582631 — 5 credits

Introduction to Machine Learning

Lecturer: Teemu Roos
Assistant: Ville Hyvönen

Department of Computer Science
University of Helsinki

(based in part on material by Patrik Hoyer and Jyrki Kivinen)

November 1st–December 16th 2016

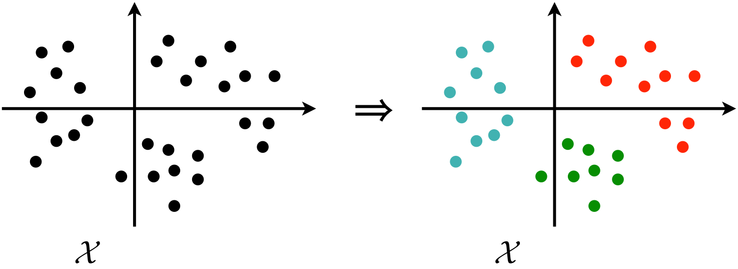
Clustering

Flat clustering: basic idea

- Each data vector $\mathbf{x}_i$ is assigned to one of K clusters

- Typically K and a similarity measure is selected by the user, while the chosen algorithm then learns the actual partition

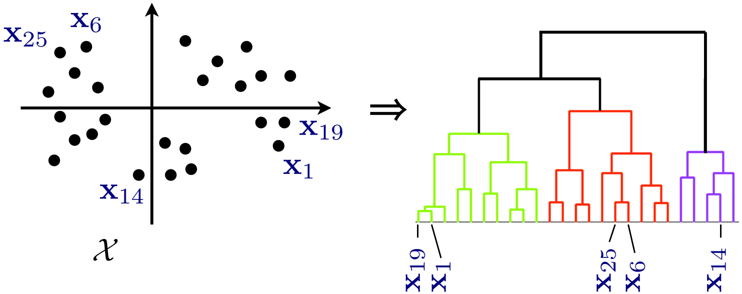
- In the example below, $K = 3$ and the partition are shown using color (red, green, blue)



Flat clustering: basic idea (2)

- In **distance-based clustering**
  - data points in same cluster are similar to (near) each other
  - data points in different clusters are dissimilar (far away) from each other
- A common strategy is to represent the clusters as K *prototypes* $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ and assigning each data point to the closest prototype
  - This can also be done for new ("test") data points by assign each new point to the nearest prototype
- Distances can be in principle anything but many methods are well defined only for *metric* distances
- Alternative: In a probabilistic approach, similarity (nearness) is replace by probability and prototypes are distributions

Hierarchical clustering: basic idea



- In this approach, data vectors are arranged in a tree, where nearby (similar) vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ should be placed close to each other: e.g., $\mathbf{x}_6$ and $\mathbf{x}_{25}$ end up being siblings while $\mathbf{x}_{14}$ is a distant cousin

- Any horizontal cut corresponds to a partitional clustering

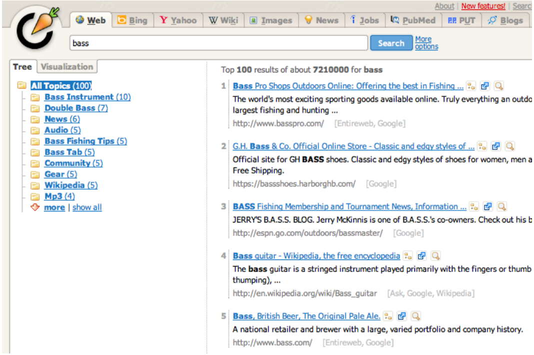
- In the example above, the 3 colors have been added manually for emphasis (they are *not* produced by the algorithm)

Motivation for clustering

Understanding the data:

- Information retrieval:

  organizing a set of documents for easy browsing (for example a hierarchical structure to the documents)



All Topics (100)
- Bass Instrument (10)
- Double Bass (7)
- News (6)
- Audio (5)
- Bass Fishing Tips (5)
- Bass Tab (5)
- Community (5)
- Gear (5)
- Wikipedia (5)
- Mp3 (4)
- more | show all

Top **100** results of about **7210000** for **bass**

1 **Bass** Pro Shops Outdoors Online: Offering the best in Fishing ...
The world's most exciting sporting goods available online. Truly everything an outdo
largest fishing and hunting ...
http://www.basspro.com/ [Entireweb, Google]

2 G.H. **Bass** & Co. Official Online Store - Classic and edgy styles of ...
Official site for GH **BASS** shoes. Classic and edgy styles of shoes for women, men a
Free Shipping.
https://bassshoes.harborghb.com/ [Google]

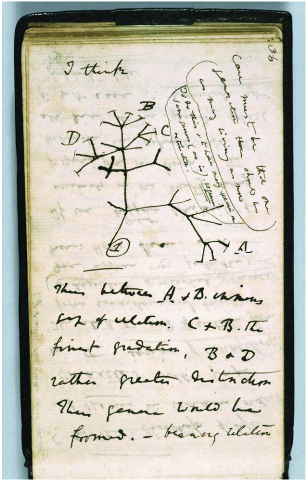
3 **BASS** Fishing Membership and Tournament News, Information ...
JERRY'S B.A.S.S. BLOG. Jerry McKinnis is one of B.A.S.S.'s co-owners. Check out his b
http://espn.go.com/outdoors/bassmaster/ [Google]

4 **Bass** guitar - Wikipedia, the free encyclopedia
The **bass** guitar is a stringed instrument played primarily with the fingers or thumb
thumping), ...
http://en.wikipedia.org/wiki/Bass_guitar [Ask, Google, Wikipedia]

5 **Bass**, British Beer, The Original Pale Ale.
A national retailer and brewer with a large, varied portfolio and company history.
http://www.bass.com/ [Entireweb, Google]

- Biology:

  creating a taxonomy of species (*phylogenetics*), finding groups of genes with similar function, etc



36

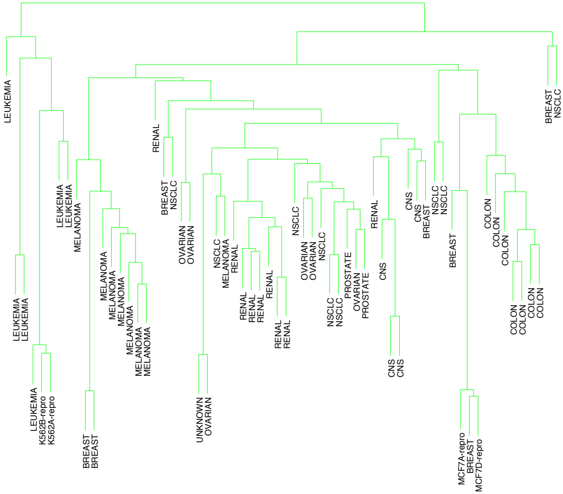
I think

Case must be that one generation then should be as many living as now. To do this & to have many species in same genus (as is). requires extinction.

Thus between A & B. immense gap of relation. C & B. the finest gradation, B & D rather greater distinction. Thus genera would be formed. — bearing relation

- Medicine:

  understanding the relations among diseases or psychological conditions, to aid in discovering the most useful treatments



- Business:

  grouping customers by their preferences or shopping behavior, for instance for targeted advertisement campaigns

  Et cetera, et cetera

- Other motivations: simplifying the data for further processing/transmission
    - Micro-clustering for Big Data:

      reduce the effective amount of data by considering only the prototypes rather than the original data vectors
    - Quantization (lossy compression):

      saving disk space/bandwidth by only representing each point by a 'close enough' prototype

Distance-based clustering

- We are given a data set $D = \{\, \mathbf{x}_1, \ldots, \mathbf{x}_n \,\} \subset \mathcal{X}$ and a notion of similarity between elements of $\mathcal{X}$

- The output will be a *partition* $(D_1, \ldots, D_K)$ of D:
  - $D_1 \cup \cdots \cup D_K = D$
  - $D_i \cap D_j = \emptyset$ if $i \neq j$

- Alternatively, we can represent the partition by giving an assignment mapping where $j(i) = c$ if $\mathbf{x}_i \in D_c$

- We usually also output K exemplars $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ where each data point is assigned to the cluster with closest exemplar

- number of clusters K is usually given as input; choosing a "good" K is a separate (non-trivial) issue

K-means

- The most popular distance-based clustering method is *K-means*

- We specifically assume that $\mathcal{X} = \mathbb{R}^p$ and use squared Euclidean distance as dissimilarity measure

- Ideally, we would wish to find partition and exemplars that minimise the total distance of data points from their assigned exemplars

$$\sum_{j=1}^{K} \sum_{\mathbf{x} \in D_j} \|\mathbf{x} - \boldsymbol{\mu}_j\|_2^2 = \sum_{i=1}^{n} \left\|\mathbf{x}_i - \boldsymbol{\mu}_{j(i)}\right\|_2^2$$

- However minimising this exactly is computationally difficult (NP-hard) so in practice we usually use heuristic algorithms

Hard vs. soft clustering

- In *soft clustering* we assign to each pair $\mathbf{x}_i$ and D_j a number $r_{ij} \in [0, 1]$ so that $\sum_{j=1}^{K} r_{ij} = 1$ for all i, and then minimise

$$\sum_{i=1}^{n} \sum_{j=1}^{K} r_{ij} \left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|_2^2$$

- *Hard clustering*, which we discuss here, is the special case where we require that for each i there is exactly one $j(i)$ such that $r_{i,j(i)} = 1$, and $r_{ij} = 0$ for $j \neq j(i)$

- Note that the optimum assignments are always hard, i.e., $r_{i,j(i)} = 1$ for some $j(i)$

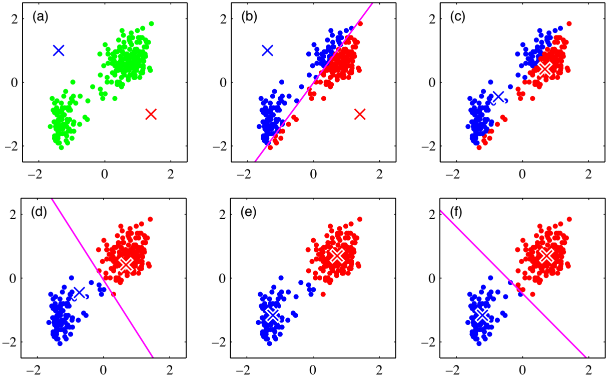
K-means algorithm

- We start by picking K initial cluster exemplars (for example randomly from our data set)
- We then alternate between the following two steps, until nothing changes any more:
  - Keeping the examplars fixed, assign each data point to the closest exemplar
  - Keeping the assignments fixed, move each exemplar to the center of its assigned data points
- In this context we call the exemplars *cluster means*. Notice that generally they are **not** sample points in our data set, but can be arbitrary vectors in $\mathbb{R}^d$
- This is also known as Lloyd's algorithm; see Algorithm 10.1 in textbook

K-means algorithm: pseudocode

- **Input**
  - data set $D = \{ x_1, \ldots, x_n \} \subset \mathbb{R}^p$
  - number of clusters K
- **Output**
  - partition $D_1, \ldots, D_K$
  - cluster means (exemplars) $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
  - assignment mapping $j \colon \{ 1, \ldots, n \} \to \{ 1, \ldots, K \}$
- **Algorithm**
  - Randomly choose initial $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
  - Repeat the following until $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ do not change:
    - for $i = 1, \ldots, n$: let $j(i) \leftarrow \arg\min_j \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2$
    - for $j = 1, \ldots, K$: let $D_j \leftarrow \{ \mathbf{x}_i \mid j(i) = j \}$
    - for $j = 1, \ldots, K$: let $\boldsymbol{\mu}_j \leftarrow \frac{1}{|D_j|} \sum_{\mathbf{x} \in D_j} \mathbf{x}_i$

K-means: 2D example



- Data from the ‘Old faithful’ geyser (horizontal axis is duration of eruption, vertical axis is waiting time to the next eruption, both scaled to zero mean and unit variance)

K-means: convergence

- We can show that the algorithm is guaranteed to converge after some finite number of steps
- We look into changes of the cost function

$$\text{Cost} = \sum_{j=1}^{K} \sum_{\mathbf{x} \in D_j} \|\mathbf{x} - \boldsymbol{\mu}_j\|_2^2 = \sum_{i=1}^{n} \left\|\mathbf{x}_i - \boldsymbol{\mu}_{j(i)}\right\|_2^2$$

at the two steps inside the main loop

  - In first step, we assign each $\mathbf{x}_i$ to $j(i)$ such that $\left\|\mathbf{x}_i - \boldsymbol{\mu}_{j(i)}\right\|_2^2$ is minimised
  - In second step, we choose each $\boldsymbol{\mu}_j$ as the mean of D_j, which minimises $\sum_{\mathbf{x} \in D_j} \|\mathbf{x} - \boldsymbol{\mu}_j\|_2^2$ for a fixed D_j
    - Showing that choosing the mean vector minimizes the sum of squared errors is left as homework
  - Hence, the cost never increases

K-means: convergence (2)

- Based on the homework property (previous slide), the minimum Cost can be computed given the cluster assignments
- There is a finite number K^n possible assignments, so there is only a finite number of possible values for Cost
- Since Cost is non-increasing, it must eventually stabilise to one value
- Notice that the value to which we converge
  - is not guaranteed to be global optimum of Cost
  - depends on initialisation of cluster means
- In practice, convergence usually takes a lot fewer than K^n steps

Space and time complexity

- Space requirements are modest, as (in addition to the data itself) we only need to store:
  1. The index of the assigned cluster for each datapoint $\mathbf{x}_i$
  2. The cluster centroid for each cluster

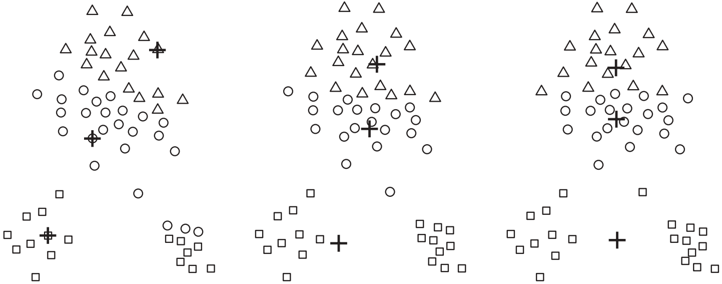
- The running time is linear in all the relevant parameters, i.e. $O(MnKp)$, where M is the number of iterations, n the number of samples, K the number of clusters, and p the number of dimensions (attributes).

  (The number of iterations M typically does not depend heavily on the other parameters.)

Influence of initialization

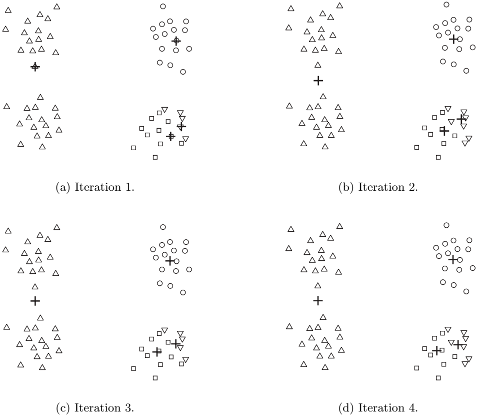
- The algorithm only guarantees that cost is non-increasing. It is still local search, and does *not* in general reach the global minimum.

  Example 1:



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3.

Example 2:



(a) Iteration 1.

(b) Iteration 2.

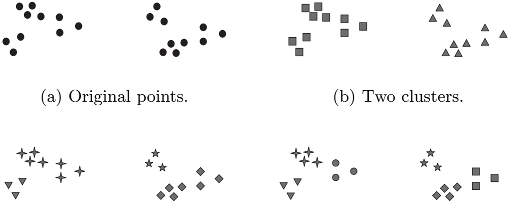
(c) Iteration 3.

(d) Iteration 4.

- One possible solution: Run the algorithm from many random initial conditions, select the end result with the smallest cost. (Nevertheless, it may still find very ‘bad’ solutions almost all the time.)

How to select the number of clusters?

- Not a priori clear what the 'optimal' number of clusters is:



(a) Original points.

(b) Two clusters.

(c) Four clusters.

(d) Six clusters.

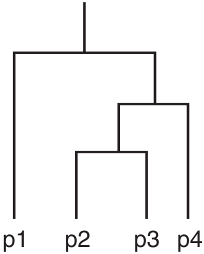
- The more clusters, the lower the cost, so need some form of 'model selection' approach

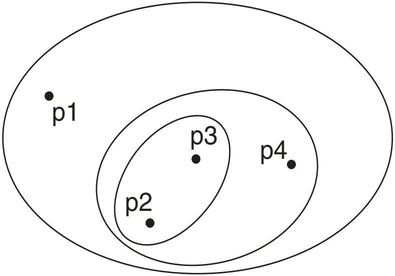
- Will discuss this a bit more in the context of clustering validation strategies later

Hierarchical clustering

- Dendrogram representation:
  - *Nested* cluster structure
  - Binary tree with datapoints (objects) as leaves
  - Cutting the tree at any height produces a partitional clustering

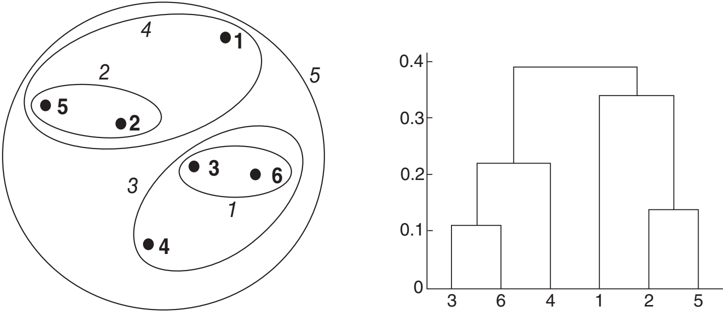
- Example 1:





Hierarchical clustering (2)

- Example 2:



- Height of horizontal connectors indicate the dissimilarity between the combined clusters (details a bit later)

Hierarchical clustering (3)

General approaches to hierarchical clustering:

- Divisive approach:
  1. Start with one cluster containing all the datapoints.
  2. Repeat for all non-singleton clusters:
     - Split the cluster in two using some partitional clustering approach (e.g. K-means)

- Agglomerative approach:
  1. Start with each datapoint being its own cluster
  2. Repeat until there is just one cluster left:
     - Select the pair of clusters which are most similar and join them into a single cluster

(The agglomerative approach is much more common, and we will exclusively focus on it in what follows.)

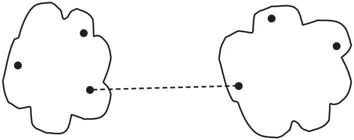
Linkage functions

- Agglomerative hierarchical clustering requires comparing similarities between pairs clusters, not just pairs of points
- There are different *linkage functions* that generalise a notion of dissimilarity $\mathrm{Dis}(\mathbf{x}, \mathbf{y})$ between two points to apply to any two sets of points A and B:
  - single linkage $L_{\mathrm{single}}(A, B)$
  - complete linkage $L_{\mathrm{complete}}(A, B)$
  - average linkage $L_{\mathrm{average}}(A, B)$
  - centroid linkage $L_{\mathrm{centroid}}(A, B)$

Linkage functions (2)

- Single linkage (minumum) considers the closest pair of points between the two clusters:

$$L_{\text{single}}(A, B) = \min_{\mathbf{x} \in A, \mathbf{y} \in B} \mathsf{Dis}(\mathbf{x}, \mathbf{y}),$$



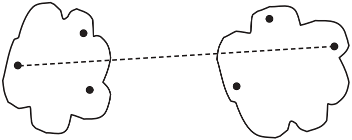
(Note that when working with *similarity* measures we instead take the object pair with *maximum* similarity!)

Linkage functions (3)

- Alternatively, we can try to enforce that clusters should have *all* pairs of points reasonably close to each other

- This gives complete linkage (maximum):

$$L_{\text{complete}}(A, B) = \max_{\mathbf{x} \in A, \mathbf{y} \in B} \mathsf{Dis}(\mathbf{x}, \mathbf{y}),$$

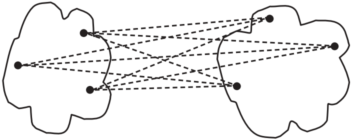


(Again, for *similarity* measures we instead take *minimum* of the objectwise similarities!)

Linkage functions (4)

- An intermediate criterion is averaging

$$L_{\text{average}}(A, B) = \frac{1}{|A|\,|B|} \sum_{\mathbf{x} \in A, \mathbf{y} \in B} \mathrm{Dis}(\mathbf{x}, \mathbf{y})$$



(With *similarity* measures we also just take the average value.)

Linkage functions (5)

- Centroid based linkage is calculated as

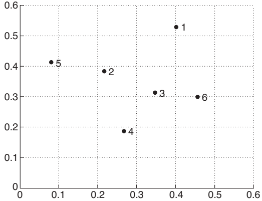
$$L_{\mathrm{centroid}}(A, B) = \mathrm{Dis}(\boldsymbol{\mu}_A, \boldsymbol{\mu}_B)$$

where $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ are the means of the vectors in each cluster:

$$\begin{aligned} \boldsymbol{\mu}_A &= \frac{1}{|A|} \sum_{\mathbf{x} \in A} \mathbf{x} \\ \boldsymbol{\mu}_B &= \frac{1}{|B|} \sum_{\mathbf{y} \in B} \mathbf{y} \end{aligned}$$

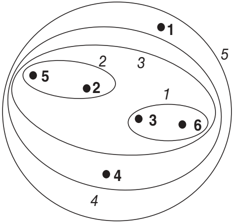
Hierarchical clustering (4)

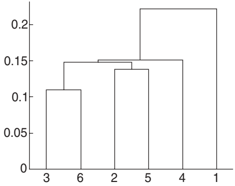
Example 1:



	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

- Single-link:

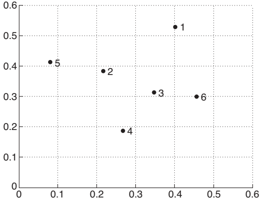




(The heights in the dendrogram correspond to linkage functions $L_{\text{single}}(A, B)$ when clusters A and B are combined.)

Hierarchical clustering (5)

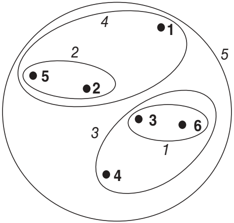
Example 2:

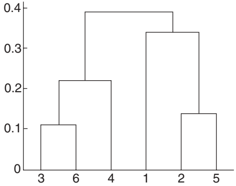


	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

- Complete-link:

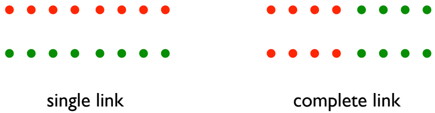
  (The heights in the dendrogram correspond to linkage functions $L_{\mathrm{complete}}(A, B)$ when clusters A and B are combined.)





Hierarchical clustering (6)

- Cluster shapes:
  - *Single-link* can produce arbitrarily shaped clusters (joining quite different objects which have some intermediate links that connect them)
  - *Complete-link* tends to produce fairly compact, globular clusters. Problems with clusters of different sizes.
  - *Group average* is a compromise between the two



- Lack of a global objective function:
  - In contrast to methods such as K-means, the agglomerative hierarchical clustering methods do not have a natural objective function that is being optimized.

Hierarchical clustering (7)

- Computational complexity
  - The main storage requirement is the matrix of pairwise distances, containing a total of $N(N-1)/2$ entries for N datapoints. So the space complexity is: $O(N^2)$.
  - Computing the distance matrix takes $O(N^2)$. Next, there are $O(N)$ iterations, where in each one we need to find the minimum of the pairwise dissimilarities between the clusters. Trivially implemented this would lead to an $O(N^3)$ algorithm, but techniques exist to avoid exhaustive search at each step, yielding complexities in the range $O(N^2)$ to $O(N^2 \log N)$.

    (Compare this to K-means, which only requires $O(NK)$ for K clusters.)

  Hence, hierarchical clustering is *directly* applicable only to relatively small datasets. (But ask Ville again about approximate nearest neighbors!)

Clustering: summary

- K-means and hierarchical clustering are among the main tools in data analysis. Everyone in the area must understand
  - what the algorithms do
  - how to interpret the results
  - computational and other limitations of the algorithms
- Often goal is understanding the data, with no clearly defined prediction or other task
  - difficult to define good performace metrics
  - difficult to give good procedures for “model selection” (e.g. choosing number of clusters)

Next week

- We'll discuss Principal Component Analysis (PCA) next week (you should have read Section 10 of the textbook by this week)
- Also, next week we'll briefly discuss *ensemble methods*
- And then we are done!
- Except of course, there's the exam...