A Neural-Network Like Logical-Combinatorial Structure of Data and Constructing Concept Lattices

Xenia Naidenova[1], Vladimir Parkhomenko[2], and Sergey Curbatov[3]

[1] Military Medical Academy, Saint-Petersburg, Russia
ksennaidd@gmail.com
[2] Peter the Great St. Petersburg Polytechnic University, Saint-Petersburg, Russia
parhomenko.v@gmail.com
[3] AO NIEVT, Moscow, Russia
kurbatow@yandex.ru

Abstract. An algorithm is proposed to implement the well-known effective inductive method of constructing sets of cardinality $(q+1)$ from their previously constructed subsets of cardinality q. A new neural network-like combinatorial data structure supporting this algorithm is advanced. Some algorithms for constructing concept lattice, inferring good maximally redundant and irredundant classification tests are given using a generalization process based on Galois connections and a direct and backward wave of network activity propagation.

Keywords: Galois lattice, concepts, closed sets, neural networks, classification, diagnostic tests

1 Introduction

There is a great interest in computer sciences to the relation between the Artificial Neuron Networks (ANN) and the Formal Concept Analysis (FCA). The first attempts to relate the FCA and the Neural Networks (NN) were given in [28]. The main goals for using the FCA with respect to the ANN are the following ones: applying concepts lattices for constructing ANN's architecture and to make ANNs to be interpretable. In [5], the FCA is applied for interpretation of neuron codes. In [33,22], the authors apply concept lattices to constructing neural network architecture.

In [11], the authors proposed another possible implementation of building interpretable NN using FCA. Their approach to generating NN architecture is based on constructing covering relation of a lattice with the use of two type of Galois connections: antitone (standard concept lattices) [7] and monotone [4] ones.

In our paper, we do not propose any approach to improve NN architecture. We advance a new configuration of neuron-like logical-combinatorial network to implement the well-known effective inductive method of constructing sets of cardinality $(q + 1)$ from their subsets of cardinality q. This method is applicable to many problems of symbolic machine learning including Concept Lattices construction and, in particularly, generating good classification tests (maximal hypotheses in the FCA and their minimal generators).

Mining logical rules (dependencies) from datasets in the form of association rules, implicative and functional dependencies, and key patterns [3,31] attracts a great interest because of its potential useful application. It has been proven that the problems of implicative and functional dependences inferring (w.r.t. classification problem) are algorithmically equivalent [19]. These problems are viewed as ones of supervised symbolic machine learning based on classification and plausible (commonsense) reasoning.

A universal algorithm (as the studies demonstrate) for inferring logical dependencies is the algorithm using an effective inductive method of constructing sets of cardinality $(q+1)$ from their subsets of cardinality q. A $(q+1)$-set can be constructed if and only if there exist all its proper q-subsets. For example, the algorithms Apriori, AprioriTid, and AprioriHybrid have been presented in [1,30] for association rule mining. Data mining using the Apriori Algorithm is described in [25]. The same principle underlies the algorithm Titanic for generating key patterns [31] and the algorithm TANE for discovering functional dependencies [9]. A level-wise method of $(q+1)$-sets' construction has also been proposed for inferring good diagnostic tests for a given classification or class of objects [17,19]. These tests serve as a basis for extracting functional dependences, implications, and association rules from a given dataset. Discovering frequent closed itemsets and generators (algorithm FCFG) is considered in [27]. This algorithm extracts frequent itemsets in a depth-first search method. Then this algorithm extracts frequent closed itemsets and frequent generator itemsets by a level-wise approach. Algorithm Titanic is used for computing Iceberg Concept Lattice [31]. Also the algorithms of fast iceberg lattice construction are described in [29,32].

A level-wise procedure is also applied in text mining, for example, for extracting association dependencies between words, extracting topic of text and contexts of topic [10,16].

In all enumerated problems, the same algorithm deals with different sets of elements (items, itemsets, attributes, object descriptions, indices of itemsets) and checks the different properties of generated subsets. These properties can be, for example: "to be a frequent (large) itemset", "to be a key pattern", "to be a test for a given class of examples", "to be a good test for a given class of examples", and some others.

If a constructed subset does not possess a required property, then it is deleted from consideration. This deletion reduces drastically the number of subsets to be built at all greater levels. Generally, this algorithm solves the task of inferring all maximal subsets of a set S (i.e., such subsets that cannot be extended) possessing a given PROPERTY. The set S can be interpreted depending on the context of a considered problem. This algorithm implements a level-wise inductive method of $(q+1)$-sets' construction.

A neural network-like combinatorial data structure for constructing $(q+1)$-sets from their q-subsets has been proposed in a number of publications, see, please, for details [21]. In this paper, two algorithms based on this structure are proposed for generating Good Maximally Redundant Tests (GMRT) and Good Irredundant Tests (GIRT) or generators.

Sec. 2 presents the basic definitions of formal concept analysis (FCA) and good test analysis (GTA). Sec. 3 is devoted to the idea of a level-wise algorithm of inferring $(q+1)$-sets of elements from their previously constructed q-subsets of elements. Some special combinatorial networks for this algorithm are discussed in Sec. 4. Some related papers are discussed in Sec. 5.

2 Basic Definitions

Let us recall the main definitions of FCA [7] and GTA [20]. Denote by M the set of attribute values such that $M = \cup\{\mathrm{dom}(attr),\ attr \in U\}$, where $\mathrm{dom}(attr)$ is the set of all values of *attr*, and U is a set of all considered attributes. Let $G = G_+ \cup G_-$ be the set of objects, where G_+ and G_- are the sets of positive and negative objects, respectively. Denote a description of $g \in G$ by $\delta(g)$ and the descriptions of positive and negative objects by $D_+ = \{\delta(g) \mid g \in G_+\}$ and $D_- =\{\delta(g) \mid g \in G_-\}$, respectively.

We give the following Galois mappings $2^G \to 2^M$ and $2^M \to 2^G$, respectively [18]: $\mathrm{obj}(B) = \{g \in G \mid B \subseteq \delta(g)\}$, returning all the objects the descriptions of which include the set B of values, and $\mathrm{val}(A) = \{\mathrm{m} \in \mathrm{M} \mid \mathrm{m} \in \cap\ \delta(g),\ g \in A\}$, returning the intersection of all objects descriptions $\delta(g)$, g belonging to A.

Two closure operators [18] are defined as follows: generalization_of(B) $=$ $\mathrm{val}(\mathrm{obj}(B))$ and generalization_of(A) $= \mathrm{obj}(\mathrm{val}(A))$. A set A is closed if $A = \mathrm{obj}(\mathrm{val}(A))$. A set B is closed if $B = \mathrm{val}(\mathrm{obj}(B))$. Similar mappings denoted by one symbol $(\cdot)'$ are given in [35], see also notation $(\cdot)^\diamond$ in [8] and other notations, for example, in [6,14,23].

The pair (A, B), where A and B are closed, is the formal concept in terms of Formal Concept Analysis and A is called concept extent and B is called concept intent. All formal concepts form Galois lattice (concept lattice) [24]. A triplet (G, M, I), where I is a binary relation between G and M determined by functions $\mathrm{obj}(B)$ and $\mathrm{val}(A)$, is a formal context $\mathbb{K}$.

According to the goal attribute, we get some possible forms of the formal context: $\mathbb{K}_\varepsilon := (G_\varepsilon,\ M,\ I_\varepsilon)$ and $I_\varepsilon := I \cap (G_\varepsilon \times M)$, where $\varepsilon \in \{+,\ -\}$ (if necessary, the value τ can be added to provide undefined objects) [13]. These contexts form a classification context $\mathbb{K}_\pm$.

Definition 1. *A Diagnostic Test (DT) for G_+ is a pair (A, B) such that $B \subseteq M$, $A = \mathrm{obj}(B) \neq \varnothing$, $A \subseteq G_+$, and $\mathrm{obj}(B) \cap G_- = \varnothing$.*

In this connection, it is worth noting that DT in Definition 1 is a special kind of semiconcept in the framework of FCA [15].

Definition 2. *A Diagnostic Test (A, B) for G_+ is maximally redundant if $\mathrm{obj}(B \cup m) \subset A$ for all $m \in M \backslash B$.*

It is worth noting that Definition 2 is equivalent to the definition of positive hypothesis given in [13].

Definition 3. *A Diagnostic Test* (A, B) *for* G_+ *is good iff any extension* $A^* = A \cup i$, $i \in G_+ \setminus A$, *implies that* $(A^*, \mathrm{val}(A*))$ *is not a test for* G_+ *.*

It is worth noting that good maximally redundant test (GMRT) is equivalent to the definition of minimal positive hypothesis given in [13] and GMRTs are formal concepts.

Definition 4. *A Diagnostic Test* (A, B) *for* G_+ *is irredundant if for all* $m \in B$, $(\mathrm{obj}(B \setminus m), B \setminus m)$ *is not a test for positive objects* (*any proper subset of* B *is not the intent for a test for* G_+).

Note that a good irredundant test (GIRT) is not generally a formal concept. If a GIRT is closed, then it is simultaneously a GMRT and it is unique in its class of equivalence with positive hypothesis in [12]. Naturally, GIRT is a minimal generator of a minimal positive hypothesis.

We use a method of GIRT construction based on the following consideration: GIRT is contained in one and only one GMRT equivalent to it with respect to its extent (a set of objects covered by it). So, we shall search for GIRTs contained in a given GMRT.

3 Idea of Level-Wise Algorithm

Let S be a set of some entities. By $s_{\mathrm{q}} = (i_1, i_2, \ldots, i_{\mathrm{q}})$, we denote a subset of S, containing q elements of S. Let $S(\text{prop-}q)$ be the set of subsets $s = \{i_1, i_2, \ldots, i_{\mathrm{q}}\}$, $q = 1, 2, \ldots, nt\text{-}1$, satisfying the PROPERTY. Here nt denotes the cardinality of S. We use an inductive rule for constructing $\{i_1, i_2, \ldots, i_{\mathrm{q}+1}\}$ from $\{i_1, i_2, \ldots, i_{\mathrm{q}}\}$, $q = 1, 2, \ldots, nt$-1. This rule relies on the following consideration: if the set $\{i_1, i_2, \ldots, i_{\mathrm{q}+1}\}$ possesses the PROPERTY, then all its proper subsets must possess this PROPERTY too. Thus the set $\{i_1, i_2, \ldots, i_{\mathrm{q}+1}\}$ in S(prop-(q+1)) can be constructed if and only if $S(\text{prop-}q)$ contains all its proper subsets.

Having constructed the set $s_{\mathrm{q}+1} = \{i_1, i_2, \ldots, i_{\mathrm{q}+1}\}$, we have to determine whether it possesses the PROPERTY or not. If not, $s_{\mathrm{q}+1}$ is deleted, otherwise $s_{\mathrm{q}+1}$ is inserted in $S(\text{prop-}(q+1))$. The algorithm is over when it is impossible to construct any element for $S(\text{prop-}(q+1))$.

The Background Algorithm has an essential disadvantage consisting in the necessity to generate all subsets $\{i_1, i_2, \ldots, i_{\mathrm{q}+1}\}$ from $\{i_1, i_2, \ldots, i_{\mathrm{q}}\}$, $q = 1, 2, \ldots, nt$-1. In next section, we consider a neuron-like combinatorial structure that makes this construction more effective.

4 Special Logical-Combinatorial Network for Background Algorithm

The idea of the following algorithm is based on the functioning of a combinatorial network structure, whose elements correspond to subsets of a finite set S

generated in the algorithm. These elements are located in the network along the layers, so that each q-layer consists of the elements corresponding to subsets the cardinality of which is equal to q. All the elements of q-layer have the same number q of inputs or connections with the elements of previous (q-1)-layer. Each element "is excited" if and only if all the elements of previous layer connected with it are active. The weight of connection going from the excited element is taken as equal to 1; the weight of connection going from the unexcited element is taken as equal to 0. An element of q-layer is activated if and only if the sum of weights of its inputs is equal to q. The possible number Nq of elements (nodes) at each layer is known in advance as the number of combinations of S on q. In the process of the functioning of the network, the number of its nodes can only diminish.

An advantage of this network consists in the fact that its functioning does not require the complex techniques for changing the weights of the connections and it is not necessary to organize the process of constructing q-sets from their (q-1)-subsets. The nodes of network can be interpreted depending on a problem to be solved. The assigned properties can be checked via different attached procedures.

If an activated node does not possess the assigned property, then it is excluded from consideration by setting to 0 all connections going from it to the nodes of above layer. The work of this combinatorial network consists of the following steps:

1. The weights going from the nodes of the first layer is set to 1. For each layer beginning with the second one:
2. The excitation of nodes, if they were not active and all their incoming traffic (links) have the weight equal to 1; checking the assigned property for the activated nodes of this layer;
3. If the assigned property of a node is not satisfied, then all the outgoing connections of this node are established to 0. If the assigned property of a node is satisfied, then its outgoing connections are set to be equal to 1;
4. The propagation of "excitation" to the nodes of the following higher layer (with respect to the current one) and the passage to analyzing the following layer;
5. "The readout" of the nodes not connected with above lying nodes. Such nodes correspond to intents of GIRTs (not extended).

The process of excitation stops if it is impossible to generate the next layer of the network.

The work of the network can be performed by several ways: 1) step by step from lower layers to upper layers; 2) with back propagation: from top to bottom and from bottom to top, simultaneously; 3) in parallel, with decomposition of the network into fragments connected via some nodes.

4.1 Examples of the Network Functioning

Example 1. Generating GMRTs with and without back propagation.

For inferring all GMRTs for G_+, let $S = G_+$. We use the level-wise algorithm, testing the property "to be test" for G_+ ", vertical mode of generating extents of tests, i. e. generating nodes as subsets of objects. An attached testing procedure verifies whether the following property fulfils: PROPERTY(s) $=$ if obj(val(s)) $\subseteq G_+$, then *true* else *false*, where $s \subseteq G_+$.

In Tables 1, 2, the sets of positive and negative object descriptions are given, for our examples.

Table 1. The set D_+ of positive object descriptions

G	D_+
1	m_1 m_2 m_5 m_6 m_{21} m_{23} m_{24} m_{26}
4	m_1 m_4 m_5 m_6 m_7 m_{12} m_{14} m_{15} m_{16} m_{20} m_{21} m_{24} m_{26}
7	m_3 m_4 m_5 m_6 m_{12} m_{14} m_{15} m_{20} m_{22} m_{24} m_{26}
8	m_3 m_6 m_7 m_8 m_9 m_{13} m_{14} m_{15} m_{19} m_{20} m_{21} m_{22}
10	m_2 m_3 m_4 m_5 m_6 m_8 m_9 m_{13} m_{18} m_{20} m_{21} m_{26}

Fig. 1 depicts the network for generating GMRTs for objects G_+ given in Table 1. In Fig. 1, dashed arrows have weight 1, all other arrows have weight 0, double circles presents excited nodes.

All the nodes of two first levels of the network are activated but nodes {4,10}, {7,10}, {1,8}, and {1,10} do not possess the given property and they have no active outgoing links. At the third level, two nodes are activated, but node {4,7,8} does not possess the given property. As a result, we have 2 nodes possessing the given property: {8,10}, {1,4,7}. Only 12 nodes have been checked and 14 ones did not require to be checked.

With testing PROPERTY, we perform the closure operation and find the extended subset of objects corresponding to a node belonging to one of upper levels, then we use a process of back propagation in the network. Then, by this process, the nodes of all lower levels connected with this node will be activated too without checking PROPERTY. Fig. 2 depicts the situation of back propagation for our example. In Fig. 2, the dashed arrows illustrate activation of the node through the closure by objects, all down arrows are established with weight 1 because of backtracking.

Example 2. Extracting good irredundant tests (GIRT) from a good maximally redundant test (GMRT).

Let $X = \{\mathrm{m}_4, \mathrm{m}_{12}, \mathrm{m}_{14}, \mathrm{m}_{15}, \mathrm{m}_{24}, \mathrm{m}_{26}\}$ be the intent of a GMRT. In Table 3, we give the initial set of negative examples for GIRTs extracting from X.

We use the level-wise algorithm, testing the property "not to be test for G_+", horizontal mode of generating intents of GIRTs, i. e. generating nodes as subsets of attribute values. An attached procedure verifies whether the following

Table 2. The set D_- of negative object descriptions

G	D_-
15	$m_3m_8m_{16}m_{23}m_{24}$
16	$m_7m_8m_9m_{16}m_{18}$
17	$m_1m_{21}m_{22}m_{24}m_{26}$
18	$m_1m_7m_8m_9m_{13}m_{16}$
19	$m_2m_6m_7m_9m_{21}m_{23}$
20	$m_{19}m_{20}m_{21}m_{22}m_{24}$
21	$m_1m_{20}m_{21}m_{22}m_{23}m_{24}$
22	$m_1m_3m_6m_7m_9m_{16}$
23	$m_2m_6m_8m_9m_{14}m_{15}m_{16}$
24	$m_1m_4m_5m_6m_7m_8m_{16}$
25	$m_7m_{13}m_{19}m_{20}m_{22}m_{26}$
26	$m_1m_2m_3m_5m_6m_7m_{16}$
27	$m_1m_2m_3m_5m_6m_{13}m_{18}$
28	$m_1m_3m_7m_{13}m_{19}m_{21}$
29	$m_1m_4m_5m_6m_7m_8m_{13}m_{16}$
30	$m_1m_2m_3m_6m_{12}m_{14}m_{15}m_{16}$
31	$m_1m_2m_5m_6m_{14}m_{15}m_{16}m_{26}$
32	$m_1m_2m_3m_7m_9m_{13}m_{18}$
33	$m_1m_5m_6m_8m_9m_{19}m_{20}m_{22}$
34	$m_2m_8m_9m_{18}m_{20}m_{21}m_{22}m_{23}m_{26}$
35	$m_1m_2m_4m_5m_6m_7m_9m_{13}m_{16}$
36	$m_1m_2m_6m_7m_8m_{13}m_{16}m_{18}$
37	$m_1m_2m_3m_4m_5m_6m_7m_{12}m_{14}m_{15}m_{16}$
38	$m_1m_2m_3m_4m_5m_6m_9m_{12}m_{13}m_{16}$
39	$m_1m_2m_3m_4m_5m_6m_{14}m_{15}m_{19}m_{20}m_{23}m_{26}$
40	$m_2m_3m_4m_5m_6m_7m_{12}m_{13}m_{14}m_{15}m_{16}$
41	$m_2m_3m_4m_5m_6m_7m_9m_{12}m_{13}m_{14}m_{15}m_{19}$
42	$m_1m_2m_3m_4m_5m_6m_{12}m_{16}m_{18}m_{19}m_{20}m_{21}m_{26}$
43	$m_4m_5m_6m_7m_8m_9m_{12}m_{13}m_{14}m_{15}m_{16}$
44	$m_3m_4m_5m_6m_8m_9m_{12}m_{13}m_{14}m_{15}m_{18}m_{19}$
45	$m_1m_2m_3m_4m_5m_6m_7m_8m_9m_{12}m_{13}m_{14}m_{15}$
46	$m_1m_3m_4m_5m_6m_7m_{12}m_{13}m_{14}m_{15}m_{16}m_{23}m_{24}$
47	$m_1m_2m_3m_4m_5m_6m_8m_9m_{12}m_{14}m_{16}m_{18}m_{22}$
48	$m_2m_8m_9m_{12}m_{14}m_{15}m_{16}$

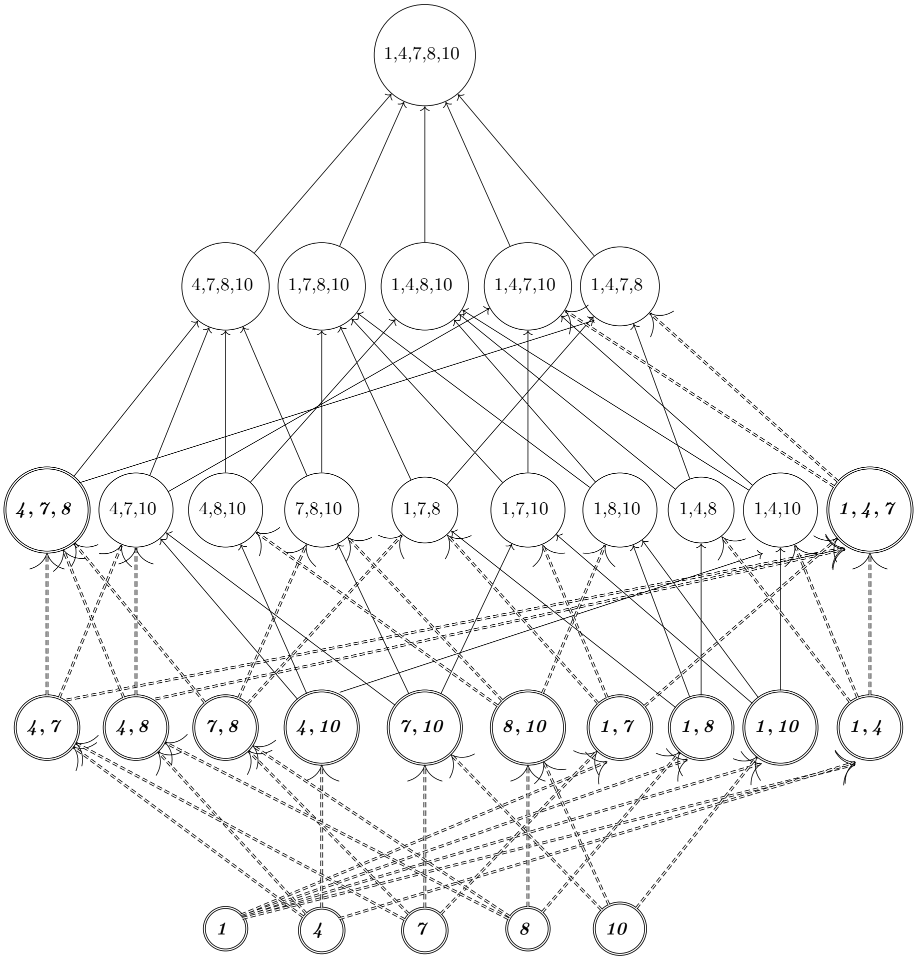

Fig. 1. Fragment of GMRTs Generation

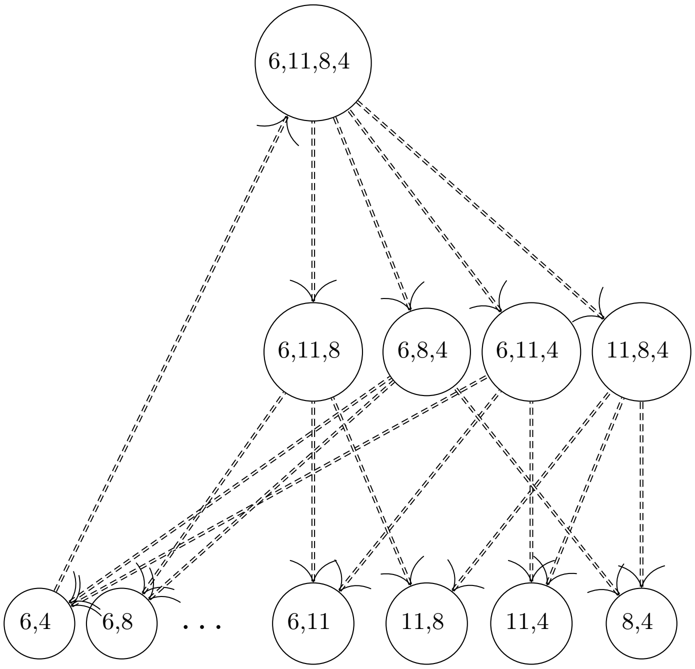

Fig. 2. Example of backtracking in GMRTs inferring

Table 3. The set D_- of negative object descriptions for example 3

G	D_-
17	$m_{24}m_{26}$
23	$m_{14}m_{15}$
30, 48	$m_{12}m_{14}m_{15}$
31	$m_{14}m_{15}m_{26}$
37,40,41,43,44,45	$m_4m_{12}m_{14}m_{15}$
39	$m_4m_{14}m_{15}m_{26}$
42	$m_4m_{12}m_{26}$
46	$m_4m_{14}m_{15}m_{24}$
47	$m_4m_{12}m_{14}$

property fulfils: PROPERTY(s) = if $\exists g$, $g \in G_-$ such that $s \subseteq \delta(g)$ then *true* else *false*, where $s \subseteq S$.

Note that intents of GIRTs consist only of essential values. Recall the definition of essential value and procedure for finding the essential values in any subset t of values.

Definition 5. *Let t be a set of values such that* $(\mathrm{obj}(t), t)$ *is a test for* G_+ *. The value $m \in M$, $m \in t$ is essential in t if* $(\mathrm{obj}(t \setminus m), (t \setminus m))$ *is not a test for* G_+.

Generally, we are interested in finding a maximal subset $sb_{\max}(t) \subset t$ such that $(\mathrm{obj}(t), t)$ is a test but $(\mathrm{obj}(sb_{\max}(t)), sb_{\max}(t))$ is not a test for G_+. Then $sb_{\min}(t) = t \setminus sb_{\max}(t)$ is one of minimal subsets of essential values in t.

In our example, GMRT contains only one essential value: m_{26}, because deleting m_{26} implies that remain part $X \setminus m_{26} = \{m_4, m_{12}, m_{14}, m_{15}, m_{24}\}$ is equal to the description of negative object 46 (see Table 3). It means that value m_{26} will be included in any GIRT. Thus, we need in a configuration of the network containing only the nodes in which m_{26} appears.

A quasi minimal subset of essential values in t can be found by the use of the following procedure.

We begin with the first value m_1 of t, then we take the next value m_2 of t and evaluate the function to_be_test((obj(m_1, m_2), (m_1, m_2))). If the value of the function is *false*, then we take the next value m_3 of t and evaluate the function to_be_test ((obj(m_1, m_2, m_3), (m_1, m_2, m_3))). If the value of the function to_be_test((obj(m_1, m_2), (m_1, m_2))) is *true*, then value m_2 is skipped and the function to_be_test((obj(m_1, m_3), (m_1, m_3))) is evaluated. We continue this process until we achieve the last value of t.

The function to_be_test(t^*), where $t^* \subseteq t$, in this procedure: if $t^* \not\subseteq g$, for all $g \in G_-$ then *true* else *false*.

As a result, we have one of quasi maximal subsets, let $sb_{\max}$, $sb_{\max} \subseteq X$ such that (obj($sb_{\max}$), $sb_{\max}$) is not a test for G_+. Then $L_{\mathrm{ev}} = \{X \setminus sb_{\max}\}$ is a quasi minimal subset of essential values in X. In our illustrative example with only one essential value m_{26} in X, we have the following configuration of the network: Fig. 3, where dashed arrows have weight 1, all other arrows have weight 0, double circles presents intents of GIRTs.

Table 4 depicts the number of nodes in the complete network (C(6,5) + C(6, 4)+C(6,3)+C(6,2)+C(6,1)), where C(n, i) means the combination from n by i.

As a result, we obtain 6 GIRTs: $\{m_2,m_{14},m_{26}\}$, $\{m_4,m_{15},m_{26}\}$, $\{m_4,m_{24},m_{26}\}$, $\{m_{12},m_{14},m_{26}\}$, $\{m_{12},m_{15},m_{26}\}$, $\{m_{12},m_{24},m_{26}\}$.

Construction of initial configuration of networks is beyond the scope of this paper.

Example 3.

The next example: $X = \{m_{19}, m_{20}, m_{21}, m_{22}, m_{26}\}$.

In this case, the only GIRT is ((3,8), $\{m_{19}, m_{20}, m_{21}, m_{22}, m_{26}\}$). We can find essential values in X by means of the procedure described above. Assume that we have find that m_{26}, m$_{22}$, m$_{21}$ are essential in X. Since essential values must enter simultaneously in an intent of a GIRT we can obtain only one node

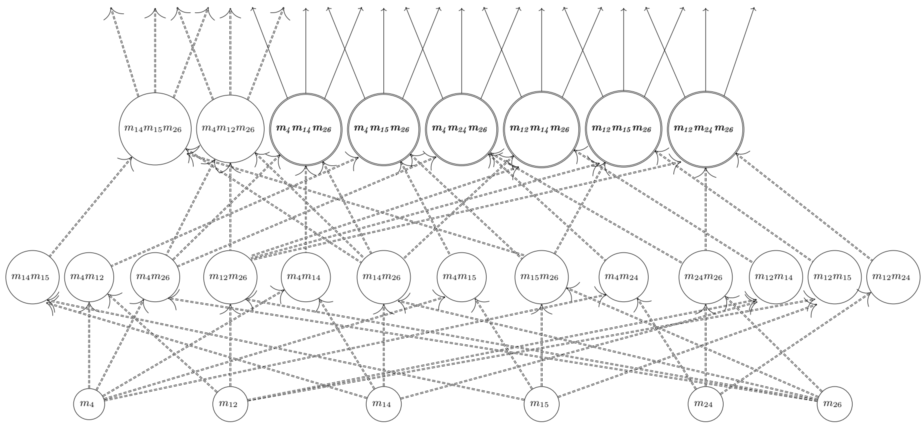

Fig. 3. Network with one essential value m_{26}

Table 4. Comparison the complete and real configurations of networks

Number of nodes in the complete network	Number of nodes in the real network
C(6,5) = 6	0
C(6,4) = 15	0
C(6,3) = 20	8
C(6,2) = 15	13
C(6,1) = 6	6

in the network, containing m_{26}, m_{22}, and m_{21} and it is $(\mathrm{m}_{19}m_{21}m_{22}m_{26})$, which is the GIRT in X.

Apparently, we can see that the size of network may be a problem if the data is large. But the decomposition of the main task into subtasks drastically diminishes the memory size of the algorithm. A subtask is determined by a sub-network generated by a node of the network.

Generally, the main advantages of combinatorial network are the following ones: 1) the size of network is computed in advance; 2) it is possible to decompose network into autonomic fragments; 3) different fragments of network can be joined via common nodes; 4) the states of nodes can be established by the use of attached procedures; 5) this can be used for problems of pattern recognition based on using logical rules [20].

5 Special Logical-Combinatorial Network as a Cognitive Structure

Let us note that the model of perceptron-type neuron with the summing up of the weights of the connections, widely utilized in technical sciences is criticized from the side of specialists, who study the work of brain. This model cannot explain the work of organism's functional systems [2].

Neuron in [34] is defined as conversion of the values of predicates P_1, P_2, P_k, which designate input excitations arriving via axons at the entrance of neuron, into the value of predicate P_0 (output of neuron). It is known that each neuron has a receptive field, whose stimulation excites it unconditionally (without learning).

In the process of learning, neurons obtain the supporting or braking (prohibitive) signals depending on varied conditions and purposes of organism.

A neuron participates in the work of different functional systems and, thus, separate neuron does not have specific fixed semantics. It is assumed that among all input excitations of each neuron there are motivational and emotional excitations, and at each moment of time in the dependence on the purpose and state of organism neuron works within the framework only one functional system.

It is assumed also that there is a mechanism including in the operation of neuron the frequency of its excitation, i.e., the ratio of the number of simultaneous excitation of all its entrances to the number of reinforcements from the side of conditional signals.

It is important for us that the model of neuron, proposed by us, coincides mainly with the new formal model of neuron, described in [34,26].

- neuron network has lattice structure;
- neuron, which is located in the lattice node obtains the excitations from the previous layers and transfers the excitations to the subsequent layers;
- the reinforcement of neuron is realized within the framework a certain functional system and only if some goal is achieved (in our model, if it is carried out a certain purposeful property). Neuron can not transfer excitation with a forbidden signal, i.e., if a certain necessary condition is not satisfied.

We do not deny existence of perceptron-type neuron network. However, neural logical-combinatorial network can interact with it. While the perceptron, as a result of learning, recognizes the objects of some class and gives an answer of the type “yes-no”, neural logical-combinatorial network gives the description of this class of objects in terms of objects’ properties.

6 Conclusion

In this paper, we proposed a neural network-like combinatorial structure of data and knowledge the advantage of which consists in the fact that the functioning of it does not require the complex techniques for changing the weights of connections. The nodes of network can be interpreted depending on a problem to be solved. The assigned properties of nodes can be checked via different attached procedures.

Furthermore, the advantages of combinatorial network are the following ones:

- the size of network is computed in advance;
- it is possible to decompose network into autonomic fragments which can operate in a parallel way;
- different fragments of network can be joined via common nodes;
- this network can be used not only for inferring logical rules from datasets but also for problems of pattern recognition based on these induced rules [20];
- a neuron is activated if and only if its weight reaches a certain value equal to the number of its inputs (the sum of the weights of its inputs) and the excitation in the network is transmitted in the same way as in the artificial neural network.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. Advances in knowledge discovery and data mining 12(1), 307–328 (1996)
2. Anokhin, P.K.: Systemic analysis of neural integrative activity, pp. 347–440. Essays on physiological functional systems, Medicine (1975)
3. Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: Mining frequent patterns with counting inference. SIGKDD Explor. Newsl. 2(2), 66–75 (2000)
4. Düntsch, I., Gediga, G.: Approximation Operators in Qualitative Data Analysis, pp. 214–230. Springer (2003)
5. Endres, D., Foldiak, P.: Interpreting the neural code with formal concept analysis. In: Advances in Neural Information Processing Systems. vol. 21, pp. 425–432. MIT Press, Cambridge (2009)
6. Ganascia, J.G.: TDIS: an algebraic formalization. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence. vol. 2, pp. 1008–1015 (1993)
7. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer, Berlin (1999)

8. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: Delugach, H.S., Stumme, G. (eds.) Conceptual Structures: Broadening the Base: Proceedings of the 9th International Conference on Conceptual Structures. pp. 129–142 (2001)
9. Huhtala, Y., Kärkkäinen, J., Porkka, P., Toivonen, H.: Tane: An efficient algorithm for discovering functional and approximate dependencies. Comput. J. 42(2), 100–111 (1999)
10. Kulkami, M., Kulkami, S.: Knowledge discovery in text mining using association rules extraction. International Journal of Computer Applications 134(12), 30–36 (2016)
11. Kuznetsov, S.O., Makhazhanov, N., Ushakov, M.: On Neural Network Architecture Based on Concept Lattices, pp. 653–663. Springer, Cham (2017)
12. Kuznetsov, S.: JSM-method as a Machine Learning System. Itogi Nauki i Tekhniki, ser. Informatika 15, 17–50 (1991), (in Russian)
13. Kuznetsov, S.: Mathematical aspects of concept analysis. Journal of Mathematical Sciences 80(2), 1654–1698 (1996)
14. Liquiere, M., Sallantin, J.: Structural machine learning with Galois lattice and graphs. In: Proceedings of the 5th International Conference on Machine Learning. pp. 305–313 (1998)
15. Luksch, P., Wille, R.: A Mathematical Model for Conceptual Knowledge Systems. In: Bock, H.H., Ihm, P. (eds.) Proceedings of the 14th Annual Conference of the Gesellschaft fur Klassifikation (GfKl 1990). pp. 156–162 (1991)
16. Mahmood, S., Shahbaz, M., Guergachi, A.: Negative and positive association rules mining from text using frequent and infrequent itemsets. The Scientific World Journal 2014 (2014)
17. Megretskaya, I.: Construction of natural classification tests for knowledge base generation. The Problem of the Expert System Application in the National Economy: Reports of the Republican Workshop. Kishinev pp. 89–93 (1988)
18. Naidenova, X.A.: Searching Good Diagnostic Tests as a Model of Reasoning. In: Proceedings of the Scientific-Practical Conference "Knowledge-Dialog-Solution" (KDS-2001), pp. 501–506. Saint-Petersburg (2001), (in Russian)
19. Naidenova, X.: Machine Learning as a diagnostic task. In: Arefiev, I. (ed.) Knowledge-Dialog-Solution, Materials of the Short-Term Scientific Seminar, pp. 26 – 36. State North-West Technical University Press, St Petersburg, Russia (1992), in Russian
20. Naidenova, X.: Good classification tests as formal concepts. In: Domenach, F., Ignatov, D., Poelmans, J. (eds.) Proceedings of the 10th International Conference on Formal Concept Analysis. vol. 7278, pp. 211–226 (2012)
21. Naidenova, X., Parkhomenko, V.: An approach to incremental learning good classification tests. In: Cellier, P., Distel, F., Ganter, B. (eds.) Contributions to the 11th International Conference on Formal Concept Analysis, pp. 51–64. Technische Universitat Dresden (2013)
22. Nguifo, E., Tsopze, N., Tindo, G.: M-CLANN: Multiclass concept lattice-based artificial neural network. Constructive Neural Networks pp. 103–121 (2009)
23. Nguifo, E.M., Njiwoua, P.: Iglue: A lattice-based constructive induction system. Intelligent data analysis 5(1), 73–91 (2001)
24. Ore, O.: Galois connections. Trans. Amer. Math. Soc 55, 494–513 (1944)
25. Palekar, A., Narwekar, K., Manjeshwar, P., Rane, Y.: Data mining using apriori algorithm. Intern. Journal of Eng. Trends and Technology 28(4), 190–192 (2015)
26. Polyakov, G.: About the principles of neural brain organization. Moscow State Univercity (1965)

27. Rani, U., Premchand, P., Govardhan, A.: A novel algorithm for discovering frequent closures and generators. International Journal on Recent and Innovation Trends in Computing and Communication 3(9), 5484–5487 (2015)
28. Rudolph, S.: Using FCA for Encoding Closure Operators into Neural Networks, pp. 321–332. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
29. Smith, D.T.: A Formal Concept Analysis Approach to Data Mining: The QuICL Algorithm for Fast Iceberg Lattice Construction. Computer and Information Science 7(1), 10–32 (2014)
30. Stumme, G.: Efficient data mining based on formal concept analysis. In: Hameurlain, A., Cicchetti, R., Traunmüller, R. (eds.) DEXA. Lecture Notes in Computer Science, vol. 2453, pp. 534–546. Springer (2002)
31. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with Titanic. Data & Knowledge Engineering 42(2), 189–222 (2002)
32. Szathmary, L., Valtchev, P., Napoli, A., Godin, R.: Constructing Iceberg Lattices from Frequent Closures Using Generators. In: Proceedings of International Conference on Discovery Science, pp. 136–147. Springer (2008)
33. Tsopzé, N., Nguifo, E.M., Tindo, G.: CLANN: Concept lattice-based artificial neural network for supervised classification. In: Proceedings of the Fifth International Conference on Concept Lattices and Their Applications. vol. 331, pp. 153–164 (2007)
34. Vityaev, E.: Knowledge acquisition from data. Cognition via computers. Models of cognitive processes. Essays on physiological functional systems, Novosibirsk State University (2006)
35. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Ferré, S., Rudolph, S. (eds.) Formal Concept Analysis: Proceedings of 7th ICFCA. pp. 314–339. Springer (2009)