A Comparison of Resampling Methods for Clustering Ensembles

Behrouz Minaei, Alexander Topchy and William Punch

Department of Computer Science and Engineering

MICHIGAN STATE UNIVERSITY

MLMTA 2004, Las Vegas, June 22th 2004

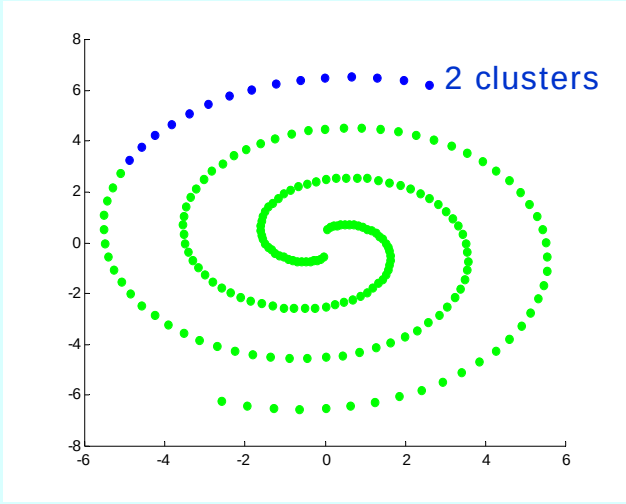
Outline

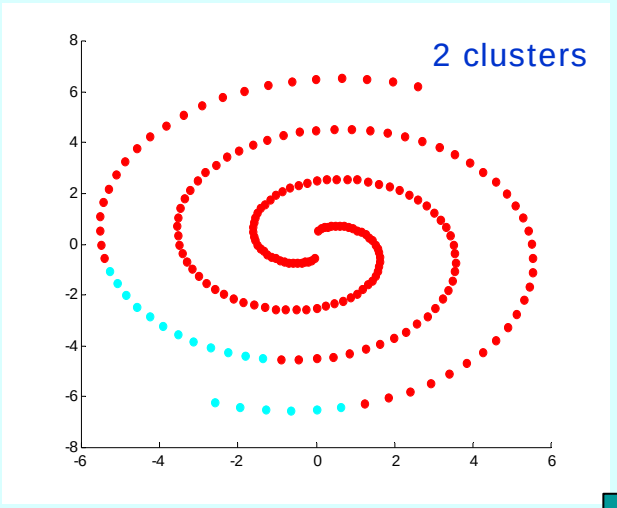
- Clustering Ensemble
  - How to generate different partitions?
  - How to combine multiple partitions?
- Resampling Methods
  - Bootstrap vs. Subsampling
- Experimental study
  - Methods
  - Results
  - Conclusion

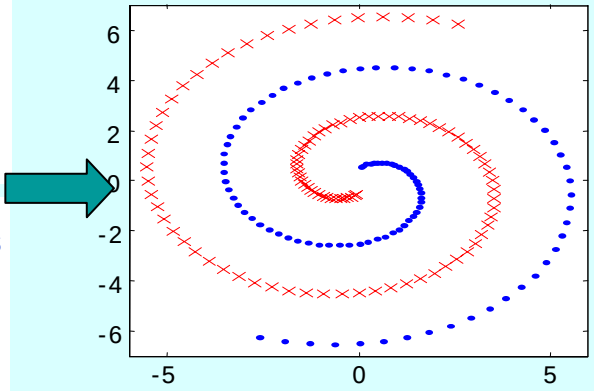
Ensemble Benefits

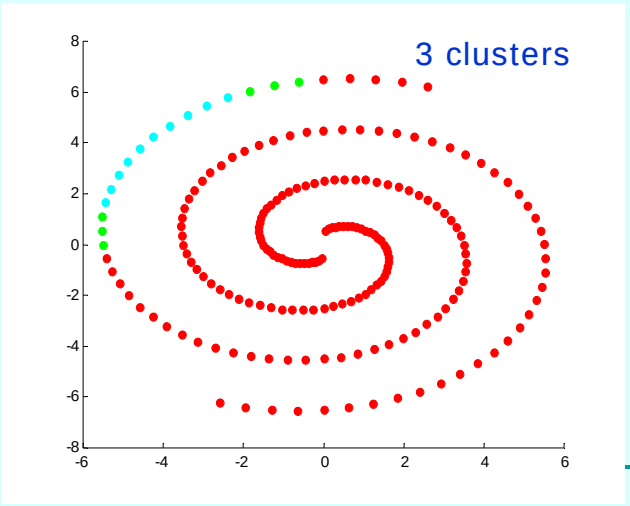
- Combinations of classifiers proved to be very effective in supervised learning framework, e.g. bagging and boosting algorithms
- Distributed data mining requires efficient algorithms capable to integrate the solutions obtained from multiple sources of data and features
- Ensembles of clusterings can provide novel, robust, and stable solutions

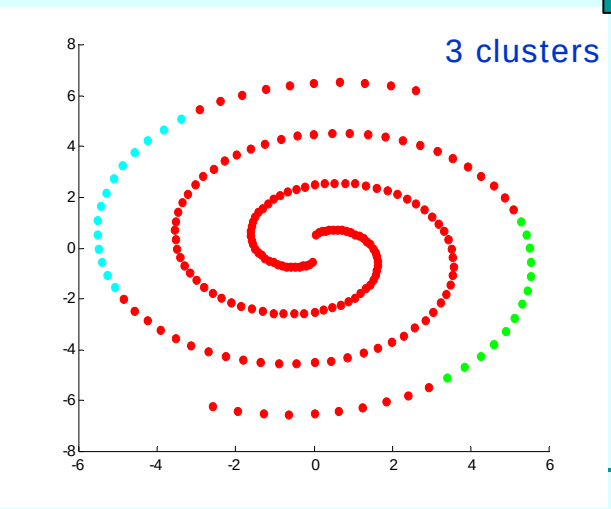
Is Meaningful Clustering Combination Possible?











"Combination" of 4 different partitions can lead to true clusters!

Pattern Matrix, Distance matrix

	Features					
X_1	x_{11}	x_{12}	…	x_{1j}	…	x_{1d}
X_2	x_{21}	x_{22}	…	x_{2j}	…	x_{2d}
…	…	…	…	…	…	…
X_i	x_{i1}	x_{i2}	…	x_{ij}	…	x_{id}
…	…	…	…	…	…	…
X_N	x_{N1}	x_{N2}	…	x_{Nj}	…	x_{Nd}

	x_1	x_2	…	x_j	…	x_N
x_1	d_{11}	d_{12}	…	d_{1j}	…	d_{1N}
x_2	d_{21}	d_{22}	…	d_{2j}	…	d_{2N}
…	…	…	…	…	…	…
x_i	d_{i1}	d_{i2}	…	d_{ij}	…	d_{iN}
…	…	…	…	…	…	…
x_N	d_{N1}	d_{N2}	…	d_{Nj}	…	d_{NN}

Representation of Multiple Partitions

- Combination of partitions can be viewed as another clustering problem, where each P_i represents a new feature with categorical values
- Cluster membership of a pattern in different partitions is regarded as a new feature vector
- Combining the partitions is equivalent to clustering these tuples

object s	P_1	P_2	P_3	P_4
x_1	1	A	α	Z
X_2	1	A	β	Y
X_3	3	D	β	?
X_4	2	D	α	Y
X_5	2	B	γ	Z
X_6	3	C	?	Z
X_7	3	C	γ	?

7 objects clustered by 4 algorithms

Re-labeling and Voting

	C-1	C-2	C-3	C-4
X1	1	A	α	Z
X2	1	A	β	Y
X3	3	B	β	?
X4	2	C	α	Y
X5	2	B	γ	Z
X6	3	C	?	Z
X7	3	B	γ	?

	C-1	C-2	C-3	C-4	FC
X1	1	1	1	2	1
X2	1	1	2	1	1
X3	3	3	2	?	3
X4	2	2	1	1	?
X5	2	3	2	2	2
X6	3	2	?	2	2
X7	3	3	2	?	3

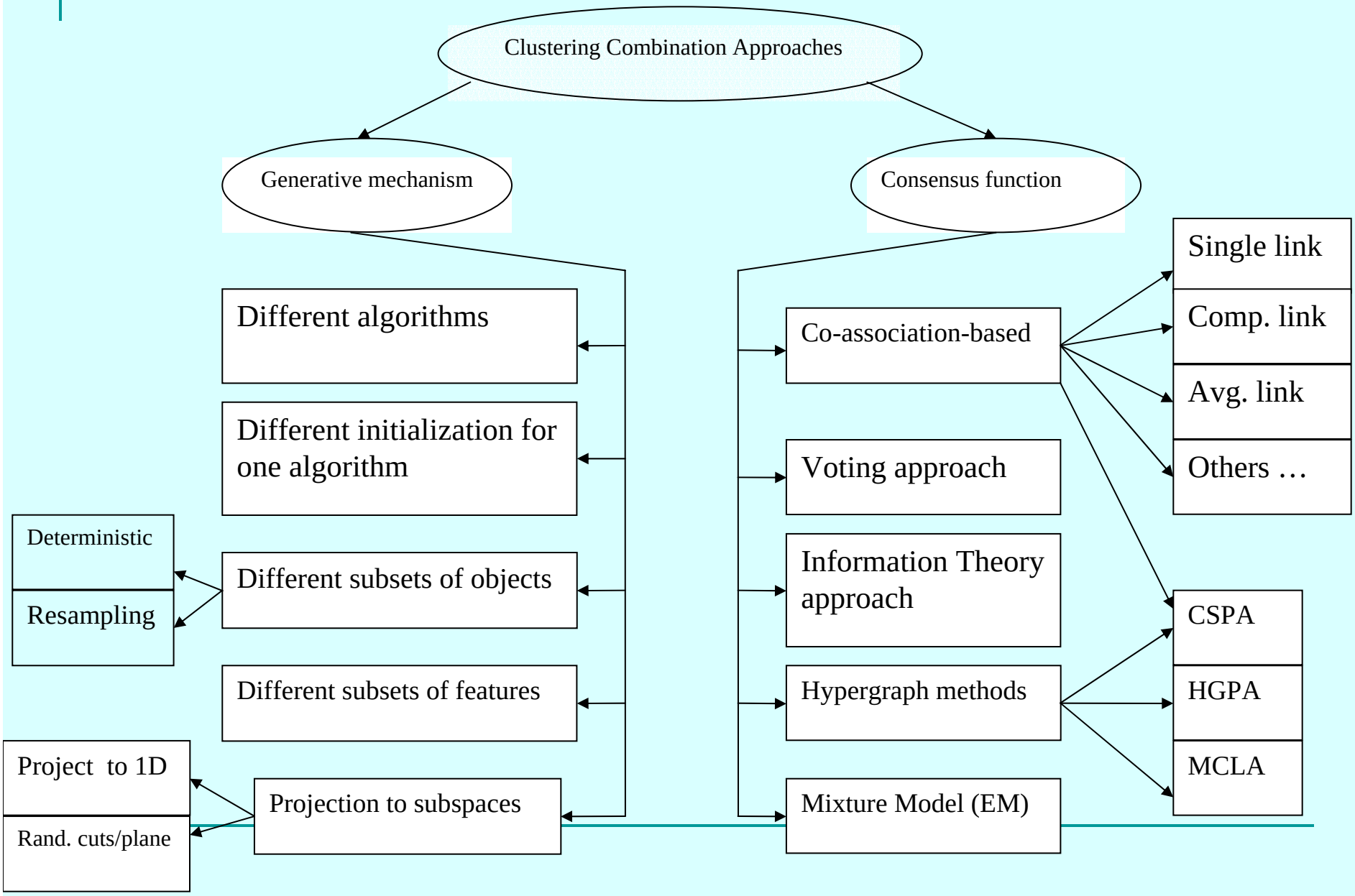
Co-association As Consensus Function

- Similarity between objects can be estimated by the number of clusters shared by two objects in all the partitions of an ensemble
- This similarity definition expresses the strength of co-association of n objects by an n x n matrix

$$C_{ij} = C(x_i, x_j) = \frac{1}{N} \sum_{k=1}^{N} I(\pi_k(x_i) = \pi_k(x_j))$$

  - x_i: the *i*-th pattern; $\pi_k(x_i)$: cluster label of xi in the *k*-th partition; I(): Indicator function; *N* = no. of different partitions
- This consensus function eliminates the need for solving the label correspondence problem

Taxonomy of Clustering Combination Approaches



Resampling Methods

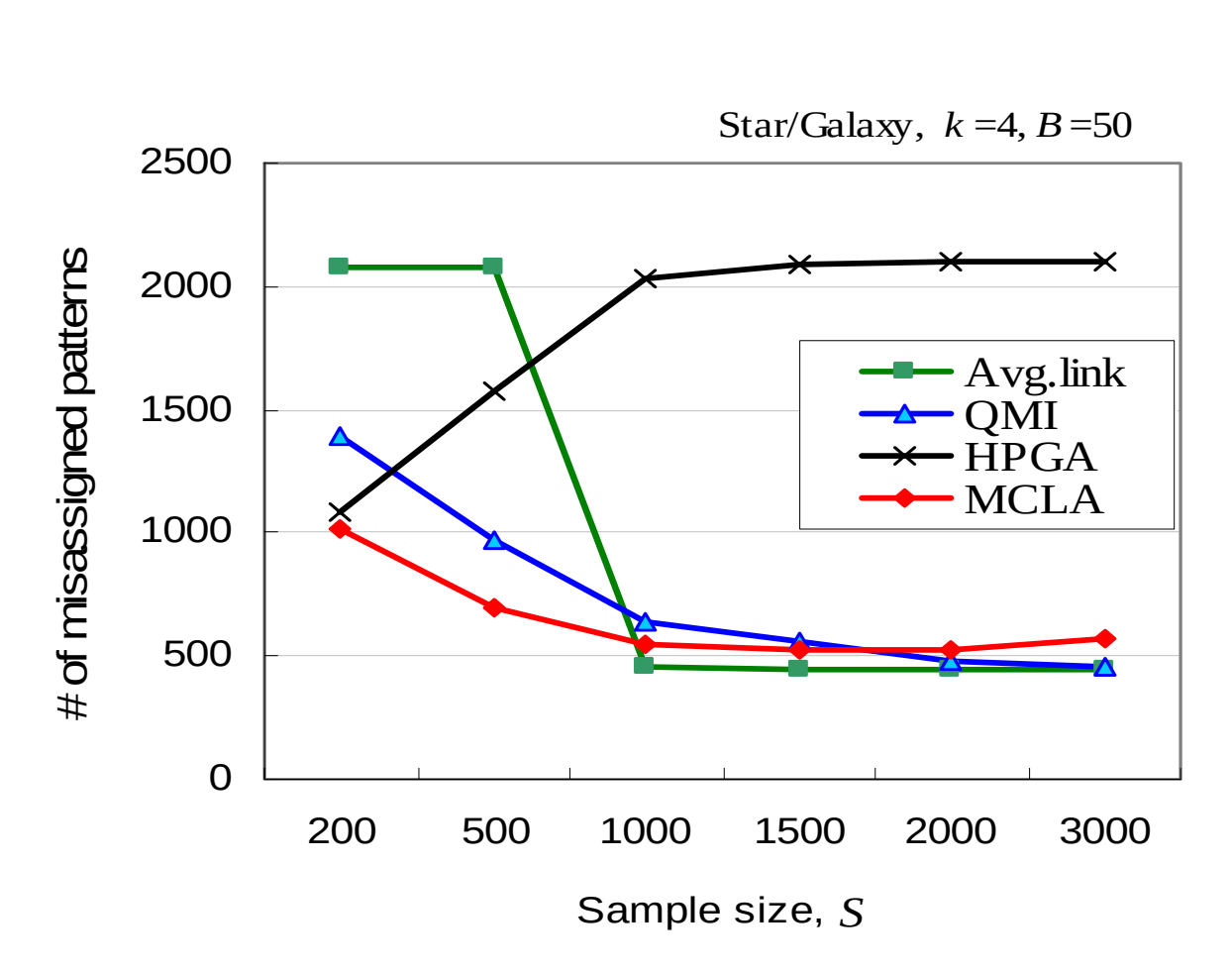
- Bootstrapping (Sampling with replacement)
  - Create an artificial list by randomly drawing *N* elements from that list. *Some elements will be picked more than once.*
  - Statistically on average 37% of elements are repeated

- Subsampling (Sampling without replacement)
  - Control over the size of subsample

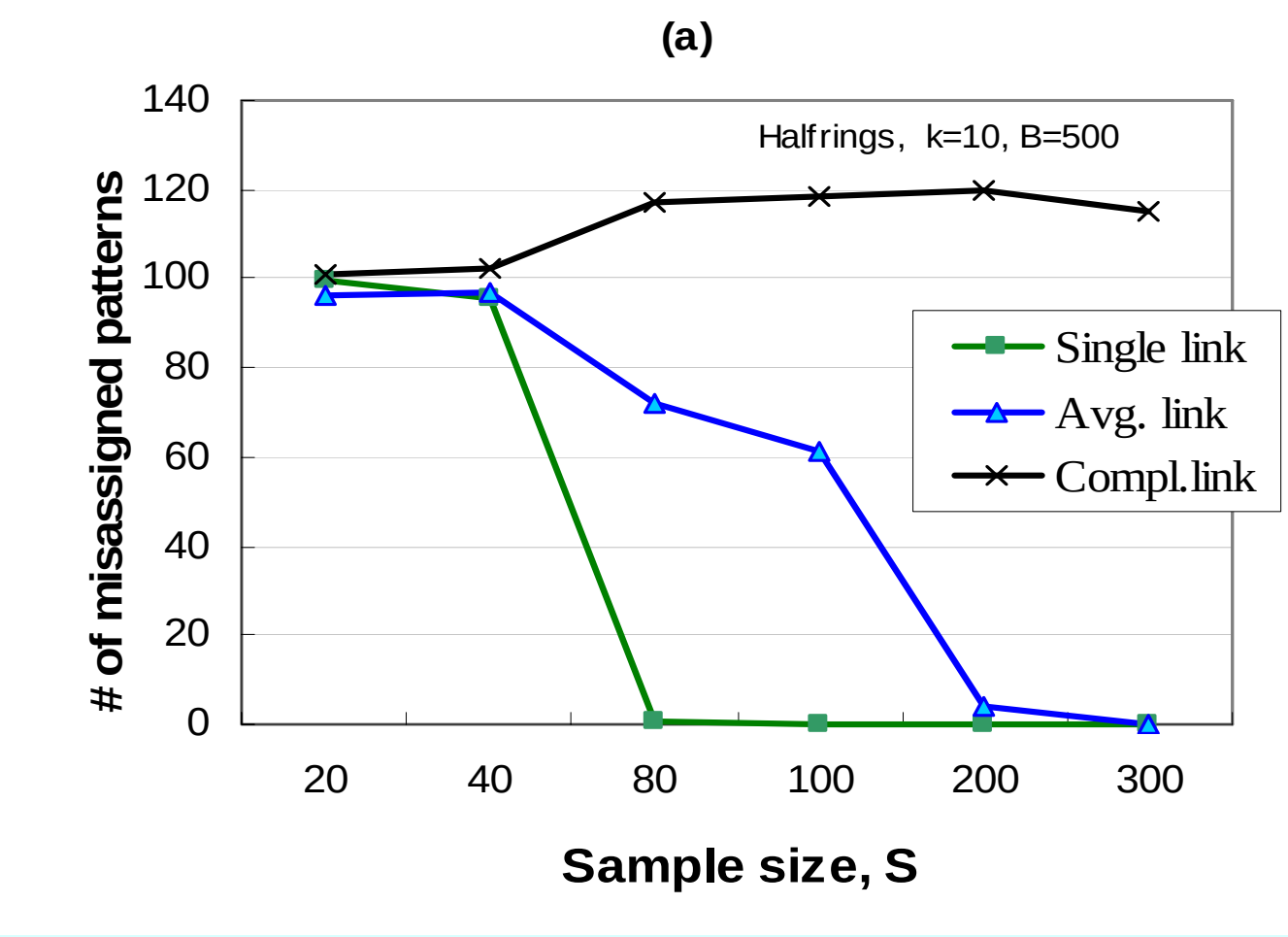
Experiment: Data sets

	Number of Classes	Number of Features	Total no of patterns	Patterns per class
Halfrings	2	2	400	100-300
2-spirals	2	2	200	100-100
Star/Galaxy	2	14	4192	2082-2110
Wine	3	13	178	59-71-48
LON	2	6	227	64-163
Iris	3	4	150	50-50-50

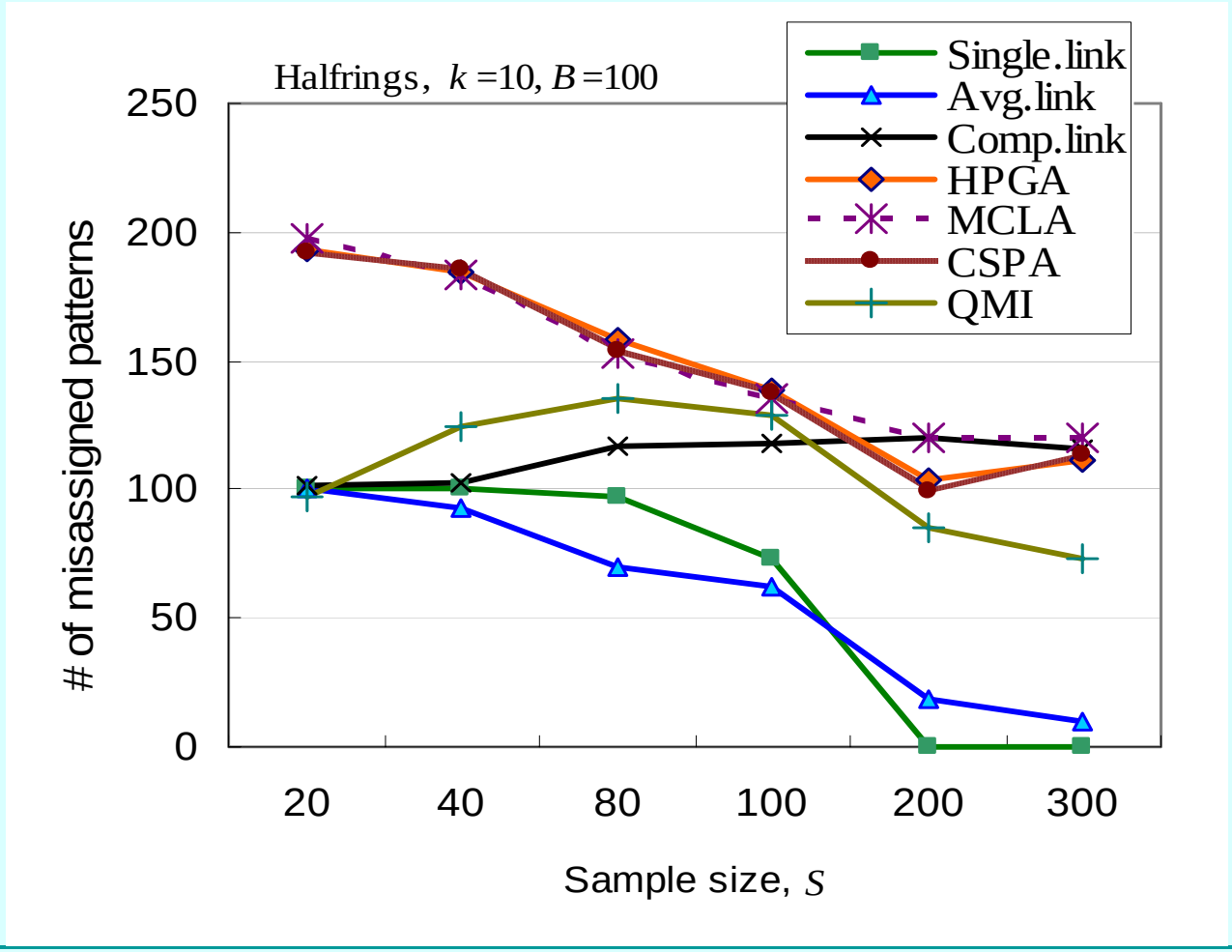
Subsampling results on Star/Galaxy



Subsampling results on Halfrings



Subsampling on Halfrings

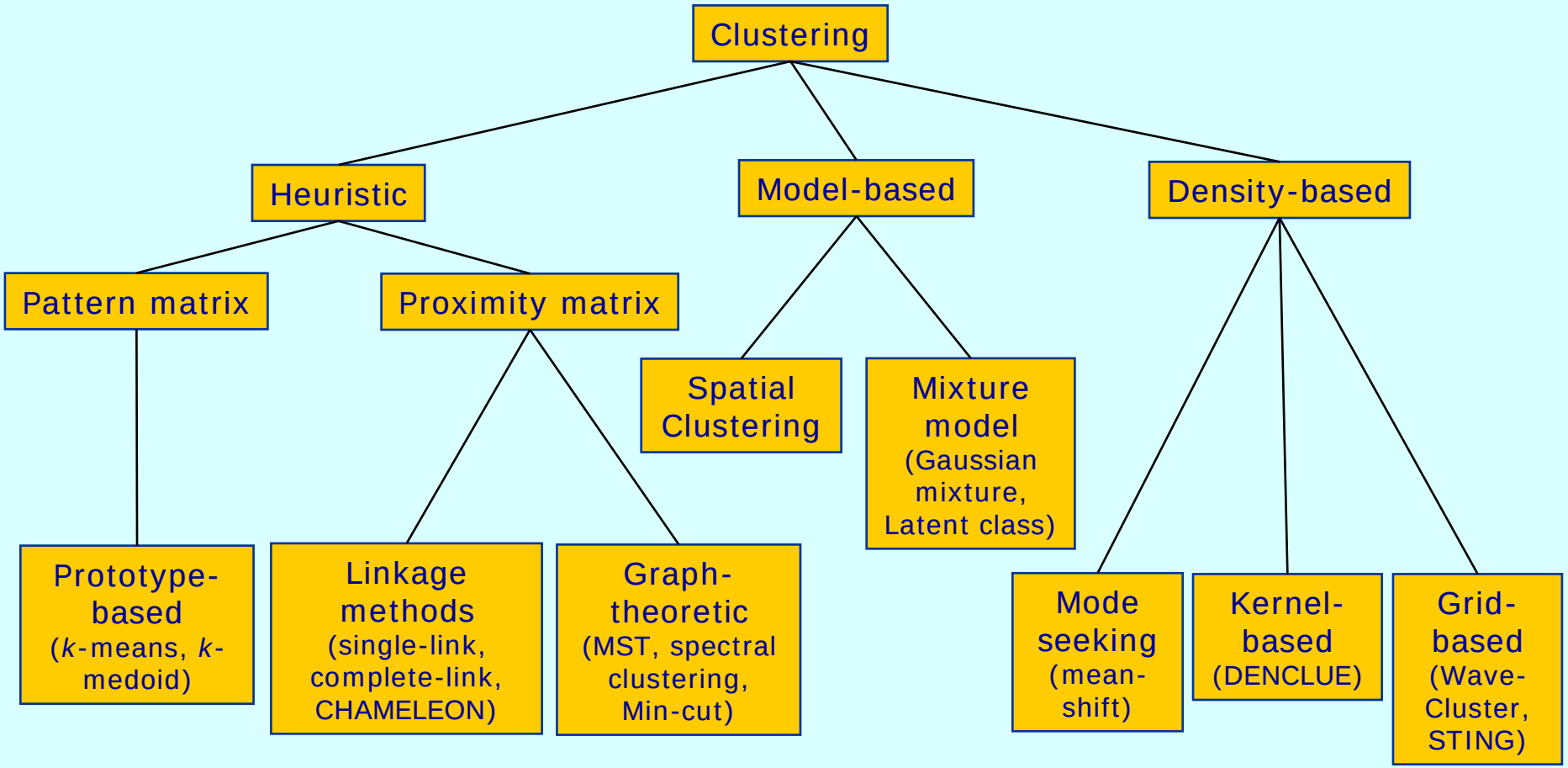


Error Rate for Individual Clustering

Data set	*k*-means	Single Link	Complete Link	Average Link
Halfrings	25%	24.3%	14%	5.3%
2 Spiral	43.5%	0%	48%	48%
Iris	15.1%	32%	16%	9.3%
Wine	30.2%	56.7%	32.6%	42%
LON	27%	27.3%	25.6%	27.3%
Star/Galaxy	21%	49.7%	44.1%	49.7%

Subsampling results	**# of clusters in partition, *k***	**# of Partitions *B***	**Sample Size *S***	**% of entire data**
Halfrings	15	50	200	50%
	15	**500**	**80**	**20%**
	20	50	200	50%
	20	500	100	25%
2 Spiral	20	500	150	75%
	20	1000	100	50%
Iris	3	500	50	33%
	4	**500**	**20**	**13%**
	5	100	50	33%
	20	50	50	33%
Wine	4	100	100	56%
	5	100	50	28%
	10	**50**	**50**	**28%**
LON	4	50	150	66%
	4	500	100	44%
Galaxy/ Star	3	50	1500	36%
	4	100	1000	24%
	10	**100**	**500**	**12%**
	10	**100**	**200**	**5%**

Taxonomy of Clustering Methods



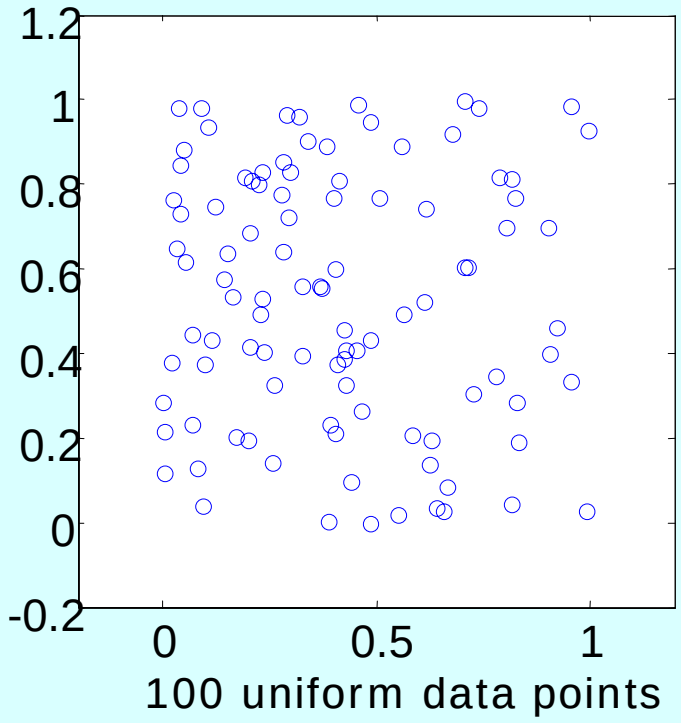
- Clustering methods call for particular clustering algorithms
- Hundreds of clustering algorithms proposed

Clustering Combination

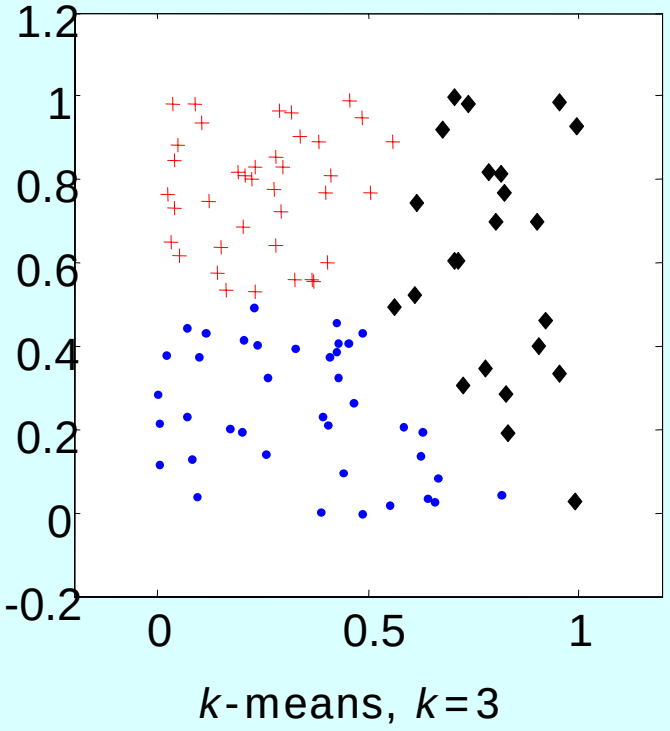
- Observations
  - Each clustering algorithm addresses differently the issues of number of clusters, and the structure imposed on the data, and produces different data partitions
  - A single clustering algorithm can produce distinct results on the same data set depending on parameter values
- Instead of a single clustering algorithm, use multiple algorithms
- Success of the ensemble of classifiers in supervised learning (bagging and boosting) is the motivation

Are There Any Clusters in Data?

- Most clustering algorithms find clusters, even if the data is uniform!



100 uniform data points



k-means, *k*=3

Clustering in new space

- Essentially consensus must be found by clustering objects in new space – space of cluster labels
- In fact, features of new space are extracted from “old” features by different clustering algorithms
- New space can possibly be extended by adding original “old” features
- Problem becomes very rich with infinite ways to solve

How to Generate the Ensemble?

- Apply different clustering algorithms
  - Single-link, EM, complete-link, spectral clustering, *k*-means
- Use random initializations of the same algorithm
  - Output of *k*-means and EM depends on initial cluster centers
- Use different parameter settings of the same algorithm
  - number of clusters, “width” parameter in spectral clustering, thresholds in linkage methods
- Use different feature subsets or data projections
- Re-sample data with or without replacement

Consensus Functions

- *Co-association matrix* (Fred & Jain 2002)
  - Similarity between 2 patterns estimated by counting the number of shared clusters
  - Single-link with max life-time for finding the consensus partition
- Hyper-graph methods (Strehl & Ghosh 2002)
  - Clusters in different partitions represented by hyper-edges
  - Consensus partition found by a *k*-way min-cut of the hyper-graph
- Re-labeling and voting (Fridlyand & Dudoit 2001)
  - If the label correspondence problem is solved for the given partitions, a simple voting procedure can be used to assign objects in clusters
- Mutual Information (Topchy, Jain & Punch 2003)
  - Maximize the mutual information between the individual partitions and the target consensus partition
- Finite mixture model (Topchy, Jain & Punch 2003)
  - Maximum likelihood solution to latent class analysis problem in the space of cluster labels via EM

Consensus functions: **Co-association approach**

(Fred 2001, Fred & Jain 2002)

- Similarity between objects can be estimated by the number of clusters shared by two objects in all the partitions of an ensemble.
- This similarity definition expresses the strength of co-association of objects by a matrix containing the values:

$$S_{ij} = S(x_i, x_j) = \frac{1}{H} \sum_{k=1}^{H} \delta(\pi(x_i), \pi(x_j))$$

- Thus, one can use numerous similarity-based clustering algorithms by applying them to the matrix of co-association values.

Consensus functions: **Hypergraph approach**

(Strehl & Ghosh 2002)

- All the clusters in the ensemble partitions can be represented as hyperedges on a graph with *N* vertices.
- Each hyperedge describes a set of objects belonging to the same clusters.
- A consensus function is formulated as a solution to *k*-way min-cut hypergraph partitioning problem. Each connected component after the cut corresponds to a cluster in the consensus partition.
- Hypergraph partitioning problem is NP-hard, but very efficient heuristics are developed for its solution with complexity proportional to the number of hyperedges *O*(|E|).

Consensus functions: **Mutual information** (Topchy et al. 2003a)

- Mutual information (MI) between the empirical probability distribution of labels in the consensus partition and the labels in the ensemble must be maximized. Under the assumption of independence of partitions, mutual information can be written as sum of pair-wise MIs between target and given partitions.

$$\sigma_{best} = \arg\max_{\sigma} I(\sigma, \Pi)$$

- An elegant solution can be obtained from a generalized definition of MI. Quadratic MI information can be effectively maximized by the *k*-means algorithm in the space of specially transformed cluster labels of given ensemble. Computational complexity of the algorithm is *O*(*kNH*)

$$U(\sigma, \pi_i) = \sum_{r=1}^{K} p(C_r) \sum_{j=1}^{K(i)} p(L_j^i \mid C_r)^2 - \sum_{j=1}^{K(i)} p(L_j^i)^2$$

Consensus functions: **Finite Mixture model** (Topchy et al. 2003b)

- The main assumption is that the labels $\mathbf{y}_i$ are modeled as random variables drawn from a probability distribution described as a mixture of multivariate multinomial component densities:

$$P(\mathbf{y}_i \mid \Theta) = \sum_{m=1}^{M} \alpha_m P_m(\mathbf{y}_i \mid \boldsymbol{\theta}_m)$$

$$P_m(\mathbf{y}_i \mid \boldsymbol{\theta}_m) = \prod_{j=1}^{H} P_m^{(j)}(y_{ij} \mid \boldsymbol{\theta}_m^{(j)})$$

$$P_m^{(j)}(y \mid \boldsymbol{\theta}_m^{(j)}) = \prod_{k=1}^{K(j)} \vartheta_{jm}(k)^{\delta(y,k)}$$

Consensus functions: **Finite Mixture model**

The objective of consensus clustering is formulated as a maximum likelihood estimation problem. To find the best fitting mixture density for a given data $\boldsymbol{Y}$ we must maximize the likelihood function with respect to the unknown parameters Θ:

$$\log L(\Theta \mid \mathbf{Y}) = \log \prod_{i=1}^{N} P(\mathbf{y}_i \mid \Theta) = \sum_{i=1}^{N} \log \sum_{m=1}^{M} \alpha_m P_m(\mathbf{y}_i \mid \boldsymbol{\theta}_m)$$

$$\Theta^* = \arg\max_{\Theta} \log L(\Theta \mid \mathbf{Y})$$

EM algorithm is used to solve this maximum likelihood problem

Consensus functions: **Re-labeling and Voting** (Fridlyand & Dudoit 2001)

- If a label correspondence problem is solved for all given partitions, then a simple voting procedure can be used to assign objects in clusters. However, label correspondence is exactly what makes unsupervised combination difficult.
- A heuristic approximation to consistent labeling is possible. All the partitions in the ensemble can be re-labeled according to their best agreement with some chosen reference partition.
- The reference partition can be taken as one from the ensemble, or from a new clustering of the dataset.

Subsampling results on Galaxy/Star