CLUSTERING

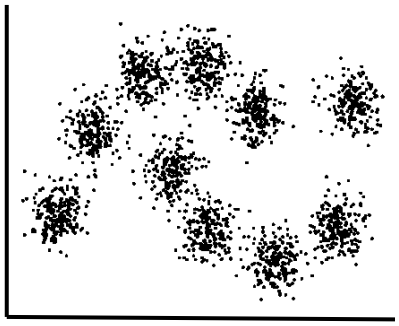
Today

- Reading
  - Introduction to Information Retrieval (IR) Ch. 16, 17
- Goals
  - Flat clustering algorithms
    - K-means
  - Hierarchical clustering algorithms
    - Agglomerative clustering
    - Divisive clustering (it's divisive!)
  - Evaluating clusters

Unsupervised Learning

- Learning without labels
  - Often comes down to clustering
  - Can be used as a surrogate for supervised learning

$$D = \begin{matrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \\ \\ \mathbf{x}_N \end{matrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1F} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2F} \\ & & & & \\ & & & & \\ x_{N1} & x_{N2} & x_{N3} & \cdots & x_{NF} \end{bmatrix}$$



Clustering

- Grouping data into (hopefully useful) sets

- Why do clustering?
  - Labeling is costly
  - Data pre-processing
    - Text Classification (e.g., search engines, Google Sets)
  - Exploratory Data Analysis
    - Clusters might suggest natural groups.
  - Visualization

Terminology

- An m-clustering of D is a partition of D into sets (clusters) C_1, C_2,…,C_m such that
  - The clusters are non-empty
  - The union of the clusters is D
  - The intersection of the clusters is empty
- The centroid of a cluster is the mean of all the elements in the cluster

Types of clustering algorithms

- Flat versus Hierarchical
  - Flat algorithms return an unstructured set of clusters
  - Hierarchical algorithms return a hierarchy of clusters
- Sequential (online) versus Batch
  - Sequential algorithms are typically fast
- Hard versus soft
  - Hard algorithms make a hard assignment of elements to clusters
  - Soft algorithms compute a distribution over clusters for each element

K-means Clustering is an old algorithm

"If someone is found slain, lying in a field in the land the Lord your God is giving you to possess, and it is not known who the killer was, your elders and judges shall go out and measure the distance from the body to the neighboring towns. Then the elders of the town nearest the body shall…"

- Deuteronomy 21

K-means Clustering

- The most well-known (widely-used) clustering algorithm
- Minimizes the sum of the squared distances of each vector from its centroid:

$$\text{RSS} = \sum_{k=1}^{K} \sum_{\vec{x} \in C_k} ||\vec{x} - \mu_k||_2^2$$

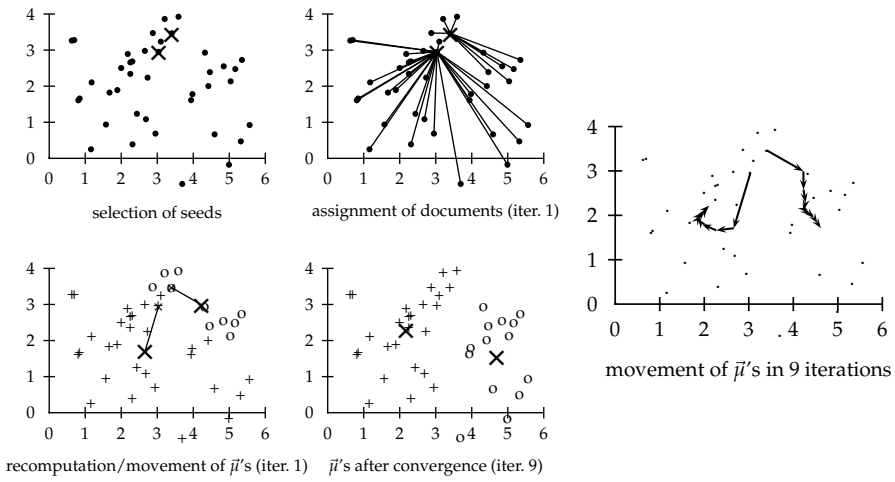
Residual sum of squares

- User must set K
- Assumes instances x represented as normalized vectors in a real-valued space

K-means Clustering

K-MEANS($\{\vec{x}_1, \ldots, \vec{x}_N\}, K$)
1 $(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow$ SELECTRANDOMSEEDS($\{\vec{x}_1, \ldots, \vec{x}_N\}, K$)
2 **for** $k \leftarrow 1$ **to** K
3 **do** $\vec{\mu}_k \leftarrow \vec{s}_k$
4 **while** stopping criterion has not been met
5 **do for** $k \leftarrow 1$ **to** K
6 **do** $\omega_k \leftarrow \{\}$
7 **for** $n \leftarrow 1$ **to** N
8 **do** $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$
9 $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*
10 **for** $k \leftarrow 1$ **to** K
11 **do** $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*
12 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

K-means Clustering



K-means Clustering

- Starting seeds (centroids)
  - Randomly select K initial data points to be the centroids
  - Run multiple K-means each with different random seeds
  - Use results from a different clustering algorithm (run on random subset of data)
- Stopping criteria
  - See IR for a number of possible stopping criteria. For example:
  - Fixed number of iterations
  - Residual sum of squares (RSS) falls below threshold
- Choosing K
  - Rules-of-thumb or experience
  - Try multiple values for K and plot RSS. Look for the elbow.
  - Add regularization term

Flat versus Hierarchical

- K-means
  - Returns unstructured set of clusters
  - Requires user to determine K
  - Non-deterministic
  - Linear run time O(IKNM)
- Hierarchical (e.g. Agglomerative clustering)
  - Returns a hierarchy of clusters
  - No need to (initially) determine K
  - Deterministic
  - Quadratic run time

Hierarchical Clustering

- Cluster data points based on distance metric or similarity measure
- A metric is a function that assigns a positive real-valued number (representing distance) to every pair of vectors in some vector space
- Minkowski distance is given by

$$d_p(\vec{x}_i, \vec{x}_j) = \left(\sum_{m=1}^{M} |x_{im} - x_{jm}|^p \right)^{1/p}$$

Hierarchical Clustering

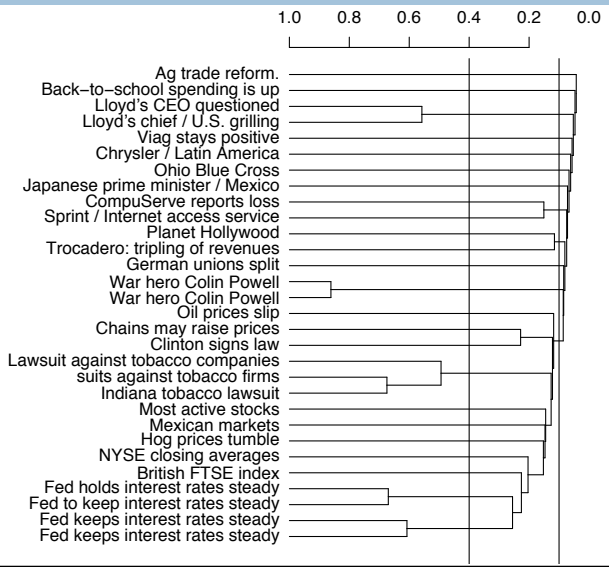
- For p = 1, Manhattan distance $d_1(\vec{x}_i, \vec{x}_j) = \sum_{m=1}^{M} |x_{im} - x_{jm}|$
- For p = 2, Euclidean distance $d_2(\vec{x}_i, \vec{x}_j) = \left(\sum_{m=1}^{M} |x_{im} - x_{jm}|^2 \right)^{1/2}$
- Cosine similarity also common measure (Note inverse of distance)

$$\cos(\vec{x}_i, \vec{x}_j) = \frac{\vec{x}_i^{\mathsf{T}} \vec{x}_j}{||\vec{x}_i||_2 \, ||\vec{x}_j||_2} = \frac{\sum_{m=1}^{M} x_{im} \cdot x_{jm}}{||\vec{x}_i||_2 \, ||\vec{x}_j||_2}$$

Hierarchical Clustering

- Agglomerative clustering
  - Start with *N* clusters each with one data point
  - Merge similar clusters to form larger clusters until there is only a single cluster left
- Divisive Clustering
  - Start with a single cluster containing all data points
  - Divide large clusters into smaller clusters until each cluster contains a single data point

Agglomerative Clustering



Agglomerative Clustering

```
SIMPLEHAC(d_1, ..., d_N)
 1  for n ← 1 to N
 2  do for i ← 1 to N
 3     do C[n][i] ← SIM(d_n, d_i)
 4     I[n] ← 1  (keeps track of active clusters)
 5  A ← []  (assembles clustering as a sequence of merges)
 6  for k ← 1 to N − 1
 7  do ⟨i, m⟩ ← arg max_{⟨i,m⟩:i≠m∧I[i]=1∧I[m]=1} C[i][m]
 8     A.APPEND(⟨i, m⟩)  (store merge)
 9     for j ← 1 to N
10     do C[i][j] ← SIM(i, m, j)
11        C[j][i] ← SIM(i, m, j)
12     I[m] ← 0  (deactivate cluster)
13  return A
```

Compute similarity matrix

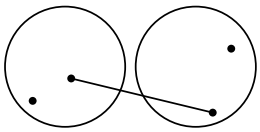
Find two clusters to merge

The i^{th} row and column of C are now the distances for the new cluster

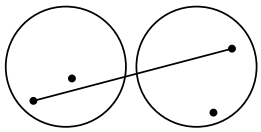
Similarity between cluster c_j and cluster $c_i \cup c_m$

► **Figure 17.2** A simple, but inefficient HAC algorithm.

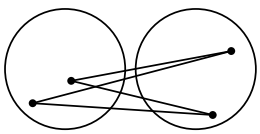
Agglomerative Clustering



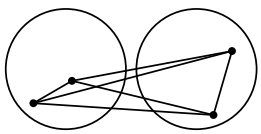
(a) single-link: maximum similarity



(b) complete-link: minimum similarity



(c) centroid: average inter-similarity

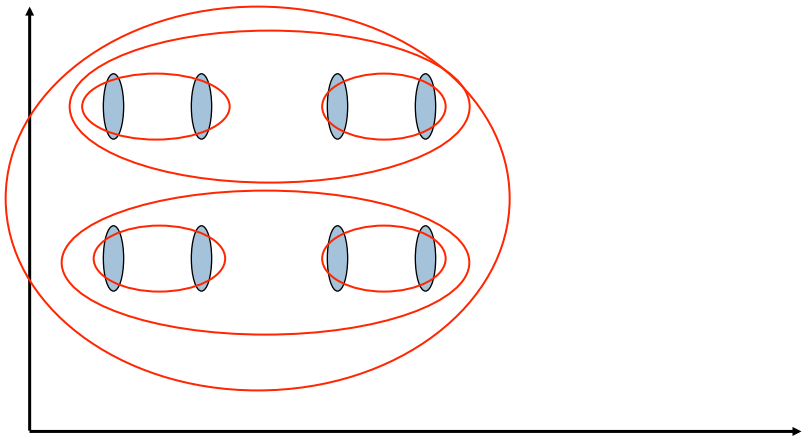


(d) group-average: average of all similarities

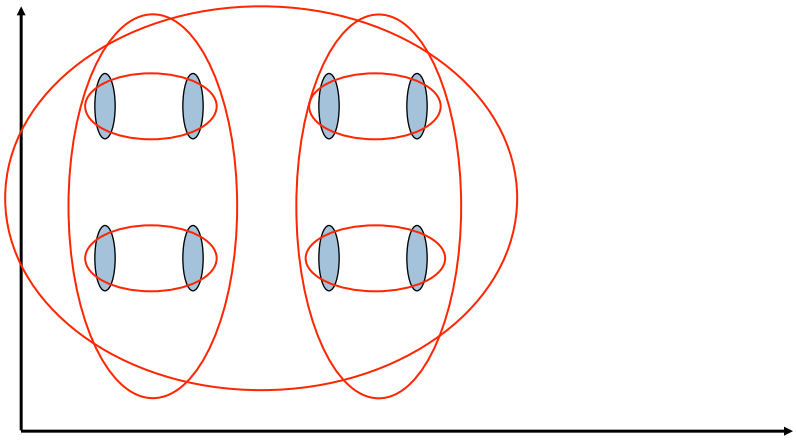
Cluster Similarity

- Single-link: similarity of most-similar elements
  - "long" clusters due to chaining
- Complete-link: similarity of least-similar elements
  - spherical clusters, sensitive to outliers
- Centroid: similarity of cluster centroids
- Group-average: average similarity btw all pairs

Cluster Similarity: Single-link



Cluster Similarity: Complete-link



Computational Complexity

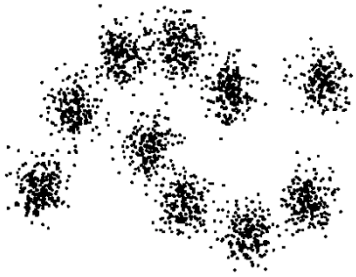
- In the first iteration, all HAC methods need to compute similarity of all pairs of N initial instances, which is $O(N^2)$.
- In each of the subsequent N-2 merging iterations, compute the distance between the most recently created cluster and all other existing clusters.
- In order to maintain an overall $O(N^2)$ performance, computing similarity to each other cluster must be done in constant time.
  - Often $O(N^3)$ if done in a naïve way
  - or $O(N^2 \log N)$ if done in a more clever way

Divisive Clustering

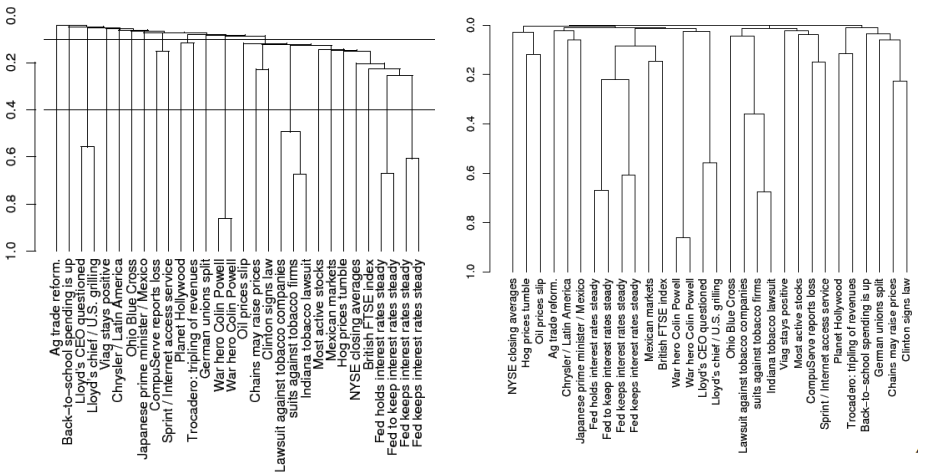
- Top-down clustering
- Divisive clustering algorithm uses a flat clustering algorithm as a subroutine
  - Start with all data points in one cluster
  - Split using a flat clustering algorithm
  - Apply recursively until each data point is in its own cluster
- Can be more efficient than agglomerative
- Benefits from complete information about the entire data set

Which clustering is correct?

- Different techniques cluster the same data set differently.
- Who is right? Is there a "right" clustering?



Which clustering is correct?

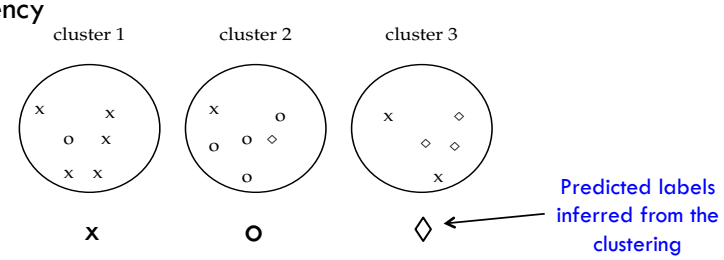


Which clustering is correct?

- Internal criteria
  - A good clustering has high intra-cluster similarity and low inter-cluster similarity
- External criteria
  - Use an external task (e.g. search, document classification) to validate the clustering
  - Requires labeled data

External Criteria

- Purity
  - Set aside labels from labeled data
  - Cluster data
  - Predicted label for each cluster is label with highest frequency



  - Compute accuracy: $\frac{5+4+3}{17} = 0.71$

External Criteria

- Normalized Mutual Information
  - Mutual Information is an information theoretic quantity similar to entropy and information gain

$$\mathrm{I}(X,Y) = \sum_y \sum_x p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = \mathrm{H}(X) - \mathrm{H}(X|Y)$$

  - How much information does the clustering contain about the class labels?

External Criteria

- Normalized Mutual Information
  - Define random variables for the clustering and for the class label:

$$I(\Omega, \mathcal{C}) = \sum_k \sum_j p(\omega_k \wedge c_j) \log \frac{p(\omega_k \wedge c_j)}{p(\omega_k)p(c_j)}$$

$$= \sum_k \sum_j \frac{|\omega_k \wedge c_j|}{N} \log\left(\frac{N|\omega_k \wedge c_j|}{|\omega_k||c_j|}\right)$$

R.V. for the cluster

R.V. for the class label

External Criteria

- Normalized Mutual Information
  - Given by the equation:

$$\mathrm{NMI}(\Omega, \mathcal{C}) = \frac{I(\Omega, \mathcal{C})}{\frac{1}{2} \cdot \left(H(\Omega) + H(\mathcal{C})\right)}$$

  - Why are we normalizing by the entropy?

Rand Index

- Two data points should be in the same cluster if and only if they have the same label
- Define contingency table:

	same cluster	different clusters
same class	true positives (TP)	false negatives (FN)
different classes	false positives (FP)	true negatives (TN)

- Once we have a contingency table, we can compute the Rand Index which is just the accuracy

$$RI = \frac{TP+TN}{TP+FP+FN+TN}$$

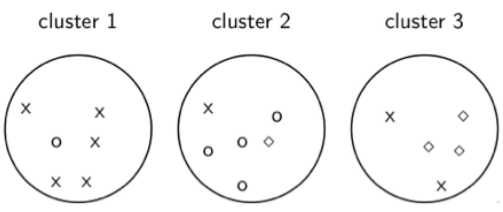
31

Rand Index Example

- There are $\binom{17}{2} = 136$ pairs of data points

	same cluster	diff. cluster
same class	20	24
diff class	20	72

RI = (20+72)/136 = 0.68



32

F-measure

- Given the contingency table, we can compute the precision, recall, and F-measure

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

- The parameter β controls the weighting between precision and recall

Clustering Evaluation

	purity	NMI	RI	F_5
lower bound	0.0	0.0	0.0	0.0
maximum	1.0	1.0	1.0	1.0
value for example	0.71	0.36	0.68	0.46

All four measures range from 0 (really bad clustering) to 1 (perfect clustering).

34