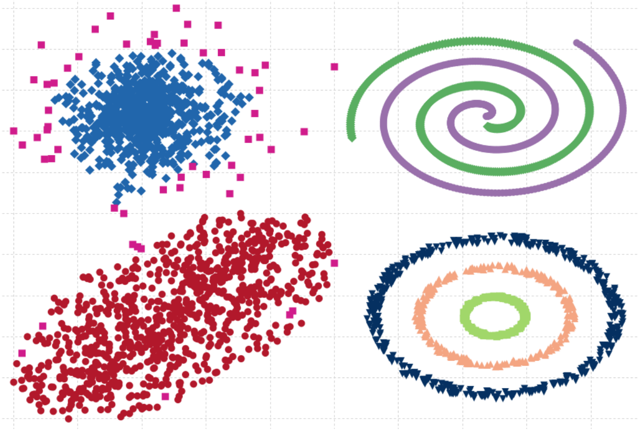
Multi-objective clustering

Tomas Barton
tomas.barton@fit.cvut.cz

01000110 Fakulta
01001001 Informačních
01010100 Technologií

Goal

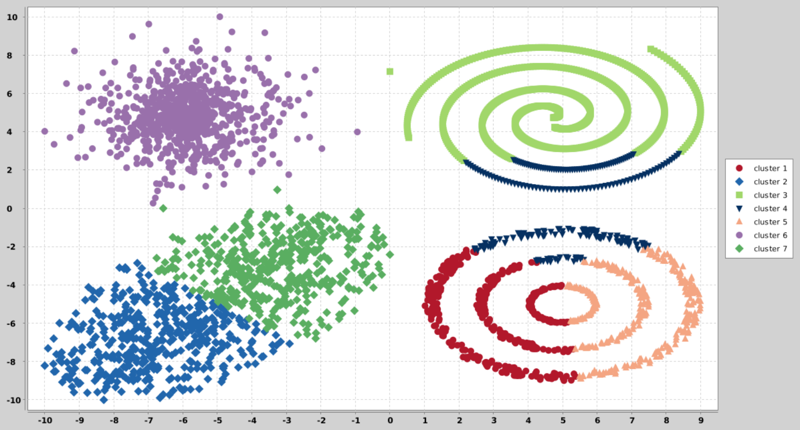
- goal: a result matching human judgment
- how many objectives do we need to obtain this?



Inspired by *Jain, Anil K. "Data clustering: 50 years beyond K-means." Pattern recognition letters 31.8 (2010): 651-666.*

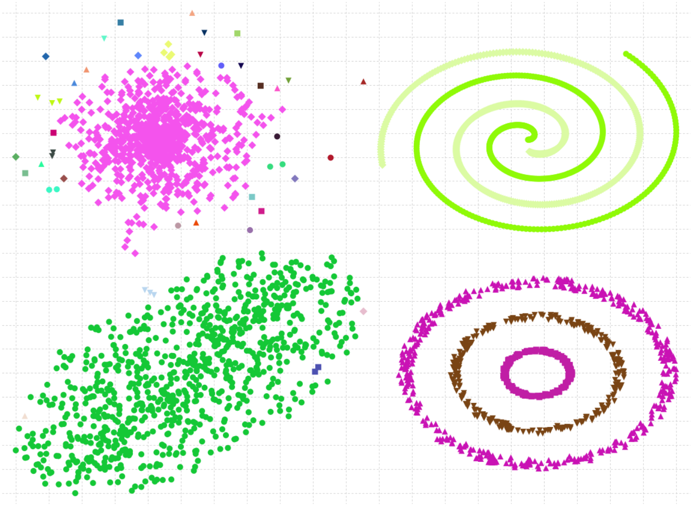
k-means clustering

- most algorithms optimize single objective
- e.g. minimize square distance inside a cluster

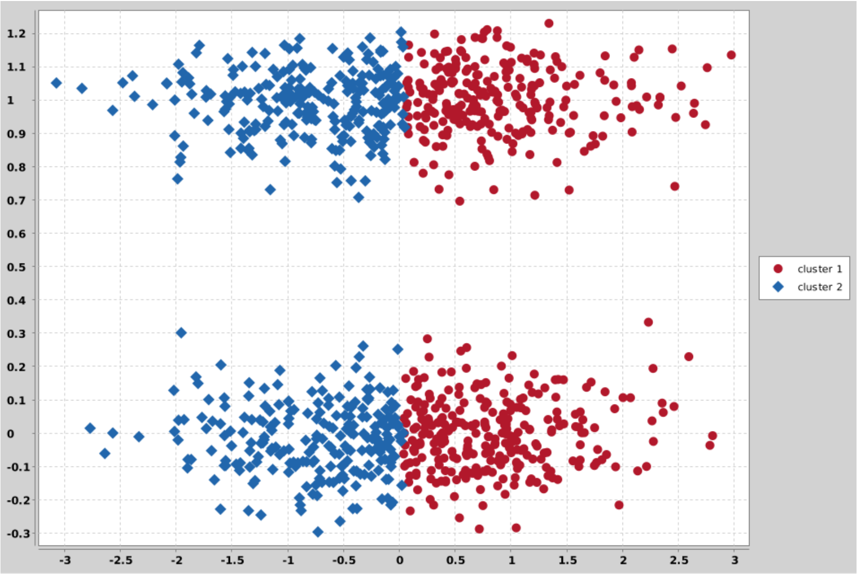


Single-Link clustering

- capable of discovering arbitrary shaped clusters
- but too sensitive to noise

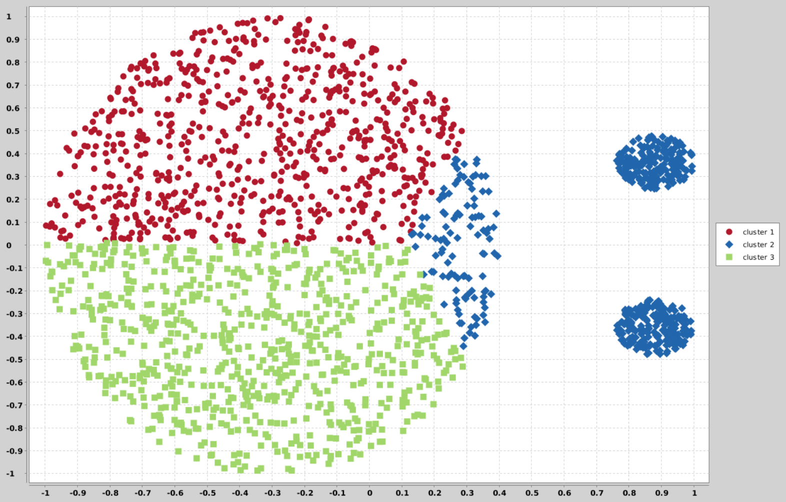


Limitation of k-means



Problems

k-means ($k = 3$)



Interesting clustering approaches

2002 – Strehl, Ghosh. "Cluster ensembles—a knowledge reuse framework for combining multiple partitions." The Journal of Machine Learning Research

2004 – Law, Topchy, Jain. "Multiobjective data clustering." Computer Vision and Pattern Recognition, IEEE Conference

2007 MOCK – Handl, Knowles. "An evolutionary approach to multiobjective clustering." Evolutionary Computation, IEEE Transactions

2010 – Forestier, Gançarski, Wemmert. "Collaborative clustering with background knowledge." Data & Knowledge Engineering

Cluster Ensembles

Strehl and Gosh (2002)

CSPA – Cluster-based Similarity Partitioning Algorithm: combines multiple clustering based pairwise similarity of instances in clusters

HGPA – Hyper-Graph Partitioning Algorithm: tries partitioning a hypergraph where hyperedges represent clusters

MCLA – Meta-CLustering Algorithm: tries to identify groups of clusters (meta-clusters) and consolidate them

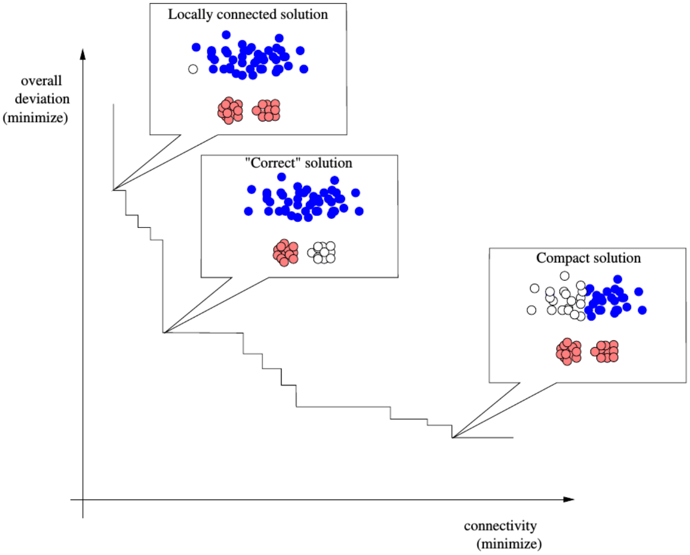
Multiobjective data clustering

Law, Topchy

- uses resampling and combines clusterings
- multi-objective clustering as NP-hard combinatorial optimization problem

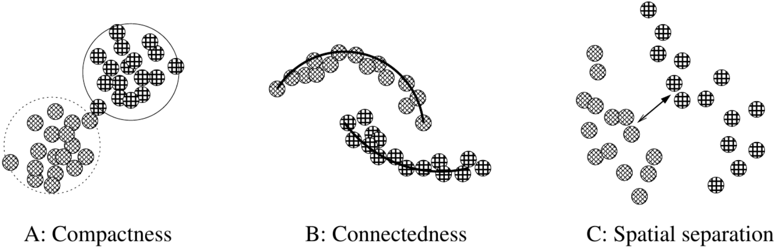
MOCK

Multiobjective clustering with automatic k-determination



MOCK

Possible objectives



$$Conn(\mathbb{C}) = \sum_{i=1}^{N} \left(\sum_{j=1}^{L} x_{i,nn_{ij}} \right), \qquad (1)$$

where

$$x_{r,s} = \begin{cases} \frac{1}{j}, & \text{if } \nexists C_k : r \in C_k \wedge s \in C_k \\ 0, & \text{otherwise,} \end{cases}$$

nn_{ij} is the jth nearest neighbour of item i, N is the size of the data set and L is a parameter determining the number of neighbours that contribute to he connectivity measure

MOCK

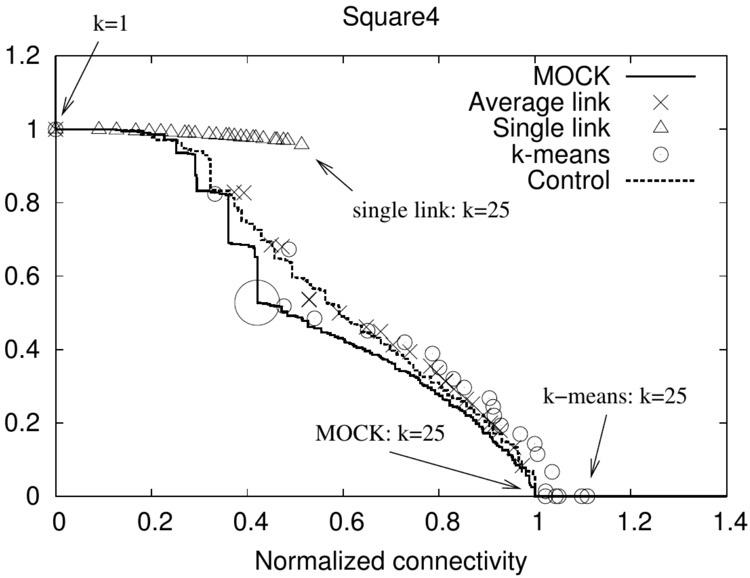
Deviation (Compactness)

$$Dev(\mathbb{C}) = \sum_{C_k \in \mathbb{C}} \sum_{i \in C_k} \delta(i, \mu_k) \tag{2}$$

where $\mathbb{C}$ is a set of all clusters, μ_k is the centroid of the cluster C_k and $\delta(.,.)$ is a chosen distance function

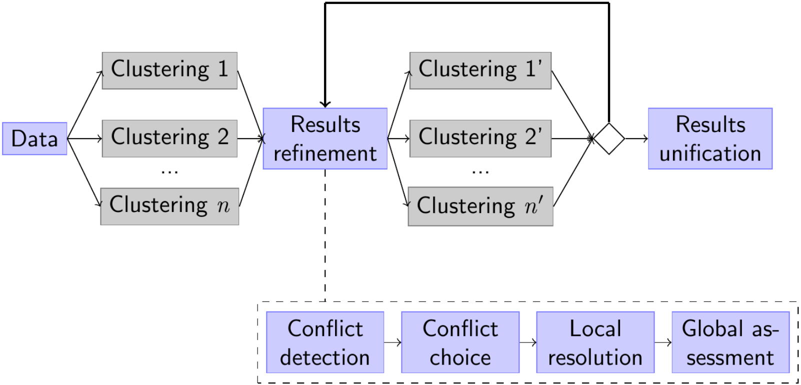
MOCK

Solution front



Collaborative clustering

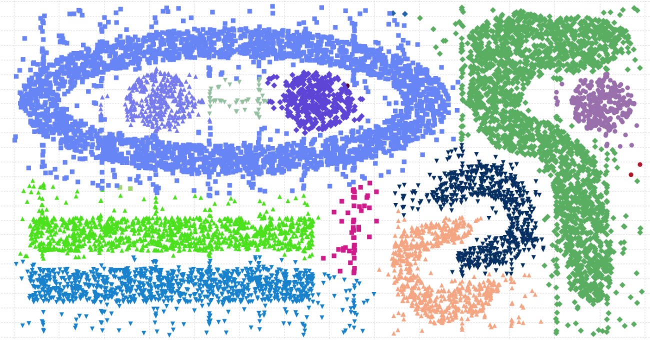
Forestier, Gançarski, Wemmert



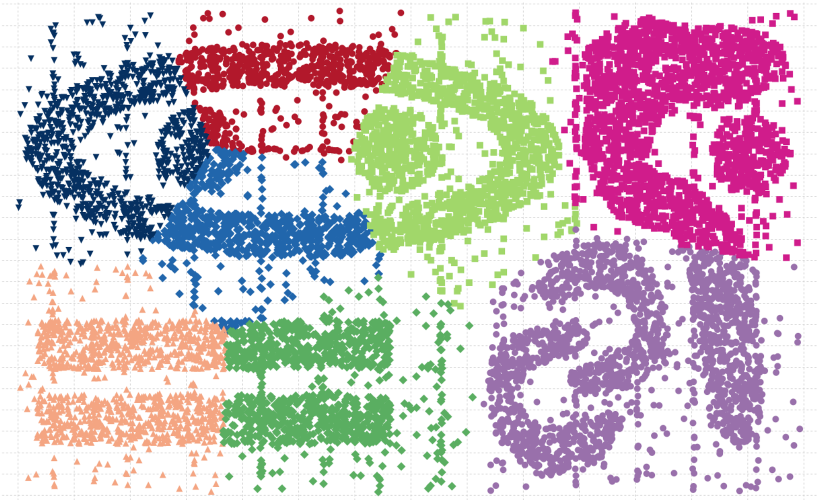
Let’s go back…

Chameleon

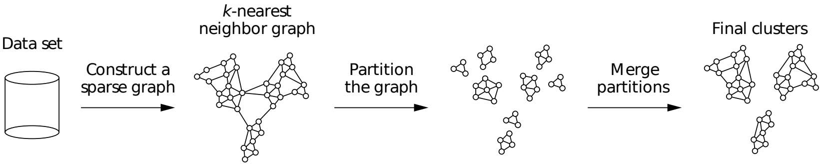
1999 Karypis, George, Eui-Hong Han, and Vipin Kumar. "Chameleon: Hierarchical clustering using dynamic modeling." Computer 32.8



k-means



Chameleon algorithm

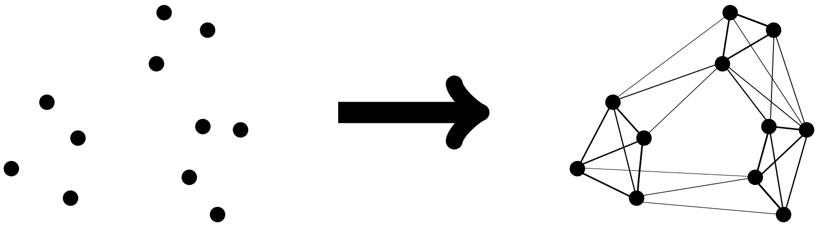


1. create k-nearest neighbor graph
2. partion the graph
3. merge partitions

Chameleon

1. k-nn

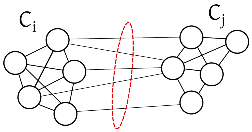
- dataset represented as a graph



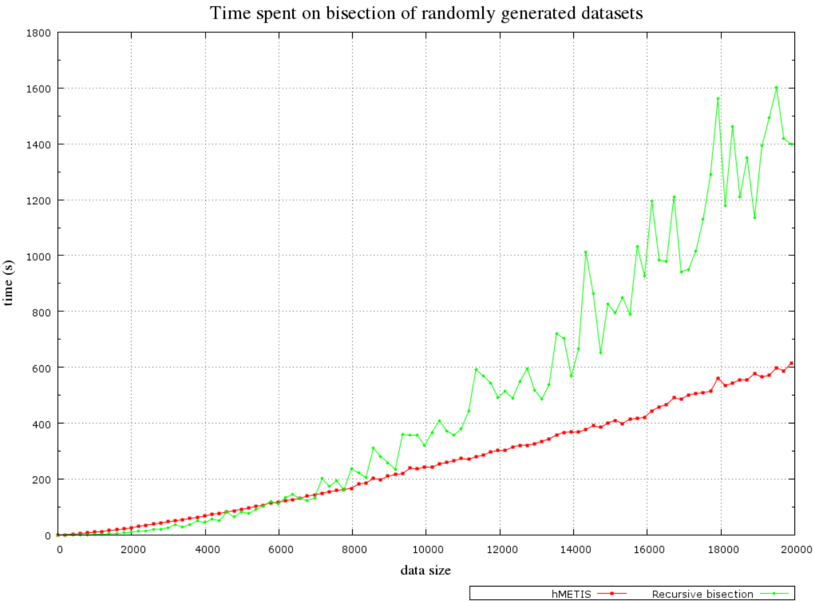
Chameleon

2. partitioning

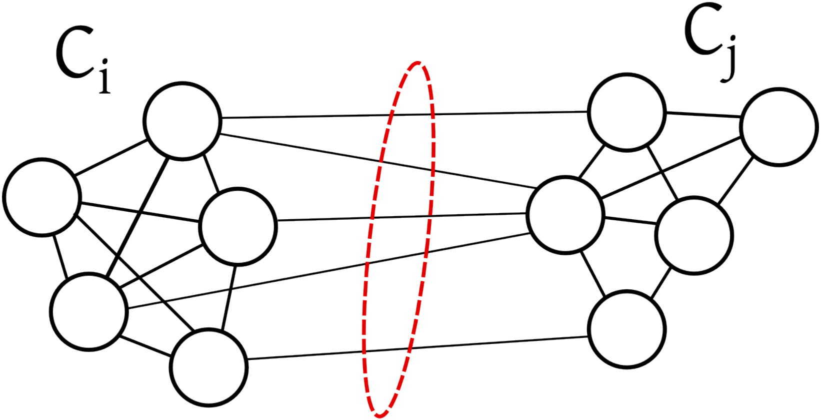
- split graph into many components
- minimalize egdes cut
- optimal bisection is NP-complete, thus we use approximation:
  - Kerighan-Lin $\mathcal{O}(n^3)$
  - Spectral bisection $\mathcal{O}(n^3)$
  - Fiduccia-Matheyses $\mathcal{O}(|E|)$
  - METIS / hMETIS



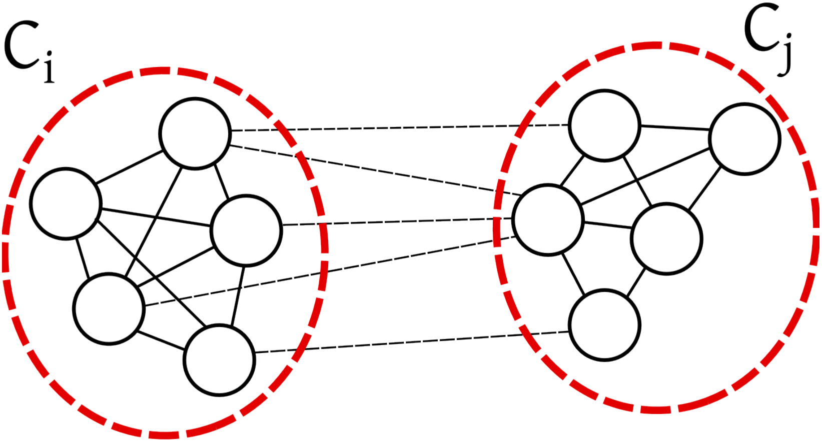
Complexity of partitioning phase



$$\bar{\phi}(C_i, C_j)$$



$$\bar{\phi}(C_i), \bar{\phi}(C_j)$$



Chameleon

3. merging

- find best candidate for merging

$$R_{IC}(C_i, C_j) = \frac{\phi(C_i, C_j)}{\frac{\phi(C_i) + \phi(C_j)}{2}} = \frac{2\phi(C_i, C_j))}{\phi(C_i) + \phi(C_j)}$$

$$R_{RC}(C_i, C_j) = \frac{\bar{\phi}(C_i, C_j)}{\frac{|C_i|}{|C_i| + |C_j|}\bar{\phi}(C_i) + \frac{|C_j|}{|C_i| + |C_j|}\bar{\phi}(C_j)}$$

$$Sim(C_i, C_j) = R_{CL}(C_i, C_j)^{\alpha} \cdot R_{IC}(C_i, C_j)^{\beta}$$

Chameleon

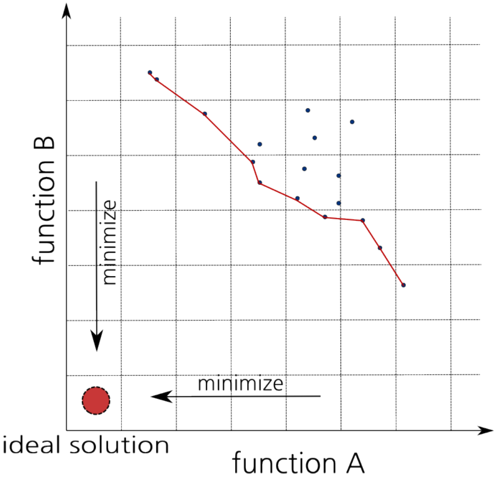
Combination of objectives

$$Sim(C_i, C_j) = R_{CL}(C_i, C_j)^{\alpha} \cdot R_{IC}(C_i, C_j)^{\beta}$$

- α, β – user defined priorities
- $\alpha = 2$
- $\beta = 1$

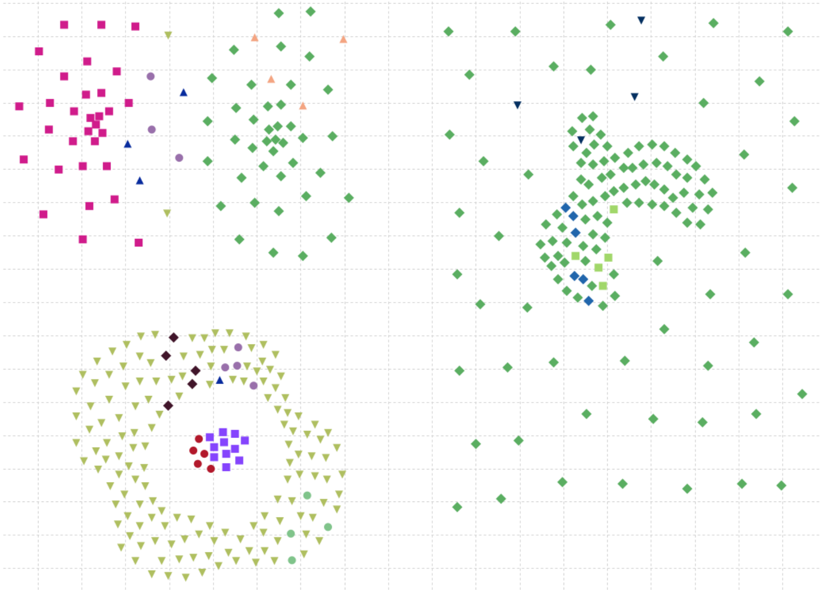
Chameleon MO

- inspired by NSGA-II
- uses same objectives as Chameleon
- relatively “fast”

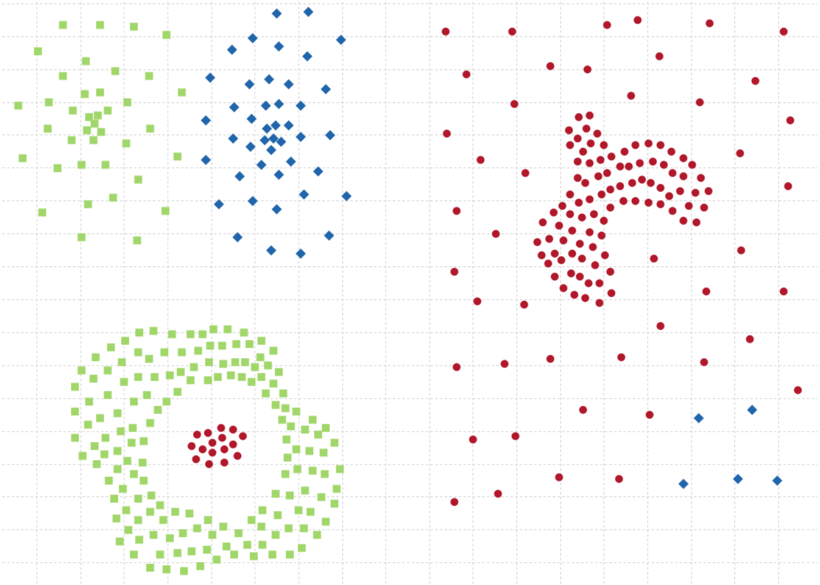


Chameleon 1

Standard similarity



Chameleon MO



Chameleon MO

- hierarchical merging has complexity $O(n^2)$
- $\Rightarrow$ multi-objective sorting (2 objectives) $O(n^4)$

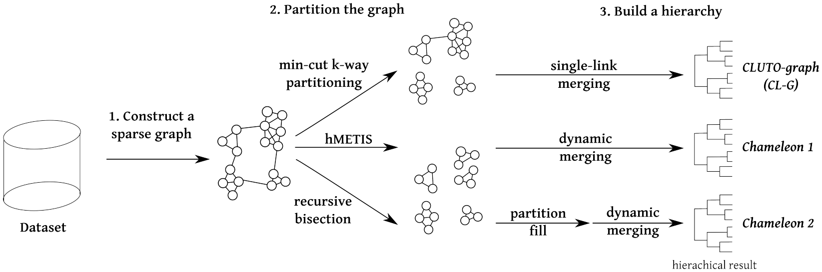
Chameleon MO

- hierarchical merging has complexity $\mathrm{O}\left(n^2\right)$
- $\Rightarrow$ multi-objective sorting (2 objectives) $\mathrm{O}\left(n^4\right)$

approximation required

- limited number of fronts
- blacklist
- 3rd level sorting
- heap rebuild

Chameleon 2


2. Partition the graph
3. Build a hierarchy
min-cut k-way partitioning
single-link merging
CLUTO-graph (CL-G)
1. Construct a sparse graph
hMETIS
dynamic merging
Chameleon 1
Dataset
recursive bisection
partition fill
dynamic merging
Chameleon 2
hierachical result

Chameleon 2

improved similarity measure

$$Sim_{\text{shat}}(C_i, C_j) = R_{\text{CLS}}(C_i, C_j)^{\alpha} \cdot R_{\text{ICS}}(C_i, C_j)^{\beta} \cdot \gamma(C_i, C_j)$$

$$R_{CLS}(C_i, C_j) = \frac{\bar{s}(C_i, C_j)}{\frac{|E_{C_i}|}{|E_{C_i}|+|E_{C_j}|}\,\bar{s}(C_i) + \frac{|E_{C_j}|}{|E_{C_i}|+|E_{C_j}|}\,\bar{s}(C_j)}$$

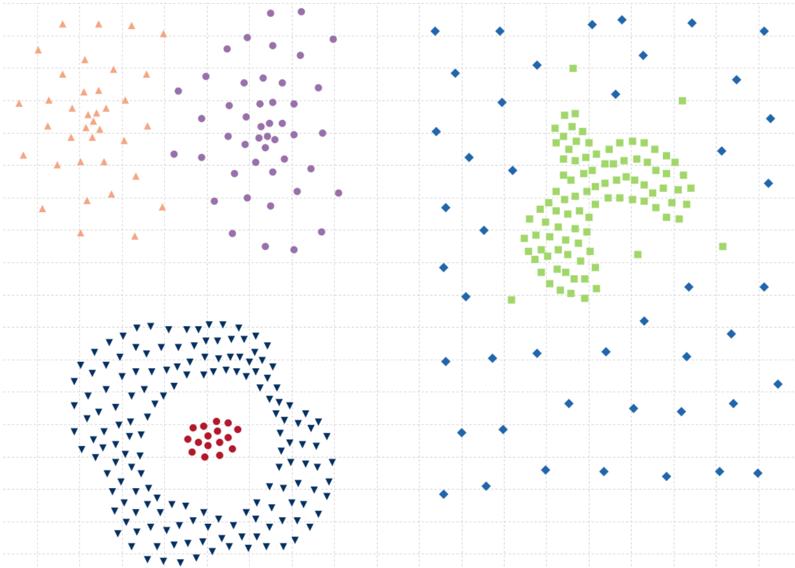
$$R_{\text{ICS}}(C_i, C_j) = \frac{\min\{\bar{s}(C_i), \bar{s}(C_j)\}}{\max\{\bar{s}(C_i), \bar{s}(C_j)\}}$$

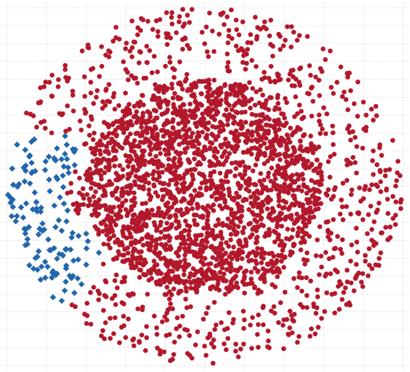
$$\gamma(C_i, C_j) = \frac{|E_{C_{i,j}}|}{\min(|E_{C_i}|, |E_{C_j}|)}$$

Where $\bar{s}(C_i)$ is defined as sum of edges' weights in a cluster

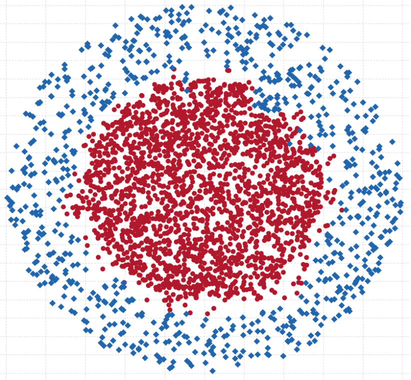
Chameleon 2

compound dataset





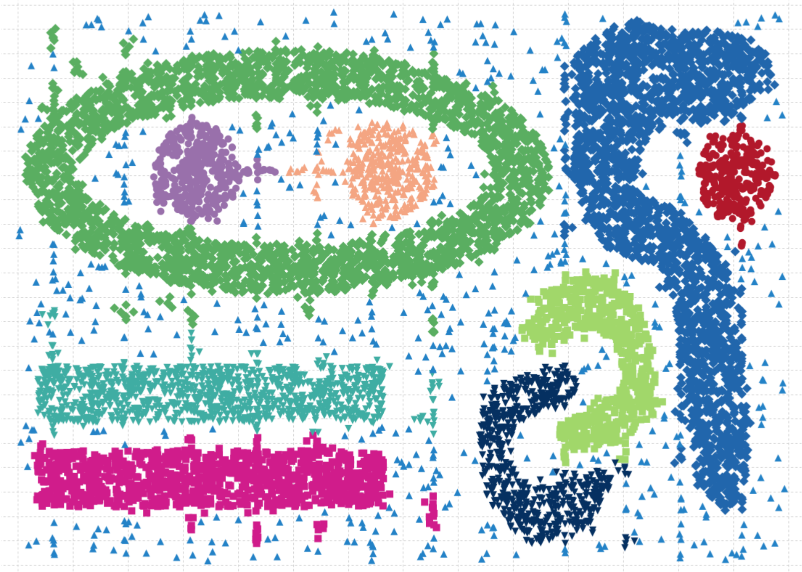
(a) CLUTO, NMI = 0.18



(b) Ch2, NMI = 0.76

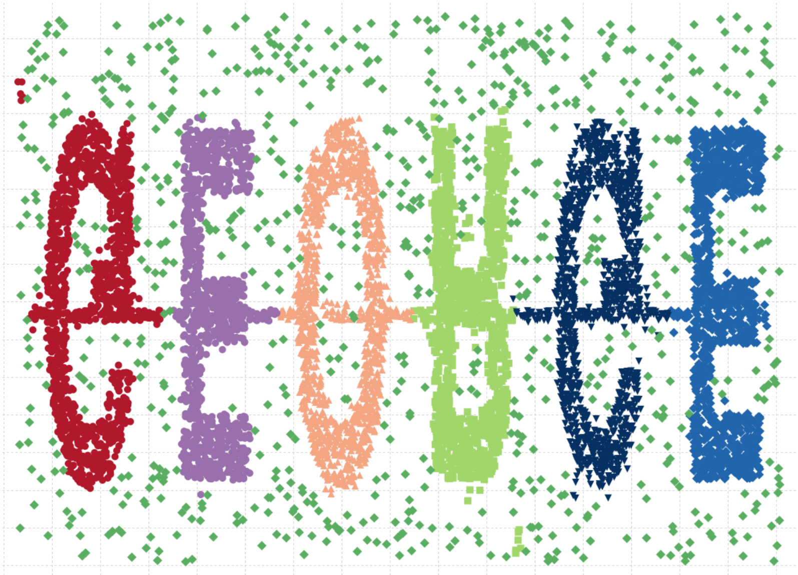
Chameleon 2

cluto-t7.10k



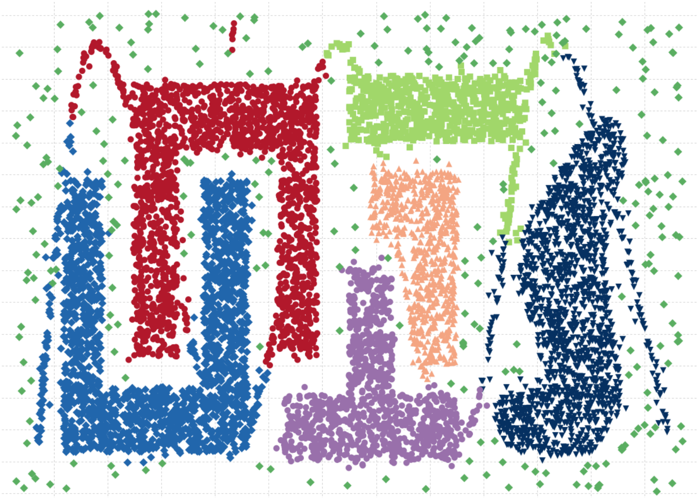
Chameleon 2

cluto-t7.10k



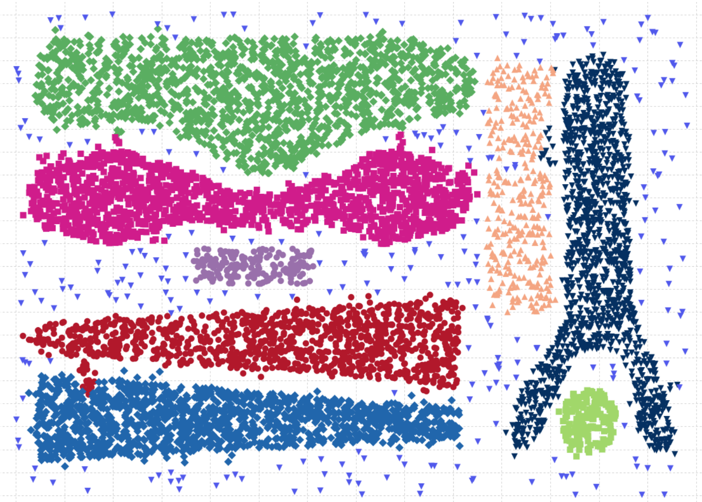
Chameleon 2

cluto-t4.8k



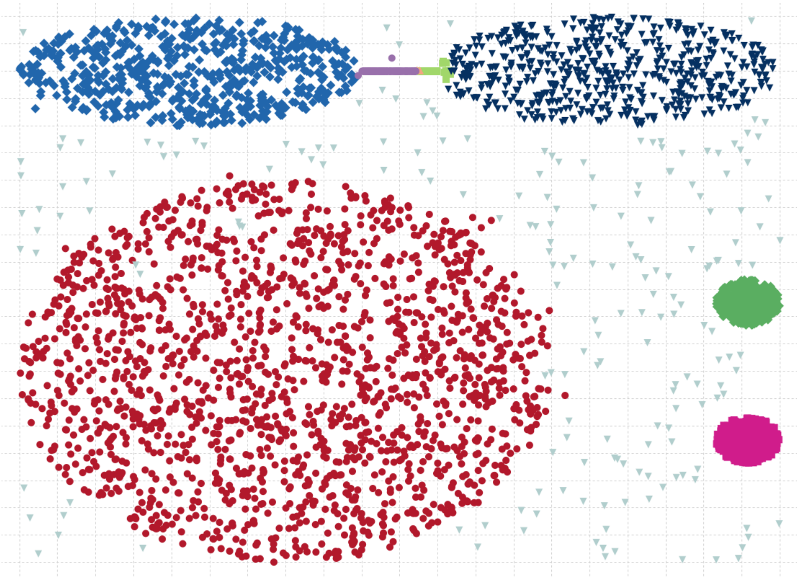
Chameleon 2

cluto-t8.8k



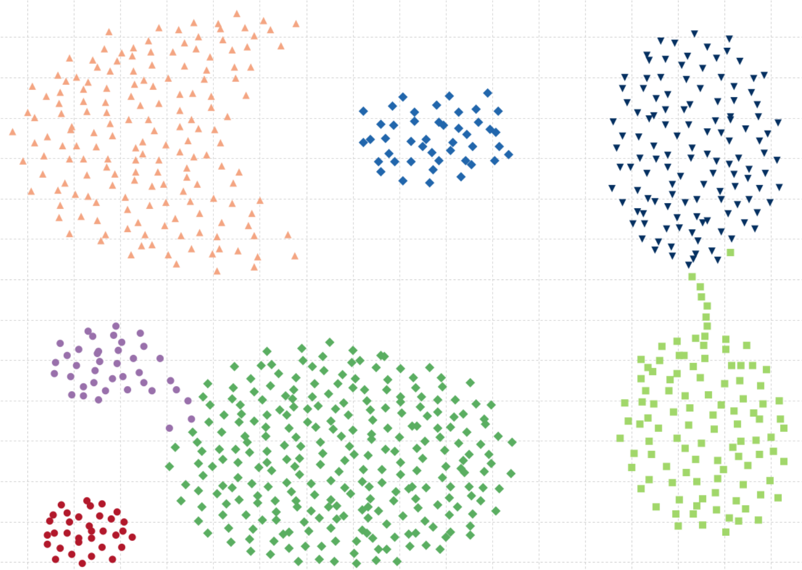
Chameleon 2

CURE



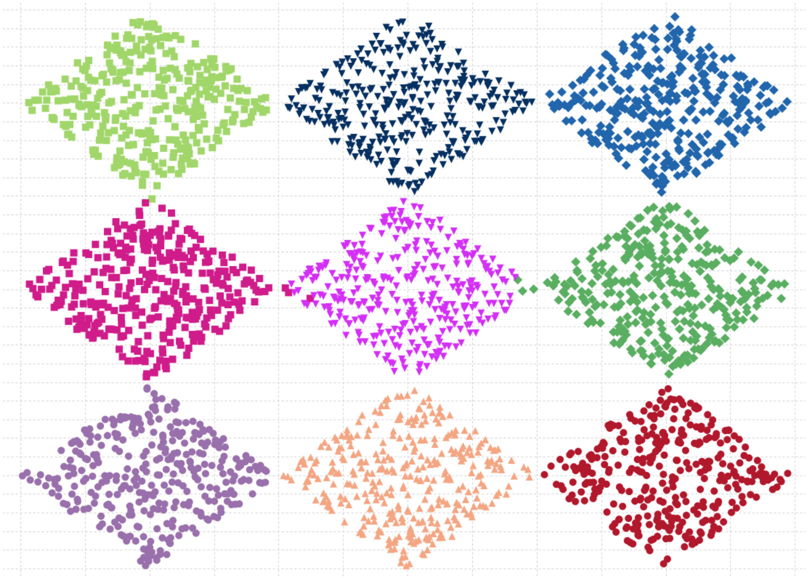
Chameleon 2

aggregation

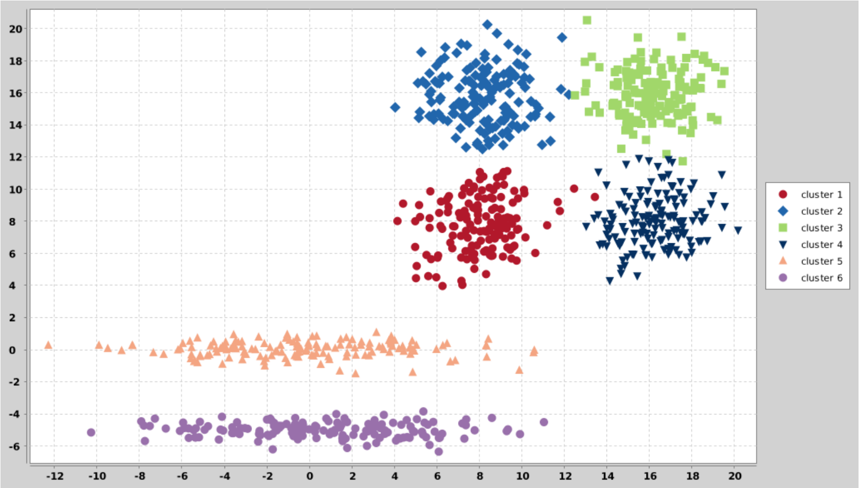


Chameleon 2

diamond9

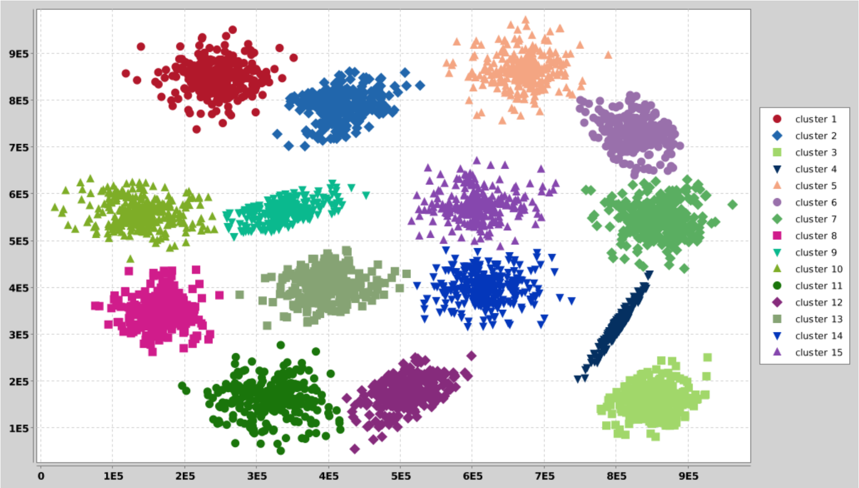


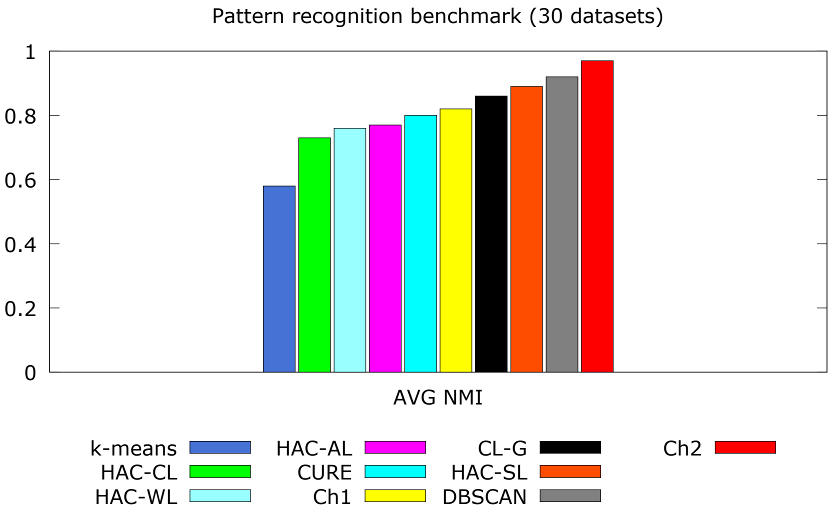
longsquare



20
18
16
14
12
10
8
6
4
2
0
-2
-4
-6
-12
-10
-8
-6
-4
-2
0
2
4
6
8
10
12
14
16
18
20
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6

s1


9E5
8E5
7E5
6E5
5E5
4E5
3E5
2E5
1E5
0
1E5
2E5
3E5
4E5
5E5
6E5
7E5
8E5
9E5
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
cluster 7
cluster 8
cluster 9
cluster 10
cluster 11
cluster 12
cluster 13
cluster 14
cluster 15


Pattern recognition benchmark (30 datasets)
1
0.8
0.6
0.4
0.2
0
AVG NMI
k-means
HAC-CL
HAC-WL
HAC-AL
CURE
Ch1
CL-G
HAC-SL
DBSCAN
Ch2

Clustering evaluation

(unsupervised)

Clustering objectives

objective function

most metrics considers following criteria:

$$p = \frac{\sum \text{distances in a cluster}}{\sum \text{distances between clusters}}$$

Clustering objectives

C-index

$$f_{\text{c-index}}(\mathbb{C}) = \frac{S_w - S_{min}}{S_{max} - S_{min}}$$

where

- S_w is the sum of the within cluster distances
- S_{min} is the sum of the N_w smallest distances between all the pairs of points in the entire dataset. There are N_t such pairs
- S_{max} is the sum of the N_w larges distances between all the pairs of points in the entire dataset

Clustering objectives

Davies-Bouldin

Davies-Bouldin indexs combines two measures, one related to dispersion and the other to the separation between different clusters

$$f_{\text{DB}}(\mathbb{C}) = \frac{1}{K} \sum_{i=1}^{K} \max_{i \neq j} \left(\frac{\bar{d}_i + \bar{d}_j}{d(\mathbf{c}_i, \mathbf{c}_j)} \right)$$

where $d(\mathbf{c}_i, \mathbf{c}_j)$ corresponds to the distance between the center of clusters C_i and C_j, $\bar{d}_i$ is the average within-group distance for cluster C_i.

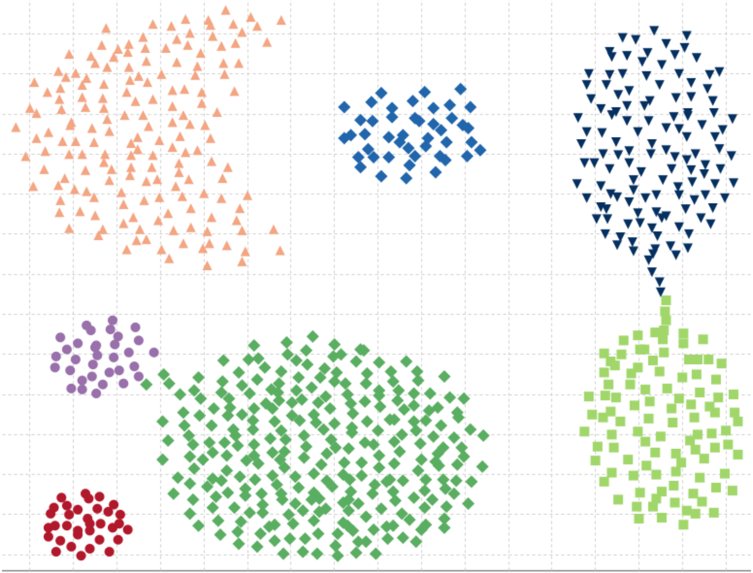
$$\bar{d}_i = \frac{1}{|C_i|} \sum_{l=1}^{|C_i|} d(\mathbf{x}_i(l), \bar{\mathbf{x}}_i)$$

Clustering objectives

- Caliński-Harabasz – VCR (1974)
- Dunn index (1974)
- AIC (1974)
- C-index (1976)
- Gamma (1976)
- BIC (1978)
- Davies and Bouldin (1979)
- Silhouette (1987)
- Gap statistic (2001)
- Connectivity (2004)
- Compactness (2006)

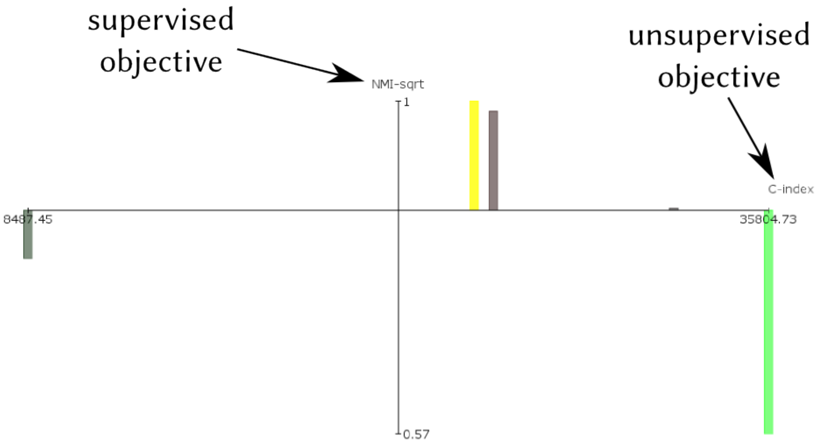
Simplified problem

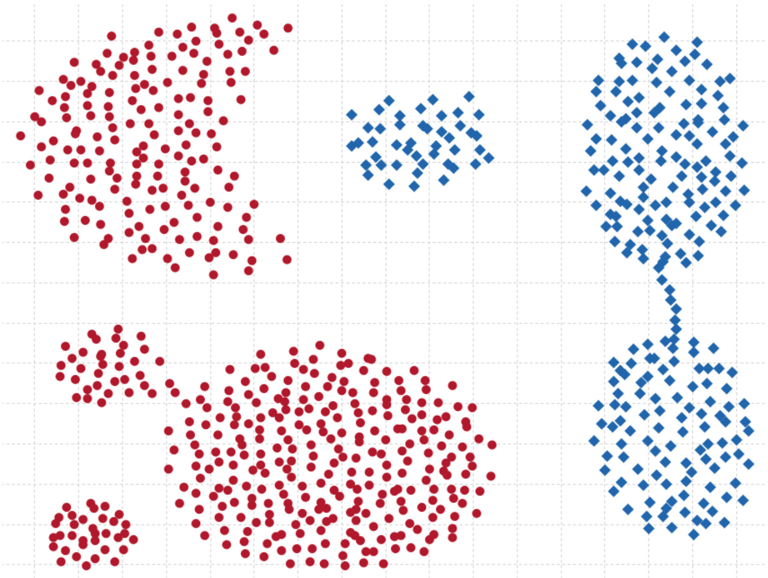
- given a set $\mathbb{C}$ of clustering solution $\{C_1, C_2, \ldots, C_k\}$ created from the same dataset
- we use a supervised function as reference $f_{supervised}(\mathbb{C}) \rightarrow \texttt{rank}\{r_1, r_2, \ldots, r_k\}$
- and an unsupervised function $g_{unsupervised}(\mathbb{C}) \rightarrow \texttt{rank}\{r_1, r_2, \ldots, r_k\}$



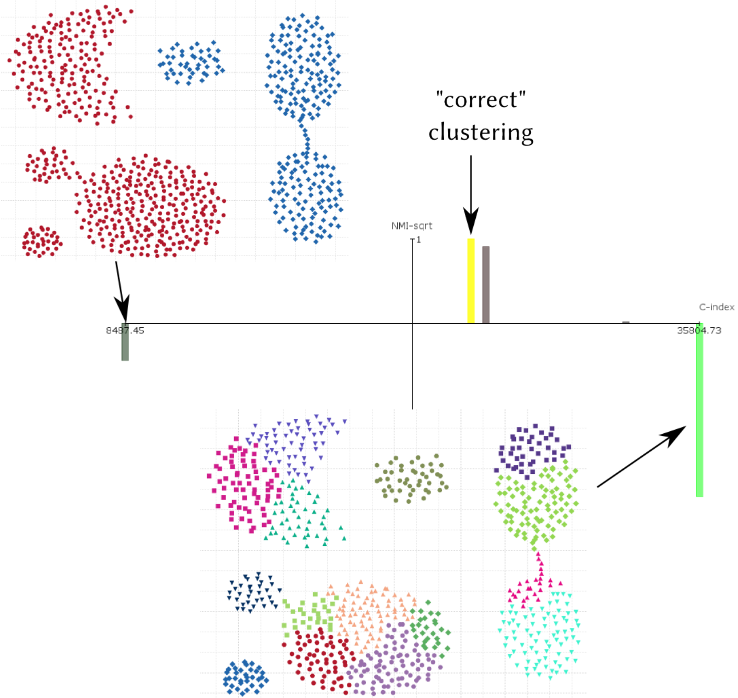
- aggregation dataset – 7 clusters

Visualization of objectives



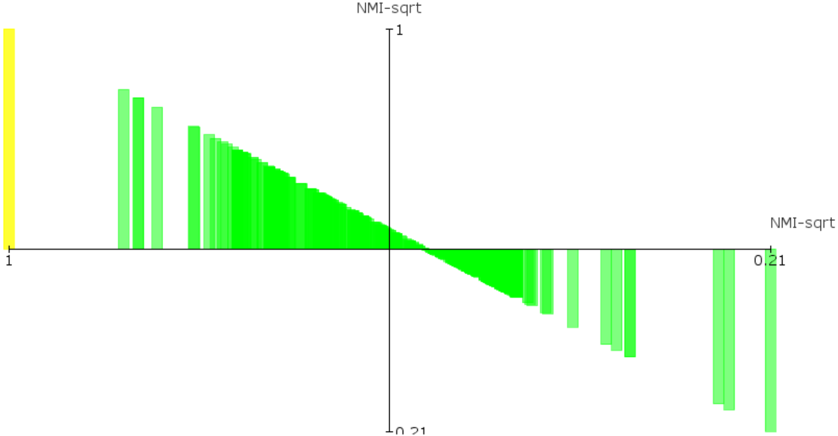


- over-optimized clustering (highest C-index)

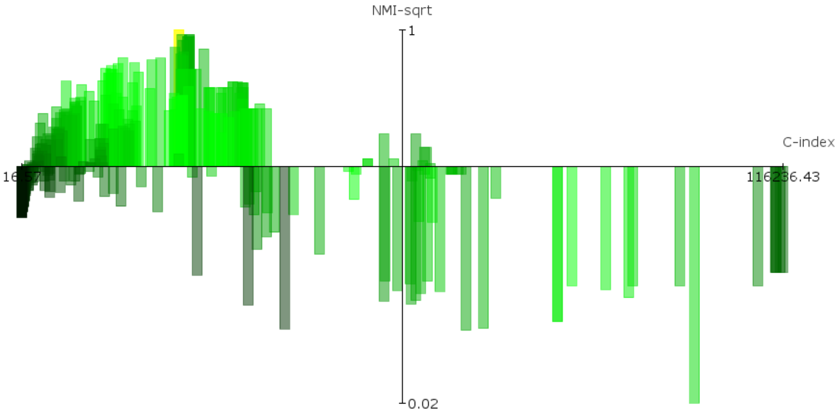


"correct"
clustering
NMI-sqrt
1
C-index
8487.45
35804.73

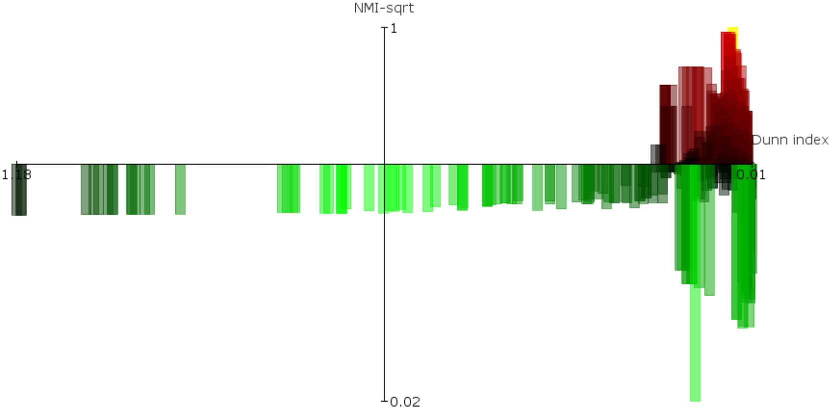
Ideal objective



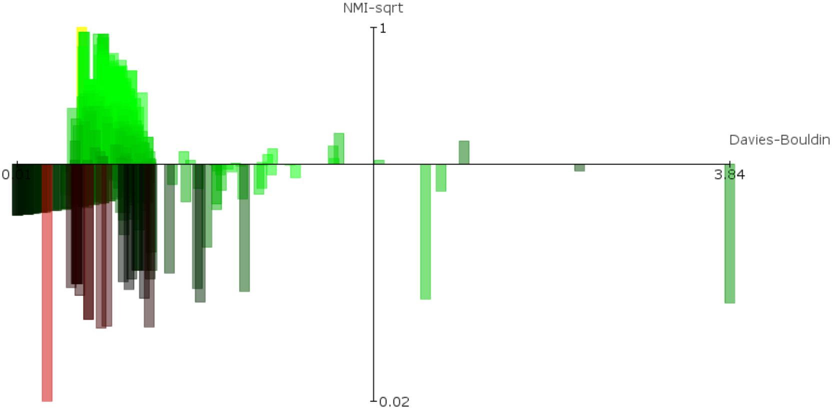
C-index



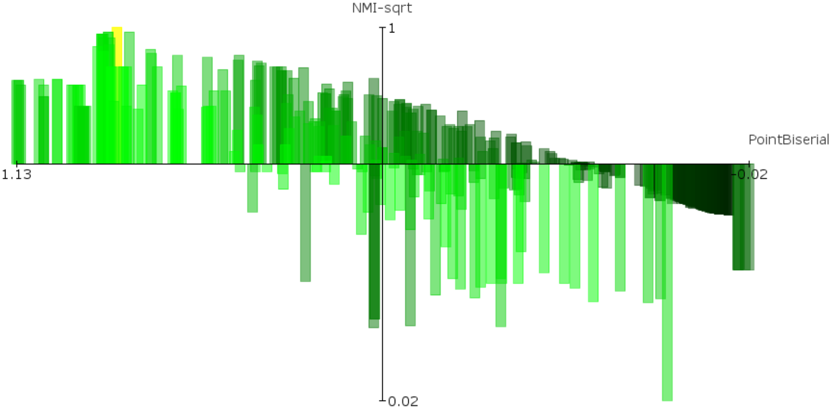
Dunn



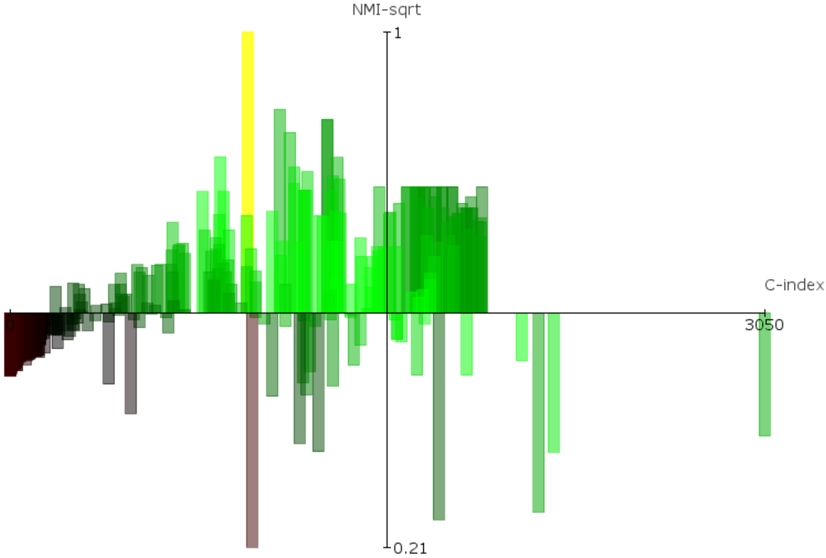
Davies-Bouldin



Point-Bi serial

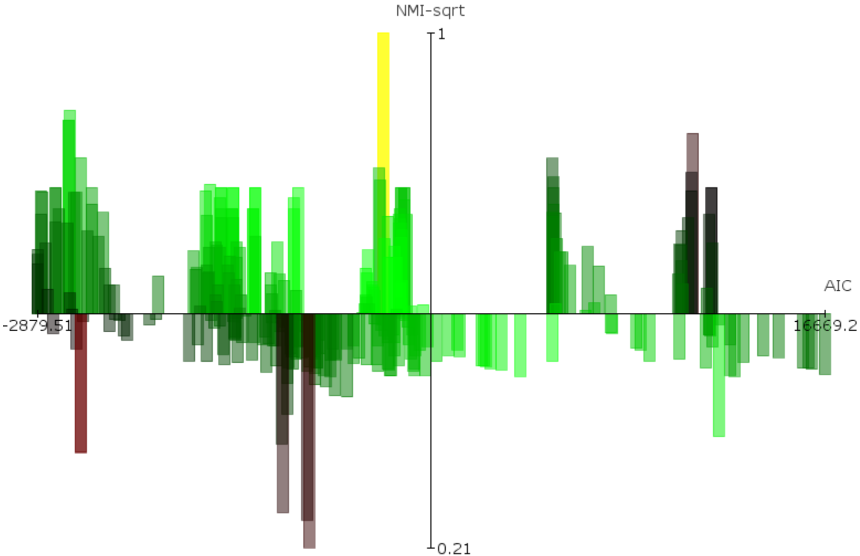


C-index (Iris datset)



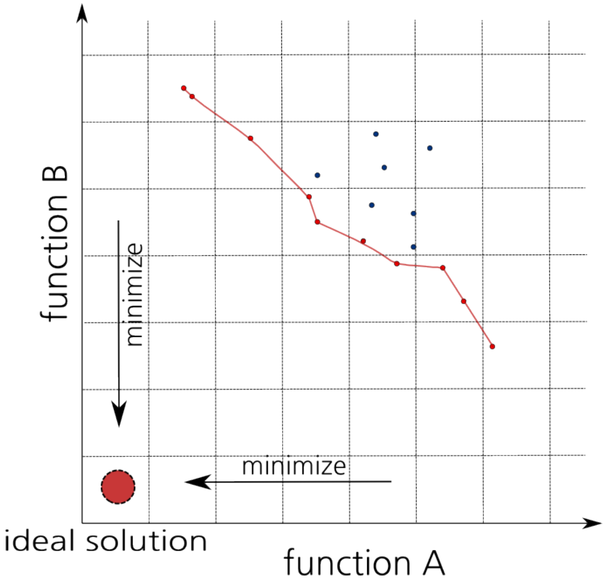
- correlation -0.81

AIC (Iris datset)

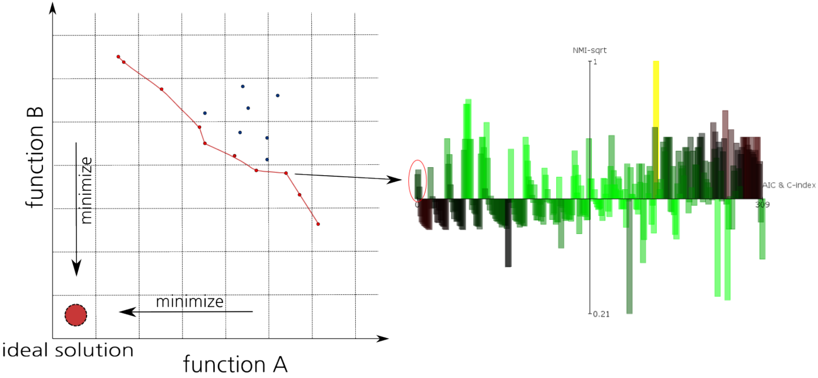


- correlation = 0.13

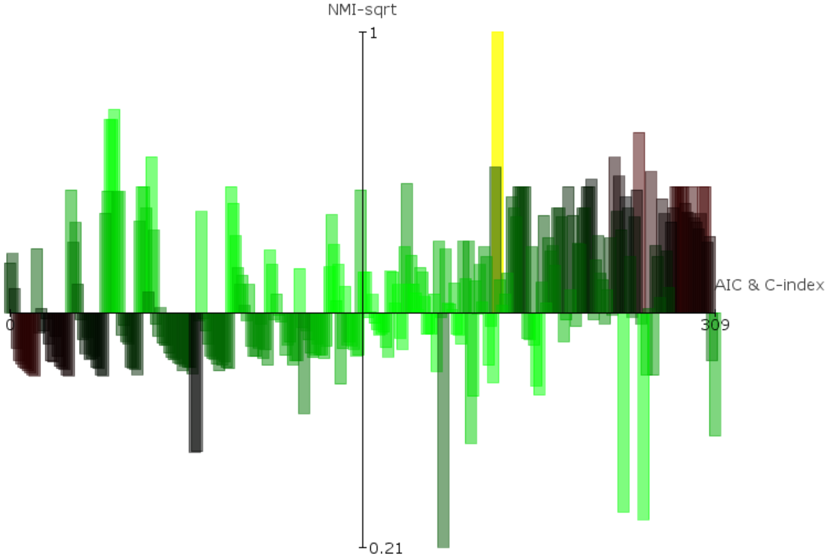
fast non-dominated sort



Pareto front projection

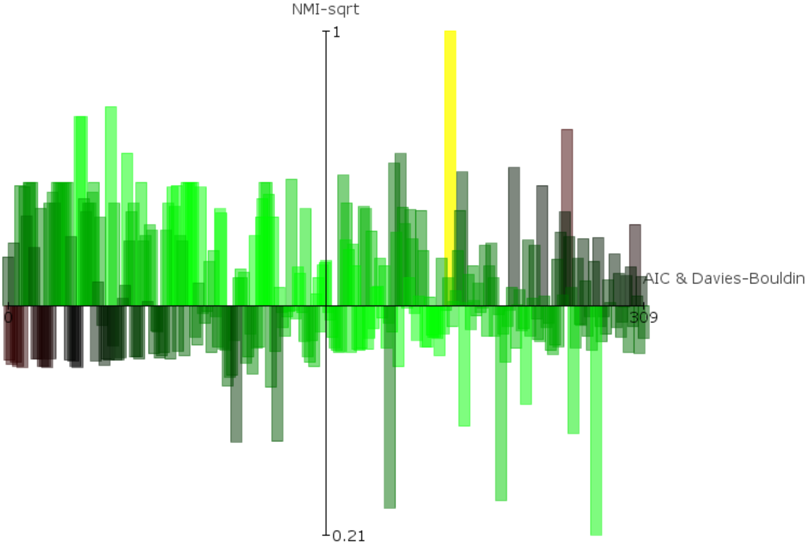


AIC & C-index (Iris datset)



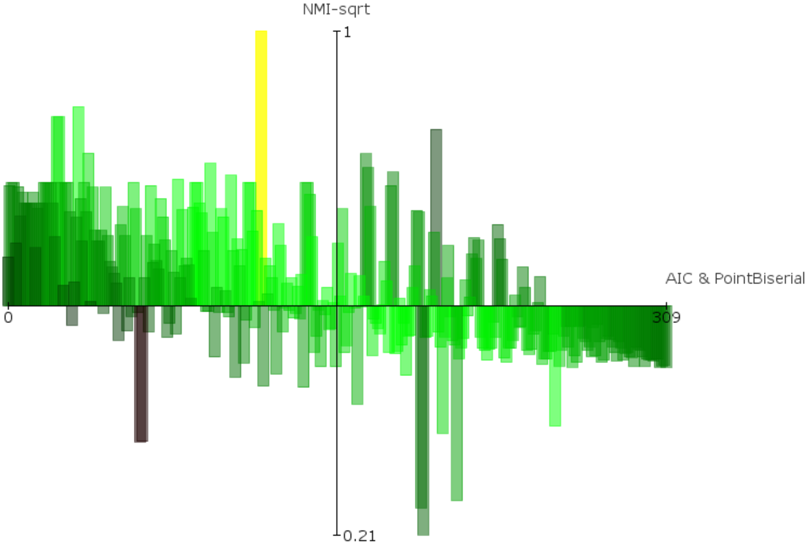
- correlation $= -0.47$

AIC & Davies-Bouldin (Iris datset)



- correlation = 0.12

AIC & Point BiSerial (Iris datset)



- correlation = 0.62

Conclusion

- after 50 years of research we have clustering objective that **actually works**
- there are combinations of objectives that work in many cases
- combining AIC (or BIC) with other objectives improves quality of the result

Questions?

Thank you for your attention

tomas.barton@fit.cvut.cz