# Constructing decision quivers

Egor Dudyrev[1,2,*], Sergei O. Kuznetsov[1] and Amedeo Napoli[2]

[1]HSE University, 20 Myasnitskaya St, Moscow, 101000, Russian Federation
[2]Université de Lorraine, CNRS, LORIA, F-54000 Nancy, France

Abstract
Rule Learning and Formal Concept Analysis (FCA) are two fields of science that study similar topic yet speak in a very different terms. This paper describes rule-based machine learning models with FCA-based terminology which results in decision quiver model. A decision quiver, discussed in the paper, is a supervised machine learning model that is based on intents, generators of intents, and predictions for each intent (or generator). We show that the finding of the optimal set of intents is a cornerstone task in constructing a decision quiver (and thus, any rule-based model). The paper finishes with the baseline algorithm to construct decision quivers. The algorithm produces machine learning models that are much smaller than the state-of-the-art ensembles of decision trees, yet that offer the similar quality of predictions.

**Keywords**
Supervised Machine Learning, Explainable Artificial Intelligence, Formal Concept Analysis

## 1. Introduction

Rule Learning [1] and Formal Concept Analysis (FCA) [2] are two fields of science that study similar topic yet speak in a very different terms. Rule Learning searches for rules that could accurately predict the attributes of unseen data. While FCA focuses on studying dependencies in only the given data. This paper attempts to combine the language of FCA and the goal of Rule Learning in one model called Decision Quivers. Thus, we combine the mathematicity of FCA with the applicability of Rule learning.

Rule learning is a mature and well-recognised research area mainly concerned with finding the Boolean rules (i.e. ”rules”) from the given attributes that are able to predict the value of “target” attribute. One specific configuration of a rule-based model – an ensemble of decision trees – is considered among the state-of-the-art machine learning models on tabular data. However, the last big increment in the prediction quality of the ensembles was introduced in the year 2016 [3] and is based on ensembling decision trees that are known since (at least) the year 1986 [4]. The later studies on Rule learning focus on interpreting and explaining big rule-based models, ensembles of decision trees especially [5] [6] [7] [8]. We attribute the lack of novel

*Published in Sergei O. Kuznetsov, Amedeo Napoli, Sebastian Rudolph (Eds.): The 11th International Workshop ”What can FCA do for Artificial Intelligence?”, FCA4AI 2023, co-located with IJCAI 2023, August 20 2023, Macao, S.A.R. China, Proceedings, pp. 69–80.*

*Corresponding author.
†These authors contributed equally.

egor.dudyrev@loria.fr (E. Dudyrev); skuznetsov@hse.ru (S. O. Kuznetsov); amedeo.napoli@loria.fr (A. Napoli)
0000-0002-2144-3308 (E. Dudyrev); 0000-0003-3284-9001 (S. O. Kuznetsov); 0000-0001-5236-9561 (A. Napoli)

state-of-the-art models in the area to the lack of a good formalism to describe rule-based models. Such formalism could also propose new ways to interpret and explain big rule-based models.

Formal Concept Analysis is a formalism aimed at analysing data based on discrete (often binary) descriptions. The FCA focus on binary descriptions promises its good applicability to become a formalism for rule-based models.

The natural connections between FCA and rule-based machine learning were covered in many works: [9] [10] [11] [12] [13] [14]. This paper does not attempt to highlight any new connection between FCA and rule-based models. Instead, we discuss a model for rule-based machine learning called Decision Quiver. The model is defined using the very basic notions of FCA – closed descriptions and generators. The notion of generators allows decision quiver describe any other rule-based model. While the notion of closed descriptions greatly shrinks the search space while constructing the model.

The paper is structured as follows. Section 2 recalls the basics of Formal Concept Analysis and Supervised Machine Learning, and defines Decision Quivers. Section 3 introduces the pipeline and the simple algorithm to construct decision quivers. Section 4 evaluates the algorithm on some of LUCS-KDD datasets. Section 5 concludes the paper.

## 2. Background

This subsection introduces definitions we use throughout the paper. Firstly, we provide the basic terms of Formal Concept Analysis to describe the rule models. Secondly, we describe the space of premises that contains the machine learning models discussed in the paper. Thirdly, we describe the main topics of a binary classification in the language in the FCA notation.

### 2.1. Formal Concept Analysis

In Formal Concept Analysis the data is represented as a formal context $K = (G, M, I)$, which is a triple of a set of objects $G$, a set of attributes $M$ and the relation $I \subseteq G \times M$ among them.

A running example of a formal context is provided in Table 1 (together with "Target $\mathbb{Y}$" column and "test objects" rows that will be introduced in the following sections). To lighten the notation, we will represent a subset of attributes from the running example as a concatenation of these attributes: e.g. we denote the subset of attributes $\{c, h\}$ as $ch$.

We call any subset of attributes $D \subseteq M$ a description. And we denote the set of all descriptions (i.e. the powerset of $M$) as $2^M$:

$$2^M = \{D \mid D \subseteq M\} \tag{1}$$

Now we can define two "prime" operations: $A'$ would describe all attributes $M$ shared by objects $A \subseteq G$, and $B'$ would describe all objects $G$ covered by attributes $B \subseteq M$. Prime operations:

$$\begin{aligned} A' &= \{m \in M \mid \forall g \in A : (g, m) \in I\}, \quad A \subseteq G \\ B' &= \{g \in G \mid \forall m \in B : (g, m) \in I\}, \quad B \subseteq M \end{aligned} \tag{2}$$

Two prime operations, combined together, result in "double prime" operator $(\cdot)''$ that has the properties of a closure operator. For example, if $B$ is a subset of attributes $M$ then $B''$ is the closure of $B$ on set $M$.

| | | *Attributes $M$* | | | | *Target $\mathbb{Y}$* |
|---|---|---|---|---|---|---|
| | | l | w | c | h | $\tau$ |
| | dog | x | | x | x | 100% |
| *Objects* | corn | x | | | | 0% |
| *G* | bream | | x | x | x | 0% |
| | egg | | | | | 0% |
| *Test* | reed | x | x | | | ? (0%) |
| *objects* | frog | x | x | x | x | ? (0%) |
| *Attr. names* | | lives on land | lives in water | can move | has limbs | is mammal |

**Table 1**
Running example of a formal context with additional target column $\mathbb{Y}$ and test objects

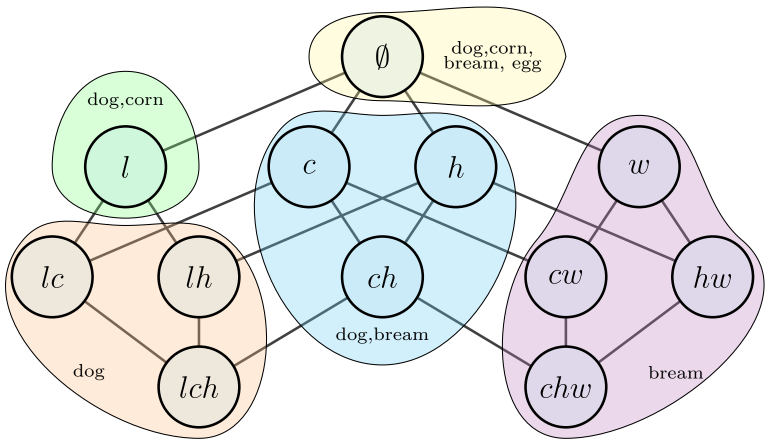

**Figure 1:** Descriptions of the example context grouped by the subsets of objects they describe. Descriptions of the empty set of objects are omitted.

In FCA terminology, a closed description $B \subseteq M, B'' = B$ is also referred to as an *intent*. The set of all intent is denoted by $\mathbb{B}$ and, together with set inclusion order, forms a lattice :

$$\mathbb{B} = \{B \subseteq M \mid B'' = B\} \tag{3}$$

Two different subsets of attributes $D, E \subseteq M, D \neq E$ can describe the same set of objects $D' = E'$ and, consequently, have the same closure $D'' = E''$. Such descriptions are called *equivalent*. Equivalence class $[D]$ of description $D \subseteq M$ is denoted as $[D]$:

$$[D] = \{E \subseteq M \mid E'' = D''\}. \tag{4}$$

Line diagram on Figure 1 shows the descriptions of a context from the running from Table 1 grouped by the equivalence classes. Descriptions equivalent to $M$ (that describe no objects) are omitted as they would make the diagram harder to read.

In each equivalence class $[D], D \subseteq M$, there is a single maximal description $D'' = B$ (also called closed description or intent) and possibly many minimal descriptions, that are called "keys". Let $B \subseteq M$ be a closed description $B'' = B$, then $keys(B)$ is the set of minimal descriptions, equivalent to $B$:

$$keys(B) = \{E \in [B] \mid \nexists D \in [B] \text{ s.t. } D \subset E\}, \quad B \subseteq M \tag{5}$$

## 2.2. Supervised Machine Learning

This subsection covers the basic ideas of Supervised Machine Learning and introduces our notation for these terms.

Let us define a formal context $(G, M, I)$, a space of target values $\mathbb{Y}$, and a target label $\tau(g) \in \mathbb{Y}$ for each object $g \in G$. The task of supervised machine learning is to find a function $\psi$ that maps any description $X \subseteq M$ to a target value $\psi(X) \in \mathbb{Y}$ so that, for any object $g \in G$, the prediction $\psi(g')$ would be close to the target label $\tau(g)$. The "closeness" of target labels $\tau$ and predictions $\psi$ on objects $G$ is evaluated by non-negative loss function $\mathscr{L}(\tau, \psi \mid G) \in \mathbb{R}_+$, where $\mathscr{L}(\tau, \psi \mid G) = 0$ means that $\psi$ is optimal. Here we provide two loss functions that can be used to evaluate the prediction function $\psi$: negative F1-score $\mathscr{L}_{F1}$ for binary classification task (i.e. when $\mathbb{Y} = \{0, 1\}$), and Mean Squared Error (MSE) $\mathscr{L}_{MSE}$ for regression task (i.e. when $\mathbb{Y} = \mathbb{R}$).

$$\mathscr{L}_{F1}(\tau, \psi \mid G) = 1 - 2\frac{\sum_{g \in G} \tau(g)\psi(g')}{\sum_{g \in G}\big(1 - \tau(g)\psi(g')\big)}$$

$$\mathscr{L}_{MSE}(\tau, \psi \mid G) = \frac{1}{|G|}\sum_{g \in G}\big(\tau(g) - \psi(g')\big)^2$$

It should be noted that, since the loss function $\mathscr{L}$ is evaluated on objects $G$, the "best" prediction function $\psi$ for these objects would be the function $\psi : g' \mapsto \tau(g)$ that predicts the label $\tau(g)$ of every objects $g \in G$ based on its description $g'$. Such function $\psi$ would give zero loss on objects $G$ but it will not extrapolate well on new descriptions, not shared by objects $G$. This issue is often mitigated by introducing the "test" formal context $(G_{\text{test}}, M, I_{\text{test}})$ with target labels $\tau_{\text{test}} : G_{\text{test}} \to \mathbb{Y}$. Then, prediction function $\psi$ is constructed by minimizing the loss on "train" context $(G, M, I)$, but the final evaluation of the loss of function $\psi$ is done on the test context $(G_{\text{test}}, M, I_{\text{test}})$. In the running example from Table 1, objects $G_{\text{test}}$ are denoted as *Test objects*.

In what follows we use the "average" operation over the target space (Equation 6). We assume, therefore, that the space $\mathbb{Y}$ is "averageble", i.e. for every list of tuple of target values $Y$, its average $\text{avg}(Y)$ is also a target value from $\mathbb{Y}$. This assumption does not make the following reasoning too specific, as many Supervised Machine Learning problems can be reformulated to satisfy it. For example, binary classification task suggests only two target values $\mathbb{Y}_{\text{bin}} = \{0, 1\}$. However, it can be reformulated as a regression task with $\mathbb{Y} = [0, 1]$ where the values of $\mathbb{Y}$ represent the probability of an object to belong to the positive class $1 \in \mathbb{Y}_{\text{bin}}$.

$$\text{avg}(Y) = \frac{1}{|Y|}\sum_{y \in Y} y, \quad Y \in \mathbb{Y}^1 \cup \mathbb{Y}^2 \cup \ldots \cup \mathbb{Y}^\infty \tag{6}$$

## 2.3. Rule set

Given a description $X \subseteq M$, a rule-based machine learning model makes prediction $\psi(X)$ based on a set of rules of the form: "if $X$ is described by $P \subseteq M$ then predict $\varrho(P) \in \mathbb{Y}$". That is, every rule is characterised by a pair $(P, \varrho(P))$ where $P \subseteq M$ is a subset of attributes called *premise*, and $\varrho(P) \in \mathbb{Y}$ is a *prediction* of premise $P$. A *rule set* is a pair $(\mathscr{P}, \varrho)$ where $\mathscr{P} \subseteq 2^M$ is a set of premises and $\varrho$ is a function that maps each premise to a target value ($\varrho : \mathscr{P} \mapsto \mathbb{Y}$).

Now, let us consider two special cases when predicting with rule set $(\mathscr{P}, \varrho)$. Let $X \subseteq M$ be a description and let $P_1, P_2 \in \mathscr{P}$ be two comparable premises covered by $X$: $P_1 \subset P_2 \subseteq X$. Then we follow the intuition that a more precise premise $P_2$ should give a more precise prediction $\varrho(P_2)$, therefore we only use premise $P_2$ to make a prediction about the target of $X$. Now, let $P_1, P_2 \in \mathscr{P}$ be incomparable premises, covered by $X$: $P_1 \not\subseteq P_2, P_2 \not\subseteq P_1, P_1 \subseteq X, P_2 \subseteq X$. Then we use both premises $P_1, P_2$ to make a prediction for $X$ by averaging their predictions $\psi(X) = \text{avg}\big((\varrho(P_1), \varrho(P_2))\big)$.

With these ideas in mind we define prediction function $\psi_{(\mathscr{P},\varrho)} : 2^M \to \mathbb{Y}$ for rule set $(\mathscr{P}, \varrho)$ as follows:

$$\psi_{(\mathscr{P},\varrho)}(X) = \text{avg}\big((\varrho(P) \mid P \in \mathscr{P}_{X,\max})\big), \quad X \subseteq M \tag{7}$$

$$\text{where } \mathscr{P}_{X,\max} = \{P \in \mathscr{P} \mid P \subseteq X, \forall P_2 \in \mathscr{P} : (P \subset P_2) \implies (P_2 \not\subseteq X)\} \tag{8}$$

Prediction $\varrho(P)$ for a premise $P \in \mathscr{P}$ is often computed as the average of target labels of objects described by $P$:

$$\varrho(P) = \text{avg}\big((\tau(g) \mid g \in P')\big) \tag{9}$$

**Example 1.** *Let us provide an example of a rule set $(\mathscr{P}, \varrho)$ constructed on the example context 1. For the sake of readability, we represent the rule set as a set of implications $\{P \implies \varrho(P) \mid P \in \mathscr{P}\}$:*

$$\{\varnothing \implies 25\%,\ l \implies 50\%,\ lc \implies 100\%,\ lh \implies 100\%,\ w \implies 0\%\}. \tag{10}$$

*Let us make a prediction for object* reed *with description $X = lw$. There are three premises from $\mathscr{P}$ that can be applied for X: $\{\varnothing, l, w\}$. Out of these three, there are two maximal descriptions: $\mathscr{P}_{X,\max} = \{l, w\}$. The former predicts that* reed *is mammal with $\varrho(l) = 50\%$ probability, and the latter predicts that* reed *is mammal with $\varrho(w) = 0\%$ probability. Therefore, the final prediction is $\psi_{(\mathscr{P},\varrho)}(X) = 25\%$ probability of* reed *being a mammal.*

*Analogously, let us predict whether object* frog *with description $X = lwch$ is a mammal. All premises of $\mathscr{P}$ can be applied for the given X. However, only three of them are maximal: $\mathscr{P}_{X,\max} = \{lc, lh, w\}$. The corresponding premise predictions are $\varrho(lc) = 100\%, \varrho(lh) = 100\%, \varrho(w) = 0\%$. Therefore, the final prediction is $\psi_{(\mathscr{P},\varrho)}(X) = 67\%$ probability of* frog *being a mammal.*

*This process of making predictions is schematically depicted in Figure 2.*

$$X = lw \xrightarrow{\mathcal{P}_{X,\max}} \{l, w\} \xrightarrow{\varrho} (50\%, 0\%) \xrightarrow{\text{avg.}} 25\% = \psi_{(\mathcal{P},\varrho)}(X)$$

$$X = lwch \xrightarrow{\mathcal{P}_{X,\max}} \{lc, lh, w\} \xrightarrow{\varrho} (100\%, 100\%, 0\%) \xrightarrow{\text{avg.}} 67\% = \psi_{(\mathcal{P},\varrho)}(X)$$

**Figure 2:** The pipeline of making predictions with Rule Set of Example 1 for a reed ($X = lw$) (top subfigure) and a frog ($X = lwch$) (bottom subfigure)

## 2.4. Implicitly equivalent premises

Let us rewrite the predictions from Example 1 in more details, considering what objects from $G$ we use to make predictions.

For object *reed* with description $X = lw$ the set $\mathscr{P}$ contains two maximal premises describing $X$: $\mathscr{P}_{X,\max} = \{l, w\}$. Premise $l$ describes training objects $l' = \{dog, corn\} \subset G$ with corresponding target labels $\tau(dog) = 100\%, \tau(corn) = 0\%$ whose average label is 50%. Premise $w$ describes training object $w' = \{bream\} \subset G$ with target label $\tau(bream) = 0\%$. So the rule set $(\mathscr{P}, \varrho)$ from Example 1 predicts that *reed* is mammal with 25% probability.

For object *frog* with description $X = lwch$ the set $\mathscr{P}$ contains three maximal premises describing $X$: $\mathscr{P}_{X,\max} = \{lc, lh, w\}$. Premise $lc$ describes training object $l' = \{dog\} \subset G$ with target label $\tau(dog) = 100\%$; premise $lh$ describes the same training object training object $w' = l' = \{dog\} \subset G$ with target label $\tau(dog) = 100\%$; and premise $w$ describes training object $w' = \{bream\} \subset G$ with target label $\tau(bream) = 0\%$. So the rule set $(\mathscr{P}, \varrho)$ from Example 1 predicts that *frog* is mammal with 67% probability.

This process of making predictions is schematically depicted in Figure 3.

$$\begin{tikzcd}
 & & \{dog, corn\} \arrow[r, "\tau"] & (100\%, 0\%) \arrow[r, "\text{avg.}"] & 50\% \arrow[dr, "\text{avg.}"] & \\
X = lw \arrow[r, "\mathcal{P}_{X,\max}"] & \{l, w\} \arrow[ur, "l'"] \arrow[dr, "w'"'] & & & & 25\% = \psi_{(\mathcal{P},\varrho)}(X) \\
 & & \{bream\} \arrow[r, "\tau"] & (0\%) \arrow[r, "\text{avg.}"] & 0\% \arrow[ur, "\text{avg.}"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
 & & \{dog\} \arrow[r, "\tau"] & (100\%) \arrow[r, "\text{avg.}"] & 100\% \arrow[dr, "\text{avg.}"] & \\
X = lwch \arrow[r, "\mathcal{P}_{X,\max}"] & \{lc, lh, w\} \arrow[ur, "lc'"] \arrow[r, "lh'"] \arrow[dr, "w'"'] & \{dog\} \arrow[r, "\tau"] & (100\%) \arrow[r, "\text{avg.}"] & 100\% \arrow[r, "\text{avg.}"] & 67\% = \psi_{(\mathcal{P},\varrho)}(X) \\
 & & \{bream\} \arrow[r, "\tau"] & (0\%) \arrow[r, "\text{avg.}"] & 0\% \arrow[ur, "\text{avg.}"] &
\end{tikzcd}$$

**Figure 3:** The explicit pipeline of making predictions with Rule Set of Example 1 for a reed ($X = lw$) (top subfigure) and a frog ($X = lwch$) (bottom subfigure)

Notice that two premises $\{lc, lh\} \subset \mathscr{P}$ used for target prediction of object *frog* ($X = lwch$) have the same extent: $lc' = lh' = \{dog\}$. Therefore, the information that subset of objects $\{dog\}$ has average target of 100% is used two times inside the rule set. In fact , if we count only one premise with extent $\{dog\}$ dog when making the final prediction $\psi_{(\mathscr{P},\varrho)}$ we will get the value 50%, which is closer to the true value 0% than 67%.

There are many possible ways to overcome this issue. For example, one can construct rule set $(\mathscr{P}, \varrho)$ such that the set $\mathscr{P}$ would only contain premises that are either comparable or contradicting (i.e. $\forall P_1, P_2 \in \mathscr{P} \ : \ P_1 \subseteq P_2$ or $P_2 \subseteq P_1$ or $\exists m \in P_1$, s.t. "not $m$" $\in P_2$). This is the approach, used by decision trees. One can also construct rule set $(\mathscr{P}, \varrho)$ such that the all premises in $\mathscr{P}$ are closed (i.e. $\mathscr{P} \subseteq \mathbb{B} \subseteq 2^M$). Then, there will be no two premises that describe the same set of objects. This is the approach commonly used in FCA literature.

In this paper we propose another approach by enriching the rule set $(\mathscr{P}, \varrho)$ with the set of closures $\mathscr{B}$ of premises $\mathscr{P}$. $\mathscr{B} = \{P'' \mid P \in \mathscr{P}\}$.

### 2.5. Decision Quiver

The previous sections introduced equivalent descriptions and covered their importance for making predictions with rule sets. This section presents Decision Quiver: a rule set model enriched with the information about equivalent descriptions.

The model of Decision Quiver was introduced in [15] as a directed multigraph (i.e. a quiver) with intents as nodes, generators as edges, and predictions for each nodes. This paper inherits the name of Decision Quiver but presents its definition in a more set-theoretic way, as we have found the latter to be more concise.

**Definition 1.** Let $(G, M, I)$ be a (training) context and $\mathbb{B}$ be its set of closed descriptions, a *decision quiver* is a triplet $Q = (\mathscr{P}, \mathscr{B}, \varrho)$ of premises $\mathscr{P} \subseteq 2^M$, their intents $\mathscr{B} \subset \mathbb{B} : \{P'' \mid P \in \mathscr{P}\} = \mathscr{B}$, and predictions $\varrho : \mathscr{B} \to \mathbb{Y}$ for every intent.

Given a description $X \subseteq M$, prediction $\psi_{(\mathscr{P},\mathscr{B},\varrho)}(X)$ is computed as the average of predictions $\varrho$ of maximal intents $\mathscr{B}_{X,\max} \subseteq \mathscr{B}$, whose generators $\mathscr{P}_X \subseteq \mathscr{P}$are covered by $X$.

$$\psi_{(\mathscr{P},\mathscr{B},\varrho)}(X) = \operatorname{avg}\big((\varrho(B) \mid B \in \mathscr{B}_{X,\max})\big), \tag{11}$$

$$\text{where } \mathscr{B}_{X,\max} = \{B \in \mathscr{B} \mid \exists P \in \mathscr{P}_{X,\max} \text{ s.t. } B = P'',\ \nexists P_2 \in \mathscr{P}_{X,\max} \text{ s.t. } B \subset P_2''\} \tag{12}$$

$$\mathscr{P}_{X,\max} = \{P \in \mathscr{P} \mid P \subseteq X, \forall P_2 \in \mathscr{P} \text{ s.t. } (P \subseteq P_2) \implies (P_2 \not\subseteq X)\} \tag{13}$$

**Example 2.** *Let us provide an example of a decision quiver $(\mathscr{P}, \mathscr{B}, \varrho)$ having the same premises as a rule set $(\mathscr{P}, \varrho)$ from Example 1. For the sake of readability, we represent the quiver $(\mathscr{P}, \mathscr{B}, \varrho)$ as two sets of implications: $\{P \implies B \mid P \in \mathscr{P}, B \in \mathscr{B}, P'' = B\}$ and $\{B \implies \varrho(B) \mid B \in \mathscr{B}\}$:*

$$\mathscr{P} \to \mathscr{B} : \{\emptyset \implies \emptyset,\ l \implies l,\ lc \implies lch,\ lh \implies lch,\ w \implies chw\},$$
$$\mathscr{B} \to \mathbb{Y} : \{\emptyset \implies 25\%,\ l \implies 50\%,\ lch \implies 100\%,\ chw \implies 0\%\} = \varrho.$$

*Contrary to the rule set $(\mathscr{P}, \varrho)$ from Example 1, the quiver $(\mathscr{P}, \mathscr{B}, \varrho)$ "knows" that premises lc and lh are equivalent as they correspond to the same closure lch. Therefore, when making prediction for a frog with description $X = lwch$, the quiver uses only two subsets of objects: $\{dog\} = lwch'$ with average target equal to 100%, and $\{bream\} = chw'$ with average target equal to 0%. This prediction process is schematically represented on Figure 4.*

$$\begin{tikzcd}
 & & lch \arrow[r, "\varrho"] & 100\% \arrow[dr, "\text{avg}"] & \\
X = lwch \arrow[r, "\mathcal{P}_{X,\max}"] & \{lc, lh, w\} \arrow[ur, "lc''"] \arrow[ur, "lh''"'] \arrow[dr, "w''"'] & & & 50\% = \psi_{(\mathcal{P},\mathcal{B},\varrho)}(X) \\
 & & chw \arrow[r, "\varrho"] & 0\% \arrow[ur, "\text{avg}"] &
\end{tikzcd}$$

**Figure 4:** The pipeline of making predictions with Decision Quiver based on a Rule Set of Example 1 for a frog ($X = lwch$)

## 3. Quivers construction pipeline

The previous section introduced target-based and arrow-based decision quivers. Now, let us discuss, how these quivers can be constructed.

## 3.1. Algorithm to find the optimal quiver

Decision quiver $Q = (\mathscr{P}, \mathscr{B}, \varrho)$ consists of three elements. Note that prediction function $\varrho : \mathscr{B} \to \mathbb{Y}$ depends on intents $\mathscr{B}$ and not the premises $\mathscr{P}$. And considering the training formal context, every premise $P \in \mathscr{P}$describes the same objects as itc closure $B \in \mathscr{B}, P'' = B$. So, when making predictions on the training context, the choice of a specific subset of premises $\mathscr{P}$is irrelevant. Therefore, the task of finding the optimal quiver $(\mathscr{P}, \mathscr{B}, \varrho)$ on the training formal context reduces to finding the optimal rule set $(\mathscr{B}, \varrho)$ whose premises are limited to intents.

After finding $(\mathscr{B}, \varrho)$ one should select the optimal set of premises $\mathscr{P}$to construct a quiver $(\mathscr{P}, \mathscr{B}, \varrho)$ that would be generalisable to the test data. However, since premises with the same intent describe the same objects, they are empirically indistinguishable. Thus, the choice of the premises $\mathscr{P}$relies on a priori intentions: for example, one may want to make premises as precise as possible (then $\mathscr{P} = \mathscr{B}$), or as general as possible (then $\mathscr{P} = \bigcup_{B \in \mathscr{B}} keys(B)$).

More formally, let $\mathscr{L}$ be a loss function, $(G, M, I)$ be a training context, and $Q_{\text{opt}} = (\mathscr{P}_{\text{opt}}, \mathscr{B}_{\text{opt}}, \varrho_{\text{opt}})$ be the quiver, that achieves the minimal loss $\mathscr{L}$ on the context $(G, M, I)$: $\mathscr{L}(\tau, \varrho_{Q_{\text{opt}}} \mid G) \to 0$. Then, the task of finding such optimal quiver $Q_{\text{opt}}$ can be separated into three independent steps:

0. Fix the search space of intents $\mathbb{B}_{\text{search}} \subseteq \mathbb{B}$ of context $(G, M, I)$,
1. Find the optimal rule set $(\mathscr{B}_{\text{opt}}, \varrho_{\text{opt}})$ of intents $\mathscr{B}_{\text{opt}} \subseteq \mathbb{B}_{\text{search}}$ and their predictions $\varrho_{\text{opt}} : \mathscr{B}_{\text{opt}} \to \mathbb{Y}$,
2. Construct the set of premises $\mathscr{P}_{\text{opt}}$ of intents $\mathscr{B}_{\text{opt}}$.

The initial step of the pipeline is fixing the search space $\mathbb{B}_{\text{search}}$. For the sake of simplicity, in this paper we assign the search space $\mathbb{B}_{\text{search}}$ to be the space of all intents $\mathbb{B}$. The ways to minimise the search space is one of the future research directions.

The first and the main step of the algorithm is finding the optimal rule set $(\mathscr{B}_{\text{opt}}, \varrho_{\text{opt}})$ where the choice of premises is limited to intents from the search space: $\mathscr{B}_{\text{opt}} \subseteq \mathbb{B}_{\text{search}}$. Remind that the prediction function $\varrho_{\text{opt}}$ is often evaluated the same way for every intent. For example, similar to Equation 9, prediction $\varrho_{\text{opt}}(B)$ for an intent $B \in \mathbb{B}$ is the average target label of objects $B'$ described by intent $B$. Thus, the search for optimal rule set $(\mathscr{B}_{\text{opt}}, \varrho_{\text{opt}})$ is a search for optimal subset of intents $\mathscr{B}_{\text{opt}} \subseteq \mathbb{B}_{\text{search}}$:

$$\mathscr{B}_{\text{opt}} = \arg \min_{\mathscr{B} \subseteq \mathbb{B}_{\text{search}}} \mathscr{L}(\tau, \psi_{(\mathscr{B}, \varrho)} \mid G) \tag{14}$$

The last step of the pipeline is to construct the set of premises $\mathscr{P}_{\text{opt}}$ of quiver $Q_{\text{opt}}$ to generalise the latter to the possible test descriptions. Here we propose two options for the set of generators $\mathscr{P}_{\text{opt}}$: the set of closed descriptions $\mathscr{P}_{\text{closed}} = \mathscr{B}_{\text{opt}}$, and the set of keys $\mathscr{P}_{\text{keys}} = \bigcup_{B \in \mathscr{B}_{\text{opt}}} \text{keys}(B)$.

## 3.2. Algorithm to find the optimal subset of intents

This paper uses the very basic greedy discrete optimisation algorithm to find the optimal subset of intents (i.e. solving the task from Equation 14).

When finding the optimal subset of intents $\mathscr{B}_{\text{opt}} \subseteq \mathbb{B}_{\text{search}}$ we operate the fixed set training context $(G, M, I)$, the fixed targets $\tau : G \to \mathbb{Y}$, and the fixed way to compute the prediction $\varrho(B)$

for an intent $B \in \mathbb{B}_{\text{search}}$ (see eq. 9). Therefore, for the sake of brevity, we define the training loss $\mathscr{L}_{\text{train}}(\mathscr{B})$ of a subset of intents $\mathscr{B}$ as follows:

$$\mathscr{L}_{\text{train}}(\mathscr{B}) = \mathscr{L}(\tau, \psi_{(\mathscr{B},\varrho)} \mid G). \tag{15}$$

The algorithm (to find $\mathscr{B}_{\text{opt}}$ given $\mathbb{B}_{\text{search}}$):

0. Start with $\mathscr{B}$ containing the top intent $\mathscr{B} = \{\emptyset''\}$,
1. Find the intent $B_{\text{opt}} \in \mathbb{B}_{\text{search}}$ that gives the minimal loss, when added to the current set of intents $\mathscr{B}$:
   $B_{\text{opt}} = \arg\min_{B \in \mathbb{B}_{\text{search}}} \mathscr{L}_{\text{train}}(\mathscr{B} \cup \{B\})$
2. Add intent $B_{\text{opt}}$ to the set $\mathscr{B}$,
3. Repeat the steps 1, 2 while the set $\mathscr{B}$ contains less than $\mathscr{B}_{\max} \in \mathbb{N}$ elements and the loss decrease is higher than $\epsilon \in \mathbb{R}_+$:
   repeat until $|\mathscr{B}| \leq \mathscr{B}_{\max}, \quad \text{and } \mathscr{L}_{\text{train}}(\mathscr{B}) - \mathscr{L}_{\text{train}}(\mathscr{B} \cup \{B_{\text{opt}}\}) > \epsilon.$

## 4. Experiments

The main limitation for the current algorithm is that considers the intents search space $\mathbb{B}_{\text{search}}$ as the set of all intents $\mathbb{B}$. Therefore, we can only apply the algorithm on the datasets where we can compute the intents $\mathbb{B}$. For the experiments we chose nine dataset from LUCS-KDD repository [16] that are discretized versions of real-world datasets from UCI repository. The set of intents $\mathbb{B}$ for every dataset was computed by LCM algorithm implemented in Scikit-mine repository.

The selected datasets give the task of multi-class classification. For example, Iris dataset asks to classify a flower into one of the three types (Setosa, Versicolour, and Virginica) based on its petal and sepal lengths and widths. We used F1 score with weighted averaging as a loss function so that it can be applied to all datasets with no adjustments.

For each dataset we fit a gradient boosting model provided by Sci-Kit learn [17] and XGBooost [3] to get the state-of-the-art prediction quality scores. Then we construct two decision quivers for each dataset: one makes predictions via intents (denoted by ”Quiver, intents”), and the other makes predictions via keys of these intents (denoted as ”Quiver, keys”).

Table 2 compares the test prediction quality of models on the selected datasets. One can see that the Gradient boosting models gave the best scores. However, in some cases, quiver models showed comparable decision quality: e.g. Iris, Congres, Breast, Flare datasets. Also, on this data, we see almost no differences between quivers that make predict with intents and quivers that prediction with keys. The only dataset where the difference occurs is Zoo dataset. Currently, we cannot explain this fact as we assumed the difference would be much more apparent. So we should proceed with more profound studies comparing intents and keys as means for predictions.

Now, let us compare the sizes of the models, provided in Table 3. One can see that quiver models contain no more that 6 intents for all the datasets. While each gradient boosting consists of one hundred of decision trees. Note that each decision tree in each gradient boosting consist of many Boolean rules. In that regard, decision quiver model are more efficient, as they are able to achieve the similar prediction quality score with much lesser number of rules.

**Table 2**
Test F1 score

| Dataset | Quiver, intents | Quiver, keys | Grad. Boosting sklearn | XGBoost |
|---|---|---|---|---|
| zoo | 0.819 | 0.869 | 1.000 | 0.929 |
| iris | 0.967 | 0.967 | 0.967 | 0.967 |
| ecoli | 0.703 | 0.703 | 0.757 | 0.808 |
| congres | 0.909 | 0.909 | 0.909 | 0.920 |
| breast | 0.936 | 0.936 | 0.936 | 0.936 |
| ticTacToe | 0.745 | 0.745 | 0.984 | 0.990 |
| flare | 0.878 | 0.878 | 0.901 | 0.880 |
| led7 | 0.460 | 0.460 | 0.774 | 0.769 |
| nursery | 0.863 | 0.863 | 0.988 | 1.000 |

**Table 3**
Model complexity in size and in construction time

| Dataset | Model size | | | Construction time (seconds) | | |
|---|---|---|---|---|---|---|
| | # intents in quiver | # trees, GB sklearn | # trees, XGBoost | Quiver | GB, sklearn | XGBoost |
| zoo | 6 | 100 | 100 | 12.333 | 0.159 | 0.057 |
| iris | 3 | 100 | 100 | 2.250 | 0.067 | 0.047 |
| ecoli | 3 | 100 | 100 | 161.508 | 0.223 | 0.125 |
| congres | 2 | 100 | 100 | 158.786 | 0.039 | 0.037 |
| breast | 3 | 100 | 100 | 0.786 | 0.028 | 0.032 |
| ticTacToe | 5 | 100 | 100 | 53.155 | 0.060 | 0.065 |
| flare | 1 | 100 | 100 | 661.792 | 0.204 | 0.196 |
| led7 | 5 | 100 | 100 | 2.047 | 0.961 | 0.726 |
| nursery | 5 | 100 | 100 | 1165.161 | 2.774 | 0.835 |

However, the better algorithm to construct decision quivers should be developed. As the current algorithm construct the quivers for too long. For example, on Nursery dataset, it took XGBoost 0.8 seconds to construct 100 decision trees, and it took decision quivers algorithm 19 minutes to select 5 intents.

## 5. Conclusion

In this paper we have presented decision quivers as a formalism for describing rule-based machine learning models. We showed that description quiver can describe any rule-based model. And, by incorporating closed descriptions, it can possibly suggest efficient algorithms to create more optimal machine learning models.

We have also presented a general pipeline to construct decision quivers. We showed that the most important part of the pipeline is finding the optimal set of intents. Thus, many FCA-based algorithm can be used to construct decision quivers. The current baseline algorithm for quivers construction is very slow, compared to the state-of-the-art models. However, it can produce much smaller models with the similar prediction quality.

## References

[1] J. Fürnkranz, D. Gamberger, N. Lavrač, J. Fürnkranz, D. Gamberger, N. Lavrač, Rule learning in a nutshell, Foundations of Rule Learning (2012) 19–55.

[2] B. Ganter, R. Wille, Formal Concept Analysis, Springer, Berlin, 1999.

[3] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[4] J. R. Quinlan, Induction of decision trees, Machine learning 1 (1986) 81–106.

[5] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence 1 (2019) 206–215. doi:10.1038/s42256-019-0048-x.

[6] C. Bénard, G. Biau, S. Da Veiga, E. Scornet, Interpretable random forests via rule extraction, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 937–945.

[7] H. Lakkaraju, S. H. Bach, J. Leskovec, Interpretable decision sets: A joint framework for description and prediction, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1675–1684.

[8] E. Dudyrev, I. Semenkov, S. O. Kuznetsov, G. Gusev, A. Sharp, O. S. Pianykh, Human knowledge models: Learning applied knowledge from the data, Plos one 17 (2022) e0275814.

[9] S. O. Kuznetsov, Machine learning and formal concept analysis, in: Concept Lattices: Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004. Proceedings 2, Springer, 2004, pp. 287–312.

[10] B. Ganter, S. O. Kuznetsov, Hypotheses and version spaces, in: ICCS, volume 2746, Springer, 2003, pp. 83–95.

[11] E. Dudyrev, S. O. Kuznetsov, Decision concept lattice vs. decision trees and random forests, in: Formal Concept Analysis: 16th International Conference, ICFCA 2021, Strasbourg, France, June 29–July 2, 2021, Proceedings 16, Springer, 2021, pp. 252–260.

[12] R. Belohlavek, B. De Baets, J. Outrata, V. Vychodil, Inducing decision trees via concept lattices, International journal of general systems 38 (2009) 455–467.

[13] T. Hanika, J. Hirth, Conceptual views on tree ensemble classifiers, International Journal of Approximate Reasoning (2023) 108930.

[14] Š. Horvát, L. Antoni, O. Krídlo, A. Szabari, S. Krajči, Generalized decision directed acyclic graphs and their connection with formal concept analysis (2022).

[15] E. Dudyrev, S. O. Kuznetsov, A. Napoli, Description quivers for compact representation of concept lattices and ensembles of decision trees, in: D. Dürrschnabel, D. López Rodríguez (Eds.), Formal Concept Analysis, Springer Nature Switzerland, Cham, 2023, pp. 127–142.

[16] F. Coenen, The lucs-kdd discretised/normalised arm and carm data library, 2003. URL: http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS_KDD_DN/.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, the Journal of machine Learning research 12 (2011) 2825–2830.