—INF4820—

Clustering

Erik Velldal

University of Oslo

Nov. 17, 2009

Topics for Today

- More on unsupervised machine learning for data-driven categorization: clustering.
  - The task of automatically grouping observations into categories.
  - A core set of tools within machine learning, data mining, and pattern recognition.
- An example of flat clustering: k-Means
- Hierarchical clustering
  - Agglomerative
  - Divisive
- Measuring the distance between clusters
  - Single-linkage, complete-linkage, average-linkage...
- Reading: Chapter 17 in Manning, Raghavan & Schütze (2008), *Introduction to Information Retrieval*; `http://informationretrieval.org/`.

Catching Up: Partitional Clustering

- Creates a flat, one-level grouping of the data.
- Can be defined as an *optimization problem*: Search for the partitioning of the data that minimizes some objective function.
  - Optimize a globally defined measure of partition quality.
- Problem: Exponentially many possible partitions of the data.
- An exhaustive search over partitions not feasible. Instead we must typically approximate the solution by iteratively refining an initial (possibly random) partition until some stopping criterion is met.

k-Means

- Unsupervised variant of the Rocchio classifier.
- Goal: Partition the n observed objects into k clusters C so that each point $\vec{x}_j$ belongs to the cluster c_i with the nearest centroid $\vec{\mu}_i$.
- Typically assumes Euclidean distance as the similarity function s.

k-Means

- Unsupervised variant of the Rocchio classifier.
- Goal: Partition the n observed objects into k clusters C so that each point $\vec{x}_j$ belongs to the cluster c_i with the nearest centroid $\vec{\mu}_i$.
- Typically assumes Euclidean distance as the similarity function s.
- The optimization problem: For each cluster, minimize the *within-cluster sum of squares*, $F_s = \mathrm{WCSS}$:

$$\mathrm{WCSS} = \sum_{c_i \in C} \sum_{\vec{x}_j \in c_i} \|\vec{x}_j - \vec{\mu}_i\|^2$$

- WCSS also amounts to the more general measure of how well a model fits the data known as the *residual sum of squares* (RSS).
- Minimizing WCSS is equivalent to minimizing the average squared distance between objects and their cluster centroids (since n is fixed), —a measure of how well each centroid represents the members assigned to the cluster.

k-Means (cont'd)

Algorithm

Initialize: Compute centroids for k random seeds.

Iterate:

- Assign each object to the cluster with the nearest centroid.
- Compute new centroids for the clusters.

Terminate: When stopping criterion is satisfied.

k-Means (cont'd)

Algorithm

Initialize: Compute centroids for k random seeds.

Iterate:

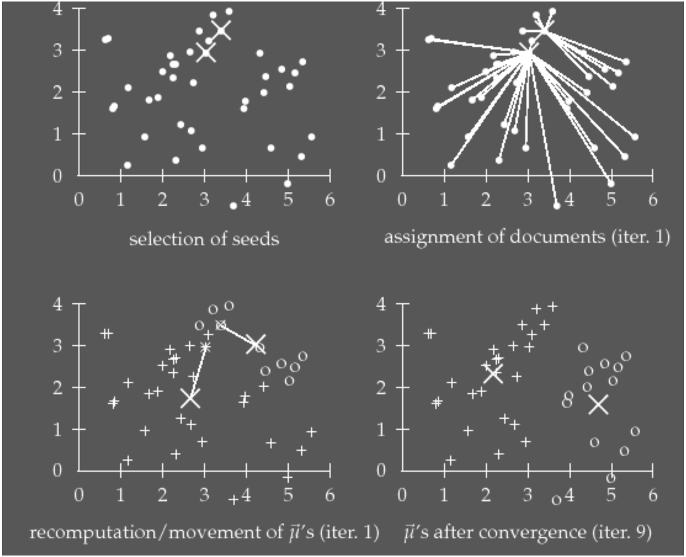
Assign each object to the cluster with the nearest centroid.
Compute new centroids for the clusters.

Terminate: When stopping criterion is satisfied.

Properties

- In short, we keep reassigning memberships and recomputing centroids until the configuration stabilizes.
- WCSS is monotonically decreasing (or unchanged) for each iteration.
- Guaranteed to converge but not to find the global minimum.
- The time complexity is linear, $\mathrm{O}(kn)$.

k-Means Example for $k = 2$ in R^2 (Manning, Raghavan & Schütze 2008)



Comments on k-Means

"Seeding"

- We initialize the algorithm by choosing random *seeds* that we use to compute the first set of centroids.
- Many ways to select the seeds:
  - pick k random objects from the collection;
  - pick k random points in the space;
  - pick k sets of m random points and compute centroids for each set;
  - compute an hierarchical clustering on a subset of the data to find k initial clusters; etc..

Comments on k-Means

"Seeding"

- We initialize the algorithm by choosing random *seeds* that we use to compute the first set of centroids.
- Many ways to select the seeds:
  - pick k random objects from the collection;
  - pick k random points in the space;
  - pick k sets of m random points and compute centroids for each set;
  - compute an hierarchical clustering on a subset of the data to find k initial clusters; etc..
- The heuristics involved in choosing the initial seeds can have a large impact on the resulting clustering (because we typically end up only finding a local minimum of the objective function).
- Outliers are troublemakers.

Comments on k-Means

Termination Criterion

- Fixed number of iterations
- Clusters or centroids are unchanged between iterations.
- Threshold on the decrease of the objective function (absolute or relative to previous iteration)

Comments on k-Means

Termination Criterion

- Fixed number of iterations
- Clusters or centroids are unchanged between iterations.
- Threshold on the decrease of the objective function (absolute or relative to previous iteration)

Some Close Relatives of k-Means

- k-Medoids: Like k-means but uses medoids instead of centroids to represent the cluster centers.

Comments on k-Means

Termination Criterion

- Fixed number of iterations
- Clusters or centroids are unchanged between iterations.
- Threshold on the decrease of the objective function (absolute or relative to previous iteration)

Some Close Relatives of k-Means

- k-Medoids: Like k-means but uses medoids instead of centroids to represent the cluster centers.
- Fuzzy c-Means (FCM): Like k-means but assigns soft memberships in $[0, 1]$, where membership is a function of the centroid distance.
  - The computations of both WCSS and centroids are weighted by the membership function.

Flat Clustering: The Good and the Bad

Pros

- Conceptually simple, and easy to implement.
- Efficient. Typically linear in the number of objects.

Cons

- The dependence on the random seeds makes the clustering *non-deterministic*.
- The number of clusters k must be pre-specified. Often no principled means of *a priori* specifying k.
- The clustering quality often considered inferior to that of the less efficient hierarchical methods.
- Not as informative as the more stuctured clusterings produced by hierarchical methods.

Hierarchical Clustering

Divisive methods

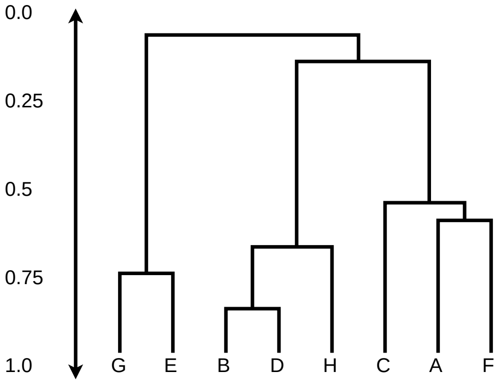
- Initially regards all k objects as part of a single cluster.
- Splits the groups top-down into smaller and smaller clusters.
- Each split defines a binary branch in the tree.
- Stops when k singleton clusters remain (unless other criterion defined).

Agglomerative methods

- Initially regards each object as its own singleton cluster.
- Iteratively merges (agglomerates) the groups in a bottom-up fashion.
- Each merge defines a binary branch in the tree.
- Stops when only one cluster remains containing all the objects (unless other criterion's defined).

Dendrograms

- A hierarchical clustering is often visualized as a binary tree structure known as a *dendrogram*.
- A merge is shown as a horizontal line connecting two clusters.
- The y-axis coordinate of the line corresponds to the *combination similarity* of the merged cluster.



- We here assume dot-products of normalized vectors; self-similarity $= 1$.

Agglomerative Clustering

parameters: $\{o_1, o_2 \ldots, o_n\}$, sim

$C = \{\{o_1\}, \{o_2\}, \ldots, \{o_n\}\}$
$T = []$
do for $i = 1$ **to** $n - 1$
$\quad \{c_j, c_k\} \leftarrow \underset{\{c_j,c_k\} \subseteq C \wedge j \neq k}{\arg\max} \; \mathrm{sim}(c_j, c_k)$
$\quad C \leftarrow C \backslash \{c_j, c_k\}$
$\quad C \leftarrow C \cup \{c_j \cup c_k\}$
$\quad T[i] \leftarrow \{c_j, c_k\}$

- At each stage, merge the pair of clusters that are most similar, as defined by some measure of *inter-cluster similarity*; sim.
- Plugging in a different sim gives us a different sequence of merges T.

Definitions of Inter-Cluster Similarity

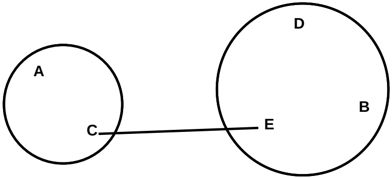
- So far we've looked at ways to the define the similarity between
  - pairs of objects.
  - objects and a class.
- Now we'll look at ways to define the similarity between classes or clusters.

Definitions of Inter-Cluster Similarity

- So far we've looked at ways to the define the similarity between
  - pairs of objects.
  - objects and a class.
- Now we'll look at ways to define the similarity between classes or clusters.
- In agglomerative clustering, a measure of cluster similarity $\text{sim}(c_i, c_j)$ is usually referred to as a *linkage criterion* (from graph theory):
  - Single-linkage
  - Complete-linkage
  - Centroid-linkage
  - Average-linkage
- The linkage criterion determines which pair of clusters we will merge to a new cluster in each step.

Single-Linkage

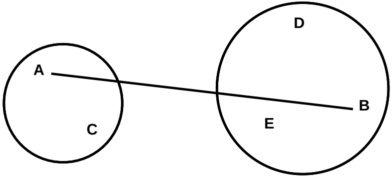
- Merge the two clusters with the minimum distance between any two members.
- Nearest-Neighbors.



- Can be computed efficiently by taking advantage of the fact that it's *best-merge persistent*:
  - The distance of the two closest members is a local property that is not affected by merging.
  - Let the nearest neighbor of cluster c_k be in either c_i or c_j. If we merge $c_i \cup c_j = c_l$, the nearest neighbor of c_k will be in c_l.
- Undesirable chaining effect: Tendency to produce to 'stretched' and 'straggly' clusters.

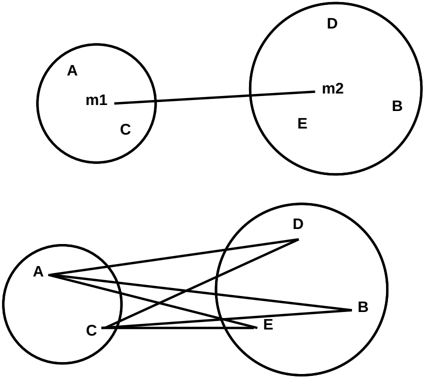
Complete-Linkage

- Merge the two clusters where the maximum distance between any two members is smallest.
- Farthest-Neighbors.



- Amounts to merging the two clusters whose merger has the smallest diameter.
- Preference for compact clusters with small diameters.
- Sensitive to outliers.
- Not best-merge persistent: Distance defined as the diameter of a merge is a non-local property that can change during merging.

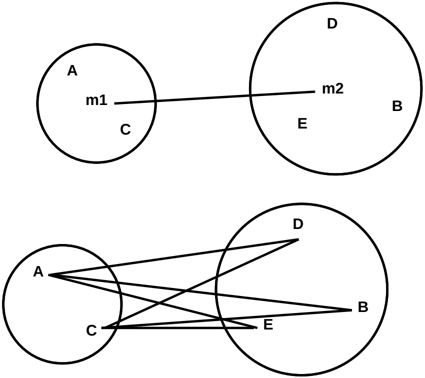
Centroid-Linkage



- Similarity of two clusters c_i and c_j defined as the similarity between their cluster centroids $\vec{\mu}_i$ and $\vec{\mu}_j$ (the mean vectors).
- Equivalent to the average pairwise similarity between objects from different clusters:

$$sim(c_i, c_j) = \vec{\mu_i} \cdot \vec{\mu_j} = \frac{1}{|c_i||c_j|} \sum_{\vec{x} \in c_i} \sum_{\vec{y} \in c_j} \vec{x} \cdot \vec{y}$$

Centroid-Linkage



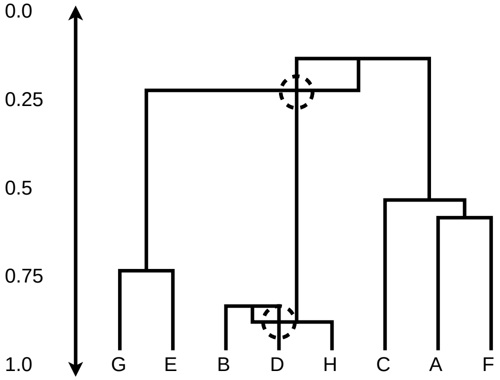
- Similarity of two clusters c_i and c_j defined as the similarity between their cluster centroids $\vec{\mu}_i$ and $\vec{\mu}_j$ (the mean vectors).
- Equivalent to the average pairwise similarity between objects from different clusters:

$$sim(c_i, c_j) = \vec{\mu_i} \cdot \vec{\mu_j} = \frac{1}{|c_i||c_j|} \sum_{\vec{x} \in c_i} \sum_{\vec{y} \in c_j} \vec{x} \cdot \vec{y}$$

- Like complete-link, not best-merge persistent.
- However, unlike the other linkage criterions, it is not monotonic and subject to s.c. *inversions*: The combination similarity can increase during the clustering.

Inversions —A Problem with Centroid-Linkage

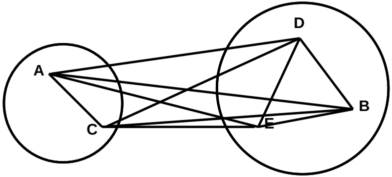
- We usually assume that the clustering is *monotonic*, i.e. the combination similarity is guaranteed to *decrease* between iterations.
- For a *non-monotonic* clustering criterion (e.g. centroid-linkage), *inversions* would show in the dendrogram as crossing lines.



- The dotted circles in the dendrogram above indicate inversions: The horizontal merge bar is lower than the bar of a previous merge.
- Violates the fundamental assumption that small clusters are more coherent than large clusters.

Average-Linkage

- AKA group average agglomerative clustering.
- Merge the two clusters where the average of all pairwise similarities in their union is highest.



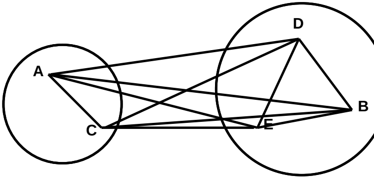
- Aims to maximize the coherency of the merged cluster by considering all pairwise similarities between the objects in the clusters.
- Let $c_i \cup c_j = c_k$, and $sim(c_i, c_j) = W(c_i, \cup c_j) = W(c_k)$:

$$W(c_k) = \frac{1}{|c_k|(|c_k| - 1)} \sum_{\vec{x} \in c_k} \sum_{\vec{x} \neq \vec{y} \in c_k} \vec{x} \cdot \vec{y}$$

- Self-similarities are excluded in order to not penalize large clusters (which have fewer self-similarities).

Average-Linkage (cont'd)

- But not best-merge persistent.
- Compromise of complete- and single-linkage.

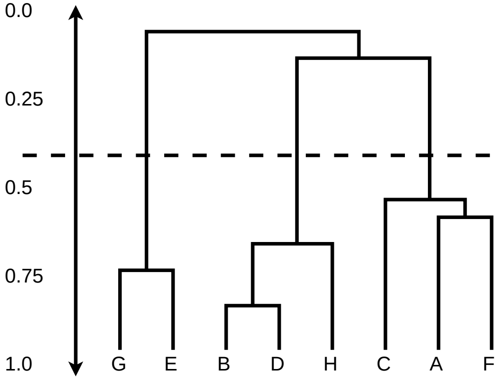


- Commonly considered the best "default" linkage criterion for agglomerative clustering.
- Can be computed very efficiently if we assume normalized vectors and that the similarity measure of the feature vectors s = dot-product:

$$W(c_k) = \frac{1}{|c_k|(|c_k| - 1)} \left((\sum_{\vec{x} \in c_k} \vec{x})^2 - |c_k| \right)$$

Cutting the Tree

- Hierarchical methods actually produce *several partitions*; one for each level of the tree.
- However, for many applications we will want to extract a set of disjoint clusters.
- In order to turn the nested partitions into a single flat partitioning, we cut the dendrogram.



- A cutting criterion can be defined as a threshold on e.g. combination similarity, relative drop in the similarity, number of root nodes, etc.

Divisive Hierarchical Clustering

- Generates the nested partitions top-down:
  - Start by considering all objects part of the same cluster (the root).
  - Split the cluster using *a flat clustering algorithm* (e.g. by applying k-means for $k = 2$).
  - Recursively split the clusters until only singleton clusters remain.
  - (Also possible to fix the desired levels and stop the clustering before we reach the singleton leaves.)
- Flat methods such as k-means are generally very effective; *linear* in the number of objects.
- Divisive methods are thereby also generally more efficient than agglomerative methods, which are *at least quadratic* (single-link).
- Also has the advantage of being able to initially consider the global distribution of the data, while the agglomerative methods must commit to early decisions based on local patterns.

Next (and Final) Week

- Summing up.
- Sample exam.