COMPUTATIONAL INTELLIGENCE

Associative Data Structures and Associative Neural Graphs





What's going on inside?



What biocybernetic secrets hides the brain and the whole nervous system?!







Internal associative representation of the data and fast associative processes allows the brain to quickly conclude and anticipate!

Brain – a biocybernetic structure forming knowledge and intelligence



Each brain can automatically and in the best-known way:

- memorize relations between data and form associations representing them,
- automatically form and broaden knowledge on the basis of the incoming data,
- remember various patterns and generalize about them,
- store important relations between data,
- work and recall facts in an associative way,
- easily use related data and information,
- quickly and context-sensitive recall adequate pieces of information,
- automatically recognize similarities and use them in thinking processes,
- transfer properties and behavior among similar objects,
- create new rules, methods, and algorithms based on the remembered ones.

Every event and experience of our lives is changing our brains to a certain extent, its way of working, and influencing future associations and actions! The dynamics and biocybernetic capabilities of our brains do not currently have a decent cybernetic equivalent or model in computational intelligence!

Brain – a dynamically changing biocybernetic structure



The brain is an unusual "computing" machine because it changes both its hardware and software as a result of the interaction with the data coming to it in the form of different stimuli. These changes concern:

- In the way of its further operation,
- In the process of further data processing,
- In its structure and properties of connections,
- In parameters of construction and functioning of neurons,
- In the previously memorized facts, rules, and objects,
- In the representation of various objects, actions, and phenomena,

In the way of associating and remembering facts, rules, and routines.
The brain allows us to memorize, but not everything and not permanently.

Definitions and ways of understanding different objects can grow, narrow, update, and even totally change throughout our lives.

The way it works is related to a nerve structure that allows it to act in an associative way and to selectively represent relations between data, objects, their groups, sequences, and classes.

BIOLOGICAL AND ARTIFICIAL NEURON



resting

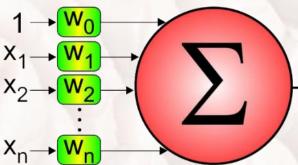
ILOSC JONOV

relaxing

Artificial neurons used today in computational intelligence are <u>very poor models of biological neurons</u>, distorting in their way of acting, plastic changes inside them, and the way of adaptation to incoming data:

charging

restina



Artificial neurons usually:

- 1. Are detached from the rest nervous system components as the senses and their receptors, cerebrospinal fluid, glial cells and their non-neglectable functions in the nervous system.
- Compute sums of weighted input signals without taking into account the automatic process of
 restoring insufficiently stimulated neurons and refracted neurons to the resting state after some time.

f(x)

- 3. Neglect and do not define or use their position in the network structure (except a few types of networks, e.g. SOM).
- 4. Diminish the significance of an activation threshold by bringing it to another weight with constant stimulation (bias), except spiking neurons.
- 5. Change the natural ability of neurons to be activated into continuous and differential activation functions.
- 6. Do not take into account the different and variable size of neurons that affect its sensitivity and specialization.
- 7. Bring down synapses to an adaptive balance that can amplify input signals many times (biologically not plausible).
- 8. Do not take into account various periods of various internal neuronal processes taking place in biological neurons.
- 9. Are mostly connected on the each-to-each basis between layers, which usually prevents them from specializing in the selected input groups.
- 10. Do not make any automatic connection or functional changes during adaptation (learning) process, bringing them to nonlinear functions that can be combined in the artificial neural network creating complex approximators.

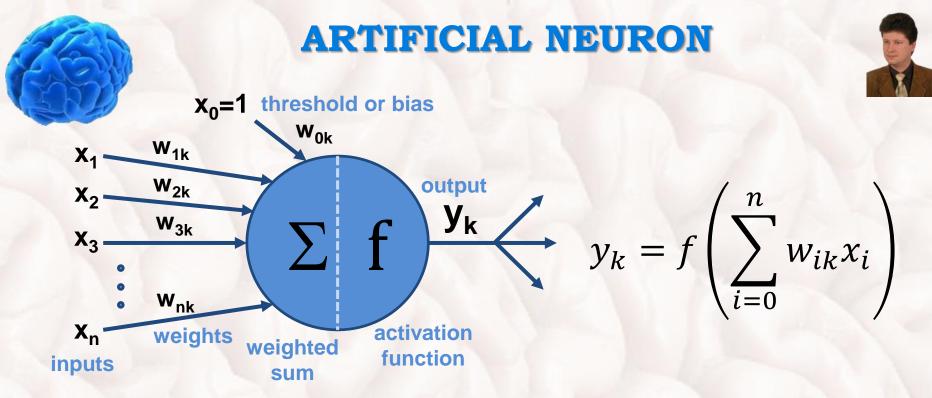


charging

relative refraction

activation

charging



- > Input data $x_1 \dots x_n$ typically simultaneously affect an artificial neuron.
- Previous states of artificial neurons have no influence on their current state, only current stimulation and weights w_{0k}, w_{1k},..., w_{nk} are taken into account.
- No temporal relationships between the states are considered.
- The response of an artificial neuron is immediate and calculated after an activation function which value depends on the weighted sum of current inputs x_{0k}, x_{1k},..., x_{nk}



THE SENSES AND RECEPTORS



- They provide the nervous system with the necessary stimuli for its functioning, development, and adaptation.
- They stimulate the neurons with certain combinations of input stimuli, which we call training patterns.
- The brain would not be able to develop without the senses and their receptors.
- The stimuli coming from the receptors form some stimulus combinations.
- Such combinations can be further associated and memorized.
- The created associations are used as a context for future associations and mental processes.





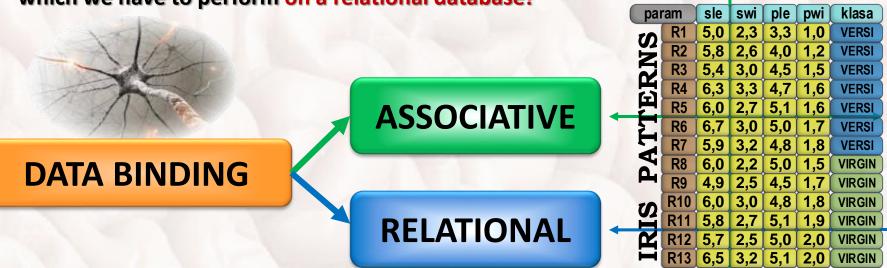
REASONS FOR THE ASSOCIATIVE REPRESENTATION OF DATA, OBJECTS, ACTIONS, AND FEATURES



Knowledge and intelligence allow us to quickly draw conclusions and make wise decisions thanks to the **associations** created and remembered in our minds.

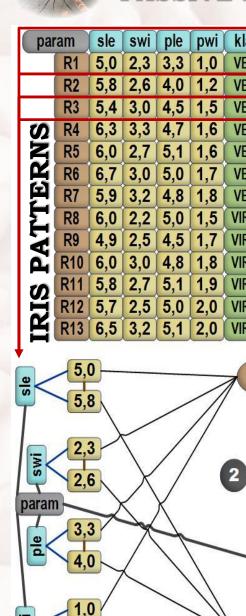
The associative data representation is much richer and gives us far more possibilities than the most commonly used relational representation used in relational databases:

- Relational databases allow us only for horizontal data binding thanks to the primary and foreign keys representing relations between objects.
- Associative systems allow us for both horizontal and vertical data binding combined with the aggregated representation of duplicates, which results in significant memory and computational time savings! Graph neural structures the with automatic vertical representation of data relationships <u>replace a lot of time-consuming operations</u>, which we have to perform on a relational database!



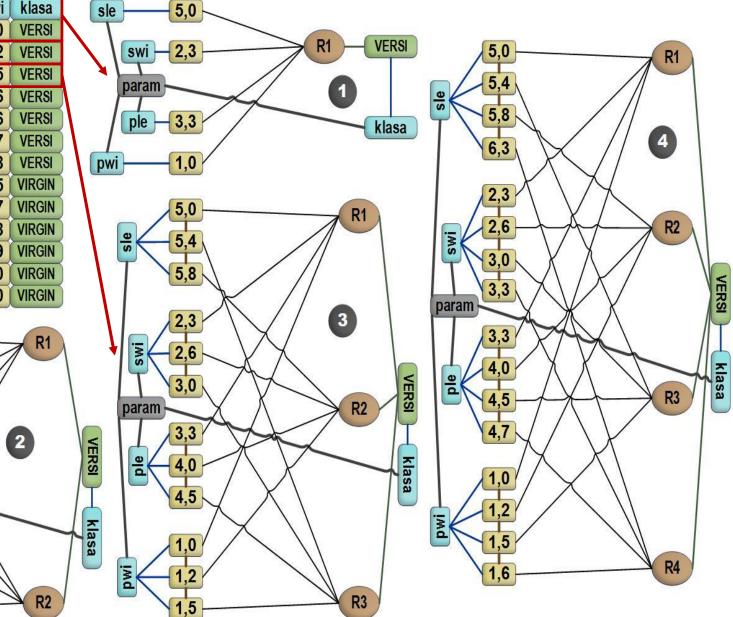
PASSIVE ASSOCIATIVE GRAPH DATA STRUCTURE - AGDS





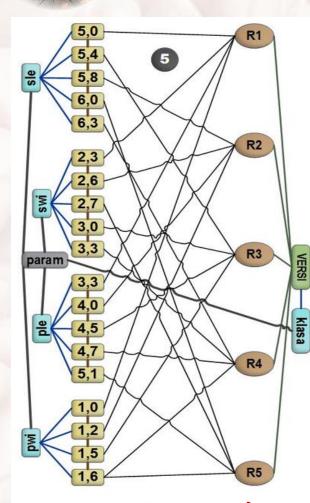
pwi

.2

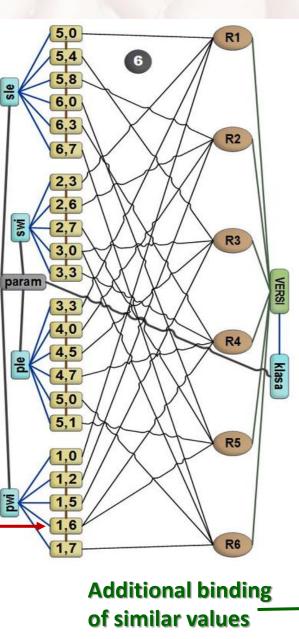


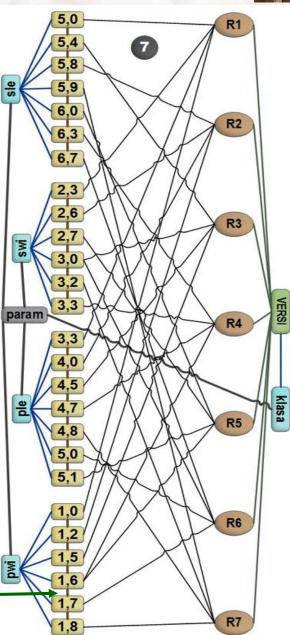
PASSIVE ASSOCIATIVE GRAPH DATA STRUCTURE - AGDS





Aggregated representation of duplicated values in table records (entities)



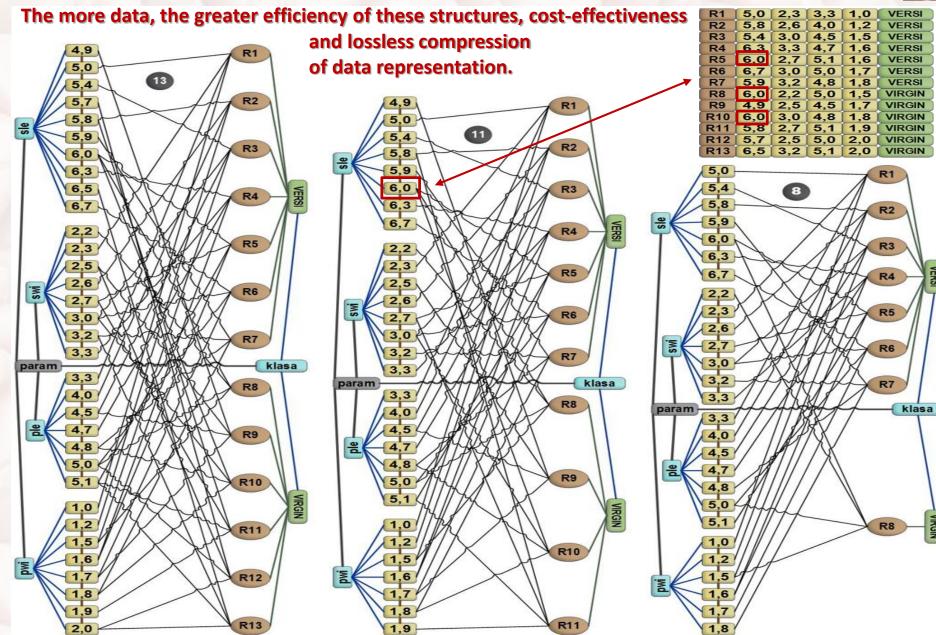


PASSIVE ASSOCIATIVE GRAPH DATA STRUCTURE - AGDS



VERSI

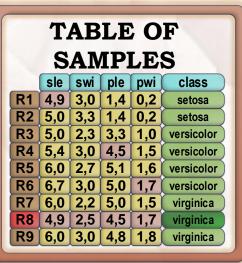
VIRGIN



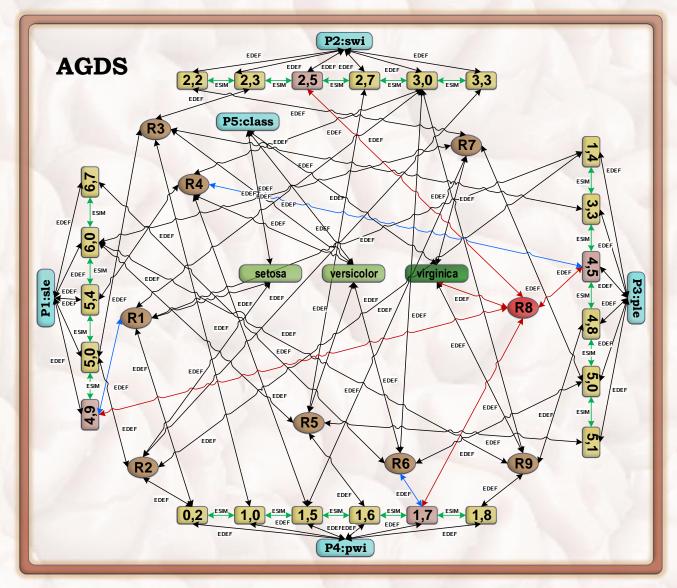


The connections point out related objects and similar data:

REPRESENTATION OF 9 IRIS SAMPLES IN TWO DATA STRUCTURES

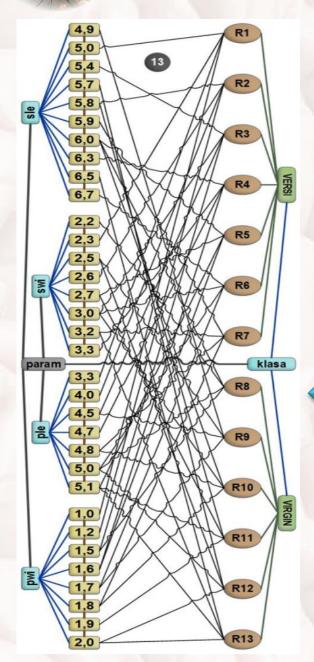


All similarities to other samples are immediately identified in the AGDS structure!



COMPARE STRUCTURES AND DRAW CONCLUSIONS





- What data relations can be simply read from these data structures and which must be found?
- What are the pros and cons of these structures?
- How do these structures affect the computational efficiency of operations on the stored data?
- Which structure is more suitable for efficient knowledge exploration and data mining?

TRANSFORMATION

pai	ram	sle	swi	ple	pwi	klasa
S	R1	5,0	2,3	3,3	1,0	VERSI
Z	R2	5,8	2,6	4,0	1,2	VERSI
	R3	5,4	3,0	4,5	1,5	VERSI
K	R 4	6,3	3,3	4,7	1,6	VERSI
E	R 5	6,0	2,7	5,1	1,6	VERSI
F	R6	6,7	3,0	5,0	1,7	VERSI
H	R7	5,9	3,2	4,8	1,8	VERSI
A	R 8	6,0	2,2	5,0	1,5	VIRGIN
Р.	R 9	4,9	2,5	4,5	1,7	VIRGIN
Ø	R10	6,0	3,0	4,8	1,8	VIRGIN
H	R11	5,8	2,7	5,1	1,9	VIRGIN
2	R12	5,7	2,5	5,0	2,0	VIRGIN
H	R13	6,5	3,2	5,1	2,0	VIRGIN



CONNECTION WEIGHTS IN THE AGDS STRUCTURES



The AGDS nodes representing neighboring (subsequent) values of each attribute a_k are connected and the weight of this connection (edge) is computed by the following formula:

$$w_{v_{i}^{a_{k}},v_{j}^{a_{k}}} = 1 - \frac{\left|v_{i}^{a_{k}} - v_{j}^{a_{k}}\right|}{r^{a_{k}}}$$

where

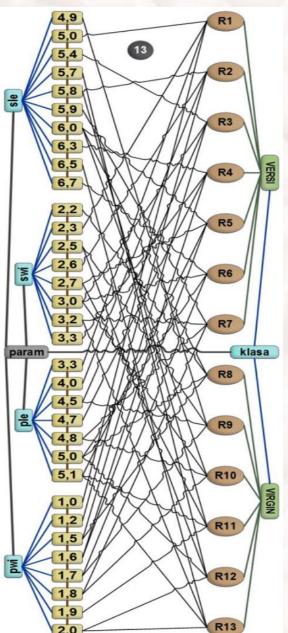
 $v_i^{a_k}$, $v_j^{a_k}$ - are values represented by the neighboring attribute nodes, which are connected by an edge in the AGDS graph,

 $r^{a_k} = v_{max}^{a_k} - v_{min}^{a_k}$ - is the current range of values of the attribute a_k . The weight of the connection from the value node $v_i^{a_k}$ of the attribute a_k to the object node R_m is determined after the number of occurrences $N_i^{a_k}$ of this value ($v_i^{a_k}$) in all objects:

$$w_{v_i^{a_k}, R_m} = \frac{1}{N_i^{a_k}} = \frac{1}{\|v_i^{a_k}\|}$$

These numbers $(N_i^{a_k} = ||v_i^{a_k}||)$ are stored in the individual value nodes of each attribute. This number is equal to the number or all connections of this value node to all object nodes if there are no duplicated objects in the table used to create the AGDS structure. In the opposite direction, the weights of connections from the object nodes to the value nodes are always equal to one:

$$W_{R_m,v_i^{a_k}} = 1$$



CREATION OF AGDS FOR A SINGLE DATABASE TABLE param ASSOCIATIVE param param **TRANSFORMATION** swi ple pwi class sle ple class sle swi pwi swi ple pwi class sle class ple pwi param sle SWI **R1** 5.0 2.3 3.3 1.0 VERSI 2.3 3.3 1.0 5.0 2.3 3.3 1.0 VERSI 4.9 2.2 3.3 1.0 VERSI VERSI 5.0 4.0 5.0 2.3 2.5 2.6 2.7 3.0 3.2 3.3 1.2 5.8 2.6 4.0 1.2 VIRGIN **R2** 5.8 2.6 4.0 1.2 VERSI **R2** 5.8 2.6 4.0 1.2 VERSI VERSI ATTERN 4.5 1.5 3.0 4.5 5.4 5.4 1.5 **R**3 4.5 1.5 VERSI 5.4 3.0 **R**3 5.4 3.0 4.5 1.5 VERSI VERSI 5.7 4.7 1.6 R4 3.3 4.7 1.6 6.3 3.3 4.7 1.6 VERSI 6.3 6.3 3.3 4.7 1.6 VERSI VERSI 4.8 5.8 1.7 **R**5 2.7 5.1 1.6 VERSI 5.1 1.6 2.7 5.1 1.6 6.0 6.0 R5 6.0 2.7 VERSI VERSI 5.0 5.1 5.9 1.8 **R6** 3.0 5.0 1.7 6.7 3.0 5.0 1.7 6.7 3.0 5.0 1.7 VERSI **R6** 6.7 VERSI VERSI 6.0 6.3 1.9 5.9 1.8 **R**7 1.8 VERSI 3.2 4.8 5.9 3.2 4.8 R7 4.8 1.8 VERSI VERSI 5.9 2.0 **R8** 6.0 2.2 5.0 1.5 6.0 2.2 5.0 1.5 VIRGIN 6.0 5.0 1.5 VIRGIN VIRGIN 2 6.5 2.5 **R**9 VIRGIN 4.9 4.5 1.7 4.9 2.5 4.5 1.7 **R**9 4.5 VIRGIN VIRGIN 4.9 2.5 1.7 3.0 4.8 1.8 3.0 4.8 1.8 6.7 VIRGIN 4.8 1.8 6.0 R10 6.0 6.0 3.0 VIRGIN VIRGIN IRIS 2.7 5.1 1.9 5.8 1.9 VIRGIN 5.8 2.7 5.1 5.1 1.9 VIRGIN VIRGIN IRI 2.5 2.0 R12 5.7 2.5 5.0 2.0 5.7 5.0 VIRGIN R12 5.7 2.5 5.0 2.0 VIRGIN VIRGIN R13 6.5 3.2 5.1 2.0 R13 6.5 3.2 5.1 2.0 3.2 5.1 2.0 VIRGIN VIRGIN 6.5 VIRGIN R4 R5 R6 R7 R8 R9 R10 R11 R12 R13 **R1** R2 R3 **IRIS PATTERNS** in the tree-based graph structure param AGDS ple class sle swi pwi

TTERNS

◄

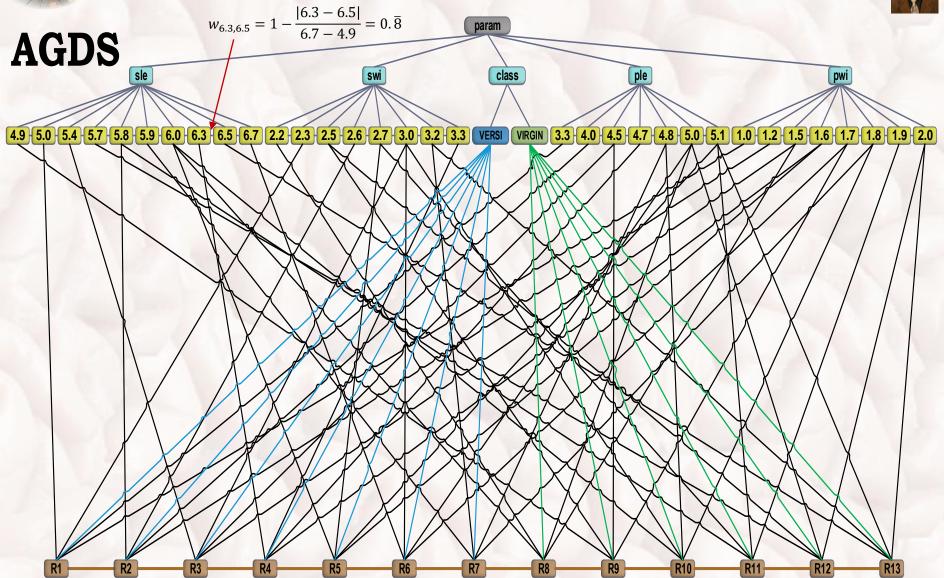
ሲ

R2 R4 R5 R6 **R**7 **R**8 R11 **R13 R**3 **R**9 R10 R12 All elements can be quickly accessed through the param root node that has connections to all parameters etc.

4.9 5.0 5.4 5.7 5.8 5.9 6.0 6.3 6.5 6.7 2.2 2.3 2.5 2.6 2.7 3.0 3.2 3.3 VERSI VIRGIN 3.3 4.0 4.5 4.7 4.8 5.0 5.1 1.0 1.2 1.5 1.6 1.7 1.8 1.9 2.0

THE TREE STRUCTURE USED IN PARALLEL COMPUTING

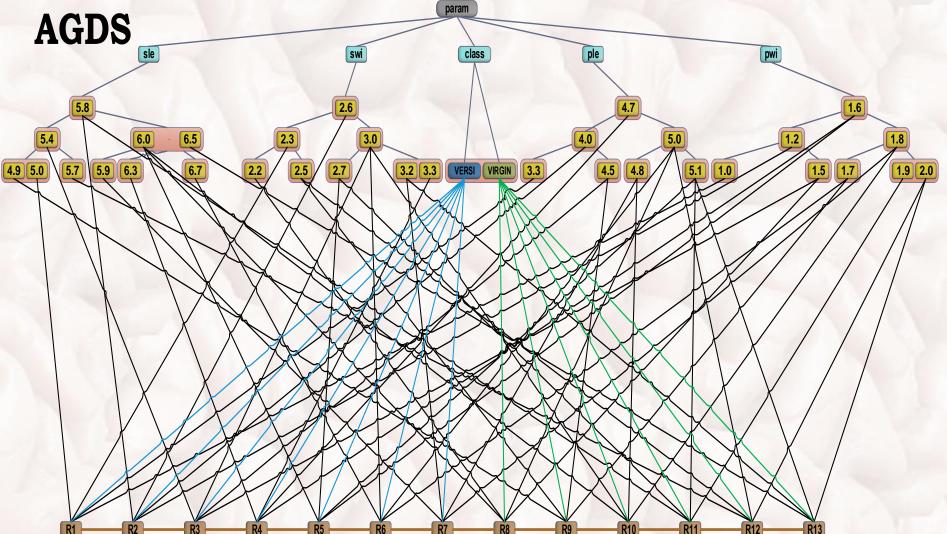




This tree-based graph gives you a very fast access to any data or relationships between these related and linked data. You can also draw various conclusions very fast.

ATTRIBUTE VALUE STRUCTURE IS BASED ON AVB-TREES

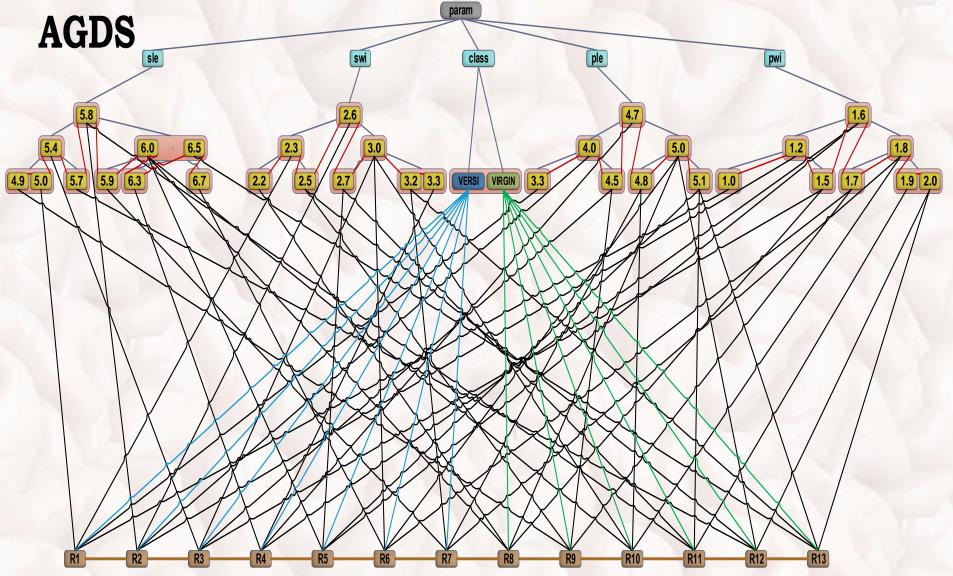




In the case of sequential (non-parallel) implementation of the AGDS structure, AVB-trees are used. The AVB-trees are the simple modification of B-trees, which aggregate representation of duplicates. The AVB-trees contain only unique attribute values for efficient access to them; duplicates are reduced.

ATTRIBUTE VALUE STRUCTURE IS BASED ON AVB+TREES





The subsequent values (keys of AVB+trees) can be additionally connected to reproduce proximity between represented unique keys, however, we can also use the AVB-tree structure to quickly find them.

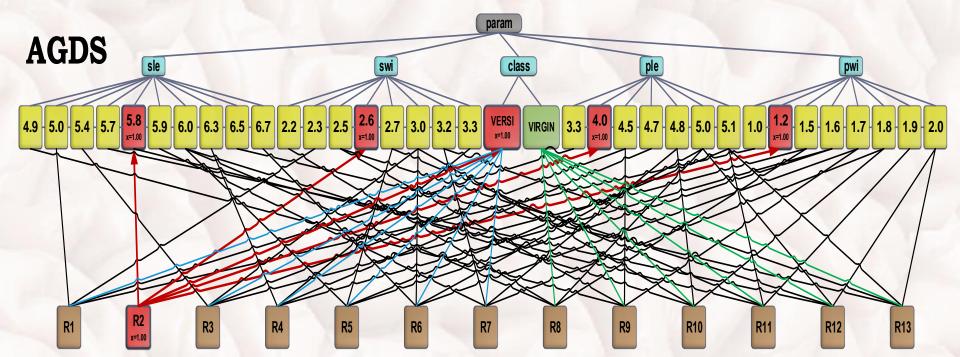


ASSOCIATIVE INFERENCE USING AGDS STRUCTURES



Associative data structures AGDS can be now used for associative inference, which is based on moving along the connections to the connected nodes and computing some values in these nodes on the basis of the send values multiplied by weights of these connections. In such a way we get the information about, e.g. similarity of objects represented by other nodes of the same kind or about the objects that satisfy some given conditions defined by the represented attribute values. Let's use our AGDS graph created for 13 Irises for such inference looking for objects (Irises) Rx which are most similar to R2.

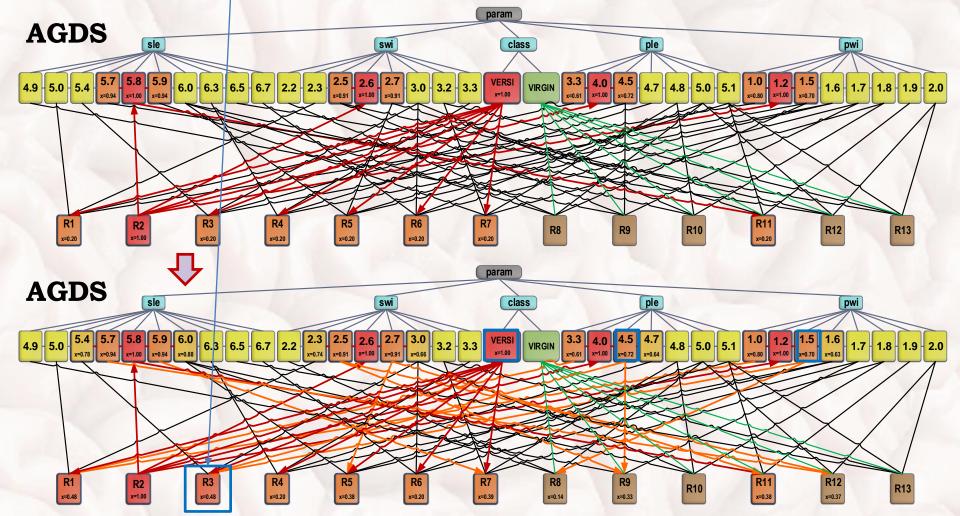
- 1. We start in the node R2 which assumes the similarity value x=1.0 because this node is 100% similar to itself.
- Next, we assign values x of the connected nodes representing the following values: 5.8, 2.6, VERSI, 4.0, and 1.2 by multiplying the value coming from the node R2 with the connection weights, which are equal 1.0. So, as a result, we achieve x=1.0 for all these connected nodes.



ASSOCIATIVE INFERENCE USING AGDS STRUCTURES



- 3. Subsequently, the values computed for these nodes are multiplied by next connection weights and send to the neighbor connected value nodes, for which we also compute their similarity values x.
- 4. Similarly, we compute the similarity values x for connected object nodes with regards to the necessity to add the passed weighted values to the sums already stored in these nodes, e.g. for the node R3 we compute x = 1.0 * 0.2 + 0.72 * 0.2 + 0.7 * 0.2 = 0.48



ASSOCIATIVE INFERENCE USING AGDS STRUCTURES



VERSI

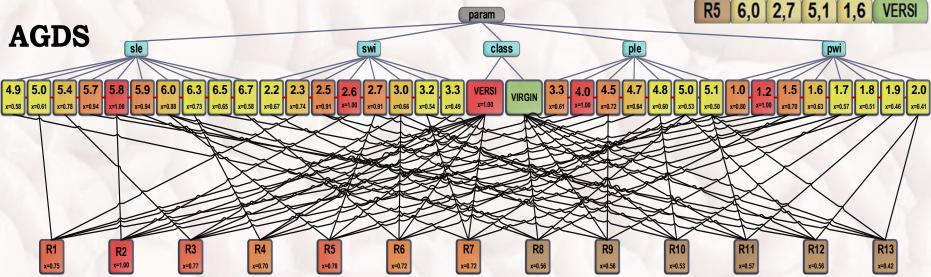
VERSI

VERSI

VERSI

1.2

5. Finally, when we go through all the connected (associated) values nodes computing theirs values of similarities by multiplying the sender similarity values by connection weights. We also computed weighted sums for all object nodes, where these weights are the same 5.0 2.3 3.3 1.0 **R1** w = 1/5 = 0.2. The computed similarity values for the nodes Rx can be 5.8 2,6 4,0 **R2** used to compare and designate the most similar objects to the object R2: **R**3 5,4 3,0 4,5 1.5 R5 (78%), R3 (77%), R1 (75%), ... 6,3 3,3 4,7 1,6 **R4**



It is also worth noting that AGDS graphs are not neural structures, so we are not obligated to multiply the nodes similarity values by connection weights, but we can also use other formulas, e.g. we can subtract the complement of the connection weight value from the similarity value represented by the sender: x' = x - (1 - w).

Consequently, we get another measure of similarity represented by the value nodes and object nodes. We can also use DASNG graph formulas to calculate weights between value nodes and object nodes to emphasize the rarity of the value using the frequency of connections coming out from value nodes: w = 1 / the number of outgoing connections.

X

♦ (4|5)

CONSTRUCTION OF B-TREES



B-trees are often used to created indices for attributes in relational databases. The construction of B-trees is a complex process that requires performing specific operations to restore assumptions and conditions: https://www.cs.usfca.edu/~galles/visualization/BTree.html

The addition of a new element to the B-tree consists of several steps:

- 1. Go from the root of the tree to one of its leaves after the following rules:
 - Go to the left if the key is less or equal to the left key value of the parent node,
 - Go to the right if the key is bigger than the right key value of the parent node,
 - Go in the middle if the key is bigger than the left key value of the parent node and less or equal to the right key value of the parent node.
- 2. When you get to the leaf, add the new key to it in order if it does not yet store two keys.
- 3. If it already contains two keys, divide this node into two nodes, leaving the smallest key in its left node, the biggest key in its right node, and pass the middle key to its parent node. If the parent node does not yet exist, create it. The parent node will be connected to these two child nodes.
- 4. If the parent node contains already two keys, the passed key is added in order and the parent node is also divided in the same way, creating two nodes and passing its middle key to its parent or creating it.

CONSTRUCTION OF B-TREE FOR THE LIST OF ELEMENTS

5

9

6

56

9

(56)

819

3

3

(5|6

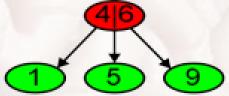
(5|6)

(1|3)

The operation of removal the keys from the B-TREE structure cannot violate the B-tree properties.

3

5



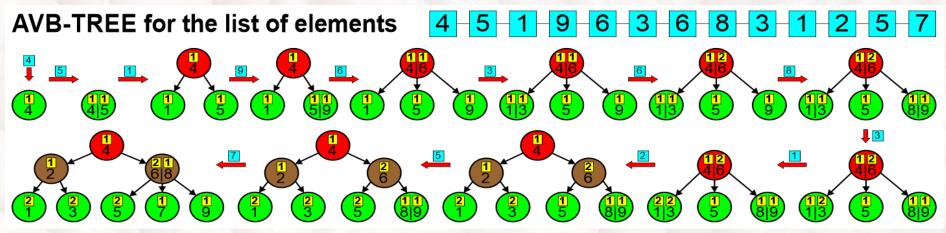


CONSTRUCTION OF AVB-TREES



AVB-trees are a simple modification of B-trees. AVB-trees aggregate the same (duplicated) values, represent them in a single node and count them up in order to know how many values have been aggregated to be able to remove the key representing several aggregated values correctly. Addition of a new key to the AVB-tree:

- 1. Go from the root of the tree to one of its leaves after the following rules:
 - Go to the left if the key is less to the left key value of the parent node,
 - Go to the right if the key is bigger than the right key value of the parent node,
 - Go in the middle if the key is bigger than the left key value and less than the right key value of the parent node.
 - Increment the counter of the left or right key of parent node if the added element is equal to one of them, and stop the
 descent process to the leaves.
- 2. When you get to the leaf, and the element is not equal to any key in it, add the new key to it in order if it does not yet store two keys.
- 3. If it already contains two keys, divide this node into two nodes, leaving the smallest key in its left node, the biggest key in its right node, and pass the middle key to its parent node. If the parent node does not yet exist, create it. The parent node will be connected to these two child nodes.
- 4. If the parent node contains already two keys, the passed key is added in order and the parent node is also divided in the same way, creating two nodes and passing its middle key to its parent or creating it.
- 5. If the leaf contains a key that is equal to the added element, increment its counter.





INSERTION OF THE KEY TO THE AVB-TREE



- 1. Start from the root and go recursively down along the edges to the descendants until the leaf is not achieved after the following rules:
 - a) Go to the left if the inserted key is less than the most left key in the node.
 - b) Go to the right if the inserted key is greater than the most right key in the node.
 - c) Go in the middle if the node contains two keys an the inserted key is greater than the left key and less then the right key.
 - d) Increment the counter of the key in the node that equals to the inserted keys.
- 2. When the leaf is achieved:
 - a) and if the inserted key is equal one of the keys in this leaf, increment the counter of this key.

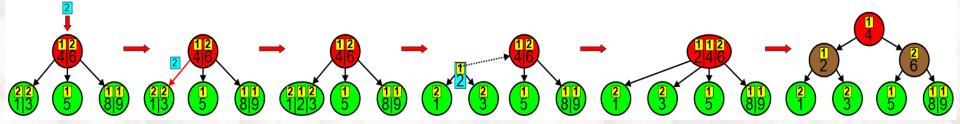
b) else insert the inserted key to the keys stored in this leaf in the increasing order, initialize its counter to one, and go to the step 3.

3. If the number of all keys stored in this leaf is greater then two, divide this node into two nodes: let the new left leaf represent the left (least) key together with its counter, the new right leaf represent the right (greatest) key together with its counter, and the middle key together with its counter and the pointers the new leaves pass to the parent node if it exists, and go to the step 4; if the parent node does not exist, create it (a new root of the AVB tree) and let it represent this middle key together with its counter and create new edges for the passed pointers to the new leaves.

4. Insert the passed key together with its counter to the key(s) represented in this node in the increasing order: if the key comes from the left branch, insert it on the left side of the key(s); if the key comes from the right branch, insert it on the right side of the key(s); if the key comes from the keys.

5. If the number of all keys stored in this node is equal to two, create two new edges for the passed pointers to the two divided nodes, where the edges are appropriately connected before and after the passed key in order instead of the edge that passed the key.

6. If the number of all keys stored in this node is greater then two, divide this node into two nodes: let the new left node represent the left (least) key together with its counter and connect the two leftmost edges to it; the new right node represent the right (greatest) key together with its counter and connect the two rightmost edges to it; and the middle key together with its counter and the pointers to the divided nodes pass to the parent node if it exists, and go to the step 4; if the parent node does not exist, create it (a new root of the AVB tree) and let it represent this middle key together with its counter the divided nodes.



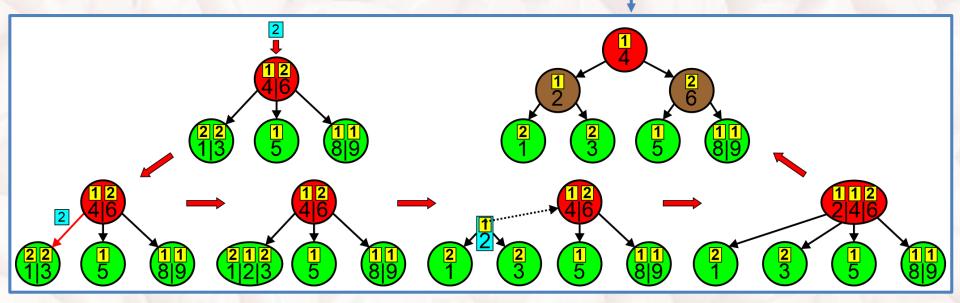


INTERMEDIATE STEPS OF PASSING THE MIDDLE KEYS



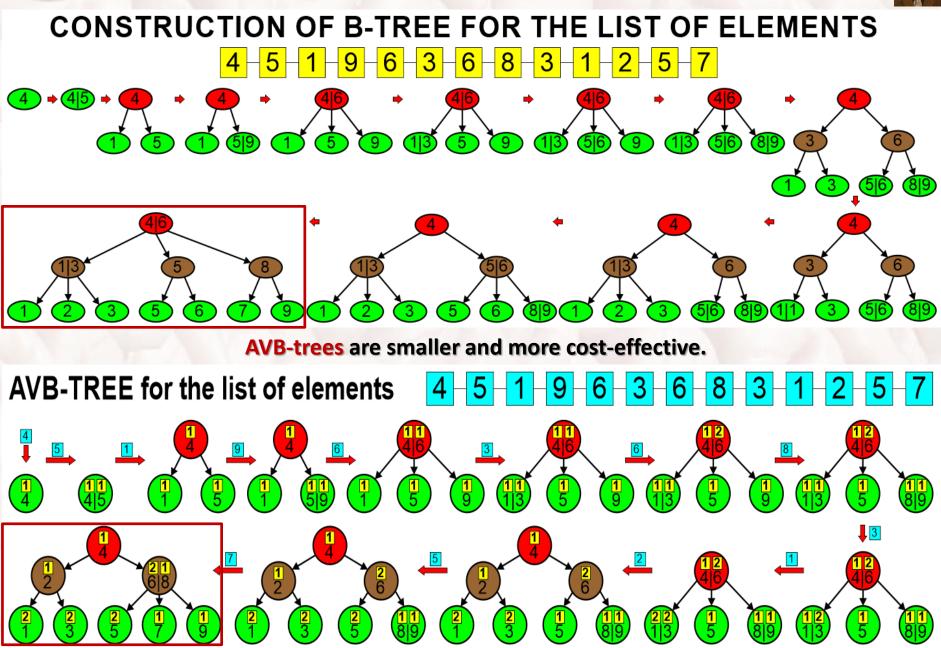
The intermediate steps of passing the middle key together with its counter and pointers to the new edges of the divided leaves/nodes to the parent node after the division of a leaf or a node or the creation of anew root.

AVB-TREE for the list of elements <mark>1</mark> 5 11 8|9 <mark>1</mark> 5 <mark>1</mark> 9 <mark>1</mark> 5 <u>1</u> 9 1 13 3 1 1 618 6 6 <mark>2</mark> 5 <mark>1</mark> 5 <mark>1</mark> 5 <mark>11</mark> 8|9, 19 <mark>11</mark> 8|9 <mark>11</mark> 8|9 <mark>22</mark> 1|3 <mark>1</mark> 5 <mark>11</mark> 8|9 2 2 12 13 23



COMPARISON OF THE B-TREES AND AVB-TREES





The AGDS for the IRIS data from ML Repository



Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka	Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka	Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka
1	Iris Setosa	5,1	3,5	1,4	0,2	51	Iris Versicolor	7,0	3,2	4,7	1,4	101	Iris Virginica	6,3	3,3	6,0	2,5
2	Iris Setosa	4,9	3,0	1,4	0,2	52	Iris Versicolor	6,4	3,2	4,5	1,5		Iris Virginica	5,8	2,7	5,1	1,9
з	Iris Setosa	4,7	3,2	1,3	0,2	53	Iris Versicolor	6,9	3,1	4,9	1,5	103		7,1	3,0	5,9	2,1
4	Iris Setosa	4,6	3,1	1,5	0,2	54	Iris Versicolor	5,5	2,3	4,0	1,3	104		6,3	2,9	5,6	1,8
5	Iris Setosa	5,0	3,6	1,4	0,2	55	Iris Versicolor	6,5	2,8	4,6	1,5	105	-	6,5	3,0	5,8	2,2
6	Iris Setosa	5,4	3,9	1,7	0,4	56	Iris Versicolor	5,7	2,8	4,5	1,3	106	Iris Virginica	7,6	3,0	6,6	2,1
7	Iris Setosa	4,6	3,4	1,4	0,3	57	Iris Versicolor	6,3	3,3	4,7	1,6	107	-	4,9	2,5	4,5	1,7
8	Iris Setosa	5,0	3,4	1,5	0,2	58	Iris Versicolor	4,9	2,4	3,3	1,0	108	Iris Virginica	7,3	2,9	6,3	1,8
9	Iris Setosa	4,4	2,9	1,4	0,2	59	Iris Versicolor	6,6	2,9	4,6	1,3	109	Iris Virginica	6,7	2,5	5,8	1,8
10	Iris Setosa	4,9	3,1	1,5	0,1	60	Iris Versicolor	5,2	2,7	3,9	1,4	110	-	7,2	3,6	6,1	2,5
11	Iris Setosa	5,4	3,7	1,5	0,2	61	Iris Versicolor	5,0	2,0	3,5	1,0	111		6,5	3,2	5,1	2,0
12	Iris Setosa	4.8	3,4	1,6	0,2	62	Iris Versicolor	5,9	3,0	4,2	1,5	112		6,4	2,7	5,3	1,9
13	Iris Setosa	4,8	3,0	1,4	0,1	63	Iris Versicolor	6,0	2,2	4,0	1,0	113		6,8	3,0	5,5	2,1
14	Iris Setosa	4,3	3,0	1,1	0,1	64	Iris Versicolor	6,1	2,9	4,7	1,4	114		5,7	2,5	5,0	2,0
15	Iris Setosa	5,8	4,0	1,2	0,2	65	Iris Versicolor	5,6	2,9	3,6	1,3	115		5,8	2,8	5,1	2,4
16	Iris Setosa	5,7	4,4	1,5	0,4	66	Iris Versicolor	6,7	3,1	4,4	1,3		Iris Virginica	6,4	3,2	5,3	2,3
17	Iris Setosa	5,4	3,9	1,3	0,4	67	Iris Versicolor	5,6	3,0	4,5	1,5	117		6,5	3,0	5,5	1,8
18	Iris Setosa	5,1	3,5	1,3	0,3	68	Iris Versicolor	5,8	2,7	4,1	1,0	118		7,7	3,8	6,7	2,2
19	Iris Setosa	5,7	3,8	1,7	0,3	69	Iris Versicolor	6,2	2,2	4,5	1,5	_	Iris Virginica	7,7	2,6	6,9	2,3
20	Iris Setosa	5,1	3,8	1,5	0,3	70	Iris Versicolor	5,6	2,5	3,9	1,1	120		6,0	2,3	5,0	1,5
21	Iris Setosa	5,4	3,4	1,7	0,2	71	Iris Versicolor	5,9	3,2	4,8	1,8	121		6,9	3,2	5,7	2,3
22	Iris Setosa	5,1	3,7	1,5	0,4	72	Iris Versicolor	6,1	2,8	4,0	1,3	122		5,6	2,8	4,9	2,0
23	Iris Setosa	4.6	3,6	1,0	0,2	73	Iris Versicolor	6,3	2,5	4,9	1,5	123		7,7	2,8	6,7	2,0
24	Iris Setosa	5,1	3,3	1,7	0,5	74	Iris Versicolor	6,1	2,5	4,7	1,2	124		6,3	2,3	4,9	1,8
25	Iris Setosa	4,8	3,4	1,9	0,2	75	Iris Versicolor	6,4	2,0	4,3	1,2	125		6,7	3,3	5,7	2,1
26	Iris Setosa	5,0	3,0	1,6	0,2	76	Iris Versicolor	6,6	3,0	4,4	1,3	125		7,2	3,2	6,0	1,8
27	Iris Setosa	5,0	3,4	1,6	0,2	77	Iris Versicolor	6,8	2,8	4,4	1,4	127		6,2	2,8	4,8	1,8
28	Iris Setosa	5,2	3,4	1,5	0,4	78	Iris Versicolor	6,7	3,0	5,0	1,4	127		6,1	3,0	4,0	1,8
20	Iris Setosa	5,2	3,3	1,5	0,2	79	Iris Versicolor	6,0	2,9	4,5	1,7	120		6,4	2,8	5,6	2,1
30		4,7	3,4	1,4	0,2	80	Iris Versicolor	5,7	2,5	3,5	1,0	130		7,2	3,0	5,8	1,6
31	Iris Setosa Iris Setosa	4,7	3,1	1,6	0,2	81	Iris Versicolor	5,5	2,0	3,8	1,0	131		7,2	2,8	6,1	1,0
32	Iris Setosa		3,1	1,5	0,2	82	Iris Versicolor	5,5	2,4	3,8	1,1	131		7,4	3,8	6,1	2,0
33	Iris Setosa	5,4			-	83	Iris Versicolor			-	-	132			-	-	2,0
34	Iris Setosa	5,2 5,5	4,1	1,5 1,4	0,1	84	Iris Versicolor	5,8	2,7	3,9 5,1	1,2		Iris Virginica Iris Virginica	6,4 6,3	2,8	5,6 5.1	1,5
35	Iris Setosa	4,9		1,4	0,2	85	Iris Versicolor	6,0	2,7		1,6	134			2,6	5,6	1,5
36	Iris Setosa	-	3,1			86		5,4		4,5	1,5	135		6,1 7,7	-	-	2,3
36		5,0 5,5	3,2 3,5	1,2	0,2		Iris Versicolor	6,0	3,4	-	1,6				3,0	6,1 5,6	
37	Iris Setosa	-		1,3	0,2	87 88	Iris Versicolor	6,7	3,1 2,3	4,7	1,5	137		6,3	3,4	-	2,4
38	Iris Setosa Iris Setosa	4,9	3,1		0,1	88	Iris Versicolor Iris Versicolor	6,3	-	4,4	-		Iris Virginica Iris Virginica	6,4 6,0	3,1 3,0	5,5 4,8	
			3,0	1,3	0,2			5,6	3,0	4,1	1,3						1,8
40	Iris Setosa	5,1	3,4	1,5	0,2	90	Iris Versicolor	5,5	2,5	4,0	1,3	140		6,9	3,1	5,4	2,1
41	Iris Setosa	5,0	3,5	1,3	0,3	91	Iris Versicolor	5,5	2,6	4,4	1,2	141		6,7	3,1	5,6	2,4
42	Iris Setosa	4,5	2,3	1,3	0,3	92	Iris Versicolor	6,1	3,0	4,6	1,4	142		6,9	3,1	5,1	2,3
43	Iris Setosa	4,4	3,2	1,3	0,2	93	Iris Versicolor	5,8	2,6	4,0	1,2	143		5,8	2,7	5,1	1,9
44	Iris Setosa	5,0	3,5	1,6	0,6	94	Iris Versicolor	5,0	2,3	3,3	1,0	144		6,8	3,2	5,9	2,3
45	Iris Setosa	5,1	3,8	1,9	0,4	95	Iris Versicolor	5,6	2,7	4,2	1,3	145		6,7	3,3	5,7	2,5
46	Iris Setosa	4,8	3,0	1,4	0,3	96	Iris Versicolor	5,7	3,0	4,2	1,2	146		6,7	3,0	5,2	2,3
47	Iris Setosa	5,1	3,8	1,6	0,2	97	Iris Versicolor	5,7	2,9	4,2	1,3	147		6,3	2,5	5,0	1,9
48	Iris Setosa	4,6	3,2	1,4	0,2	98	Iris Versicolor	6,2	2,9	4,3	1,3	148		6,5	3,0	5,2	2,0
49	Iris Setosa	5,3	3,7	1,5	0,2	99	Iris Versicolor	5,1	2,5	3,0	1,1		Iris Virginica	6,2	3,4	5,4	2,3
50	Iris Setosa	5,0	3,3	1,4	0,2	100	Iris Versicolor	5,7	2,8	4,1	1,3	150	Iris Virginica	5,9	3,0	5,1	1,8



ATTRIBUTE VALUES ON THE LEFT and OBJECTS ON THE RIGHT



P1.sepal length	/	P2. sepal width 📝	6	P3. petal length 📝		P4. petal width	V		C1.Iris-setosa		C2.Iris-versicolor	C3.1	tris-virginica
pe: ordered		Type: ordered		Type: ordered		Type: ordered			Similarity = 0,000		Similarity = 0,000	Sin	milarity = 0,00
pense: 1,00		Expense: 1,00		Expense: 1,00		Expense: 1,00			1 0	10	49[0		9910
alue:		Value:		Value:		Value:			210	1	5010		100[0
4,3[0	<u>n</u>	210		1 0	1	0,110	1		310	41	51 0	711-	101 0
4,510		2,210		1.1 0		0,2 0	-		4 0	ET.	52 0		102 0
4,410		2,3 0		1.2 0		0,210			510		5310	<u></u>	102 0
4,6 0	-	2,4 0	1	1,3 0	223	0,410			610		54[0	12/1-	104 0
4,7 0	-77-	2,5 0		1,4 0	1200	0,410	Attac	C ALL	7 0		5510		105 0
4,8 0		2,6 0	100	1,5 0	1992	0.6 0			8 0	1211	5610		106[0
4,9 0	100	2,7 0	DH-	1.6 0	1111	1 0	- ALLENGER		9 0	11-	57 0	12/-	107 0
5 0		2.8 0	5.44	1.7 0	1111 200	1,1 0	State State		10 0	1.1	58 0		108 0
5,1 0	->>>	2.9 0		1.9 0		1.2 0			11 0	2012	59]0		109 0
5,2 0		3 0		3 0	1-20.20	1,3 0		COR CARLOS	12 0	140	6010		110 0
5,3 0		3,1 0	1	3.3 0		1,510			13 0	-	61 0		111 0
5,4 0	-	3,210	1220	3,510	- 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1	1,5 0			14 0	100	6210		112 0
5,510	1200	3,310	12	3,6]0	1000	1,510		100000	14 0	the section	6310	-	112 0
5,6 0		3,310	Call Call	3,7 0		1,5 0	and the second second	the second second	15 0	-	6410		113 0
5,810		3,5 0	-	3,8 0	1000	1,7 0			15 0	42	6510		114 0
5,8 0	1	3,510	1000	3.9 0		1,8 0		112122	17 0		6610		115 0
		Name of Concession, Name o		4 0		Q-second statements			And and a second s	8.8	And and a second s		
5,9 0		3,7 0		the second se	1.1.1.1.1.1.1.1	2 0	- Alter and the second		19 0		67 0	×~-	117 0
6 0		3,8 0	5555	4,1 0	200	2.1 0	and the second second	And the second second	20 0		68 0		118 0
6.1 0		3,9 0		4.2 0		2,2 0			21 0		69 0		119 0
6.2 0	\sim	4 0		4.3 0		2,3 0			22 0	2	7010		120 0
6,3 0	- 200	4,1 0		4,4]0	1942-201	2,4 0	- CONTRACT N		23 0		71 0	>	121 0
6,4 0		4,2 0		4,5 0	S	2,5 0		South Street States	24 0		72 0		122 0
6,5 0		4,4 0		4,6]0	(C) (S)	2000	200 - 10 - 10 - 10 - 10 - 10 - 10 - 10 -	Chi mante	2510		7310	>	123[0
6,6 0			2	4,7 0		1	and the second	110 1425 - 3	26 0	1	74 0		124 0
6,7 0	->~			4,8 0	1000			Section 18 18	27 0		75[0		125 0
6,8 0	22			4.910		1990		2151161616	28 0	8200	76[0		126 0
6,9 0				5 0	X				29 0	242	77 0		127 0
7 0				5,1 0		202		1 1 m 2 m - 2 m - 2 m	30 0		78 0	~_	128 0
7,1 0	-			5.2[0			CU Sauce	S. S. S. C.	31 0		79[0		129 0
7,2 0		/		5,3 0					32 0		80 0		130 0
7,3 0				5,4 0			いちょうちょうちょう		3310	224	81 0		131 0
7,4 0				5,510		42		11111 14 16 16 16 16	34 0		8210		132 0
7,6 0				5,6]0	1	1 74	XXXXXXX	5. 1 65 W 1. 1. 2 F.	35 0	1000	8310		133 0
7,7 0				5,7 0	14	272		Section Ballie	3610		8410		134 0
7,9 0				5,8 0	7/ /	TA-			37 0	1000	85 0		135 0
				5,9 0	174	4			38 0	12-5-5-6	86[0		136 0
				6 0		1-1		4_1600000	39 0	20220	87 0		137 0
				6,1 0	/	11			40 0	11-1-1-2	88 0		138 0
				6,3]0		7/		0 - 1/148362	41 0		8910		139 0
				6,4 0	7/			1 11/11/11	4210	2000	90100		140 0
				6,6]0				I WAN	4310	14533	91 0		141 0
				6,7 0				11 111	4410	and an	9210		142 0
				6,9 0	r			IN NO	45 0	11111	9310		143 0
								11	4610	18000	9410		144 0
								1	47 0	141	95 0		145 0
								8	48 0	110.	96[0		146]0
										11	97[0		147 0
											98[0		



REMOVING OF REDUNDANCY (REDUCTION OF DUPLICATES)



P1.sepal length	P2.1	sepal width	P3. petal length 📝	P4. petal width 👿		CL.Iris-setosa	C2.Iris-versicolor	C3.tris-virginica	Lp. klasa	długość	szerokość			Lp. klasa		szerokość			Lp. klasa		szerokość		szerokość
Type: ordered		e: ordered	Type: ordered	Type: ordered		Similarity = 0,000	Similarity = 0,000	Similarity = 0,000	1 Iris Setos	5,1	liścia 3,5	płatka 1,4	płatka 0,2	51 Iris Versicolor	liścia 7,0	liścia 3,2	płatka 4,7	płatka 1,4	101 Iris Virginica	liścia 6,3	liścia 3,3	płatka 6,0	płatka 2,5
Expense: 1,00	Expe	ense: 1,00	Expense: 1,00	Expense: 1,00		10	49/0	9910	2 Iris Setos	4,9	3,0	1,4	0,2	52 Iris Versicolor	6,4	3,2	4,5	1,5	102 Iris Virginica	5,8	2,7	5,1	1,9
volue:	VOIV	ue:	value:	value:	1.	210	5010	100/0	3 Iris Setos	4,7	3,2	1,3	0,2	53 Iris Versicolor	6,9	3,1	4,9	1,5	103 Iris Virginica	7,1	3,0	5,9	2,1
4,3 0	< C	210	1 0	0,1 0		310	51/0	101/0	4 Iris Setos	4,6	3,1	1,5	0,2	54 Iris Versicolor	5,5	2,3	4,0	1,3	104 Iris Virginica	6,3	2,9	5,6	1,8
4,4 0		2,2 0	1.1/0	0,2 0		4 0	5210	102/0	5 Iris Setos	5,0	3,6	1,4	0,2	55 Iris Versicolor	6,5	2,8	4,6	1,5	105 Iris Virginica	6,5	3,0	5,8	2,2
45 0		23/0	12/0	0,3 0		5 0	53/0	103/0	6 Iris Setos	5,4	3,9	1,7	0,4	56 Iris Versicolor	5,7	2,8	4,5	1,3	106 Iris Virginica	7,6	3,0	6,6	2,1
4,6 0	16	24/0	S 1310	6,410	Contraction of the second	6 0	54/0	104 0	7 Iris Setos	4,6	3,4	1,4	0,3	57 Iris Versicolor	6,3	3,3	4,7	1,6	107 Iris Virginica	4,9	2,5	4,5	1,7
4,7 0	11	2.5 0	14/0	0,5 0		7 0	5510	105 0	8 Iris Setos	5,0	3,4	1,5	0,2	58 Iris Versicolor	4,9	2,4	3,3	1,0	108 Iris Virginica	7,3	2,9	6,3	1,8
4,8 0	SH-	2.6 0	15 0	0,6 0		8 0	5510	105 0	9 Iris Setos	4,4	2,9	1,4	0,2	59 Iris Versicolor	6,6	2,9	4,6	1,3	109 Iris Virginica	6,7	2,5	5,8	1,8
4.9 0	10	2.7 0	1.5 0	1 0 1.1 0		9 0 10 0	57/0	107/0	10 Iris Setos	4,9	3,1	1,5	0,1	60 Iris Versicolor	5,2	2,7	3,9	1,4	110 Iris Virginica	7,2	3,6	6,1	2,5
5.1 0	X-	2.9 0	1910	1.2 0		11/0	59 0	109/0	11 Iris Setos	5,4	3,7	1,5	0,2	61 Iris Versicolor	5,0	2,0	3,5	1,0	111 Iris Virginica	6,5	3,2	5,1	2,0
5.2 0	200-	310	10	1.3/0		1210	6010	110/0	12 Iris Setos	4,8	3,4	1,6	0,2	62 Iris Versicolor	5,9	3,0	4,2	1,5	112 Iris Virginica	6,4	2,7	5,3	1,9
5,3 0	111-	3.1/0	3,310	14 0		13/0	61/0	111/0	13 Iris Setos	4,8	3,0	1,4	0,1	63 Iris Versicolor	6,0	2,2	4,0 4,7	1,0	113 Iris Virginica	6,8	3,0	5,5	2,1
5,410	ET.	3.210	3.5 0	1.5 0		14 0	6210	112 0	14 Iris Setosi 15 Iris Setosi	4,3 5,8	3,0 4,0	1,1 1,2	0,1 0,2	64 Iris Versicolor 65 Iris Versicolor	6,1 5,6	2,9 2,9	4,7	1,4 1,3	114 Iris Virginica 115 Iris Virginica	5,7 5,8	2,5 2,8	5,0 5,1	2,0 2,4
5,5 0	17	3,3 0	3,6 0	1,6 0		15/0	63 0	113 0	16 Iris Setos	5,7	4,4	1,2	0,2	66 Iris Versicolor	6,7	3,1	4,4	1,5	116 Iris Virginica	6,4	3,2	5,3	2,4
5,6 0	R	3,4 0	3,7[0	1,7 0		16 0	5110	114 0	17 Iris Setos	5,4	3,9	1,3	0,4	67 Iris Versicolor	5,6	3,0	4,5	1,5	117 Iris Virginica	6,5	3,0	5,5	1,8
5,7 0	R.	3,5 0	3.8 0	1,8 0	Sec. 10 2 30	17 0	65 0	115 0	18 Iris Setos	5,1	3,5	1,4	0,3	68 Iris Versicolor	5,8	2,7	4,1	1,0	118 Iris Virginica	7,7	3,8	6,7	2,2
5,8 0	22	3.6 0	3,9 0	1,9 0	and the second	18 0	66 0	116 0	19 Iris Setos	5,7	3,8	1,7	0,3	69 Iris Versicolor	6,2	2,2	4,5	1,5	119 Iris Virginica	7,7	2,6	6,9	2,3
5.9 0	2	1.7 0	410	2/0	in the second second	19 0	67/0	117/0	20 Iris Setos	5,1	3,8	1,5	0,3	70 Iris Versicolor	5,6	2,5	3,9	1,1	120 Iris Virginica	6,0	2,2	5,0	1,5
6 0 6.1 0		3.910	4210	2.2/0	and the second second	21/0	69 0	119 0	21 Ins Setus	5,4	3,4	1,7	0,2	71 Iris Versicolor	5,9	3,2	4,8	1,8	121 Iris Virginica	6,9	3,2	5,7	2,3
6.210	5	410	4310	2310		2210	7010	120/0	22 Iris Setos	5,1	3,7	1,5	0,4	72 Iris Versicolor	6,1	2,8	4,0	1,3	122 Iris Virginica	5,6	2,8	4,9	2,0
6.3 0	0.41	41/0	4410	2410		23/0	710	> 121/0	22 Tric Cotoc	4.6	3,6	1.0	0.2	73 Iris Versicolor	6,3	2,5	4,9	1,5	123 Iris Virginica	7,7	2,8	6,7	2,0
6,4]0		42 0	4510	2.5 0	A BALLING COS	24/0	1210	122/0	24 Iris Setos	5,1	3,3	1,7	0,5	74 Iris Versicolor	6,1	2,8	4,7	1,2	124 Iris Virginica	6,3	2,7	4,9	1,8
6,5 0		4410	4610 🔀	STATES ST	Cart Cart South	25/0	73 0	> 123/0		1,0	2,1	1,0	0,2	To This Versionles	641 6.6	2,0	4,3	1,3		6,7	3,3	5,7	2,1
6,6 0	\gg	1983.	4710	Seres III		26 0	7410	124 0	26 Iris Setosi 27 Iris Setosi	5,0 5,0	3,0 3,4	1,6 1,6	0,2 0,4	76 Iris Versicolor 77 Iris Versicolor	6,6 6,8	3,0 2,8	4,4 4,8	1,4 1,4	126 Iris Virginica 127 Iris Virginica	7,2 6,2	3,2 2,8	6,0 4,8	1,8
6,7 0	S	X D	4310			27 0	75 0	125 0	28 Iris Setos	5,0	3,5	1,5	0,4	78 Iris Versicolor	6,7	3,0	4,0 5,0	1,4	127 Tris Virginica 128 Iris Virginica	6,1	3,0	4,0	1,8
6,8 0	××	X I	0164			28 0	7610	126 0	29 Iris Setos	5,2	3,4	1,4	0,2	79 Iris Versicolor	6,0	2,9	4,5	1,5	129 Iris Virginica	6,4	2,8	5,6	2,1
6.9 0 7 0	74	X	5.10	CLAR A		29 0	710	127/0	30 Iris Setos	4,7	3,2	1,6	0,2	80 Iris Versicolor	5,7	2,6	3,5	1,0	130 Iris Virginica	7,2	3,0	5,8	1,6
7,10	17	7 HS	5,210	5-9+33		3110	7910	129/0	31 Iris Setos	4,8	3,1	1,6	0,2	81 Iris Versicolor	5,5	2,4	3,8	1,1	131 Iris Virginica	7,4	2,8	6,1	1,9
7,2 0	4	7	5.310	UHR 23		32/0	8010	330/0	32 Iris Setos	5,4	3,4	1,5	0,4	82 Iris Versicolor	5,5	2,4	3,7	1,0	132 Iris Virginica	7,9	3,8	6,4	2,0
7,3 0	/ /		5.410	XXXX	CALLER MONTH	33/0	81/0	131/0	33 Iris Setos	5,2	4,1	1,5	0,1	83 Iris Versicolor	5,8	2,7	3 <mark>,</mark> 9	1,2	133 Iris Virginica	6,4	2,8	5,6	2,2
7,4 0	/	1	5,5 0			34 0	8210	112/0	34 Iris Setos	5,5	4,2	1,4	0,2	84 Iris Versicolor	6,0	2,7	5,1	1,6	134 Iris Virginica	6,3	2,8	5,1	1,5
7,6 0	1		5,6 0	ALS		35 0	8310	133/0	35 Iris Setos	4,9	3,1	1,5	0,1	85 Iris Versicolor	5,4	3,0	4,5	1,5	135 Iris Virginica	6,1	2,6	5,6	1,4
7,7 0	-		5,710	UT B	Section in such	3610	8410	134 0	36 Iris Setos	5,0	3,2	1,2	0,2	86 Iris Versicolor	6,0	3,4	4,5	1,6	136 Iris Virginica	7,7	3,0	6,1	2,3
7,9 0			5,8 0	I DA		37 0	8510	135 0	37 Iris Setosi 38 Iris Setosi	5,5 4,9	3,5 3,1	1,3 1,5	0,2 0,1	87 Iris Versicolor 88 Iris Versicolor	6,7 6,3	3,1 2,3	4,7 4,4	1,5 1,3	137 Iris Virginica 138 Iris Virginica	6,3 6,4	3,4 3,1	5,6 5,5	2,4
			5,9 0	A-621	ZSHAMMAR	38 0	8510	136 0	39 Iris Setos	4,9	3,0	1,3	0,1	89 Iris Versicolor	5,6	3,0	4,4	1,3	139 Iris Virginica	6,0	3,0	4,8	1,0
			6.10	412	<pre>></pre>	4010	8810	138/0	40 Iris Setos	5,1	3,4	1,5	0,2	90 Iris Versicolor	5,5	2,5	4,0	1,3	140 Iris Virginica	6,9	3,1	5,4	2,1
				11	11 21-2445303		8910	139/0	41 Iris Setos	5,0	3,5	1,3	0,3	91 Iris Versicolor	5,5	2,6	4,4	1,2	141 Iris Virginica	6,7	3,1	5,6	2,4
							0100	14010	42 Iris Setos	4,5	2,3	1,3	0,3	92 Iris Versicolor	6,1	3,0	4,6	1,4	142 Iris Virginica	6,9	3,1	5,1	2,3
		D	FNA		AL OF		91/0	14110	43 Iris Setos	4,4	3,2	1,3	0,2	93 Iris Versicolor	5,8	2,6	4,0	1,2	143 Iris Virginica	5,8	2,7	5,1	1,9
							92 0	142 0	44 Iris Setos	5,0	3,5	1,6	0,6	94 Iris Versicolor	5,0	2,3	3,3	1,0	144 Iris Virginica	6,8	3,2	5,9	2,3
							0 66	143 0	45 Iris Setos	5,1	3,8	1,9	0,4	95 Iris Versicolor	5,6	2,7	4,2	1,3	145 Iris Virginica	6,7	3,3	5,7	2,5
		D			ANCY		014	144 0	46 Iris Setos	4,8	3,0	1,4	0,3	96 Iris Versicolor	5,7	3,0	4,2	1,2	146 Iris Virginica	6,7	3,0	5,2	2,3
				INU	HILL		9510	145 0	47 Iris Setos	5,1	3,8	1,6	0,2	97 Iris Versicolor	5,7	2,9	4,2	1,3	147 Iris Virginica	6,3	2,5	5,0	1,9
							9510	14610	48 Iris Setos	4,6	3,2	1,4	0,2 0,2	98 Iris Versicolor	6,2 5,1	2,9	4,3	1,3	148 Iris Virginica	6,5 6,2	3,0	5,2	2,0
							3710	- And De	49 Iris Setosi 50 Iris Setosi	5,3 5,0	3,7 3,3	1,5 1,4	0,2	99 Iris Versicolor 100 Iris Versicolor	5,1	2,5 2,8	3,0 4,1	1,1 1,3	149 Iris Virginica 150 Iris Virginica	6,2 5,9	3,4 3,0	5,4 5,1	2,3
_									SU TIS SELOS	3,0	JJJ	1/4	VIZ	100 The versicolor	3,1	2,0	4/1	1,0	130 THS VIrginiCa	5,9	5,0	3/1	1,0



AGGREGATION AND REMOVAL OF REDUNDANCY IN AGDS



P1.sepal length 🚺	P2. sepal width 🕡 P3. petal length 🕡	P4. petal width [C1.Iris-setosa	C2.tris-versicolor	C3.tris-virginica	Lp. klasa		rokość długoś ścia płatka	ć szerokość	É Lp. klasa	długość liścia	szerokość długość liścia płatka	szerokoś płatka	ść Lp. klasa	długość liścia	szerokość dł liścia p		erokość ołatka
Type: ordered	Type: ordered Type: ordered	Type: ordered	Similarity n 0,00	Similarity = 0,000	Similarity = 0,000	1 Iris Setosa		3,5 1,4	0,2	51 Iris Versicolor	7.0	3,2 4,7	1,4	101 Iris Virginica		3,3		2,5
Expense: 1,00 Value:	Expense: 1.00 Expense: 1.00	Expense: 1,00 Value:	1]0	49[0	99 0	2 Iris Setosa		3,0 1,4	0,2	52 Iris Versicolor	6,4	3,2 4,5	1,5	102 Iris Virginica				1,9
teres.	Turks	Tant.	210	5010	100 0	3 Iris Setosa	4,7	3,2 1,3	0,2	53 Iris Versicolor	6,9	3,1 4,9	1,5	103 Iris Virginica	a 7,1	3,0	5,9	2,1
43 0	210 110	0,1 0	310	51/0	101 0	4 Iris Setosa	4,6	3,1 1,5	0,2	54 Iris Versicolor	5,5	2,3 4,0	1,3	104 Iris Virginica	6,3	2,9	5,6	1,8
44 0	2210 1110	0,2 0	410	52 0	102 0	5 Iris Setosa	5,0	3,6 1,4	0,2	55 Iris Versicolor	6,5	2,8 4,6	1,5	105 Iris Virginica	6,5	3,0	5,8	2,2
45 0	1310 1310	0,310	510	5310	103 0	6 Iris Setosa	5,4	3,9 1,7	0,4	56 Iris Versicolor	5,7	2,8 4,5	1,3	106 Iris Virginica		3,0	· .	2,1
46 0	2410 1310	0,410	610	510	104 0	7 Iris Setosa	· ·	3,4 1,4	0,3	57 Iris Versicolor	6,3	3,3 4,7	1,6	107 Iris Virginica				1,7
4.7 0	2.5 0 1.4 0 2.6 0 1.5 0	0.510	10	55 0	105 0	8 Iris Setosa	<u> </u>	3,4 1,5	0,2	58 Iris Versicolor	4,9	2,4 3,3	1,0	108 Iris Virginica		2,9	· •	1,8
4.9 0	27/0 15/0	10	910	5710	107/0	9 Iris Setosa	<u> </u>	2,9 1,4	0,2	59 Iris Versicolor	6,6	2,9 4,6	1,3	109 Iris Virginica		2,5	-	1,8
510	2810 2110 2170	Lilo	10/0	5810	10810	10 Iris Setosa 11 Iris Setosa		3,1 1,5 3,7 1,5	0,1	60 Iris Versicolor 61 Iris Versicolor	5,2 5,0	2,7 3,9 2,0 3,5	1,4 1,0	110 Iris Virginica 111 Iris Virginica	· ·			2,5
5.1 0	2.910 1.910	1.2 0	11/0	59 0	109 0	12 Iris Setosa		3,4 1,6	0,2	62 Iris Versicolor	5,0	3,0 4,2	1,0	112 Iris Virginica		2,7	· ·	1,9
5210	olt and all	1,3 0	12/0	6010	110/0	13 Iris Setosa	<u> </u>	3,0 1,4	0,1	63 Iris Versicolor	6,0	2,2 4,0	1,0	113 Iris Virginica				2,1
5,3 0	0162 0162	1,4 0	13/0	61 0	111/0	14 Tris Setosa		30 11	0,1	64 Iris Versicolor	6,1	2,9 4,7	1,4	114 Iris Virginica			· ·	2,0
5.410	1210 3510	1,510	14/0	6210	112/0	15 Iris Setosa	5,8	4,0 1,2	0,2	65 Iris Versicolor	5,6	2,9 3,6	1,3	115 Iris Virginica			· .	2,4
5.5 0		1,6 0	15/0	6310	113/0	16 Iris Setosa	5,7	4,4 1,5	0,4	66 Iris Versicolor	6,7	3,1 4,4	1,4	116 Iris Virginica	6,4	3,2	5,3	2,3
5,6 0	3.410 3.710	1,7 0	15/0	6110	114/0	17 Iris Setosa	5,4	3,9 1,3	0,4	67 Iris Versicolor	5,6	3,0 4,5	1,5	117 Iris Virginica	a 6,5	3,0	5,5	1,8
5.710	3.5 0 3.8 0 3.9 0	1.8 0	0/11	65 0	115 0	18 Iris Setosa		3,5 1,4	0,3	68 Iris Versicolor	5,8	2,7 4,1	1,0	118 Iris Virginica		3,8	· ·	2,2
5.90	3.710 410	2/0	190	67/0	117/0	19 Iris Setosa		3,8 1,7	0,3	69 Iris Versicolor	6,2	2,2 4,5	1,5	119 Iris Virginica		2,6	-	2,3
610	3810 4110	2.110	20 0	6810	118/0	20 Iris Setosa		3,8 1,5	0,3	70 Iris Versicolor	5,6	2,5 3,9	1,1	120 Iris Virginica			-	1,5
6.1 0	1.910 4.110	2.2 0	21/0	69 0	119 0	21 Iris Setosa		3,4 1,7	0,2	71 Iris Versicolor	-7-	3,2 4,8	1,8	121 Iris Virginica				2,3
62/0	410 4310	2.3 0	110	2010	130/0	22 Iris Setosa		3,7 1,5	0,4	72 Iris Versicolor	6,1	2,8 4,0	1,3	122 Iris Virginica				2,0
6310	41 0	2410	23/0	71 0	121/0	23 Iris Setosa 24 Iris Setosa		3,6 1,0 3,3 1,7	0,2 0,5	73 Iris Versicolor 74 Iris Versicolor	6,3 6.1	2,5 4,9 2,8 4,7	1,5 1,2	123 Iris Virginica 124 Iris Virginica		2,8 2,7	· .	2,0 1,8
64 0	4210 4510	2,5 0	24/0	1210	122 0	25 Iris Setosa	-1-	3,4 1,9	0,2	75 Iris Versicolor	6,4	2,0 4,7	1,2	125 Iris Virginica		3,3	· ·	2,1
6510	4410 4510		15/0	7310	123/0	26 Iris Setosa	<u> </u>	3,0 1,6	0,2	76 Iris Versicolor	6,6	3,0 4,4	1,4	126 Iris Virginica	· ·	3,2	· .	1,8
66 0	4710	1. 1480	25/0	7410	124/0	27 Iris Setosa	<u> </u>	3,4 1,6	0,4	77 Iris Versicolor	6,8	2,8 4,8	1,4	127 Iris Virginica		2,8	· .	1,8
6,7 0	4.10		23/0	75/0	125 0	28 Iris Setosa		3,5 1,5	0,2	78 Iris Versicolor	6,7	3,0 5,0	1,7	128 Iris Virginica		3,0		1,8
6.9 0	50		29/0	710	117/0	29 Iris Setosa	5,2	3,4 1,4	0,2	79 Iris Versicolor	6,0	2,9 4,5	1,5	129 Iris Virginica	6,4	2,8	5,6	2,1
7 0			30/0	7810	128 0	30 Iris Setosa	4,7	3,2 1,6	0,2	80 Iris Versicolor	5,7	2,6 3,5	1,0	130 Iris Virginica	7,2	3,0	5,8	1,6
7,1 0	5210		31/0	79 0	0 011	31 Iris Setosa	<u> </u>	3,1 1,6	0,2	81 Iris Versicolor	5,5	2,4 3,8	1,1	131 Iris Virginica	· ·	2,8	-	1,9
7,2 0	5.310	6110520	32/0	8010	130 0	32 Iris Setosa	<u> </u>	3,4 1,5	0,4	82 Iris Versicolor	5,5	2,4 3,7	1,0	132 Iris Virginica		3,8		2,0
7.3 0	5410	10.52	0110	81 0	131/0	33 Iris Setosa	- ·	4,1 1,5	0,1	83 Iris Versicolor	5,8	2,7 3,9	1,2	133 Iris Virginica				2,2
74 0	5510	H	14/0	8210	11510	34 Iris Setosa 35 Iris Setosa		4,2 1,4 3,1 1,5	0,2	84 Iris Versicolor 85 Iris Versicolor	6,0 5,4	2,7 5,1 3,0 4,5	1,6 1,5	134 Iris Virginica 135 Iris Virginica		· · ·	· .	1,5 1,4
7,6 0	5.610	ALA		810	133/0	36 Iris Setosa		3,2 1,2	0,1	86 Iris Versicolor	6,0	3,4 4,5	1,5	136 Iris Virginica				2,3
7,710	5/ U	UAT 19	3710	8510	134 0	37 Iris Setosa	<u> </u>	3,5 1,3	0,2	87 Iris Versicolor	6,7	3,1 4,7	1,5	137 Iris Virginica	· · ·	3,4		2,4
			3810	8510	13610	38 Iris Setosa	· ·	3,1 1,5	0,1	88 Iris Versicolor	6,3	2,3 4,4	1,3	138 Iris Virginica			· .	1,8
	AGGRE	<u>с</u> л.		87 0	137 0	39 Iris Setosa	4,4	3,0 1,3	0,2	89 Iris Versicolor	5,6	3,0 4,1	1,3	139 Iris Virginica	6,0	3,0	4,8	1,8
	AUGRE	JA		88/0	138 0	40 Iris Setosa	5,1	3,4 1,5	0,2	90 Iris Versicolor	5,5	2,5 4,0	1,3	140 Iris Virginica	6,9	3,1	5,4	2,1
			41 0	89 0	139 0	41 Iris Setosa	5,0	3,5 1,3	0,3	91 Iris Versicolor	5,5	2,6 4,4	1,2	141 Iris Virginica	6,7	3,1	5,6	2,4
		ъл/		0100	140 0	42 Iris Setosa		2,3 1,3	0,3	92 Iris Versicolor	6,1	3,0 4,6	1,4	142 Iris Virginica			· .	2,3
	AND RE			9110	141 0	43 Iris Setosa	<u> </u>	3,2 1,3	0,2	93 Iris Versicolor	5,8	2,6 4,0	1,2	143 Iris Virginica	-1-	2,7	-	1,9
			44(0	9210	14210	44 Iris Setosa	<u> </u>	3,5 1,6	0,6	94 Iris Versicolor	5,0	2,3 3,3	1,0	144 Iris Virginica	-7-	3,2		2,3
			45/0	0110	143/0	45 Iris Setosa	<u> </u>	3,8 1,9	0,4	95 Iris Versicolor	5,6	2,7 4,2	1,3	145 Iris Virginica		· · ·		2,5
	OF DUP			94 0	14410	46 Iris Setosa 47 Iris Setosa		3,0 1,4 3,8 1,6	0,3 0,2	96 Iris Versicolor 97 Iris Versicolor	5,7 5,7	3,0 4,2 2,9 4,2	1,2 1,3	146 Iris Virginica 147 Iris Virginica			· .	2,3 1,9
				9510	14610	47 Iris Setosa 48 Iris Setosa	<u> </u>	3,0 1,0 3,2 1,4	0,2	98 Iris Versicolor	6,2	2,9 4,2	1,3	147 Tris Virginica				2,0
				97[0	147/0	49 Iris Setosa	<u> </u>	3,7 1,5	0,2	99 Iris Versicolor	5,1	2,5 4,5	1,1	149 Iris Virginica				2,3
				98[0		50 Iris Setosa	<u> </u>	3,3 1,4	0,2	100 Iris Versicolor	5,7	2,8 4,1	1,3	150 Iris Virginica	5,9		· ·	1,8
													_		<u> </u>	<u></u>		



Pl.sepal length V

P2. sepal width 📝

P3. petal length 💡

P4. petal width 🗸

AGGREGATION OF DUPLICATED OBJECTS IN AGDS

długość szerokość długość szerokość

długość szerokość długość szerokość

C3.tris-virginica

C2.Iris-versicolor

C1.Iris-setosa



Table and a function of the second of the se	a liścia	a liścia	płatka płatka
Expense: 1.0 Expen	inica 6,3	3,3	6,0 2,5
Valie: Wale: Wale: Wale: Valie: 10 40 10 10 10 10 10 10 10 10 10 10 10 10 10	5,8	2,7	5,1 1,9
10 W/0 J W/0 J Ins Versicolor 6,9 3,1 4,9 1,5 103 Iris Ver	inica 7,1	3,0	5,9 2,1
410 10 10 410 10 410 10 10 10 10 10 10 10 10 10 10 10 10 1	inica 6,3	2,9	5,6 1,8
40 120 110 110 110 110 110 110 110 110 11	inica 6,5	3,0	5,8 2,2
430 2210 210 410 510 510 510 111 111 111 111 111 111 1	inica 7,6	3,0	6,6 2,1
460 240 140 40 40 40 40 10 10 10 10 10 10 10 10 10 10 10 10 10	inica 4,9	2,5	4,5 1,7
4/10 13/0 14/0 65/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1	inica 7,3	2,9	6,3 1,8
410 2610 1510 660 2,9 4,6 1,3 109 Iris Vir		2,5	5,8 1,8
4/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1			6,1 2,5
50 110 170 110 110 110 110 110 110 110 11	inica 6,5	3,2	5,1 2,0
SIID SIID <th< th=""><th></th><th></th><th>5,3 1,9</th></th<>			5,3 1,9
	inica 6,8	3,0	5,5 2,1
			5,0 2,0
			5,1 2,4
			5,3 2,3
	inica 6,5		5,5 1,8
	· ·		6,7 2,2
			6,9 2,3
			5,0 1,5
5,10 210 210 210 210 210 210 210 210 210 2	- ·		5,7 2,3
Line Line <thline< th=""> Line Line <thl< th=""><th></th><th></th><th>4,9 2,0</th></thl<></thline<>			4,9 2,0
Coll Coll <th< th=""><th></th><th>2,8</th><th>6,7 2,0</th></th<>		2,8	6,7 2,0
SAID SAID SAID MU <		2,7	4,9 1,8
KID KID <th></th> <th>· ·</th> <th>5,7 2,1</th>		· ·	5,7 2,1
660 NO		<u> </u>	6,0 1,8
67/0 16/0 17/1 17/1 17/1 17/1 17/1 17/1 17/1 17		·	4,8 1,8
SAID SAID THIS VERSION 6,7 3,0 5,0 1,7 128 Iris Version SAID		·	4,9 1,8
Sile Tile Tile <th< th=""><th>· · ·</th><th></th><th>5,6 2,1 5,8 1,6</th></th<>	· · ·		5,6 2,1 5,8 1,6
			6,1 1,9
			6,1 1,9
			5,6 2,2
			5,1 1,5
			5,6 1,4
		- <u>'</u>	6,1 2,3
DIF CONSOLIDATED H/I H/I <t< th=""><th></th><th>- · ·</th><th>5,6 2,4</th></t<>		- · ·	5,6 2,4
13 130		- · ·	5,5 1,8
me un 39 Iris Setora 4.4 3.0 1.3 0.2 89 Iris Versicolor 5.6 3.0 4.1 1.3 139 Iris Vir			4,8 1,8
			5,4 2,1
			5,6 2,4
900 400 42 Iris Setosa 4,5 2,3 1,3 0,3 92 Iris Versicolor 6,1 3,0 4,6 1,4 142 Iris Vir			5,1 2,3
	_	_	5,1 1,9
OF DUPLICATED 1/0 1/0 1/1 <	_		5,9 2,3
JIJO JIJO <th< th=""><th></th><th>-</th><th>5,7 2,5</th></th<>		-	5,7 2,5
M/0 M/0 M/0 A B 3,0 1,4 0,3 96 Iris Versicolor 5,7 3,0 4,2 1,2 146 Iris Versicolor	inica 6,7	3,0	5,2 2,3
OBJECTS Mile	inica 6,3	2,5	5,0 1,9
UDJEUID #10 #16 48 Iris Setosa 4,6 3,2 1,4 0,2 98 Iris Versicolor 6,2 2,9 4,3 1,3 148 Iris Vir		3,0	5,2 2,0
Still MUID 49 Iris Setosa 5,3 3,7 1,5 0,2 99 Iris Versicolor 5,1 2,5 3,0 1,1 149 Iris Versicolor	inica 6,2	3,4	5,4 2,3
310 50 Iris Setosa 5,0 3,3 1,4 0,2 100 Iris Versicolor 5,7 2,8 4,1 1,3 150 Iris Vir	inica 5,9	3,0	5,1 1,8

ELIMINATION OF REDUNDANCY IN AGDS REPRESENTATION



Manual Instate	83 martulate (20)	63 continent (1)	Re and aldeb		the second	en tels secolastes	en ede al adadada		JI		11				JL		JL				11		41	
P1.sepal length V	P2. sepal width 📝	P3. petal length	P4. petal width 🛛		C1.Iris-setosa	C2.tris-versicotor	C3.tris-virginica	Lp. klasa	liścia	szerokość liścia	płatka	szerokość płatka	Lp.	klasa	długosc liścia	szerokość liścia	długosc płatka	szerokość płatka	Lp. kla		Hugosc !! liścia	szerokość liścia	diugosc : platka	szerokosc płatka
Type: ordered	Type: ordered	Type: ordered	Type: ordered		Similarity = 0,000	Similarity = 0,000	Similarity = 0,000	1 Iris Setos		3,5	1,4	0,2	51	ris Versicolor	7,0	3,2	4,7	1,4	101 Iris Vi	_	6,3	3,3	6,0	2,5
Expense: 1,00 Value:	Expense: 1,00	Expense: 1,00	Expense: 1,00		110	49/0	9910	2 Iris Setos		3,0	1,4	0,2	52	ris Versicolor	6,4	3,2	4,5	1,5	102 Tris Vi	-	5,8	2,7	5,1	1,9
Value:	VOIUC:	Value:	Value:		2/0	5010	100/0	3 Iris Setos	<u> </u>	3,2	1,3	0,2	53	ris Versicolor	6,9	3,1	4,9	1,5	103 Iris Vi	-	7,1	3,0	5,9	2,1
43 0	210	1 0	0,1 0	Ala	310	5110	101/0	4 Tris Setos	<u> </u>	3,1	1,5	0,2	54	ris Versicolor	5,5	2,3	4,0	1,3	104 Iris Vi		6,3	2,9	5,6	1,8
4.4 0	2210	1.1/0	0,2 0		410	52 0	102 0	5 Iris Setos	_	3,6	1,4	0,2	55	ris Versicolor	6,5	2,8	4,6	1,5	105 Iris Vi	-	6,5	3,0	5,8	2,2
4.5 0	2310	1210	0,3/0	N HA	510	5310		6 Iris Setos		3,9	1,7	0,4	56	ris Versicolor	5,7	2,8	4,5	1,3	106 Iris Vi	-	7,6	3,0	6,6	2,1
4.6 0	2410	1.310	0,410		610	5410	104/0	7 Iris Setos		3,4	1,4	0,3	57	ris Versicoloi	6,3	3,3	4,7	1,6	107 Iris Vi	_	4,9	2,5	4,5	1,7
4,7 0	25/0	1410	0,510		710	55 0	105/0	8 Iris Setos		3.4	1.5	0,2	58	ris Versicolor	4,9	2,4	3,3	1,0	108 Iris Vi		7,3	2,9	6,3	1,8
4,8 0	2.610	15 0	0.610		8 0	24 10 0	106 0	9 Iris Setos	<u> </u>	2,9	1,4	0,2	59	ris Versicolor	6,6	2,9	4,6	1,3	109 Iris Vi	-	6,7	2,5	5,8	1,8
4.9 0	2.7/0	1.6 0	1 0	Call - Aller	910	57/0	107 0	10 ris Setos	<u> </u>	3,1	5	0,1	60	ris Versicoloi	5,2	2,7	3,9	1,4	110 Iris Vi	-	7,2	3,6	6,1	2,5
5 0	2.8/0	1.7/0	1.1 0	And the second second	10 0	58 0	108/0	11 Iris Setos		3,7	1,5	0,2	61	ris Versicolor	5,0	2,0	3,5	1,0	111 Iris Vi		6,5	3,2	5,1	2,0
5.1 0	2.910	1.9 0	1.2 0		11/0	59 0	109[0	12 Iris Setu	4,8	3,4	1,6	0,2	62	ris Versicoloi	5,9	3,0	4,2	1,5	112 Iris Vi	-	6,4	2,7	5,3	1,9
5,2 0	310	310	1,3 0		12/0	6010	110/0	13 Iris Setos		3,0	1,0	0,1	53	ris Versicolor	6,0	2,2	4,0	1,0	113 Iris Vi	-	6,8	3,0	5,5	2,1
5,3 0	3110	3,3 0	1,4 0		13/0	61/0	111]0	14 Iris Setos		3,0	1,1	0,1	64	Versicolor	6,1	2,2	4,7	1,0	114 Iris Vi	-	5,7	2,5	5,0	2,0
5,4 0	3210	3,5 0	1,510		14/0	62 0	112 0	15 Iris Setos		4,0	1,1	0,1	65	ris Versicolor	5,6	2,9	3,6	1,4	115 Iris Vi	_	5,8	2,5	5,0	2,0
5,5 0	3310	3,6 0	1,6 0		15/0	63 0	113 0	16 Iris Setos	<u> </u>	4,0	1,2	0,2	66	ris Versicolo	6,7	3,1	4,4	1,5	116 Iris Vi	-	6,4	3,2	5,3	2,4
5,6 0	3.4 0	3,7 0	1,7 0	a man a sea that a sea a s	16/0	6410		17 Iris Setos	· · ·	3,9	1,3	0,4	67	ris Versicolor	0,7 5 C	3,0	4,4	1,4	117 Iris Vi	-	6,5	3,2	5,5	1,8
5,7 0	3.5 0	3,8 0	1,8 0	Carl Constant	17 0	65 0	115/4	17 Tris Setos		3,9	1,5	0,4	68	ris Versicolor ris Versicolor	5,8	3,0 2,7	4,5	1,5	117 Tris Vi 118 Tris Vi	-	7,7	3,0	5,5 6,7	2,2
5,8 0	3.6 0	3,9 0	1,9 0		18/0	66 0	116/0	19 Iris Setos	<u> </u>	3,8	1,7	0,3	69	ris Versicoloi	6,2	2,2	4,1	1,5	119 Iris Vi	-	7,7	2,6	6,9	2,2
5.9 0	3.7 0	410	2 0		0 01	67 0	2> 117 0	20 Iris Setos	-	3,8	1,7	0,3	70	ris Versicoloi	5,6	2,2	4,5 2.9	1,5	120 Iris Vi	-	6,0	2,0	5,0	1,5
6 0	3.8/0	4.1]0	2.1 0		20/0	68 0	118 0	20 Iris Setos	-	3,4	1,5	0,5	70	ris Versicoloi	5,0	3,2	4,8	1,1	120 Iris Vi		6,9	3,2	5,7	2,3
6.1 0	1.9 0	42/0	2.2/0	and the second second	21/0	69 0	119 0	22 Iris Setos	<u> </u>	3,7	1,7	0,4	72	ris Versicolor	6,1	2,8	4,0	1,0	122 Iris Vi	-	5,6	2,8	4,9	2,0
6.2 0	40	4,3 0	2,3 0	1.191 1. 10 m	22/0	70/0	120 0	2 Iris Setos	<u> </u>	3,6	1,0	0,4	72	ris Versicolor	6,3	2,5	4,0	1,5	122 Iris Vi	-	7,7	2,8	6,7	2,0
6,3 0		4.4 0	2410	A STATE AND A STATE	23/0	71/0	121 0	24 Iris Setos	<u> </u>	3,3	1,0	0,2	74	ris Versicolor	6,1	2,5	4,5	1,5	124 Iris Vi	-	6,3	2,0	4,9	1,8
6,4 0	42/0	45 0	2,5 0	en na maria	24/0	12/0	122 0	25 Iris Setos	<u> </u>	3,3	1,7	0,5	74	ris Versicolor ris Versicolor	6,1	2,0	4,7	1,2	124 Tris Vi 125 Tris Vi	-	6,7	3,3	4,9 5,7	2,1
6,5 0	4410	4.610		and the state of the	25/0	7310	123/0	26 Iris Setos	· · ·	3,4	1,5	0,2	76	ris Versicolor	6,6	3,0	4,5	1,5	125 Iris Vi	-	7,2	3,2	6,0	1,8
6,6 0	$\rightarrow \rightarrow \rightarrow \sim$	4.710	Sec. 1		26 0	74/0	124 0	27 Iris Setos	· · ·	3,0	1,6	0,2	70	ris Versicolor	6,8	2,8	4,4	1,4	120 Iris Vi	-	6,2	2,8	4,8	1,0
6,7 0		4.310			27/0	75 0	125 0	28 Iris Setos	· ·	3,4	1,0	0,4	70	ris Versicolor	6,7	3,0	4,0 5,0	1,4	127 Iris Vi	-	6,1	3,0	4,0	1,0
6,8 0	ADAL)	0164			28 0	7610	126 0	29 Iris Setos		3,5	1,5	0,2	70	ris Versicoloi	6,0	2,9	4,5	1,7	120 Iris Vi	-	6,4	2,8	5,6	2,1
6.9 0	\sim	510	10.360		29 0	77/0	127 0	30 Iris Setos		3,4	1,4	0,2	79	ris Versicoloi	5,7	2,9	4,5 3,5	1,5	130 [ris Vi	-	7,2	3,0	5,8	1,6
7 0	1743	5,1 0	-4113)	26. 19 11 11 11 11	30 0	78/0	128 0	31 Iris Setos	-	3,1	1,6	0,2	81	ris Versicoloi	5,5	2,0	3,8	1,0	131 Iris Vi	-	7,2	2,8	6,1	1,0
7.1 0	14	5 5.210			31 0	79 0		32 Iris Setos	<u> </u>	3,4	1,5	0,2	82	ris Versicolor	5,5	2,4	3,7	1,1	132 Iris Vi	-	7,9	3,8	6,4	2,0
7,2 0	77 2	5.310	<u>MIN X</u>		32/0	80 0	130/0	33 Iris Setos		4,1	1,5	0,4	83	ris Versicoloi	5,5	2,4	3,9	1,0	133 [ris Vi		6,4	2,8	5,6	2,0
7,3 0		5.410	SEL		3310	81 0	131/0	34 Iris Setos		4,1	1,5	0,1	0.0	ris Versicolor	6,0	2,7	5,5	1,2	134 Iris Vi	_	6,3	2,8	5,0	1,5
7,4 0	7/	5510	HE	ANN Some Field	34 0	8210	13210	35 Iris Setos	<u> </u>	3,1	1,4	0,2	85	ris Versicolor	5,4	3,0	4,5	1,0	135 Iris Vi	-	6,1	2,6	5,6	1,5
7,6 0		5,610	ALA	A CONTRACTOR OF CONTRACTOR	3510	8310	13310	36 Iris Setos		3,2	1,2	0,1	86	ris Versicolor	6,0	3,4	4,5	1,5	136 Iris Vi	-	7,7	3,0	6,1	2,3
7,7 0		\$710	RTP	102.00 Mar 16012	3610	810	134 0	37 Iris Setos		3,5	1,2	0,2	87	ris Versicolor	6,7	3,1	4,7	1,5	137 Iris Vi	_	6,3	3,4	5,6	2,5
7,9 0		5,8 0	DA-	A WARRANT KS	37 0	8510	135 0	38 Iris Setos	<u> </u>	3,5	1,5	0,2	88	ris Versicolor	6,3	2,3	4,7	1,5	137 Iris Vi	-	6,4	3,4	5,5	1,8
		5,9 0	74/	ZSHALMAN HAR	(l)	85 0	136 0	39 Iris Setos		3,0	1,3	0,1	89	ris Versicoloi	5,6	3,0	4,4	1,3	139 [ris Vi	-	6,0	3,0	4,8	1,0
		610	7-12		39 0	8710		40 Iris Setos	<u> </u>	3,0	1,5	0,2	90	ris Versicoloi	5,6	2,5	4,1	1,5	139 Iris Vi 140 Iris Vi	-	6,9	3,1	5,4	2,1
		6.1 0	A	11 11 - 4434834	410	8810	138 0	40 Iris Setos	· · ·	3,4	1,5	0,2	91	ris Versicoloi	5,5	2,5	4,0	1,5	140 Iris Vi	-	6,7	3,1	5,6	2,1
		0.010	2111	111 76 11119024076	ALL PAL	9010	14010	41 Iris Setos		2,3	1,3	0,3	92	ris Versicoloi	5,5 6,1	3,0	4,4	1,2	141 Iris Vi 142 Iris Vi	-	6,9	3,1	5,0	2,4
		DEE			71/	910	14010	43 Iris Setos		3,2	1,3	0,3	92	ris Versicolor	5,8	2,6	4,0	1,4	142 Iris Vi 143 Iris Vi		5,8	2,7	5,1	1,9
		KEL		NDANC	Y	92/0	14210	44 Iris Setos		3,2	1,5	0,2	95	ris Versicolor ris Versicolor	5,0	2,0	3,3	1,2	143 Iris Vi 144 Iris Vi	_	5,0 6,8	3,2	5,1	2,3
						9310	14310	45 Iris Setos	· ·	3,5	1,0	0,6	94	ris Versicolor ris Versicolor	5,0	2,5	3,3 4,2	1,0	144 Tris Vi 145 Tris Vi	-	6,7	3,2	5,9	2,5
						94(0	14310	45 Tris Setos	<u> </u>	3,8	1,9	0,4	95	ris Versicolor ris Versicolor	5,6 5,7	3,0	4,2	1,3	145 Tris Vi 146 Tris Vi	-	6,7	3,3	5,7	2,5
					C	95 0	145 0	40 Iris Setos	· ·	3,0	1,4	0,3	90	ris Versicolor ris Versicolor	5,7	2,9	4,2	1,2	140 Iris Vi 147 Iris Vi	_	6,3	2,5	5,2	1,9
	AN	$\boldsymbol{\nu}$	UP	LICAIE	3	96[0	14510	47 Tris Setos	<u> </u>	'	1,6	0,2	0.0	ris Versicoloi ris Versicoloi		,	4,2	1,3		-	6,5 6,5	2,5	5,0	2,0
						97/0	147 0	48 Tris Setos	<u> </u>	3,2 3,7	1,4		90		6,2	2,9 2,5	4,3 3,0	· ·		-	6,5	,		2,0
						98 0	And be		a 5,3 a 5,0	3,7	1,5	0,2	100	ris Versicoloı ris Versicoloı	5,1 5,7	2,5	3,0 4,1	1,1 1,3		-	5,9	3,4 3,0	5,4	1,8
						and a		50 Iris Setos	a 5,0	აკა	1,4	0,2	100	ris versicoloi	5,7	2,8	4,1	1,5	150 Iris Vi	rginica	5,9	3,0	5,1	1,0



FAST SEARCH FOR RELATIONS AND CORRELATIONS



1.sepal length 📝		P2. sepal width	0	P3. petal length 📝		P4. petal width 📝		C1.Iris-setosa		C2.Iris-versicolor		C3.Iris-virginica
pe: ordered		Type: ordered		Type: ordered		Type: ordered		outCR=0/0		outCR=0/0]	outCR=0/0
pense: 1,00		Expense: 1,00		Expense: 1,00		Expense: 1,00		11-68-0		401-00-0		001-010-010-
ilue:		Value:		Value:		Value:		1 oCR=0		49]oCR=1		99 oCR=2/1x
4,3joutCorr=0		2 outCorr=0		1 outCorr=0	6	0,1 outCorr=0		2/oCR-1		50 oCR=2 51 oCR=2		100 oCR=2/10 101 oCR=1/10
4,3 outCorr=0 4,4 outCorr=0		2 outCorr=0		1,1 outCorr=0		0,2 outCorr=0		3[oCR=1 4[oCR=1	41	Committee and the second scheme and the	1	101 oCR=1/15 102 oCR=1/35
							A	4/0CR=1		52 0CR=0		
4,5 outCorr=0		2,3 outCorr=0		1,2 outCorr=0		0,3 outCorr=0			7	= 53 oCR=2		103 oCR=1/2
4,6 outCorr-0		2,4 outCorr=0		1,3 outCorr=0		0,4 outCorr=0		6 oCR-0		= 54 oCR-1		104 oCR-1/10
4,7 outCorr=0		2,5 outCorr=2	1	1,4 outCorr=0		0,5 outCorr=0		7/oCR=1		55 oCR=2		105 oCR=1/4x
4,8 outCorr-0		2,6 outCorr=1		1,5 outCorr=0		0,6 outCorr=0		B oCR-1	-27	56 oCR-1		106 oCR-1/25
4,9 outCorr=1		2,7 outCorr=2	130	1,6 outCorr=0		1 outCorr=0		9 oCR=1		57 oCR=1		107 oCR=1/3x
5 outCorr=0		2,8 outCorr=2		1,7 outCorr=0		1,1 outCorr=0	Contract 14	10 oCR-1		58 oCR=1		108 oCR-1/1x
5,1 outCorr=0		2,9 outCorr=1	62	1,9 outCorr=0		1,2 outCorr=0		11 oCR=0		59 oCR=0	-	109 oCR=1/3x
5,2 outCorr=0		3 outCorr=2	1000	3 outCorr=0	1350	1,3 outCorr=0	Charles A	12 oCR=1	_	60 oCR=2	1	110 oCR-1/2x
5,3 outCorr=0		3,1 outCorr=2	1.12	3,3 outCorr=0	-	1,4 outCorr=2		13 oCR=1		61 oCR=2	-	111 oCR-1/2
5,4 outCorr=0		3,2 outCorr=2		3,5 outCorr=0	100	1,5 outCorr=2		14 oCR=1		62 oCR=2	2	112 oCR=1/3x
5,5 outCorr=0		3,3 outCorr=2	-888	3,6 outCorr=0		1,6 outCorr=1		15 oCR=1		63 oCR=1	~	113 oCR=1/3
5,6 outCorr=1		3,4 outCorr=1	12210	3,7 outCorr=0		1,7 outCorr=1		16 oCR=1		= 64]oCR=2	0	114 oCR=1/20
5,7 outCorr=1	CH.	3,5 outCorr=0		3,8 outCorr=0	6025	1,8 outCorr=2	COLOR COLOR	17 oCR=0		65 oCR-1		115 oCR-1/30
5,8 outCorr=2	0	3,6 outCorr=1		3,9 outCorr=0		1,9 outCorr=0		18joCR=0	200	> 661oCR=2	~	116 oCR=1/13
5,9 outCorr=2	0	3,7 outCorr=0		4 outCorr=0	1996	2 outCorr=0		19 oCR-1		- 67 oCR=2		117[oCR-1/1x
6 outCorr=2	<	3,8 outCorr=1	$\langle \rangle$	4,1joutCorr=0	Ser 15	2,1 outCorr=0		20 oCR=1		68 oCR=1		118 oCR=1/4x
6,1 outCorr~2		3,9 outCorr=0		4,2 outCorr=0	6 AM	2,2 outCorr=0		21 oCR-1		69 oCR-2	2	119 oCR-1/2
6,2 outCorr=1		4 outCorr=0		4,3 outCorr=0	aller a	2,3 outCorr=0		22 oCR=0		= 70 oCR=1	2	120 oCR=1/3a
6,3 outCorr=2		4,1 outCorr-0		4,4 outCorr=0	A COL	2,4 outCorr=0	approx 1742	23 oCR-1	26502	711oCR-2		121 oCR=1/1x
6,4 outCorr=2		4,2 outCorr=0		4,5 outCorr=1		2,5 outCorr=0		24 oCR=1		72 oCR=1		122 oCR=2/1x
6,5 outCorr=1		4,4 outCorr-0	500	4,6 outCorr=0	22	A B		25 oCR-1		73 oCR-1		123 oCR-1/2x
6,6 outCorr=0	>			4,7 outCorr=0	SY	A CA	PH STATES CON	26 0CR=1		74 oCR=1		124 oCR=2/1x
6,7 outCorr=2			00	4,8joutCorr-2	BXC?			27 oCR-1		75 oCR-2		125 oCR=2/2x
6,8 outCorr=1				4,9 outCorr=2		8000	A State March	28]oCR=0	200	76[oCR=2		126 oCR=1/4x
6,9 outCorr=2	$\langle \rangle$			5 outCorr=1		all to	Call Call and Call	29 oCR=1		77 oCR=2		127 oCR=1/2x
7 outCorr=0				5,1 outCorr=2		4422	1201241121	30 0CR=1		78 oCR=1		128 oCR=1/2x
7,1 outCorr=0				5,2 outCorr=0		1112	a all all the	31 oCR=1		79 oCR=0	-	129 oCR-1/1x
7,2 outCorr=0				5,3 outCorr=0			ALL AND I DA	32 oCR=1	2634	80]oCR=0		130[oCR=1/1x
7,3 outCorr=0				5,4 outCorr=0		- 10-	1911 IN HEARD	33joCR=0	22	81 oCR=2	200	131 oCR=1/2x
7,4 outCorr=0				5,5 outCorr=0	1		1112A+11 212A	34/oCR=0	BCK	82 oCR=2		132 oCR=1/4x
7,6 outCorr=0				5,6 outCorr=0	1-2		VIII. LET TO LE	35 oCR=1		83]oCR=1	1	133 oCR=2/1x
7,7]outCorr=0				5,7 outCorr=0	44		ANT USAN	36 oCR=0		84 oCR=1	2	134 oCR-1/10
7,9 outCorr=0				5,8 outCorr=0	17		11.11.62.62.11	37 oCR=1		851oCR=2	210-	135 oCR=1/25
				5,9 outCorr=0	14		111000	38 0CR-1	2005	86 oCR-1	200	136 oCR-1/3
				6 outCorr=0	17		111 11	39 oCR=0	4116	87 oCR=1		137 oCR=2/1a
				6,1 outCorr=0	1			40 oCR=0	H2	88 oCR-1	00	138 oCR-1/2
				6,3 outCorr=0				41/oCR=1	200	89 oCR=1		139 oCR=2/1x
				6,4 outCorr=0	10			42 oCR-0		901oCR-2		140 oCR=1/3
				6,6 outCorr=0				43 oCR=1		91 oCR=1	10	141 oCR=1/2
					4			aninetter.	1000	and and the state of the state		
								44/of R-1	Charles -	921oCB=0	760	1421oCR=1/2
				6,7 outCorr=0				44 oCR-1		92 oCR=0	22	the second s
								44 oCR-1 45 oCR=1 46 oCR=1		92 oCR=0 93 oCR=1 94 oCR=1		142 oCR=1/2x 143 oCR=1/2x 144 oCR=1/3x

48 oCR=1

96|oCR=1

97|oCR=1

98[oCR=1

146|oCR=1/2x

147 |oCR=2/1x

FAST CORRELATION SEARCH



FAST SEARCH FOR RELATIONS BETWEEN OBJECTS



98 0,57277

1.sepal length 🔽 ype: ordered		P2. sepal width Type: ordered	×	P3. petal length		P4. petal width Type: ordered	N N	C1.Iris-setosa	and the second se	versicolor	C	3.Iris-virginica
			-	Type: ordered				Similarity = 0,413	Simil	arity = 0,823		Similarity - 1,0
opense: 1,00 alue:		Expense: 1,00 Value:		Expense: 1,00 Value:		Expense: 1,00 Value:	_	1 0,38479	49	10,76885		99[0,76162
ive.		value.		value.		value.		210,28599		0,71120		10010,75814
4,3 0,19516		210,35872		1/0,17394		0,110,22965		310,31670	51	10,78886	12	101 0,88849
4,410,20648		2,2 0,42691		1,1 0,17999		0,210,25006		410,29183	52	0,51405	14	102 0,84262
4,5 0,21844		2,3 0,46483		1,210,18625		0,310,27227		510,33329	53	10,69250		103 0,78917
4,6 0,23110	1	2,4 0,50613		1,3[0,19273	K	0,4[0,29646		610,35760	54	10,59779	44	104 0,80150
4,710,24450	144	2,5 0,55110		1,410,19943	k	0,510,32280		7 0,35611	55	10,72337		105 0,63856
4,810,25867		2,6 0,60007		1,5]0,20637	122	0,610,35148		810,37201	36	0,45469	6/-	106 0,9365
4,910,27366		2,710,65338	1224	1,610,21355		1 0,50613		910,23659	57	10,68335		107 0,8540
510,28952		2,8 0,71143		1,710,22097		1,1[0,55110	Hard Hard	10-0,30299		10,58168	1/1	10810,8263
5,1 0,30630		2,9 0,77464		1,910,23675		1,210,60007		1110,36836	59	10,42130		109 0,7881
5,210,32406	194	310,84346		310,35770	Dette	1,3 0,65338	Alter and a state of the state	1210,36130	60	10,62898	12-	110 0,77872
5,3 0,34284		3,1 0,91840		3,310,39705	6	1,4 0,71143	A Contraction of the second se	13 0,27312	61	10,49035		111 0,8116
5,4 0,36271		3,2[1,00000		3,510,42540		1,510,77464		1410,24745	62	0,66775		112 0,7181
5,5 0,38373	K	3,3[0,91840		3,6[0,44020	12 CO	1,610,84346		15 0,32153	63	0,55872		113 0,6474
5,610,40597		3,4 0,84346		3,7[0,45551	10000	1,7 0,91840		1610,29813	200 64	10,69005		114 0,7358
5,7 0,42950		3,5 0,77464		3,8[0,47135	No.	1,5 1,00000		1710,35023	65	10,62025		115 0,8457
5,810,45440		3,610,71143	P-335	3,910,48774	K -	1,910,91840		1810,39050	66	10,53247	-	116 0,7352
5,910,48073		3,710,65338		4[0,50471	Res Co	2 0,84346		19 0,34309	67	0,60477		117 0,7001
6 0,50860		3,8 0,60007		4,1[0,52226		2,110,77464		2010,32371	68	0,49581		118 0,6150
6,1 0,53807		3,9 0,55110		4,2[0,54042	1000	2,2 0,71143		21 0,39234	69	10,65259		119 0,8211
6,210,56926	1	4 0,50613		4,310,55922		2,310,65338		2210,36732	70	0,58855		120 0,7397
6,310,60225		4,1 0,46483		4,410,57867	P - M	2,410,60007		2310,30802	71	0,67570		121 0,7810
6,410,63716	1422	4,210,42691		4,510,59880	Ka	2,510,55110		24 0,34826	72	0,61541		122 0,7720
6,5 0,67409		4,4]0,35872		4,6[0,61962	12A	No and Contraction	CHART AND AND	2510,36749	73	0,64872		123 0,8345
6,6 0,71316		-74		4,710,64117		1 Acres	A Contraction of the second	2610,29361	74	0,68303	>	12411,0000
6,710,75449		14		4,8 0,66347	BX-1	423	And North And The	2710,38711	75	0,71375	\sim	125 0,7795
6,8 0,79822				4,910,68655	1		ALL MARK	2810,39310	76	0,82311		12610,8207
6,910,84449	62			5 0,71043				29[0,37861	1	10,66692	>	127 0,7634
7 0,89343				5,1[0,73514		200		3010,32285	78	10,48373		128 0,9345
7,1[0,94522	/			5,2[0,76070			X X THINK AND	31 0,30236	79	0,48534		129 0,8917
7,211,00000	-			5,3[0,78716		12/1/1	NO CONCENCIÓN OF	3210,40122	80	10,46896		130 0,7761
7,3 0,94522		/		5,410,81454		-11	and the second	3310,30504	81	0,55148		131 0,7442
7,4 0,89343				5,5 0,84287	45	11 total		3410,30010	82	0,67980		132 0,6928
7,610,79692	< _			5,610,87218	1	4-1		35 0,33051	68	0,68240		133 0,7428
7,7 0,75326				5,710,90252	44	TH?		3610,40389	84	10,65933		134 0,7980
7,9[0,67189				5,8[0,93391	41	114	TH ACTING	3710,26121	85	0,73561		135 0,7109
_			02	5,9[0,96639	4A	2/_	XXVIII AM	3810,37410	86	10,57776		136 0,8518
II SN12	4ofC3			6] 1,00000	11	1		3910,38717	87	10,56462		137 0,8171
DefSamp	des=126			6,1 0,96639	4			4010,24605	88	10,53129		138 0,8221
DetClasse	es=3. Iris-virginia	2		6,3 0,90198		-11	1 1 Letter	4110,30546	89	10,54699		139 0,7640
Contraction of the local division of the loc				6,4[0,87167	1			42 0,41350	90	0,65291		140 0,7596
PONo=24	CON0=9 ; OutVa	1=1.00000		6,610,81357				4310,33906	91	0,54581		141 0,8313
CI-1 CO-4	0 P1-VN7,2 : We	ight=0.3029 : OutVal=1	.00000	6,7 0,78623		R	ELATIONS	4410,28632	92	0,44397	100	142 0,7630
0-100-1	1 P2-VN3.2 : We	ight-0.1165 : OutVal-1	.00000	6,910,73383				45 0,31948	93	0,56110	624	143 0,7370
100000000		nt-0.3029 : OutVal-1.0						46 0,31472	94	10,56892		144 0,7297
and the second se			1000000					47[0,36316	95	0,60285		145 0,7765
C=1C0=	THE WILS: W	ight=0.2777 : OutVal=1	,00000				SEARCH	4810,31508	96	0,63734		146[0,7099
									97	10,46662		147 0,8161



FAST FINDING OF VARIOUS GROUPS AND CLASSIFICATION



.sepal length 🛛 🗹		2. sepal width 🛛 😨	1	P3. petal length Type: ordered		P4. petal width Type: ordered	X	C1.Iris-setosa Similarity = 0,526	C2.tris-versicolor Similarity = 0,980	C3.Iris-virg
pense: 1,00	E	pense: 1,00		Expense: 1,00		Expense: 1,00				
lue:	V	alue:		Value:		Value:		1 0,49085	49 0,91474	9910,9
1			-	e malazer			-	2 0,35519	50 0,87058	100 0,9
4,310,24450		2]0,50422		110,23661	2	0,1[0,27227		310,36919	51 0,94705	101 1.0
4,4[0,25867		2,2 0,60007		1,1 0,24484		0,210,29646		410,35395	52 0,66236	102 0,9
4,5 0,27366		2,310,65338		1,210,25336		0,310,32280		510,43978	5310,89592	103(0,9
4,610,28952	100	2,410,71143	10	1,3[0,26217	1	0,4]0,35148		610,50847	54 0,76905	104/1,0
4,710,30630		2,510,77464	100	1,4 0,27129	52-	0,510,38271		710,43563	5510,90682	105 0,7
4,810,32406		2,610,84346	0000	1,5[0,28072		0,610,41671	- All	810,45559	56 0,59610	106 1,0
4,910,34284		2,710,91840	220	1,6[0,29049		1 0,60007	the second secon	910,29884	57 0,86782	107(0,9
510,36271		2,811,00000		1,710,30059		1,1 0,65338		10 0,36586	5810,75527	
5,1 0,38373	- 10 C	2,911,00000	1000	1,910,32205		1,2 0,71143		11/0,51022	5910,54859	109 0,9
5,2 0,40597	-	311,00000		310,48657	1	1,3 0,77464	A COLOR OF THE OWNER	12]0,44293	6010,78700	110 0,9
5,310,42950		5,111,00000	200	3,310,54010	222	1,4 0,84346		13 0,33998	61[0,63343	
5,410,45440		1,211,00000	200	3,510,57867	2	1,5[0,91840		14 0,30717	62 0,85129	112 0,9
5,5 0,48073		3,311,00000	Ser	3,610,59880				15 0,46995	53 0,70677	113 0,9
5,6[0,50860	\geq	3,413,00000	100	3,710,61962 3,810,64117	200	1,7 (5,00000		16 0,41985 17 0,49845	64[0,85476	114 0,9
5,810,56926	225	3,511,00000	100	3,810,64117 3,910,66347	1000	1,014,00000		1710,49845	6610,69412	115 0,9
5,910,50920		3,711,00000		410,68655		314 00000		19 0,48843	67 0,77728	117]0,9
6 0,63716		3,811,00000	1000	4,1 0,71043	2000	211,0000		2010,45317	6810,63840	118 0,8
6,1 0,67409		ala la la deserve		4,2 0,73514		3,211,00000		2010,49317	69 0,78801	119 10
6,2[0,71316		3,910,91840		4,310,76070		2,311,00000		2210,48159	70(0,75573	12010,9
6,310,75449	>>>	4,110,77464	200	4,410,78716		2,414,00000		2310,40616	7010,75573	121/14
6,4[0,79822		4,210,71143	K	4,510,81454	200	2,511,00000		24[0,42136	7210,79528	122 0,9
6,510,84449		4,410,59780	2	4,610,84287				2510,45135	73 0,82506	123 0,9
6,6[0,89343	\geq	4/410/23/40		4,710,87218	20			2610,36555	74 0,85661	124 10
6,7 0,94522		4		4,810,90252	600	130 B		2710,47384	75 0,91309	125(0,9
6.811.00000	24			4,9[0,93391	X	XXX		2810,50178	76 0,97980	12510,9
6,911,00000	ZR		~	510,96639				2910,46349	77 0,84645	127 0,9
211.00000	>	\rightarrow		5,111,00000				3010,37755	7810,62670	128110
7.111.00000	//		2	5.211.00000				31/0,36697	79 0,63252	129 1.0
7,211,00000				5,313,00000	\gg	100		3210,49141	80 0,61237	
7.311.00000				5.411,00000				33[0,43523	81 0,71473	131 0.9
7,411,00000				5,511,00000	200	115		34[0,43112	82(0,86592	132/0,9
2.611.00000	/ /			5,611,00000	1	12 1		35(0,38454	83 0,85284	133]0,8
2,211,00000				5211.00000	21	7-16-		3610,51617	8410,82982	
2.911.00000				5.811.00000	17	TH	- CHARLOURS	3710,32440	85 0,91374	135(0,9
and a present of				5,911,00000	44	1	A ALLERA DON	3810,45646	B6[0,74703	136[0,9
				611.00000	17	1		3910,49346	87[0,70569	137[0,9
				6.111.00000	4			4010,31445	88 0,68192	13811.0
				6,311,00000	//	11		410,35511	89 0,70846	139[0.9
				6,411,00000	1	//	11114	4210,52562	90/0,82053	14011.0
				6,611,00060	1/			4310,47286	91 0,70603	141 12.0
				6,711,00000	//			4410,35562	92 0,57720	14210.9
				5.913.00000				45 0,44851	93 0,72509	143 0.5
				of a Laborator				46 0,36692	94 0,71285	144 0,8
								47 0,50167	95 0,76574	and the second second

48|0,38019

9610,80998

97 0,60124

98|0,73682

146 0,94622

147 0,94839

SIMILARITIES DEFINE GROUPS



DATA STRUCTURE AND EFFICIENCY OF DATA PROCESSING



The introduced associative graph data structures (AGDS) essentially reduce the speed of data access and eliminates loops that have to be used on data organized in tables. Thus, the applied data structures have fundamental importance in data mining and its efficacy. Appropriately organized data can also facilitate various cognitive processes as well as intelligent inference.

In the AGDS structures, there is possible to:

- Storing always sorted data for all attributes at the same time,
- Lossless compression of data by removing any redundancy by eliminating all duplicates of attribute values and objects,
- Linking attribute data through additional relationships not presented in tabular structures, mapping different vertical relationships, e.g. similarity, differences, order, minima, maxima, and thus also additional relationships between objects,
- Instantaneous data access (in constant time),

Automatic grouping of similar data and objects is built-in this structure and accessible in constant time.





The AGDS is not only another way of storing data in the graph structure, but it also replaces many operations and methods that have to be executed on tabular structures, looking for vertical relationships, e.g.:

- ✓ search for similar, different, correlated, inverse, neighbor, or duplicate objects,
- ✓ filter and search for various groups (e.g. clusters, classes) against given restrictions or constraints, selected attribute values, or their ranges,
- ✓ organize objects by all attributes simultaneously.

DEFINITION: We say that the structure replaces operations performer on another data structure when the computational complexity of the resulting data decreases to constant computational complexity O(1).

Generally speaking, if you have reached the computational goal in constant time (as in AGDS structures without loops) then your structure replaces more time-consuming operations that must be processed on another structure.

Due to the fact that in computer science we lose most of the time for data search operations, the AGDS structure can accelerate many operations and applications several dozens, hundred, or even thousand Times depending on the size of the browsed data! Intelligence demands such an efficiency!

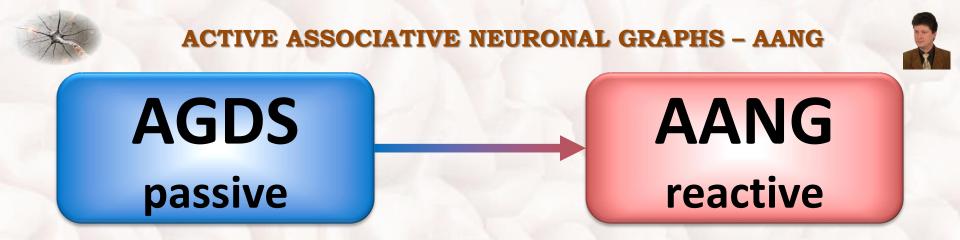
×

ACTIVE ASSOCIATIVE NEURONAL GRAPHS – AANG



- In the human brain, we find reactive neurons and active neuronal structures that not only quickly and effectively associate data but also are able to actively respond or react to incoming data from senses, i.e. receptors.
- Despite the slow-acting neurons in relations to the clocking speed of modern processors, mental processes are rapidly overwhelmed by the constant computational complexity of neuronal associative and recall processes.
- Such structures in the human brain do not have to go through the processes of crawling, searching, comparing, and exploring data in many nested loops, nor passive tables are used for storing data as in relational databases.
- Biological processes of knowledge formation, data storage, information, and reasoning are based on plastic associative processes that reach for relevant data if they are fixed in them through learning, experience, introspection, inference, or other cognitive processes in our minds.
- In addition, the human mind has the ability to compile various triggers from the memory of events regardless of the actual place and time of their occurrence. This ability allows you to create new cognitive contexts for next thought processes as well as provide creativity and generosity at the high mental, logical, emotional, and abstract levels.

Modern computer science is very expensive in finding and exploring large amounts of data. That is why we talk about BIG DATA PROBLEMS!

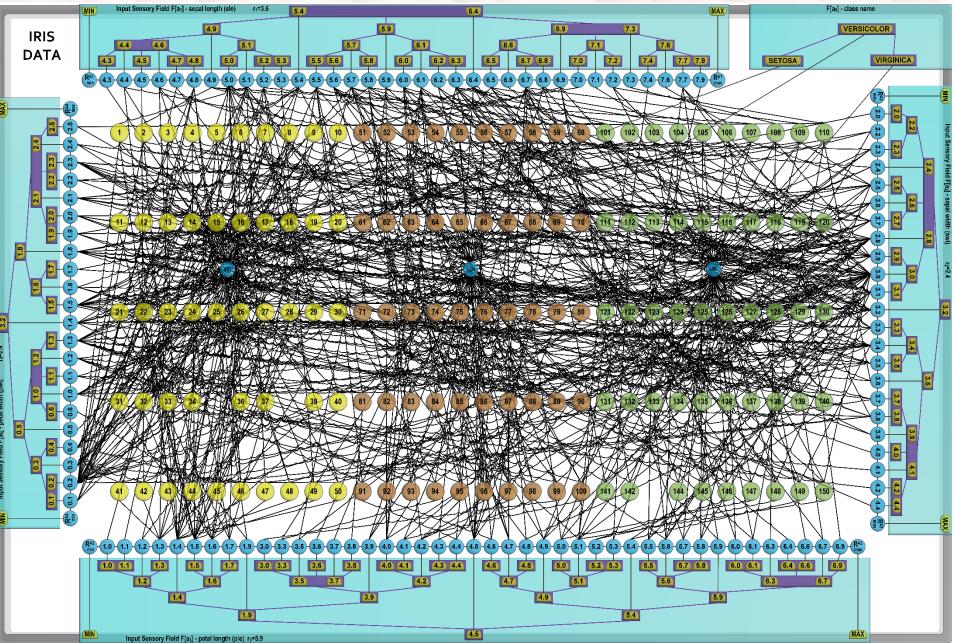


PASSIVE DATA STRUCTURES are designed to store data in their intact form

REACTIVE DATA STRUCTURES react to new data and allow data to interact with each other automatically

The AANG constructed for all Iris data from ML Repository using AVB-trees for representation of all attribute values



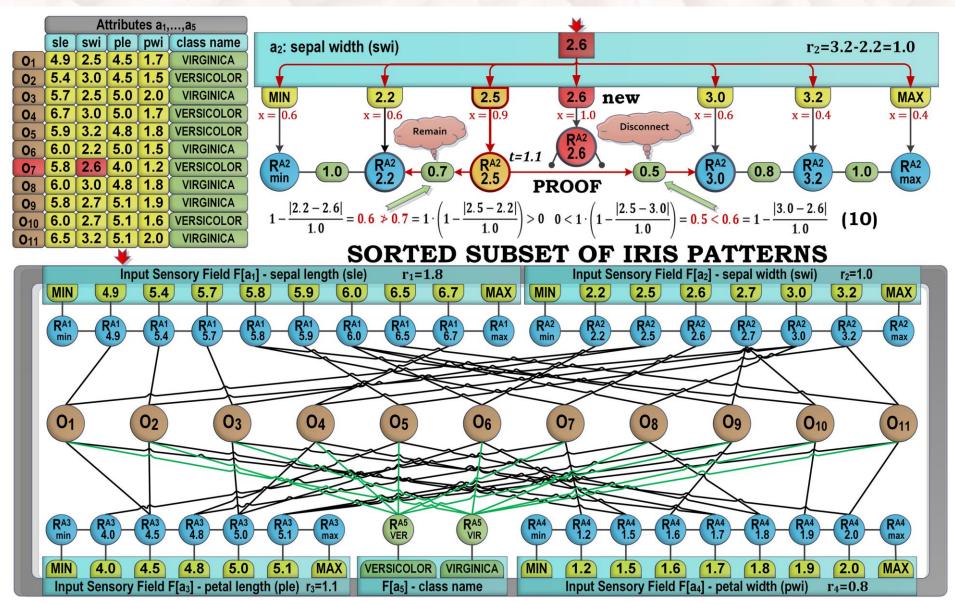




CREATION OF REACTIVE ASSOCIATIVE NEURONAL STRUCTURES



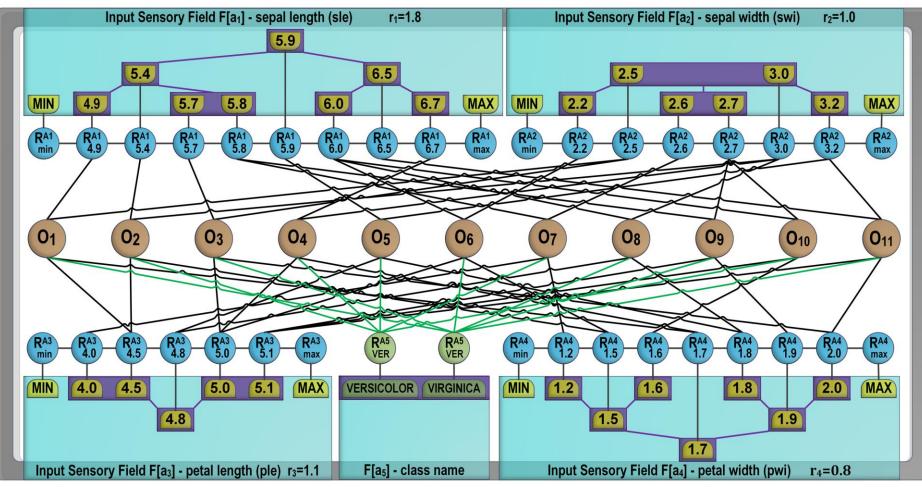
The ASSORT-2 algorithm automatically creates the basic associative neuronal structure for any table.





REACTIVE ASSOCIATIVE NEURAL STRUCTURES ON SEQUENTIAL MACHINES

Buy in large, contemporary computers work sequentially, have sequential cores in processors, sequential memories and sequential ways of executing operations and programs. Brains are parallel, and all internal processes run simultaneously. When implementing reactive neuronal structures on contemporary computers, we have to keep in mind these limitations and use AVB-trees to efficiently organize and access attribute data represented by sensors in sensory







Suppose we have objects $o_1, ..., o_N$ defined by the attributes $a_1, ..., a_K$ in such a way that each object is defined by a set of values of these attributes (K is a number of all attributes):

 $o_n = (v_{n_1}^{a_1}, ..., v_{n_K}^{a_K})$

Let these values react to certain sensory fields, modeling the senses, having sensors, modeling the receptors, enabling them to react to these values at a certain intensity. Determination of ranges of represented values by the input sensory fields is computed after:

$$r^{a_k} = v^{a_k}_{max} - v^{a_k}_{min} \qquad where \ v^{a_k}_{max} = max\{v^{a_k}_i\}, \ v^{a_k}_{min} = min\{v^{a_k}_i\}$$

Sensors in the sensory fields are created after the presentation of the stimulus that is not yet represented by any of the existing sensors, i.e. none of the existing sensors does not react enough, i.e. the distance between the presented and represented the value by this sensor is bigger than a defined certain minimum sensitivity:

$$d(v^{a_k}, v^{a_k}_i) = |v^{a_k} - v^{a_k}_i| \qquad \forall_i \, d(v^{a_k}, v^{a_k}_i) > \varepsilon^{a_k}$$

In case, when one of the sensors recognizes a certain value of the stimulus with a certain force, then the new sensor is not created: $d(v^{a_k}, v_i^{a_k}) \le \varepsilon^{a_k}$

To the extreme (minimum and maximum) values of external stimuli react extreme sensors:

 $S_{min}^{a_k}$ and $S_{max}^{a_k}$





Sensors react (respond) to external stimuli with a specific force depending on the proximity of the stimulus value to the value represented by that sensor, for which it is the most sensitive. Extreme sensors $S_{min}^{a_k}$ and $S_{max}^{a_k}$ use the following formulas to compute their responses:

$$x_{min}^{a_k} = \begin{cases} \frac{v_{max}^{a_k} - v^{a_k}}{r^{a_k}} & \text{if } r^{a_k} > 0\\ v_{min}^{a_k} - v^{a_k} + 1 & \text{if } r^{a_k} = 0 \end{cases}$$

$$x_{max}^{a_k} = \begin{cases} \frac{v^{a_k} - v_{min}^{a_k}}{r^{a_k}} & \text{if } r^{a_k} > 0\\ v^{a_k} - v_{max}^{a_k} + 1 & \text{if } r^{a_k} = 0 \end{cases}$$

Value sensors $S_{v_i}^{a_k}$ represent attribute values and calculate their responses on the sensory input stimulations v^{a_k} on the basis of the following formula:

$$x_{v_i}^{a_k} = \begin{cases} 1 - \frac{|v_i^{a_k} - v^{a_k}|}{|v_i^{a_k}|} & \text{if } r^{a_k} > 0\\ \frac{|v_i^{a_k}|}{|v_i^{a_k}| + |v_i^{a_k} - v^{a_k}|} & \text{if } r^{a_k} = 0 \end{cases}$$

The stimulated sensors $S_{v_i}^{a_k}$ start to stimulate connected sensory neurons $R_{v_i}^{a_k}$ with the computed strength $x_{v_i}^{a_k}$ as long as the value v^{a_k} is presented at the input sensory field. This can lead to activation of the connected neurons after a certain period of time which can be computed after:

$$t_{v_i^{a_k}} = \begin{cases} \frac{r^{a_k}}{\theta_{R_{v_i}^{a_k}} \left(r^{a_k} - |v_i^{a_k} - v^{a_k}| \right)} & if |v_i^{a_k} - v^{a_k}| < r^{a_k} \\ \infty & if |v_i^{a_k} - v^{a_k}| = r^{a_k} \\ 1 + \frac{|v_i^{a_k} - v^{a_k}|}{|v_i^{a_k}|} & if r^{a_k} = 0 \end{cases}$$



 $x_{min}^{a_k}$

 $x_{max}^{a_k}$

CREATION OF ACTIVE ASSOCIATIVE NEURAL GRAPHS



Next, extreme neurons $R_{min}^{a_k}$ and $R_{max}^{a_k}$ can react to extreme values according to their stimulation by extreme sensors. Their reactions can be divided into three categories (ranges):

< 1 for non-extreme values

- = 1 to the values equal to the current extremum (minimum or maximum)
- > 1 to the values that are new extremum to the current one

$$y_{min}^{a_k} = f(x_{min}^{a_k}) = \begin{cases} 1 & \text{if } x_{min}^{a_k} \ge \theta_{min}^{a_k} \\ 0 & \text{if } x_{min}^{a_k} < \theta_{min}^{a_k} \end{cases}$$

$$y_{max}^{a_k} = f(x_{max}^{a_k}) = \begin{cases} 1 \ if \ x_{max}^{a_k} \ge \theta_{max}^{a_k} \\ 0 \ if \ x_{max}^{a_k} < \theta_{max}^{a_k} \end{cases}$$

Extreme neurons are connected to the value neurons representing extreme values. The connection weights are always equal the activation thresholds of the connected neurons:

$$\begin{split} w_{R_{min}^{a_{k}}, R_{v_{min}}^{a_{k}}} &= \theta_{R_{v_{min}}^{a_{k}}} & w_{R_{max}^{a_{k}}, R_{v_{max}}^{a_{k}}} &= \theta_{R_{v_{max}}^{a_{k}}} \\ w_{R_{v_{min}}^{a_{k}}, R_{min}^{a_{k}}} &= \theta_{R_{min}^{a_{k}}} & w_{R_{v_{max}}^{a_{k}}, R_{max}^{a_{k}}} &= \theta_{R_{max}^{a_{k}}} \end{split}$$

The activation thresholds in this model are always equal one ($\theta = 1$).



Sensory neurons react to stimulations coming from sensors $S_{v_i}^{a_k}$, neighbor sensory neurons $R_{v_i}^{a_k}$, and object neurons O_n according to the following formula:

$$X_{R_{v_{i}}^{a_{k}}} = t_{v_{i}^{a_{k}}} \cdot x_{v_{i}}^{a_{k}} + \sum_{j}^{R_{v_{j}}^{a_{k}}} y_{R_{v_{j}}^{a_{k}}} \cdot w_{R_{v_{j}}^{a_{k}}, R_{v_{i}}^{a_{k}}} + \sum_{n}^{O_{n} \rightsquigarrow R_{v_{i}}^{a_{k}}} y_{O_{n}} \cdot w_{O_{n}, R_{v_{i}}^{a_{k}}}$$

And calculate their output value depending on the achievement of their activation thresholds:

$$y_{R_{v_j}^{a_k}} = \begin{cases} 1 & \quad if \ X_{R_{v_j}^{a_k}} \ge \theta_{R_{v_j}^{a_k}} \\ 0 & \quad if \ X_{R_{v_j}^{a_k}} < \theta_{R_{v_j}^{a_k}} \end{cases}$$

While sensors can stimulate them for some time, charging them until they reach their activation thresholds which is determined by the following formula:

$$t_{v_i^{a_k}} = \begin{cases} \frac{r^{a_k}}{\theta_{R_{v_i}^{a_k}} \left(r^{a_k} - |v_i^{a_k} - v^{a_k}| \right)} & if |v_i^{a_k} - v^{a_k}| < r^{a_k} \\ \infty & if |v_i^{a_k} - v^{a_k}| = r^{a_k} \\ 1 + \frac{|v_i^{a_k} - v^{a_k}|}{|v_i^{a_k}|} & if r^{a_k} = 0 \end{cases}$$





Sensory neurons are connected by synapses which weights are determined by:

$$w_{R_{v_i}^{a_k}, R_{v_j}^{a_k}} = 1 - \frac{\left|v_i^{a_k} - v_j^{a_k}\right|}{r^{a_k}}$$

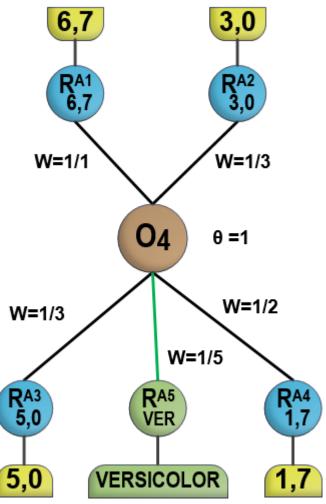
Sensory neurons are connected to object neurons representing objects defined by values represented by these sensory neurons.

The weights of synaptic connections leading from sensory neurons to object neurons are determined by:

$$w_{R_{v_{i}}^{a_{k}},O_{n}} = \frac{1}{\|v_{i}^{a_{k}}\|}$$

The weights of synaptic connections leading from object neurons to sensory neurons are equal their activation thresholds:

$$w_{O_n, R_{v_i}^{a_k}} = \theta_{R_{v_i}^{a_k}} = 1$$







The stimulation of object neurons is determined by:

$$X_{O_n} = \sum_{k}^{R_{v_{n_k}}^{a_k} \leadsto O_n} y_{R_{v_{n_k}}^{a_k}} \cdot w_{R_{v_{n_k}}^{a_k}}, O_n$$

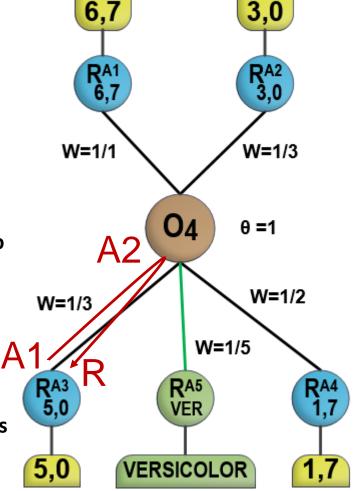
And their output value is computed as follows:

 $y_{O_n} = \begin{cases} 1 & \text{if } X_{O_n} \ge \theta_{O_n} \\ 0 & \text{if } X_{O_n} < \theta_{O_n} \end{cases}$

Where the neuron activation thresholds are initially equal to one:

$$\theta_{O_n} = 1$$

Thanks this, if there is presented an input combination defining a known object on the sensory input fields, the neuron representing this combination will activate at first. The other neurons representing similar combinations will activate later if the input combination is further presented on the input sensory fields of the AANG.



Neurons, which were activated (e.g. A1) are for some time in the refractory states (R), so they are not reactive to any stimulations, e.g. the one coming back from the neuron A2.





Sensory neurons should not only react to sensory stimuli of a specific intensity but also stimulate other connected sensory neurons with the most similar values.

Hence, there is necessary the self-organizing capability of the AANG network.

The sensory connective plasticity rule determines in which cases the plasticity results in the creation or reconfiguration of existing connections between sensory neurons.

The sensory connective plasticity rule between sensory neurons says that the sensory neuron $R_{v_j}^{a_k}$ will disconnect with the neuron $R_{v_i}^{a_k}$ which stimulates it weaker than the connected sensor $S_{v_j}^{a_k}$:

$$0 < y_{R_{v_i}^{a_k}} \cdot w_{R_{v_i}^{a_k}, R_{v_j}^{a_k}} < x_{v_j^{a_k}} - \varepsilon^{a_k}$$

Sensory neurons are thus programmed to require precisely two connections with the remaining sensory neurons or extreme neurons.

Disconnection thus triggers the neuronal process of connective plasticity, which will look for other neurons that wish to connect at a given moment.

Therefore, if a new sensor and its new sensory neuron for a not yet represented new value in a given sensory field is created then this new sensory neuron will try to connect to these two disconnected neurons.

In result, the new sensory neuron representing the value v^{a_k} will join the others in an orderly way: $v_j^{a_k} < v^{a_k} < v_i^{a_k}$ or $v_j^{a_k} > v^{a_k} > v_i^{a_k}$





However, this plasticity is only possible in the sensory neuron stimulated simultaneously by the sensor and another previously activated sensory neuron.

Therefore, it is important to take into consideration and computation the time and the order of activations of the individual sensory neurons in time to make such plasticity.

The activation time of sensory neurons as a result of their stimulation by the connected sensors will vary depending on the similarity of represented values by sensors to the presented value on their input sensory fields: $\int \frac{r^{a_k}}{1 + r^{a_k}} = \frac{i f |v_k^{a_k} - v_k^{a_k}|}{1 + r^{a_k}} \leq r^{a_k}$

$$t_{v_i^{a_k}} = \begin{cases} \frac{\overline{\theta_{R_{v_i}^{a_k}} \left(r^{a_k} - |v_i^{a_k} - v^{a_k}| \right)}}{\sum \\ \infty & \text{if } |v_i^{a_k} - v^{a_k}| = r^{a_k} \\ 1 + \frac{|v_i^{a_k} - v^{a_k}|}{|v_i^{a_k}|} & \text{if } r^{a_k} = 0 \end{cases}$$

The neuron, which activates first as a result of such sensory stimulation, sends a weighted signal to the two connected sensory (or extreme) neurons. Always only one of it will satisfy the connective plasticity condition:

$$0 < y_{R_{v_i}^{a_k}} \cdot w_{R_{v_i}^{a_k}, R_{v_j}^{a_k}} = 1 \cdot \left(1 - \frac{|v_i^{a_k} - v_j^{a_k}|}{r^{a_k}} \right) < 1 - \frac{|v_j^{a_k} - v^{a_k}|}{r^{a_k}} = x_{v_j^{a_k}}$$

and breaks its connection to this neuron because:

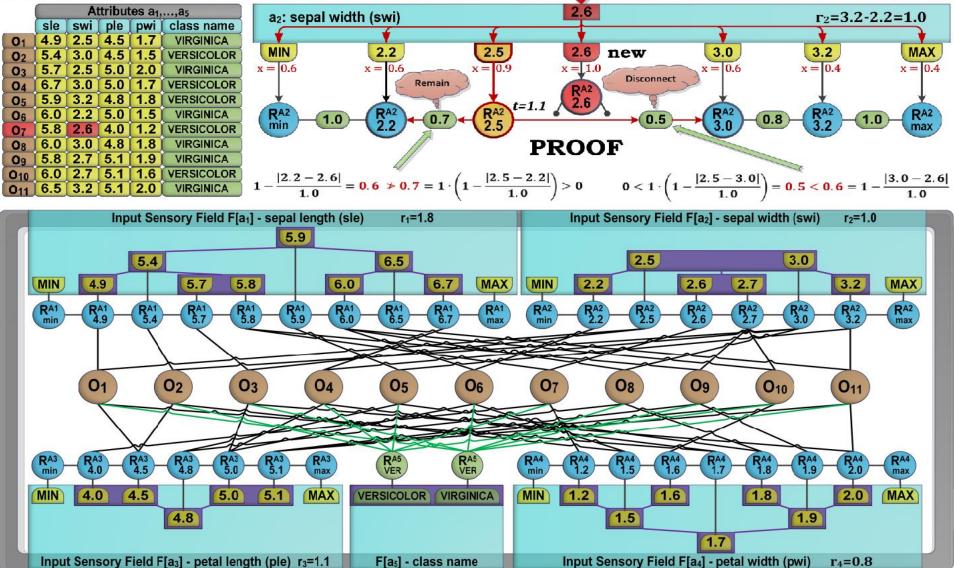
$$\left| v_i^{a_k} - v_j^{a_k} \right| > \left| v_j^{a_k} - v^{a_k} \right| if and only if v_j^{a_k} < v^{a_k} < v_i^{a_k} \text{ or } v_j^{a_k} > v_i^{a_k} > v_i^{a_k}$$

The presented algorithm is called the ASSORT-2 associative sort algorithm and is used for the automatic and incremental construction of the AANG neural network for any set of patterns.



We can get the following graph structure built by the ASSORT-2 algorithm:

CONDITIONAL PLASTICITY PROCESS PLASTICITY CONDITION







SORTED SUBSET OF IRIS PATTERNS						
Attributes						
sle	swi	- Y		class name	S / III	
	\rightarrow	<u>· · · · · · · · · · · · · · · · · · · </u>		VERSICOLOR	< 🗆	
R2 5.4	3.0	4.5 [^]	1.5	VERSICOLOR		
R3 6.0	2.7	5.1 ′	1.6	VERSICOLOR		
R4 6.7	3.0	5.0	1.7	VERSICOLOR		
R5 5.9	3.2	4.8	1.8	VERSICOLOR		
R6 6.0	2.2	5.0	1.5	VIRGINICA		
R7 4.9	2.5	4.5	1.7	VIRGINICA		
R8 6.0	3.0	4.8	1.8	VIRGINICA		
R9 5.8	2.7	5.1 ′	1.9	VIRGINICA		
	\rightarrow		2.0	VIRGINICA		
R11 6.5	3.2	5.1	2.0	VIRGINICA		
STEP 1						

STEP 1. CREATION OF A NEW GRAPH

Create a new AANG graph for a set of objects stored in a tabular structure (table).





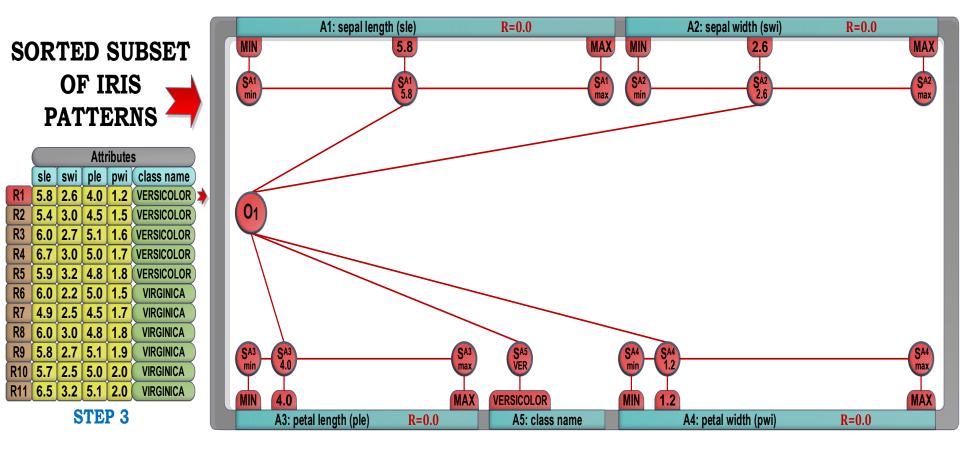
	A1: sepal leng	th (sle) R:	=0.0	A2: sepal width (swi)	R=0.0
SORTED SUBSET					
OF IRIS 🛁					
PATTERNS 🔫					
Attributes					
sle swi ple pwi class name 🖈					
R1 5.8 2.6 4.0 1.2 VERSICOLOR					
R2 5.4 3.0 4.5 1.5 (VERSICOLOR)					
R3 6.0 2.7 5.1 1.6 VERSICOLOR					
R4 6.7 3.0 5.0 1.7 VERSICOLOR					
R5 5.9 3.2 4.8 1.8 VERSICOLOR					
R6 6.0 2.2 5.0 1.5 VIRGINICA					
R7 4.9 2.5 4.5 1.7 VIRGINICA					
R8 6.0 3.0 4.8 1.8 VIRGINICA					
R9 5.8 2.7 5.1 1.9 VIRGINICA					
R10 5.7 2.5 5.0 2.0 VIRGINICA					
R11 6.5 3.2 5.1 2.0 VIRGINICA					
STEP 2	A3: petal length (ple)	R=0.0	A5: class name	A4: petal width (pwi)	R=0.0

STEP 2. CREATION OF SENSORY FIELDS IN THE GRAPH (INPUT INTERFACES FOR THE AANG)

Create new sensory fields for all known attributes defined in the transformed table.





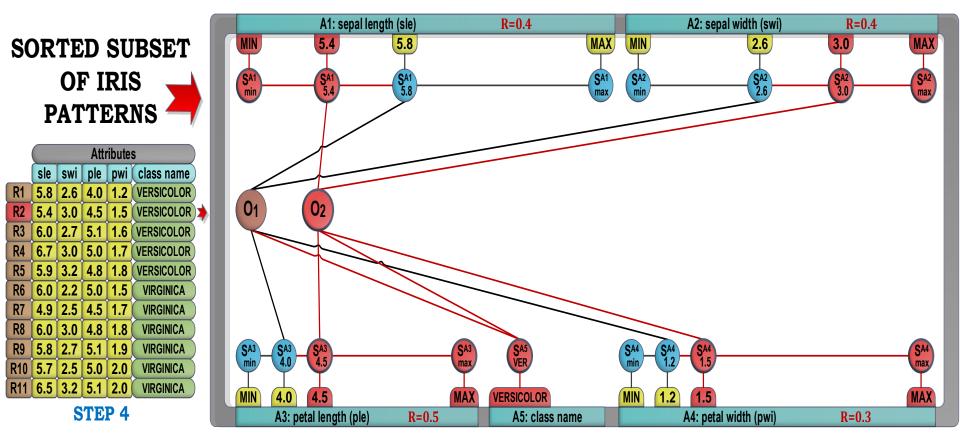


STEP 3. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R1

Create an object representation related to new sensory neurons and sensors using ASSORT **the sensory connective plasticity rule**, which connects new sensory neurons in order.





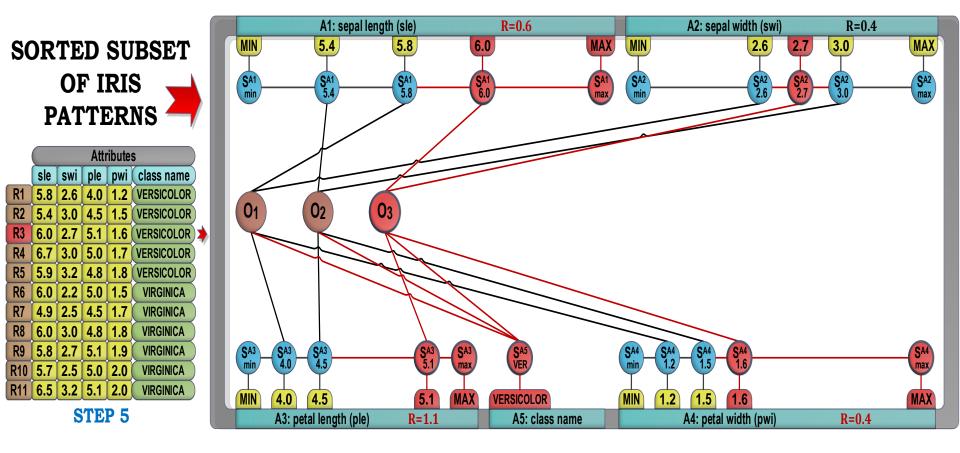


STEP 4. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R2

Create a representation of another object (R2) in the AANG structure using already created sensors and sensory neurons, aggregating, not duplicating, representation of the same values (VERSICOLOR).





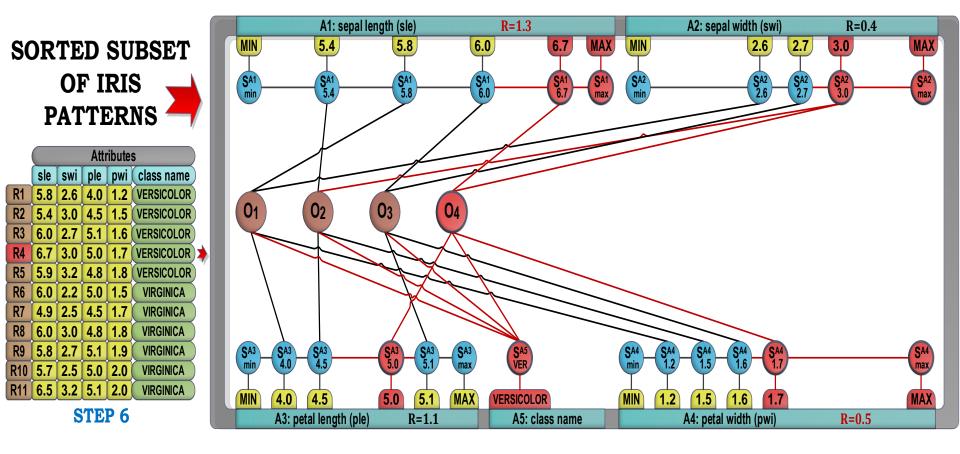


STEP 5. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R3

Presentation of further objects causes stimulation and activation of sensory neurons if they represent presented values or addition of new ones if the presented values are not yet represented.





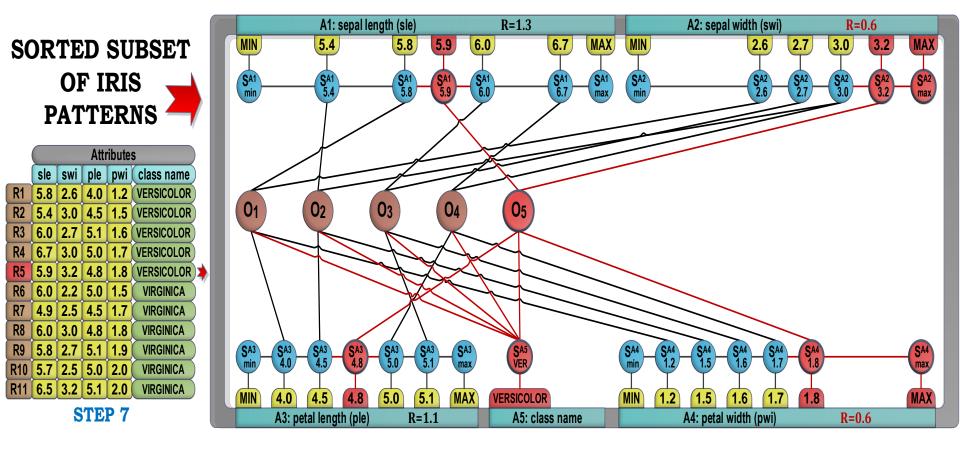


STEP 6. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R4

There is visible the aggregated representation of the same attribute values by the same sensors and sensory neurons, e.g. 3.0 for the attribute A2: sepal width.





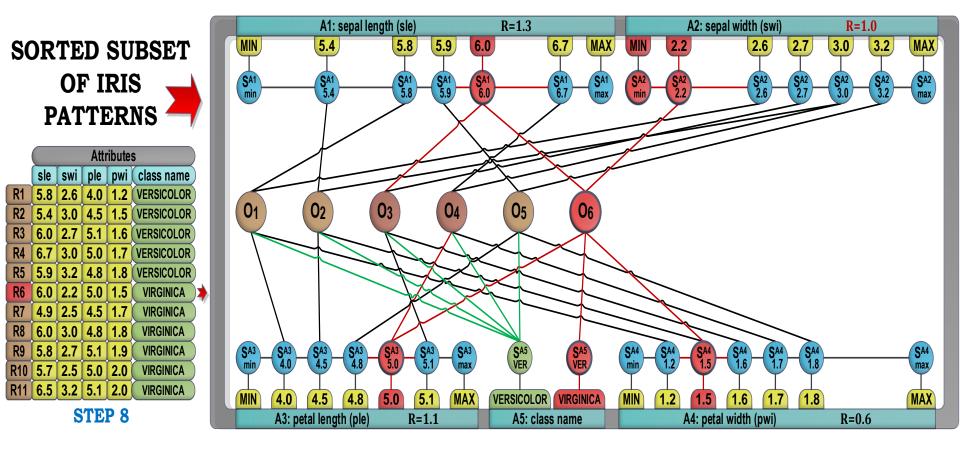


STEP 7. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R5

Sometimes MIN or MAX sensors are also activated when the values presented on the sensory input fields are minimum or maximum in the range of given attributes.





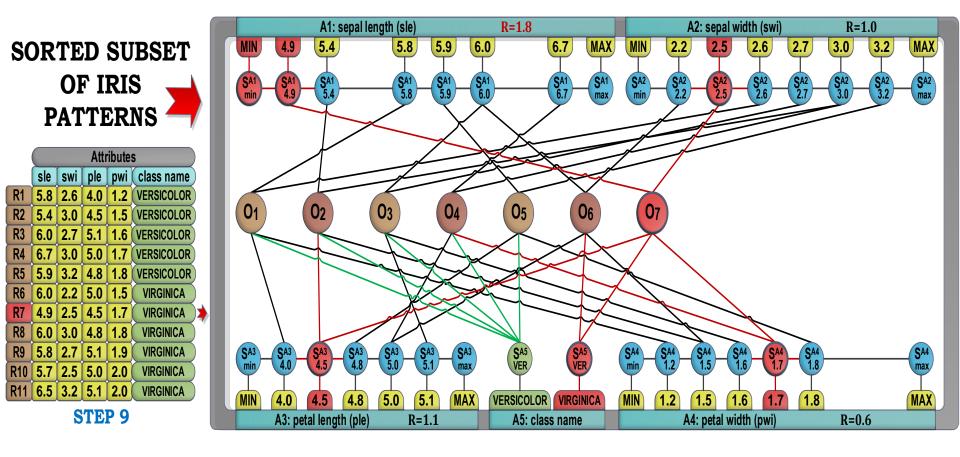


STEP 8. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R6

The number and level of aggregations will grow together with the number of represented objects, e.g. 6.0 for the attribute A1 and 5.0 for the attribute A3), which reduces the cost of representation.





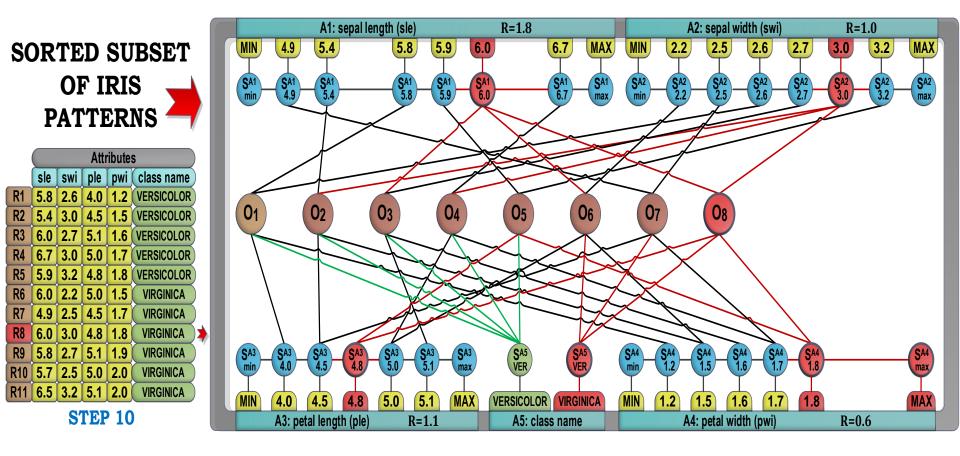


STEP 9. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R7

Subsequent object aggregations enable automatic associations between objects, e.g. the sensory neuron 4.5 for the attribute A3 links together objects R2 and R7, and 1.7 for the attribute A4 links together objects R3 and R7.





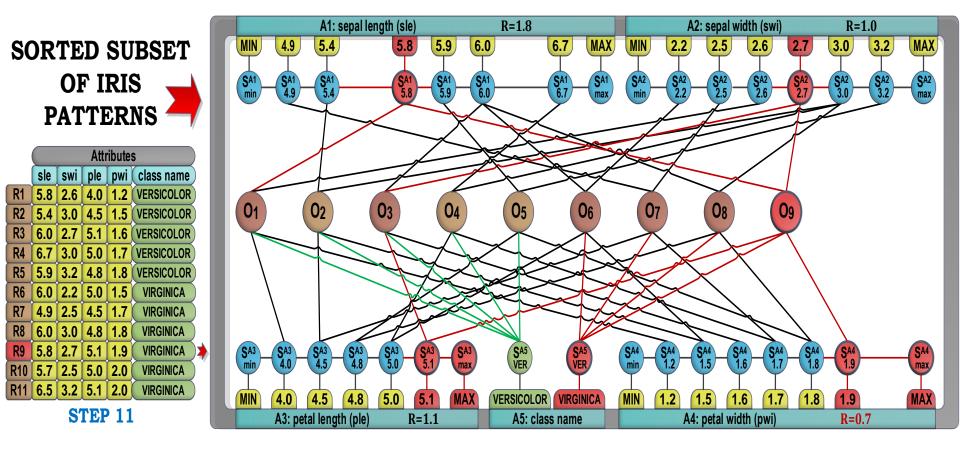


STEP 10. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R8

Neuronal aggregates also allow for automatic grouping (clustering and classification) of objects against any combination of input values as well as similar values that are linked together.





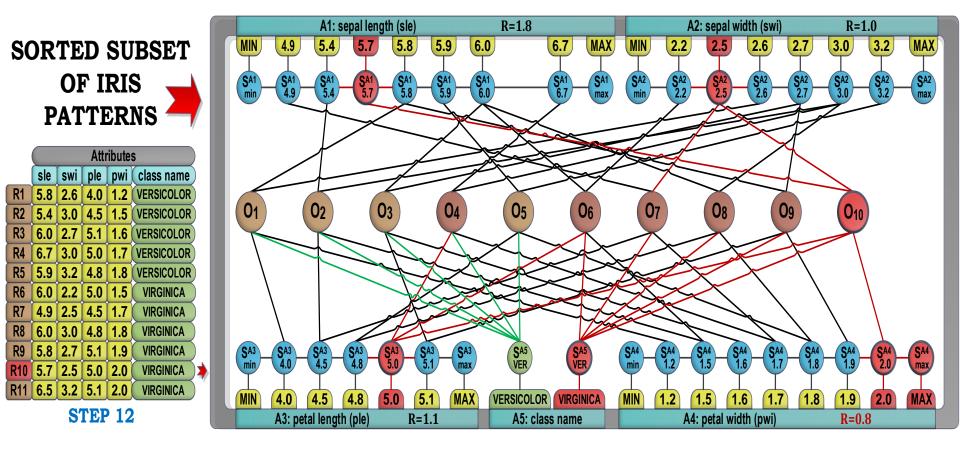


STEP 11. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R9

Most aggregations occur where there are natural classes (e.g. VERSICOLOR or VIRGINICA), but in this model, the class can be defined by any aggregation of the same values or any range of them. The AANG graph allows to quickly process any grouping or filtration.





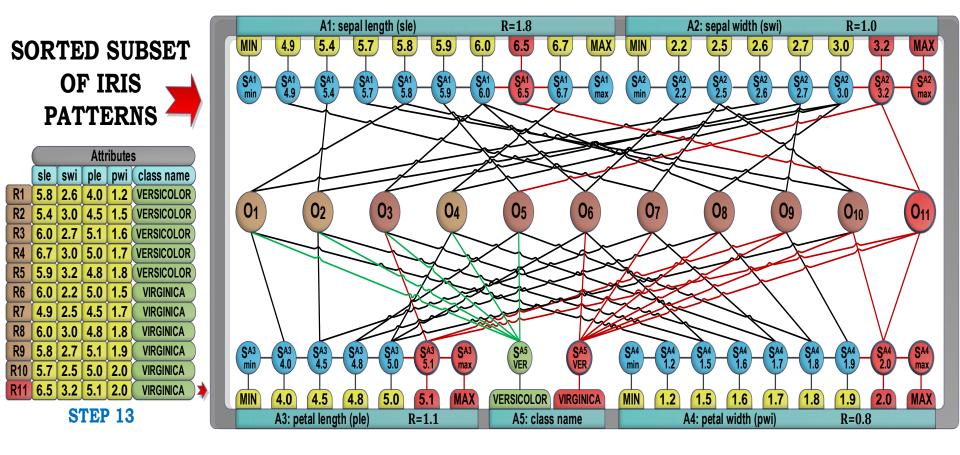


STEP 12. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R10

Aggregates can group and associate multiple objects, e.g. 5,0 for the attribute A3 naturally associates the objects R4, R6 and R10. Such associations can be found instantly in constant time.





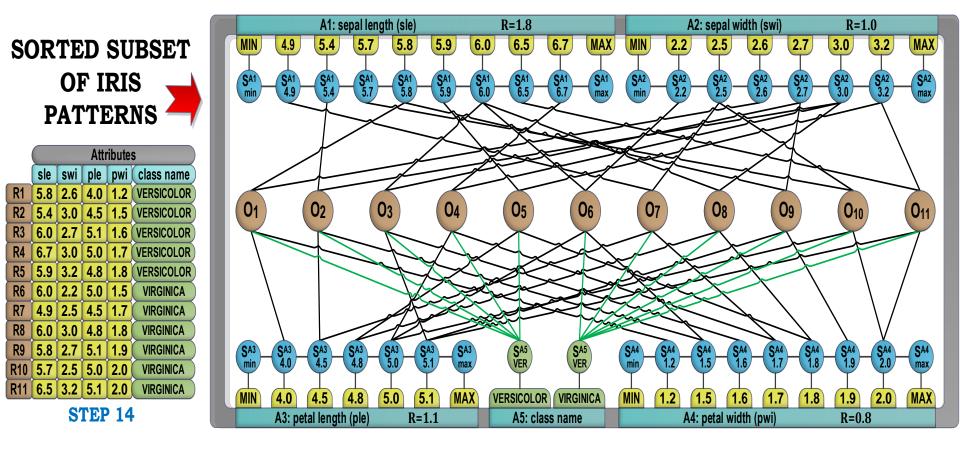


STEP 13. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R11

The built-in associative neural graph with the ASSORT algorithm can then serve to quickly and automatically infer the different relationships encoded in this associative structure!







THE AANG HAS BEEN CREATED THANKS TO THE ASSORT ALGORITHM

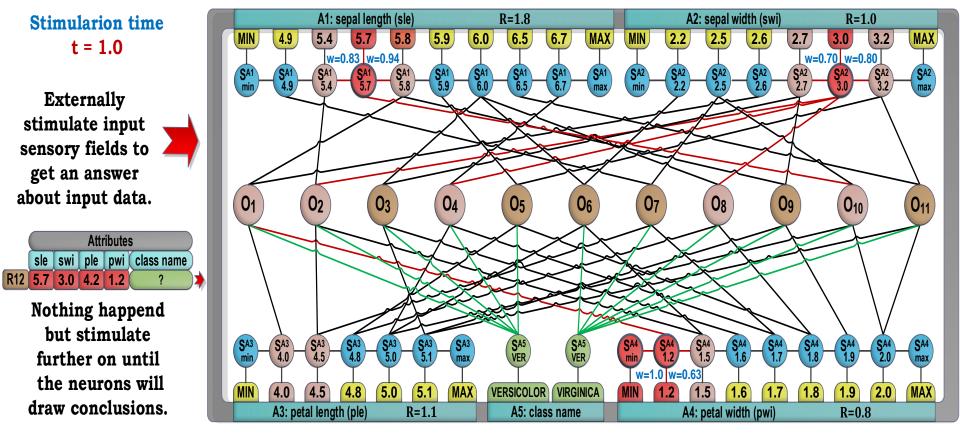
The final associative neuronal graph structure represents all sorted objects simultaneously for all attributes! Hence, we do not need to sort anything again...





WE HAVE TO STIMULATE SENSORS OR NEURONS IN ORDER TO USE AANG

We start to present a new pattern (object) of unknown class to the AANG, stimulating the appropriate sensory fields for a a specified period of time:



THERE IS NO ACTIVE NEURON REACTION?

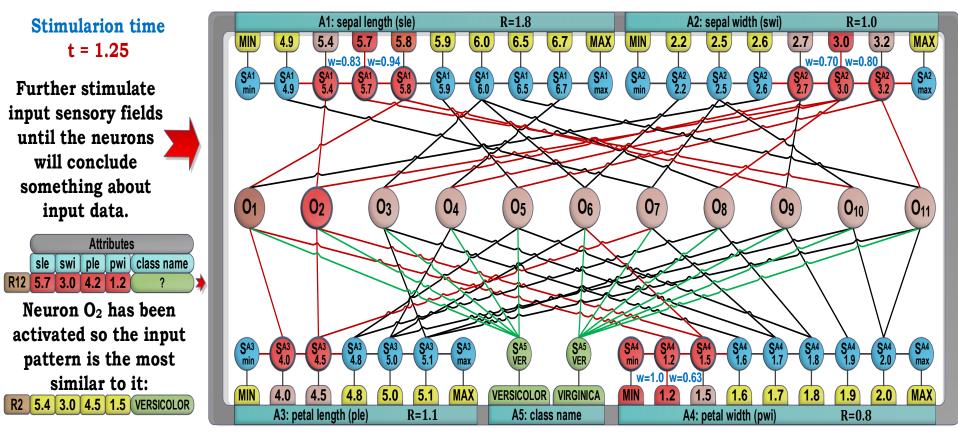
So far, the neurons have not yet reacted? Why? Maybe we were too short stimulating the network? Remember that in artificial associative systems time is a computational factor, so let's keep on stimulating.





WE HAVE TO STIMULATE THE NETWORK FURTHER

After a slightly longer period of time, neurons were activated because their charging process was slower.



THIS TIME THE NEURON O₂ WAS ACTIVATED

The O_2 neuron was activated the fastest, indicating the most similar object to the pattern presented on the input of this network, but we still do not have an answer for its class.

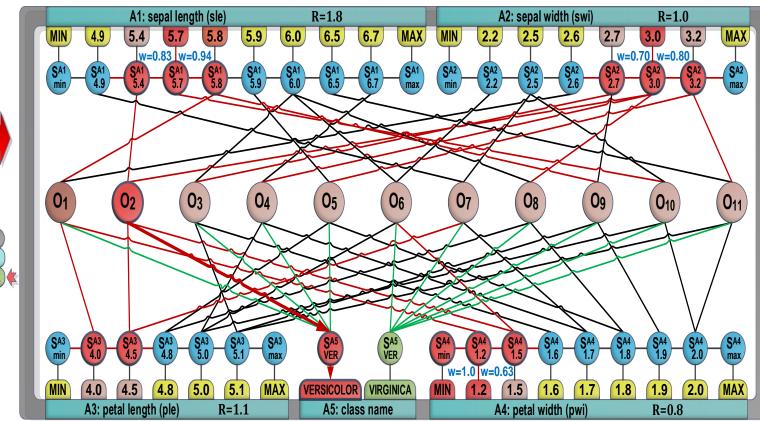




ACTIVATED NEURONS AUTOMATICALLY STIMULATE THE OTHER CONNECTED NEURONS

The O_2 neuron is connected to the neuron representing the VERSICOLOR class. The weight of this connection is equal to the activation threshold of that neuron, so it produces the correct response.

Stimularion time t = 1.30 Activated neuron O₂ stimulates the connected neuron representing the class Versicolor drawing the correct conclusion about the class of pattern R12. Attributes sle swi ple pwi class name R12 5.7 3.0 4.2 1.2 VERSICOLOR (Pattern R12 has been correctly classified as VERSICOLOR.

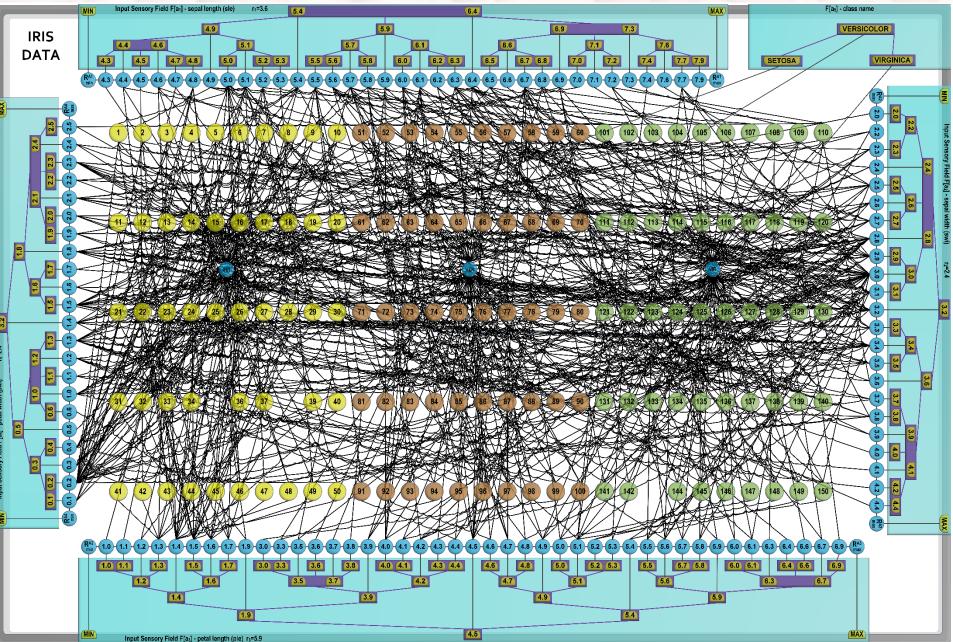


ACTIVATION OF NEURONS IS THE RESPONSE OF THE AANG NETWORK

In this way, without the use of any inference, search, or comparison algorithm, we have received an answer coming from associations represented by the neurons of this neural network.

THE AANG CREATED FOR ALL IRIS PATTERNS FROM ML REPOSITORY USING AVB-TREES TO REPRESENT ALL FEATURES







WHAT CAN BE ACHIEVED WITH ASSOCIATIONS



Active associative neural graphs AANG enable us to:

- Actively interact data with each other through the use of neurons and a graph structure that integrates data, groups, and sequences that are properly consolidated.
- ✓ Automatic recall relationships between data in the table.
- Sort objects using neurons with context-sensitive plasticity triggers for their actions, i.e.:
 - creating new connections,
 - breaking old connections,
 - update of synaptic weights.
- Perform local implementation of all calculations performed during ASSORT sorting without the involvement of external algorithms to iterate on neurons or consolidate their results. The neurons unintentionally do the sorting, which results from their characteristic plasticity operations, known from the biological nervous system.
- Sort objects against all attributes simultaneously and simultaneously with linear computational complexity O (n) and without the need to create additional indices for relational databases to speed up the operations performed on them.
- Add new objects in parallel for all attributes while maintaining order with constant computational complexity O (1) for each of them.
- ✓ Automatic inference by stimulating associated data or objects.
 - Automatic construction of **intelligently concluding cognitive associative systems** after the given constraints, conditions, or for the given value or their ranges.



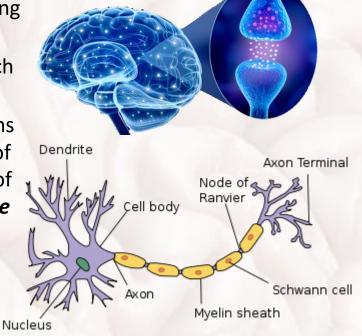
SPIKING NEURONS

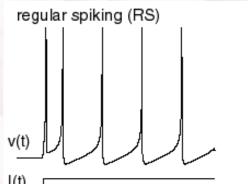


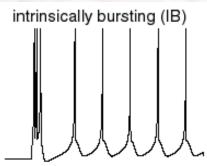
Spiking neurons are models of biological neurons that fall into the third generation of neural network models, increasing the realism in neural simulations.

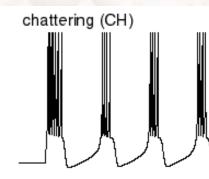
These models incorporate the concept of time into the operating model of the previously used artificial McCulloch-Pitt's model. Artificial neurons do not fire even if they implement hard-switch activation functions.

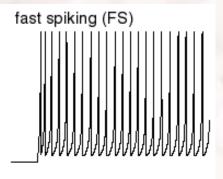
The **fundamental problem** is to propose the model that explains how information is encoded and decoded by a series of trains of pulses, i.e. action potentials. Thus, the fundamental question of neuroscience is to determine *if neurons communicate by a rate* or temporal code? Temporal coding suggests that a single spiking neuron can replace hundreds of hidden units on a sigmoidal neural network. Is that true?













SPIKING ASSOCIATIVE NEURONS



absolute refraction

The spiking associative neuron is a functional model of biological neurons. This is a simplistic model in comparison to spiking neurons, but it does not emphasize or try to model biological platform truly. It focuses on efficient modeling of time-dependent functional aspects that are responsible for the information processes that take place in biological neurons and their networks. This model also has some built-in routines that enable it automatically connect to other neurons according to some

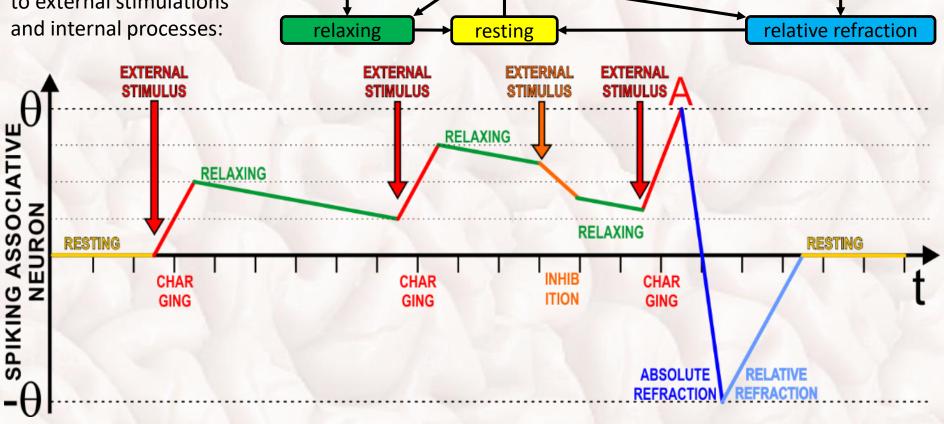
suppression

charging

activation

plasticity rules.

This model is reactive to external stimulations







Active Associative Neural Graphs (AANGs) are able to combine **inferences based on similarity and sequences** based on the neuronal graph and its parameters. For now, let's focus on the **conclusions based on similarity**. Let's take a little clipping of this graph, assuming that the chosen neuron representing the object O_j can be externally stimulated for some time in order

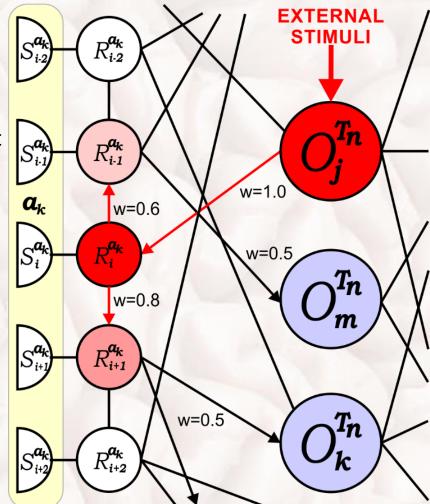
to find out other neurons of the same kind in this graph, which are most similar to it.

The degree of similarity will be determined by the time after which such neurons will be activated.

For ease of analysis, let us consider only one object neuron O_k , which may potentially be indirectly influenced by another object neuron O_j that is **externally stimulated for some period of time.**

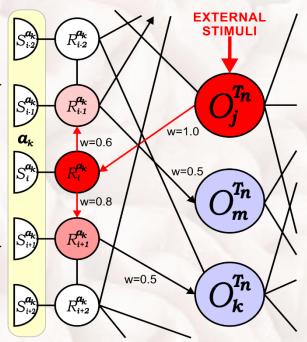
This period can be determined by the user in order to perform analysis of the similarity between these two objects represented by these neurons. The interaction between neurons is possible thanks to **neuronal affinity associations** fixed in the associative neuronal graph.

These associations are represented by weighted connections between sensory neurons R_i and R_{i+1}.

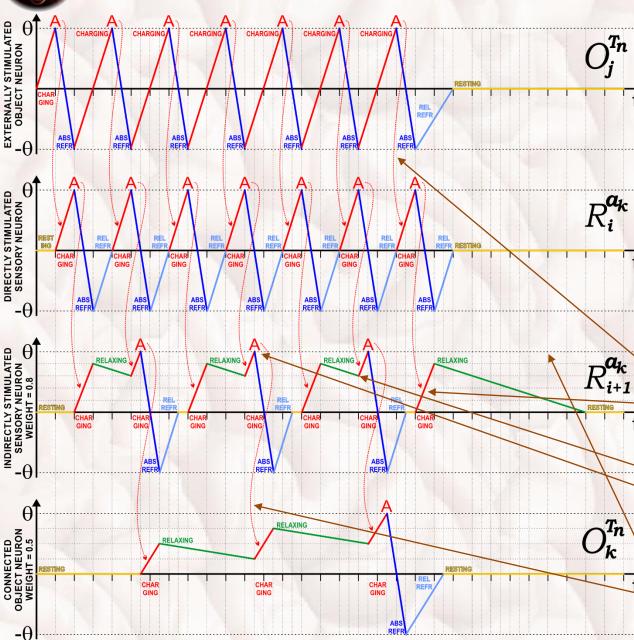






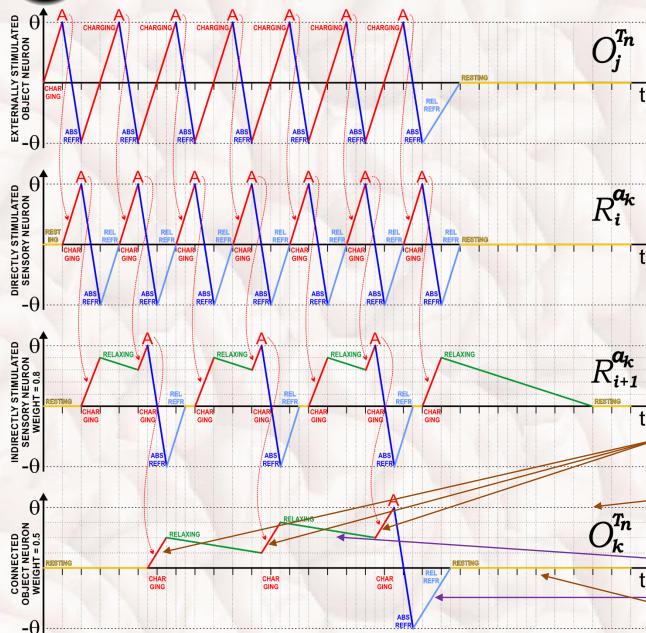


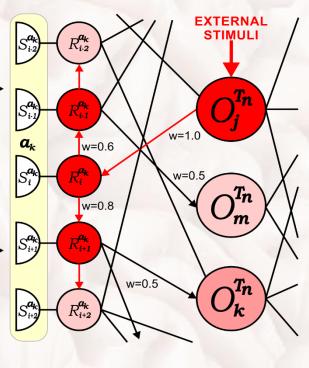
Each activation of the neuron O_j stimulates and activates the neuron R_i which stimulates the neighboring t sensory neurons R_{i+1} and R_{i-1} with the force equal to the weights of these connections, i.e. 0.8 and 0.6, appropriately. Therefore, it is needed to stimulate these neurons twice, so that, with regards to relaxation, they achieve a total stimulus equal to their activation thresholds $\theta = 1$. This will allow them for activation and then to start stimulation of the connected neurons, e.g. the neuron O_k .









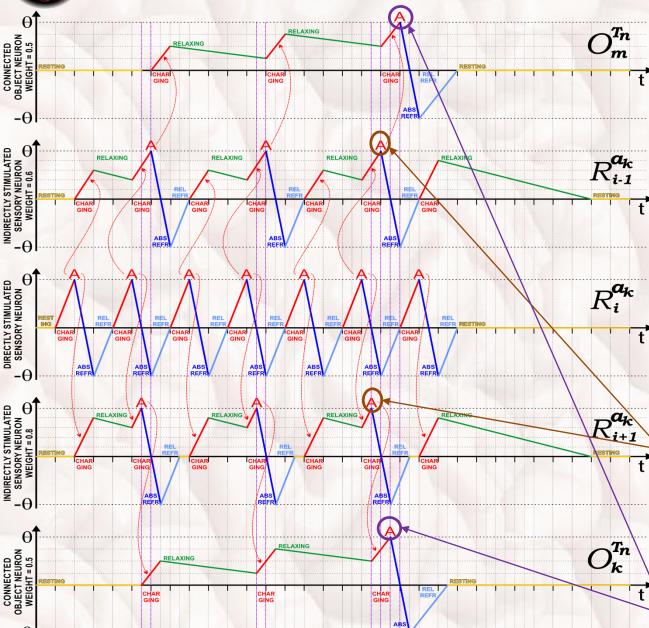


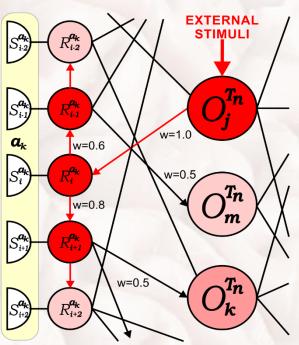
As we can notice, the neuron O_k needs to be **stimulated triple times** through the connection coming from the neuron R_{i+1} and weighted with 0.5 to reach the **activation threshold** $\theta = 1.0$.

When a neuron is not externally stimulated, the **relaxation** and **refraction** processes try to restore the **resting state** in it.







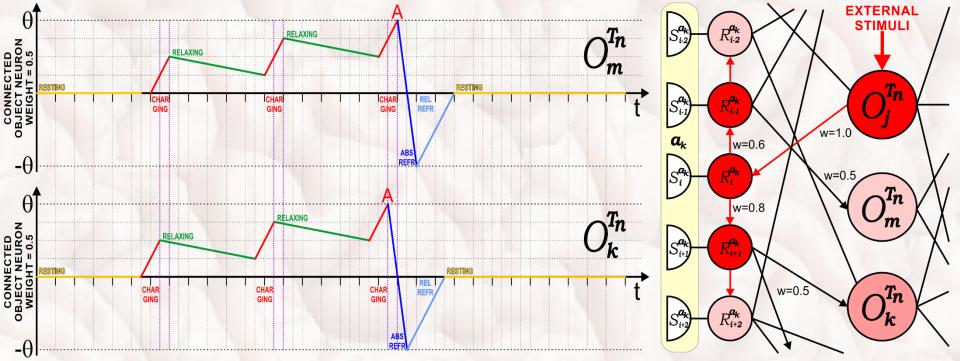


The sensory neurons R_{i+1} and R_{i-1} are stimulated with different strength according to the weights (0.8 and 0.6) of connections coming from the neuron R_i . It induces different **excitation levels** in them and **different activation moments**. The neuron R_{i+1} achieves this threshold **earlier** than the neuron R_{i-1} , so the neuron R_{i+1} starts **earlier** to stimulate the neuron O_k than the neuron R_{i-1} starts to influence the neuron O_m . Thus, the neuron O_k will be activated **earlier** than the neuron O_m . It implies **greater similarity** of the object represented **by the neuron** O_k than **by the neuron** O_m . This is consistent with **intuition** of the real **similarity**.





The small shift in activation of the neurons O_k and O_m may seem to be insignificant or negligible, but this phenomenon is crucial for the working way of biological neural networks as well as of the introduced **associative neural graphs (e.g. AANG, DASNG, ANAKG or AAS)**. **The difference in activation time of these neurons** representing different objects informs us of **weaker and stronger associations** with these objects, i.e. **less or greater similarity** of them. In general, these time differences determine **cognitive processes in the human brain** like mental, motor, or sensory responses... influencing the behavior of the whole network. In this way, **associative neurons** automatically **conclude**, revealing their various relationships with other objects and data represented by other directly or indirectly connected neurons.

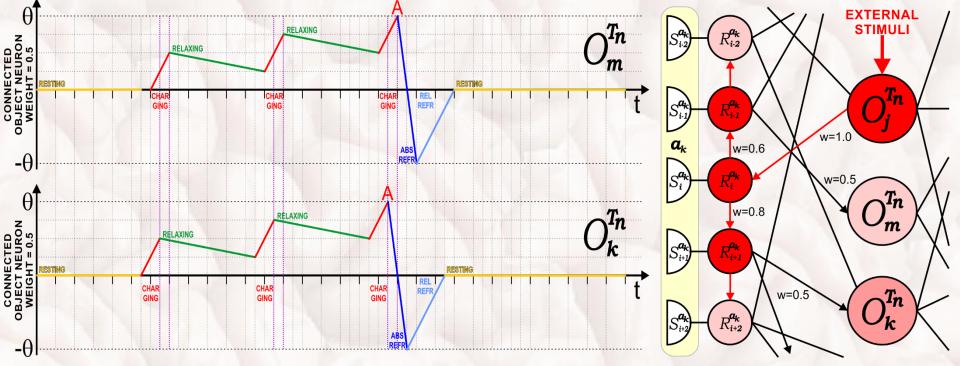






Looking for answers about similarities, groups of similar patterns to a given pattern, any subset of patterns, any given set of features, or any combination of their ranges, it is enough to stimulate the appropriate neurons or sensors, and just wait for activations of neurons, which **activation moments** determine the network answers.

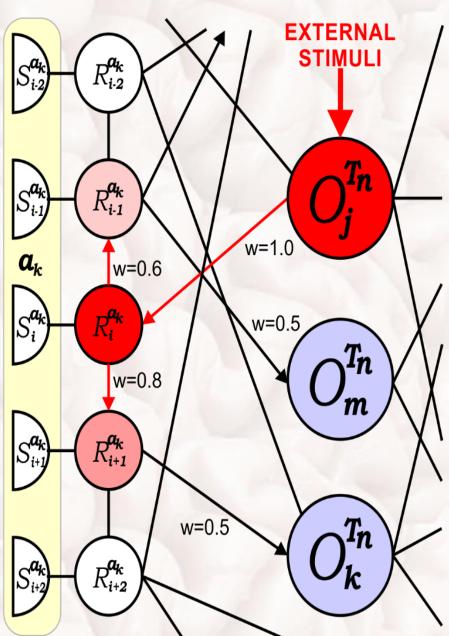
The chronology of neuronal activity automatically points out similar objects or their groups (in clusterization problems), missing features or component objects (in cognition problems), or indicates which classes they belong to (in classification problems). Neurons can therefore automatically explore the knowledge from the relations represented by connections and from objects represented by other neurons.





GENERALIZATION OF CONSIDERATIONS

This very simple (trivial) example does not introduce the full scope of the inference, applicability or possible combinations of neuronal activities in the whole network (graph)! In the whole **associative neural graph**, such an externally stimulated and activated object **neuron** will simultaneously inference with several sensory neurons which will subsequently stimulate various object neurons. The activated object neuron can also have direct connections to other object neurons (representing associative succession or defining) and thereby stimulate them. It is very tough to precisely, clearly, and sequentially describe all these parallelly running processes, it must be seen! All these processes run parallelly in the brain, so the **inference is very fast** in comparison to the classic inference methods of computational intelligence or knowledge engineering, where you must repeatedly search through tables, their elements, transaction, join data together etc.







Using the knowledge of how **associative neurons** behave when representing data and objects, we can try to transform any **relational database** that represents horizontally related objects to the form of a **deep associative neural graph DASNG**.

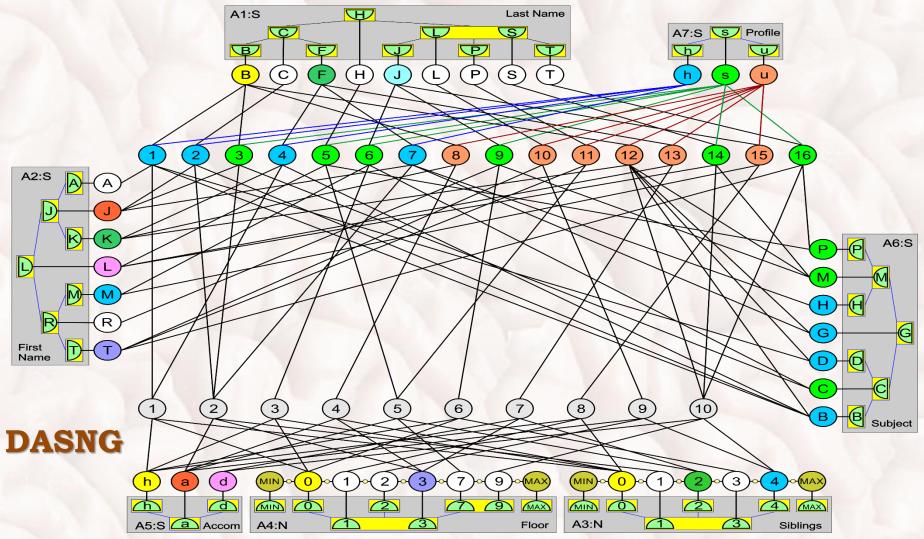
	TABLE D: Likes				TABLE A: Pupils						TABLE B: Live with/in				
		Foreign	Foreign		Primary	A1:S	A2:S	Foreign	Foreign		Primary	A3:N No	A4:N	A5:S	
		KeyC	KeyA		KeyA	Last Name	First Name	KeyE	KeyB		KeyB	of Siblings	Floor	Accommodation	
		9 1	1 -		• 1	Brown	Amy	1	1 🗣		• 1	0	0	house	
TABLE C: Su	bjects	• 1	2 🗕	/	• 2	Cruise	Jack	1 🗨	2 🕶	$- \downarrow$	P 2	1	3	appartament	
A6:S	Primary	93	1 🕤		ø 3	Brown	Jack	2 🔍	2 🖝	7	9 3	0	0	house	
Subjects	KeyC	• 1	4 🗕		• 4	Ford	Kate	3 🔍	1 🖌		9 4	0	3	appartament	
business	1	2	3 🕤		9 5	Hanks	Luke	2 🍳	5 🖝	\mathcal{A}	• 5	2	7	appartament	
chemistry	2	9 6	12 🥄	\downarrow	• 6	Jolie	Mary	3 🗨	3.		• 6	3	0	house	
drama	3 🌒 🗌	~ 2	6 🗨	$\langle \rangle$	P 7	Ford	Jack	2 🔍	2 🖌		9 7	2	3	dormitory	
geography	4	• 4	7 🖝	\times /	8	Brown	Kate	1 🗨	4 🗸		98	0	1	house	
history	5	• 7	5 🗉	\bigvee	9	Jolie	Rose	2 🔍	6	XL	9	4	2	appartament	
mathematics	6	• 4	12 🔍	K	10	Lopez	Tom	1 🗨	9		P 10	4	9	dormitory	
physics	7 🔨 🏹	6	9 🗨	\wedge	11	Pitt	Tom	3 🔍	5		/				
		• 3	12 🎝		• 12	Brown	Kate	1 🔍	10	() () () () () () () () () ()		TABLE	E: Pr	ofile	
		• 5	7 🗉		13	Ford	Luke	2 🔍	ZČ		Prim	ary KeyE	A	7:S Profile	
	\backslash	• 7	16 🔍		• 14	Smith	Luke	2 🔍	10	$\langle X \rangle \rangle$		1		humanistic	
, in the second s		1 12		\times	15	Jolie	Mary	2 🔸	8		2	2		science	
		6	14 🗨		• 16	Trump	Tom	3 🖝	10			3		unselected	

The presented relational database consists of 4 tables containing data and one link table for representing many-to-many relationships (N:M type). Now, we can ask questions: *Which pupils have similar interests?* OR *Which pupils do live in apartments?*





Deep associative neural graph DASNG represents all **horizontal and vertical relations** which associate data and objects and can be automatically retrieved from the database. They also **aggregate** the representation of all duplicates, occurring in a relational database:





WEIGHTS COMPUTATION FOR DASNG

 $S_{i_1}^{a_k}$

 a_k

 $W_{O_i^{T_n}O}$

 $\boldsymbol{N}_{R_{i_2},O_j^{T_n}}$

 $N_{i,T_1}^{T_n}$

 $N_{j,T_{i}}^{T_{n}}$



 $N_{O_2^{T_2}}$

The weights of connections between neurons are computed after very simple formulas, so they do not need to be stored or updated, but computed before using them:

Orderable sensory neurons are connected, the connections are weighed, and the **weights** are:

$$w_{R_{v_{i}}^{a_{k}}, R_{v_{i}}^{a_{k}}} = 1 - \frac{\left|v_{i}^{a_{k}} - v_{j}^{a_{k}}\right|}{r^{a_{k}}}$$

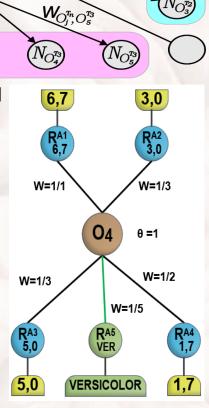
The connections between the sensory and object neurons are **weighted** in the following way:

$$w_{R_{v_i}^{a_k}, O_j^{T_n}} = \frac{1}{\|v_i^{a_k}\|} \qquad w_{O_j^{T_n}, R_{v_i}^{a_k}} = \theta_{R_{v_i}^{a_k}} = 1$$

The weights of synaptic connections between various object neurons are computed on the basis of the number of objects represented by the object neurons of the considered layer of the DASNG, which represents a single database table. If the given object neuron of the considered layer is connected to M object neurons of another layer, then the weight is computed in the following way:

$$w_{O_{j}^{Tn},O_{k}^{Tm}} = \frac{1}{N_{j,T_{m}}^{T_{n}}} \cong \frac{1}{M} \qquad w_{O_{k}^{Tm},O_{j}^{Tn}} = \frac{1}{N_{k,T_{n}}^{Tm}} \cong \frac{1}{N}$$

where $N_{k,T_n}^{T_m} = N = 1$ for the relations one-to-many (1:M) and the relations many-to-many (N:M). The equation is precise when there are no duplicates of the whole records in the database. We need to create separate lists of connections in each neuron to represent connections to neurons of various layers in order to easily compute the number of objects $N_{j,T_m}^{T_n}$ or the number of connections M.



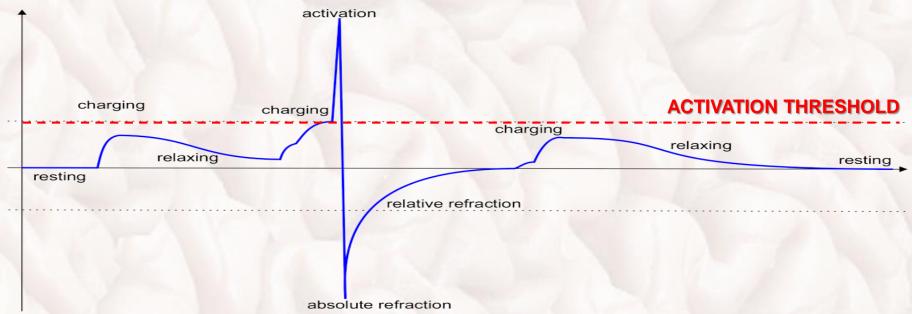
 $W_{O_{j}^{T_{n}},O_{2}^{T_{2}}}$



ACTIVATION THRESHOLDS OF NEURONS



Activation thresholds of neurons play a key role in McCulloch-Pitts' neurons of the first generation, spiking neurons of the third generation and real (biological) neurons while they determine neuronal activity and to which combinations neurons react.



Moreover, activations of the neuron determine which combinations of input stimuli are represented by this neuron.

Therefore, it is very essential to be able to track the state coming from neuronal stimulations with regards to its activation threshold. During simulation, we have no possibility to check all neuronal states constantly, so we need to foresee and compute the predictable time when the neuron achieves its activation threshold.



ACTIVATION THRESHOLDS OF NEURONS



Activation thresholds of sensory neurons are always equal one in this model:

 $\theta_{R_{v_i}^{a_k}} = 1$

Activation thresholds of object neurons are computed according to the following formula:

$$\theta_{O_{j}^{Tn}} = \begin{cases} 1 & \text{if } \sum_{R_{v_{i}}^{a_{k}}} w_{R_{v_{i}}^{a_{k}}, O_{j}^{Tn}} \ge 1 \\ \sum_{R_{v_{i}}^{a_{k}}} w_{R_{v_{i}}^{a_{k}}, O_{j}^{Tn}} & \text{if } \sum_{R_{v_{i}}^{a_{k}}} w_{R_{v_{i}}^{a_{k}}, O_{j}^{Tn}} < 1 \end{cases}$$

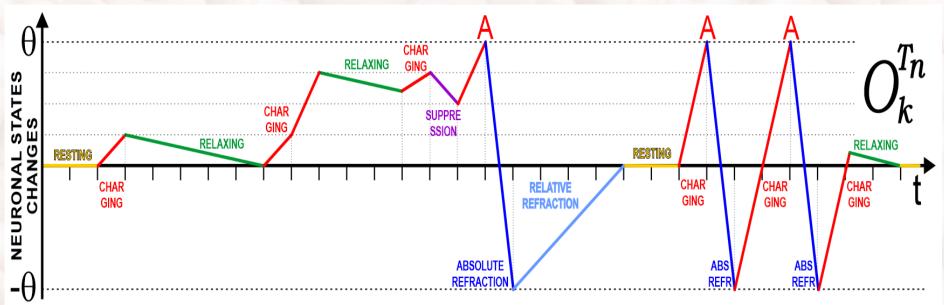
This definition of activation thresholds allows for activation of an object neuron whenever it is stimulated by the **whole defining combination** of this neuron, or when it is stimulated by a **sufficiently representative subset of rare or unique features** defining this neuron, e.g. if a feature defines only one object neuron, then it is enough to recognize it when this feature appears.



LINEAR APPROXIMATION



The DASNG model uses a linear approximation of all processes that take place in Associative Spiking Neurons (ASNs) as it greatly simplifies and speeds up calculations:



Each neuron creates an **internal neuronal process queue (IPQ)** of successive processes sorted after the time of their beginning. New processes are added to this queue on the basis of stimuli coming from other neurons or a sensor.

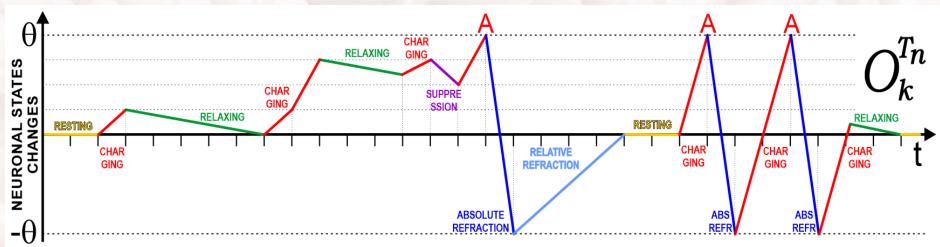
This queue can be modified at any time as a result of the appearance of a new <u>external stimulus</u>, which are appropriately combined (added) with the **charging** or suppression processes already added to this queue. <u>They</u> can also interrupt the relaxation or relative refraction processes or the resting state of the neuron.



INTERNAL NEURONAL PROCESS QUEUE (IPQ)



The use of the **internal neuronal process queue (IPQ)** is necessary because **associative spiking neurons (ASN)** <u>operate over time</u>, so subsequent stimuli and processes must be managed and ordered in time:



Internal neuronal process queue is implemented as a sorter list relative to the start time of the pipelined and ordered processes.

Although neuronal processes may overlap in time (e.g. external stimuli), new processes are added or combined with the existing ones, or they replace them.

As a result, we get a queue of sorter processes that come one after another and do not overlap in time. This way of operating this model significantly simplifies and speeds up all operations, and all results can be updated in the rare discrete moments.

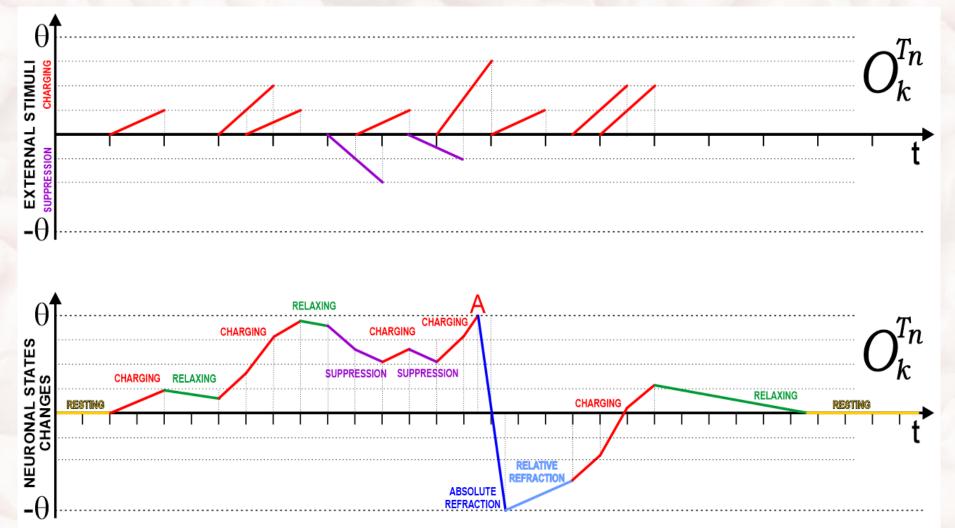
In order to appropriately order parallel processes in time, we use **global event queue** (GEQ) which stores and orders all processes of all IPQs in the DASNG graph.



CREATION OF INTERNAL NEURONAL PROCESS QUEUE



The neuronal process queue is created because of external stimuli that can come to the neuron at different moments and in a varying number depending on the number of neuronal connections and the activity of presynaptic neurons and sensors.



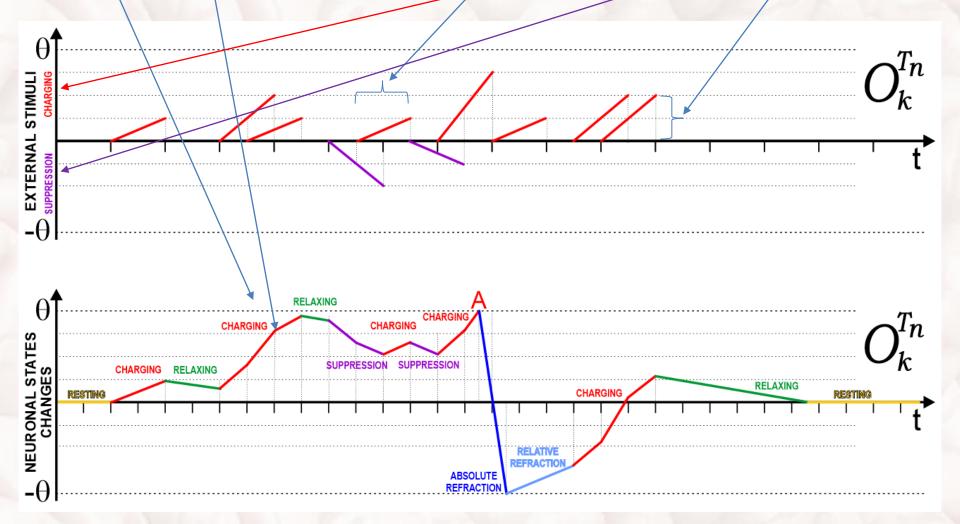


INTERNAL NEURAL PROCESSES



Neuronal processes are thus defined as:

(process type, process start time, process duration, positive or negative process power, the pointer to the event representing this process in the global event queue.





GLOBAL EVENT QUEUE - GEQ



Global Event Queue orders all events related to the neuronal processes in time.

Global Event Queue is responsible for running updating methods in the nodes of the DASNG graphs and switch between the subsequent processes, e.g.:

- After charging finishes, relaxation begins if a neuron is in the excitation state.
- After the activation threshold is achieved, this neuron spikes and starts its absolute refraction process.
- After the absolute refraction process finishes, the relative refraction process is automatically started.
- When the relaxation or relative refraction process finishes, neuron switches to its resting state.

The asynchronous parallelism model is based on the global DASNG event queue, which stores information about the predicted end time of the processes started in various neurons which are not in the resting state.

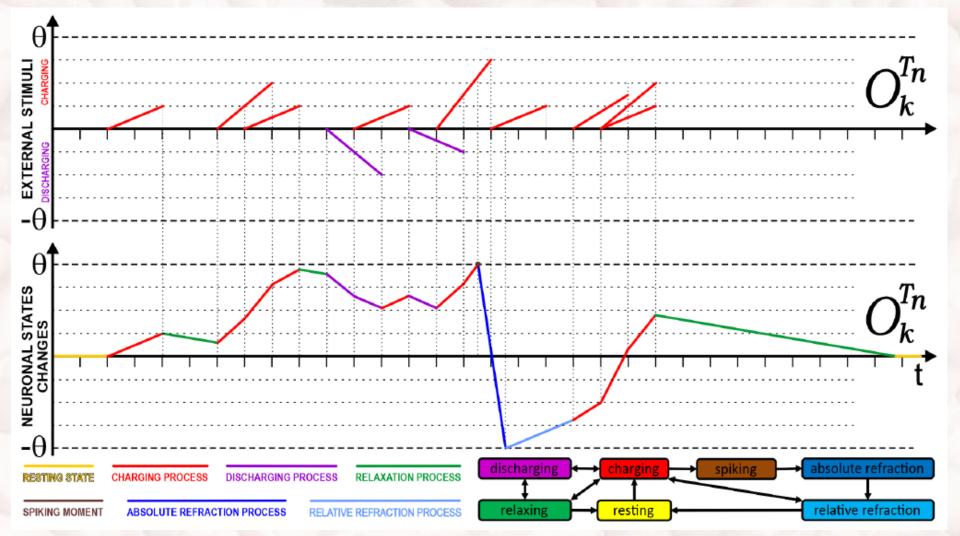
The event in this queue is represented by the pair (end time of the process, pointer), where pointer indicates the DASNG element (e.g. neuron) in which the process is about to terminate. Such a process is the first in the **internal process queue (IPQ)** in this neuron, so we don't need to look for it. Moreover, the IPQs typically consist of a few processes, so the operations on them are very fast!



COMBINING PROCESSES WITH NEW STIMULI



An Internal Process Queue (IPQ) chronologically orders neuronal processes that represent internal changes of a neuron and external stimuli. It avoids collisions of overlapping external stimuli, which are transformed to subsequent processes:





PROCESSES AND EVENTS



A neuronal process is defined as $P_k = (r_k, t_k, d_k, s_k, p_k)$, where:

 r_k - specifies the type of the process: charging (CH), discharging (DC), relaxation (RX), relative refraction (RR), or absolute refraction (AR),

- t_k the starting time of the process,
- d_k the duration of the process (a given period of time),

 s_k - the strength of the process = relative neuronal change after the finished process,

 p_k - a pointer to an event in the global event queue (GEQ) that tracks the end of this neuronal process and launches the neuron update.

An event is defined as an ordered pair $E_n = (t_n, p_n)$, where:

 $t_n = t_k + d_k$ - is the end time of the process that should be finished and switched to another one or to a resting state, and the indicated neuron appropriately updated,

 p_n - a pointer to the updated neuron which the current process should be finished. All events from the entire neural network triggered by the internal neural processes are chronologically ordered in the **global event queue (GEQ)** after their end time.

Sometimes some events become to be outdated when an internal process queue is updated under the influence of new external stimuli. The outdated events E_n are automatically removed from the GEQ by the processes which indicate them (p_k) , and usually swapped to new ones watching the ends of the new processes.



ACHIEVEMENT OF THE SPIKING THRESHOLD



Some charging processes can achieve a spiking (activation) threshold or a resting state during their run, so it is necessary to check all charging processes for such an eventuality before addition of a new process event to the GEQ:

If during the neuron charging process the condition $X_{t_s} + s_s > \theta$ is satisfied, then the time $t^{SP} = t_s + d_s \cdot \frac{\theta - x_s}{s_s}$ of achievement of the neuron spiking (activation) threshold must be calculated to correctly set a watching event $E_n = (t^{SP}, p_n)$ to the GEQ. If during the neuron discharging process the condition $X_{t_s} + s_s < 0$ is true, then the time of achievement of the neuron resting state must be calculated in the following way $t^{RS} = t_s + d_s \cdot \frac{x_s}{-s_s}$, and the appropriate event $E_n = (t^{SP}, p_n)$ put into the GEQ. In the other cases, the state of the neuron is updated at the end of the charging or discharging process after:

$$X_{t_s} = X_{t_0} + s_0 \cdot \frac{t_s - t_0}{d_0}$$

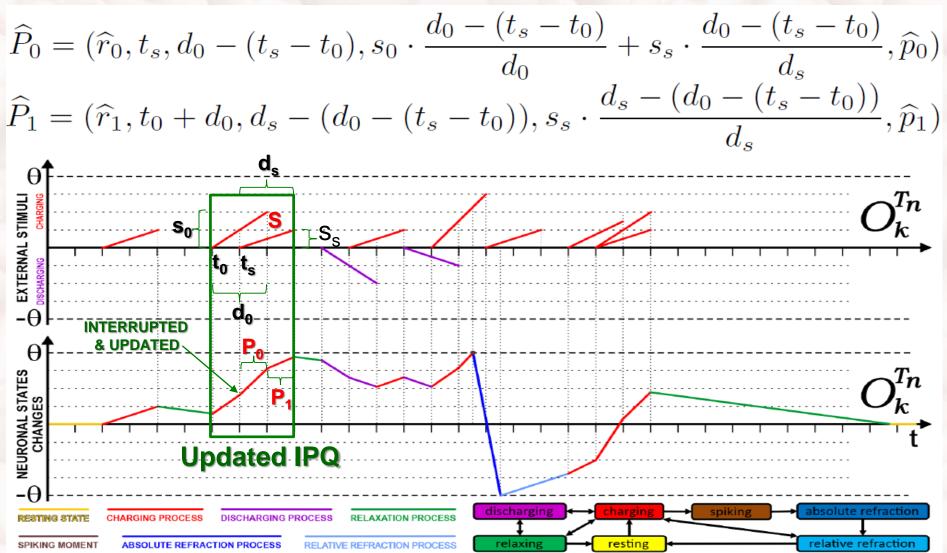
If the neuron achieves it spiking threshold then the IPQ is cleared of all remaining processes and an absolute refraction process is added to the IPQ. During this process the neuron does not react to any further stimuli:

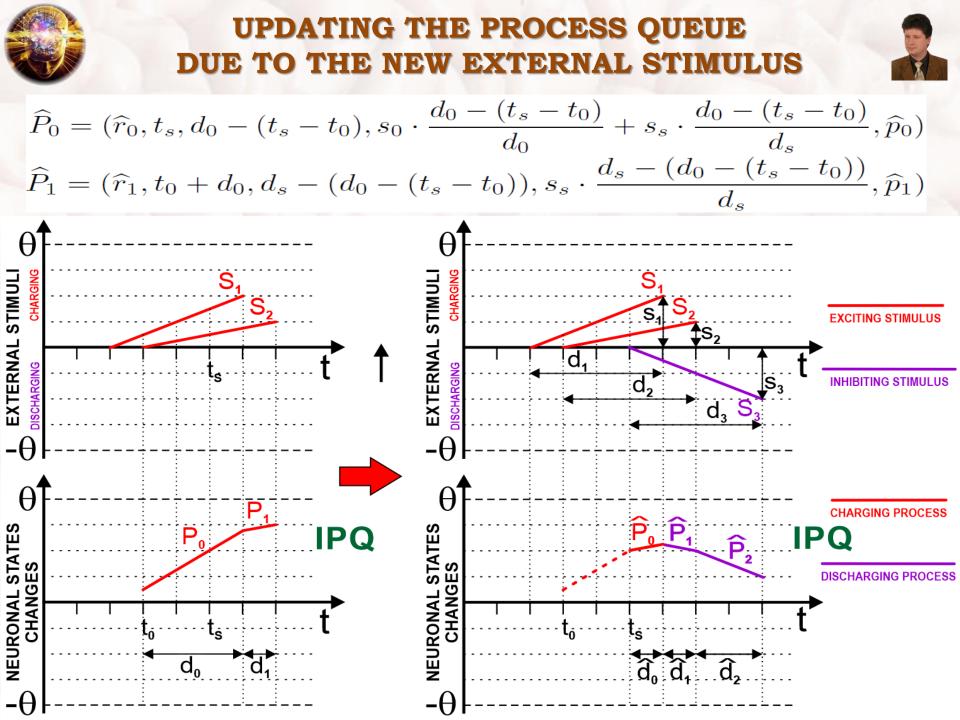
$$\boldsymbol{P}_{AR} = \left(AR, t^{SP}, \boldsymbol{1}, -2 \cdot \boldsymbol{\theta}, \boldsymbol{p}_{AR}\right)$$



CREATING AND UPDATING THE PROCESS QUEUE AFTER THE EXTERNAL STIMULI

The processes in the IPQ are automatically added or removed for each new stimulation $S = (t_s, d_s, s_s)$ which overlap some processes in the IPQ:







DURATION OF RELAXATION AND RELATIVE REFRACTION



The duration of the relaxation process d^{RX} depends on the current state of the neuron X, its spiking threshold θ , and on the assumed maximum relaxation period $p^{RX} = 10$:

$$d^{RX} = \frac{p^{RX} \cdot X_{t_0}}{\theta}$$

The duration of the relative refraction process d^{RR} depends on the state of the neuron X, its spiking threshold θ , and on the assumed maximum relative refraction period p^{RR} = 5:

$$d^{RR} = -\frac{p^{RR} \cdot X_{t_0}}{\theta}$$



TRANSFORMATION ORDER OF TABLES



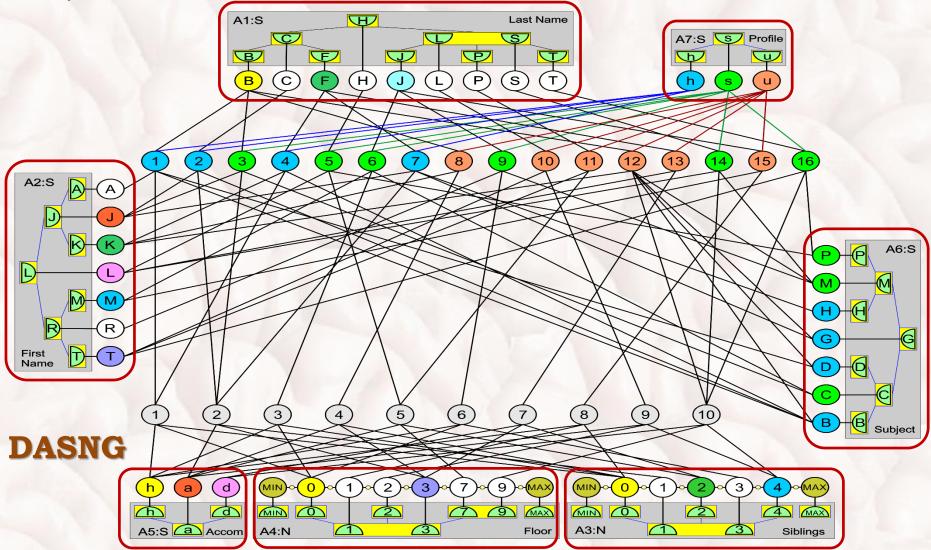
During the transformation of the relational database, note that only the tables which all foreign keys are already represented in the DASNG structure can be transformed. On the other hand, they have to wait until all their foreign keys will be transformed during the associative transformation of other tables. So the sequence of transformed tables is important and the tables must be transformed in an appropriate order as we can see in the figure below and our sample tables. This associative transformation of database tables can be performed also to the passive AGDS structures.

TABLE D: Likes					TAE	BLE A: Pupi	s			TABLE B: L			Live with/in	
	Foreign	Foreign		Primary	A1:S	A2:S	Foreign	Foreign		Primary	A3:N No	A4:N	A5:S	
	KeyC	KeyA		KeyA	Last Name	First Name	KeyE	KeyB		KeyB	of Siblings	Floor	Accommodation	
) 1	1 🗕		• 1	Brown	Amy	1	1 🗕		• 1	0	0	house	
TABLE C: Subjects	• 1	2 🗕	-	• 2	Cruise	Jack	1 🗨	2 🗕	$- \not$	P 2	1	3	appartament	
A6:S Primary	9 3	1 💕		ø 3	Brown	Jack	2 🔍	2 🖝	7	93	0	0	house	
Subjects KeyC	• 1	4 🗕	-	• 4	Ford	Kate	3 🔍	1 🖌		9 4	0		appartament	
business 1	2	3 💕		9 5	Hanks	Luke	2 🌒	5 🖝	A +	9 5	2		appartament	
chemistre 2	• 6	12 🥄	A	• 6	Jolie	ry ry	3 🔍	3 🖌		96	3	•	house	
drama 3 🗨	- 2 📕	6 🔿		P 7	Ford	<mark>' </mark> k	2 🔍	2		9 7	2	3	dormitory	
geograph 4	-• 4	7 🔹	\times /	8	Brown	Kate	1 🗨	4	$\langle $	98	0	1	house	
history 5	• 7	5 🖌	\vee	9	Jolie	Rose	2 🔍	6	XL	9	4	2	appartament	
mathematics 6	• 4	12 🔍	X	10	Lopez	Tom	1 🔍	9		/ 10	4	9	dormitory	
physics 7	6	9 💕	\wedge	11	Pitt	Tom	3 🔍	5		/				
	• 3	12 🚽		• 12	Brown	Kate	1 🔍	10			TABLE	E: Pr	ofile	
	• 5	7 🕌		13	Ford	Luke	2 🔍	ZÝ		Prima	ary KeyE	A	7:S Profile	
7 16		16 🔍		• 14	Smith	Luke	2 🔍	10	X	🕨 1 <mark>humanist</mark>			humanistic	
• 1		12 🖌	\times	15	Jolie	Mary	2 🗣	8			2		science	
6 14				• 16	Trump	Tom	3 🖝	10 •		S 3 Unselect			unselected	





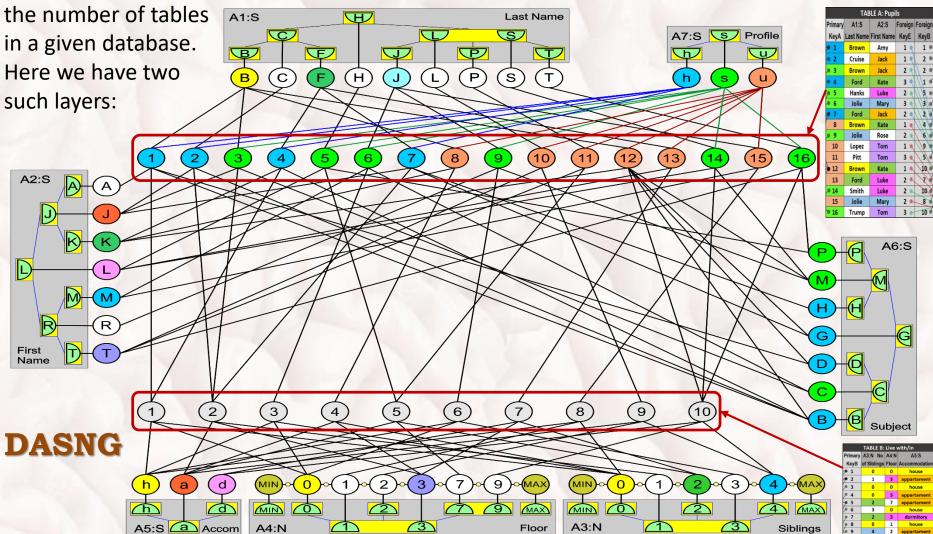
For each attribute separately, the **unique attribute values** are represented by sensors and sensory neurons in this **deep associative graph.** Thanks to AVB-trees, we get access to all data usually in constant time.







The **unique records** of each relational database table containing objects defined by several attribute values and/or foreign keys pointing out records of other tables are represented by a separate **layer** in this **deep associative neural graph**. The number of such layers depends on







KeyA

1

2 •

1 •

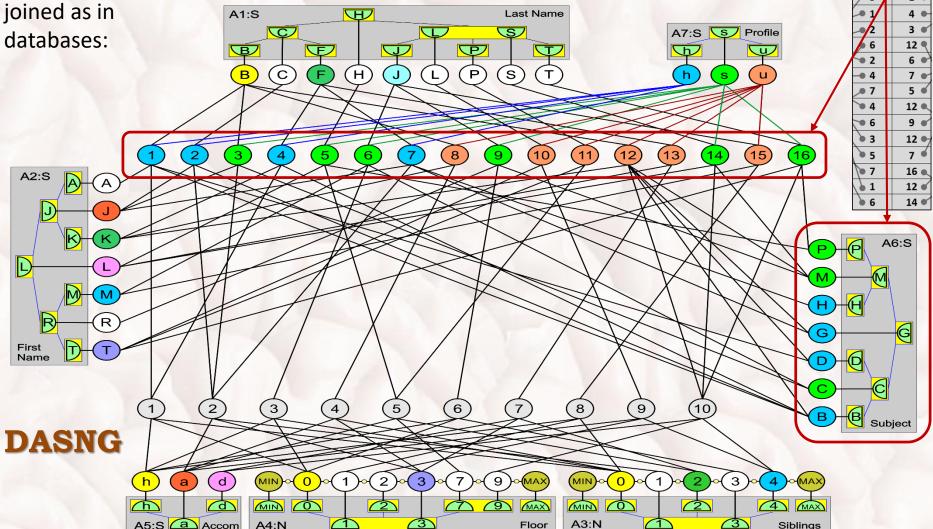
FABLE D: Likes

oreign Foreign KeyC

1

In deep associative neural graphs, there is no need to use link tables representing many-to-many relations (N:M), because they can be replaced by direct connections between neurons representing related objects. Thus, the records do not need to be

databases:







Now, try to answer the question: Which pupils have similar interests? using the DASNG which will response after stimulating appropriate sensors separately. The sensors will stimulate and activate the linked sensory neurons which will then stimulate and activate the

P

P

Last Name

T

S/

s

A7:SSProfile

3

 $\overline{3}$

Siblings

h/

appropriate object **neurons** representing the pupils who are the answer to the given question.

A1:S

B

В

d

h

а

A5:S 🔼 Accom

์MIN

MIN

A4:N

0

 \mathbf{C}

F

H

J

2

1

3

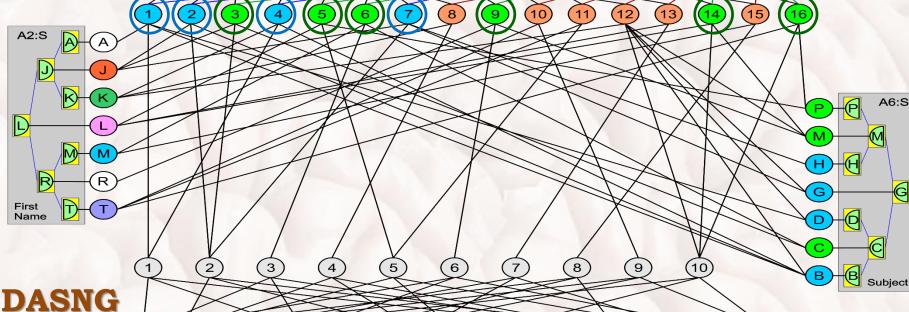
Ĺ

F

A2:S

J)

First Name



9

Floor

MIN

A3:N

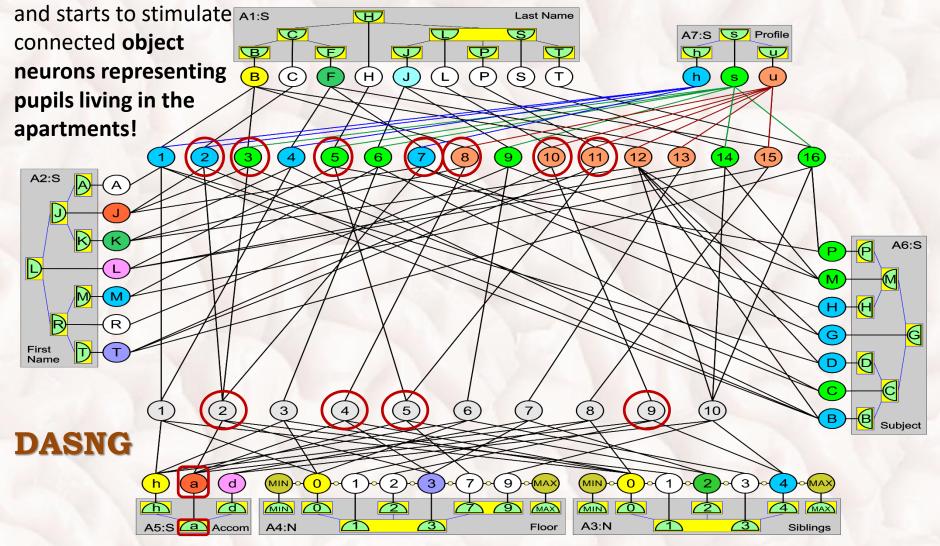
0

7





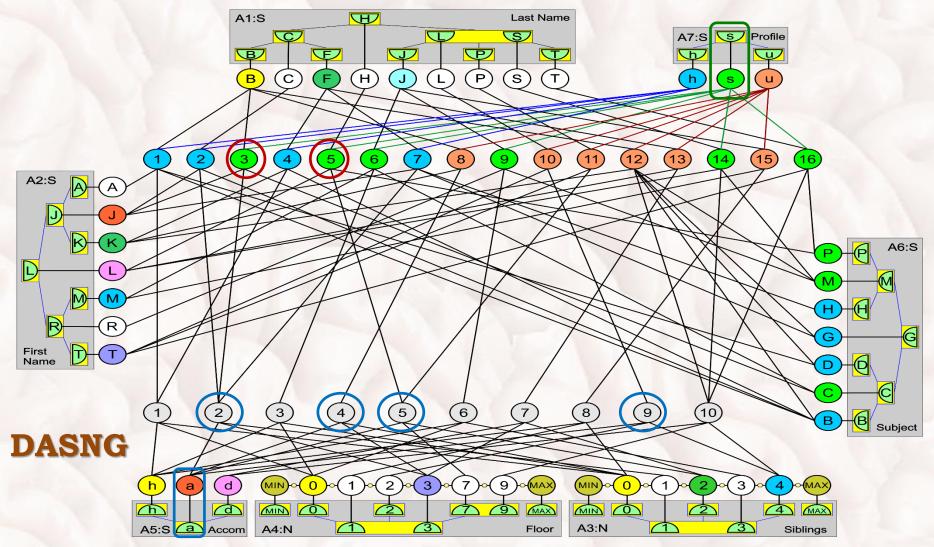
We can also answer the second question: *Which pupils do live in apartments?* by stimulating the **sensor "apartment"**, which stimulates and activates its **sensory neuron** that stimulates **the object neurons representing living conditions**, which are activated after some time,







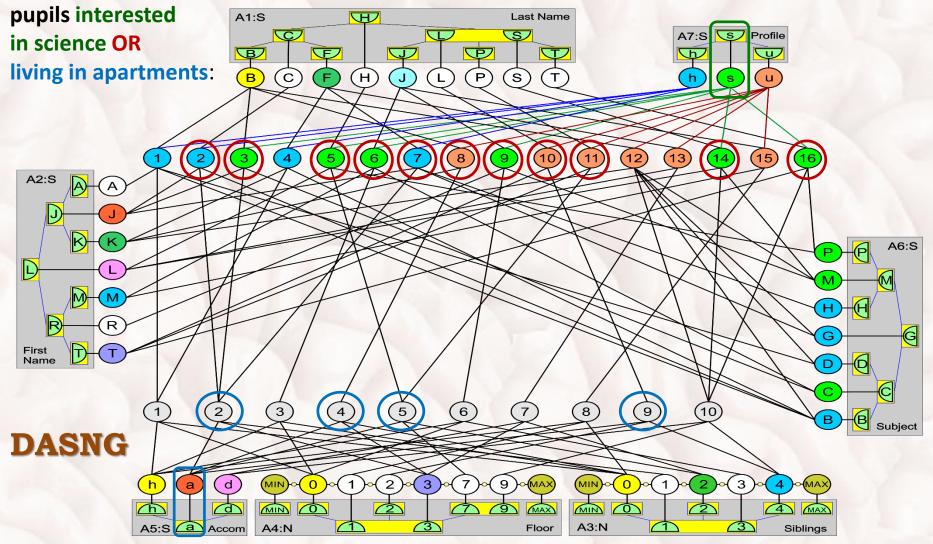
Note that we can also stimulate various combinations of sensors representing the logical conjunction of selected features, which will result in stimulation and the fastest activation of those neurons which represent **pupils interested in science AND living in apartments**.







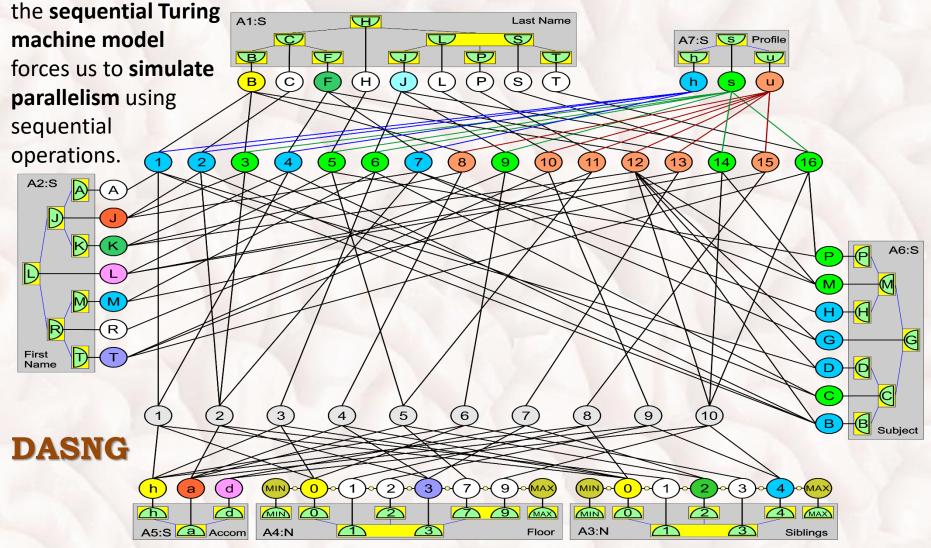
In the case of logical alternative, we wait for the activity of additional pupil neurons longer, where the chronology of activations points out how strong the pupils satisfy the alternative, i.e. the activation moments represent the adaptation degree of pupils to the condition:







Sensors and neurons in such associative graphs can be and should be stimulated in parallel alike in the human brain in order to achieve responses and answers in constant time. Unfortunately, our contemporary computers and computational technology based on





EXPLORATION OF THE KWNOWLEDGE

A5:S Accom

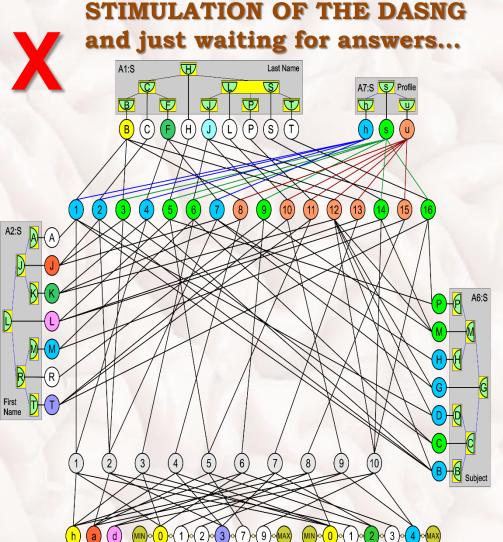
A4:N



We can use tables or deep associative neural graphs for data mining or knowledge exploration:

SEARCHING THROUGH TABLES using classic data mining approaches, calculation of frequent patterns, supports, linking and comparing elements, using ECLAT transformation and Apriori algorithm...

			TABLE D: Likes			TABLE A: Pupils							TABLE B:	: Live with/in	
			Foreign	Foreign		Primary	A1:S	A2:S	Foreign	Foreign		Primary	A3:N No	A4:N	A5:S
			KeyC	KeyA		KeyA	Last Name	First Name	KeyE	KeyB		KeyB	of Siblings	Floor	Accommodation
			•1	1.		• 1	Brown	Amy	1	1 •		• 1	0	0	house
TABLE C: Su	ubjects		• 1	2 🖝	\vdash	• 2	Cruise	Jack	1	2 🗣	-	P 2	1	3	appartament
A6:S	Primary		9 3	1 0	ľ,	ø 3	Brown	Jack	2 🍳	2 🖤	11	93	0	0	house
Subjects	KeyC		•1	4 •	\vdash	• 4	Ford	Kate	3 🌒	1		• 4	0	3	appartament
business	1 📢		• 2	3 0	ſ	9 5	Hanks	Luke	2 🌒	\ \\$ •-	\mathcal{H}	• 5	2	7	appartament
chemistry	2 •	K	ø 6	12		• 6	Jolie	Mary	3 🌒	3		• 6	3	0	house
drama	3 🌒		ו 2	6 🖬		• 7	Ford	Jack	2 🌒	2		9 7	2	3	dormitory
geography	4 🗨	$\mathbb{A}/$	-• 4	7 r	K	8	Brown	Kate	1	4	/	• 8	0	1	house
history	5 🌒	Å	•7	5 0		9	Jolie	Rose	2 🌒	6		0 9	4	2	appartament
mathematics	6 🌒	X	• 4	12 🔍	LK.	10	Lopez	Tom	1 🌒	9		p 10	4	9	dormitory
physics	7 📢	(\uparrow)	• 6	9 💕	\mathbb{N}	11	Pitt	Tom	3 🌒	5		/			
			03	12 🔸	$\left(\right)$	12	Brown	Kate	1 🔍	10	()))()		TABLE	E: Pr	ofile
			05	7 🕯	/	13	Ford	Luke	2 🔍	X		Prim	ary KeyE	ŀ	A7:S Profile
			• 7	16 🔍	V	• 14	Smith	Luke	2 🔍	10	$\langle \rangle \rangle$		1		humanistic
			1	12 🗹	\mathbb{X}	15	Jolie	Mary	2 🖡	8			2		science
			6	14 🗸	ſ	• 16	Trump	Tom	3 🖝	10			3		unselected



Floor A3:N

Siblings

BIBLIOGRAPHY

- A. Horzyk, J. A. Starzyk, J. Graham, <u>Integration of Semantic and Episodic Memories</u>, IEEE Transactions on Neural Networks and Learning Systems, Vol. 28, Issue 12, Dec. 2017, pp. 3084 - 3095, <u>DOI:</u> <u>10.1109/TNNLS.2017.2728203</u>, IF = 6.108.
- 2. A. Horzyk, *Deep Associative Semantic Neural Graphs for Knowledge Representation and Fast Data Exploration*, Proc. of KEOD 2017, SCITEPRESS Digital Library, 2017, pp. 67-79. <u>presentation</u>, <u>DOI: 10.13140/RG.2.2.30881.92005</u>
- A. Horzyk and J.A. Starzyk, Fast Neural Network Adaptation with Associative Pulsing Neurons, IEEE Xplore, In: 2017 IEEE Symposium Series on Computational Intelligence, pp. 339-346, 2017, DOI: 10.1109/SSCI.2017.8285369. presentation, movie Iris-12
- 4. A. Horzyk, Neurons Can Sort Data Efficiently, Proc. of ICAISC 2017, Springer-Verlag, LNAI, 2017.
- A. Horzyk, J. A. Starzyk and Basawaraj, <u>Emergent creativity in declarative memories</u>, IEEE Xplore, In: 2016 IEEE Symposium Series on Computational Intelligence, Greece, Athens: Institute of Electrical and Electronics Engineers, Curran Associates, Inc. 57 Morehouse Lane Red Hook, NY 12571 USA, 2016, ISBN 978-1-5090-4239-5, pp. 1-8, <u>DOI:</u> 10.1109/SSCI.2016.7850029.
- 6. A. Horzyk, <u>Human-Like Knowledge Engineering</u>, <u>Generalization and Creativity in Artificial Neural Associative</u> <u>Systems</u>, Springer-Verlag, AISC 11156, ISSN 2194-5357, ISBN 978-3-319-19089-1, ISBN 978-3-319-19090-7 (eBook), DOI 10.1007/978-3-319-19090-7, Springer, Switzerland, 2016, pp. 39-51.
- Innovative Types and Abilities of Neural Networks Based on Associative Mechanisms and a New Associative Model of Neurons - referat na zaproszenie na międzynarodowej konferencji ICAISC 2015, Springer-Verlag, LNAI 9119, 2015, pp. 26-38, DOI 10.1007/978-3-319-19324-3 3
- **8.** Horzyk, A., How Does Generalization and Creativity Come into Being in Neural Associative Systems and How Does It Form Human-Like Knowledge?, Neurocomputing, 2014, IF = 1,634.
- **9.** Horzyk, A., Human-Like Knowledge Engineering, Generalization and Creativity in Artificial Neural Associative Systems, Springer, AISC 11156, 2014.
- 10. Horzyk, A., <u>Sztuczne systemy skojarzeniowe i asocjacyjna sztuczna inteligencja</u>, EXIT, Warszawa, 2013.
- 11. Tadeusiewicz, R., **Horzyk, A.**, *Man-Machine Interaction Improvement by Means of Automatic Human Personality Identification*, Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen (Eds.), Springer, LNCS 8104, 2013.
- 12. Horzyk, A., Gadamer, M., Associative Text Representation and Correction, Springer Verlag Berlin Heidelberg, LNAI 7894, 2013, pp. 76-87.
- **13.** Horzyk, A., Information Freedom and Associative Artificial Intelligence, Springer Verlag Berlin Heidelberg, LNAI 7267, 2012, pp. 81-89.
- **14.** Horzyk, A., *Self-Optimizing Neural Network 3*, L. Franco, D. Elizondo, J.M. Jerez (eds.), Constructive Neural Networks, Springer, Series: Studies in Computational Intelligence, Vol. 258, 2009, pp. 83-101.
- 15. Dudek-Dyduch, E., Tadeusiewicz, R., **Horzyk, A.**, *Neural Network Adaptation Process Effectiveness Dependent of Constant Training Data Availability*, **Neurocomputing** 72, 2009, pp. 3138-3149, **IF = 1,440**.



Adrian Horzyk

Sztuczne systemy skojarzeniowe i asocjacyjna sztuczna inteligencja

