



**OPPA European Social Fund
Prague & EU: We invest in your future.**

Cluster analysis – advanced and special algorithms

Jiří Kléma

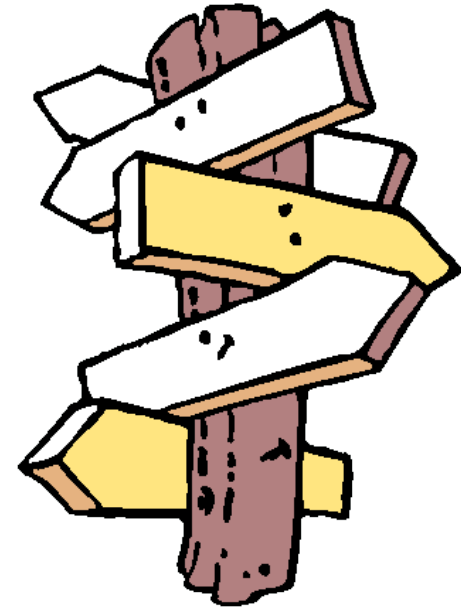
Department of Cybernetics,
Czech Technical University in Prague



<http://ida.felk.cvut.cz>

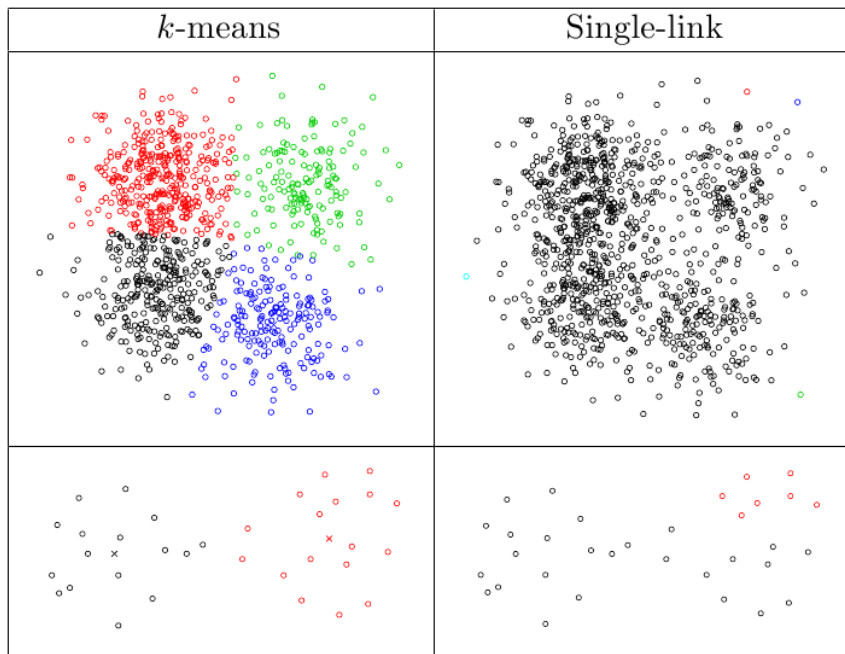
Outline

- robustness of the introduced methods – examples,
- k-means and hierarchical agglomerative clustering – complexity,
- clustering quality – evaluation
 - internal versus external partitioning evaluation,
- clustering definition?
- an advanced method
 - spectral clustering,
- special methods
 - conceptual clustering,
 - bi-clustering,
 - semi-supervised clustering.

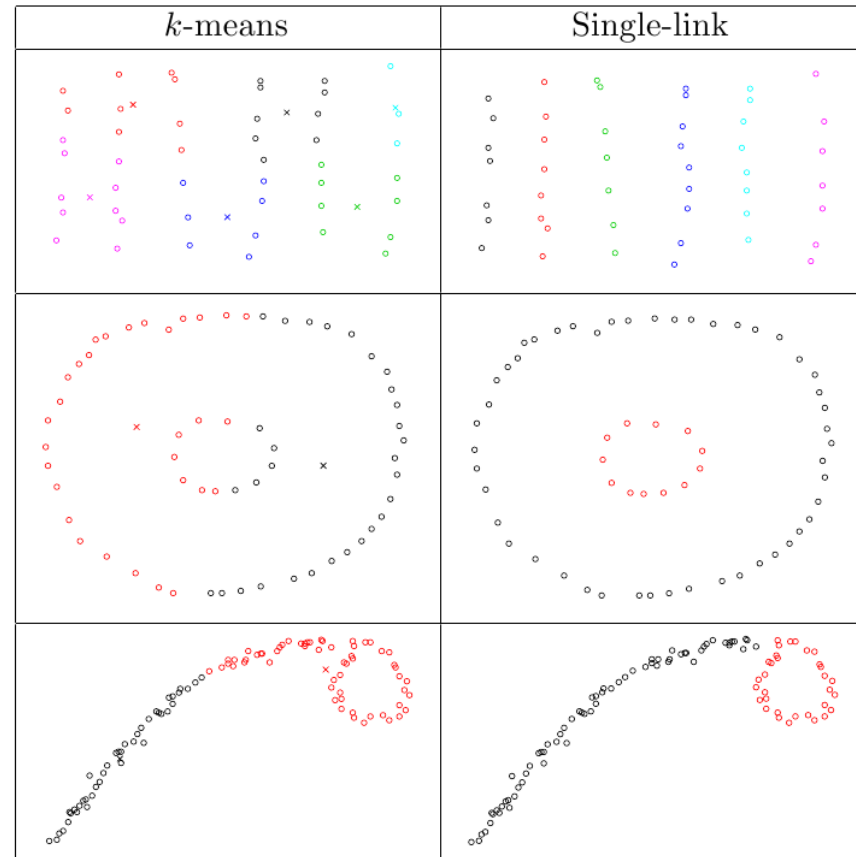


Comparison: k-means and hierarchical single-link

- single linkage tends to generate longer non-compact clusters,
- k-means makes compact clusters, complete linkage is outlier sensitive,



k-means intuitively correct



single linkage intuitively correct

Complexity – comparison

- assumed: $d(x_i, x_j) \in \mathcal{O}(n)$,
- k-means algorithm
 - assign instances into clusters: $\mathcal{O}(km)$ distance computations $\rightarrow f_E \in \mathcal{O}(knm)$,
 - modify centroids: $f_M \in \mathcal{O}(nm)$ (each instance used exactly once in one of the centroid),
 - unknown iteration number before stop: i (estimates vary from the constant with m up to $\mathcal{O}(m^{kn})$),
 - summary: $f = i(f_E + f_M) \in \mathcal{O}(iknm)$,
- hierarchical agglomerative clustering (single link)
 - initialize
 - * compute distances among all instance pairs $f_I \in \mathcal{O}(m^2n)$,
 - * *next-best-merge* array – the nearest neighbor with its distance for each cluster (complexity hidden in the previous line),
 - $m - 1$ iterations to complete the dendrogram
 - * find the smallest distance in the next-best-merge array $f_{E_0} \in \mathcal{O}(m)$,
 - * adjust the distance matrix $f_{E_1} \in \mathcal{O}(m)$,
 - * adjust the next-best-merge array $f_{E_2} \in \mathcal{O}(m)$,
 - summary: $f = f_I + (m - 1)(f_{E_0} + f_{E_1} + f_{E_2}) \in \mathcal{O}(m^2n)$.

Clustering quality – evaluation

- **internal:** quantifies three partitioning characteristics

- homogeneity – are instances within clusters similar?

$$hom = \frac{1}{m} \sum_{i=1}^k \sum_{x_j \in C_i} d(x_j, \mu_i),$$

- separability – are instances in different clusters dissimilar?

$$sep = \frac{1}{k(k-1)} \sum_{i=1}^k \sum_{j=1, j \neq i}^k \exp\left(-\frac{d^2(\mu_i, \mu_j)}{2\sigma^2}\right),$$

- stability – how many relations brakes by adding noise or random instance sampling?

- internal evaluation criteria in the clustering algorithms

- intra-cluster variability: (see homogeneity for k-means),
- model likelihood: (for probabilistic models, see EM GMM),
- the size of the balanced cut of similarity graph: (see spectral clustering later)

$$\frac{1}{2} \sum_{ij} s(x_i, x_j) (1 - \delta(C_i - C_j)) \left(\frac{1}{|C_i|} + \frac{1}{|C_j|} \right),$$

- clustering is subjective, the “objective” internal measure can be improper.

Clustering quality – evaluation

- **external:** match the partitioning Ω with a known annotation $G = \{G_1, \dots, G_c\}$ (gold standard),

- basic external evaluation criteria

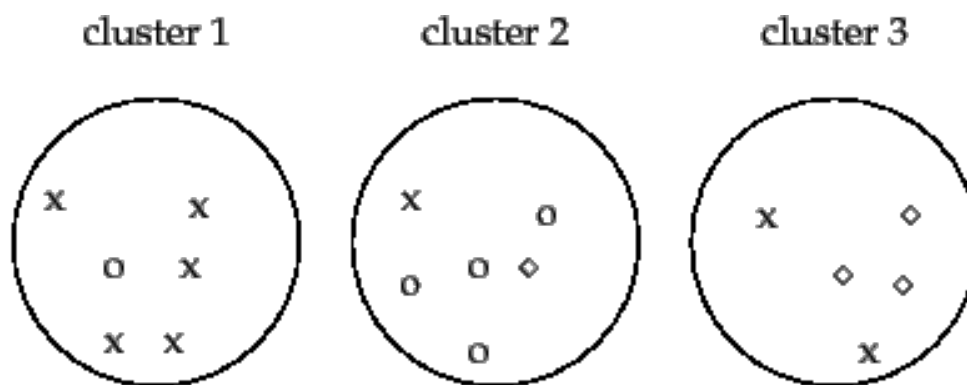
– purity

- * the total major instance class ratio across clusters
(each cluster has a major class, the higher its ratio in the cluster the better),

$$purity(\Omega, G) = \frac{1}{m} \sum_{i=1 \dots k} \max_{j=1 \dots c} |C_i \cap G_j|$$

- * disadvantage: cannot compare partitionings with different k ,

- * example (figure): $purity = \frac{5+4+3}{17} = 0.71$



Manning et al.: Introduction to Information Retrieval.
<http://nlp.stanford.edu/IR-book/html/htmledition/irbook.html>

Clustering quality – evaluation

- basic external evaluation criteria (continuation)

- normalized mutual information

- * based on information entropy,

$$NMI(\Omega, G) = \frac{2I(\Omega, G)}{H(\Omega)H(G)}$$

$$I(\Omega, G) = - \sum_{i=1\dots k} \sum_{j=1\dots c} P(C_i \cap G_j) \log_2 \left(\frac{P(C_i \cap G_j)}{P(C_i)P(G_j)} \right)$$

$$H(\Omega) = - \sum_{i=1\dots k} P(C_i) \log_2(P(C_i)), \quad H(G) = - \sum_{j=1\dots c} P(G_j) \log_2(P(G_j))$$

- * example (figure): $H(\Omega) = -2\frac{6}{17} \log_2\left(\frac{6}{17}\right) - \frac{5}{17} \log_2\left(\frac{5}{17}\right) = 1.58$, $H(G) = 1.52$

$$NMI = \frac{2 \times 0.96}{1.58 \times 1.52} = 0.8$$

- rand index

- * clustering is interpreted as a sequence of $\frac{m(m-1)}{2}$ decisions,

- * decision = the partitioning either puts a instance pair into the same cluster or not,

- * TP ... an instance pair in the same cluster in Ω and the same class in G ,

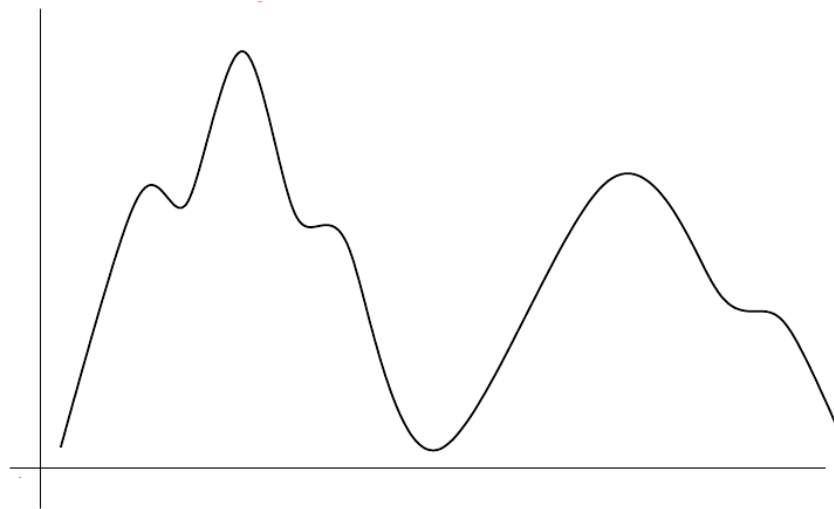
- * TN ... an instance pair in different clusters in Ω and different classes in G ,

- * $RI(\Omega, G) = \frac{TP+TN}{TP+TN+FP+FN} = \frac{2(TP+TN)}{m(m-1)}$

- * example (figure): $RI = \frac{2\left(\binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2}\right) + 5 \times 8 + 1 \times 7 + 4 \times 5 + 1 \times 2 + 1 \times 3}{17 \times 16} = 0.68$

Defining clustering by ...

- a quality function – see the previous slides
 - + can use standard optimization techniques,
 - heuristic choice of quality function, usually NP hard,
- high density areas
 - a cluster is a high density area, clusters are separated by low density areas,
 - + intuitively makes sense, density estimation is the well-known problem,
 - for non-trivial dimension even more demanding than clustering itself,



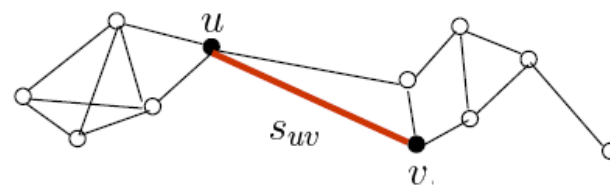
Defining clustering by ...

- a model-based approach
 - assumed that data were generated by a (probabilistic) model
 - the model implicitly defines clusters,
 - + model more than a partitioning = a clear interpretation,
 - + standard techniques such as ML, EM, Bayesian approaches available,
 - often too strong assumptions eventually unsatisfied, parameter estimation is not easy,
- an axiomatic view
 - clustering function from the distance matrix to partitioning defined indirectly by properties,
 - an axiomatic system example:
 - * scale invariance: distance scaling does not change the partitioning,
 - * richness: for any clustering there exists a distance matrix which induces it,
 - * consistency: shrink/expand distances inside/outside cluster → partition unchanged,
 - + an elegant way of definition,
 - seemingly harmless axiom sets contradictory, ad hoc choice, often not pract. helpful,
- information theory
 - clustering=lossy compression, defined by acceptable amount of loss or code length,
 - what is the “original information”? cannot be solved analytically.

Graph theory – basic terms

- vertex (object) similarity (affinity)

- $s_{uv} = \langle u, v \rangle$,



- vertex degree (volume), degree matrix

- $d_u = \sum_{v=1}^m s_{uv}$,

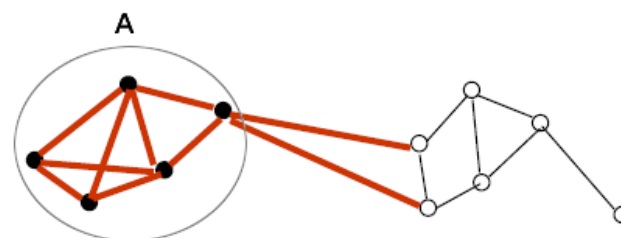
- $\mathcal{D} = \text{diag}(d_1, \dots, d_m)$,



- size and degree of a vertex set (cluster)

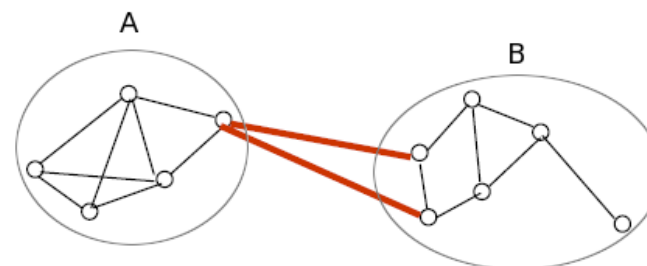
- $|A|$... the number of vertices in A ,

- $\text{vol}(A) = \sum_{u \in A} d_u$,



- an edge cut between two components

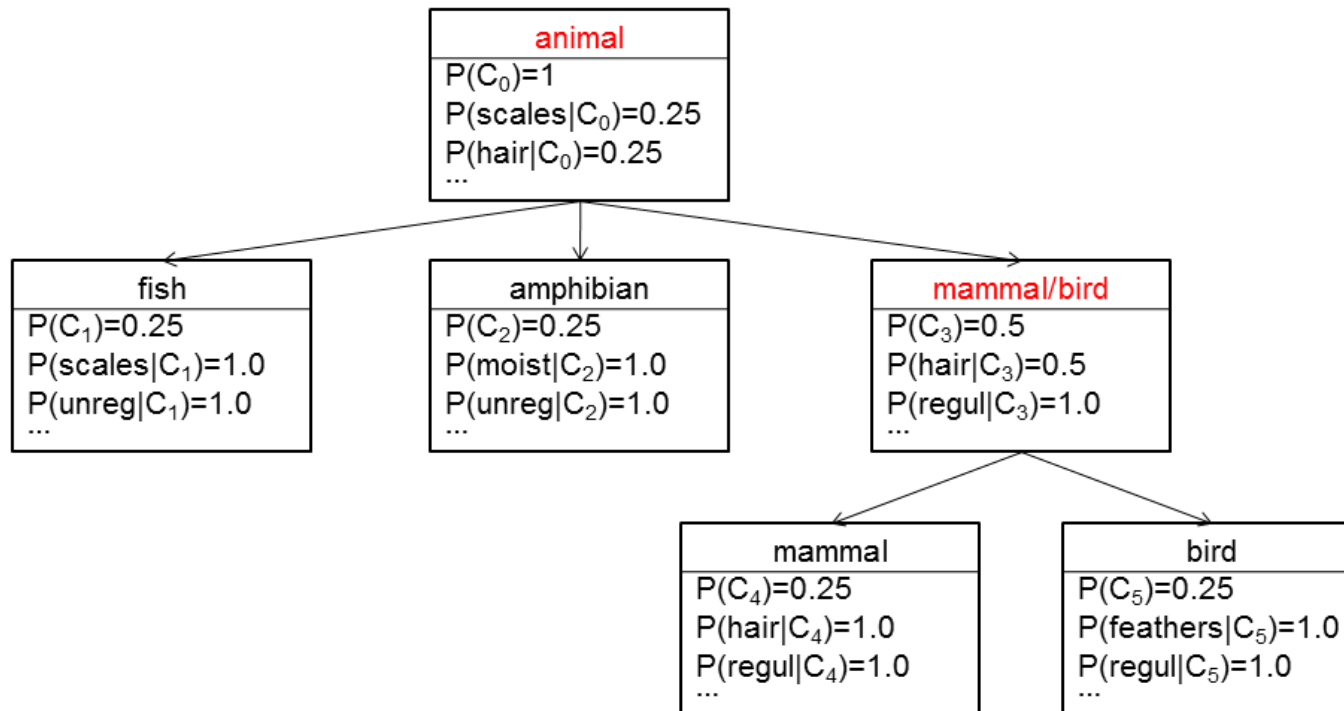
- $\text{cut}(A, B) = \sum_{u \in A} \sum_{v \in B} s_{uv}$.



Azran: A Tutorial on Spectral Clustering

Conceptual clustering: COBWEB (Fisher, 1987)

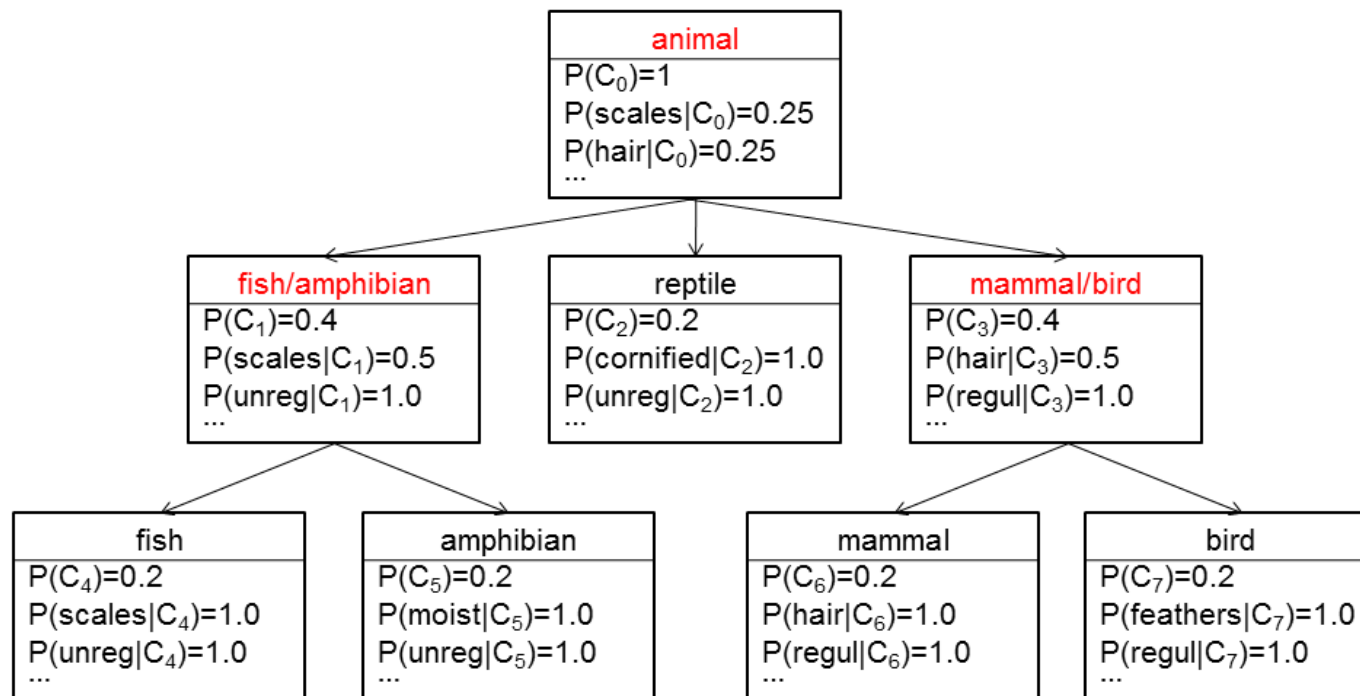
Name	BodyCover	HeartChamber	BodyTemp	Fertilization
fish	scales	two	unregulated	external
amphibian	moist_skin	three	unregulated	external
mammal	hair	four	regulated	internal
bird	feathers	four	regulated	internal
reptile	cornified_skin	imperfect_four	unregulated	internal



Fisher: Conceptual Acquisition Via Incremental Conceptual Clustering

Conceptual clustering: COBWEB (Fisher, 1987)

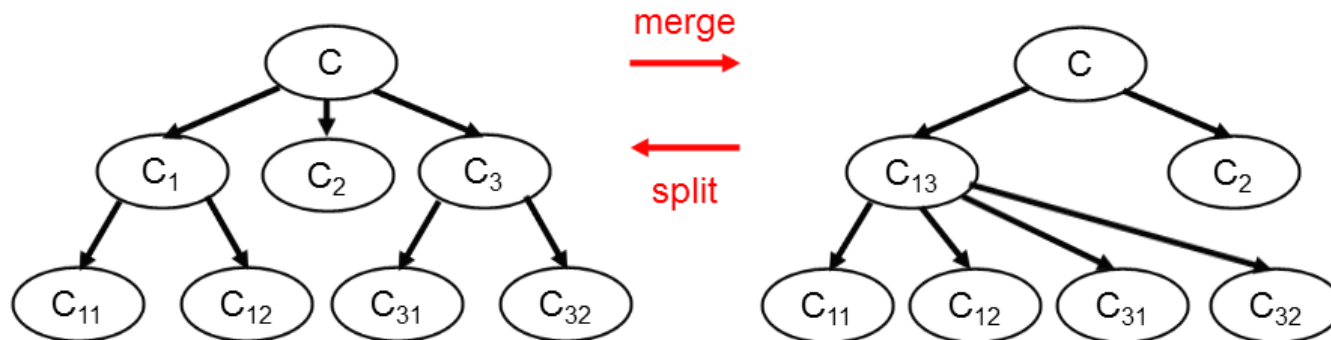
Name	BodyCover	HeartChamber	BodyTemp	Fertilization
fish	scales	two	unregulated	external
amphibian	moist_skin	three	unregulated	external
mammal	hair	four	regulated	internal
bird	feathers	four	regulated	internal
reptile	cornified_skin	imperfect_four	unregulated	internal



Fisher: Conceptual Acquisition Via Incremental Conceptual Clustering

COBWEB algorithm

- instance by instance makes a (classification) tree,
 - internal node = (probabilistic) concept, leaf = instance,
- for each instance takes one of the following operators
 - create a new class – find a position in the hierarchy for a new subclass,
 - assign the instance into an existing class – an object is similar with the objects belonging to one of the classes,
- to minimize the influence of instance ordering, concern the learning operators
 - merge classes – two classes replaced by one class,
 - split class – one class breaks to subclasses or individual objects,
- bi-directional hill-climbing search driven by the category utility function.



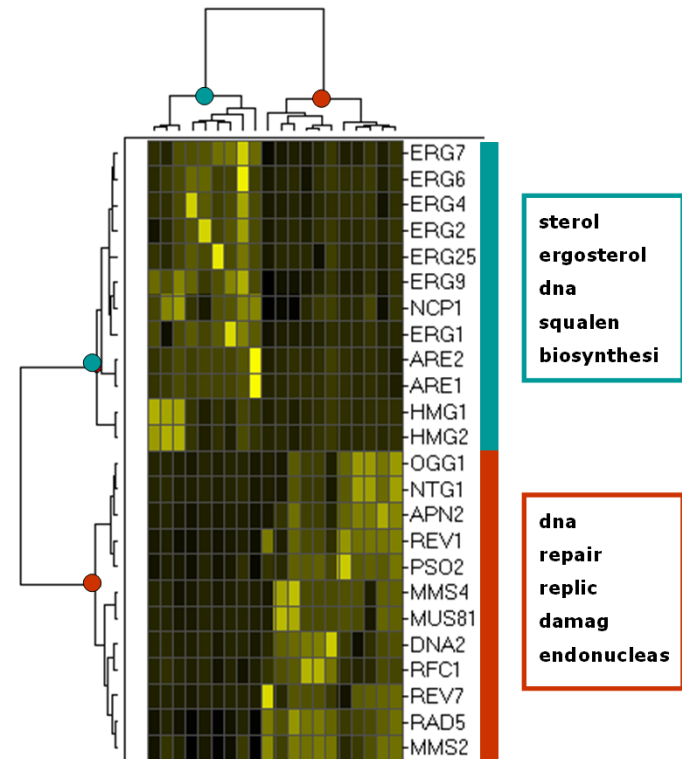
COBWEB: summary

- algorithm properties
 - symbolic: nominal features,
 - hierarchical: creates a taxonomy,
 - incremental: the output made gradually, instance by instance,
- what **language** it works with?
 - the probabilistic concept description is a weak language (a vague one),
 - it permits an arbitrary concept
 - * an example for three binary features: $Pr(x|C_j) = [0.6, 0.5, 0.7]$,
 - when enhanced by constraints it is a stronger language
 - * define α factor for minimum deviation from 0.5,
 - * the constraint can apply to the maximum deviation or all the deviations (features),
 - * $\alpha = 0.3$ for one feature at least
 - the vector $[0.6, 0.5, 0.7]$ is not a concept, $[0.6, 0.5, 0.9]$ vice versa,
 - * the ultimate case $\alpha = 0.5$ for all the features
 - the language of logical conjunctions,
- how would you classify an unseen example?



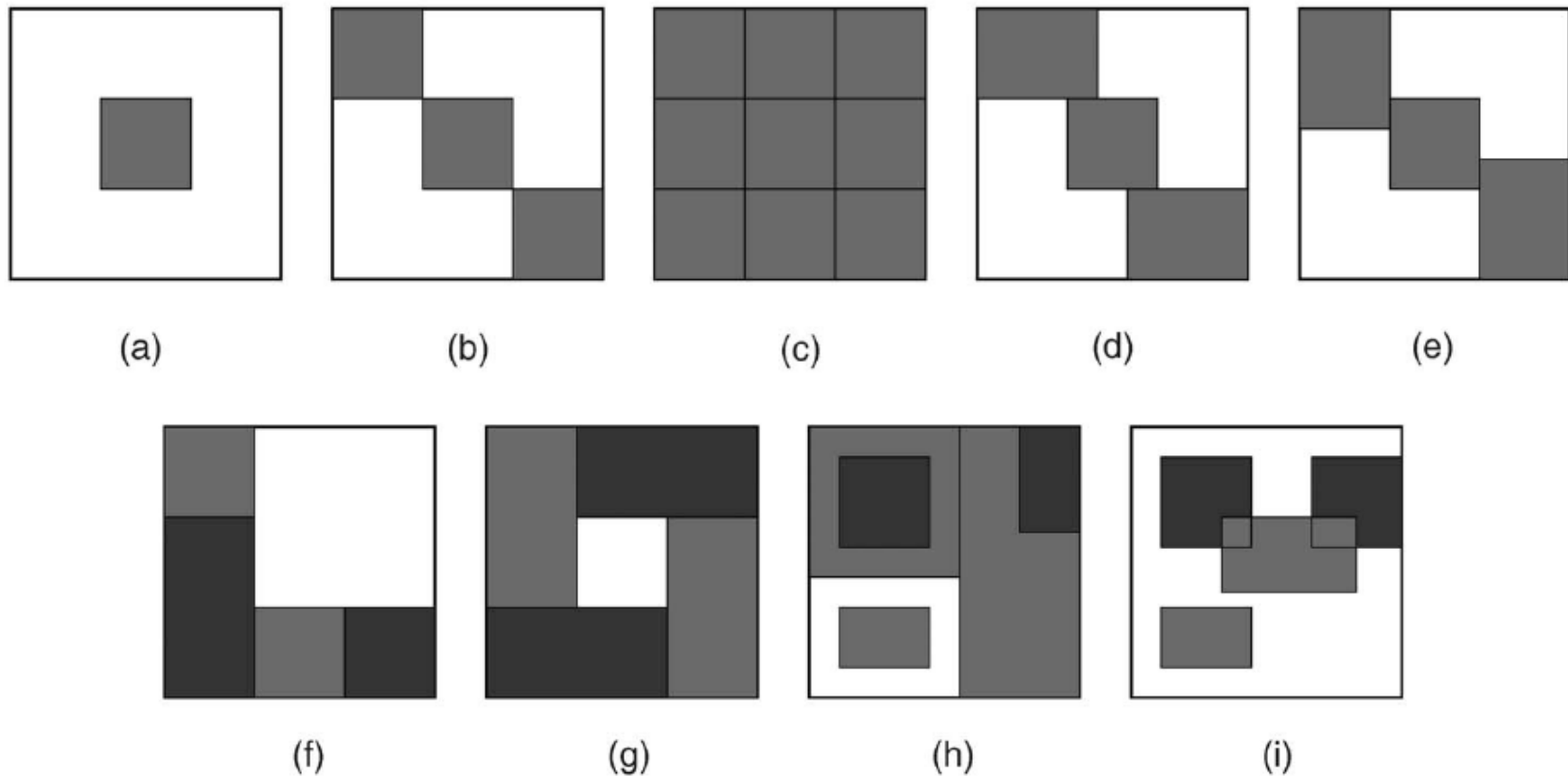
Bi-clustering (co-clustering)

- (hierarchical) both for instances and features,
- frequent in gene expression data analysis or text mining,
 - instances: tissues, features: genes,
- figure: two-way clustering carried out independently (successively) in both dimensions
 - comprehensible visualisation,
 - a global model
 - * all tissues used to cluster genes,
 - * all genes used to cluster tissues,
- bi-clustering clusters **concurrently** in both dimensions
 - a local model
 - * only relevant tissues to cluster particular genes,
 - * only relevant genes to cluster particular tissues,
 - genes can have multiple functions, different biological conditions trigger different functions,
 - incomplete and inexclusive (skip object/feature, cluster overlaps).



Bi-clustering – cluster structures

(a) single bi-cluster, (b) exclusive rows and columns, (c) checkerboard, (d) exclusive rows, (e) exclusive columns, (f) hierarchical nonoverlapping, (g) nonoverlapping nonexclusive, (h) hierarchical overlapping, (i) arbitrarily positioned overlapping.



Madeira, Oliveira: Biclustering Algorithms for Biological Data Analysis.

Bi-clustering – an algorithm example

- **block clustering** (also direct clustering, Hartigan)

1. order rows (columns) by row (column) means,
2. find the best row or column split,
 - (a) criterion is the decrease in the internal block variance,

$$\sum_{c=1}^k \sum_{i \in I, j \in J} (x_{ij} - x_{IJ})^2$$

(I, J) is a bi-cluster, $I \subseteq \{1 \dots m\}$, $J \subseteq \{1 \dots n\}$, x_{IJ} is the bi-cluster mean,

- (b) splits always keep the row/column order – the linear number of tests with m and n ,
3. repeat the previous step in both sub-blocks until the given k is reached,
 - hierarchical divide and conquer approach,
 - trivial, fast algorithm with roughly suboptimal solutions,
 - searches for (f) type bi-clusters, ie. hierarchical nonoverlapping,
 - preference for constant bi-clusters – ideally the only value in whole bi-cluster.



Block clustering – US presidential elections

- the republican vote for president (the republican candidate vote percentage),
- the southern US states in 1900-1968.

State	Year																	
	12	36	32	40	44	48	16	04	68	08	24	00	20	28	56	60	52	64
SC	1	1	2	4	4	4	2	5	39	6	2	7	4	9	25	49	49	59
MI	2	3	4	4	6	3	5	5	14	7	8	10	14	18	24	25	40	87
GA	4	13	8	15	18	18	7	18	30	31	18	29	29	45	33	37	30	54
LA	5	11	7	14	19	17	7	10	23	12	20	21	31	24	53	29	47	57
AA	8	13	14	14	18	19	22	21	14	24	27	35	31	48	39	42	35	70
TS	9	12	11	19	17	25	17	22	40	22	20	31	24	52	55	49	53	37
FA	8	24	25	26	30	34	18	21	41	22	28	19	31	57	57	52	55	48
AS	20	18	13	21	30	21	28	40	31	37	29	35	35	39	46	43	44	44
VA	17	29	30	32	37	41	32	37	43	38	33	44	38	54	55	52	56	46
NC	12	27	29	26	33	33	42	40	40	46	40	45	43	55	49	48	46	44
TE	24	31	32	33	39	37	43	43	38	46	44	45	51	54	49	53	50	44
KY	25	40	40	42	45	41	47	47	44	48	49	49	49	59	54	54	50	36
MD	24	37	36	41	48	49	45	49	42	49	45	52	55	57	60	46	55	35
MO	30	38	35	48	48	42	47	50	45	49	50	46	55	56	50	50	51	36
WV	21	39	44	43	45	42	49	55	40	53	49	54	55	58	54	47	48	32
DE	33	43	51	45	45	50	50	54	45	52	58	54	56	65	55	49	52	39

Hartigan: Direct Clustering of a Data Matrix.

Bi-clustering – another algorithm

■ Cheng and Church

- the first bi-clustering algorithm used for microarray data,
- stems from the modified variance definition (residues)

$$\sum_{c=1}^k \frac{1}{|I||J|} \sum_{i \in I, j \in J} (x_{ij} - x_{Ij} - x_{iJ} + x_{IJ})^2$$

- * superpose background (x_{IJ}), gene effect (x_{iJ}) and effect of biological conditions (x_{Ij}),
- problem: some trivial bi-clusters having zero residue ($1 \times m$ or $1 \times n$)
 - * introduces a threshold residue δ , searches for the largest bi-clusters having residue $< \delta$,
- greedy algorithm (bi-clusters of the type (i))
 1. start with the whole matrix,
 2. remove a row or column causing the top residual decrease,
 3. repeat step 2 until the residue cannot be further decreased or it is $< \delta$,
 4. randomize the values in the found bi-cluster, repeat (1-3) for k bi-clusters.

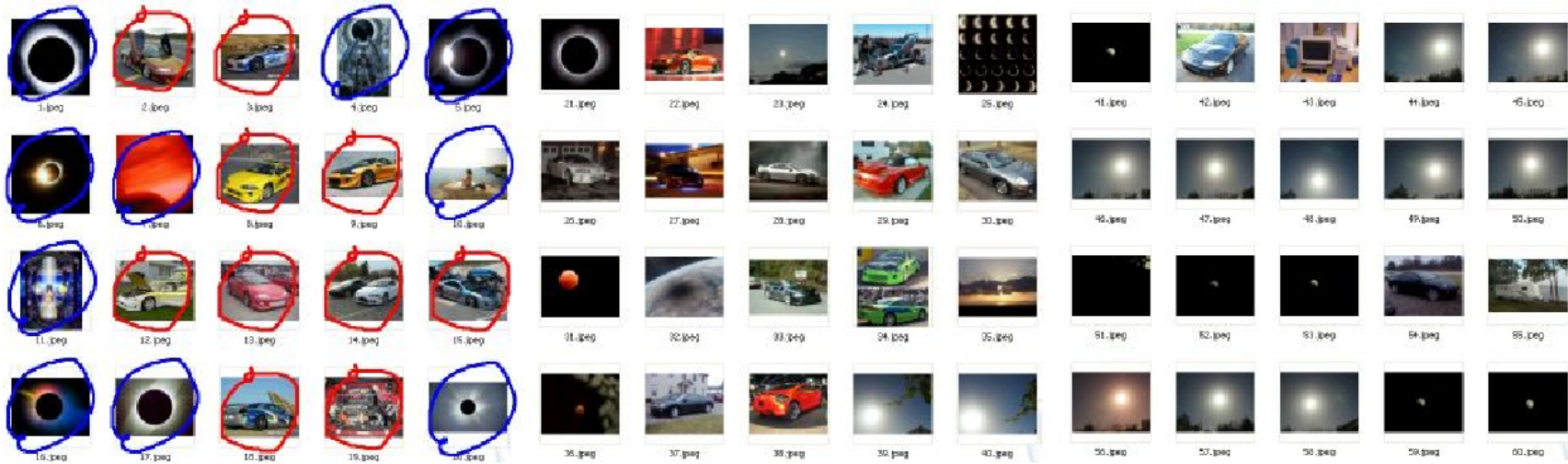
- figure: additive (left pane) a multiplicative (right pane) bi-cluster.

1.0	2.0	5.0	0.0
2.0	3.0	6.0	1.0
4.0	5.0	8.0	3.0
5.0	6.0	9.0	4.0

1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

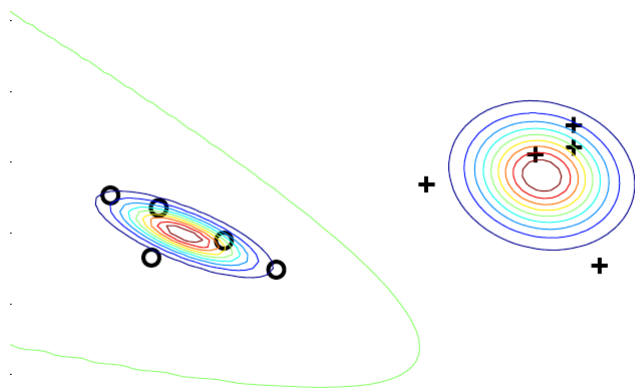
Semi-supervised clustering

- how to learn to recognize eclipse photographs in reasonable time?

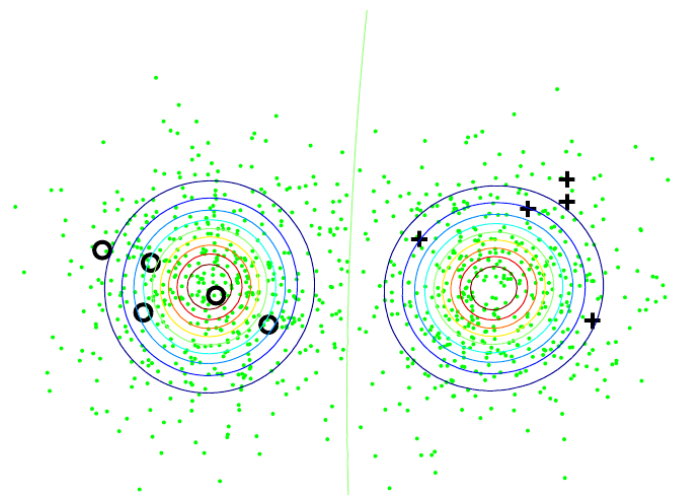


Semi-supervised clustering

- success story: document classification into 20 discussion groups [Nigam, 2006]
 - 2 classified training instances per discussion group available only,
 - further we have 10000 unclassified instances,
 - naïve Bayes model made from 40 documents (without clustering): 27% class. accuracy,
 - the same model type for 10040 documents (with clustering): 43% classification accuracy,
 - necessary condition to succeed: data clusters must match classes.
- an explanation for 16% increase in accuracy?



Decision boundary for annotated instances only



Decision boundary with unannotated instances

Zhu: Semi-Supervised Learning Tutorial



**OPPA European Social Fund
Prague & EU: We invest in your future.**
