— INF4820 —
Algorithms for AI and NLP

Hierarchical Clustering

Erik Velldal & Stephan Oepen

Language Technology Group (LTG)

October 7, 2015

Agenda

Last week

- Evaluation of classifiers
- Machine learning for class discovery: Clustering
  - Unsupervised learning from unlabeled data.
  - Automatically group similar objects together.
  - No pre-defined classes: we only specify the similarity measure.
- Flat clustering, with k-means.

Today

- Hierarchical clustering
  - Top-down / divisive
  - Bottom-up / agglomerative
- Crash course on probability theory
- Language modeling

Agglomerative clustering

- Initially: regards each object as its own singleton cluster.
- Iteratively 'agglomerates' (merges) the groups in a bottom-up fashion.
- Each merge defines a binary branch in the tree.
- Terminates: when only one cluster remains (the root).

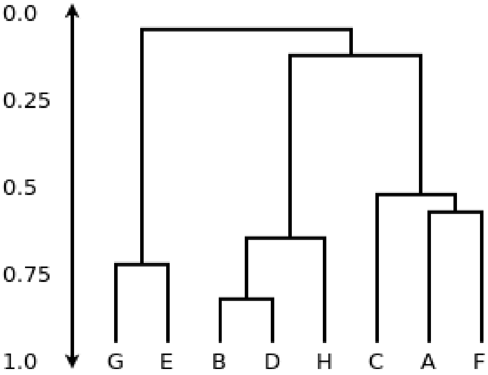
parameters: $\{o_1, o_2, \ldots, o_n\}$, sim

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

- At each stage, we 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.

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 similarity of the merged clusters.



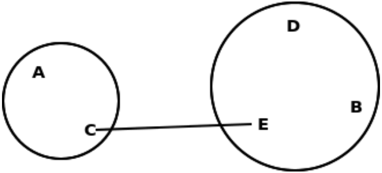
- We here assume dot-products of normalized vectors (self-similarity $= 1$).

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 collections.
- In agglomerative clustering, a measure of cluster similarity $\text{sim}(c_i, c_j)$ is usually referred to as a linkage criterion:
  - Single-linkage
  - Complete-linkage
  - Average-linkage
  - Centroid-linkage
- Determines the pair of clusters to merge in each step.

Single-linkage

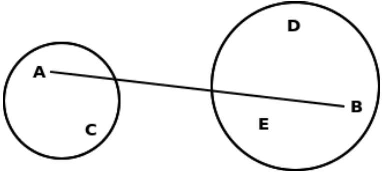
- Merge the two clusters with the minimum distance between any two members.
- 'Nearest neighbors'.



- Can be computed efficiently by taking advantage of the fact that it's best-merge persistent:
  - 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.
  - The distance of the two closest members is a local property that is not affected by merging.
- Undesirable chaining effect: Tendency to produce 'stretched' and 'straggly' clusters.

Complete-linkage

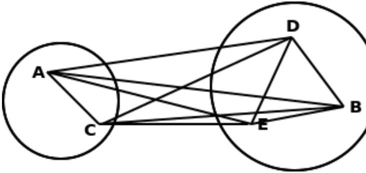
- Merge the two clusters where the maximum distance between any two members is smallest.
- 'Farthest neighbors'.



- 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.

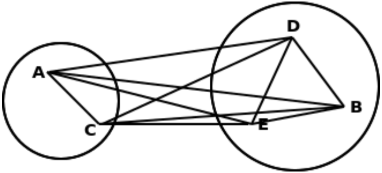
Average-linkage (1:2)

- AKA group-average agglomerative clustering.
- Merge the clusters with the highest average pairwise similarities in their union.



- Aims to maximize coherency by considering all pairwise similarities between objects within the cluster to merge (excluding self-similarities).
- Compromise of complete- and single-linkage.
- Not best-merge persistent.
- Commonly considered the best default clustering criterion.

- Can be computed very efficiently if we assume (i) the *dot-product* as the similarity measure for (ii) *normalized* feature vectors.

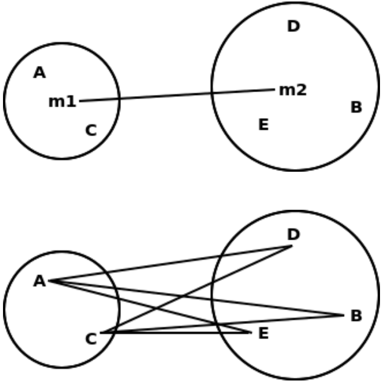


- Let $c_i \cup c_j = c_k$, and $sim(c_i, c_j) = W(c_i \cup c_j) = W(c_k)$, then $W(c_k) =$

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

- The sum of vector similarities is equal to the similarity of their sums.

Centroid-linkage



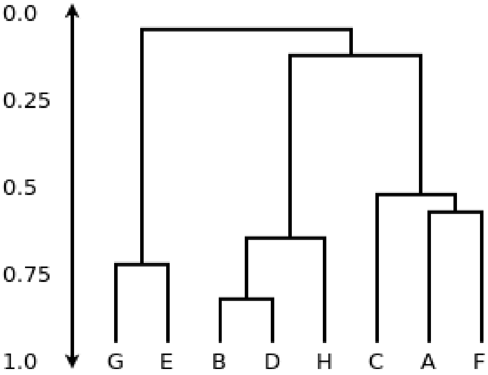
- Similarity of clusters c_i and c_j defined as the similarity of their cluster centroids $\vec{\mu}_i$ and $\vec{\mu}_j$.
- Equivalent to the average pairwise similarity between objects from different clusters:

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

- Not best-merge persistent.
- Not monotonic, subject to inversions: The combination similarity can increase during the clustering.

Monotinicity

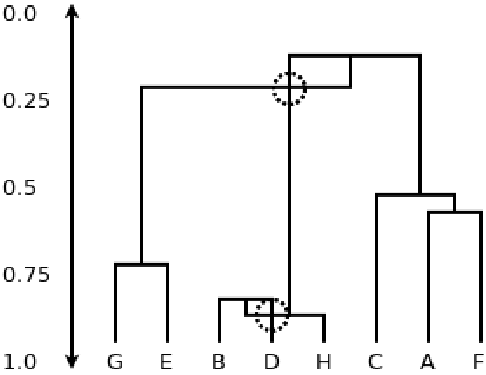
- A fundamental assumption in clustering: small clusters are more coherent than large.
- We usually assume that a clustering is monotonic:
- Similarity is *decreasing* from iteration to iteration.



- This assumpion holds true for all our clustering criterions except for centroid-linkage.

Inversions – a problem with centroid-linkage

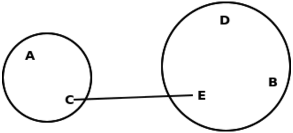
- Centroid-linkage is non-monotonic.
- We risk seeing so-called inversions:
- Similarity can increase during the sequence of clustering steps.
- Would show as crossing lines in the dendrogram.



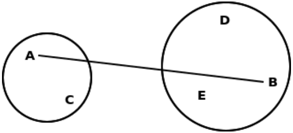
- The horizontal merge bar is lower than the bar of a previous merge.

Linkage criterions

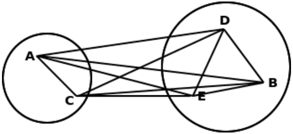
Single-link



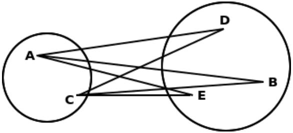
Complete-link



Average-link



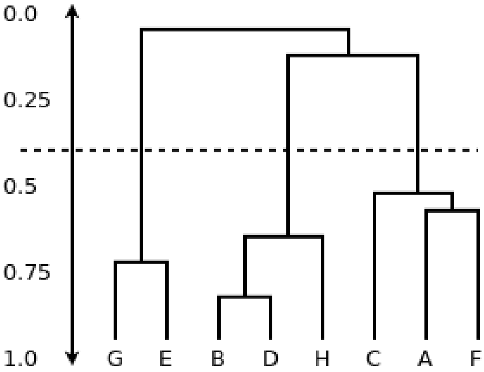
Centroid-link



- All the linkage criterions can be computed on the basis of the object similarities; the input is typically a proximity matrix.

Cutting the tree

- The tree actually represents several partitions:
- one for each level.
- If we want to turn the nested partitions into a single flat partitioning. . .
- we must cut the tree.



- 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: all objects considered 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 (or some specified number of levels is reached).

- Flat methods are generally very effective (e.g. k-means is linear in the number of objects).
- Divisive methods are thereby also generally more efficient than agglomerative, which are at least quadratic (single-link).
- Also able to initially consider the global distribution of the data, while the agglomerative methods must commit to early decisions based on local patterns.

INF4820: Algorithms for Artificial Intelligence and Natural Language Processing

Basic Probability Theory & Language Models

Stephan Oepen & Erik Velldal

Language Technology Group (LTG)

October 7, 2015

Changing of the Guard

So far: Point-wise classification; geometric models.

Next: Structured classification; probabilistic models.

- sequences
- labelled sequences
- trees

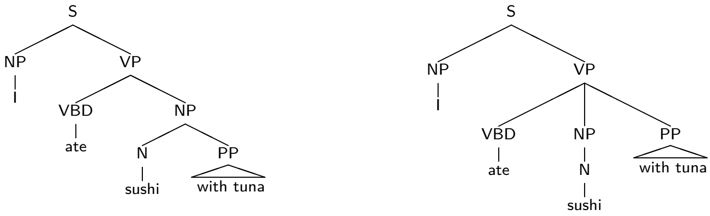
Kristian (December 10, 2014)

Guro (March 16, 2015)

By the End of the Semester . . .

. . . you should be able to determine

- which string is most likely:
  - *How to recognise speech* vs. *How to wreck a nice beach*
- which category sequence is most likely for *flies like an arrow*:
  - **N V D N** vs. **V P D N**
- which syntactic analysis is most likely:



Probability Basics (1/4)

- Experiment (or trial)
  - the process we are observing
- Sample space (Ω)
  - the set of all possible outcomes
- Event(s)
  - the subset of Ω we are interested in

$P(A)$ is the probability of event A, a real number $\in [0, 1]$

Probability Basics (2/4)

- Experiment (or trial)
  - rolling a die
- Sample space (Ω)
  - $\Omega = \{1, 2, 3, 4, 5, 6\}$
- Event(s)
  - $A =$ rolling a six: $\{6\}$
  - $B =$ getting an even number: $\{2, 4, 6\}$

$P(A)$ is the probability of event A, a real number $\in [0, 1]$

Probability Basics (3/4)

- Experiment (or trial)
  - flipping two coins
- Sample space (Ω)
  - $\Omega = \{HH, HT, TH, TT\}$
- Event(s)
  - $A =$ the same both times: $\{HH, TT\}$
  - $B =$ at least one head: $\{HH, HT, TH\}$

$P(A)$ is the probability of event A, a real number $\in [0, 1]$

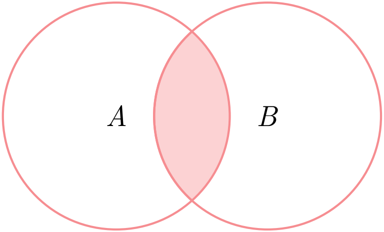
Probability Basics (4/4)

- Experiment (or trial)
  - rolling two dice
- Sample space (Ω)
  - $\Omega = \{11, 12, 13, 14, 15, 16, 21, 22, 23, 24, \ldots, 63, 64, 65, 66\}$
- Event(s)
  - $A =$ results sum to 6: $\{15, 24, 33, 42, 51\}$
  - $B =$ both results are even: $\{22, 24, 26, 42, 44, 46, 62, 64, 66\}$

$P(A)$ is the probability of event A, a real number $\in [0, 1]$

Joint Probability

- $P(A, B)$: probability that both A and B happen
- also written: $P(A \cap B)$

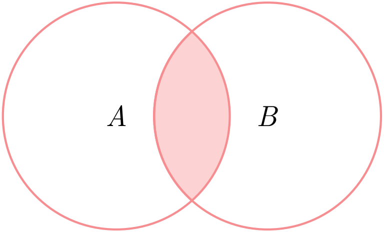


What is the probability, when throwing two fair dice, that

- A: the results sum to 6 and
- B: at least one result is a 1?

Joint Probability

- $P(A, B)$: probability that both A and B happen
- also written: $P(A \cap B)$

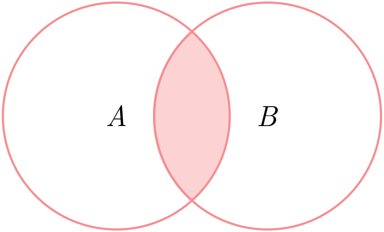


What is the probability, when throwing two fair dice, that

- A: the results sum to 6 and
- B: at least one result is a 1?

Joint Probability

- $P(A, B)$: probability that both A <u>and</u> B happen
- also written: $P(A \cap B)$

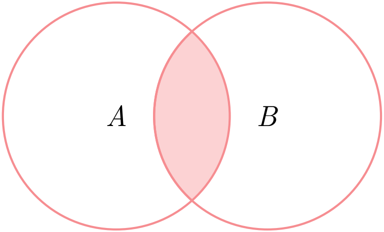


What is the probability, when throwing two fair dice, that

- A: the results sum to 6 <u>and</u> $\frac{5}{36}$
- B: at least one result is a 1?

Joint Probability

- $P(A, B)$: probability that both A and B happen
- also written: $P(A \cap B)$



What is the probability, when throwing two fair dice, that

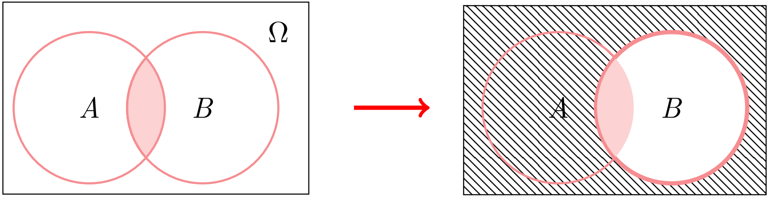
- A: the results sum to 6 and $\frac{5}{36}$
- B: at least one result is a 1? $\frac{11}{36}$

Conditional Probability

Often, we know something about a situation.

What is the probability $P(A|B)$, when throwing two fair dice, that

- A: the results sum to 6 given
- B: at least one result is a 1?



$$\boxed{P(A|B) = \frac{P(A \cap B)}{P(B)} \quad \text{(where } P(B) > 0\text{)}}$$

The Chain Rule

Joint probability is symmetric:

$$\begin{aligned} P(A \cap B) &= P(A)\,P(B|A) \\ &= P(B)\,P(A|B) \quad \text{(multiplication rule)} \end{aligned}$$

More generally, using the chain rule:

$$P(A_1 \cap \cdots \cap A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2)\ldots P(A_n| \cap_{i=1}^{n-1} A_i)$$

The chain rule will be very useful to us through the semester:

- it allows us to break a complicated situation into parts;
- we can choose the breakdown that suits our problem.

(Conditional) Independence

If knowing event B is true has no effect on event A, we say

A and B are independent of each other.

If A and B are independent:

- $P(A) = P(A|B)$
- $P(B) = P(B|A)$
- $P(A \cap B) = P(A)\,P(B)$

Intuition? (1/3)

Let's say we have a rare disease, and a pretty accurate test for detecting it. Yoda has taken the test, and the result is positive.

The numbers:

- disease prevalence: 1 in 1000 people
- test false negative rate: 1%
- test false positive rate: 2%

What is the probability that he has the disease?

Intuition? (2/3)

Given:

- event A: have disease
- event B: positive test

We know:

- $P(A) = 0.001$
- $P(B|A) = 0.99$
- $P(B|\neg A) = 0.02$

We want

- $P(A|B) = ?$

Intuition? (3/3)

	A	¬ A	
B	0.00099	0.01998	0.02097
¬ B	0.00001	0.97902	0.97903
	0.001	0.999	1

$$P(A) = 0.001; \quad P(B|A) = 0.99; \quad P(B|\neg A) = 0.02$$

$$P(A \cap B) = P(B|A)P(A)$$

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{0.00099}{0.02097} = 0.0472$$

Bayes' Theorem

- From the two 'symmetric' sides of the joint probability equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- reverses the order of dependence (which can be useful)
- in conjunction with the chain rule, allows us to determine the probabilities we want from the probabilities we know

Other useful axioms

- $P(\Omega) = 1$
- $P(A) = 1 - P(\neg A)$

Bonus: The Monty Hall Problem

- On a gameshow, there are three doors.
- Behind 2 doors, there is a goat.
- Behind the 3rd door, there is a car.
- The contestant selects a door that she hopes has the car behind it.
- Before she opens that door, the gameshow host opens one of the other doors to reveal a goat.
- The contestant now has the choice of opening the door she originally chose, or switching to the other unopened door.

What should she do?

Coming up Next

- Do you want to come to the movies and ___?___
- Det var en ___?___
- Je ne parle pas ___?___

Natural language contains redundancy, hence can be predictable.

Previous context can constrain the next word

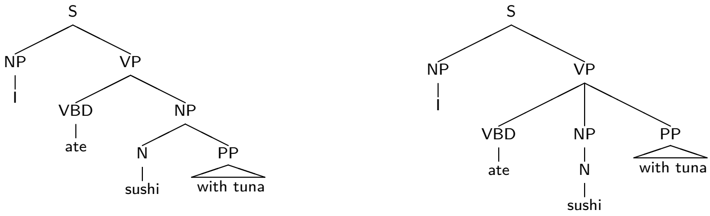
- semantically;
- syntactically;

→ by frequency.

Recall: By the End of the Semester . . .

. . . you should be able to determine

- which string is most likely:
  - *How to recognise speech* vs. *How to wreck a nice beach*
- which category sequence is most likely for *flies like an arrow*:
  - **N V D N** vs. **V P D N**
- which syntactic analysis is most likely:



Language Models

- A probabilistic (also known as stochastic) language model M assigns probabilities $P_M(x)$ to all strings x in language L.
  - L is the sample space
  - $0 \leq P_M(x) \leq 1$
  - $\sum_{x \in L} P_M(x) = 1$
- Language models are used in machine translation, speech recognition systems, spell checkers, input prediction, ...
- We can calculate the probability of a string using the chain rule:

$$P(w_1 \dots w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1 \cap w_2) \dots P(w_n| \cap_{i=1}^{n-1} w_i)$$

$P($*I want to go to the beach*$) =$
$P($*I*$)\ P($*want*$|$*I*$)\ P($*to*$|$*I want*$)\ P($*go*$|$*I want to*$)\ P($*to*$|$*I want to go*$) \dots$

N-Grams

We simplify using the Markov assumption (limited history):

the last $n-1$ elements can approximate the effect of the full sequence.

That is, instead of

- $P($*beach*$|$ *I want to go to the*$)$

selecting an n of 3, we use

- $P($*beach*$|$ *to the*$)$

We call these short sequences of words n-grams:

- bigrams: *I want*, *want to*, *to go*, *go to*, *to the*, *the beach*
- trigrams: *I want to*, *want to go*, *to go to*, *go to the*
- 4-grams: *I want to go*, *want to go to*, *to go to the*