Description Quivers for Compact Representation of Concept Lattices and Ensembles of Decision Trees

Egor Dudyrev, Sergei O Kuznetsov, Amedeo Napoli

Description Quivers for Compact Representation of Concept Lattices and Ensembles of Decision Trees

Egor Dudyrev[1,2] [0000−0002−2144−3308]
Sergei O. Kuznetsov[1] [0000−0003−3284−9001]
Amedeo Napoli[2] [0000−0001−5236−9561]

[1] HSE University, Moscow, Russia
[2] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Abstract. In this paper we introduce and study *description quivers* as compact representations of concept lattices and respective ensembles of decision trees. Formally, description quivers are directed multigraphs where vertices represent concept intents and (multiple) edges represent generators of intents. We study some properties of description quivers and shed light on their use for describing state-of-the-art symbolic machine learning models based on decision trees. We also argue that a concept lattice can be considered as a cornerstone in constructing an efficient machine learning model. We show that the proposed description quivers allow us to fuse decision trees just as we can sum linear regressions, while proposing a way to select the most important rules in decision models, just as we can select the most important coefficients in regressions.

Keywords: Formal Concept Analysis · Supervised Machine Learning · Explainable Artificial Intelligence

1 Introduction

Intents and their generators are important tools of Formal Concept Analysis [9]. This paper introduces generators of difference between comparable intents and proposes some ways of using them, such as simplifying the lattice visualization and summation of decision trees.

Formal Concept Analysis (FCA) is a mathematically well-founded theory aimed at data analysis. One of its tools for analysing data consists in providing computable representations of a dataset [6]. These are, e.g., intents (the maximal subsets of attributes describing a specific subset of objects), and their generators (subsets of intents that describe the same subset of objects as intents). This paper generalises the notion of generators to difference generators that describe the same subset of objects as the difference of two comparable intents. Here we study difference generators and provide some of their use-cases.

First, the difference generators help data analysis by presenting more concise line diagrams of concept lattices given by intents. Intents, being maximal

subsets of attributes corresponding to a set of objects often make diagrams too overloaded with text. The well-known solution to this problem is to show only the "new" (w.r.t. to predecessors intents) attributes of intents, but not any of its smaller intents. Difference generators allow to present only the key new attributes of an intent. Thus, they require less text on a line diagram of a lattice of intents and help to highlight the lattice structure.

Difference generators also make it possible to introduce description quivers as a graph theory-based view of a lattice of intents. A description quiver is a directed multigraph, also known as a "quiver", with intents as nodes and difference generators between intents as edges. This graph-theoretic definition proposes a useful interface between FCA and graph theory. Moreover, due to a versatile choice of its edges, description quivers can both copy the "behaviour" of a lattice of intents, or refer only to a subset of attributes. Therefore, description quivers can be studied as mathematical objects on their own. This paper introduces the first results of such a study.

Finally, description quivers can be easily applied in a supervised machine learning scenario, resulting in decision quiver model. This allows us to describe decision trees [5] and their ensembles (e.g. random forest [4]) in terms of FCA. The connections between FCA and decision trees were extensively studied in [1], [2], [3],[7], [10], [11], and [12].

Accordingly, this paper presents a new simpler notation to describe connections between FCA and decision trees. The notation also presents a simple way to combine an ensemble of loosely connected decision trees into one interconnected model. This paper continues the work started in [7] and [8]. It merges these two works and proposes a better mathematical language for describing relations between FCA and decision trees.

The paper is organised as follows. Section 2.1 recalls basic definitions of Formal Concept Analysis (FCA). Then Section 2.2 introduces difference generators and how to use them for providing more concise visualizations of lattices of intents. Section 2.3 introduces and studies description quiver: a directed multidigraph with intents as nodes and difference generators as edges. Section 2.4 introduces decision quivers that allow summation of decision trees. Experimental results presented in Section 3 provide empirical validation of propositions of Sections 2.1 and 2.3. Finally, Section 4 concludes the paper.

2 Theoretical Background

2.1 Formal Concept Analysis

This section recalls basic definitions of Formal Concept Analysis to facilitate their generalizations in the following sections.

As usual, data are given in the form of a **formal context** $K = (G, M, I)$, where G is the set of objects, M is the set of (binary) attributes, and I is a relation between objects and attributes: $I \subseteq G \times M$.

For a given subset of attributes $X \subseteq M$ and an attribute $m \in M$, we often want to consider the cases when attribute m belongs to X : $m \in X$, when

attribute m does not belong to $X : m \notin X$, also represented as $\overline{m} \in X$, and when the presence of attribute m is irrelevant: $m \notin X$ and $\overline{m} \notin X$. Therefore, we define a dichotomised set of attributes $M^{\pm}$ as the union of attributes M and their negations:

$$M^{\pm} = \{m, \overline{m} \mid m \in M\}. \tag{1}$$

Standard **prime operators** are defined as follows: A' gives the maximal subset of attributes (i.e. a **description**) shared by all objects from $A \subseteq G$; analogously, B' gives the maximal subset of objects (that we call an **extent**) shared by all attributes from $B \subseteq M$.

$$A' = \{m \in M^{\pm} \mid \forall g \in A : gIm\}, \quad A \subseteq G \tag{2}$$

$$B' = \{g \in G \mid \forall m \in B : gIm\}, \quad B \subseteq M^{\pm} \tag{3}$$

Formal Concept Analysis studies formal concepts and the partial order (lattice) over them. A formal concept if a pair (A, B) of subsets of objects A and attributes B, such that A is the extent of $B : A = B'$, and B is the **intent** (i.e. maximal description) of $A : B = A'$. To make the notation of this paper concise we will ignore the extents and concentrate only on intents. It is well-known that the set of all intents forms a lattice dual to the lattice of extents:

$$\mathbb{L} = \{B \subseteq M^{\pm} \mid B'' = B\} \tag{4}$$

Each intent $B \subseteq M^{\pm} : B'' = B$ corresponds to many subsets of attributes $D \subseteq M^{\pm}$ with the same closure $B : D'' = B$. Such subset of attributes D is called a **generator** of B. The set of all generators of B gives the **equivalence class** of $B : [B] = \{D \subseteq M^{\pm} \mid D'' = B\}$. A subset of attributes $D \subseteq M^{\pm}$ is called a **key** (or a **minimal generator**) if it is the smallest subset of attributes with closure B: i.e. $\forall E \subset D : E'' \neq D'' = B$. Note that for each intent $B \subseteq M^{\pm}$ there can be multiple (even exponentially many) incomparable keys.

Table 1 presents a formal context that is going to be a running example in the paper. The purpose for the target column Y (so as the reason for using numbers 0 and 1 instead of crosses) will be covered in section 4. This formal context is inspired by the classic "Live in water" formal context.

2.2 Difference generators

The previous subsection describes our motivation to study the keys of set differences between comparable intents. This subsection defines these differences mathematically.

Definition 1. *Subset of attributes D is called a **difference generator** between intents B_2, B_1, such that $B_1 \subseteq B_2$, if its closure is equal to the closed difference between intents B_2 and B_1:*

$$D \subseteq M^{\pm} : D'' = (B_2 \setminus B_1)'', \quad \textit{where } B_1, B_2 \in \mathbb{L}, B_1 \subseteq B_2. \tag{5}$$

		Attributes M				*Target Y*
		l	w	c	h	
	dog	×		×	×	1
Objects	corn	×				0
G	bream		×	×	×	0
	egg					0
Test	reed	×	×			0
objects	sea snake	×	×	×		0
Attr. names		lives on land	lives in water	can move	has limbs	breast feeds

Table 1: Running example of a formal context with additional target column Y and test objects

A trivial example of a difference generator between intents B_2 and B_1 is the difference $B_2 \setminus B_1$ itself. In fact, we are not interested in any difference generator larger than the difference.

Proposition 1. *Subset D of the difference between two comparable intents B_2 and B_1 is a difference generator between these intents if and only if the union of its closure with B_1 is the closure of B_2:*

$$\forall B_1, B_2 \in \mathbb{L}, B_1 \subseteq B_2, D \subseteq B_2 \setminus B_1 : D'' = (B_2 \setminus B_1)'' \iff B_1 \cup D'' = B_2 \quad (6)$$

Proof. Let us derive prove the proposition in two directions:

- From left to right:
  If $D'' = (B_2 \setminus B_1)''$ then $(B_2 \setminus B_1) \subseteq (B_2 \setminus B_1)'' = D''$ and $D'' \subseteq B_2$
  therefore $B_1 \cup D'' = B_2$;
- From right to left:
  First, if $B_1 \cup D'' = B_2$ then $(B_2 \setminus B_1) \subseteq D''$;
  second, if $D \subseteq B_2 \setminus B_1$ then $D'' \subseteq (B_2 \setminus B_1)''$;
  therefore $D'' = (B_2 \setminus B_1)''$.

Note that any generator $D \subseteq M^{\pm}$ of an intent $B \subseteq M^{\pm} : D'' = B$ can be represented as a difference generator between B and the minimal intent $\emptyset''$: $D'' = B \iff D'' = (B \setminus \emptyset'')''$.

Intent differences often used in FCA to shorten the labels of nodes in a line diagram of lattice of intents. That is, instead of presenting the full intent B in a lattice L, one only shows the "new" attributes of the intent, that are not included in smaller intents of the lattice: $B \setminus (\bigcup_{\tilde{B} \in L, \tilde{B} \subset B} \tilde{B})$. However, such set differences can also contain many attributes. So keys, i.e., minimal generators of the differences, can be used.

Figure 1 presents these three ways to label the nodes on a line diagram. The left plot labels the nodes with the full intents, the middle plot only shows new attributes, contained in the intent, and the right plot gives a key of the set of "new" attributes. The lattice in the figure is based on ten intents having the

biggest values of stability lower bound for the Zoo dataset (plus the smallest and the biggest intents). It can be seen that the right plot shows the minimal set of attributes, thus presenting only the most important ones. A qualitative comparison of label lengths is presented in section Experiments.

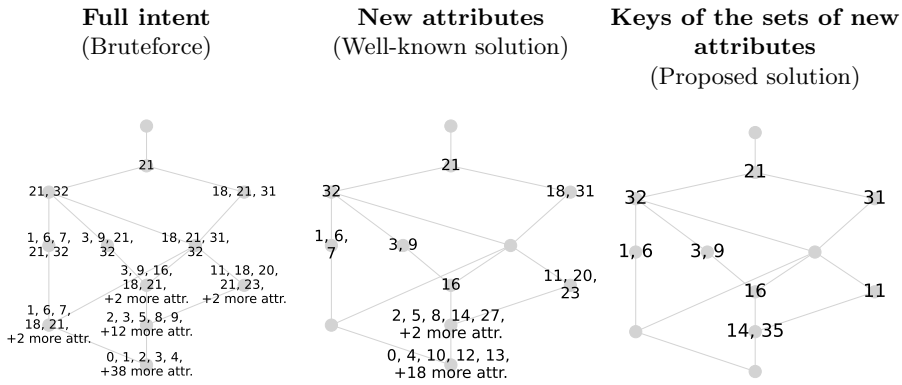

Fig. 1: Three ways to denote intents in a lattice. Given lattice represents ten most stable intents in Zoo dataset (plus top and bottom intents). Numbers stand for indices of binary attributes. The left subfigure visualizes indices of all attributes from intents that makes the diagram almost incomprehensible. In contrast, the right subfigure shows only a few attribute indices that makes it easier to read the diagram and also allows to increase the font size.

3 Description Quivers for unsupervised setting

In the previous section introduced the basic definitions of FCA, the task we solve, and the difference keys aimed at connecting the FCA terms of intents and their keys. This section is devoted to describing description quiver: a graph-like model that merges intents and difference keys. Description quiver is designed to balance the length of intents with the redundancy of equivalent keys.

3.1 Description quiver

Definition 2. ***Description quiver*** (L, E) *is a pair of subset of intents* L *and a subset of difference generators* E *between intents* L*. If the directed multigraph* (L, E) *is weakly-connected, it is called a* description quiver*:*

$$\begin{aligned}(L, E) \text{ is a description quiver, where} \\ \bullet\ & L \subseteq \mathbb{L} : \emptyset'' \in L \\ \bullet\ & E = \{(B_1, B_2, D) \mid B_1 \cup D'' = B_2, \\ & \qquad B_1, B_2 \in L, B_1 \subset B_2, D \subseteq B_2 \setminus B_1\}\end{aligned} \tag{7}$$

An example of a description quiver for the context from Table 1 is given on Figure 2. Note that a quiver can contain have multiple edges between the same nodes as there can be multiple difference generators between two intents.

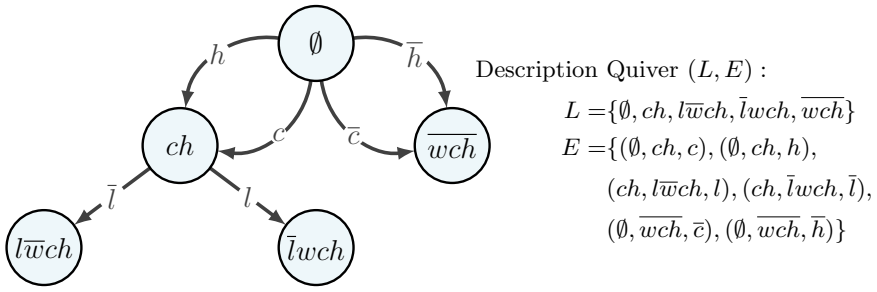

Fig. 2: Description Quiver example. Nodes represent a subset of intents of the contexts. And edges show how to "generate" these intents

3.2 Path in quiver

As for any graph, we can define traversal procedure for a decision quiver. This subsection introduces paths in quivers adapted to intents and difference generators (i.e., to decision quivers).

First, let us define a path in quiver (L, E).

Definition 3. *Given quiver (L, E), a sequence of k edges $\langle e_i \rangle_{i=1}^{k}, e_i \in E$ is called a **path in the quiver** if each element e_i of the path is a difference generator between intents B_i and B_{i-1}, and the first element e_1 is a difference generator between B_1 and $\emptyset''$:*

$$\langle e_i \rangle_{i=1}^{k} \subseteq E : \forall i = 1, \ldots, k : e_i = (B_{i-1}, B_i, D_i), B_0 = \emptyset'' \tag{8}$$

For example, consider the quiver in Figure 2. Its five intents and six edges are given on the right of the figure. An example of a path in this quiver would be the tuple of edges $\langle (\emptyset, ch, c), (ch, l\overline{w}ch, l) \rangle$.

Now we can define a set of paths P in quiver (L, E) as follows:

Definition 4. *Given quiver (L, E), **set of paths** P in the quiver is the set of all possible paths in the quiver:*

$$P = \{\langle e_i \rangle_{i=1}^{k} \subseteq E \mid \forall i = 1, \ldots, k : e_i = (B_{i-1}, B_i, D_i), B_0 = \emptyset'', k \in \mathbb{N}\}. \tag{9}$$

For the quiver in Figure 2, the set of paths $P = \{ \langle (\emptyset, ch, c), (ch, l\overline{w}ch, l) \rangle,$ $\langle (\emptyset, ch, h), (ch, l\overline{w}ch, l) \rangle, \langle (\emptyset, ch, c), (ch, \overline{l}wch, \overline{l}) \rangle, \langle (\emptyset, ch, h), (ch, \overline{l}wch, \overline{l}) \rangle, \langle (\emptyset, \overline{wch}, \overline{c}) \rangle,$ $\langle (\emptyset, \overline{wch}, \overline{h} \rangle$ }.

Definition 5. *Given quiver (L, E) and description $X \subseteq M$, the* ***set of paths P_X, following description X,*** *is the set of maximal paths in quiver (L, E) such that each edge description in a path is a subset of X:*

$$\begin{aligned} P_{X,any\ k} &= \{\langle e_i \rangle_{i=1}^{k} \in P \mid D_i \subseteq X, \forall i \in \mathbb{N} : i \leq k\}, \\ P_X &= \{\langle e_i \rangle_{i=1}^{k} \in P_{X,any\ k} \mid \nexists \langle e_i \rangle_{i=1}^{k+1} \in P_{X,any\ k}\}, \end{aligned} \tag{10}$$

where $e_i = (B_{i-1}, B_i, D_i)$.

For the quiver in Figure 2, the set of paths P_{lwc}, following description lwc would be $P_{lwc} = \{\langle(\emptyset, ch, c), (ch, l\overline{w}ch, l)\rangle, \langle(\emptyset, ch, h), (ch, l\overline{w}ch, l)\rangle, \langle(\emptyset, \overline{wch}, \overline{h})\rangle\}$.

We are specifically interested in the set of intents L_X we can arrive to via paths P_X, and the set of edges we pass E_X while traversing the quiver:

Definition 6. *Given quiver (L, E) and description $X \subseteq M$, the subset of the terminal intents L_X in paths, following X, is called the set of* ***targets of X.*** *The set of* ***maximal targets*** *of X is denoted by $L_{X,\max}$. The set of edges E_X from paths P_X is called the set of* ***arrows of X.***

$$\begin{aligned} L_X &= \{B_k \mid \langle e_i \rangle_{i=1}^{k} \in P_X\}, \\ L_{X,\max} &= \{B \in L_X \mid \nexists \tilde{B} \in L_X : B \subset \tilde{B}\} \\ E_X &= \{e_1, \ldots, e_k \mid \langle e_i \rangle_{i=1}^{k} \in P_X\} \end{aligned} \tag{11}$$

where $e_i = (B_{i-1}, B_i, D_i)$.

For the quiver in Figure 2, the targets L_{lwc} of description lwc are $L_{lwc} = \{l\overline{w}ch, \overline{wch}\}$. The targets are incomparable, therefore the set of maximal targets of description lwc is the same: $L_{lwc,\max} = L_{lwc}$. The arrows of description lwc is $E_{lwc} = \{(\emptyset, ch, c), (ch, l\overline{w}ch, l), (\emptyset, \overline{wchh})\}$.

Definition 7. *Description quiver (L, E) is called a* ***description tree*** *if for each non-top node of the quiver there is only one node leading to it:*

$$\begin{aligned} (L, E) \text{ is a tree } \iff & (L, E) \text{ is a quiver, and} \\ & \forall B_2 \in L \setminus \{\emptyset''\} : \left|\{B_1 \mid (B_1, B_2, D) \in E\}\right| = 1 \end{aligned} \tag{12}$$

Description tree (L, E) is called **dichotomic** when each node $B \in L$ with edges, going from this node, $\{(B_1, B_2, D) \in E \mid B_1 = B\} \neq \emptyset$ has exactly two such edges and their descriptions are dichotomic attributes $m, \overline{m} \in M^{\pm}$: $\{(B_1, B_2, D) \in E \mid B_1 = B\} = \{(B, B_3, \{m\}), (B, B_4, \{\overline{m}\})\}$, where $B_3, B_4 \in L$.

4 Decision Quivers for supervised setting

In this subsection we show the connections between description quivers and decision trees, a basic supervised machine learning model.

In supervised machine learning setup, we are given a context (G, M, I), the set of outcomes Y and the mapping $\tau : G \to Y$ from (training) objects outcomes. The set of outcomes is usually defined as $Y = \{0, 1\}$ for binary classification task, or as a set of real values $\mathbb{R}$ for regression task. Both tasks aims to finding a function $\varphi : 2^M \to Y$ that maps each description $X \subseteq M$ to a value from Y. Prediction function φ is chosen w.r.t. prediction quality measures that compare the outcomes of τ and φ on objects G and their descriptions.

In this subsection we study two types of decision quivers: target-based decision quiver $\dot{Q} = (L, E, \dot{\varphi} : L \to Y)$, and arrow-based decision quiver $\vec{Q} = (L, E, \vec{\varphi} : D \to Y)$. The small dots above the signs for $\dot{Q}$ and $\dot{\varphi}$ symbolize that target-based quivers assign predictions to the nodes L. Analogously, arrow-based quivers assign predictions to the edges E that is highlighted by arrow above the signs for $\vec{Q}$ and $\vec{\varphi}$.

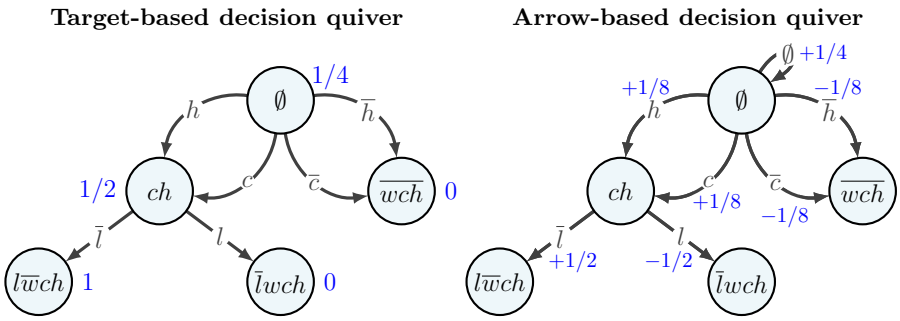

Fig. 3: Two types of Decision Quivers: target-based decision quiver assigns labels (or predictions) to the nodes, and arrow-based decision quiver assigns labels to the edges. Here, the numbers represent the probability of an object being "breast-feeding" (target Y) based on its attributes M.

4.1 Target-based decision quiver

Definition 8. ***Target-based decision quiver*** *is a triple* $(L, E, \dot{\varphi} : L \to Y)$, *where the pair* (L, E) *makes a description quiver and* $\dot{\varphi}$ *is a function that maps intents from* L *to values from* Y. *We will denote target-based decision quivers by* $\dot{Q}$*:*

$$\dot{Q} = (L, E, \dot{\varphi} : L \to Y) \tag{13}$$

To make a prediction $\dot{Q}(X)$ for description $X \subseteq M$ with target-based decision quiver $\dot{Q} = (L, E, \dot{\varphi})$, we average the individual predictions of maximal targets of description X:

$$\dot{Q}(X) = \frac{1}{|L_{X,\max}|} \sum_{B \in L_{X,\max}} \dot{\varphi}(B). \tag{14}$$

Target-based decision quivers make it easy to define a machine learning model in the FCA framework. First, we select a subset of intents L based on some interestingness measure. Second, we define a prediction function $\dot{\varphi}$ for each intent $B \in L$ (e.g. $\dot{\varphi}(B) = \frac{1}{|B'|}\sum_{g\in B'}\tau(g)$). Third, we select a subset of directed generators D to connect intents from L.

In fact, decision tree – a basic machine learning model – can be represented as a target-based decision quiver.

Proposition 2. ***Decision tree*** *is a target-based decision quiver* $(L, E, \dot{\varphi})$ *where* (L, E) *forms a description tree.*

4.2 Arrow-based decision quiver

Definition 9. ***Arrow-based decision quiver*** *is a triple* $(L, E, \vec{\varphi} : E \to Y)$*, where the pair* (L, E) *is a description quiver and* $\vec{\varphi}$ *is a function that maps directed generators from* D *to values from* Y*. We denote arrow-based decision quivers by* $\vec{Q}$*:*

$$\vec{Q} = (L, E, \vec{\varphi} : E \to Y) \tag{15}$$

To make a prediction $\vec{Q}(X)$ for description $X \subseteq M$ with arrow-based decision quiver $\vec{Q} = (L, E, \vec{\varphi})$, we sum the predictions $\vec{\varphi}$ of all arrows E_X following description X:

$$\vec{Q}(X) = \sum_{e\in E_X} \vec{\varphi}(e). \tag{16}$$

Often the models are required to make nonzero basic prediction. In target-based decision quivers such predictions are expressed by the prediction of the top intent $\dot{\varphi}(\emptyset'')$. In arrow-based decision quivers this can be represented with the trivial edge $(\emptyset'', \emptyset'', \emptyset)$: $\vec{\varphi}((\emptyset'', \emptyset'', \emptyset))$.

4.3 Summation of arrow-based decision quivers

Let us define a summation operation on arrow-based decision quivers:

$$\begin{gathered}\vec{Q}_1 + \vec{Q}_2 = \vec{Q}_\Sigma, \\ \text{where } \vec{Q}_1 = (L_1, E_1, \vec{\varphi}_1),\ \vec{Q}_2 = (L_2, E_2, \vec{\varphi}_2) \\ \vec{Q}_\Sigma = (L_1 \cup L_2, E_1 \cup E_2, \vec{\varphi}_1 + \vec{\varphi}_2)\end{gathered} \tag{17}$$

Arrow-based decision quivers can be multiplied by scalar:

$$(L, E, \vec{\varphi}) \cdot k = (L, E, k \cdot \vec{\varphi}), \quad \text{where } k \in \mathbb{R} \tag{18}$$

Therefore, arrow-based decision quivers allow us to "sum up" many quivers to one. Thus, they form a vector space.

Let us study how predictions of the sum of quivers relate to the predictions of the initial quivers. Consider an example of the averaging of two arrow-based

decision quivers on Figure 4. Figure represents two quivers $\vec{Q}_1 = (L_1, E_1, \vec{\varphi}_1)$, $\vec{Q}_2 = (L_2, E_2, \vec{\varphi}_2)$ and their average quiver $\vec{Q}_3 = (\vec{Q}_1 + \vec{Q}_2)/2 = (L_3, E_3, \vec{\varphi}_3)$. For object *bream* with description *wch* the quivers predictions will be $\vec{Q}_1(wch) = 0, \vec{Q}_2(wch) = 0, \vec{Q}_3(wch) = 0$.

Therefore, the average of two initial quivers predictions $(\vec{Q}_1(wch) + \vec{Q}_2(wch))/2$ equals the prediction of the averaged quiver $\vec{Q}_3(wch)$. This property does not hold for any description $X \subseteq M$. Consider object *sea snake* with description *lwc*, the predictions will be $\vec{Q}_1(lwc) = 1, \vec{Q}_2(lwc) = 0, \vec{Q}_3(lwc) = 3/4$.

In this case, the average of two initial predictions $(\vec{Q}_1(lwc) + \vec{Q}_2(lwc))/2$ is not equal to the prediction of the averaged quiver $\vec{Q}_3(lwc)$. However, the latter lies between the two initial predictions $\vec{Q}_2(lwc) < \vec{Q}_3(lwc) < \vec{Q}_1(lwc)$.

This problem of keeping predictions the same after the averaging requires an extensive study. For now, we can formulate the following proposition:

Proposition 3. *Given two arrow-based decision quivers $\vec{Q}_1 = (L_1, E_1, \vec{\varphi}_1), \vec{Q}_2 = (L_2, E_2, \vec{\varphi}_2)$ with only one common intent: $L_1 \cap L_2 = \{\emptyset''\}$, the prediction of the sum $\vec{Q}_\Sigma(X) = (\vec{Q}_1 + \vec{Q}_2)(X)$ is equal to the sum of predictions $\vec{Q}_1(X) + \vec{Q}_2(X)$ for any description $X \subseteq M$.*

Proof. Without loss of generality, assume that both initial quivers have the trivial edge $e_t = (\emptyset'', \emptyset'', \emptyset)$ with labels $\vec{\varphi}_1(e_t), \vec{\varphi}_2(e_t)$ that can be equal to zero.

Let the sum quiver be $\vec{Q}_\Sigma = (L_\Sigma, E_\Sigma, \vec{\varphi}_\Sigma)$. Since $\emptyset''$ is the only intent contained in both quivers, the only edge containing in both quivers would be the trivial edge $E_1 \cap E_2 = \{e_t\}$. Therefore, by the definition of the sum operation, any edge $e \in E_1 \setminus \{e_t\}$ of the first quiver would have the same label in the sum quiver: $\vec{\varphi}_1(e) = \vec{\varphi}_\Sigma(e)$. Analogously for the second quiver, $\forall e \in E_2 \setminus \{e_t\}, \vec{\varphi}_2(e) = \vec{\varphi}_\Sigma(e)$. And the label of the trivial edge e_t would be the sum of labels of the initial quivers: $\vec{\varphi}_\Sigma(e_t) = \vec{\varphi}_1(e_t) + \vec{\varphi}_2(e_t)$.

Predictions of the quivers are computed by equation 16. Now, putting everything together, we can write the sum of predictions as follows:

$$\begin{aligned}
\vec{Q}_1(X) + \vec{Q}_2(X) &= \sum_{e \in E_{1,X}} \vec{\varphi}_1(e) + \sum_{e \in E_{2,X}} \vec{\varphi}_2(e) \\
&= \Big(\sum_{e \in E_{1,X} \setminus \{e_t\}} \vec{\varphi}_1(e) + \sum_{e \in E_{2,X} \setminus \{e_t\}} \vec{\varphi}_2(e) \Big) + \Big(\vec{\varphi}_1(e_t) + \vec{\varphi}_2(e_t) \Big) \\
&= \sum_{e \in E_\Sigma \setminus \{e_t\}} \vec{\varphi}_\Sigma(e) + \vec{\varphi}_\Sigma(e_t) = \vec{Q}_\Sigma(X), \ \forall X \subseteq M
\end{aligned}$$

The requirement of two quiver $(L_1, E_1, \vec{\varphi}_1), (L_2, E_2, \vec{\varphi}_2)$ having only one common intent $L_1 \cap L_2 = \{\emptyset''\}$ might seem too restrictive. However, this often happens in practice for big data machine learning problems as the big datasets contain massive amount of possible intents $\mathbb{L}$. Much greater than the number of intents in both quiver: $|L_1 \cup L_2| \ll |\mathbb{L}|$.

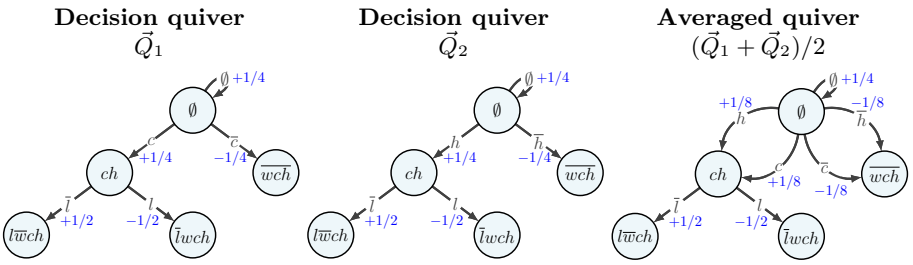

Fig. 4: Example of averaging arrow-based decision quivers.

4.4 Conversion of target-based quivers to arrow-based quivers

The previous subsections introduced target-based and arrow-based decision quivers. They showed that one can construct target-based decision quivers and one can sum arrow-based decision quivers. This section introduces "differentiation" operator $\delta : \dot{Q} \mapsto \vec{Q}$ that allows converting target-based decision quivers to arrow-based ones.

The desirable property of differentiation δ operator is that it should not affect the training objects predictions of the quivers:

$$\text{Given } \dot{Q}, \qquad \delta : \dot{Q} \mapsto \vec{Q}, \text{ s.t. } \dot{Q}(g') = \delta(\dot{Q})(g') = \vec{Q}(g'), \qquad \forall g \in G \tag{19}$$

It should be noted that, given an arbitrary target-based decision quiver $\dot{Q} = (L, E, \dot{\varphi})$, we cannot define a differentiation δ operator that would not affect the quiver predictions for any given description $X \subseteq M$. This is due to the fact that the training data can contain implications that would not hold true for the test data. And difference generators of the quiver can be reflect such implications.

Example 1. Consider the example of two decision quivers from Figure 3. Specifically, their intents $\emptyset$ and ch, and difference generators $(\emptyset, ch, c)$ and $\emptyset, ch, h)$. Both quivers start with prediction $\dot{\varphi}(\emptyset) = \vec{\varphi}((\emptyset, \emptyset, \emptyset) = 1/4$. Then, the target-based quiver predicts $1/2$ for intent ch, that is $1/4$ higher than the start prediction: $\dot{\varphi}(ch) = 1/2 = \dot{\varphi}(\emptyset) + 1/4$. The arrow-based decision quiver reflects this difference with the labels for two difference generators: $\vec{\varphi}((\emptyset, ch, c)) + \vec{\varphi}((\emptyset, ch, h)) = 1/8{+}1/8 = 1/4$. Now, the description of test object *sea snake* contains attribute c but no attribute h. Therefore, while making a prediction, the target-based quiver will follow the edge $(\emptyset, ch, c)$ and change its prediction by $1/4$: from $\dot{\varphi}(\emptyset) = 1/4$ to $\dot{\varphi}(ch) = 1/2$. And the arrow-based quiver will only change its prediction by $\vec{\varphi}((\emptyset, ch, c)) = 1/8$. Thus, two quivers would result in different predictions.

It is impossible to define what implication will be falsified on the test data. However, decision trees avoid these problem because of their dichotomic and tree properties.

Proposition 4. *Given a target-based decision* tree $\dot{Q} = (L, E, \dot{\varphi})$*, its differentiated version* $\delta(\dot{Q})$ *gives the same predictions for any input* $X \subseteq M$ *if differentiation operator* δ *defined as follows:*

$$\begin{aligned} &\text{For } \dot{Q} = (L, E, \dot{\varphi}),\ \dot{Q}(X) = \delta(\dot{Q})(X), \forall X \subseteq M \text{ if:} \\ &\quad (L, E) \text{ is a description tree, and } \delta(\dot{Q}) = (L, E, \vec{\varphi}), \text{ where} \\ &\quad \vec{\varphi}((B_1, B_2, D)) = \begin{cases} \dot{\varphi}(\emptyset), & \text{if } B_1 = B_2 = D = \emptyset, \\ \dot{\varphi}(B_2) - \dot{\varphi}(B_1), & \text{otherwise} \end{cases} \end{aligned} \tag{20}$$

Proof. The proof, although in different notation, is given in [8].

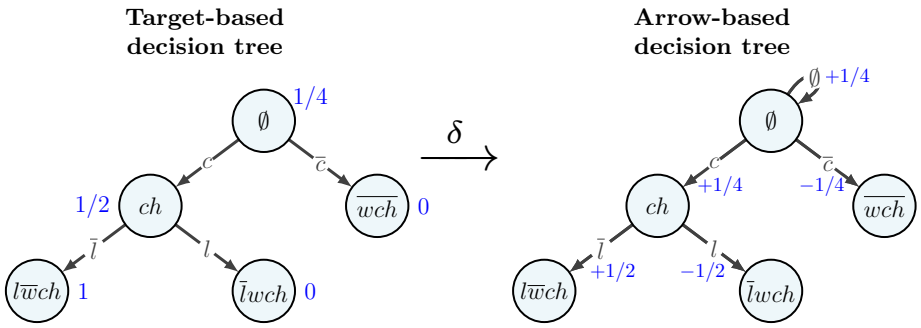

Fig. 5: Differentiation example. The arrow-based decision quiver on the right can be obtained by "differentiating" the target-based decision tree on the left.

With the introduced differentiation and summation operators we can merge ensembles of decision trees into a single decision quiver by 1) differentiating all decision trees of an ensemble, and 2) summing up these differentiated decision trees.

5 Experiments

5.1 Datasets

For this study we selected 14 real-world binary datasets from LUCS-KDD repository[3].

For all provided datasets we computed concept lattices and then selected subsets of the most stable concepts of various sizes. We use only a half of the

[3] Coenen, F. (2003), The LUCS-KDD Discretised/normalised ARM and CARM Data Library, `http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS_KDD_DN/`, Department of Computer Science, The University of Liverpool, UK.

id	context	# rows $\lvert G\rvert$	# columns $\lvert M\rvert$	# connections $\lvert I\rvert$	density $\frac{\lvert I\rvert}{\lvert G\times M\rvert}$
1	zoo	101	43	1717	0.40
2	iris	150	20	750	0.25
3	wine	178	69	2492	0.20
4	glass	214	49	2140	0.20
5	heart	303	53	4236	0.26
6	ecoli	336	35	2688	0.23
7	dematology	366	50	4750	0.26
8	breast	699	21	6974	0.48
9	pima	768	39	6912	0.23
10	anneal	898	74	12847	0.19
11	ticTacToe	958	30	9580	0.33
12	flare	1389	40	15279	0.28
13	led7	3200	25	25600	0.32
14	pageBlocks	5473	47	60203	0.23

Table 2: The description of contexts used in the experiments

datasets from the repository as the omitted ones result in too many concepts, thus making computation of the whole lattice infeasible. Table 2 provides information about the contexts used in the experiments.

5.2 Sizes of difference generators

In practice we often want to visualize the lattice of intents while presenting only the necessary attributes. In this subsection we measure to what extent difference keys allow us to shorten the labels of diagrams.

For each dataset from Table 2 we compute the whole concept lattice and estimate the stability lower bound for each concept. Then we select 10, 30, 100, and most stable concepts and compute the intent labels for the line diagram. We consider two types of labels: 1) labels drawn for each intent (i.e. attributes of an intent, that belong to no smaller intent), and 2) labels drawn between two connected intents (i.e. attributes of an intent, that do not belong to each of the preceding intents). Finally, for each label we calculate the cardinality of key of this label divided by the cardinality of the label itself.

The results are shown in Figure 6. We see that the ratio between the lengths of the difference key and the difference often lies between 80 and 100 percent. That is, the keys of difference are not always shorter than the original differences. However, for some datasets and small number of selected most stable concepts, the ratio reaches 60 percent. Therefore, replacing intent differences by the keys of these differences can reduce the total size of attribute subsets given as labels of diagram edges up to 40 percent.

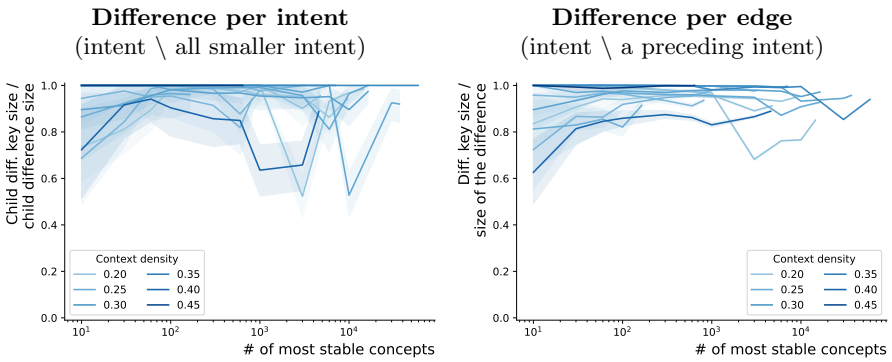

Fig. 6: Ratios between the lengths of the keys of differences and the lengths of the differences themselves. To the left: each difference consists of intent attributes that do not belong to smaller intents. To the right: each difference consists of intent attributes that do not belong to one of its preceding intents.

5.3 Summation of decision trees

Decision Quivers allow us to sum many decision trees in one model with no changes in the overall predictions. Here we test the summation of quivers empirically.

We test experimentally the correctness of summation operation by comparing the predictions of a random forest model and the arrow-based decision quiver constructed from this random forest. Constructing the arrow-based decision quiver from a random forest consists of three steps: 1) differentiation of all decision trees of the random forest, 2) summation of differentiated decision trees into one arrow-based decision quiver; and 3) division of the resulting decision quiver by the number of trees. That is, when random forest averages predictions of its decision trees, we "average" the decision trees themselves.

To make experiments closer to practice, we run them on the numerical (non-binarized) versions of datasets of Table 2 from UCI repository. We only take classification dataset from the table, therefore the set of outcomes Y for each dataset represents the list of probabilities of an object belonging to a specific class. For the sake of efficiency, we only consider the contexts with less than 1000 objects. For each classification dataset from the table, we make 50 random splits of the dataset to disjoint "train" and "test" subsets, where the test subset contains 20% of rows. Then, for each train-test split we fit a random forest of 100 decision trees on the train subset of the dataset. We proceed by constructing a set of arrow-based decision quivers from the first $k = 1, 2, 3, \ldots, 100$ decision trees of the forest. Finally, we use the test subset of the dataset to compute the mean average difference between the averaged predictions of the first k decision trees and the predictions of the arrow-based decision quiver constructed by averaging these k decision trees.

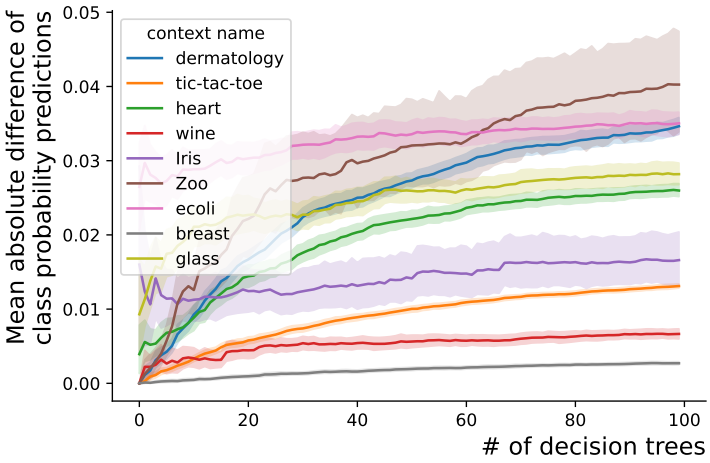

Fig. 7: The mean absolute difference between predictions of a random forest and an arrow-based decision quiver obtained by summing decision trees of the forest.

Figure 7 presents the obtained results. It can be seen that the difference between the probabilities is not null and grows with the number of decision trees. However, the difference remains small and does not exceed 5 percent. On average for our 12 datasets, the class probability predicted by decision quiver differs from the class probability predicted by the corresponding random forest by no more than 5%.

The difference in predictions appears because of two factors. First, as it is shown in Proposition 3, the sum of two quivers can give different predictions if these quiver have more than one common intent. Rarely, different decision trees of a random forest find the same intents while fitting. Thus, such decision trees will not satisfy the conditions of Proposition 3.

The second factor is the use of floating-point numbers. We run the experiments on numerical datasets, therefore each binary attribute used by a decision tree is constructed by comparing a real value x_i of i-th element in description x with real-valued threshold θ. We cannot perfectly fit to the precision of floating-point threshold used by sci-kit learn package. Thus, rarely, when the difference between x_i and θ is extremely small (e.g. 10^{-8}), our implementation can consider $x_i \leq \theta$ as True, and sci-kit learn implementation would say the opposite. Overall, this discrepancy is of engineering nature and can be considered as a measurement error.

6 Conclusion

We have introduced and studied description quivers as compact representation of concept lattices and respective ensembles of decision trees. We have studied

some properties of description quivers and tried to justify their use for describing state-of-the-art symbolic machine learning models based on decision trees.

We have shown that the proposed description quiver allows one to fuse decision trees, while proposing a way to select the most important rules in decision models. Computer experiments that we performed justified our expectations about the usefulness of decision quivers.

Acknowledgments

The work on Sections 1 and 2 was done by Sergei O. Kuznetsov under support of the Russian Science Foundation under grant 22-11-00323 and performed at HSE University, Moscow, Russia.

Egor Dudyrev and Amedeo Napoli are carrying out this research work as part of the French ANR-21-CE23-0023 SmartFCA Research Project.

References

1. Aldinucci, T., Civitelli, E., Di Gangi, L., Sestini, A.: Contextual Decision Trees. arXiv preprint arXiv:2207.06355 (2022)
2. Bělohlávek, R., De Baets, B., Outrata, J., Vychodil, V.: Characterizing trees in concept lattices. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **16**, 1–15 (2008)
3. Bělohlávek, R., De Baets, B., Outrata, J., Vychodil, V.: Inducing decision trees via concept lattices. International journal of general systems **38**(4), 455–467 (2009)
4. Breiman, L.: Random forests. Machine Learning **45**, 5–32 (2001)
5. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth (1984)
6. Buzmakov, A., Dudyrev, E., Kuznetsov, S.O., Makhalova, T., Napoli, A.: Experimental Study of Concise Representations of Concepts and Dependencies. In: Cordero, P., Krídlo, O. (eds.) Proceedings of the Sixteenth International Conference on Concept Lattices and Their Applications (CLA 2022). pp. 117–132. CEUR Workshop Proceedings 3308, CEUR-WS.org (2022)
7. Dudyrev, E., Kuznetsov, S.O.: Decision concept lattice vs. decision trees and random forests. In: Proceedings of the 16th International Conference on Formal Concept Analysis (ICFCA 2021). pp. 252–260. Springer (2021)
8. Dudyrev, E., Kuznetsov, S.O.: Summation of Decision Trees. In: Kuznetsov, S.O., Napoli, A., Rudolph, S. (eds.) Proceedings of the 9th International Workshop FCA4AI co-located with IJCAI 2021. pp. 99–104. CEUR Workshop Proceedings 2972, CEUR-WS.org (2021)
9. Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Berlin (1999)
10. Hanika, T., Hirth, J.: Conceptual Views on Tree Ensemble Classifiers. arXiv preprint arXiv:2302.05270 (2023)
11. Kuznetsov, S.O.: Machine Learning and Formal Concept Analysis. In: Proceedings of the Second International Conference on Formal Concept Analysis (ICFCA). pp. 287–312. Springer (2004)
12. Strecht, P.: A Survey of Merging Decision Trees Data Mining Approaches. In: Proceedings of the 10th Doctoral Symposium in Informatics Engineering (DSIE'15). pp. 36–47 (2015)