



Wrocław  
University  
of Science  
and Technology

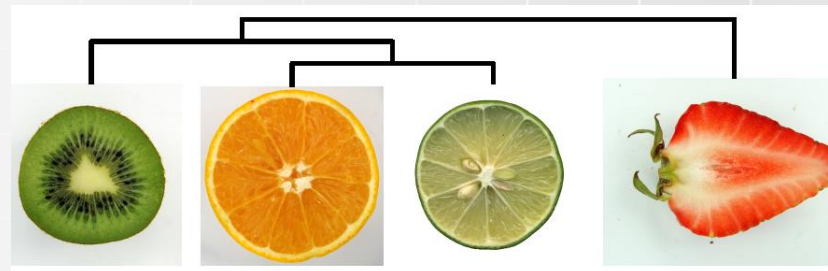
# Object Cluster Hierarchy – a new paradigm of hierarchical clustering

Halina Kwaśnicka

Department of Artificial Intelligence

Wrocław University of Science and Technology

Wrocław, Poland



GHOST DAY 2024

Poznań, 05.04.2024 – 06.04.2024

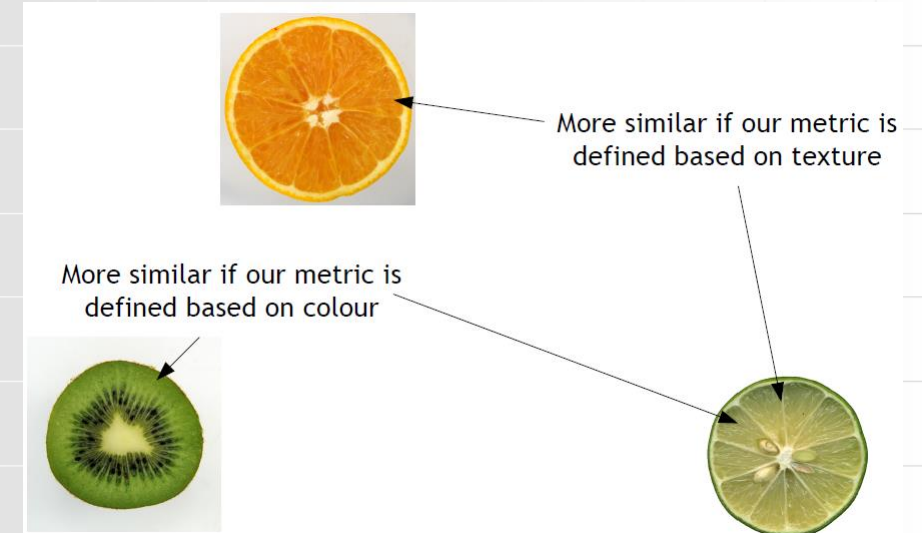


# OUTLINE

1. Cluster Analysis – introduction
2. Hierarchical Clustering (HC) methods vs Human perception of hierarchical data
3. Tree-Structured Stick Breaking for Hierarchical Data (TSSBH) – an extension to the Hierarchical Clustering paradigm
4. Proposed modifications – OCH (IRV-HC)
5. How to evaluate – Partial Order F-Score
6. Benchmark data generator
7. Summary

# THE CLUSTERING PROBLEM

- The task: generating meaningful groups (clusters) of data points in the provided dataset
- Can we define „meaningfulness of groups“?
  - points within any particular cluster are as similar as possible
  - points that belong to different clusters are as different as possible
- What is similarity?
  - Interpretation will vary depending on the considered set of objects



- Usually we say that one pair of objects is more similar than another

# TYPES OF CLUSTERING METHODS

- The clustering methods differ in the type of results they produce

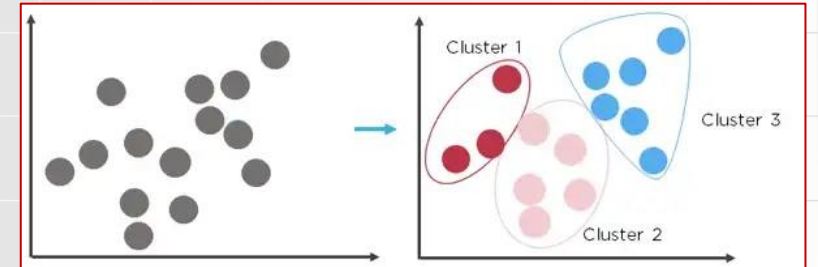
## 1. Flat Clustering

- it organizes points into a predefined number of clusters
- the only relation between them is spatial

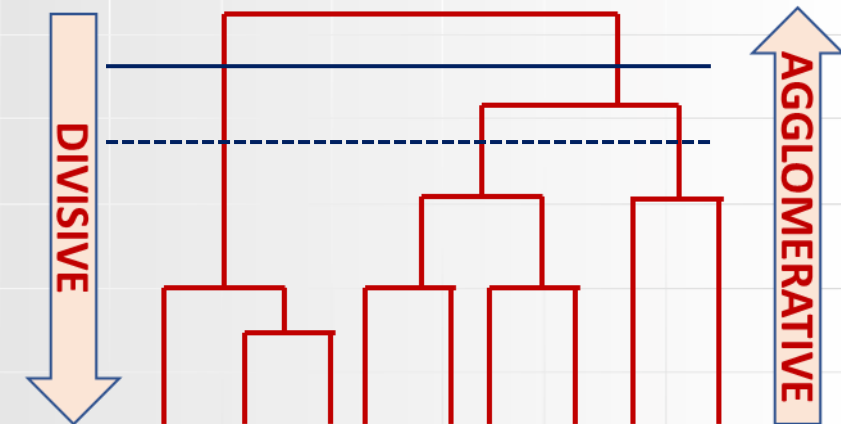
## 2. Hierarchical Clustering (HC) or Hierarchical Cluster Analysis (HCA)

- produces several flat clustering solutions
- they are organized into a tree structure (dendrogram)

- Hierarchical clustering assigns objects to clusters
- It also build the relation between clusters

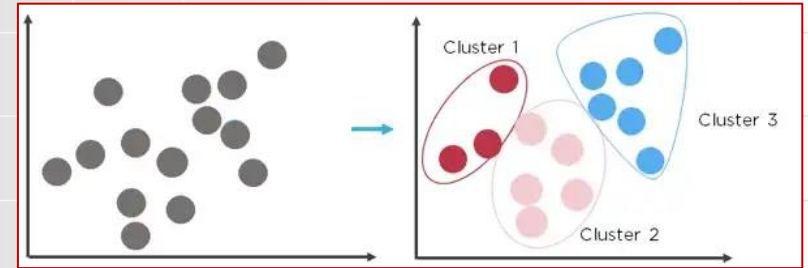


<https://www.oneai.com/learn/text-clustering>

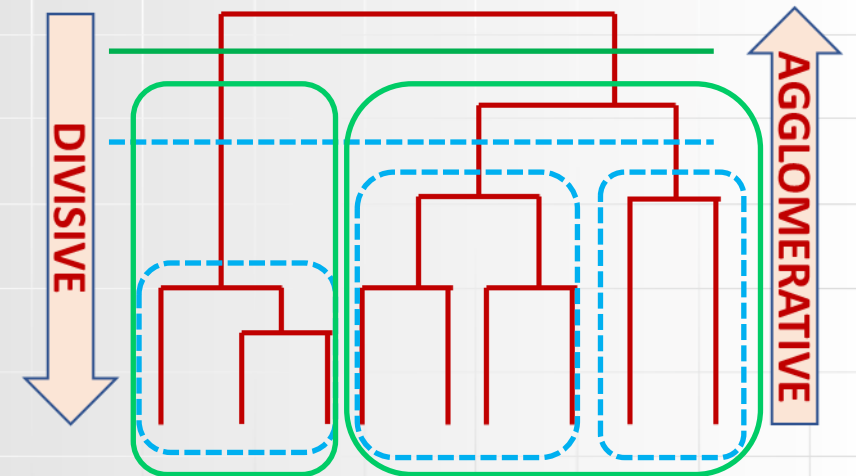


# THE TYPES OF CLUSTERING METHODS

- The clustering methods differ in the type of results they produce
  1. Flat Clustering
    - it organizes points into a predefined number of clusters
    - the only relation between them is spatial
  2. Hierarchical Clustering (HC) or Hierarchical Cluster Analysis (HCA)
    - produces several flat clustering solutions
    - they are organized into a tree structure (dendrogram)
- Hierarchical clustering assigns objects to clusters
- It also build the relation between clusters



<https://www.oneai.com/learn/text-clustering>



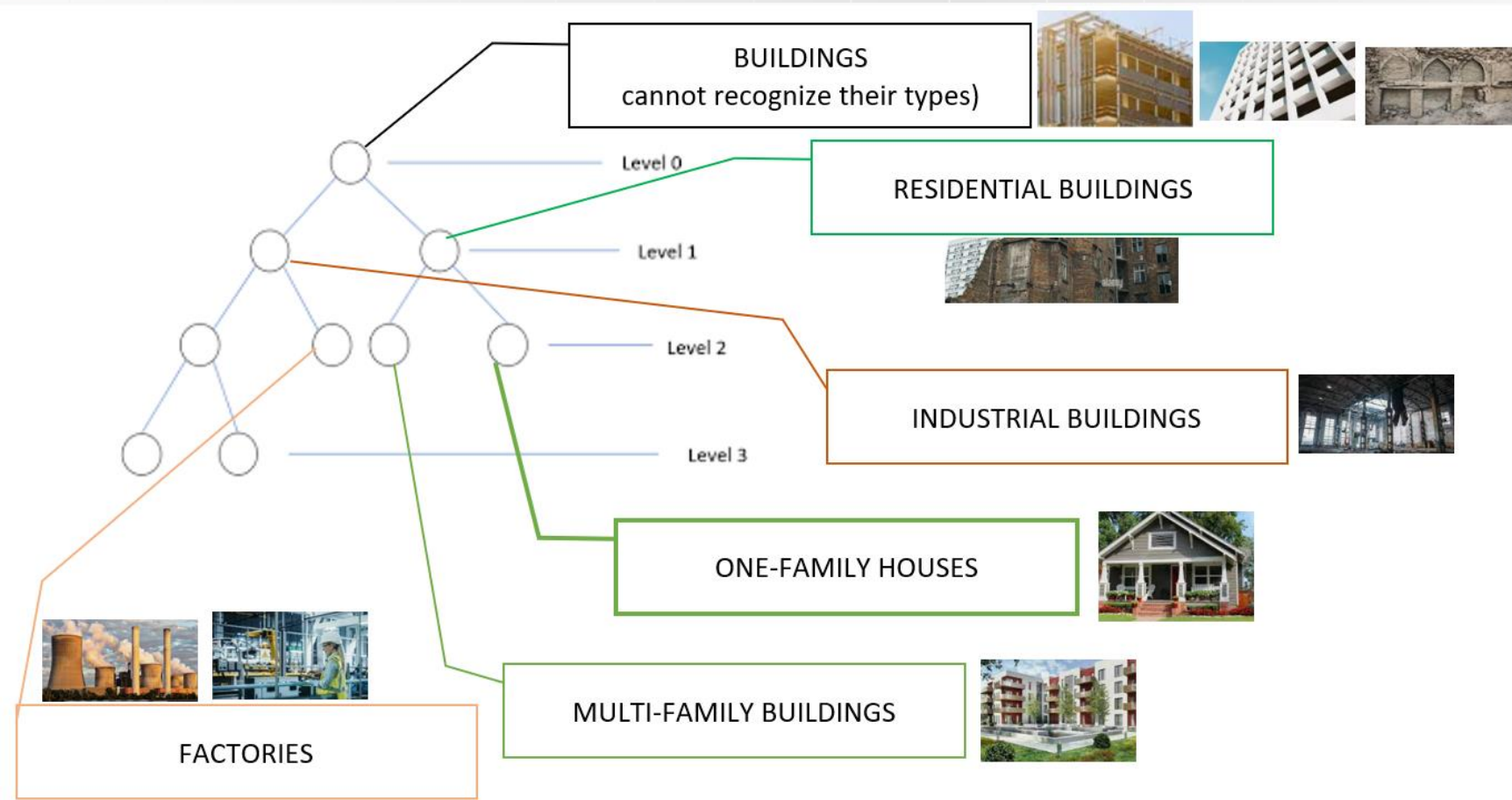
The objects located only in leaf clusters are not in a hierarchical relation (a partial order) with each other

# THE RATIONALE BEHIND THE RESEARCH

We focus on hierarchical methods where the primary issue is a semantic gap between how humans perceive hierarchies and the results produced by Hierarchical Clustering methods

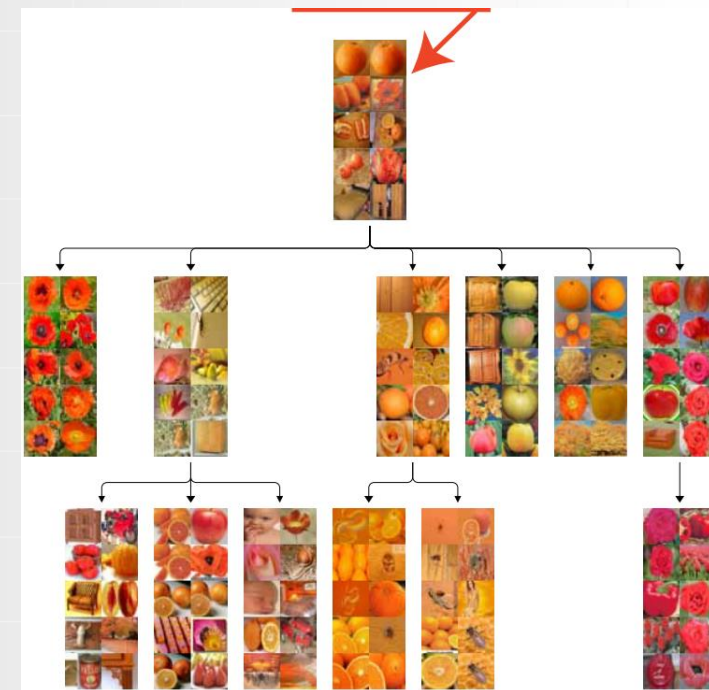
- Our goal: generation of hierarchy structures of data
- Human perception describes hierarchical data as possessing the following properties:
  1. the data can be present in any node in the hierarchy and belongs to that node without being propagated to the child nodes;
  2. the data in the child nodes should represent equal or more precise concepts than the data in the corresponding parent, which in turn should resemble more general concepts;
  3. the data in a node should be more similar to data in the parent and child nodes than to unrelated nodes located in other subtrees of the hierarchy

# DESIRED SEMANTIC RESULTS OF OCH



# THREE REQUIRED PROPERTIES OF CLUSTERING

- We put following requirements behind the method:
  1. **Inheritance** – if an element belongs to a group it also belongs to the parents' groups, up to the root
  2. **Retention** – elements do not need to be located in the tree's leaves
  3. **Variance** – groups located lower in the hierarchy are more specific  
(children cannot have higher variation than their parents)
- Interesting approach is proposed in [1] (TSSB)



A part of Fig. 3 from [1]

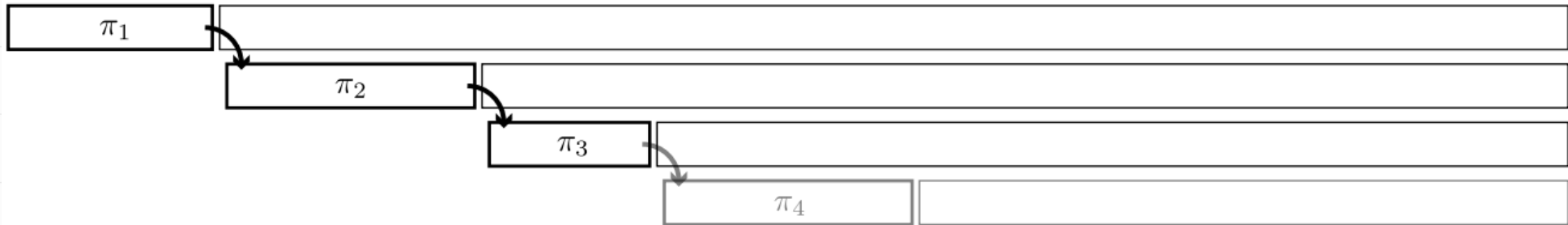
[1] Zoubin Ghahramani, Michael Jordan, Ryan P. Adams: Tree-Structured Stick Breaking for Hierarchical Data. *Statistics* 23(1) (2010)



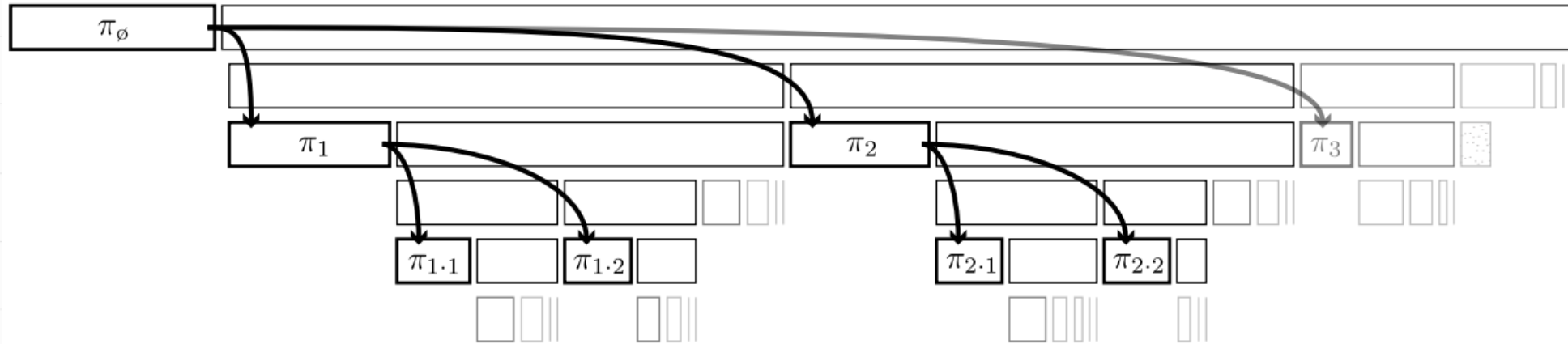
# ABOUT THE METHOD BEING THE ROOT OF OURS

- It uses **nested stick-breaking processes** to allow for trees of **unbounded width and depth**
- A stick-breaking approach allows **applying Markov Chain Monte Carlo methods based on Slice Sampling to perform Bayesian Inference** and simulate from the posterior distribution on trees
- PROS and CONS of that method
  - + The method allows for data to be assigned to every node
  - + The Tree Structured Stick Breaking (TSSB) process allows for trees with different structures to form depending on the hyperparameters values
  - This method does not guarantee the **first** and **third** required clustering property
- This method **inspired us to develop a modified variant with altered properties**

# NESTED STICK-BREAKING PROCESSES



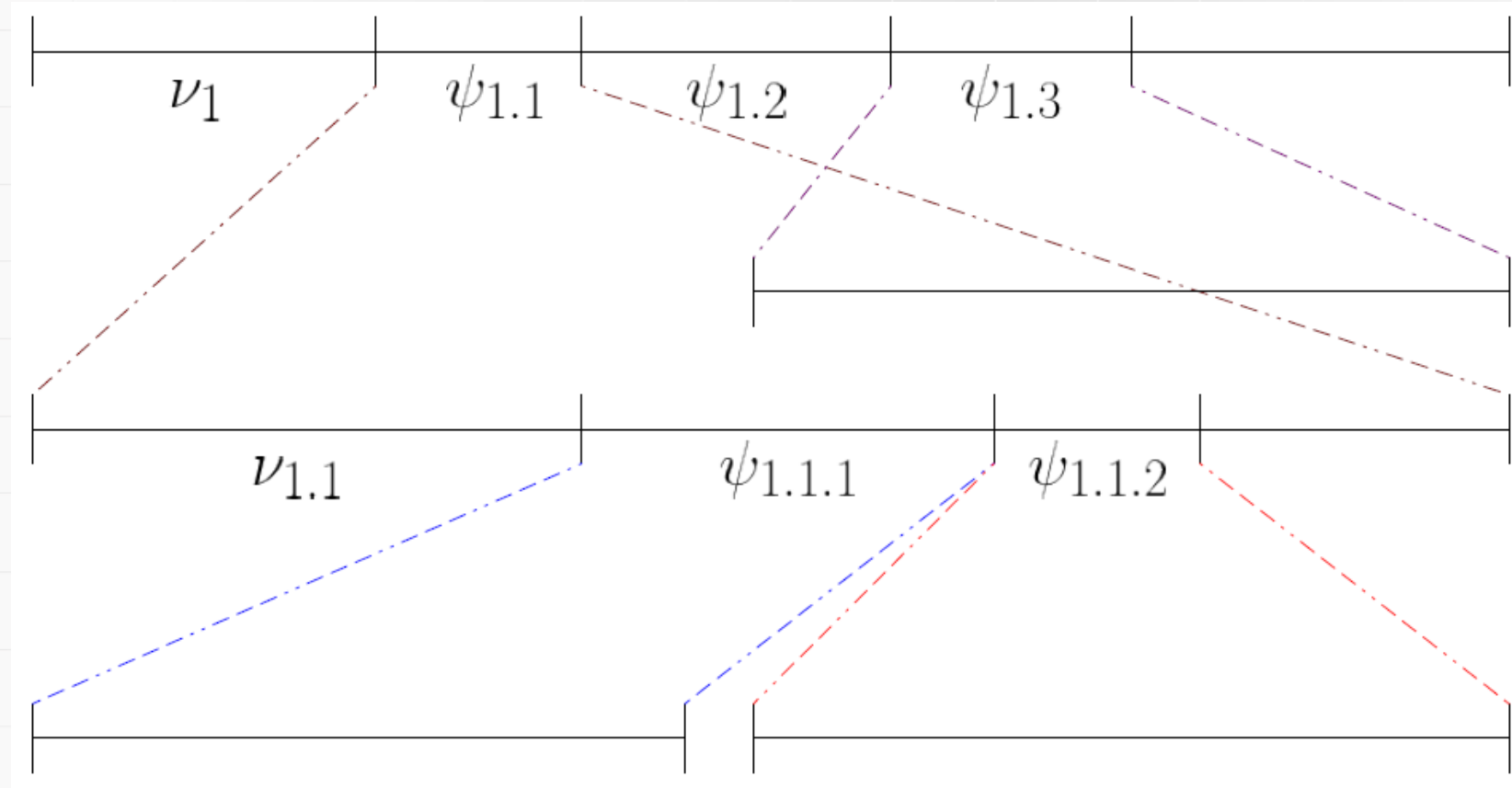
(a) Dirichlet process stick breaking



(b) Tree-structured stick breaking

- Dirichlet processes and tree-structured stick breaking – the idea [1]

# TREE STRUCTURED STICK BREAKING



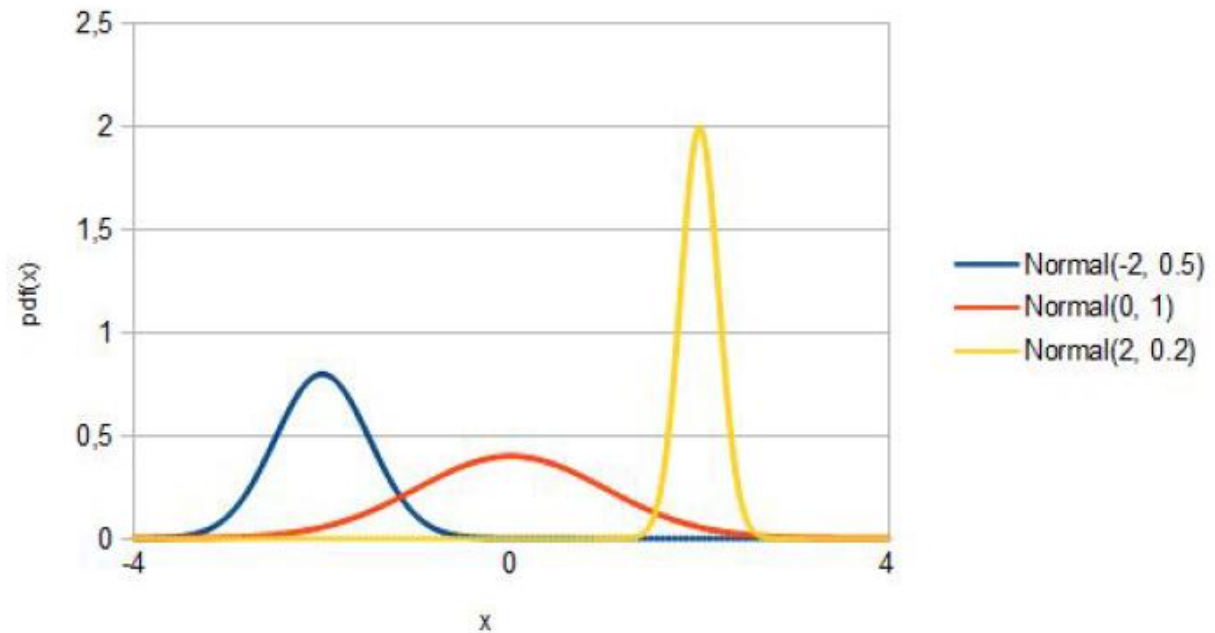
# DISTRIBUTIONS: GAUSS (A.K.A. THE NORMAL)

Domain:

$$D_{Norm} = \mathbb{R}$$

Parameters:

$$\mu \in \mathbb{R}, \sigma \in \mathbb{R}_+$$



PDF:

$$pdf(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

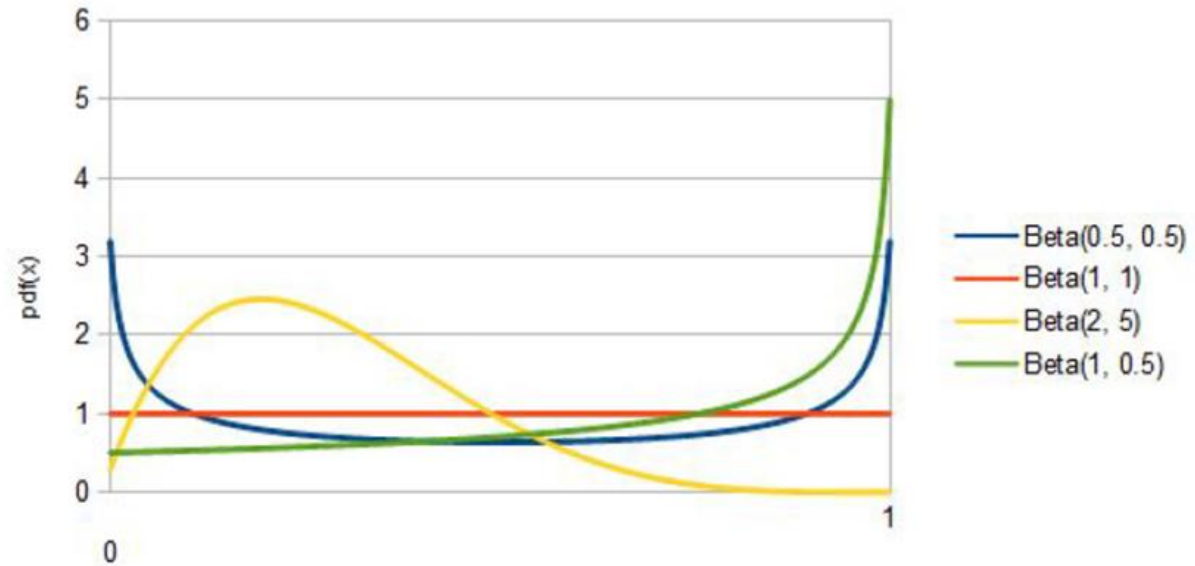
# DISTRIBUTIONS: BETA

Domain:

$$D_{Beta} = (0, 1)$$

Parameters:

$$\alpha \in R_+, \beta \in R_+$$



PDF:

$$pdf(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}$$

In **mathematics**, the **beta function**, also called the **Euler integral** of the first kind, is a **special function** defined by

$$B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1} dt$$

for  $\text{Re}(x), \text{Re}(y) > 0$ .

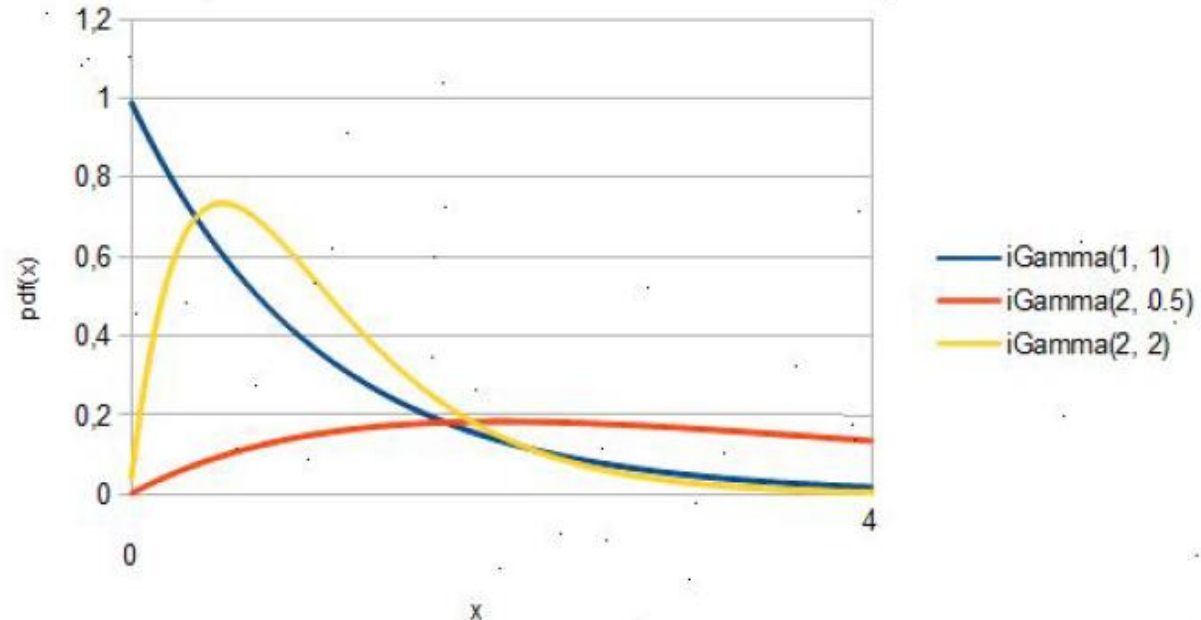
# DISTRIBUTIONS: INVERCE GAMMA

Domain:

$$D_{iGamma} = \mathbb{R}_+$$

Parameters:

$$\alpha \in \mathbb{R}_+, \beta \in \mathbb{R}_+$$



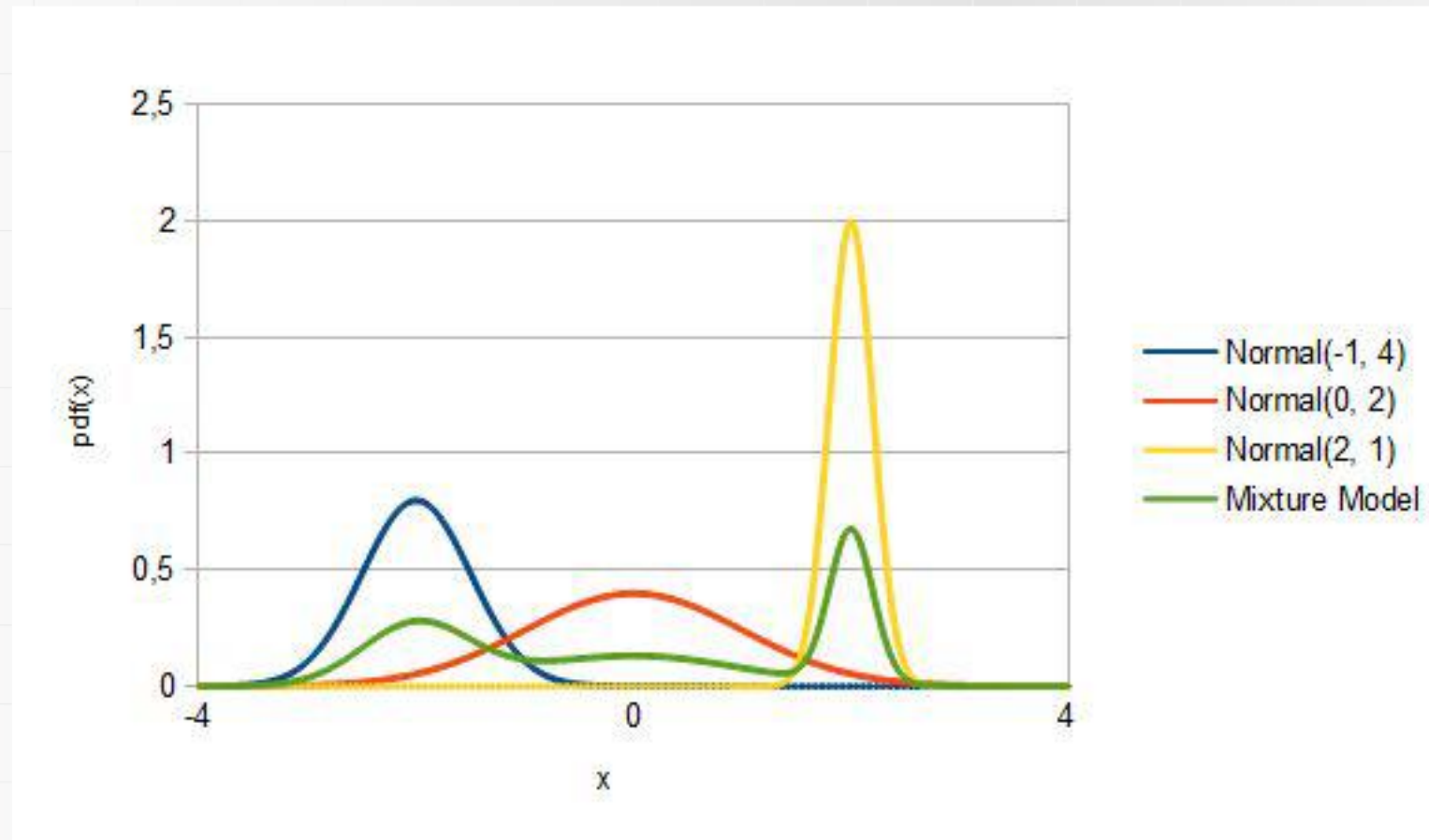
PDF:

$$pdf(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\left(\frac{-\beta}{x}\right)$$

The gamma function is defined for all complex numbers except the non-positive integers. For complex numbers with a positive real part, it is defined via a convergent improper integral:

$$\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx.$$

# MIXTURE MODEL



Example of Mixture Model: three Normal Distributions, each with equal probability of  $1/3$

# MODEL

- We assume that:
  - Our data comes from a *Mixture Model*
  - This is an infinite *Mixture Model*
  - **The weights of the *mixture* come from a *Dirichlet Process* parametrized with a *Beta* distribution**
  - The mixtures are uncorrelated *Normal* distributions
  - In these distributions the *means* is drawn from a Normal distribution and the *variance* from an *Inverse Gamma* distribution



# EXAMPLES OF TREES OF 50 DATA WITH DIFFERENT HYPERPARAMETERS VALUES [1]

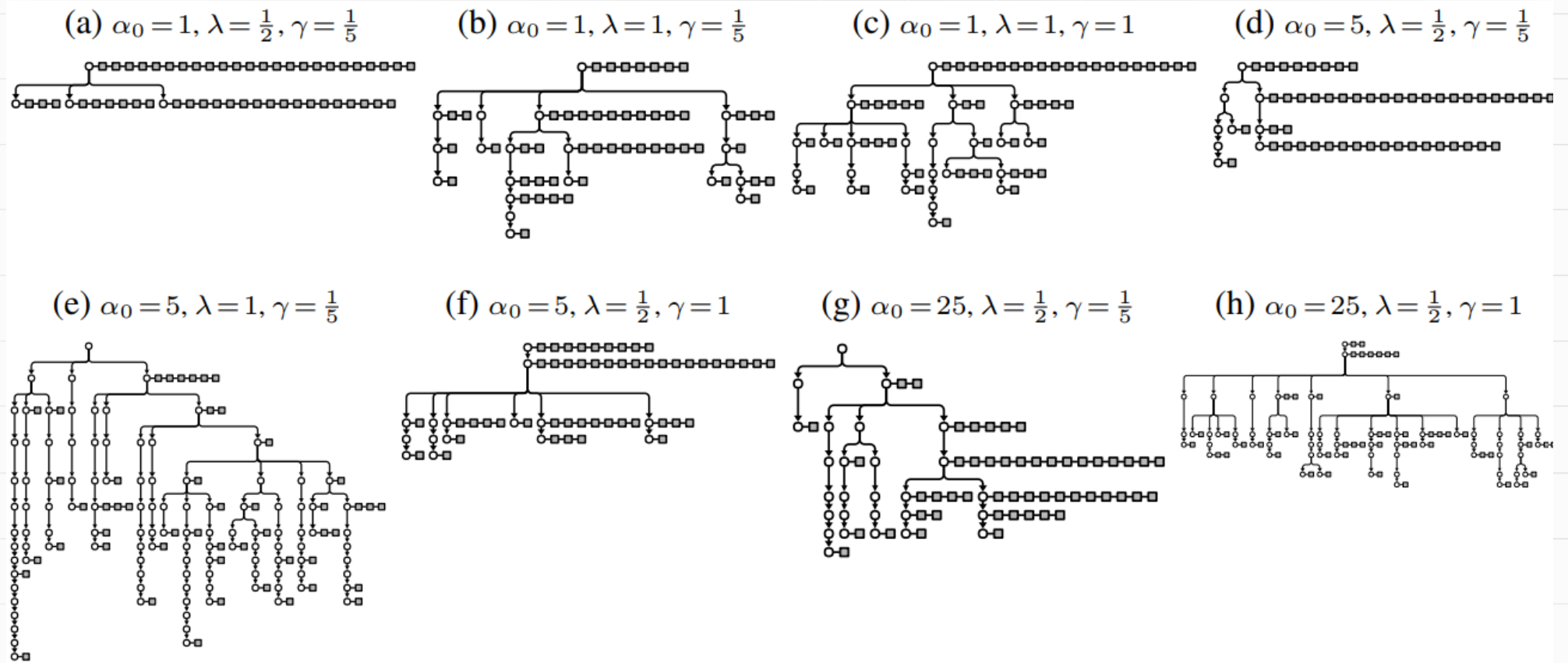


Figure 2: Eight samples of trees over partitions of fifty data, with different hyperparameter settings. The circles are represented nodes, and the squares are the data. Note that some of the sampled trees have represented nodes with no data associated with them and that the branch ordering does not correspond to a size-biased permutation.

# Pros and cons

## Strengths:

- No assumptions about the number of levels or children per node
- Data assigned to any node
- Adaptable to different types of data

## Weaknesses:

- Relation between clusters still does not match common understanding of hierarchy (nodes located lower are not more specific)
- Not a robust method, prone to numerical error

The TSSB method does not have the required properties, can it be modified to have them?

# OUR REQUIREMENTS FOR THE METHOD

## Three properties:

- **I**nheritance
  - **R**etention
  - **V**ariance
- Initially we called our method *Inheritance, Retention, Variance Hierarchical Clustering* (IRV-HC),
  - Now we prefer the shorter name *Object Cluster Hierarchy* (OCH)

# CHANGES TO MODEL

The new kernel:

$$\begin{aligned} \sigma_{\epsilon\epsilon_i} &= \sigma_{\epsilon} \text{Beta}(\alpha_{\sigma}, \beta_{\sigma}) \Rightarrow \\ \Rightarrow \sigma_{\epsilon\epsilon_i} &\in (0, \sigma_{\epsilon}) \end{aligned}$$

$\sigma$  – variance

$\epsilon$  – specific cluster

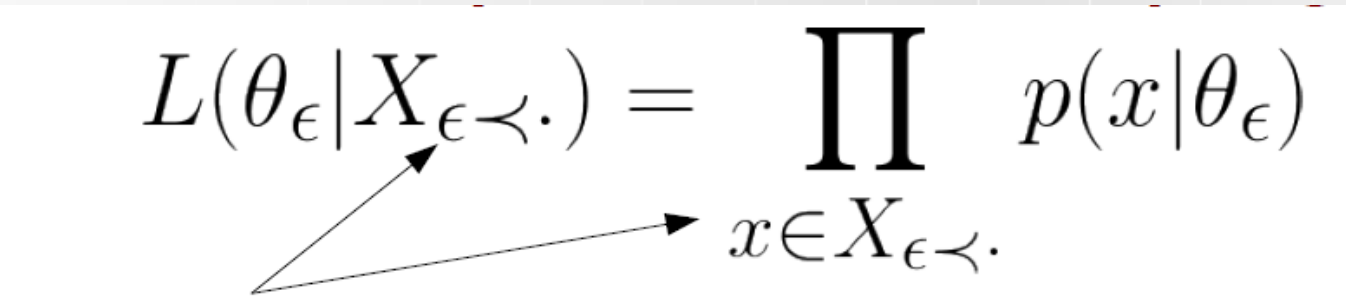
$\epsilon\epsilon_i$  – the  $i$ -th child of cluster  $\epsilon$

Implications:

- All children will have lower variance than their parent
- Root will have highest variance of all nodes
- Variance still cannot fall to 0

# CHANGES TO OPERATORS

## Cluster parameter resampling:

$$L(\theta_\epsilon | X_{\epsilon \prec \cdot}) = \prod_{x \in X_{\epsilon \prec \cdot}} p(x | \theta_\epsilon)$$


Data in node and all descendant nodes

Previously:

$$L(\theta_\epsilon | X_\epsilon, \Theta) = p(\theta_\epsilon | \theta_p) \prod_{x \in X_\epsilon} p(x | \theta_\epsilon) \prod_{\epsilon \epsilon_i \succ \epsilon} p(\theta_{\epsilon \epsilon_i} | \theta_\epsilon)$$

## New operator: Resampling parent-child assignments

$\theta$  - parameters of the node

$\epsilon \prec \cdot$  - the descendants of node  $\epsilon$

# EXPERIMENTS WITH THE ORIGINAL TSSB-HC AND THE PROPOSED VERSION IRV-HC (OCH)

- The goal: to evaluate **how the modified IRV-HC method performs in comparison with the baseline TSSB-HC method**
- Data were generated from a three dimensional model corresponding to a hierarchical mixture model
- **400 sets of data** were drawn from the test mixture model
- Both methods were applied to each set
- The parameters for both methods were identical when possible
- **Two measures were calculated** for the methods in each of the 400 experiments, the results were averaged
- **The variance of the nodes at specific levels of the generated tree was analyzed**

# Proposed method (IRV-HC) vs. the original TSSB-HC – EXPERIMENT RESULTS

Comparison of average variance by level for 10 random sets of data

Test:	Root	Level 1	Level 2	Level 3	Level 4	Level 5	Meets variance property?
TSSB-HC	0.74	0.51	0.55	0.62	0.67	0.70	No
IRV-HC	2.49	1.89	0.52	0.50	n/a	n/a	Yes
TSSB-HC	6.76	0.51	1.31	1.43	0.50	n/a	No
IRV-HC	2.50	0.70	0.53	0.51	n/a	n/a	Yes
TSSB-HC	5.13	0.89	1.07	0.67	n/a	n/a	No
IRV-HC	2.50	1.37	0.73	0.61	0.85	0.50	No
TSSB-HC	0.73	0.53	0.59	0.54	n/a	n/a	No
IRV-HC	2.54	1.22	0.50	0.50	n/a	n/a	Yes
TSSB-HC	2.03	0.90	0.77	0.74	n/a	n/a	Yes
IRV-HC	2.29	1.37	0.50	0.50	0.50	n/a	Yes
TSSB-HC	6.32	0.52	0.77	0.94	0.65	n/a	No
IRV-HC	2.33	0.63	0.54	0.50	0.50	n/a	Yes
TSSB-HC	4.90	0.53	0.60	n/a	n/a	n/a	No
IRV-HC	2.47	1.44	0.50	0.50	0.50	n/a	Yes
TSSB-HC	8.77	0.53	1.33	0.79	0.84	n/a	No
IRV-HC	2.24	1.89	0.51	n/a	n/a	n/a	Yes
TSSB-HC	1.14	0.62	6.15	0.72	n/a	n/a	No
IRV-HC	2.54	1.25	0.51	0.50	n/a	n/a	Yes
TSSB-HC	6.31	0.51	0.58	0.61	0.70	n/a	No
IRV-HC	2.29	0.82	0.51	0.50	n/a	n/a	Yes

# Proposed method (IRV-HC) vs. the original TSSB-HC – EXPERIMENT RESULTS

Comparison of average variance by level for 10 random sets of data

Test:	Root	Level 1	Level 2	Level 3	Level 4	Level 5	Meets variance property?
TSSB-HC	0.74	0.51	0.55	0.62	0.67	0.70	No
IRV-HC	2.49	1.89	0.52	0.50	n/a	n/a	Yes
TSSB-HC	6.76	0.51	1.31	1.43	0.50	n/a	No
IRV-HC	2.50	0.70	0.53	0.51	n/a	n/a	Yes
TSSB-HC	5.13	0.89	1.07	0.67	n/a	n/a	No
IRV-HC	2.50	1.37	0.73	0.61	0.85	0.50	No
TSSB-HC	0.73	0.53	0.59	0.54	n/a	n/a	No
IRV-HC	2.54	1.22	0.50	0.50	n/a	n/a	Yes
TSSB-HC	2.03	0.90	0.77	0.74	n/a	n/a	Yes
IRV-HC	2.29	1.37	0.50	0.50	0.50	n/a	Yes
TSSB-HC	6.32	0.52	0.77	0.94	0.65	n/a	No
IRV-HC	2.33	0.63	0.54	0.50	0.50	n/a	Yes
TSSB-HC	4.90	0.53	0.60	n/a	n/a	n/a	No
IRV-HC	2.47	1.44	0.50	0.50	0.50	n/a	Yes
TSSB-HC	8.77	0.53	1.33	0.79	0.84	n/a	No
IRV-HC	2.24	1.89	0.51	n/a	n/a	n/a	Yes
TSSB-HC	1.14	0.62	6.15	0.72	n/a	n/a	No
IRV-HC	2.54	1.25	0.51	0.50	n/a	n/a	Yes
TSSB-HC	6.31	0.51	0.58	0.61	0.70	n/a	No
IRV-HC	2.29	0.82	0.51	0.50	n/a	n/a	Yes



## SUMMARY OF THE RESULTS

- **IRV-HC (OCH) repeatedly produces clustering with average variance dropping with further levels** (deviations in less than 5% of cases)
- TSSB-HC appears almost random (variance property met in approximately 10% of cases)
- Both methods use randomization, **IRV-HC works in a more predictable manner**
- The average variance of test data is approximately 2.50, which is represented in the IRV-HC method
- TSSB-HC produces results with significantly larger variance – root cluster contains outliers or data with significantly smaller variance

# PROBLEM WITH MEASURES SUITABLE FOR HIERARCHIES EVALUATION

- In hierarchical clustering there is a relation only between groups
- In Object Cluster Hierarchy (OCH), previously called IRV-HC, there is also a hierarchical relation (partial order) between the objects assigned to the clusters
- This is why maximum separation between clusters is not desirable
- Clusters that are in relation to each other should be less separated than unrelated clusters
- To at least partially fill this gap, we propose one, **new external measure – Partial Order F-Score**

Spytkowski, M., Olech, Ł.P., Kwaśnicka, H. (2016). Hierarchy of Groups Evaluation Using Different F-Score Variants. In: Nguyen, N.T. et al. (eds) Intelligent Information and Database Systems. ACIIDS 2016. Lecture Notes in Comp. Sc., vol 9621. Springer.  
[https://doi.org/10.1007/978-3-662-49381-6\\_63](https://doi.org/10.1007/978-3-662-49381-6_63)

# HIERARCHICAL F-SCORE – Partial Order F-Score

- Relationship between the **ground truth classes**:  $C_c = \{c\} \cup \bigcup_{i=1}^n C_{cc_i}$ 
  - $C_c$  – set containing class  $c$  and all its descendant classes;
  - $C_{cc_i}$  – set containing class  $cc_i$  and all its descendant classes;
  - $n$  – number of children for class  $c$
- Relationship between **groups in the hierarchy**:  $E_\epsilon = \{\epsilon\} \cup \bigcup_{i=1}^m E_{\epsilon\epsilon_i}$ 
  - $E_\epsilon$  – set containing node  $\epsilon$  and all its descendant nodes;
  - $E_{\epsilon\epsilon_i}$  – set containing node  $\epsilon\epsilon_i$  and all its descendant nodes;
  - $m$  – number of children for node  $\epsilon$

- The set of points belonging to a class (cluster) and its descendants

$$X_{C_c} = \bigcup_{C' \in C_c} X_{C'} \quad X_{E_\epsilon} = \bigcup_{\epsilon' \in E_\epsilon} X_{\epsilon'}$$

- **Classic version F-Score:** F-Score:  $F_1 = 2T_p / (2T_p + F_n + F_p)$
- Data points belong to only one class, in hierarchies this is no longer applicable
- **Hierarchical F-Score:** for each class  $c$ , find a cluster  $\epsilon$  in the hierarchy with the maximal F-measure:

$$F_c = \max_{\epsilon \in \Theta} \frac{2|X_{E_\epsilon} \cap C_c|}{|X_{E_\epsilon}| + |X_{C_c}|}$$

- The final quality of a hierarchy is calculated:

$$F_1 = \frac{\sum_{c \in \mathcal{C}} |X_{C_c}| F_c}{\sum_{c \in \mathcal{C}} |X_{C_c}|}$$

- **Partial order F-Score:**
- We propose to use the partial order relations between points ( $\subseteq$  instead of  $=$ ):

$$x_i G x_j \Leftrightarrow c_{x_i} \subseteq c_{x_j}, \quad x_i M x_j \Leftrightarrow \epsilon_{x_i} \subseteq \epsilon_{x_j} \text{ or } x_i G x_j \Leftrightarrow c_{x_i} \supseteq c_{x_j}, \quad x_i M x_j \Leftrightarrow \epsilon_{x_i} \supseteq \epsilon_{x_j}$$

# EXPERIMENTS

- Goal: to compare three measures:
  1. Classic F-score
  2. Hierarchical F-score
  3. Partial order F-score
- Experiments were conducted on eight dataset, generated using the Tree Structured Stick Breaking Process with different parameters, they control:
  - Average density of data per level
  - Tree depth
  - Sparsity of tree
- The three types of experiments were conducted:
  - A. Random Error Introduction Tests
  - B. Reduction to a Single Cluster
  - C. Removing the Cluster Hierarchy

# EXPERIMENTS – SUMMARY THE RESULTS

- Classic F-Score
  - + it reflects relations found in the flat and hierarchical clustering
  - + it can reach both the maximum and minimum value
  - + is simple to calculate
  - it does not work properly for hierarchies of clusters
  - it notices fewer types of errors than the other measures

# EXPERIMENTS – SUMMARY THE RESULTS

- Hierarchical F-Score
  - + reflects relations found in many types of structures: flat clustering, hierarchies, forests of hierarchies
  - + in some cases it can be optimized to work more efficiently
  - + it focuses more on the numerous classes (because it is a weighted sum)
  - it cannot possibly reach its minimal value
  - points on lower levels of the hierarchy contribute to the final result with a higher weight than points higher up
  - unoptimized version requires complex calculation

# EXPERIMENTS – SUMMARY THE RESULTS

- Partial Order F-Score

- + reflects relations found in many types of structures, including flat clustering, hierarchies, and forests of hierarchies
- + is capable of reaching both the maximum and minimum value
- + with points assigned only to leaf nodes, it is indistinguishable from the classic F-Score
- + it can be optimized to work as fast as classic F-score
- when unoptimized, is more complex to calculate than classic F-Score

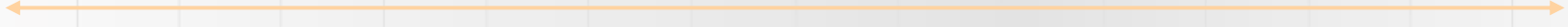


# LACK OF TOOLS FOR SYSTEMATIC ANALYSIS OF OBJECT CLUSTER HIERARCHIES (OCH)

- Working on ML methods require
  - Methods
  - Measure to evaluate and compare the method
  - Benchmark data available to every scientist
- Methods
  - TSSB-HC
  - OCH (earlier name IRV-HC)
  - BRT, ...
- Evaluation measures
  - Classic clustering indices (should be adapted)
  - Partial Order F-Score
- Benchmarks – ?

# BENCHMARKS

- Benchmarks – data to be used for understanding and testing the quality of the methods
  - CIFAR-10 dataset (the classes are completely mutually exclusive)
  - ?
- Benchmarks – expectations
  - A method should generate hierarchical structures of data with assumed, user-defined properties
  - It should produce data sets with hierarchical structures and with the ground truth assignment



# BENCHMARKS

- At [http://kio.pwr.edu.pl/?page\\_id=396](http://kio.pwr.edu.pl/?page_id=396) are freely available:
  - The implemented generator
  - The benchmarking datasets
  - Instructions on how to use it

Łukasz P. Olech, Michał Spytkowski, Halina Kwaśnicka, Zbigniew Michalewicz, Hierarchical data generator based on tree-structured stick breaking process for benchmarking clustering methods, *Information Sciences*, Volume 554, 2021, <https://doi.org/10.1016/j.ins.2020.12.020>.

# WHAT CHARACTERISTICS OF DATA CAN WE CONTROL?

- Parameters allow to control:
  - hierarchy depth
  - hierarchy width
  - data specificity
- Kernel parameters ( $p$  and  $q$ ) control the rate at which the children nodes become more specific than parents
- The parent distribution and the two kernel parameters influence the distribution of data points in a group
- The data generated from the model can be
  - scaled afterwards to any desired values
  - moved in any direction along any dimension

# REASSIGNMENT AS A POST-PROCESSING

- A point is assigned to the first node for which certain conditions are met
- This process is greedy and stochastic, it does not guarantee that a node with the highest probability of generating that point will be selected
- A hierarchy can undergo post-processing (reassignment)
- Reassignment moves the data between clusters so that each object is assigned to the cluster from which it is most likely to be generated

# SUMMARY DESCRIPTION OF HIERARCHIES PUBLISHED AS BENCHMARKING DATASET

- s00 – s07: datasets with an initial assignment of data
- s00r – s07r: the same data but with the reassigned hierarchies

<i>Set</i>	<i>Summary Description</i>
<i>s00 s00r</i>	<ul style="list-style-type: none"> <li>•the smallest hierarchical structures</li> <li>•similar depth and breadth</li> <li>•the highest number of instances per node</li> </ul>
<i>s01 s01r</i>	<ul style="list-style-type: none"> <li>•the most longitudinal hierarchies (high and narrow)</li> <li>•high number of instances per node</li> <li>•low number of nodes</li> <li>•low average number of children per node</li> </ul>
<i>s02 s02r</i>	<ul style="list-style-type: none"> <li>•medium number of nodes</li> <li>•wide hierarchies</li> <li>•low average number of instances per node</li> <li>•medium number of leaves</li> </ul>
<i>s03 s03r</i>	<ul style="list-style-type: none"> <li>•similar to <i>s00</i> and <i>s00r</i> but a bit larger structures</li> </ul>
<i>s04 s04r</i>	<ul style="list-style-type: none"> <li>•the highest hierarchies</li> <li>•the largest number of nodes</li> <li>•very narrow</li> <li>•the lowest average number of children per node</li> </ul>
<i>s05 s05r</i>	<ul style="list-style-type: none"> <li>•very wide hierarchies</li> <li>•medium number of nodes and leaves</li> <li>•the largest number of children per node</li> </ul>
<i>s06 s06r</i>	<ul style="list-style-type: none"> <li>•similar to <i>s01</i> and <i>s01r</i> but a bit larger structures</li> </ul>
<i>s07 s07r</i>	<ul style="list-style-type: none"> <li>•very large number of nodes</li> <li>•the widest hierarchies</li> <li>•the largest number of leaves</li> <li>•very low average number of instances per node</li> <li>•very high average number of children per node</li> </ul>

## SUMMARY OF THE PRESENTATION

- We've started research on methods generating a hierarchy of groups of objects, **providing a partial order relation between objects, not only clusters**
- The primary goal – clustering of images from a given domain, projection onto the ontology and inference in the ontology about the semantic meaning of the images
- We have developed
  1. A clustering method that meets 3 main conditions
  2. One measure of clustering quality assessment was adapted
  3. (a) Benchmark collections with their characteristics (available)
  3. (b) A benchmark data generator with instructions for use (available)
- Further work
  - Each of the above points requires further research, improvement or/and development of other approaches



Time for questions, but let me end, unusually,  
with my question to you:

Do you think that the topic is worth continuing  
in the era of rapidly developing deep models?