Generalized decision directed acyclic graphs and their connection with Formal Concept Analysis

Šimon Horvát[1,*], L'ubomír Antoni[1], Ondrej Krídlo[1], Alexander Szabari[1] and Stanislav Krajči[1]

[1]Institute of Computer Science, Faculty of Science, Pavol Jozef Šafárik University in Košice, 041 80 Košice, Slovakia

Abstract

We propose a generalization of decision jungles from a binary decision directed acyclic graph to a generalized decision directed acyclic graph. We describe the classification method based on a generalized decision directed acyclic graph. Moreover, we explore the properties of our proposed method and illustrate it by example. We present our experiments on several datasets and provide the comparison of our results with related studies. The comparison of the generalized decision directed acyclic graph with the concept lattice of approval for a loan concludes our paper.

Keywords

Classification task, Decision directed acyclic graphs, Concept lattice, Generalization

1. Introduction

Decision jungles (Shotton et al. [30]) provide the compact ensemble method for classification tasks which extend the popular classification methods of decision trees [27, 28, 29, 4] and random forests [5, 6]. In particular, decision jungles represented by the ensemble of decision directed acyclic graphs require less memory in comparison with binary decision trees.

Decision directed acyclic graphs for classification tasks were explored in several recent studies [8, 17, 16, 21, 22, 23, 24, 26] including their connections with operation of convolution. Laptev and Buhmann [21] proposed a novel classification method for images, which is based on so-called transformation-invariant convolutional jungles. Ioannou et al. [17] explored the connections between decision jungles and convolutional neural networks. They proposed the hybrid conditional network which is based on trees and convolutional neural networks. Ignatov and Ignatov [16] proposed an algorithm of decision streams given by tree structure which is generated by merging nodes from various branches based on their two-sample test statistics similarity.

Moreover, the connections between trees structures and Formal concept analysis were recently investigated in [20, 15, 2, 12, 19]. Kuznetsov [20] described the model of learning from positive

Published in Pablo Cordero, Ondrej Kridlo (Eds.): The 16th International Conference on Concept Lattices and Their Applications, CLA 2022, Tallinn, Estonia, June 20–22, 2022, Proceedings, pp. 57–68.

*Corresponding author.

simon.horvat@upjs.sk (Š. Horvát); lubomir.antoni@upjs.sk (L. Antoni); ondrej.kridlo@upjs.sk (O. Krídlo); alexander.szabari@upjs.sk (A. Szabari); stanislav.krajci@upjs.sk (S. Krajči)

0000-0002-3191-8469 (Š. Horvát); 00000000-0002-7526-8146 (L. Antoni); 0000-0001-8166-6901 (O. Krídlo); 0000-0001-5372-1096 (A. Szabari); 0000-0001-5612-3534 (S. Krajči)

and negative examples in the terms of Formal concept analysis. Guillas et al. [15] presented the first structural links between dichotomic lattices and decision trees. Bělohlávek et al. [2] proposed a novel method of induction of decision trees. The concept lattice is seen as the collection of overlapping decision trees in this approach. Links between random forests and pattern structures in Formal Concept Analysis were investigated by Krause et al. [19], as well. Dudyrev and Kuznetsov [12] proposed a combination of concept lattices and decision trees.

In this paper, we propose a generalization of decision jungles from the binary decision directed acyclic graphs to the generalized decision directed acyclic graphs. In Section 2, we present the basic notions of classifications tasks for categorical attributes. Our novel method for the construction of a classifier for a generalized oriented acyclic graph is proposed in Section 3. We present the experiments on our novel method and a comparison of our results with other related studies in Section 4. Connections of our approach with Formal Concept Analysis are illustrated in Section 5.

2. Classification tasks with nominal attributes

In this section, we will formally describe the basic notions of classification tasks with nominal (categorical) input attributes.

Let A be a set and let $b \notin A$. Consider a set L_a for each $a \in A$ and a set L_b for $b \notin A$. Then, $\prod_{a\in A} L_a$ denotes a set of all functions f with a domain A such that $f(a) \in L_a$ for all $a \in A$.

Table 1
Example of training set for loan approval

No.	Income	Savings	Self-Employed	Married	Loan Approved
1	high	high	no	no	1
2	high	high	yes	no	1
3	high	high	yes	yes	1
4	high	high	yes	no	1
5	low	high	no	yes	1
6	low	high	no	yes	1
7	low	high	no	yes	1
8	high	low	no	yes	1
9	high	low	yes	yes	1
10	high	low	yes	no	0
11	low	high	no	no	0
12	low	low	yes	yes	0
13	low	low	yes	no	0
14	low	low	yes	yes	0
15	low	high	yes	no	0
16	average	low	no	no	0

For classification tasks with nominal input attributes, the set A represents the set of input attributes and b is a target attribute. The system of sets $(L_a : a \in A)$ represents the unordered sets of values (or so-called classes, or categories) of input attributes. The set L_b is an unordered set of values (or so-called classes, or categories) of the target attribute. The number of values in L_b is usually low in classification tasks.

In Table 1, the example for $A = \{\text{Income}, \text{Savings}, \text{Self-employed}, \text{Married}\}$ is shown. For $b \notin A$, the target attribute of loan approval is assigned, whereby the sets $L_{\text{Income}} = \{\text{low}, \text{average}, \text{high}\}$, $L_{\text{Savings}} = \{\text{low}, \text{high}\}$, $L_{\text{Self-employed}} = \{\text{yes}, \text{no}\}$, $L_{\text{Married}} = \{\text{yes}, \text{no}\}$, and $L_{\text{LoanApproved}} = \{1, 0\}$ are considered.

For $k \in \mathbb{N}$, a set S given by

$$S = \{(\boldsymbol{x}^1, y^1), \dots, (\boldsymbol{x}^k, y^k)\} \subseteq \prod_{a \in A} L_a \times L_b$$

is called the training set with k objects (also called the labeled objects). In our example from Table 1, we have $k = 16$. In some cases, it can hold that $\boldsymbol{x}^i = \boldsymbol{x}^j$ for some $i, j \in \{1, 2, \dots, k\}, i \neq j$, but $y^i \neq y^j$. It means that we have two people with the same values of input attributes. However, both have different values of the target attribute. This is not the case in our example.

For $m \in \mathbb{N}$ such that $k < m$, a set T given by

$$T = \{\boldsymbol{x}^{k+1}, \dots, \boldsymbol{x}^m\} \subseteq \prod_{a \in A} L_a$$

is called the testing set with $m - k$ objects (also called the unlabeled objects). In Table 2, we illustrate the possible testing set for our example of loan approval.

Table 2
Example of testing set for loan approval

Income	Savings	Self-Employed	Married
high	low	no	no
low	high	yes	yes

A function

$$c : T \to L_b$$

is called a classifier. For its possible fuzzy modification, we can consider the function

$$c : T \to [0, 1]^{L_b}.$$

In the following sections, we will present the construction of a novel classifier based on generalized decision acyclic graphs and describe the possible connections of this classifier with concept lattice in Formal Concept Analysis.

3. Generalized decision directed acyclic graph for classification tasks

Shotton at al. [30] and Pohlen [26] provide the ensemble of classifiers that is based on the rooted binary decision directed acyclic graphs [24]. A rooted binary decision directed acyclic graph is a directed graph with one root node (i.e., in-degree 0), multiple split nodes (in-degree at least 1, out-degree 2), and multiple leaf nodes (in-degree at least 1, out-degree 0).

For constructing a classifier based on a decision directed acyclic graph, Shotton et al. [30] optimize the parameters of the attribute for split node, the numerical threshold for split node, left child of a split node, and right child node of a split node for root and each internal node of the decision directed acyclic graph. They optimize these parameters by LSearch or ClusterSearch algorithms regarding the numerical attributes. Pohlen [26] noted that an attribute map can be used to process the categorical data in this environment, as well. However, it can lead to the growing depth of a binary decision directed acyclic graph.

These approaches inspire us to propose the generalized version of decision directed acyclic graph to process the categorical data without attribute transformation. Moreover, we aim to eliminate the growing depth of a binary decision directed acyclic graph by a generalized decision directed acyclic graph. Finally, we will compare the performance and memory efficiency of our proposed solution with those from [30, 26].

Table 3
Comparison of two types of directed acyclic graphs

	Rooted binary directed acyclic graph [30, 26]		Rooted generalized directed acyclic graph (in our approach)	
Type of node	in-degree	out-degree	in-degree	out-degree
root	0	2	0	≥ 2
internal node	≥ 1	2	≥ 1	≥ 2
leaf node	≥ 1	0	≥ 1	0

First, the definition of a rooted generalized directed acyclic graph follows.

Definition 1. *A rooted generalized directed acyclic graph is a directed graph $G = (V, E)$ such that:*

(1) A unique root node $r \in V$ has zero incoming edges and at least two outgoing edges;
(2) There exist no directed cycles in G;
(3) There exists a directed path from a root node $r \in V$ to each other node $v \in V$;
(4) Each node $v \in V \setminus \{r\}$ has either zero outgoing edges (then it is called a leaf) or has at least two outgoing edges (then it is called an internal node), and at least one incoming edge.

The comparison of a rooted generalized directed acyclic graph with a rooted binary directed acyclic graph applied by [26, 30] is shown in Table 3.

In the following part, we will define a generalized decision directed acyclic graph.

Definition 2. *Let A be a set of n categorical attributes and let $b \notin A$ is a target attribute. A generalized decision directed acyclic graph is a rooted generalized directed acyclic graph $G = (V, E)$ such that*

(1) All directed paths from the root node r to a node v have the same lengths;

(2) The triple (d_v, b_v, U_v) is assigned for the root node and for each internal node $v \in V$, whereby

(a) $d_v \in \{1, 2, \dots, n\}$ is an index of the attribute,

(b) $b_v \in \{1, 2, \dots, |L_b|\}$ is a class index of target attribute b,

(c) $U_v = \{u \in V : (v, u) \in E\}$ is a set of children nodes of v such that $|U_v| \leq |L_{a_{d_v}}|$ and $g_v : U_v \to L_{a_{d_v}}$ is an injective mapping;

(3) A class index $b_w \in \{1, 2, \dots, |L_b|\}$ of the target attribute b is assigned for each leaf node $w \in V$.

In the following subsection, we will describe the algorithm for the generation of the generalized decision directed acyclic graph from a given training set S described in Section 2.

3.1. Algorithm for learning a generalized decision directed acyclic graph

Suppose that we have a training set S and we aim to learn a generalized decision directed acyclic graph from S. By Definition 2, we aim to find a triple (d_v, b_v, U_v) for the root node and each internal node $v \in V$. Moreover, we aim to find a class index b_w for each leaf node $w \in V$.

In Algorithm 1, we describe our proposed procedure for learning a generalized decision directed acyclic graph from a training set S. Due to a limited number of pages, we will explain only the substantial parts of our algorithm in this paper. Additional important notions, the formal descriptions of mappings and their properties will be included in the extended version of our paper.

Without loss of generality, we can suppose that A is a set of attributes, b is a target attribute, and P_i is a set of all parent nodes at some fixed level $i \in I$ (see Figure 1). Let U_p be a set of all child nodes for all $p \in P_i$.

Let S_u denote the set of all labeled objects from a training set S that reaches a node $u \in U_p$. The procedure **AttributeIndexOfNode** assigns the attribute index d_u to u (last but one line of Figure 1). This assignment is based on the entropy function h with a discrete probability distribution $\mathscr{T}^{S_u}$. Note that for $u \in U_p$ and $z \in L_{a_{d_u}}$, a set

$$S_u^z = \{(\boldsymbol{x}, y) \in S_u : x_{d_u} = z\}$$

is a subset of a training set S corresponding to a given child node u and an attribute value z (see Algorithm 1). The procedure **TargetClassIndexOfNode** assigns the class index y_{b_u} of the target attribute b to u (last line of Figure 1). The class index of the target attribute with the maximal number of objects in S_u is selected. If there exist two or more classes with the same maximal number, we consider the class with the least index.

Finally, the parent nodes P_{i+1} for level $i + 1$ are obtained from a system $(U_p : p \in R_i)$ by procedure **MergeChildNodes** with the same attribute and class indices at a level i (Figure 2).

We present the full procedure for learning a generalized decision directed acyclic graph in Algorithm 1. Note that the set of parents nodes that are internal at a level i is denoted R_i.

4. Experiments

We conducted several experiments to provide the performance and memory efficiency of our proposed method in comparison with other recent studies, mainly with binary decision directed

Figure 1: Procedures AttributeIndexOfNode and TargetClassIndexofNode at level *i*

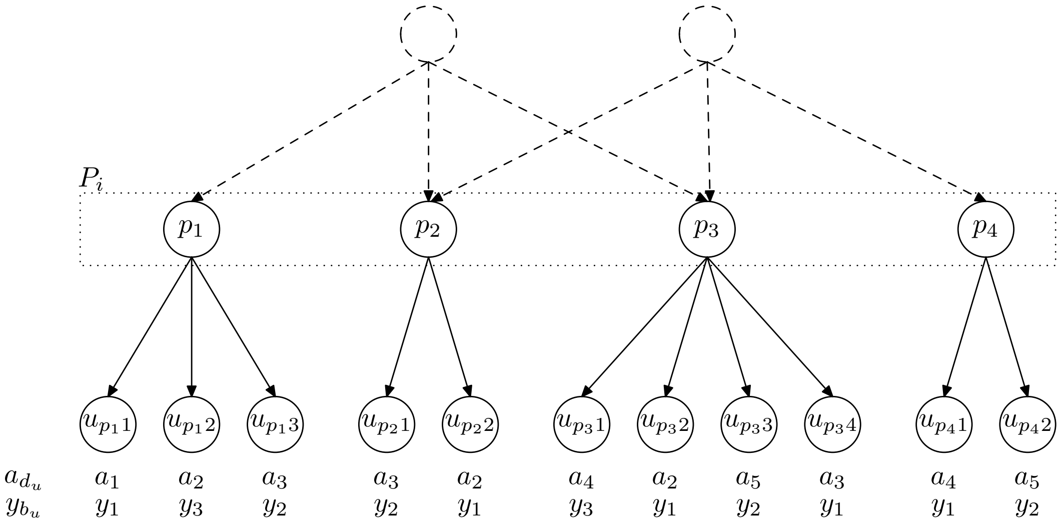
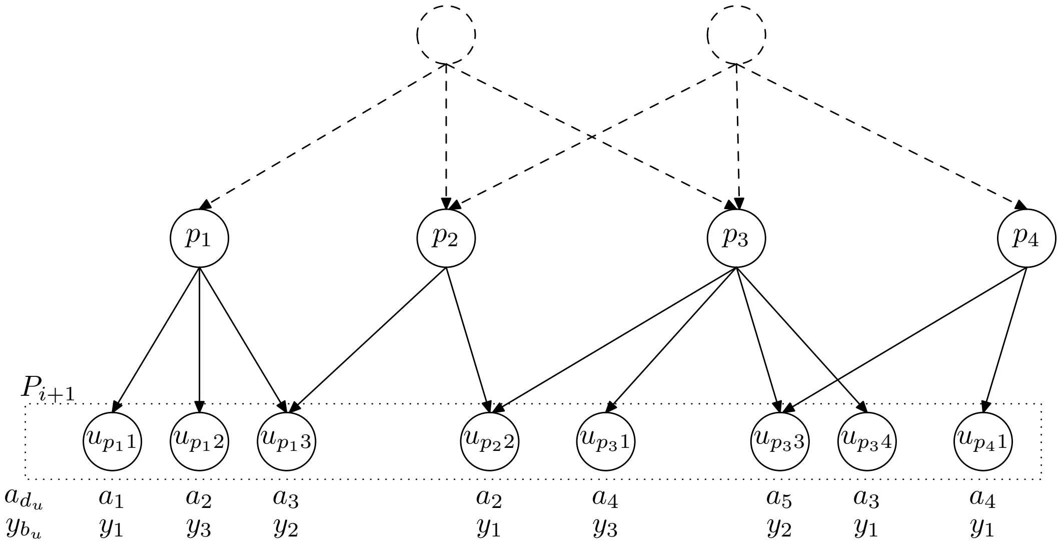

P_i
p_1
p_2
p_3
p_4
$u_{p_1}1$ $u_{p_1}2$ $u_{p_1}3$ $u_{p_2}1$ $u_{p_2}2$ $u_{p_3}1$ $u_{p_3}2$ $u_{p_3}3$ $u_{p_3}4$ $u_{p_4}1$ $u_{p_4}2$
a_{d_u} a_1 a_2 a_3 a_3 a_2 a_4 a_2 a_5 a_3 a_4 a_5
y_{b_u} y_1 y_3 y_2 y_2 y_1 y_3 y_1 y_2 y_1 y_1 y_2

Figure 2: Procedure MergeChildNodes at level *i*


P_{i+1}
p_1
p_2
p_3
p_4
$u_{p_1}1$ $u_{p_1}2$ $u_{p_1}3$ $u_{p_2}2$ $u_{p_3}1$ $u_{p_3}3$ $u_{p_3}4$ $u_{p_4}1$
a_{d_u} a_1 a_2 a_3 a_2 a_4 a_5 a_3 a_4
y_{b_u} y_1 y_3 y_2 y_1 y_3 y_2 y_1 y_1

acyclic graphs and decision forests [30, 26]. For comparison of memory consumption of our method with other studies, we use the estimation of the model sizes which are described in [30, 26]. The complete table of estimated memory consumption for nodes and leaves of various structures and approaches (including our approach) is shown in Table 4.

We tested the performance of our method and its memory efficiency on four datasets (Connect4, Letter Recognition, Shuttle, and USPS) from UCI Machine Learning Repository [10] to compare our results with [30, 26].

The final results of random forest, the ensemble of 15 decision directed acyclic graphs, and our ensemble of 15 generalized decision directed acyclic graphs on memory consumption

Algorithm 1: Procedure for learning a generalized decision directed acyclic graph

```
Data: Set of attributes $A$, target $b$, training set $S$
Result: Generalized decision directed acyclic graph $G$ matching $S$
$i \leftarrow 1$;                                          /Level of graph/
$P_i \leftarrow \{r\}$;                                    /Parent node at level 1/
$d_r \leftarrow$ AttributeIndexOfNode($r$);
$b_r \leftarrow$ TargetClassIndexOfNode($r$);
while $P_i \neq \emptyset$ do
    $R_i \leftarrow \emptyset$;                            /Internal nodes/
    for $p \in P_l$ do
        if $h(\mathscr{T}^{S_p}) \neq 0$ then
            $s \leftarrow |\{z \in L_{a_{d_p}} : S_p^z \neq \emptyset\}|$;   /Number of child nodes/
            $U_p \leftarrow \{u_{pj} : j \in \{1, \dots, s\}\}$;            /New child nodes/
            for $u \in U_p$ do
                $d_u \leftarrow$ AttributeIndexOfNode($u$);
                $b_u \leftarrow$ TargetClassIndexOfNode($u$);
            end
            if $|\{d_u : u \in U_p\}| = s$ then
                $R_i \leftarrow R_i \cup \{p\}$;
            end
        end
    end
    $P_{i+1} \leftarrow$ MergeChildNodes($\{U_p : p \in R_i\}$);
    $i \leftarrow i + 1$;
end
```

and test accuracy are shown in Table 5. For Accuracy, we present the arithmetic mean of the test accuracies which were obtained by ensembles of 15 classifiers with a random 5-fold cross-validation.

5. Connections with Formal Concept Analysis

The generalized decision directed acyclic graph for our loan approval example (see Table 1) is illustrated in Figure 3. Note that the proportion of two numbers inside the nodes corresponds to the proportion of the count of labeled objects by two classes of target attribute (categories 1 or 0). The sets described in the text on the left side of the nodes in Figure 3 correspond to the indices of people (first column in Table 1) with category 1 of loan approval.

Regarding the interesting research results in the area of Formal Concept Analysis [14, 13], the positive examples from a training set S (i.e., the labeled objects with $L_b = 1$ in Table 1) can be considered as (binary) formal context $\langle O, A^{\times} \cup \{\langle b, 1\rangle\}, R\rangle$ such that $1 \in L_b$, $R(o, \langle b, 1\rangle) = 1$, and

Table 4
Estimation of memory consumption for each node

Type of node	Variable	Size in bytes	Total size in bytes
	d_v	4	
	θ_v	8	
Internal node v of tree [5]	l_v	4	16
	d_v	4	
	θ_v	8	
Internal node v of decision	l_v	4	
directed acyclic graph [30, 26]	r_v	4	20
	d_v	4	
	$u_1 \in U_v$	4	
Internal node v of our generalized	neighbor	4	
decision directed acyclic graph	$g_v(u_1)$	4	16
Leaf v (all structures)			$4.|L_b|$

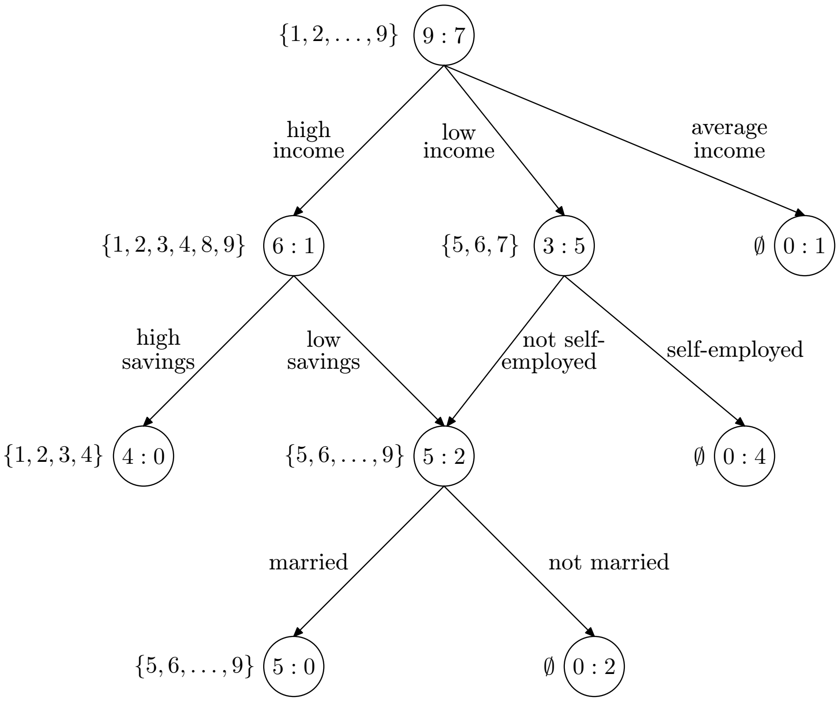
Table 5
Comparison of methods on different datasets

Dataset	Indicator	Random forest [26] (15 trees)	Ensemble of 15 decision directed acyclic graphs [26, 30]	Ensemble of 15 generalized decision directed acyclic graphs (our paper)
Connect4	Accuracy	81.5%	82.0%	82.1%
	Memory size	4.75 MB	1.51 MB	0.22 MB
Letter	Accuracy	95.6%	95.7%	96.3%
Recognition	Memory size	4.89 MB	1.50 MB	0.21 MB
Shuttle	Accuracy	99.9%	99.9%	99.9%
	Memory size	0.03 MB	0.11 MB	0.005 MB
USPS	Accuracy	96.0%	96.0%	90.0%
	Memory size	0.36 MB	0.30 MB	0.035 MB

$$A^{\times} = \bigcup_{a \in A} (\{a\} \times L_a).$$

Note that a conceptual scale [14] or several fuzzy extensions in Formal Concept Analysis [1, 3, 7, 9, 11, 18, 25] can be also applied here.

Figure 3: Generalized decision directed acyclic graph of approval for loan

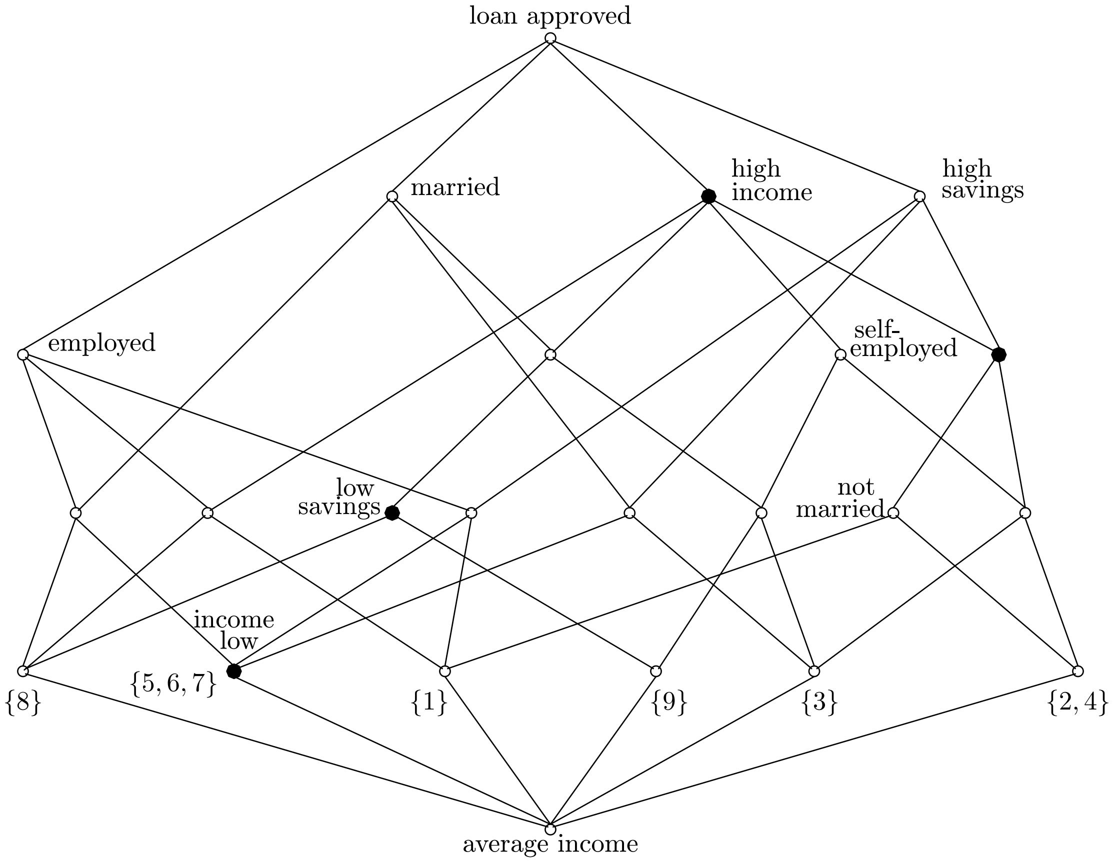


In our example from Table 1, the set of objects O contains people with loan approval equals 1 (rows 1 − 9), and the set of attributes includes pairs $\langle$income, high$\rangle$, $\langle$income, low$\rangle$, $\langle$savings, high$\rangle$, $\langle$savings, low$\rangle$, $\langle$self-employed, yes$\rangle$, $\langle$self-employed, no$\rangle$, $\langle$married, yes$\rangle$, $\langle$married, no$\rangle$, $\langle$loan approved, 1$\rangle$. The corresponding concept lattice of loan approval is shown in Figure 4.

The circles of concept lattice filled with black color (Figure 4) correspond to the nodes of generalized decision directed acyclic graph (Figure 3) with the most of node people who are approved for a loan. For example, the extent of formal concept $\{1, 2, 3, 4\}$ (highlighted by a black circle on the right part of concept lattice) corresponds to the path of high income and high savings in the generalized decision directed acyclic graph. Moreover, the intent of the mentioned formal concept is $\{\langle$income, high$\rangle$, $\langle$savings, high$\rangle\}$, as well. However, the concept lattice provides the possibility to explore other possible groups of people which are not obtained by generalized directed acyclic graphs since the entropy function returns only the best splitting attribute for each node.

The second concept lattice of Table 1 can be constructed by taking the formal context $\langle O, A^{\times} \cup \{\langle b, 0\rangle\}, R\rangle$ such that O is a set of labeled objects with $L_b = 0$. Then, we obtain the formal concepts that can correspond to the paths of people who are not approved for a loan by generalized decision directed acyclic graph.

Figure 4: Concept lattice of approval for loan (own figure)



6. Conclusions and future work

In this paper, we proposed a classification method of generalized decision directed acyclic graphs which is a generalization of decision jungles from the rooted binary directed acyclic graph to rooted generalized directed acyclic graph. In the extended version of our paper, we will present the properties of generalized decision directed acyclic graphs in general. Moreover, we will emphasize the description of the functions used in our algorithm in more detail and their importance for our experimental results. The more detailed connections between generalized decision directed acyclic graphs and Formal Concept Analysis will be explored in the extended version of the paper, as well.

Our future research aims to propose the algorithms for the generalized decision directed acyclic graphs which can be applied for regression tasks with a numerical target attribute. Moreover, another possibility is to apply the genetic algorithms in the search of optimal parameters, optimize the learning process, and improve the performance of algorithms for testing datasets.

Acknowledgments

This work was partially supported by the Scientific Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic under contract VEGA 1/0645/22 Proposal of novel methods in the field of Formal Concept Analysis and their application. This article was partially supported by the project KEGA 012UPJŠ-4/2021. This work was supported by the Slovak Research and Development Agency under contract No. APVV-21-0468.

References

[1] Bělohlávek, R.: Fuzzy concepts and conceptual structures: induced similarities. In: JCIS '98 proceedings, International Conference on Computer Science and Informatics, pp. 179–182. Association for Intelligent Machinery (1998)

[2] Bělohlávek, R., De Baets, B., Outrata, J., Vychodil, V.: Inducing decision trees via concept lattices, Int. J. Gen. Syst. **38**(4), 455–467 (2009)

[3] Ben Yahia, S., Jaoua, A.: Discovering knowledge from fuzzy concept lattice. In: Kandel, A., Last, M., Bunke, H. (eds.) Data Mining and Computational Intelligence, pp. 169–190. Physica-Verlag (2001).

[4] Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: Classification and regression trees. Chapman and Hall/CRC (1984)

[5] Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)

[6] Breiman, L.: Random Forests. Mach. Learn. **45**, 5–32 (2001)

[7] Burusco, A., Fuentes-González, R.: The study of L-fuzzy concept lattice. Mathw. Soft Comput. **3**, 209–218 (1994)

[8] Busa-Fekete, R., Benbouzid, D., Kegl, B.: Fast classification using sparse decision DAGs. In: Proceedings of the 29th International Conference on Machine Learning 2012, 8 pages. Omnipress (2012)

[9] Cordero, P., Enciso, M., Mora, Á., Ojeda-Aciego, M., Rossi, C.: A Formal Concept Analysis Approach to Cooperative Conversational Recommendation. Int. J. Comput. Intell. Syst. **13**(1), 1243–1252 (2020)

[10] Dua, D., Graff, C.: UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science (2019), http://archive.ics.uci.edu/ml. Last accessed 27 March 2022

[11] Dubois, D., Medina, J., Prade, H., Ramírez-Poussa, E.: Disjunctive attribute dependencies in formal concept analysis under the epistemic view of formal contexts. Inf. Sci. **561**, 31–51 (2021)

[12] Dudyrev, E., Kuznetsov, S. O.: Decision Concept Lattice vs. Decision Trees and Random Forests. In: Braud, A., Buzmakov, A., Hanika, T., Le Ber, F. (eds.) ICFCA 2021, LNCS (LNAI), vol. 12733, pp. 252-260. Springer, Heidelberg (2021).

[13] Ferré, S., Huchard, M., Kaytoue, M., Kuznetsov, S. O., Napoli, A.: Formal concept analysis: from knowledge discovery to knowledge processing. In: A Guided Tour of Artificial Intelligence Research, Vol. II: AI Algorithms, pp. 411–445. Springer (2020)

[14] Ganter, G., Wille, R.: Formal Concept Analysis, Mathematical Foundation. Springer Verlag (1999)

[15] Guillas, S., Bertet, K., Ogier, J.M., Girard, N.: Some links between decision tree and dichotomic lattice. In: Bělohlávek, R., Kuznetsov, S. O. (eds.) CLA 2008, Proceedings of the Sixth International Conference on Concept Lattices and Their Applications, pp. 193–205. Palacký University, Olomouc (2008)

[16] Ignatov, D., Ignatov, A.: Decision stream: Cultivating deep decision trees. In: IEEE 29th International Conference on Tools with Artificial Intelligence 2017, pp. 905–912. IEEE (2017)

[17] Ioannou, Y., Robertson, D., Zikic, D., Kontschieder, P., Shotton, J., Brown, M., Criminisi, A.: (2016). Decision forests, convolutional networks and the models in-between, https://arxiv.org/pdf/1603.01250.pdf. Last accessed 27 March 2022.

[18] Krajči, S.: A generalized concept lattice, Logic J. IGPL **13**, 543–550 (2005)

[19] Krause, T., Lumpe, L., Schmidt, S.: A link between pattern structures and random forests. In: Valverde-Albacete, F. J., Trnečka, M (eds.) Proceedings of the 15th International Conference on Concept Lattices and Their Applications, CLA 2020, pp. 131–143. CEUR-WS.org (2020)

[20] Kuznetsov, S.O.: Machine learning and formal concept analysis. In: Eklund, P. (ed.) ICFCA 2004: Concept Lattices, pp. 287–312. Springer, Berlin, Heidelberg (2004)

[21] Laptev, D., Buhmann, J. M.: Transformation-invariant convolutional jungles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3043–3051. IEEE (2015)

[22] Oliver, J. J.: Decision graphs – an extension of decision trees. In: Proceedings of the Fourth International Workshop on Artificial Intelligence and Statistics, pp. 343–350. Fort Lauderdale, Florida (1993)

[23] Peterson, A.H., Martinez, T.R.: Reducing decision trees ensemble size using parallel decision DAGs. Int. J. Artif. Intell. Tools **18**(4), 613–620 (2009)

[24] Platt, J. C., Cristianini, N, Shawe-Taylor, J.: Large margin DAGs for multiclass classification. In: NIPS'99: Proceedings of the 12th International Conference on Neural Information Processing Systems, pp. 547–553. MIT Press, Cambridge (1999)

[25] Pócs, J., Pócsová, J.: Basic theorem as representation of heterogeneous concept lattices, Front. Comput. Sci. **9**(4), 636–642 (2015)

[26] Pohlen, T.: Decision Jungles, In: Seminar Report, Fakultät für Mathematik, Informatik und Naturwissenschaften Lehr – und Forschungsgebiet Informatik (2014)

[27] Quinlan, J. R.: Induction of Decision Trees. Mach. Learn. **1**(1), 81–106 (1986)

[28] Quinlan, J. R.: Unknown attribute values in induction. In: Proceedings of the Sixth International Machine Learning Workshop, pp. 164–168. Morgan Kaufmann (1989)

[29] Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1992)

[30] Shotton, J., Sharp, T., Kohli, P., Nowozin, S., Winn, J., Criminisi, A.: Decision jungles: Compact and rich models for classification. In: NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems, pp. 234–242. Curran Associates Inc. (2013)