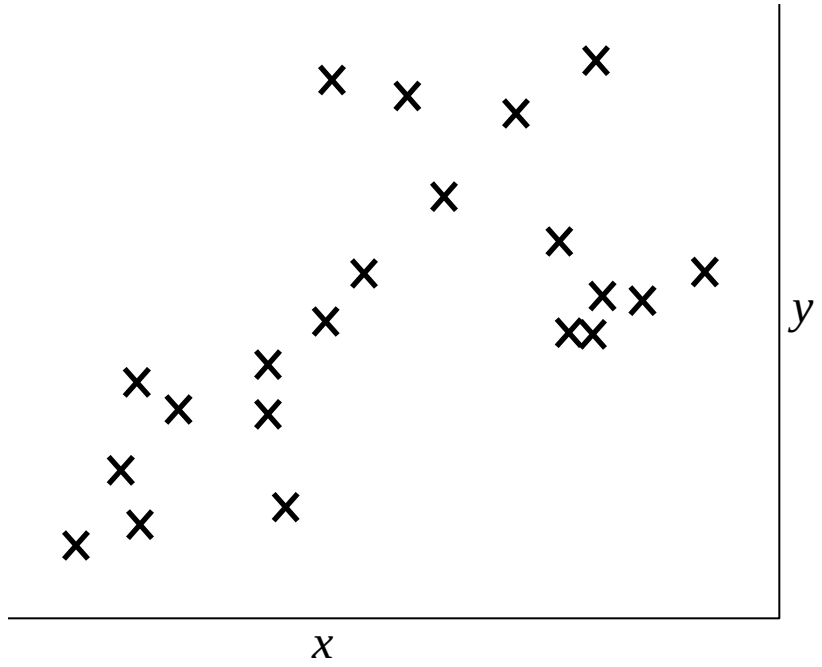
Cluster analysis

Andrew Torda, wintersemester 2009 / 2010, AST…

- classification and prediction
- methods
  - *k*-means
  - hierarchical
    - nearest neighbour
    - divisive
- Übung

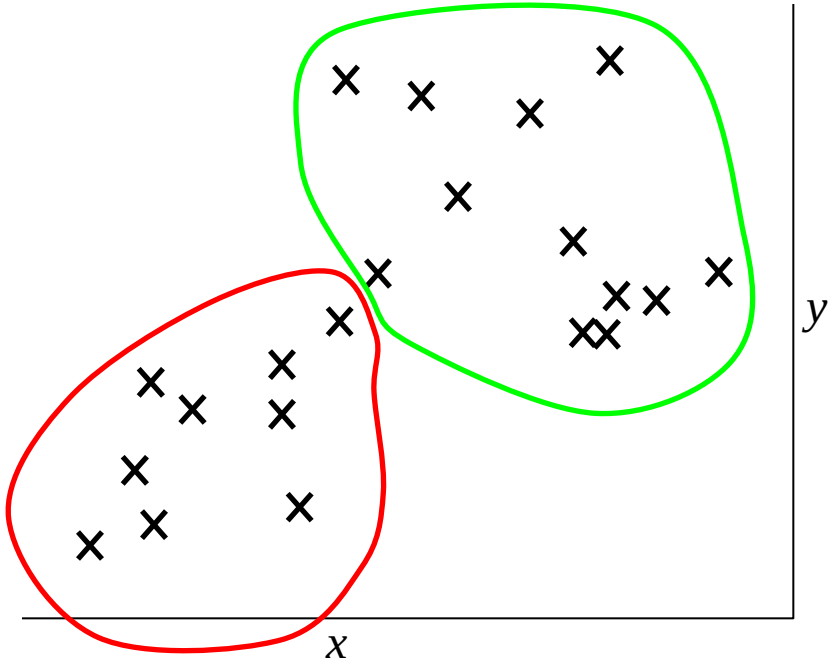
Classification versus prediction ?

- Easy data two clusters

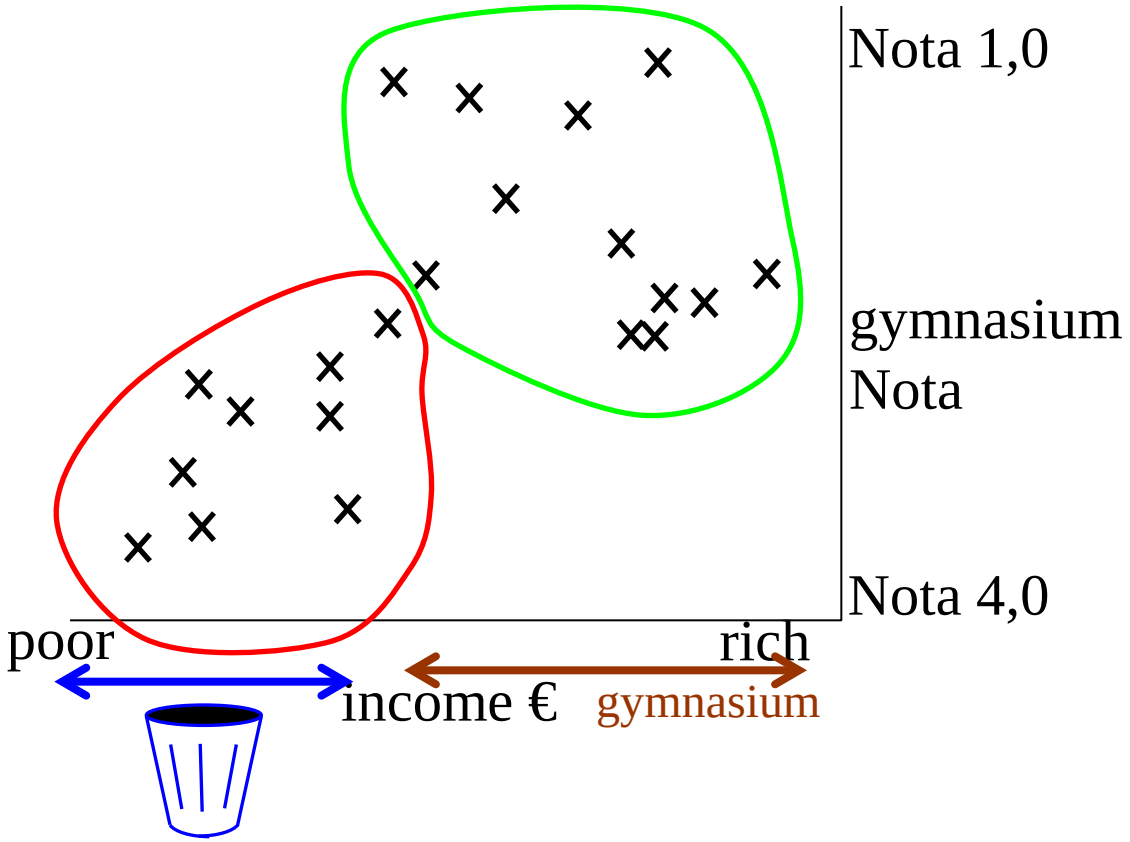


Classification versus prediction ?

- Easy data two clusters
- can this be predictive ?
  - put labels on



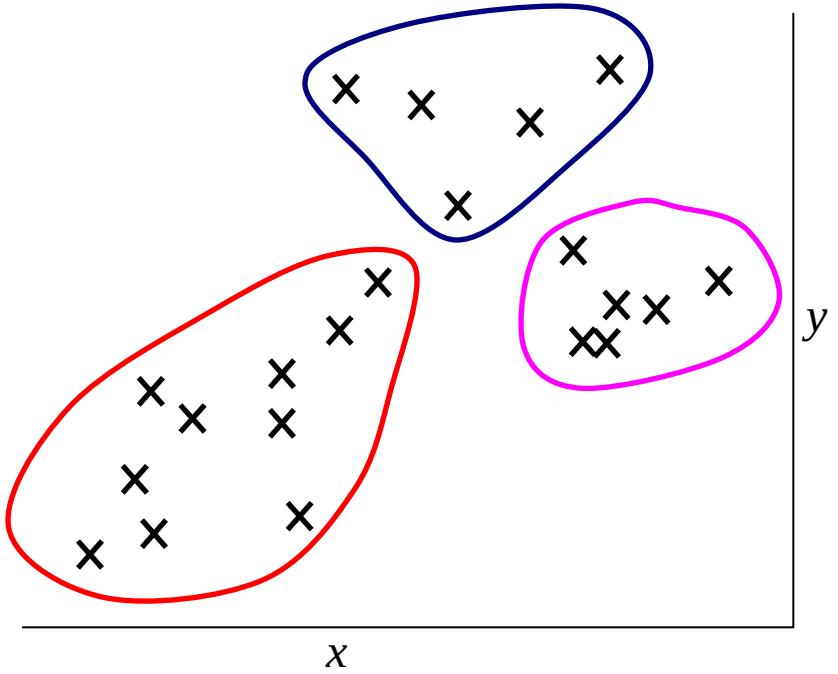
Classification versus prediction ?



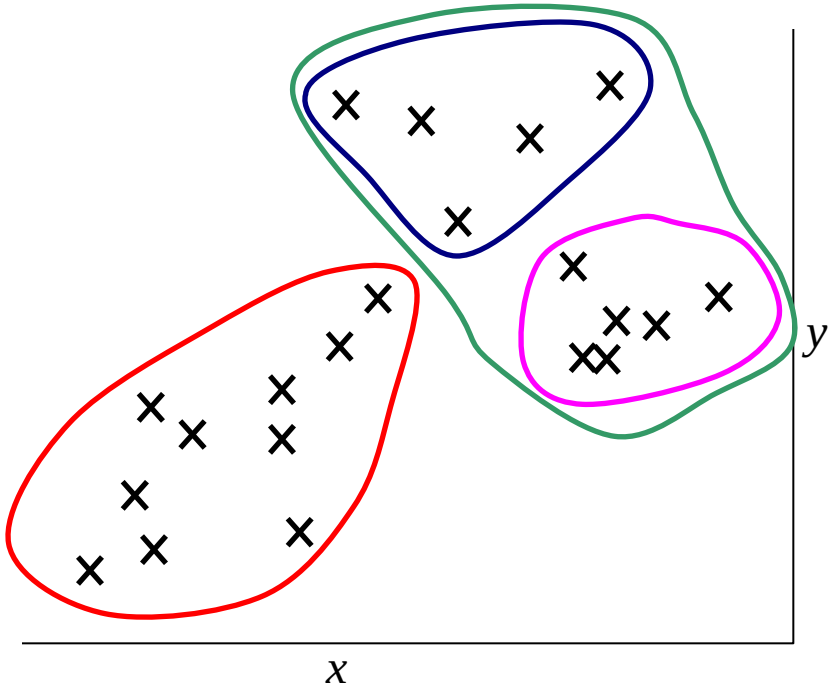
- Try a prediction based on income
- In general
  - if we know of some properties, we can guess others

Problems

- Easy data two clusters
  - is it really ?
- alternative ?

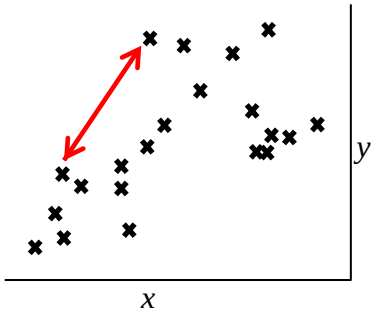


Problems



- two clusters with sub-clusters ?

Distance Measures (Euclidean)



- For any two points
  - want a distance /dissimilarity
- Euclidean distance (easy in two dimensions)

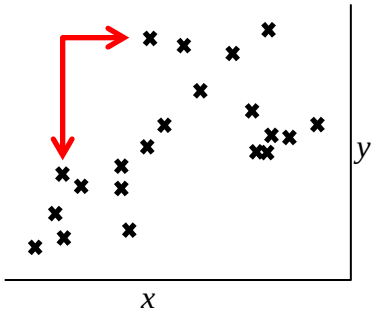
$$d_{ij} = \left(\left(x_i - x_j \right)^2 + \left(y_i - y_j \right)^2 \right)^{1/2}$$

- in 3D $d_{ij} = \left(\left(x_i - x_j \right)^2 + \left(y_i - y_j \right)^2 + \left(z_i - z_j \right)^2 \right)^{1/2}$

- in *n*D (nomenclature does not work)

$$d_{ij} = \left(\left(x_i - x_j \right)^2 + \left(y_i - y_j \right)^2 + \left(z_i - z_j \right)^2 + \ldots \right)^{1/2}$$

Distance Measures (Manhattan)



- 2D $d_{ij} = |x_i - x_j| + |y_i - y_j|$
- *n*D $d_{ij} = |x_i - x_j| + |y_i - y_j| + |z_i - z_j + \ldots|$

- Euclidean versus Manhattan versus …
  - depends on belief
  - if one is lucky, results will not be too different
- Worse cases
  - category data
    - cars have
      - speeds, size, **colour, 2 door/4door**

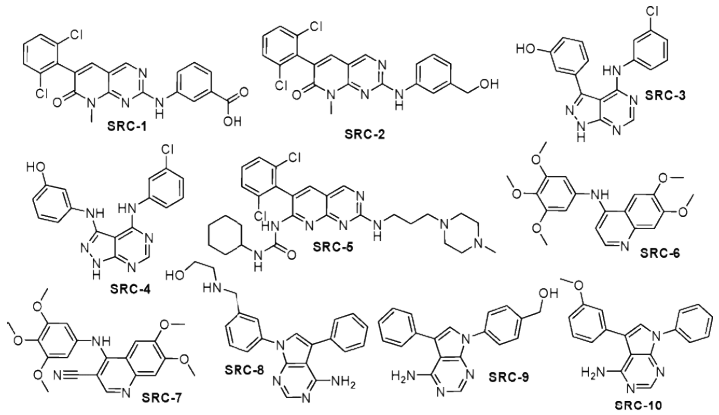
not x, y continuous descriptor

- a possible Manhattan measure

A set of discrete descriptors

- identify properties
  - make long bit-vector
- dissimilarity ?
  - count matching bits
- typically 10^2 - 10^3 properties
- crude ?
  - enough properties that mistakes do not matter

- is this a Manhattan distance ?
  - probably



$n_{polar} < 5$	0
$5 < n_{polar} < 10$	0
$n_{polar} \geq 10$	1
contains sulfur	0
acidic (can lose H^+)	0
has ether O	1
… lots more ...	
mol. wt. < 300	0
300 < mol. wt. < 500	1
mol. wt. ≥ 500	0

ligands from, Wang, N., DeLisle, R. K. and Diller, D.J. (2005), J. Med. Chem., 48, 6980-6990

General versus Specific

- When I know nothing
  - invent a distance / dissimilarity based on descriptors *x*, *y*, ..
- If I know more, use an appropriate distance
  - sequence example
    - Jukes-Cantor distance, *p*-value measure
  - protein structures, metabolic pathways, small molecules
    - (geometric differences, similar reactions, bit strings)

- Given some distances what are the methods ?

Clustering Methods

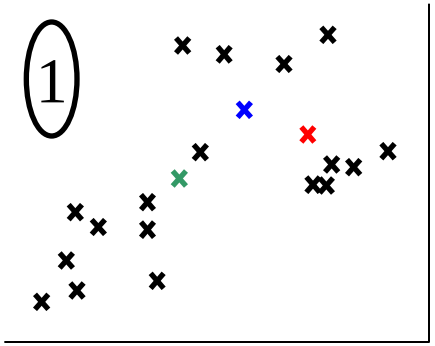
- *k*-means
- hierarchical
- fuzzy (not here)

k-means

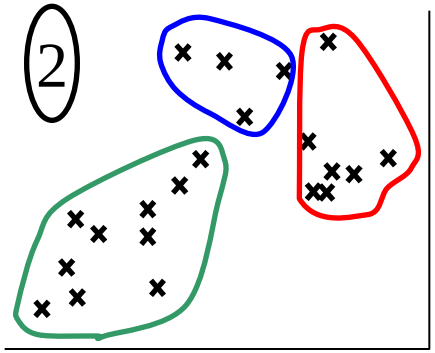
- Pick *k* points - call them cluster centres

```
while (there is substantial change)
   assign each data-point to nearest centre
   re-calculate centres
```

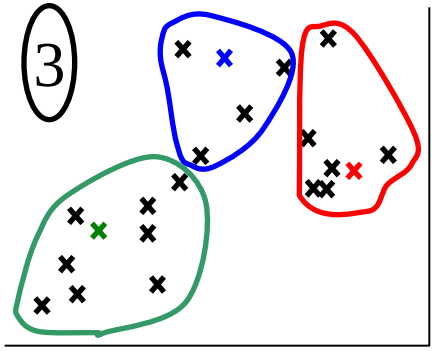
k-means steps



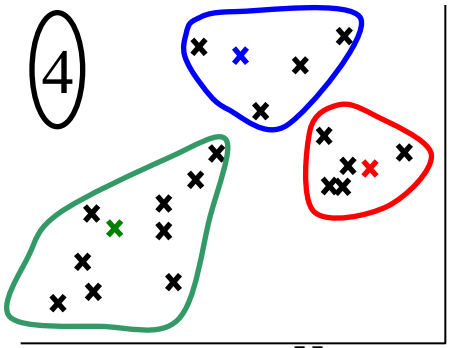
pick 3 points



allocate all other points



pick new centres



allocate all other points

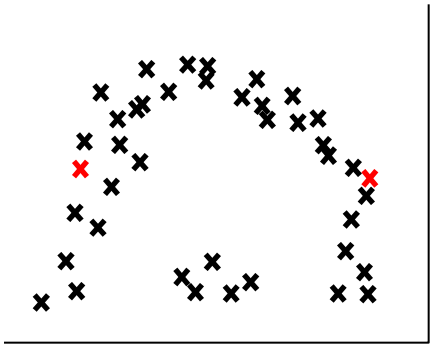
k-means problems

- What is *k* ?
  - guess, experiment, preconception
- Initial choice of cluster centres
- requires concept of cluster centre (mean)

- non deterministic

- convergence

- cluster shape
  - what if red points become centres ?



Hierarchical

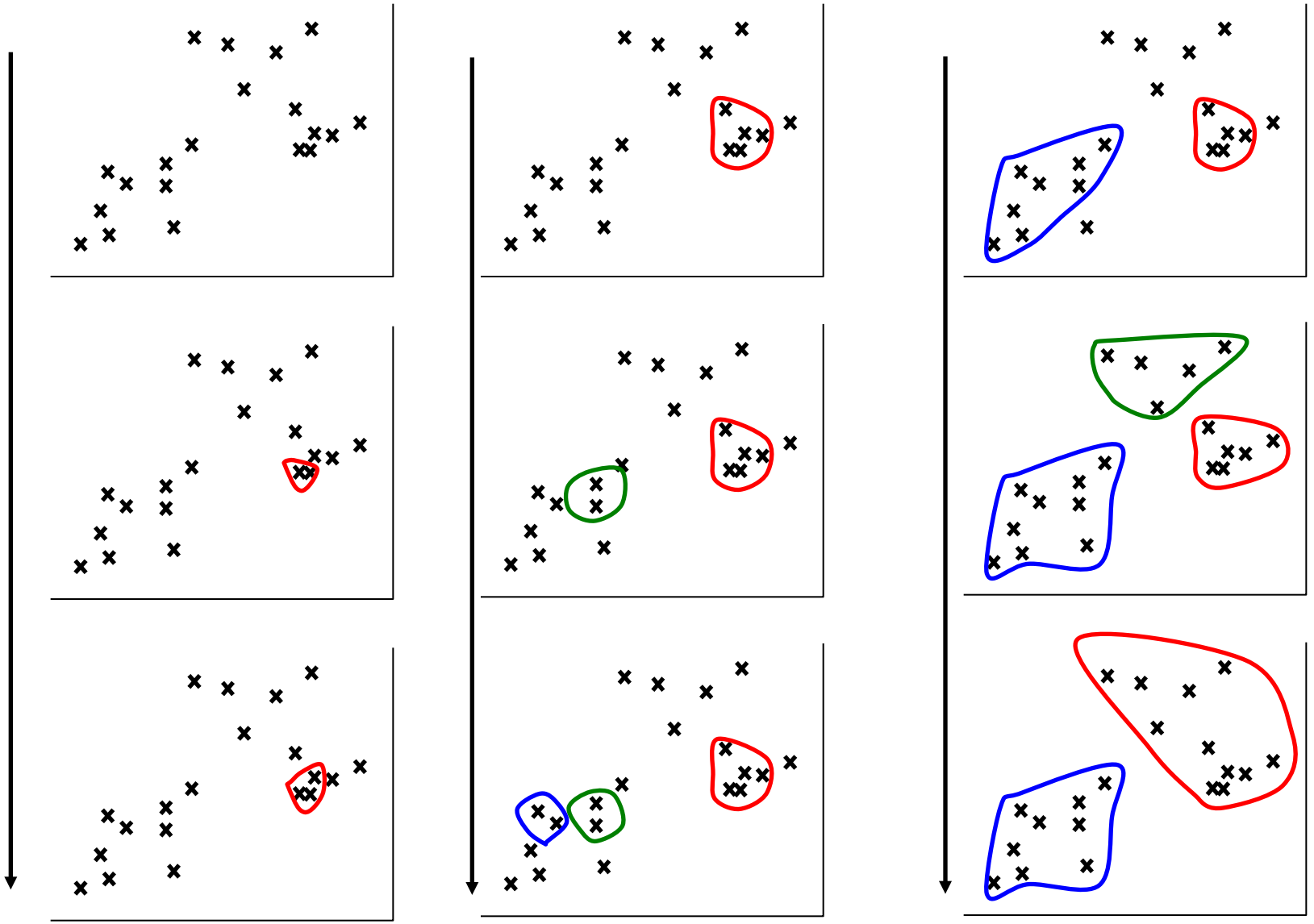
- Two flavours
  - divisive
  - agglomerative / joining / nearest neighbour

agglomerative / joining / nearest neighbour

•For *n* observations form *n* clusters (each point is separate)

```
while (not finished)
      find two nearest clusters (details later)
      join
```

agglomerative / joining example



Divisive

- split_into_two (cluster)

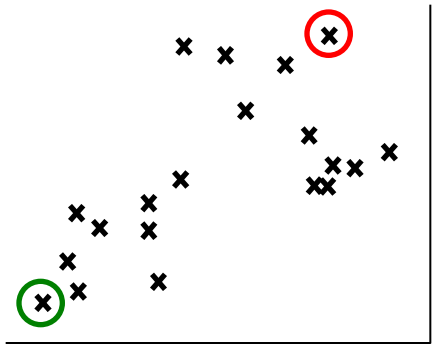
```
split_into_two (cluster)
   select two most separated points as centres of new clusters
   for each point in cluster
      allocate to nearest cluster centre
```

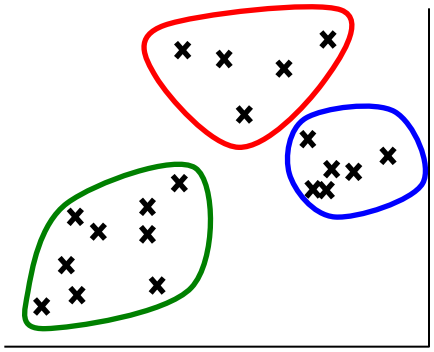
- main procedure

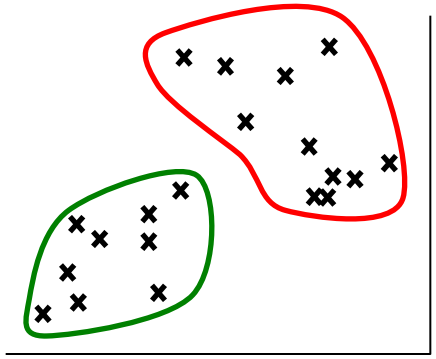
```
all points in one cluster
while (not finished)
    find largest cluster
    split_into_two (cluster)
```

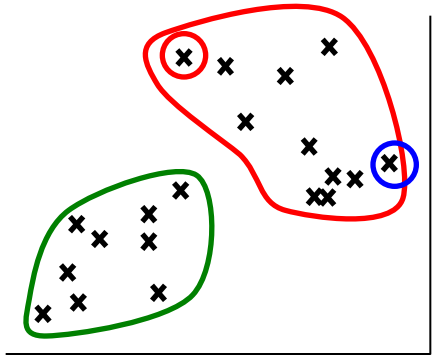
- example

Divisive example

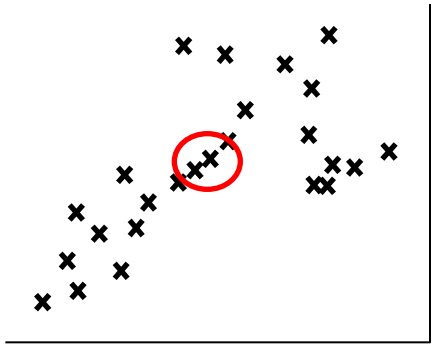


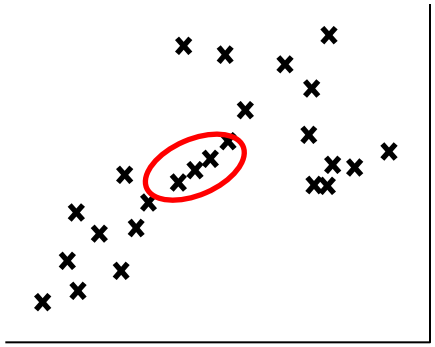


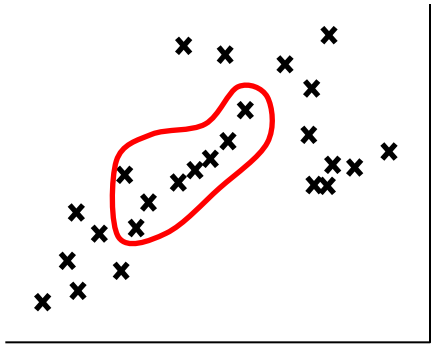




Breaking a joining method



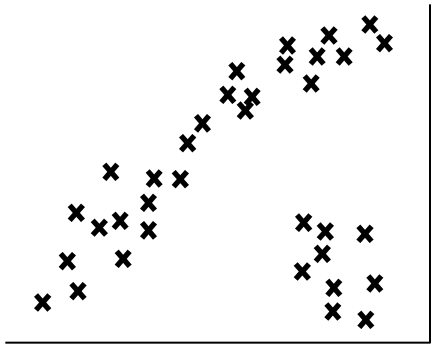


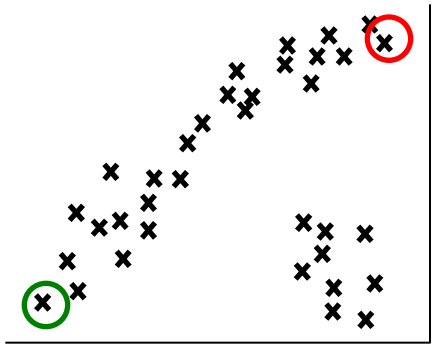


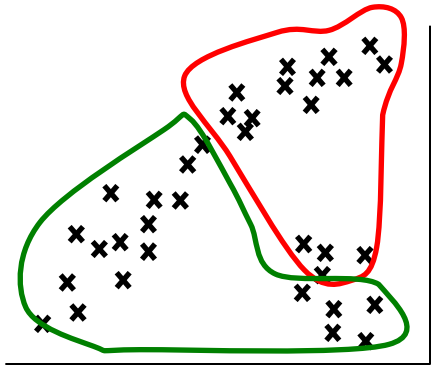
- consider this data with an agglomerative method
- distances are important, not compactness

- is this always true ?

breaking a divisive method





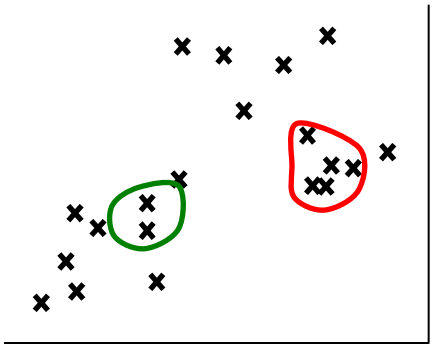


- method considers distances
- in this case
  - compactness of points is more important

- in many problems
  - we only trust measures of high similarity
  - example
    - molecular similarity
      - very different versus very very different

cluster distances

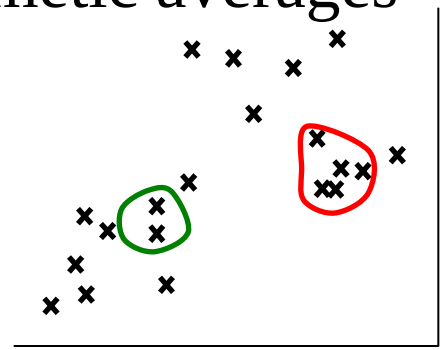
- many details glossed over
  - what is cluster distance ? cluster centre ?
- distance between clusters ?



- distance between points is clear
  - between point and cluster
  - between clusters ?
- sensible choices
  - from cluster to nearest point
  - from cluster to most typical point in other cluster

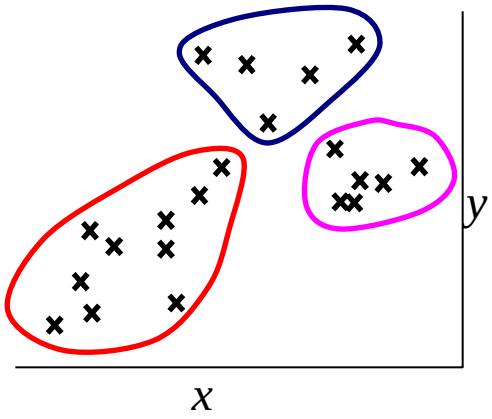
UPGMA

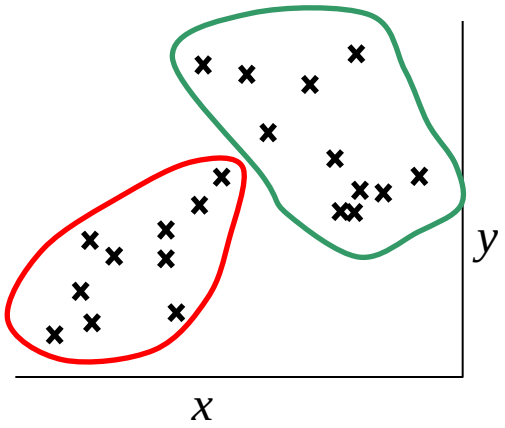
- in many bioinformatics texts
- unweighted pair group method using arithmetic averages
- take red points (5)
- take green points (2)
  - take average of all 2×5 distances
- debate over distance measures
  - similar to agglomerative versus divisive discussion
  - depends on structure of data



How complicated is clustering ?

- in practice
  - distance based methods are best when a table of distances is available O(n^2)
- problem in most fundamental form
  - unknown *k*-clusters
  - combinatorial possibilities huge
- formalise our goal
  - maximise density within clusters
  - maximise distance between clusters
  - should be able to distinguish
    - 2 from 3 cluster answers





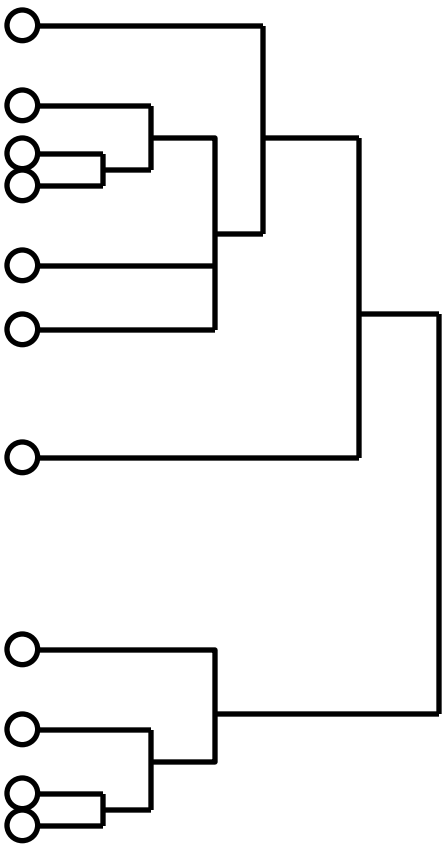
Are we finished ?

- lots of decorations
  - iterations over cluster memberships
  - different definitions of distances, centres
- mixing *x*, *y*, *z* continuous descriptors and categories (red/blue/..)

- fuzzy clustering

Dendrograms

joining →



← divisive

- assumption of hierarchy
- where you call the "classification" depends on where you want to cut tree
- protein shape example
  - most detailed level
    - very similar protein sequences

Applications - sequences

Sequence comparison

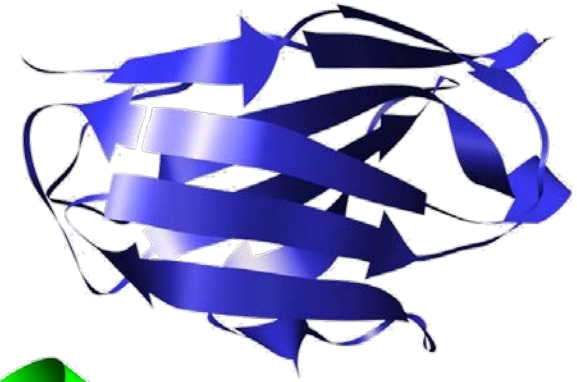
- distances ?
  - evolutionary estimates or
  - similarity based on statistics (*p*-values)
  - clear model (evolution) - suits hierarchy

  - related sequences
    - distances OK
  - less related sequences
    - alignments unreliable

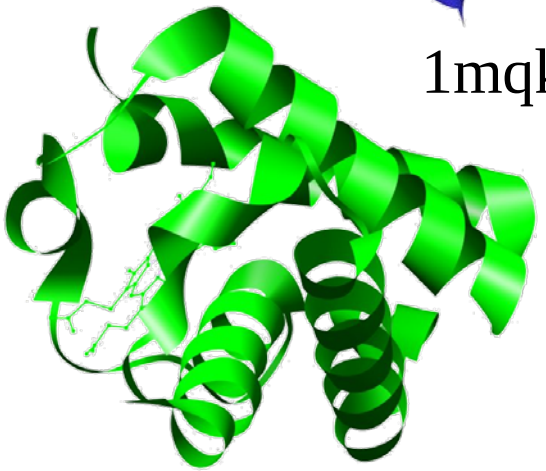
Applications - protein structure

- 3 proteins of similar size
- 1bww and 1mqk easy (immunoglobulins human/mouse)
  - not easy to compare against 1dlw (globin shape)

1bww



1mqk



1dlw

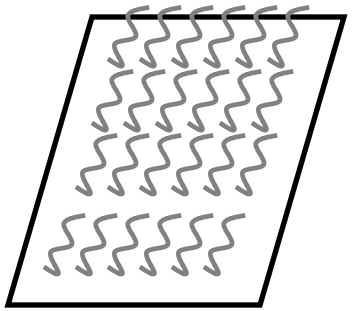
	1bww	1mqk	1dlw
1bww	0	easy	?
1mqk		0	?
1dlw			0

Applications - protein structure

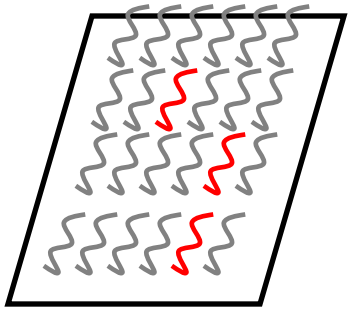
- Are we lost ?
  - easiest to tackle problems with joining methods

Applications - microarray data

- what are microarrays ?
  - little slabs with pieces of DNA bound



microarrays



- lots of bits of DNA from known genes (complementary)
- pour on a sample from cells with mRNA
  - some binds
  - detect by fluorescence
  - have a look which bits of DNA on chip were affected - tells us which genes were involved
  - we know which genes were activated in the original soup

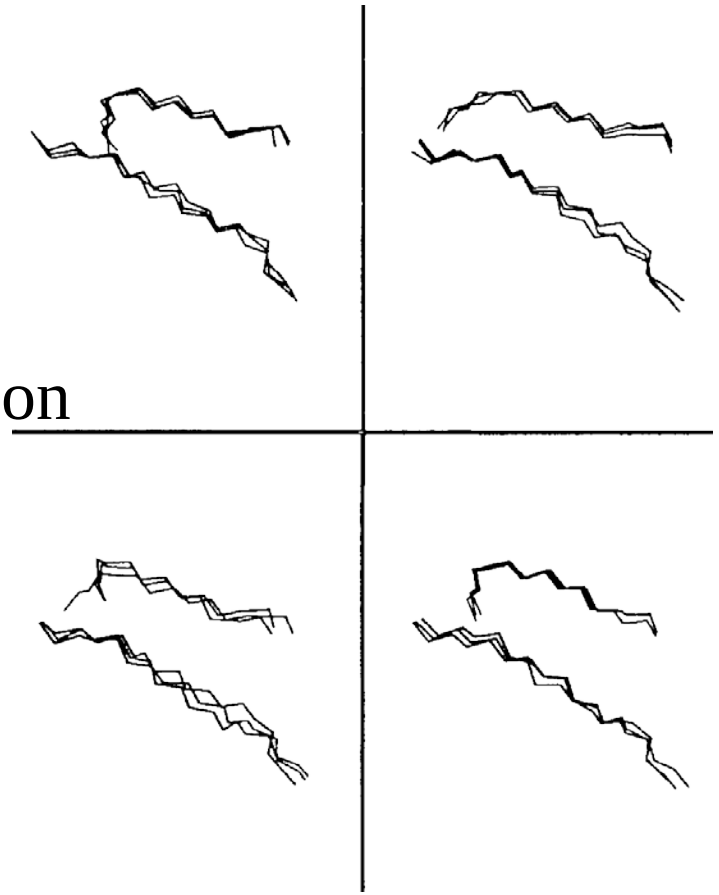
microarrays

- feed sugar to cells
  - pour on to microarray - who lights up ?
  - boring
- feed lipids to cells
  - who lights up
- feed … to cells
- starve cells, heat cells, find cells with disease

- are there groups of genes whose regulation is similar ?
  - should let you find genes in pathways / regulation mechanisms

protein structure

- I simulate a protein molecule and see 10^6 configurations
  - is the molecule constantly changing or sometimes leaving and returning to conformations ?
  - does not look like much..
    backbone atoms only
  - long molecular dynamics simulation
    - 4 major clusters selected
    - each represented by centre
      + two outliers



Summary

- Rarely is there a correct answer
- Method of choice may depend on data

- best case
  - reliable distances known between all points
- real problems
  - noise / outliers

- running time ?
  - O(n^2) for dissimilarity matrix
  - method dependent - usually less than O(n^2)