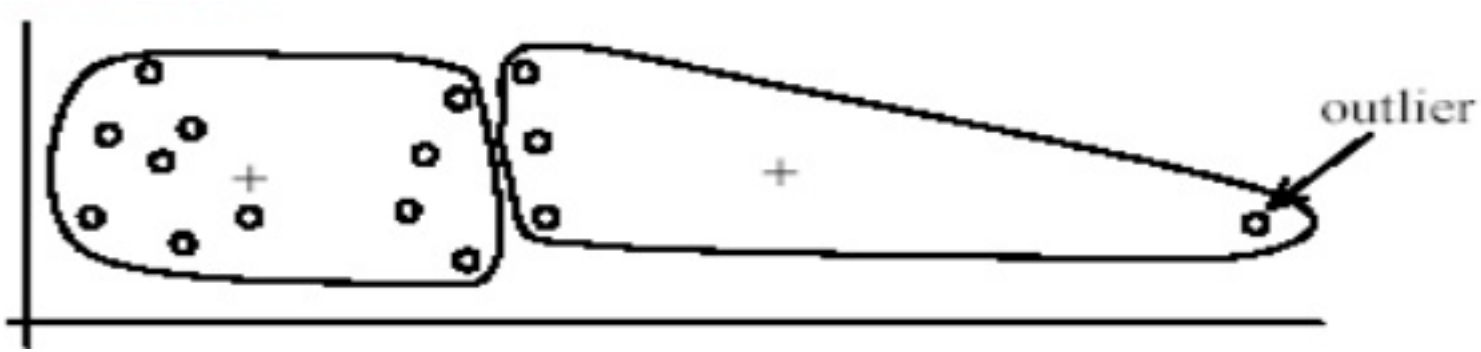


New

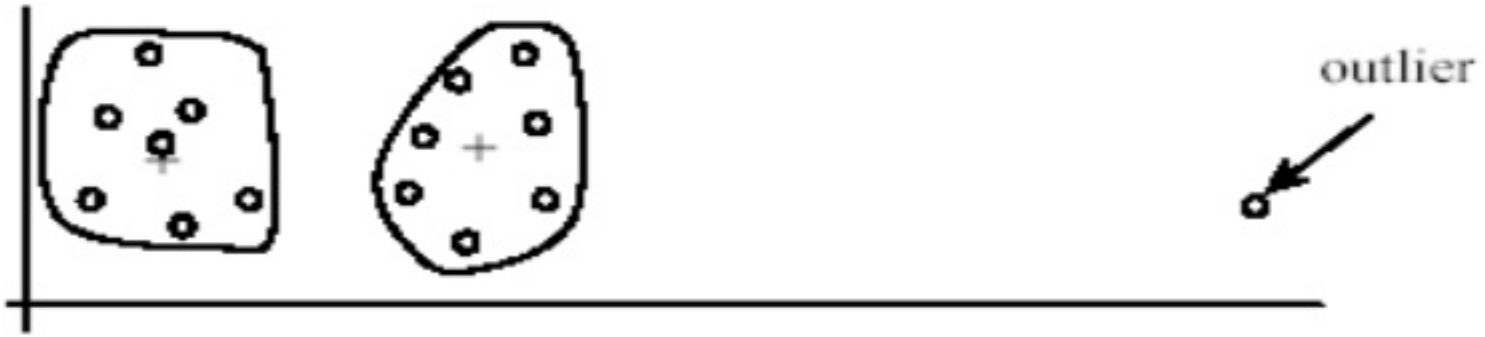


Old

Issue 2. K-Means is sensitive to noisy data and outliers. How may K-Medoids and DBSCAN help?



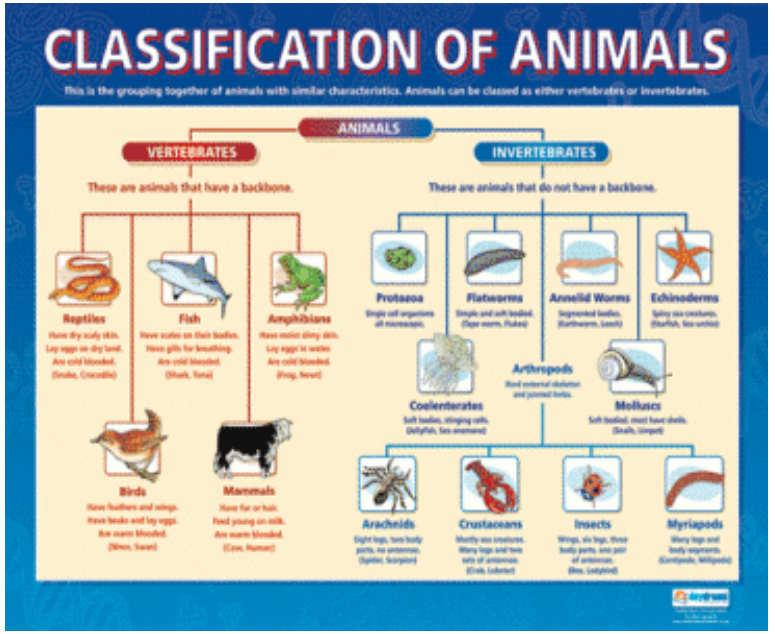
Undesirable clustering



Desirable clustering

Hierarchical clustering arises when clusters are nested

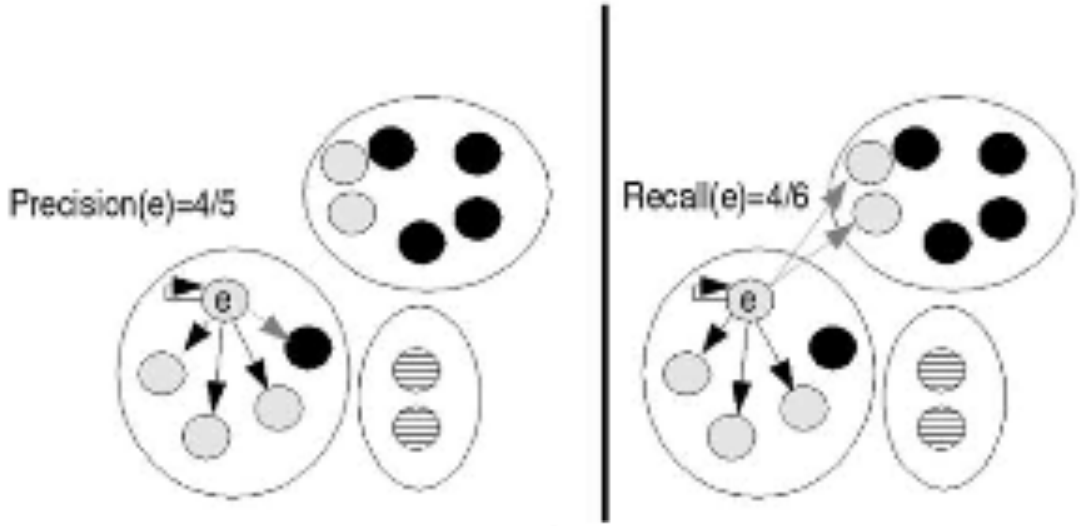
- The output is a dendrogram, not yet separate clusters
- Need more info to actually separate clusters, e.g., number of clusters.
  - The good thing is we do not have to guess the number of clusters ahead. We can do that based on the dendrogram.



Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape: dense area
  - Discover noise: if points do not belong to a dense area
  - Need density parameters as termination condition
  - Require one scan

B-Cubed
Precision & Recall



$$\text{Precision}_i = \frac{\textit{number of correct elements in the output chain containing entity}_i}{\textit{number of elements in the output chain containing entity}_i}$$

$$\text{Recall}_i = \frac{\textit{number of correct elements in the output chain containing entity}_i}{\textit{number of elements in the truth chain containing entity}_i}$$

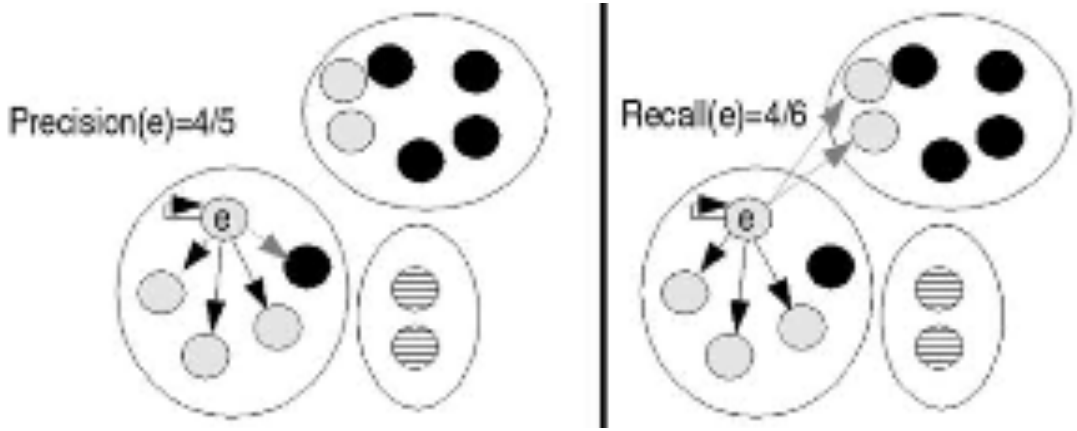
$$\text{Final Precision} = \sum_{i=1}^{N} w_i * \textit{Precision}_i$$

$$\text{Final Recall} = \sum_{i=1}^{N} w_i * \textit{Recall}_i$$

By default, $w_i = \frac{1}{N}$

Note: The B-Cubed formula in the textbook is imprecise! Use this one (from [Bagga & Baldwin 1999)!

B-Cubed
Precision & Recall Formulas



- Notation:
  - Data points: $o_{1..N}$
  - $L(o_i)$ is the name of the cluster containing o_i in the ground truth
  - $C(o_i)$ is the name of the cluster containing o_i in the clustering output
- Formulas (note: it is not necessary that j $\neq i$):

$$Correct(o_i, o_j) = \begin{cases} 1 & if\ [L(o_i) = L(o_j)\ and\ C(o_i) = C(o_j)] \\ 0 & otherwise \end{cases}$$

$$Precision_i = \frac{\sum_{o_j : C(o_j) = C(o_i)} Correct(o_i, o_j)}{\| \{o_j | C(o_j) = C(o_i)\} \|}$$

$$Recall_i = \frac{\sum_{o_j : L(o_j) = L(o_i)} Correct(o_i, o_j)}{\| \{o_j | L(o_j) = L(o_i)\} \|}$$

- By default, all data points are equally important:

$$Precision = \frac{\sum_{i=1}^{N} Precision_i}{N}$$

$$Recall = \frac{\sum_{i=1}^{N} Recall_i}{N}$$

B-Cubed Precision & Recall Intuition

- An entity = a data point
- A chain = a cluster
- Precision: Among all the entities that the algorithm groups with $entity_i$, how many of them, including $entity_i$ itself, (correctly) belong to the same cluster with $entity_i$ in the ground truth?
- Recall: Among all the entities that belong to the same cluster with $entity_i$ in the ground truth, how many of them, including $entity_i$ itself, are (correctly) grouped with with $entity_i$ by the algorithm?
- Because we count the data point itself, precision and recall for each data point is always bigger than 0.

Handling each outlier as an individual cluster

- Why all outliers do not belong to a single cluster?
  - Because they are not closed to each other!
- Example: https://piazza.com/class/idqujg4tiae3q0?cid=568