

Data analysis and stochastic modeling

Lecture 3 – Cluster analysis

Guillaume Gravier

`guillaume.gravier@irisa.fr`

with a lot of help from Dr. HOI Chu's course

<https://svn.mosuma.net/r4000/doc/course/ci6227/public/lectures/lecture07cluster.pdf>



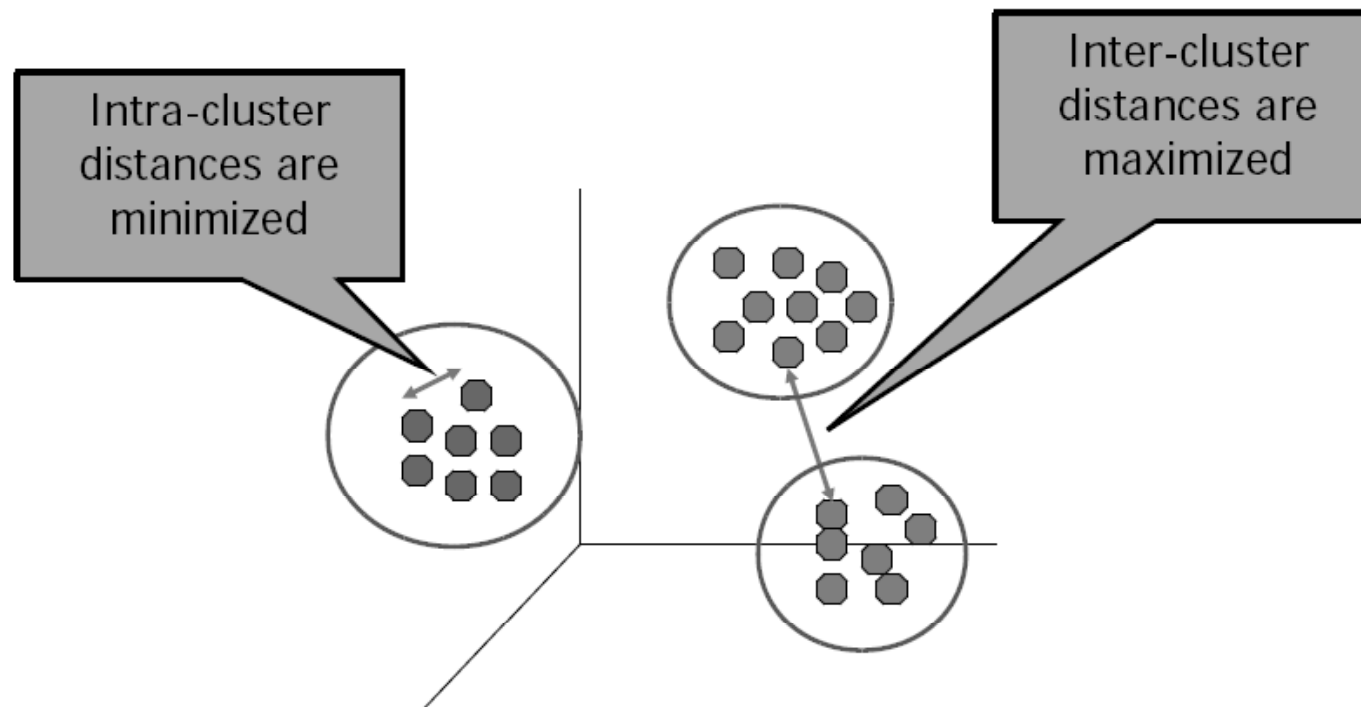
What are we gonna talk about today?

- What's cluster analysis?
- Partitioning clustering
 - k-means and the likes
- Hierarchical clustering
 - bottom-up clustering, linkage methods
- A quick survey of other methods
 - density methods, spectral clustering, etc.
- Case study

In short: an overview of the art of grouping data according to their similarity.

What is clustering?

Clustering consists in grouping data together in “classes” where objects in a class are similar and objects from different classes are different.



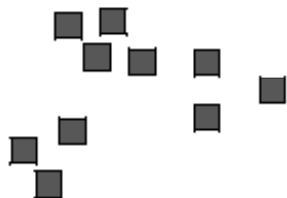
An ambiguous notion



How many clusters?



Six Clusters



Two Clusters



Four Clusters



What's clustering good for?

- **(exploratory) data analysis and understanding** (data mining)
 - ▷ Biology: taxonomy of living things
 - ▷ Information retrieval: grouping similar documents
 - ▷ Land use: identifying areas with similar properties
 - ▷ Marketing: discover distinct groups of customers
 - ▷ City planning, earth quake analysis, etc.

- **pre-processing tool**
 - ▷ quantization, coding and compression
 - ▷ classification, segmentation, etc.
 - ▷ k nearest neighbor search



Quality

Good clustering

= low within-class similarity

= high across-class similarity (eventually)

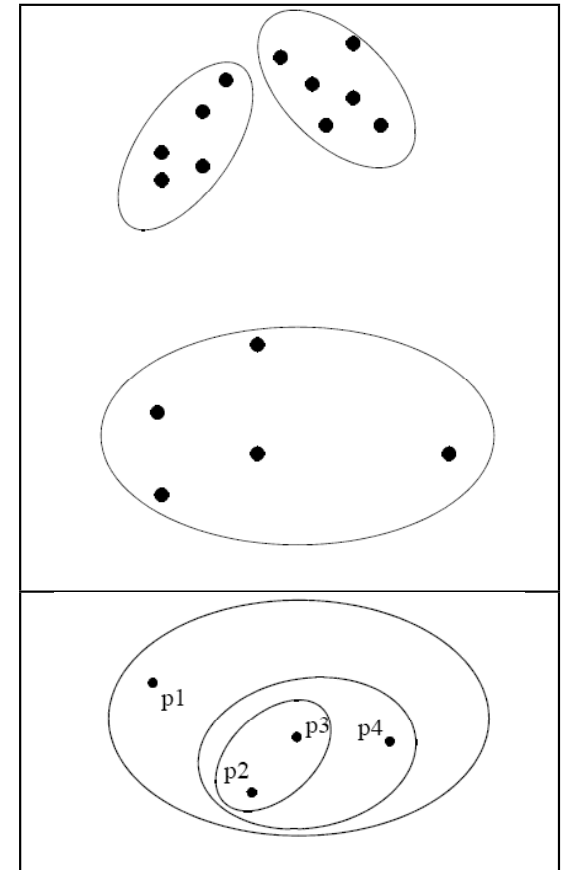
but other factors are to be considered:

- scalability
- dynamic behavior
- ability to deal with noise and outliers
- (in)sensitive to the order of input
- etc.

Two main philosophies

Clustering \Rightarrow a set of clusters

- partitioning data: divide the entire data set
 - ▷ non-overlapping clusters
 - ▷ each object belongs to one and only one cluster
- aggregating data: group similar data
 - ▷ nested clusters
 - ▷ hierarchical structure



But other philosophies do exist: density-based, grid-based, model-based, constraints-based, etc.

About pairwise distances

similar objects \Rightarrow notion of distance

- Any measure of the similarity $d(x_i, x_j)$ between two objects can be used to solve the problem
- The adequate measure highly depends on the nature of the data and on the nature of the problem
 - ▷ distance measures
 - ◇ Euclidian, Manhattan, Mahalanobis, χ^2 , etc.
 - ▷ similarity (no triangular inequality)
 - ◇ cosine, template matching, edit distance, generalized likelihood, etc.
 - ▷ conceptual measures
 - ◇ whatever one can think of...

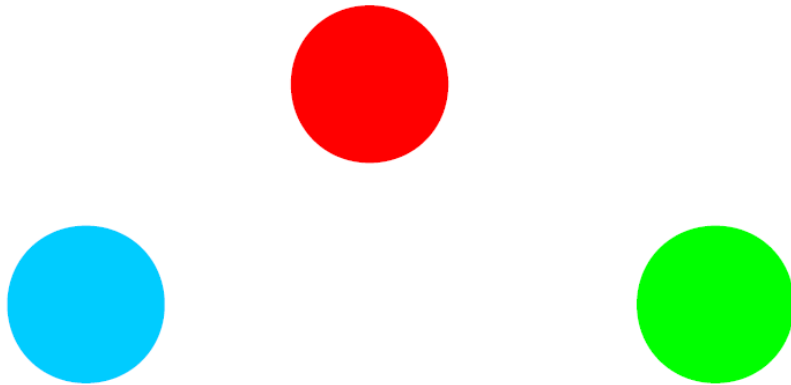
Some typology elements

- **exclusive or not?**
 - ▷ points might belong to several clusters (with or without weights)
- **fuzzy or not?**
 - ▷ in fuzzy algorithms, a point belong to all clusters with a weight $\in [0, 1]$
- **partial or not?**
 - ▷ only part of the data is clustered
- **homogeneous or not?**
 - ▷ clusters of very different shape

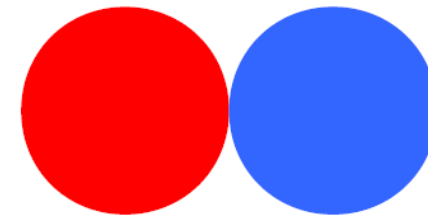
Some typology elements (cont'd)

- **well separated clusters**
 - ▷ every point in the cluster is closer to every point in the cluster than to any point outside the cluster
- **center clusters**
 - ▷ every point in the cluster is closer to the center of the cluster than to the center of any other cluster
- **contiguous cluster**
 - ▷ every point in the cluster is closer to at least one point in the cluster than to any point in another cluster
- **density-based cluster**
 - ▷ dense region of points in a cluster separated from other clusters by low-density regions

Some typology elements (cont'd)



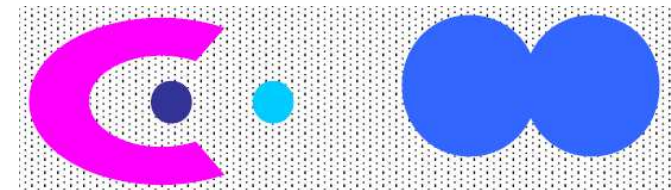
well separated



center



contiguous



dense

Some typology elements (cont'd)

To select the most appropriate solution one must look at the following elements

- Type of proximity or density measure
- Sparseness
 - ▷ Dictates type of similarity; Adds to efficiency
- Attribute type
 - ▷ Dictates type of similarity
- Type of Data
- Dimensionality
- Noise and Outliers
- Type of Distribution

The k-means algorithm

Idea: Divide some data x_i into K clusters represented by the mean value of their members c_k (centroids), so as to minimize the overall quantization error

$$e = \sum_i d(x_i, c_{f(i)})$$

Algorithm:

initialize K centroids c_k

while not converged **do**

for $i = 1 \rightarrow N$ **do**

 assign x_i to the closest centroid ($f(i) \leftarrow \arg \min_k d(x_i, c_k)$)

end for

for $i = 1 \rightarrow K$ **do**

 update centroid c_k from all assigned points

end for

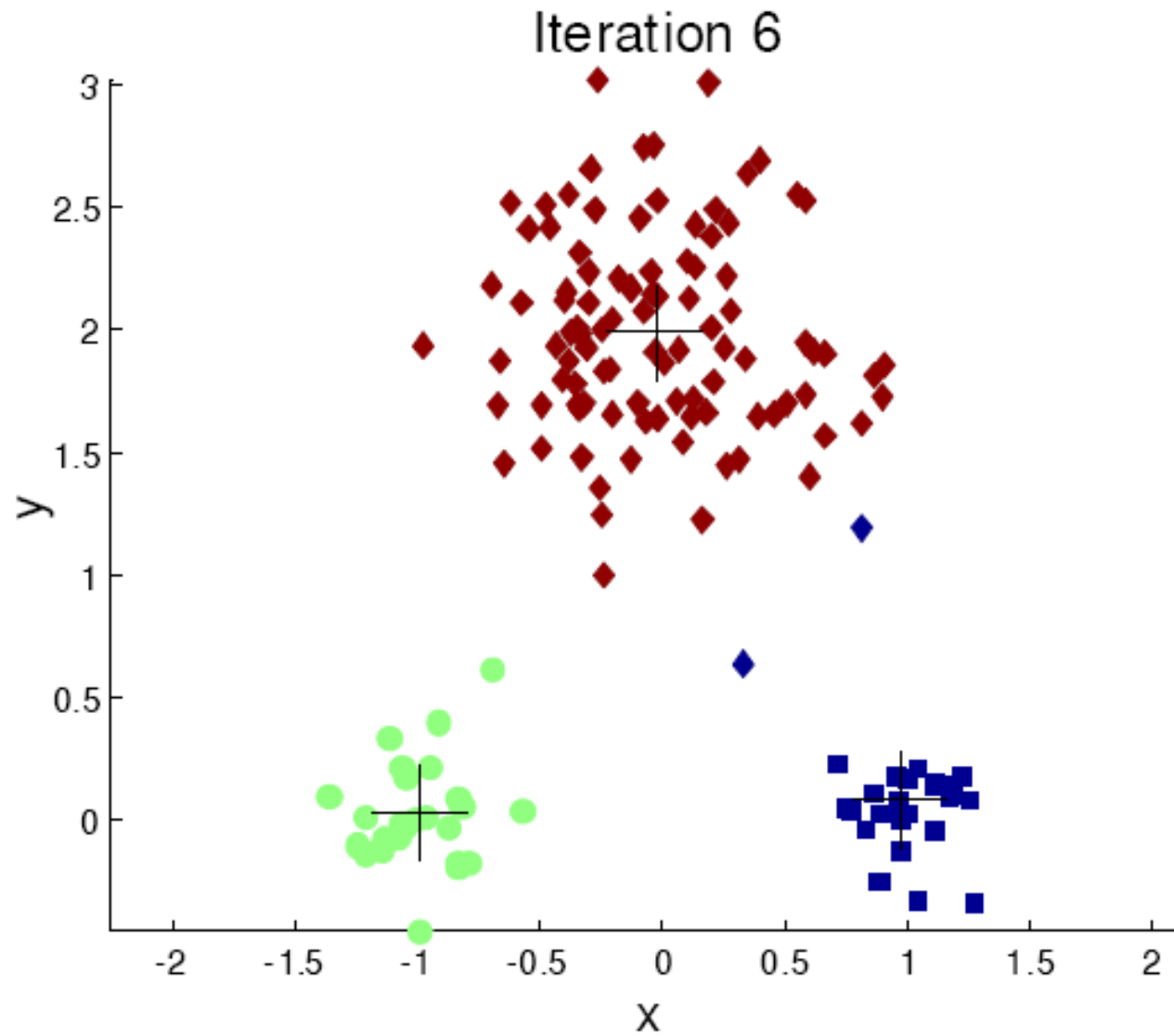
end while

[J. B. McQueen. Some methods for classification and analysis of multivariate observations. Proc. Symposium on Math., Statistics, and Probability, pp. 281-297, 1967]

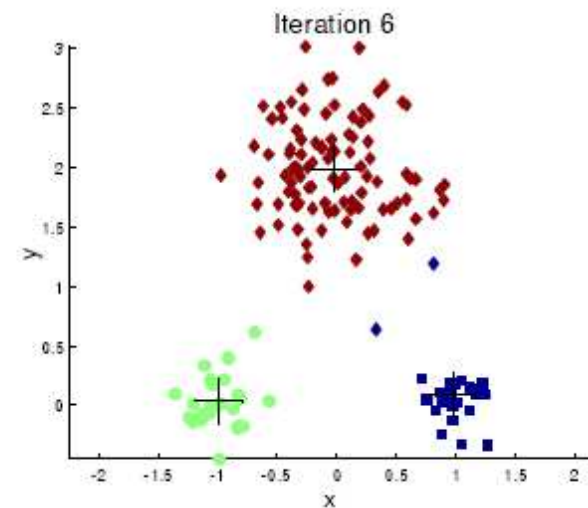
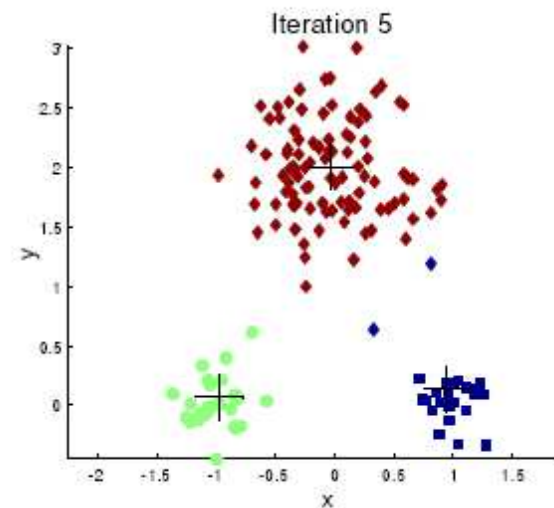
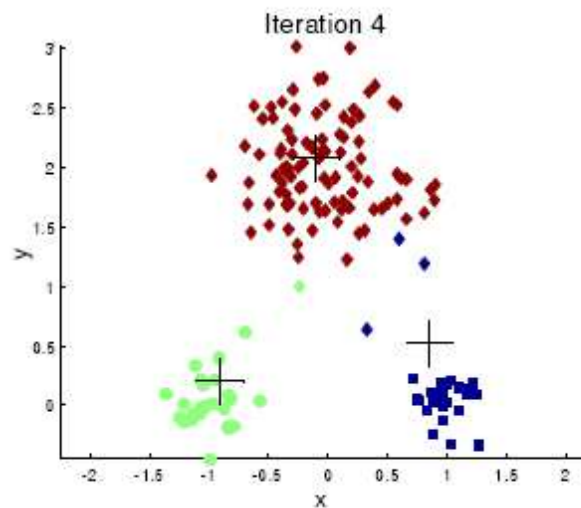
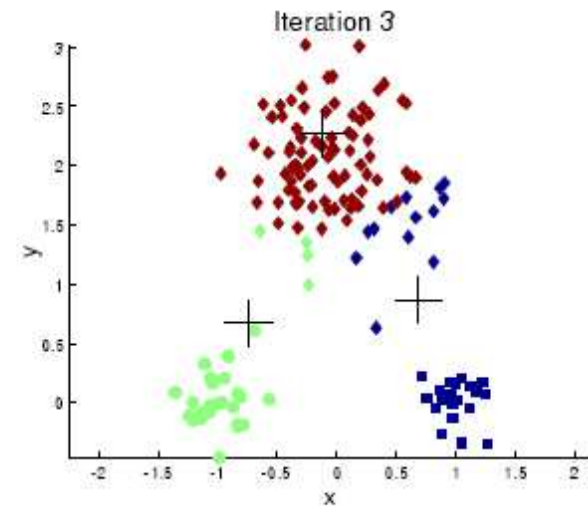
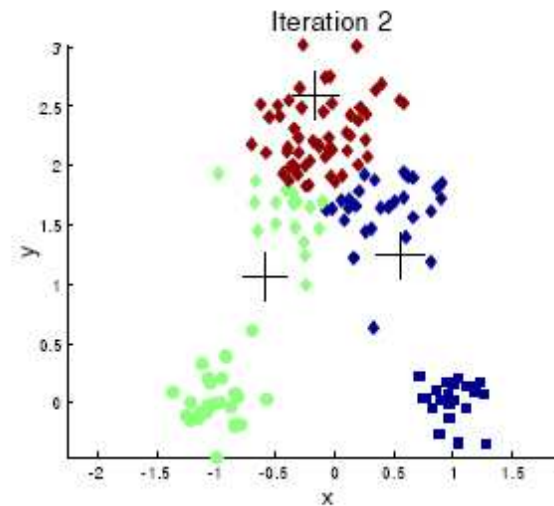
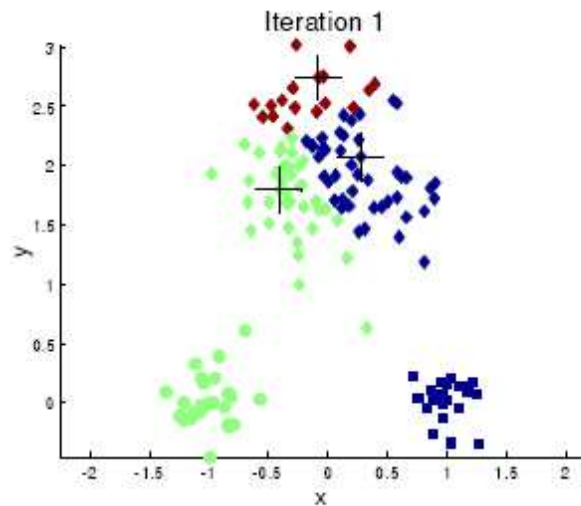
The art of k-means clustering

- the distance is the key
 - ▷ mean has to have a meaning ...
 - ▷ ... but we can use the median instead (the *k-medians* or *k-medioids* algorithm)
- convergence = nothing moves anymore!
 - ▷ convergence is guaranteed
 - ▷ often in 10 to 20 iterations
- complexity = $O(iKNd)$
 - ▷ i = #iterations, d =dimension(x_i)
- initialization is tricky!
 - ▷ random choice
 - ▷ multiple runs
 - ▷ hierarchical k-means (LBG)

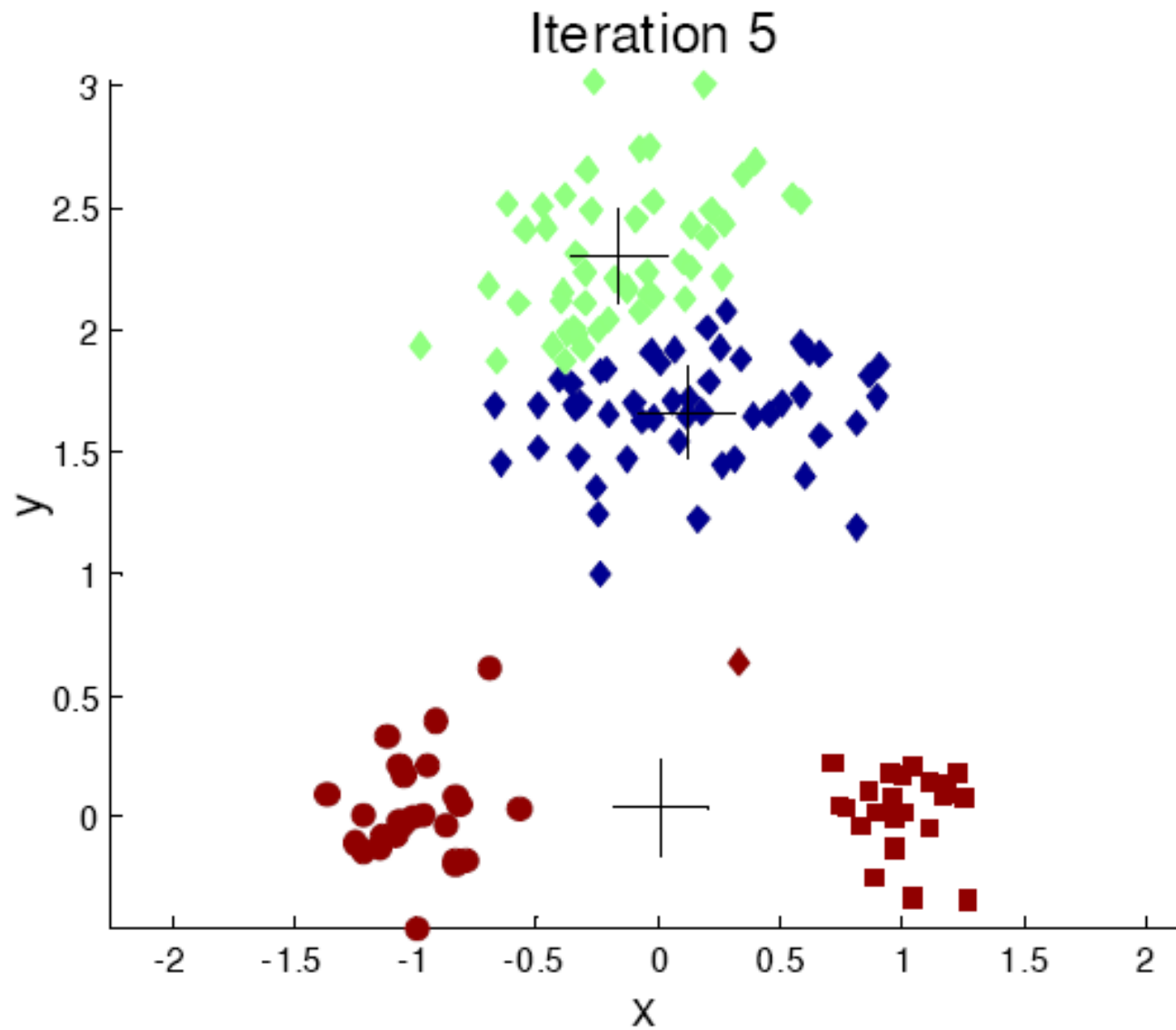
Initialization issues



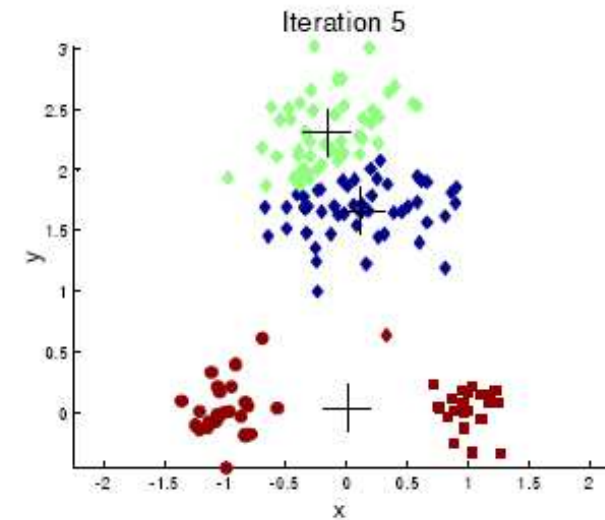
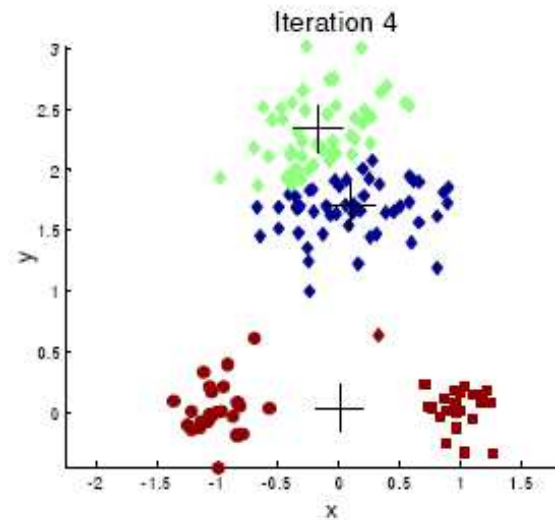
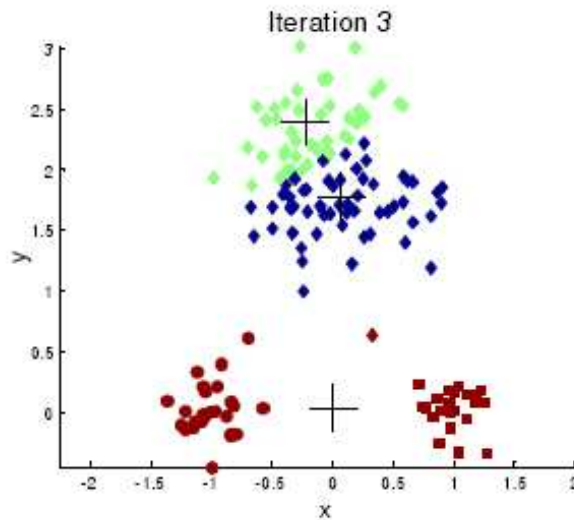
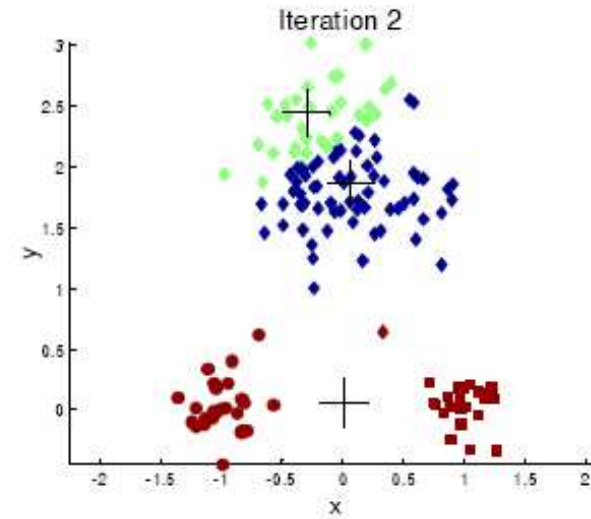
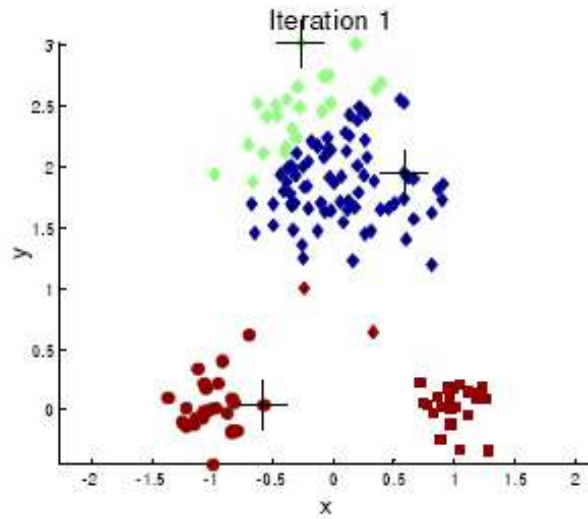
Initialization issues (cont'd)



Initialization issues (cont'd)



Initialization issues (cont'd)



Initialization issues (cont'd)

- **multiple runs**

- ▷ but that's costly!
- ▷ have to deal with “*dead*” clusters (e.g., replace them)

- **hierarchical k-means**

- ▷ the original Linde-Buzo-Gray algorithm (aka bisecting k-means)
-

initialize centroids $c_1(1)$ to gravity center

$i \leftarrow 1$

while not enough clusters ($i < p$) **do**

 split each centroids $c_i(j)$ (along maximum variance line)

 run k-means

end while

- ▷ and its many variants

- ◇ split only the biggest cluster → arbitrary number of clusters instead of 2^p
- ◇ points stay within their parent cluster → much faster

In any case, local optima!

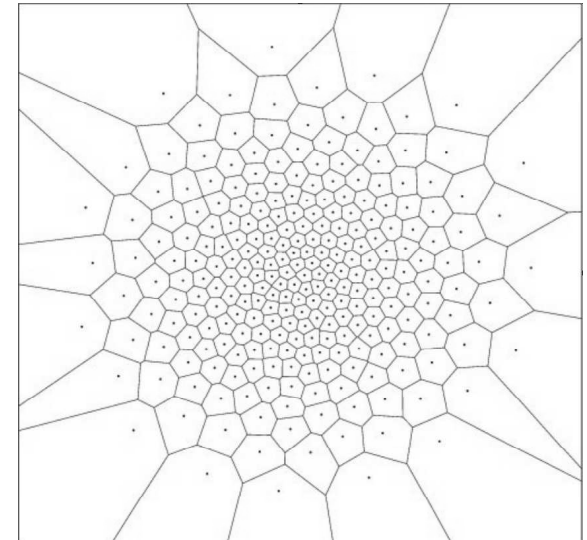
Pros and Cons

○ Pros

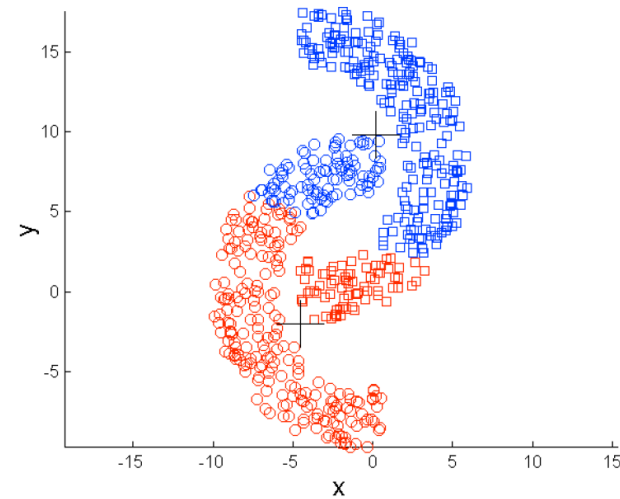
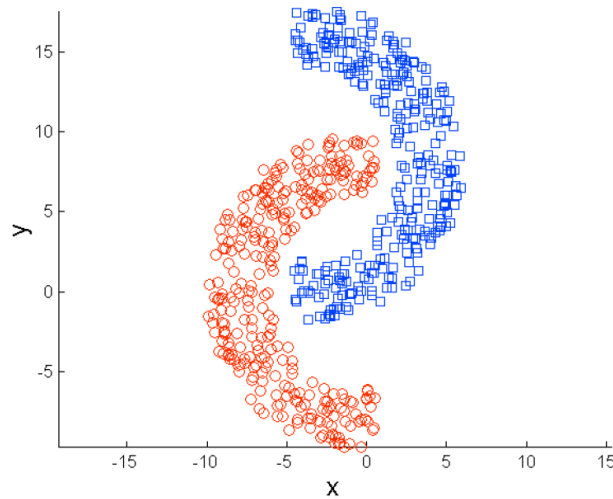
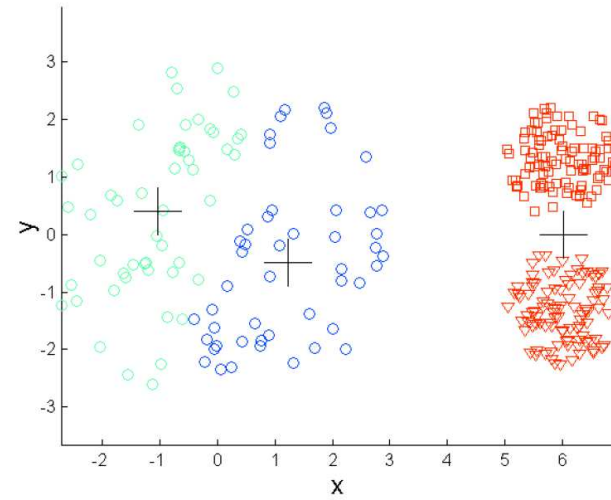
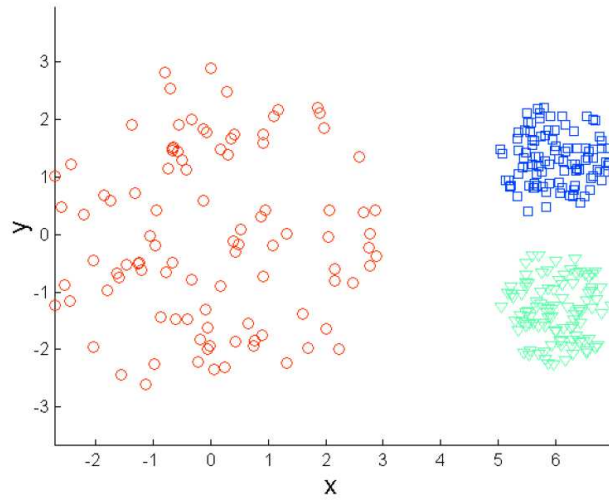
- ▶ simple, clear and popular
- ▶ decently efficient
- ▶ guaranteed convergence
- ▶ can accomodate any shape (with enough clusters)

○ Cons

- ▶ initialization and local optima
- ▶ need to define the number of clusters
- ▶ convex clusters of roughly the same size and density
- ▶ tends to create unbalanced cells
- ▶ highly sensitive to noise and outliers



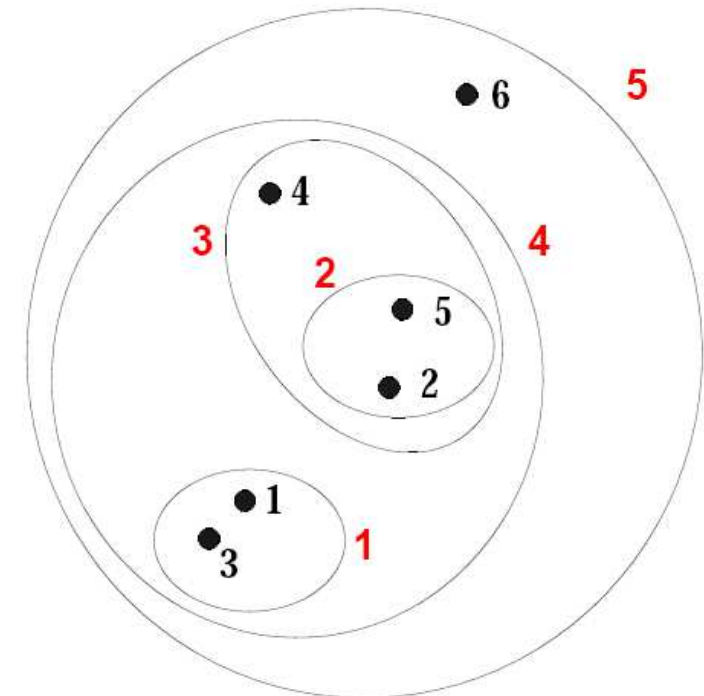
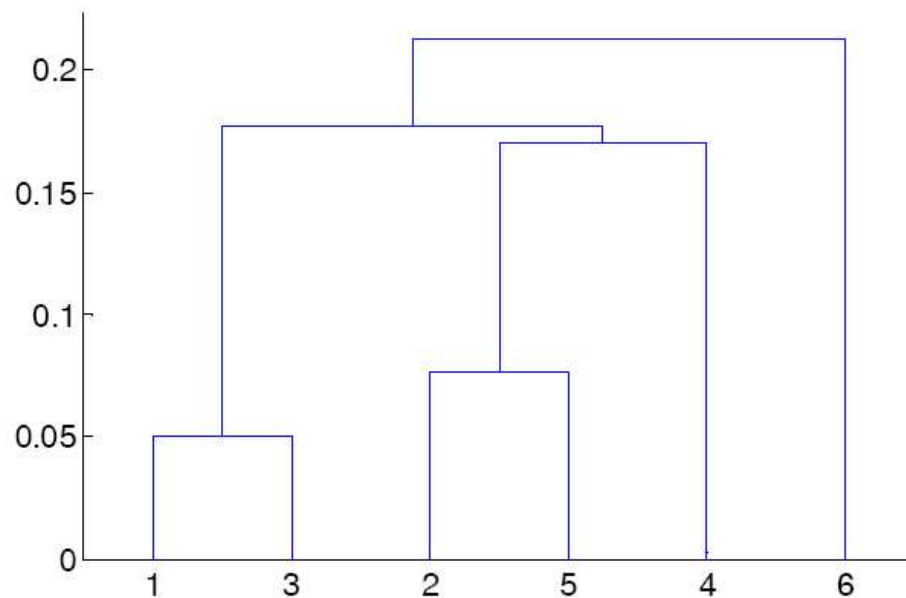
Pros and Cons illustrated



Hierarchical clustering

Idea: Progressively generate clusters by merging or dividing data

- generate nested clusters
- can be visualized as a dendrogram



Agglomerative vs. divisive

- **Agglomerative bottom-up clustering**

- ▶ Bottom-up construction of the dendrogram by progressively merging clusters

initialize N singleton clusters

nclusters $\leftarrow N$

while nclusters > 0 **do**

merge the two closest clusters

nclusters \leftarrow nclusters $- 1$

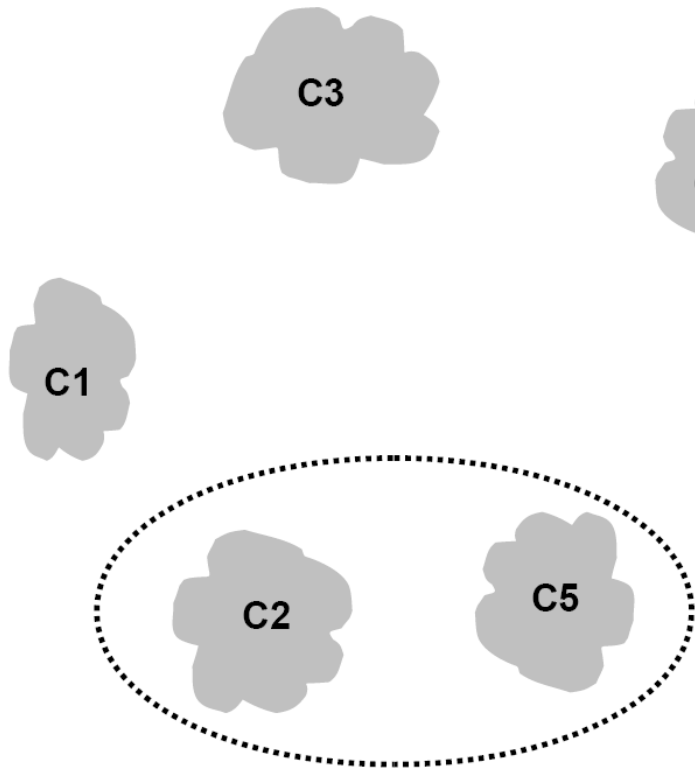
end while

- **Divisive bottom-up clustering**

- ▶ top-down construction of the dendrogram
- ▶ DIANA (Divisive ANALysis)

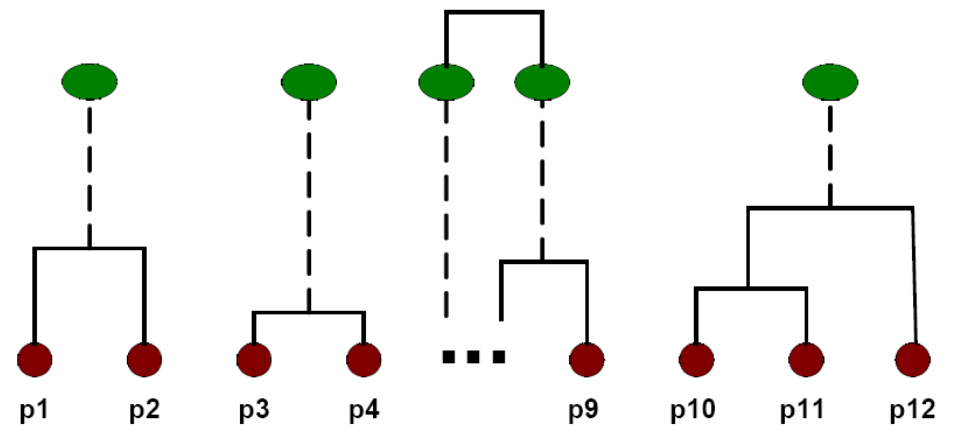
Proximity matrix and bottom-up clustering

update the proximity matrix.

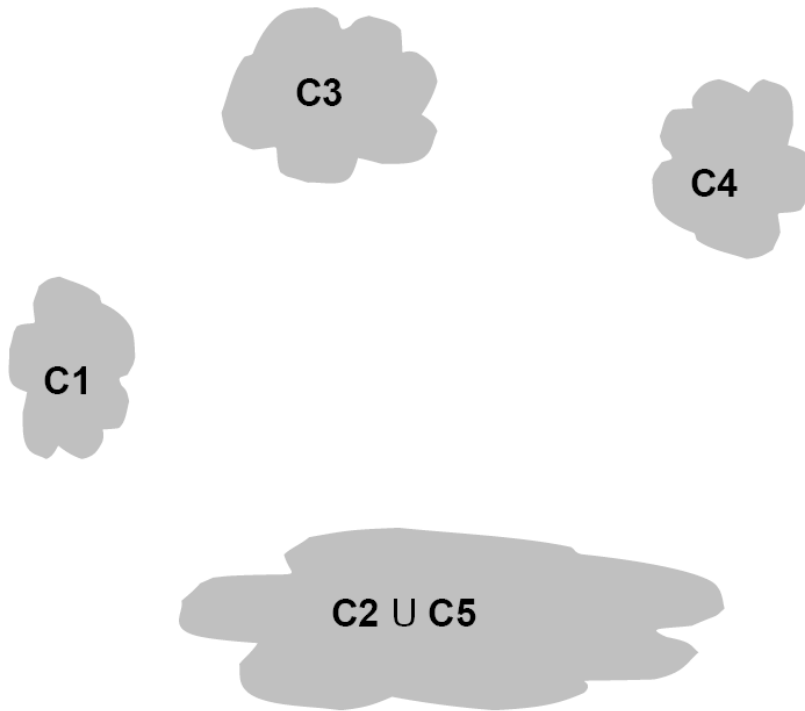


	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix

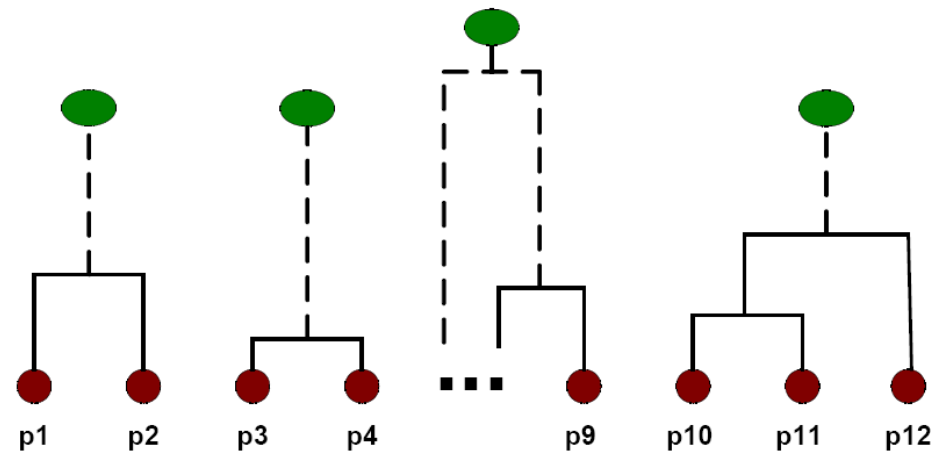


Proximity matrix and bottom-up clustering (cont'd)



	C1	$C2 \cup C5$	C3	C4
C1		?		
$C2 \cup C5$?	?	?	?
C3		?		
C4		?		

Proximity Matrix

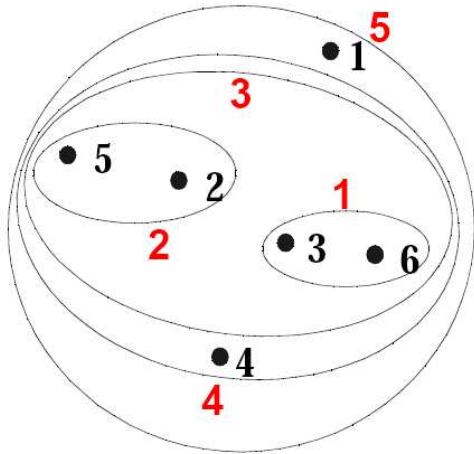


“Linkage” types

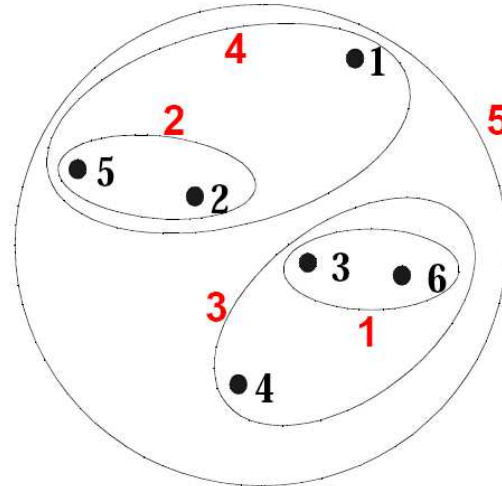
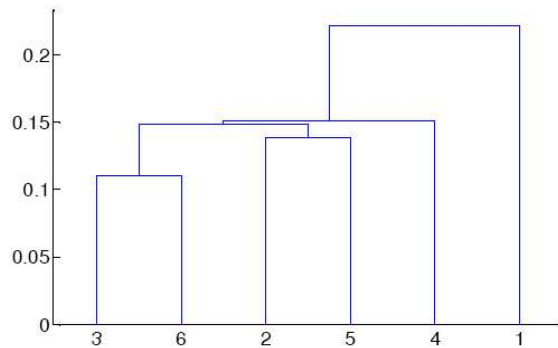
linkage = how to measure the distance between clusters

- single linkage: $D(A, B) = \min(d(x, y) \quad \forall (x, y) \in A \times B)$
→ for well-separated classes only
- total linkage: $D(A, B) = \max(d(x, y) \quad \forall (x, y) \in A \times B)$
→ favors large clusters
- average linkage = average distance between elements in A and B
→ robust to noise and outliers but biased towards globular clusters
- Ward’s linkage = increase in variance for the cluster being merged
- and many others, including distance between mean/median or between (statistical) models of the data

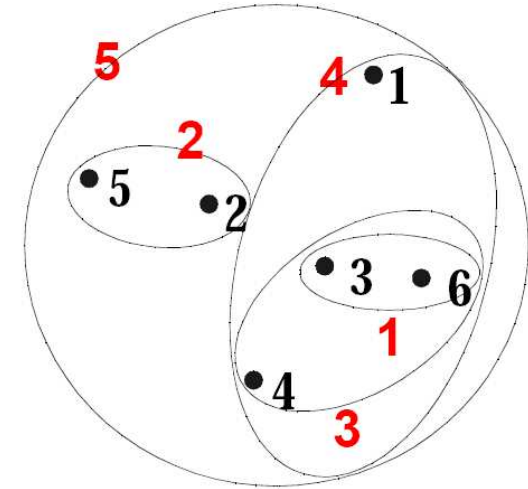
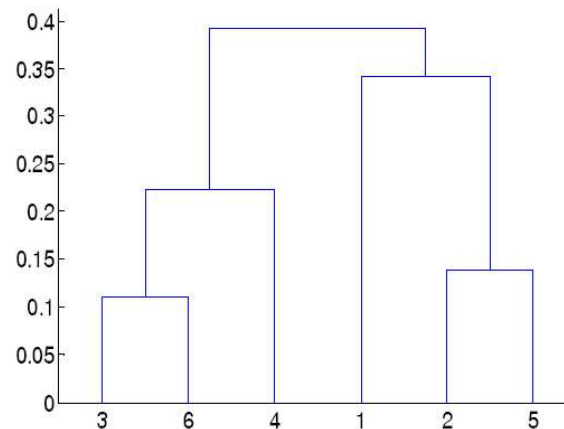
“Linkage” types (cont'd)



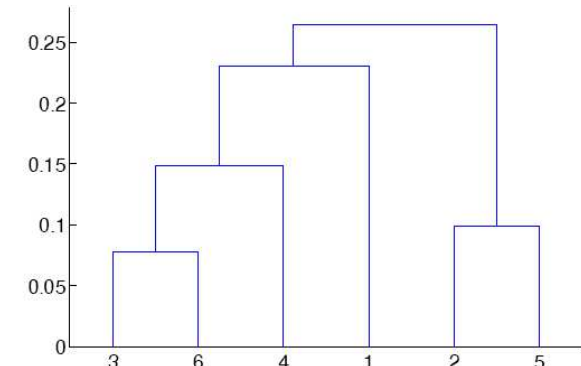
Sing Link (MIN)



Complete Link (MAX)



Group Average



Pros and Cons

- **Pros**

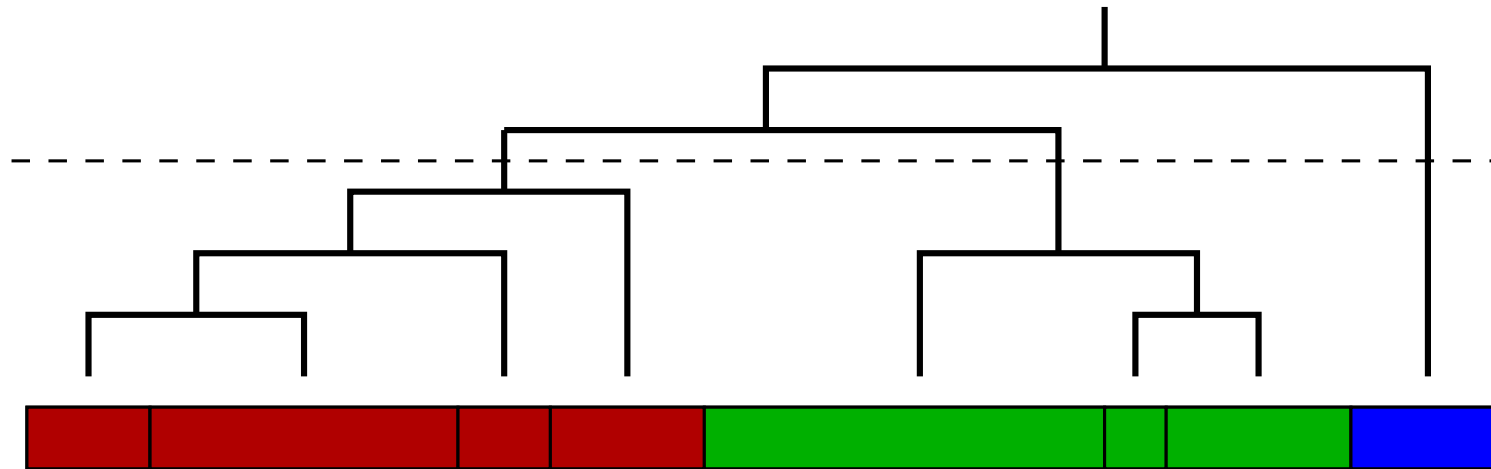
- ▷ better than k-means with non-metric distances
- ▷ can combine metrics and cluster balancing criteria
- ▷ possibility to define a posteriori the number of clusters (though not so easy in practice)

- **Cons**

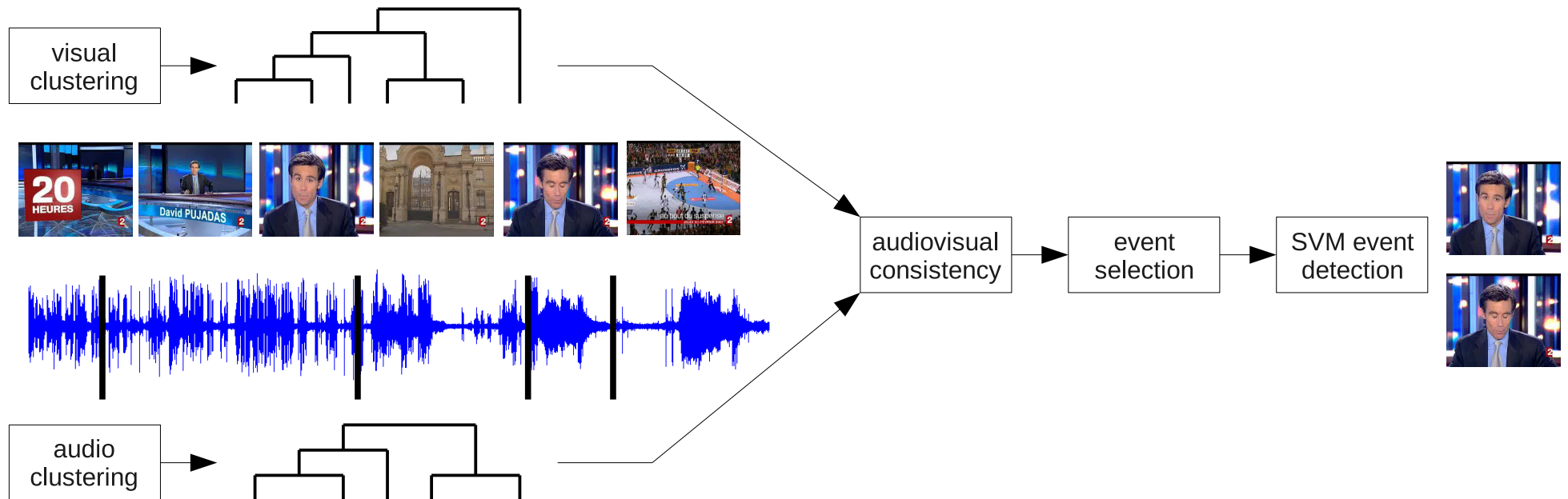
- ▷ quite slow and computationally demanding ($O(N^3)$ or $O(N^2 \log(N))$)
- ▷ local optima may not be globally good
 - ◇ cannot undo what was done previously
 - ◇ require relocation methods
- cutting the dendrogram is not as easy as it seems to be

Bottom-up clustering for temporal partitioning

- detect boundaries of segments → see hypothesis testing
- group together segments with similar characteristics
 - ▷ model based representation of clusters (Gaussian densities and mixtures)
 - ▷ Kullback-Leibler divergence, generalized likelihood ratio
- find out where to cut the dendrogram
 - ▷ model selection approaches: Bayesian information criterion

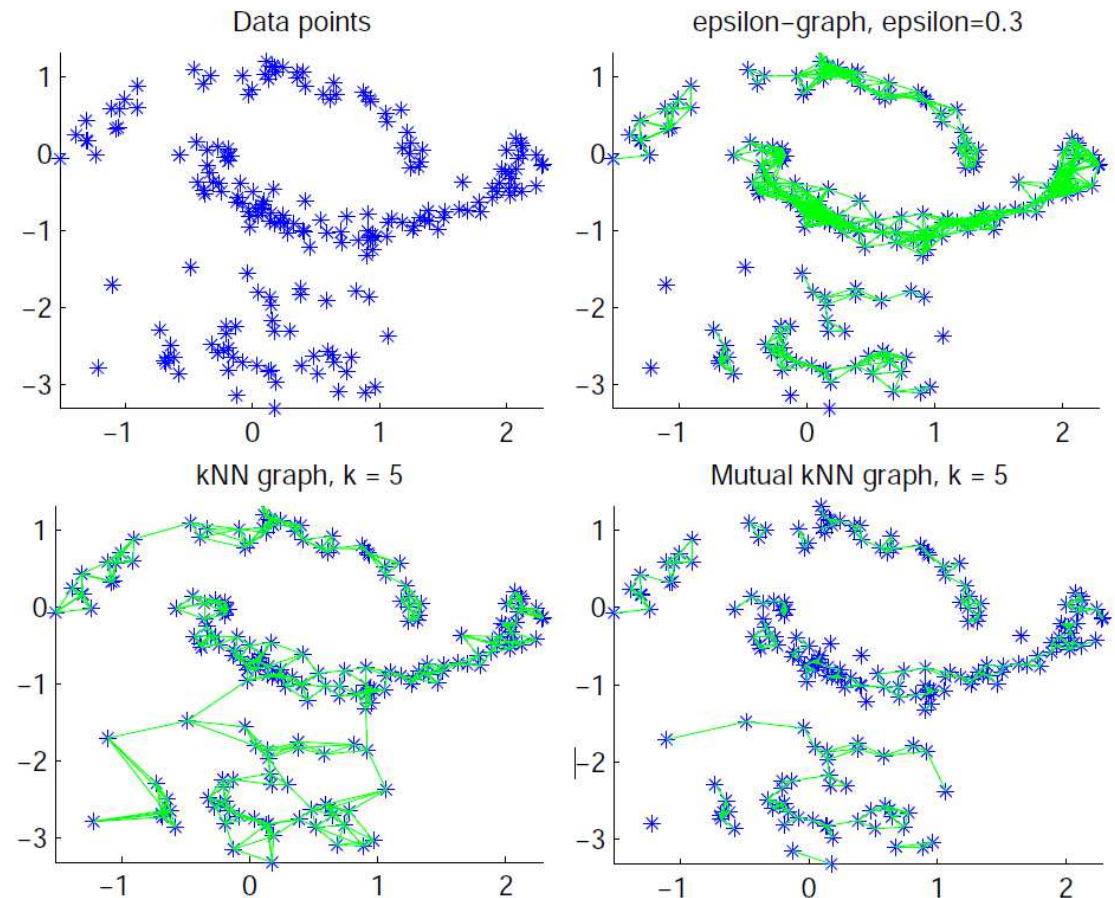


Mining TV sequences with bottom-up clustering



Similarity graphs for spectral clustering

- data point = node in a graph
- edges encode the similarity between two nodes with weights w_{ij}
 - ▷ ϵ -neighbor graphs
 - ▷ k -nearest neighbor graphs
 - ▷ fully connected graphs



[Ulrike von Luxburg. A tutorial on spectral clustering. Statistics and Computing, 17(4):395-416, 2007]

Graph Laplacian

The unnormalized graph Laplacian is the $n \times n$ matrix defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

with \mathbf{D} a diagonal matrix with elements $d_i = \sum_j w_{ij}$.

Important properties of \mathbf{L} :

- \mathbf{L} as n non-negative, real valued eigenvalues
- the smallest eigenvalue is 0 and corresponds to the unit eigen vector $\mathbb{1}$
- the multiplicity of the eigenvalue 0 is the number of connected components

\Rightarrow exploit the eigenvalues of the Laplacian of the similarity graph to perform clustering



Normalized versions of the Laplacian are often used in practice.

The algorithmics of spectral clustering

Input: similarity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$, number of clusters k

Algorithm:

construct a similarity graph from \mathbf{S} with adjacency matrix \mathbf{W}

compute the Laplacian \mathbf{L}

compute the k eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_k$ of \mathbf{L} associated with the k lowest eigenvalues

define $\mathbf{U} \in \mathbb{R}^{n \times k}$ with $\mathbf{u}_1, \dots, \mathbf{u}_k$ as columns

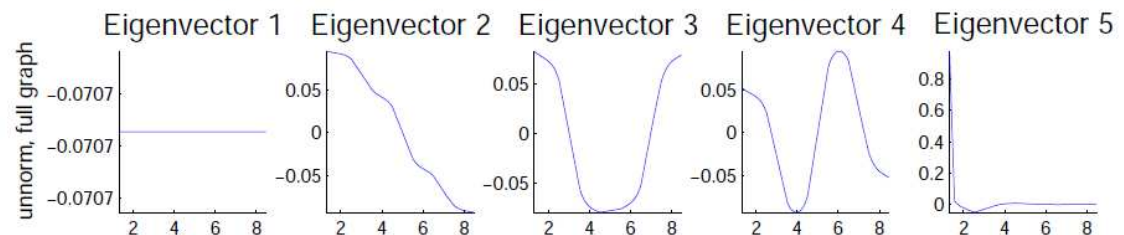
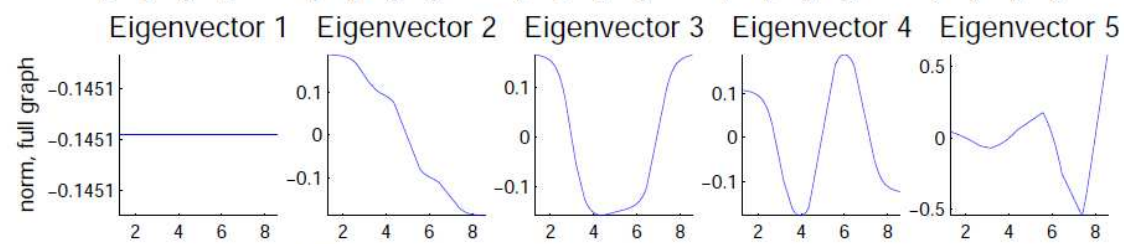
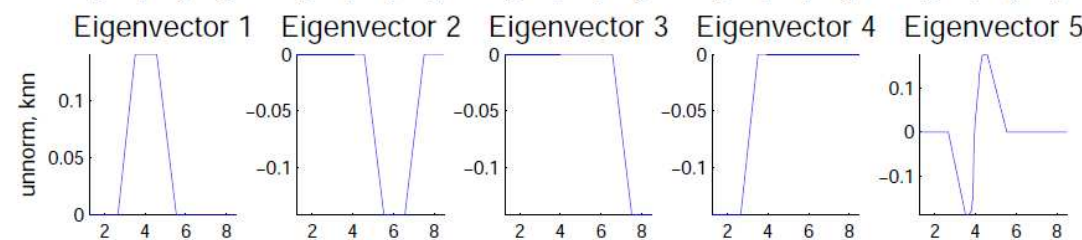
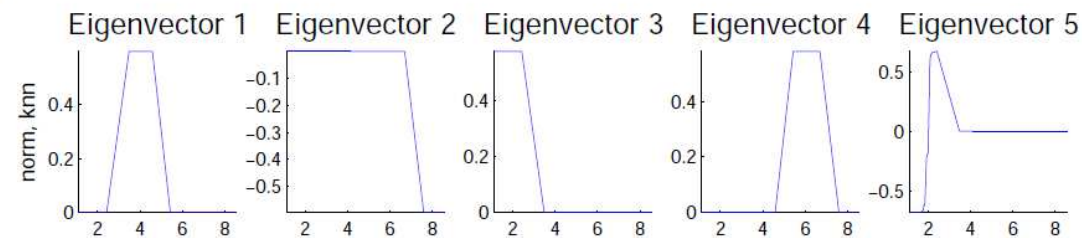
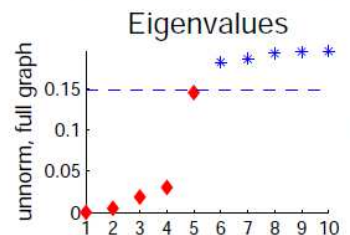
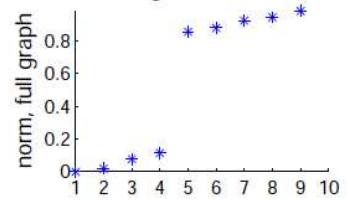
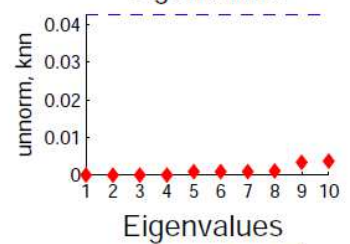
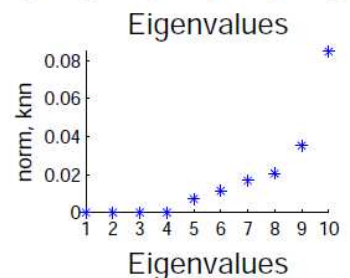
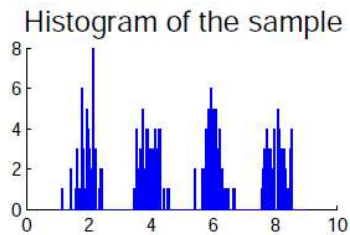
define $\mathbf{y}_i \in \mathbb{R}^k$ the i -th row of \mathbf{U} for $i \in [1, n]$

run k-means clustering (or other) on the vector \mathbf{y}_i

Strong links with

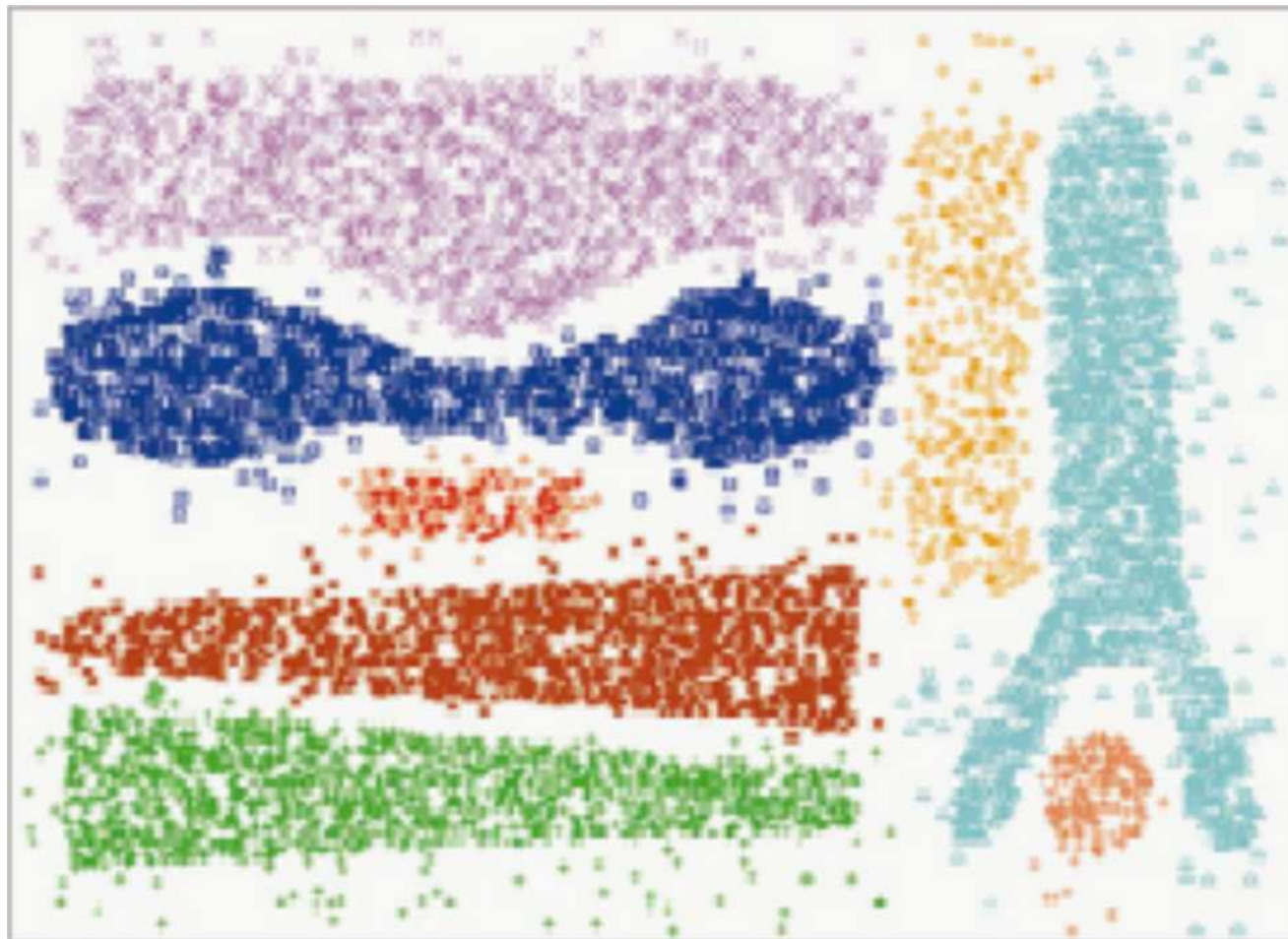
- graph cut algorithms (NCut, RatioCut, MinMaxCut)
- random walks theory (Markov clustering)

Spectral clustering: Toy example



Density-based clustering in a nutshell

- Track density-connected points = neighborhood analysis
- arbitrary shaped clusters, robustness to noise, one pass over the data
- Typical algorithms: DBSCAN, OPTICS, DENCLUE



Density-based clustering in a nutshell (cont'd)

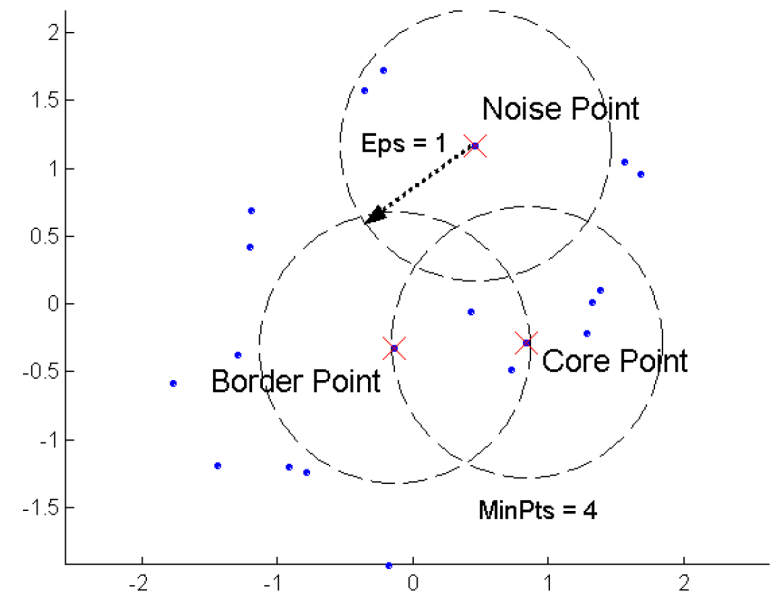
1. label points

- core points: enough points around
- border points: not enough points around but close to a core point
- noise points: none of the above

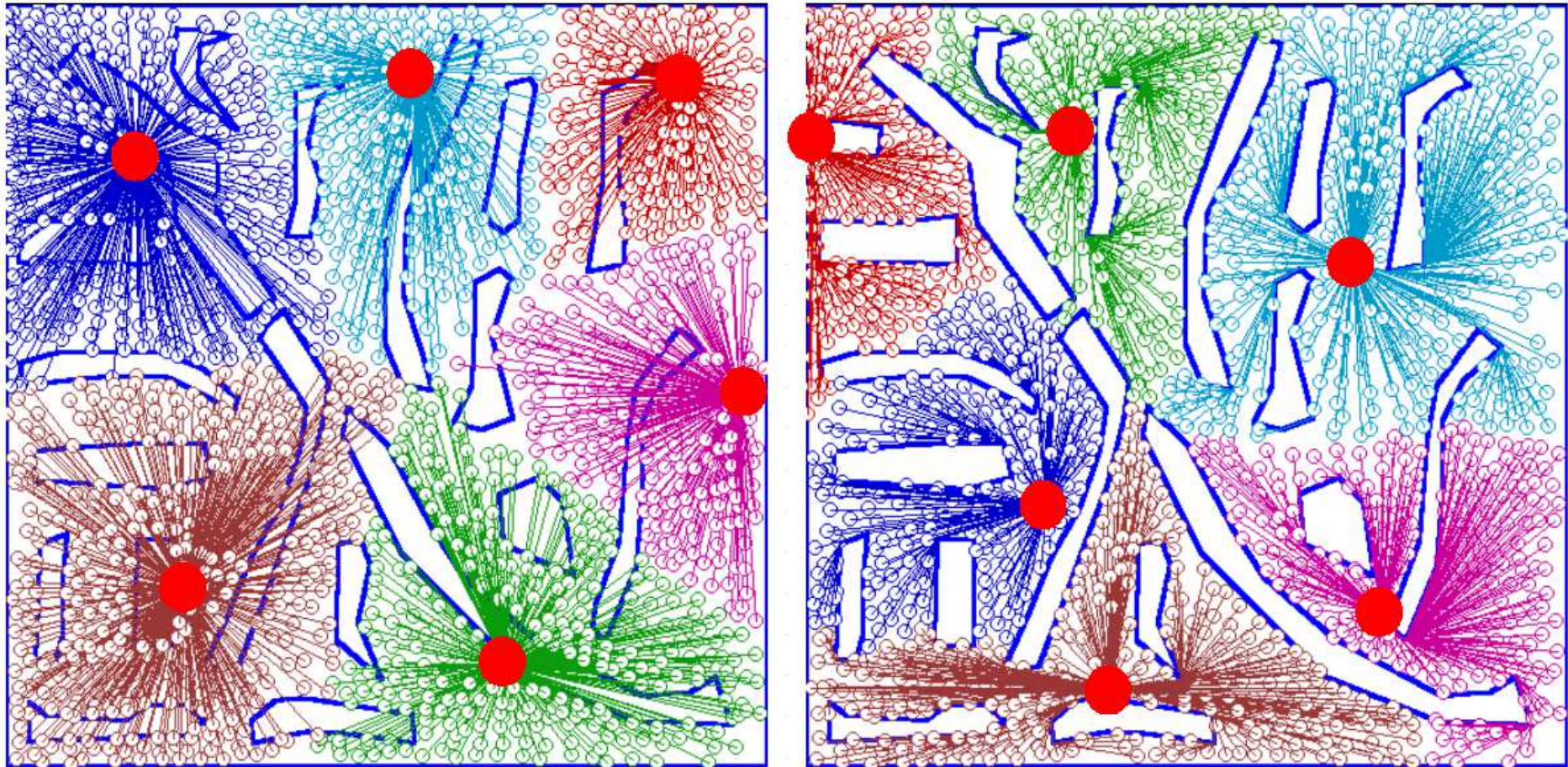
2. eliminate noise points

3. create clusters from core points

4. assign border points to core clusters



Clustering with obstacles



Clustering in high dimension space

- many applications require high-dimension spaces: text documents, images, DNA data, etc.
- high dimensions raises new challenges
 - ▷ many irrelevant dimensions might mask clusters
 - ▷ distance measure becomes meaningless (curse of dimensionality)
 - ▷ cluster may exist only in some subspaces
- several workarounds
 - ▷ feature transformation
 - ◇ PCA/SVD if features are correlated/redundant
 - ▷ feature selection
 - ◇ select feature where nice clusters appear
 - ▷ subspace clustering
 - ◇ find clusters in all possible subspaces (CLIQUE, Proclus)

A fun (and practical) use of text clustering

The screenshot shows a search engine interface with a search bar containing 'rolling stone'. The search results are clustered into a single group titled 'Cluster Rolling Stones Biography with 8 documents (search for more like this)'. The cluster contains 8 documents, each with a title, a brief description, and a URL. The documents are numbered 21, 26, 47, 58, 61, 76, 79, and 96. The search engine interface includes a navigation bar with various search engines (Web, Bing, Yahoo, Wiki, Images, News, Jobs, PubMed, PUT, Blogs) and a search bar with a 'Search' button and 'More options' link. A sidebar on the left shows a tree view of topics related to 'rolling stone', including 'All Topics (98)', 'Rolling Stones News (19)', 'Rolling Stones Biography (8)', 'Rolling Stones Lyrics (8)', 'Rolling Stones are an English (7)', 'Mick Jagger (6)', 'Rolling Stone Magazine (6)', 'Rolling Stones Began (6)', 'Rolling Stones Tickets (5)', 'Links (4)', and 'MP3 Downloads (4)'. The footer of the search engine interface shows the query 'rolling stone -- Source: Yahoo (98 results: 883 ms) -- Clusterer: Lingo (78 ms)' and the version information 'v3.4.1 | build 61 | 2010-09-07 08:19:05 © 2002-2010 Stanislaw Osinski, Dawid Weiss'.

Cluster **Rolling Stones Biography** with 8 documents ([search for more like this](#))

- 21 [The Rolling Stones - WolfgangVault.com](#)
... with **The Rolling Stones** biography, detailed summary of the shows and related **The Rolling Stones** memorabilia items, including vintage tees, poster ...
<http://www.wolfgangsvault.com/the-rolling-stones/>
- 26 [The History of Rock Music. Rolling Stones: biography ...](#)
A guide to **Rolling Stones**: biography, discography, reviews, links
<http://www.scaruffi.com/vol1/stones.html>
- 47 [The Rolling Stones Biography - Biography.com](#)
A look at the legendary group **The Rolling Stones** on Biography.com.
<http://www.biography.com/articles/The-Rolling-Stones-9462754>
- 58 [The Rolling Stones Lyrics - Lyrics, Video, Biography](#)
Find the **Rolling Stones** Lyrics to 'Satisfaction' and Many More Here! ... **The Rolling Stones** are a British rock band who rose to prominence during the mid-1960s. ...
<http://www.rolling-stones-lyrics.com/>
- 61 [The Rolling Stones - VH1.com](#)
Rolling Stones biography, interviews, news, tour dates, and a discography with album information and audio clips.
http://www.vh1.com/artists/az/rolling_stones/artist.jhtml
- 76 [Rolling Stones - rock and roll music at the RockSite](#)
The Rolling Stones at The RockSite - biography, links, discography, albums, bio FAQ, pics, concerts, tour dates, tickets
<http://www.rocksite.info/r-rolling-stones.htm>
- 79 [Rolling Stones Discography and Music at CD Universe](#)
Low prices on **Rolling Stones** discography of music albums at CD Universe, with top rated service, **Rolling Stones** songs, discography, biography, cover ...
http://www.cduniverse.com/sresult.asp?style=music&HT_Search_Info=Rolling+Stones&HT_Search=ARTIST&frm=lk_sway
- 96 [The Rolling Stones - Discography, biography, music, MP3s ...](#)
The Rolling Stones is a group formed in 1962. Their discography includes Hot Rocks 1964-1971, Exile on Main St., Sticky Fingers, Forty Licks and Let It Bleed. ...
<http://www.soundunwound.com/music/the-rolling-stones/382>

Query: rolling stone -- Source: Yahoo (98 results: 883 ms) -- Clusterer: Lingo (78 ms)

v3.4.1 | build 61 | 2010-09-07 08:19:05 © 2002-2010 Stanislaw Osinski, Dawid Weiss

Another example mixing all you've seen

OVERALL APPROACH

Images → Lucene → Sparse Feature Matrix → Combined Sparse Matrix → DBSCAN + Spectral → Clusters

INCREMENTAL CLUSTERING

- 1) CLUSTER INITIAL WINDOW
- 2) GROW WINDOW, CLUSTER AGAIN
- 3) IDENTIFY STABLE CLUSTERS
- 4) CONTINUE IGNORING STABLE CLUSTERS

SPARSE SIMILARITY MATRIX

LUCENE INDEX

... CONSTRUCT A LUCENE INDEX OF ALL THE IMAGES WE HOPE TO CLUSTER. CONSTRUCT A QUERY FOR EACH IMAGE AND USE ONLY RETURNED RESULTS FOR THE NEXT STAGE...

FEATURE SIMILARITY MATRIX

... EXTRACT FEATURES FOR EACH IMAGE, USING CUSTOM DISTANCE METRICS COMPUTE THE SIMILARITY OF TWO IMAGES. CONSTRUCT A SIMILARITY MATRIX FOR EACH FEATURE FOR THE NEXT STAGE...

MATRIX COMBINATION

$$\sum W_{ij} = 1$$

... WEIGHT EACH FEATURE SIMILARITY MATRIX BY SOME LEARNT WEIGHT AND SUM TOGETHER TO FORM A SINGLE FEATURE FUSED SIMILARITY MATRIX READY FOR CLUSTERING...

CLUSTERING

DBSCAN

DBSCAN FINDS DATA WHICH ARE MUTUALLY DENSITY-CONNECTED AND IDENTIFIES NOISE AS THOSE DATA WHICH ARE NOT. OUR DBSCAN WORKS DIRECTLY WITH THE AGGREGATED SIMILARITY MATRIX AS WELL AS WITH STANDARD GEOMETRIC POINTS

SPECTRAL CLUSTERING

- 1) RAW DATA
- 2) LAPLACIAN EIGENVECTORS
- 3) CLUSTERED USING DBSCAN

SPECTRAL CLUSTERING USES THE SIMILARITY MATRIX TO CALCULATE A GRAPH LAPLACIAN WHOSE LOW EIGENVALUED EIGENVECTORS CAN BE USED AS A SPACE WITHIN WHICH ANOTHER CLUSTERING ALGORITHM CAN BETTER CLUSTER DATA. WE USE DBSCAN FOR THIS.

RESULTS

WEIGHT SELECTION

A SIMPLEX SEARCH WAS USED TO FIND THE BEST WEIGHTING OF FEATURES. HOWEVER THE DIFFERENCE BETWEEN THE TOP 1000 POINTS ON THE SIMPLEX WERE SMALL SO AN AVERAGE WEIGHTING COMBINING THE TOP 1000 WEIGHTS WAS ALSO USED

SETTING	TIME TAKEN	TIME POSTED	LOCATION	TEXT DESC	TEXT TITLE	TEXT TAGS
BEST	2	0	1	1	0	3
AVERAGE	2.1	1.8	1.4	0.7	0.3	1.7

OFFICIAL RESULTS

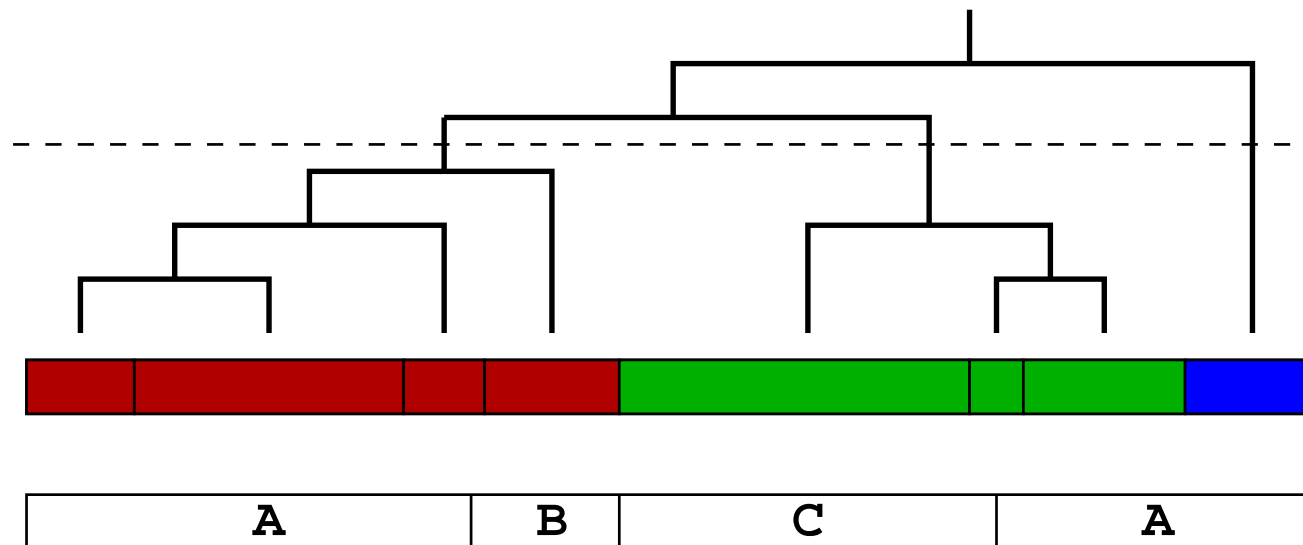
IN ALL EXPERIMENTAL SETTINGS SUBMITTED THE INCREMENTAL ALGORITHM WAS USED. THOUGH OUR INCREMENTAL TECHNIQUE ALLOWED FOR SPECTRAL CLUSTERING ACROSS A RELATIVELY LARGE DATASET, WE FOUND DBSCAN PERFORMED BEST OVERALL

	F1	NMI	F1(DIV)	RB F1	DIV F1
DBSCAN (BEST)	0.945	0.985	0.935	0.059	0.887
SPECTRAL (BEST)	0.911	0.977	0.882	0.058	0.853
DBSCAN (AVG)	0.946	0.985	0.936	0.060	0.886
SPECTRAL (AVG)	0.902	0.974	0.866	0.057	0.846

MediaEval 2013 Social Event Detection Task

About the evaluation of clustering

- subjective evaluation by inspecting clusters
- within cluster distortion (if a meaningful metric exists)
- any arbitrary objective quality criterion (but none out of the shelf)
- special case with temporal segmentations



To get to know more ...

- So many textbooks (look for clustering and data mining)!
- So many resources on the Web (not in Wikipedia this time)
 - ▷ free code available everywhere
 - ▷ lot's of tutorials/lessons
 - ◇ Dr. HOI Chu's course:
<https://svn.mosuma.net/r4000/doc/course/ci6227/public/lectures/lecture07cluster.pdf>
 - ◇ Prof. Jiawei Han courses: <http://www.cs.uiuc.edu/hanj>