Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Learning decision catalogues for situated decision making: The case of scoring systems

Stefan Heid [a,1], Jonas Hanselle [a,b,1], Johannes Fürnkranz [c], Eyke Hüllermeier [a,b,*]

[a] *Institute of Informatics, LMU Munich, Germany*
[b] *Munich Center for Machine Learning, Germany*
[c] *Johannes-Kepler-University Linz, Austria*

ARTICLE INFO

Keywords:
Machine learning
Algorithmic decision-making
Scoring systems

ABSTRACT

In this paper, we formalize the problem of learning coherent collections of decision models, which we call decision catalogues, and illustrate it for the case where models are scoring systems. This problem is motivated by the recent rise of algorithmic decision-making and the idea to improve human decision-making through machine learning, in conjunction with the observation that decision models should be situated in terms of their complexity and resource requirements: Instead of constructing a single decision model and using this model in all cases, different models might be appropriate depending on the decision context. Decision catalogues are supposed to support a seamless transition from very simple, resource-efficient to more sophisticated but also more demanding models. We present a general algorithmic framework for inducing such catalogues from training data, which tackles the learning task as a problem of searching the space of candidate catalogues systematically and, to this end, makes use of heuristic search methods. We also present a concrete instantiation of this framework as well as empirical studies for performance evaluation, which, in a nutshell, show that greedy search is an efficient and hard-to-beat strategy for the construction of catalogues of scoring systems.

1. Introduction

With the increasing access to technology, computational resources, and massive amounts of data, the idea of taking advantage of machine learning (ML) methodology to optimize decision support is becoming more and more feasible. Automated or partially automated decision-making with models constructed in a data-driven way is indeed appealing for various reasons, especially as it is potentially more rational, objective, and accurate than decision-making by humans alone, which may be subjective or error-prone. For example, think of decisions in the context of employees recruitment, such as hiring or placement decisions [43], or the data-driven construction of individualized treatment rules in personalized medicine [66]. Further examples can be found in other domains, including jurisdiction [34] and disaster management [65].

However, highly accurate models generated by modern ML algorithms, such as deep neural networks, tend to be complex and difficult to comprehend, and may not be appropriate in every situation. In fact, a certain degree of transparency of a model and explainability of decisions is often desirable. Besides, depending on the situation and application context, time and computational

[*] Corresponding author at: Institute of Informatics, LMU Munich, Germany.
E-mail addresses: stefan.heid@ifi.lmu.de (S. Heid), jonas.hanselle@ifi.lmu.de (J. Hanselle), juffi@faw.jku.at (J. Fürnkranz), eyke@lmu.de (E. Hüllermeier).

[1] Authors contributed equally.

https://doi.org/10.1016/j.ijar.2024.109190
Received 28 February 2023; Received in revised form 4 April 2024; Accepted 4 April 2024

Available online 12 April 2024

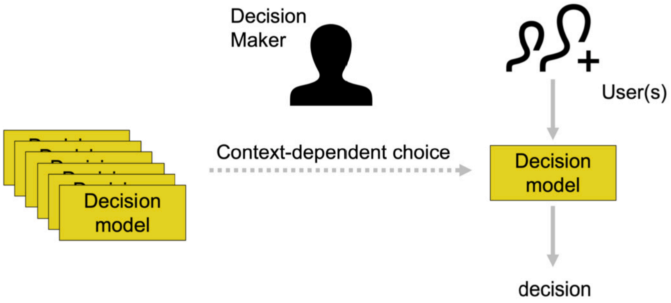

Fig. 1. Illustration of the basic setting studied in this paper: In a given decision situation, a decision maker can choose a model from a catalogue of (candidate) models (left). Once a suitable model has been selected, it is used to make a decision for the current user(s).

resources for applying decision models might be limited. For example, a human's resources to collect, validate, and enter data might be scarce, or decisions must be taken quickly. In the extreme case, instead of running an algorithm on a computer, the decision is taken by a human expert herself — imagine, for example, a medical doctor who needs to take decisions in emergency cases. In such situations, the decision maker may only be able to rely on simple, easy-to-evaluate knowledge and rules of thumb. Whatever the case, the complexity of a decision model should be well adapted to the experience of the decision maker, the information at hand, the computational resources available, and any possible time constraints that have to be met. For example, recent studies have confirmed that human decision makers rely on weighing multiple features characterizing a decision situation when given enough time, whereas decisions are based on very few or even a single feature if time pressure is high [36].

Motivated by the huge potential to improve decision-making through machine learning, together with the problems in exploiting this potential due to the difficulties just mentioned, we promote the idea of a data-driven, ML-based process for constructing catalogues of decision models. Their intention is to support a seamless transition from very simple to more sophisticated models. This transition and the most appropriate level of complexity can be controlled by the human decision maker (DM). For example, the DM might be a medical doctor diagnosing a patient. To this end, she will be supported by a set of models that recommend a decision (e.g., diagnosis, training programme) on the basis of characteristics of the patient (and perhaps further information about the case at hand). Based on the resources and information available to the DM, she can select an appropriate model from the set, and possible refine it later on, in case new resources or information becomes available. An illustration of this scenario is shown in Fig. 1.

Our aim is to develop a machine learning methodology suitable for a scenario of that kind, that is, a methodology for constructing a family of decision models — or *decision catalogue* for short — with increasing complexity. Each of the models is particularly suitable for a specific situation, and constructed in a learning process with the help of data, possibly supported by expert knowledge. Essentially, the machine learning system creates solutions within the space of candidate models that are "satisficing" [47,48], i.e., which meet a certain aspiration level while being as simple as possible. In other words, the proposed models are not necessarily (Bayes-)optimal in the sense of decision performance, but achieve an optimal compromise between a desired decision performance on the one side and complexity on the other.

Following a brief overview of related work in the next section, we introduce and formalize the problem of learning decision catalogues in Section 3, where we also propose a general (search-based) framework for tackling this problem. The latter is then instantiated for the specific case of scoring systems in Section 4. Experimental results are presented in Section 5, prior to concluding the paper with an outlook on extensions and future work in Section 6.

2. Related work

Work on the combination of decision modelling and machine learning can be found mainly in the fields of multi-criteria decision aid (MCDA) and preference learning. MCDA aims at helping a decision maker to rank or sort choice alternatives on the basis of their values on multiple criteria [35,33]. To this end, MCDA has developed a wide variety of decision models, most of which aggregate the evaluations on individual criteria into an overall assessment of an alternative. An important example of a model of that kind is the (discrete) Choquet integral [8], that is able to capture the interaction between pairs or within groups of variables (in this context also referred to as *criteria*), and combines nonlinearity, interpretability, and monotonicity elegantly. In fact, while being a nonlinear aggregation function, the Choquet integral offers measures (such as the Shapley value) for quantifying the importance of individual predictor variables as well as their interaction. Although our focus is on scoring systems in this paper, models of that kind can be seen as interesting alternatives.

The specification of models in MCDA is typically accomplished during the course of an interactive process, in which a decision analyst seeks to elicit the decision maker's preferences by asking informative questions [42]. In contrast to such a human-centric *construction* of preference models, preference learning (PL) is geared toward the automated data-driven *induction* of models [19]. This idea has received increasing attention in the recent past, and has been exemplified in different ways. Examples include approaches to learning the majority rule model [51], the non-compensatory sorting model [52], the TOPSIS model [1], and the OWA operator [37]. As already said, especially prevalent in MCDA are approaches based on the Choquet integral, and its qualitative counterpart, the Sugeno integral. Various methods for identifying such models (or, more precisely, the non-additive measure on which the integral is defined) have been proposed in the literature [23,24]. The problem is essentially considered as a parameter identification problem and commonly formalized as a constraint optimization problem—for example, using the sum of squared errors as an objective function [55,22].

Another important and closely related research direction is the learning of simple decision heuristics that are considered plausible from the perspective of cognitive psychology. However, this is a relatively unexplored field, in which only a few publications can be found so far [49]. Related to this are methods for learning decision models that are often used in practical applications. In this paper, we will focus on a specific example of that kind, namely so-called scoring systems. Roughly speaking, a scoring system proceeds from a set of (binary) features characterizing a decision context. The presence of a feature contributes a specific score, and a positive decision is made if the cumulative score exceeds a threshold (cf. Section 4). Models of that kind are especially comprehensible and used in many applications and fields of applied research [20]. Often, standard machine learning methods, such as support vector machines or logistic regression, are used to train a (sparse) linear model, and the real-valued coefficients of that model are then turned into integers, e.g., through rounding or by taking the sign. Obviously, approaches of that kind are rather ad-hoc, and indeed, can be shown to yield suboptimal performance in practice [54]. From a theoretical perspective, certain guarantees for the rounded solutions can nevertheless be given [7]. Moreover, more principled approaches have been developed in the recent past. In a series of papers, Ustun and Rudin developed the so-called Supersparse Linear Integer Model (SLIM) for inducing scoring systems from data, as well as an extension called RiskSLIM [59–61]. Their methods are based on formalizing the learning task as an integer linear programming problem, with the objective to find a meaningful compromise between sparsity (number of variables included) and predictive accuracy. The problem can then essentially be tackled by means of standard ILP solvers. Another method with provable guarantees, which includes a binarization of real-valued features, has been developed by Sokolovska et al. [53].

All the above methods are intended to construct a single model, wherefore they may serve as building blocks within the methodology we intend to develop. Yet, they do not support an on-demand adjustment of model complexity, which is at the core of our work—that is, the construction of an entire family of models covering a wide spectrum from simple to complex, and allowing one to adapt the model to the situation at hand and the needs of the user. This area has received little attention in the machine learning literature. A notable exception is cascaded generalization [21], which learns a cascade of models, each aiming to correct the mistakes of the previous one. Several related techniques, such as arbitrating [41], grading [46] or patching [32] share the same abstract goal. Particularly interesting is delegating [21], which formulates and analyses a general threshold-based framework where decisions are passed on to the next decision element in case the current cannot decide with a certain minimum confidence. Somewhat similar in spirit are the cascaded classifiers by Clertant et al. [9], which seek to find an optimal compromise between the quality and cost of prediction: given a query instance, additional feature values are determined (causing certain costs) and dynamically added until a sufficiently accurate prediction can be made. Decision lists [45], which are widely used in inductive rule learning, may also be seen as an instantiation of this approach: whenever the current rule is applicable to the example at hand, it makes a prediction, otherwise it passes the decision on to the next rule in the list. One can therefore also view decision lists as a complexity-ordered catalogue of models, where each model is one rule longer than the previous one. An algorithm explicitly targetted towards learning a sequence of rule-based models pruned to different degrees, and thus with increasing complexity, was also investigated by Fürnkranz [16,17]. However, in these and other cases, the goal was not to support a human decision maker with a catalogue of models of different complexity, but to rely on incremental model refinement and the power of multiple models, similar to ensemble methods such as boosting. Close in philosophy to our goal is early work on the interactive concept learning system CLINT, which was able to dynamically shift its learner along a series of pre-defined language biases with different degrees of expressivity [13].

Related are also techniques that aim at learning models with minimal feature acquisition costs [e.g. 57,12]. They share with our work the goal of obtaining models that are cheap to evaluate, but also focus on single models, whereas we explicitly aim at a catalogue of models, each being more complex or more expensive but also more accurate than the previous one.

3. Learning decision catalogues

Consider a scenario where decisions need to be made in different contexts, which are characterized in terms of a set of features $\mathcal{F} = \{f_1, \dots, f_K\}$. A concrete situation is specified by a vector $\boldsymbol{x} = (x_1, \dots, x_K)$, where x_i is the value observed for the feature f_i—using machine learning jargon, the set of all conceivable vectors of that kind forms the instance space $\mathcal{X}$. Features can be of various kinds, i.e., binary, (ordered) categorical, or numeric. Decisions are taken from a decision space $\mathcal{Y}$, which we suppose to be finite (typically comprising a small to moderate number of alternatives to choose from) and, for simplicity, independent of the context (i.e., the decision maker has always the same choice options, regardless of the context).

A decision model is a mapping $h : \mathcal{X} \longrightarrow \mathcal{Y}$, i.e., $y = h(\boldsymbol{x})$ is the decision suggested by h in the context $\boldsymbol{x}$. The task of a learning algorithm is to induce such a model from a set of training data

$$\mathcal{D} = \left\{(\boldsymbol{x}_i, y_i)\right\}_{i=1}^{N}, \tag{1}$$

i.e., to select a model from a predefined model class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ in light of data $\mathcal{D}$ collected from decisions taken in the past. In machine learning, a model h is called a hypothesis and $\mathcal{H}$ the hypothesis space. Note that different learning methods represent models in different ways, for example in the form of (generalized) linear models, decision trees, neural networks, etc. As already mentioned, we are mostly interested in simple, genuinely interpretable model classes.

3.1. Structured model classes

Our idea of complexity control requires the possibility to increase or decrease the complexity of a decision model h when the need arises, optimally adapting it to the situation at hand. For the decision maker, a complete change in the type of model (e.g., replacing a decision tree with a neural network) is certainly undesirable. Instead, it would be preferable to remain within the same model class. To support seamless complexity control, this space should be equipped with a mathematical structure, e.g., a nested sequence $\mathcal{H}_0 \subset \mathcal{H}_1 \subset \ldots \subset \mathcal{H}_M$ of subspaces sorted by complexity: $\mathcal{H}_i$ is (in a sense to be specified) less complex than $\mathcal{H}_{i+1}$ $(i = 0, 1, \ldots)$. Structures of that kind are known in statistical learning theory, where they have been explored for the purpose of structural risk minimization [62].

The type of models considered in this regard are mostly function spaces, for example polynomials of different degrees, and the subspaces are sorted according to their capacity. Such models do not appear to be appropriate for our purposes, however, mainly due to a lack of interpretability and transparency. More relevant are symbolic models and models expressed in terms of logical formalisms, or even simple decision heuristics, such as decision lists, lexicographic models [15], tallying [11], and so forth. As already said, our focus in this paper will be on so-called scoring systems.

For model classes of that kind, it is less obvious how a suitable structure should be defined. An obvious idea is to restrict the space $\mathcal{H}$ by means of constraints that can be relaxed step by step. For example, constraints might be imposed on the number of variables included in a model. In this way, a smooth transition from very simple to more sophisticated decision rules can be realized. It is not immediately clear, however, in which way such constraints influence the expressiveness and capacity of the model class, let alone the "cognitive complexity", that is, the complexity as perceived by the decision maker.

Our goal, therefore, is to identify a structured space of models that meets two key criteria. First, it should be possible to vary the capacity of the subspaces smoothly, that is, to increase or decrease the capacity in sufficiently small steps. We assume that a higher capacity allows for learning models with better predictive performance—at least in principle. In practice, this is of course not completely true, because increasing the capacity will also increase the difficulty of learning and the danger of generalizing poorly due to overfitting the training data. Second, the differences in capacity should align well with the differences in complexity, where the latter is actually meant to summarize (at least) three different types of complexity:

- *Computational complexity*: Which computational resources are needed to apply the model in practice, that is, to compute a prediction in a specific situation? Note that we are mainly interested in the complexity of *applying* the decision model, and less in the complexity of *training* it. In fact, if training can be done "offline", it is normally less critical from this point of view. That said, (re-)training time could become an issue in the case of an *interactive* process, when a model or a catalogue is not constructed in a purely data-driven way but with a "human in the loop" [29].
- *Cognitive complexity*: How comprehensible is the model, how transparent, and how difficult is it to explain the model to the decision maker? Research in explainable AI often equates this with the syntactic complexity of a model. While this is certainly an important aspect, the concept appears to be considerably more complex [18,38].
- *Cost*: Different models may also have different costs, for example, in a monetary sense or in terms of time and effort [58]. In particular, cost values can often be attached to features. In a medical context, for example, measuring the temperature of a patient is much cheaper than conducting a blood test.

With regard to the search for a most suitable ("best satisficing") model, it appears reasonable to go beyond simple linear structures and equip $\mathcal{H}$ with a more general ordering, such as a lattice structure. This is because complexity can normally be increased in different ways. Formally, we seek a structure $(\mathbb{H}, \sqsubseteq)$, where $\mathbb{H} \subseteq 2^{\mathcal{H}}$ and $\sqsubseteq$ is a partial order relation on $\mathbb{H}$, such that $\mathcal{H}' \sqsubseteq \mathcal{H}''$ if the model class $\mathcal{H}''$ is in a sense more expressive (or at least as expressive as) model class $\mathcal{H}'$ (cf. Fig. 2). An important special case is the subset relation, i.e., $\mathcal{H}' \sqsubseteq \mathcal{H}''$ iff $\mathcal{H}' \subseteq \mathcal{H}''$, meaning that all models $h \in \mathcal{H}'$ are also available in $\mathcal{H}''$. For example, a model class could be restricted by the subset of features that the learner is allowed to use.

3.2. Learning decision catalogues

Suppose $\mathbb{H}$ to be finite, i.e., $\mathbb{H}$ is a set $\{\mathcal{H}_1, \ldots, \mathcal{H}_M\}$ of model classes, and let $\mathcal{H} = \bigcup_{i=1,\ldots,M} \mathcal{H}_i$. $\mathbb{H}$ is equipped with a partial order relation $\sqsubseteq$, the strict part of which we denote by $\sqsubset$ (i.e., $\mathcal{H}_i \sqsubset \mathcal{H}_j$ if $\mathcal{H}_i \sqsubseteq \mathcal{H}_j$ and $\mathcal{H}_j \not\sqsubseteq \mathcal{H}_i$). Moreover, we assume that each class $\mathcal{H}_i$ can be associated with a (numerical) complexity degree $c(\mathcal{H}_i) \in \mathbb{R}$, where the complexity function $c : \mathbb{H} \longrightarrow \mathbb{R}$ is coherent with $\sqsubseteq$ in the sense that $\mathcal{H}_i \sqsubseteq \mathcal{H}_j$ iff $c(\mathcal{H}_i) \leq c(\mathcal{H}_j)$. Individual decision models $h \in \mathcal{H}$ inherit the complexity of model classes by letting $c(h) = \min\{c(\mathcal{H}_i) \mid h \in \mathcal{H}_i\}$.

As already said, instead of creating only a single decision model, we are interested in constructing a complete family of models on various levels of complexity, i.e., a catalogue of decision models, so as to be prepared for making decisions under different conditions. Formally, a decision catalogue is a finite sequence

$$H = (h_1, \ldots, h_J) \in \mathcal{H}^J \tag{2}$$

of increasing complexity: there exists a $\mathcal{H}_1 \sqsubseteq \mathcal{H}_2 \sqsubseteq \ldots \sqsubseteq \mathcal{H}_J$ in the partially ordered space of models such that $h_j \in \mathcal{H}_j$, $j \in \{1, \ldots, J\}$, and $c(h_1) < c(h_2) < \ldots < c(h_J)$; see again Fig. 2 for an illustration. Note that the length of a catalogue is not fixed, i.e., catalogues can be of different length and do not necessarily need to comprise all level of complexity. For example, a catalogue could make bigger "jumps" in $\mathbb{H}$, although this does not well align with the goal of a seamless transition from simple to complex models. It could,

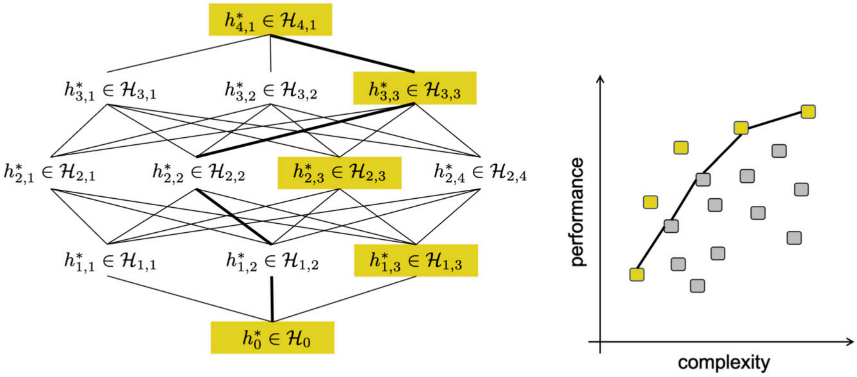
$h^*_{4,1} \in \mathcal{H}_{4,1}$
$h^*_{3,1} \in \mathcal{H}_{3,1}$
$h^*_{3,2} \in \mathcal{H}_{3,2}$
$h^*_{3,3} \in \mathcal{H}_{3,3}$
$h^*_{2,1} \in \mathcal{H}_{2,1}$
$h^*_{2,2} \in \mathcal{H}_{2,2}$
$h^*_{2,3} \in \mathcal{H}_{2,3}$
$h^*_{2,4} \in \mathcal{H}_{2,4}$
$h^*_{1,1} \in \mathcal{H}_{1,1}$
$h^*_{1,2} \in \mathcal{H}_{1,2}$
$h^*_{1,3} \in \mathcal{H}_{1,3}$
$h^*_0 \in \mathcal{H}_0$
performance
complexity

Fig. 2. Left: Illustration of partially (level-wise) ordered model classes (the higher the level, the more complex), with an optimal model $h^*_{i,j}$ in each space $\mathcal{H}_{i,j}$. Right: Models as points in a complexity/performance diagram. The Pareto-optimal models are marked in green. The line connects the models of a decision catalogue. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

however, simplify the decision maker's task of selecting a suitable model from the catalogue in a concrete context, which, although not considered in this paper, is an interesting and non-trivial problem by itself — having too many choices may then cause a cognitive overload.

In addition to increasing the complexity, a catalogue might be assumed to satisfy various *coherence constraints*, to guarantee practical usefulness and cognitive plausibility. For example, if the h_i are linear functions $\sum_{x_k \in F_i} w_k \cdot x_k$ on feature subsets $F_i \subseteq \mathcal{F}$, the decision maker may request that the weight of a feature does not change from model to model: If $x_k \in F_i \cap F_j$, then the weight w_k is the same in h_i and h_j. In general, the type of constraint will of course strongly depend on the model class and the application at hand. We denote by $\mathbb{C}$ the set of all feasible catalogues over $\mathbb{H}$, i.e., catalogues of increasing complexity and meeting all coherence constraints.

To evaluate the performance of catalogues, we suppose a loss function

$$L : \mathbb{C} \longrightarrow \mathbb{R} \tag{3}$$

to be defined, so that $L(H)$ is the loss assigned to a catalogue $H = (h_1, \dots, h_J)$. This loss can be an aggregation of "local losses" $\ell(h_i)$ assigned to individual models, but can also be defined in a more general way, maybe taking other criteria into account (e.g., the length of the catalogue). Then, the task of the learner is to find a catalogue that minimizes L (in expectation[2]), i.e., to find

$$H^* \in \arg\min_{H \in \mathbb{C}} \mathbb{E}\big[L(H)\big] . \tag{4}$$

Note that the problem of finding H^* is normally not decomposable, i.e., cannot simply be solved by following a divide-and-conquer strategy: Finding optimal models h^*_i (according to the loss ℓ on individual models) on different levels of complexity and combining these into a catalogue H^* will yield a catalogue that is likely to violate coherence constraints; see again Fig. 2, where the (optimal) decision catalogue contains models that are not Pareto-optimal.

Also note that the space of candidate catalogues will typically be huge. As an example, consider again the case where models can use different feature subsets $F \subseteq \mathcal{F}$ and complexity is measured in terms of $|F|$. There are

$$\prod_{k=1}^{K} \binom{K}{k} = \mathcal{O}\left(K^{K^2}\right)$$

many sequences of feature subsets progressively increasing complexity from 1 to K, and this number does not yet account for any model parameters. Even if coherence constraints may reduce the space of candidate catalogues (e.g., only nested sequences of feature subsets might be allowed, reducing the above number to $K!$), $\mathbb{C}$ will remain extremely large. Combined with a complex (non-decomposable) loss function L, this means that the learning task (4) can have a very high complexity. In fact, to guarantee that a best catalogue is found, e.g., one that minimizes empirical loss on the training data, a complete enumeration of $\mathbb{C}$ might be unavoidable. Therefore, exact optimization will usually not be tractable.

In the next section, we propose to tackle the problem by means of generic search techniques. In particular, heuristic search methods can then be used to make the learning problem tractable, albeit at the cost of losing optimality. Of course, other approaches might be conceivable as well. For example, one may think of using regularization, which is a common approach in machine learning. More concretely, by varying the regularization parameter, one may try to construct a model catalogue in the form of a "regularization path" [27]: A stepwise reduction of regularization will produce a model sequence of increasing complexity. However, this is a very

[2] The loss of a predictive model is normally based on its generalization performance, which can only be estimated.

indirect way of controlling complexity, and using it to construct model catalogues satisfying arbitrary coherence constraints will be quite difficult.

3.3. Constructing catalogues through search

As a generic approach to learning catalogues, we propose the use of systematic (heuristic) search methods to explore the space of feasible catalogues $\mathbb{C}$, taking advantage of the structure $\sqsubseteq$ on $\mathbb{H}$. Thus, the idea is to construct a catalogue step by step by navigating the structure $(\mathbb{H}, \sqsubseteq)$ in one way or the other. Such a process will eventually build on two basic operations, namely, model simplification and refinement (reduction and increase in complexity).

Definition 3.1 *(Simplification, refinement).* Consider a subspace $\mathcal{H}_t \in \mathbb{H}$ and a decision model $h_t \in \mathcal{H}_t$. Let $\mathcal{H}_t^+$ be the set of direct extensions of $\mathcal{H}_t$, that is, the set of all $\mathcal{H} \in \mathbb{H}$ such that $\mathcal{H}_t \sqsubset \mathcal{H}$, and there is no $\mathcal{H}' \in \mathbb{H}$ with $\mathcal{H}_t \sqsubset \mathcal{H}' \sqsubset \mathcal{H}$. Likewise, denote by $\mathcal{H}_t^-$ the set of direct reductions of $\mathcal{H}_t$, that is, the set of $\mathcal{H} \in \mathbb{H}$ such that $\mathcal{H} \sqsubset \mathcal{H}_t$, and there is no $\mathcal{H}' \in \mathbb{H}$ with $\mathcal{H} \sqsubset \mathcal{H}' \sqsubset \mathcal{H}_t$. Then, a transition from $(\mathcal{H}_t, h_t)$ to some $(\mathcal{H}_{t+1}, h_{t+1})$, where $\mathcal{H}_{t+1} \in \mathcal{H}_t^+$, is called a *direct refinement*. Likewise, a transition to some $(\mathcal{H}_{t+1}, h_{t+1})$ with $\mathcal{H}_{t+1} \in \mathcal{H}_t^-$ is a *direct simplification*.

We suppose a search strategy to provide suitable simplification and refinement operators, which allow the learner to navigate in the space of decision models. Thus, if $\mathcal{H}_t \in \mathbb{H}$ is the subspace considered in the t^{th} iteration of the search, and $\hat{h}_t \in \mathcal{H}_t$ the model learned in that space, simplification, or refinement operators can be applied to realize a transition to $(\mathcal{H}_{t+1}, \hat{h}_{t+1})$, where $\mathcal{H}_{t+1} \in \mathcal{H}_t^+$ or $\mathcal{H}_{t+1} \in \mathcal{H}_t^-$. More specifically, to accomplish a transition of that kind, two subproblems must be solved. First, the set of *candidate* transitions must be determined, that is, the set of possible simplifications or refinements. These are determined essentially by the sets $\mathcal{H}_t^-$ and $\mathcal{H}_t^+$. Second, a preferred simplification or refinement must be selected among the candidates, in line with the global loss L to be minimized.

In addition, a global search strategy is needed. In principle, any (informed) search method can be used for this purpose, including A^*, Monte Carlo tree search, etc. The strategy determines how the entire space $\mathbb{H}$ of subclasses (and hence the space of catalogues $\mathbb{C}$) is traversed. On the one side, search within this space must be greedy to some extent, because an exhaustive search will generally be infeasible. On the other side, the strategy should, as much as possible, avoid the danger of getting trapped in local optima, that is, of ending up with a catalogue H that can no longer be improved locally (in terms of the loss $L(H)$).

4. The case of scoring systems

Scoring systems have a long history of active use in safety-critical domains such as healthcare and justice. Exemplary applications include the diagnosis of acute coronary syndrome in patients with chest pain [50] or criminal recidivism prediction [63]. In a nutshell, they consist of a set of simple criteria (presence or absence of certain characteristics or features) that are checked, and if satisfied, contribute a certain number of points to a total score. The final decision is then based on comparing this score to one or more thresholds. Formally, scoring systems can be seen as a specific type of generalized additive models [28] defined over a set of features.

Definition 4.1 *(Scoring system).* A *scoring system* (over a set of candidate features $\mathcal{F}$ and score set $\mathcal{S} \subset \mathbb{Z}$) is a triple $h = \langle F, S, t \rangle$, where $F = (f_1, \ldots, f_K)$ is a sequence of (pairwise distinct) features $f_j \in \mathcal{F}$, $S = (s_1, \ldots, s_K) \in \mathcal{S}^K$ are scores assigned to the corresponding features, and $t \in \mathbb{Z}$ is a decision threshold. For a given decision context $\boldsymbol{x} = (x_1, \ldots, x_K)$, i.e., the projection of an instance to the feature set F, the decision prescribed by h is given by

$$h(\boldsymbol{x}) = \left[\!\!\left[\sum_{i=1}^{K} s_i \, x_i \geq t \right]\!\!\right] , \tag{5}$$

where $[\![\cdot]\!]$ is the indicator function, i.e., $[\![P]\!] = 1$ if predicate P is true (positive decision) and $[\![P]\!] = 0$ if P is false (negative decision).

A remark on notation: In the following, we will treat selections of features from $\mathcal{F}$ interchangeably as sequences, like in the previous definition, or sets $F \subset \mathcal{F}$. In both cases, $|F|$ denotes the number of features included in the sequence or set. The meaning should always be clear from the context.

While scoring systems have often been handcrafted by domain experts in the past, more recent methods for the data-driven construction of scoring systems aim to achieve a good trade-off between the complexity of models and the quality of their recommendations [59]. As already said, this is crucial for the successful adoption of decision models in practice, as overly complex models are difficult to analyze by domain experts and impede the manual application by human practitioners.

Given a (maximal) set of candidate features $\mathcal{F}$, there are various ways to equip the space of scoring systems over $\mathcal{F}$ with a structure $(\mathbb{H}, \sqsubseteq)$, as well as various possibilities to define the complexity of scoring systems. For example, complexity could be defined by the cardinality of the underlying score set $\mathcal{S}$: A decision model using only scores ±1 is clearly simpler but also less expressive than a model using scores between -10 and $+10$. Or, complexity could be controlled by the number of features: A decision model with only 5 features is simpler but also less expressive than a model with 10 features. In this case, that we shall elaborate on further in this and the next section, a structure $(\mathbb{H}, \sqsubseteq)$ is naturally induced by the subset relation on feature sets: $\mathbb{H} = \{H_F \mid F \subseteq \mathcal{F}\}$ and $H_F \sqsubseteq H_G$ iff

$F \subseteq G$. Moreover, a meaningful coherence constraint on catalogues is monotonicity, in the sense that later members of a catalogue are formed by adding features and scores, and maybe by adaptation of the threshold, but without changing previously added features and scores.

Definition 4.2 *(Monotonic scoring systems).* A catalogue of scoring systems $H = (h_1, \dots, h_M)$ with $h_j = \left\langle F_j, \mathcal{S}_j, t_j \right\rangle$ is *monotonic*, iff

(i) $F_j = F_i \parallel (f_{j,i+1}, \dots, f_{j,j})$ for all $i < j$, and
(ii) $\mathcal{S}_j = \mathcal{S}_i \parallel (s_{j,i+1}, \dots, s_{j,j})$,

where $\parallel$ denotes vector concatenation. Thus, with $F_i = (f_{i,1}, \dots, f_{i,i})$ and $F_j = (f_{j,1}, \dots, f_{j,j})$ we have $(f_{i,1}, \dots, f_{i,i}) = (f_{j,1}, \dots, f_{j,i})$. Likewise, $s_{j,k} = s_{i,k}$ for $k = 1, \dots, i$.

Note that this definition does not impose any restrictions on the thresholds t_j used by the individual models. Because of this, but also in general, monotonic scoring systems do not imply any monotonicity w.r.t. the examples classified as positive, i.e., examples classified as positive by some catalogue members may be classified as negative by later ones, and vice versa. There are, however, interesting special cases, as we will see in the next section.

4.1. *Expressivity of scoring systems*

Looking at (5), it is clear that scoring systems can be seen as restricted linear classifiers, namely, linear classifiers with weights s_i restricted to the score set $\mathcal{S}$. In spite of this restriction, the expressivity of scoring systems is not much lower, at least theoretically. Obviously, every linear model can be approximated arbitrarily closely by increasing the size of $\mathcal{S}$. But even if scores are restricted to ±1, it is easy to see that K instances can be shattered by scoring systems (take instances $\boldsymbol{x}_1, \dots, \boldsymbol{x}_K$, where $\boldsymbol{x}_k$ has the k^{th} feature present and all others absent), wherefore the VC dimension is (at least) K in this case (compared to $K+1$ for linear classifiers).

Interestingly, scoring systems may also be viewed as generalizations of other feature-based classifiers. Let us consider the simplest type of scoring systems with score set $\mathcal{S}_{+1} = \{+1\}$, i.e., all added features receive the same constant score $+1$. Thus, the scoring system $\left\langle F, \mathcal{S}_{+1}, t \right\rangle$ is determined by the set of selected features F and a threshold $t \in \mathbb{N}_0$. Special cases include the all-positive classifier $\left\langle F, \mathcal{S}_{+1}, 0 \right\rangle$, which classifies all examples as positive, the all-negative classifier $\left\langle F, \mathcal{S}_{+1}, |F|+1 \right\rangle$, but also elementary operators such as disjunction ($\left\langle F, \mathcal{S}_{+1}, 1 \right\rangle$) or conjunction ($\left\langle F, \mathcal{S}_{+1}, |F| \right\rangle$) of a subset of features F. More generally, the scoring system $\left\langle F, \mathcal{S}_{+1}, m \right\rangle$ represents a so-called m-of-n concept, also called criteria table [56], which classifies an example as positive if at least m of the specified $n = |F|$ features are positive. Such m-of-n concepts have primarily been used for feature construction in decision tree [39] or rule learning [64,3] algorithms.

A catalogue of scoring classifiers based on $\mathcal{S}_{+1}$ therefore corresponds to $\mathbb{H} = \{\mathcal{H}_F \mid F \subseteq \mathcal{F}\}$, where $\mathcal{H}_F$ is the set of scoring systems over the feature subset F, i.e., the elements of $\mathcal{H}_F = \{\left\langle F, \mathcal{S}_{+1}, m \right\rangle \mid m \leq |F| \in \mathbb{N}_0\}$ differ only with respect to their thresholds. The relation $\sqsubseteq$ on $\mathbb{H}$ is then naturally inherited from the inclusion relation $\subseteq$ on subsets, i.e., $\mathcal{H}_F \sqsubseteq \mathcal{H}_G$ iff $F \subseteq G$. The set of refinements of $\mathcal{H}_F$ is given by $\mathcal{H}_F^+ = \{\mathcal{H}_G \mid G = F \cup \{f_i\}, f_i \notin F\}$, and the set of simplifications by $\mathcal{H}_F^- = \{\mathcal{H}_G \mid G = F \setminus \{f_j\}, f_j \in F\}$. A natural measure of model complexity is the number of features used by a scoring system, i.e., $c(h) = |F|$ for all $h \in \mathcal{H}_F$. Moreover, as already said, a meaningful coherence constraint on catalogues $(h_1, \dots, h_M)$ is the requirement $F_i \subset F_j$ for $i < j$, which means that all features used by a simpler model h_i are also used by a more complex model h_j.

In this setting, it is easy to see the following property[3]:

Remark 4.1. In a monotonic catalogue $(h_1, \dots, h_M)$ defined over constant scores $\mathcal{S}_{+1} = \{+1\}$, h_{i+1} is either a generalization or a specialization of h_i.[4]

Proof. Let $h_i = \left\langle F_i, \mathcal{S}_{+1}, t_i \right\rangle$ and $h_{i+1} = \left\langle F_{i+1}, \mathcal{S}_{+1}, t_{i+1} \right\rangle$ with $F_{i+1} = F_i \parallel (f_{i+1})$. Thus, F_{i+1} extends F_i by adding a single feature f_{i+1} and changing the threshold from t_i to t_{i+1}. We can distinguish two cases:

(i) $t_{i+1} > t_i$: As $t_i, t_{i+1} \in \mathbb{N}_0$, $t_{i+1} \geq t_i + 1$. Let us consider the case $t_{i+1} = t_i + 1$. An example that is classified as positive by h_{i+1} must satisfy $t_{i+1} = t_i + 1$ of the features in $F_{i+1} = F_i \cup \{f_{i+1}\}$. If it satisfies f_{i+1}, it must then satisfy t_i features of F_i, and is therefore also classified as positive by h_i. On the other hand, if it does not satisfy f_{i+1}, it must satisfy $t_{i+1} > t_i$ of the features in h_i, and is therefore also classified as positive by h_i. Thus, every example that is classified as positive by h_{i+1} is also classified as positive by h_i, and therefore h_{i+1} is a specialization of h_i. This argument can be easily extended to all $t_{i+1} > t_i$.
(ii) $t_{i+1} \leq t_i$: Every example that is classified as positive by h_i has t_i or more of the features F_i. As $t_{i+1} \leq t_i$ and $F_i \subset F_{i+1}$, it also satisfies at least t_{i+1} of the features in F_{i+1}. Thus, F_{i+1} is a generalization of F_i. □

[3] Recall that a model h is called a generalization of h', and h' a specialization of h, if $h(\boldsymbol{x}) = 1$ implies $h'(\boldsymbol{x}) = 1$ for all $\boldsymbol{x} \in \mathcal{X}$, i.e., all $\boldsymbol{x}$ predicted positive by h' are also predicted positive by h.

[4] Note that this proposition is non-tautological, as it is possible that none of the two relations holds (h_{i+1} is neither a generalization of h_i nor a specialization).

Of course, the simple case of constant scores is only a delimiting case. It is easy to see that extending the score set increases the expressivity of the scoring classifiers. We illustrate this for the following special case:

Remark 4.2. Extending the score set from $\mathcal{S}_{+1} = \{+1\}$ to $\mathcal{S}_{\pm 1} = \{-1, +1\}$ strictly increases the expressivity of a catalogue of scoring systems.

Proof. Trivially, every catalogue defined over $\mathcal{S}_{+1}$ is also defined over $\mathcal{S}_{\pm 1} \supset \mathcal{S}_{+1}$. To see that the latter strictly increases the expressivity, we consider $h_i = \left\langle F_i, \mathcal{S}_{+1}, t \right\rangle$, and refine it to $h_{i+1} = \left\langle F_{i+1}, \mathcal{S}_{\pm 1}, t \right\rangle$ by adding a feature f_{i+1} with a negative weight of -1 and leaving the threshold t unchanged. An example that does not satisfy f_{i+1} must therefore satisfy t of the features in F_i, whereas an example that does satisfy f_{i+1} must satisfy at least $t+1$ features in F_i in order to meet the threshold t. Thus, the resulting classifier corresponds to a conditional statement,

$$\textbf{if } f_{i+1} \textbf{ then } h_{i+1} = \left\langle F_i, \mathcal{S}_{+1}, t \right\rangle \textbf{ else } h_{i+1} = \left\langle F_i, \mathcal{S}_{+1}, t+1 \right\rangle,$$

which can in general not be modelled with a single threshold over $\mathcal{S}_{+1}$. □

As an illustration, scoring systems with three features, respective scores +1, +1, −1, and threshold 1 induce the following model:

x_1	x_2	x_3	$h(x)$
0	0	0	0
0	0	1	0
0	1	0	1
1	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1
1	1	1	1

Obviously, this model cannot be produced with score set $\mathcal{S}_{+1}$. For example, that latter implies monotonicity, which is violated here (e.g., $h(1,0,0) = 1 < 0 = h(1,0,1)$).

Note, however, that including negative weights is equivalent to including positive weights with negated features. More precisely, assume that $F = P \cup N$, where P is the set of features that are associated with a positive weight $s_i > 0$, and N are the features associated with a negative weight $s_j < 0$, and $S_P = \{s_i \mid f_i \in P\}$ and $S_N = \{s_j \mid f_j \in N\}$ the sets of positive and negative scores, respectively. Also let us denote with $\bar{X} = \{\neg f \mid f \in X\}$ the set of negated features in X, and with $|S| = \{|s| \mid s \in S\}$ the set of absolute values of the scores S. We can then show the following result, which is also part of Theorem 9.6 in [10].

Remark 4.3. Scoring system $h = \left\langle P \cup N, S, t \right\rangle$ is equivalent to scoring system $\bar{h} = \left\langle P \cup \bar{N}, |S|, t + t_{\bar{N}} \right\rangle$, where $t_{\bar{N}} = \sum_{s_j \in S_N} |s_j|$.

In principle, we can thus restrict ourselves to scoring systems with only positive scores, including (or excluding) negated features as desired. In practice, one may nevertheless prefer the use of possibly negative scores, because the original features might be more natural and easier to interpret.

4.2. Probabilistic scoring systems

Scoring systems as introduced above provide deterministic decisions. In safety-critical domains like medical decision-making, one may also be interested in a representation of the systems uncertainty in the form of probability distributions. This suggests a probabilistic extension of scoring systems [9,26,61].

Definition 4.3 *(Probabilistic scoring system).* A *probabilistic scoring system* over candidate features $\mathcal{F}$ and score set $\mathcal{S} \subset \mathbb{Z}$ is a quadruple $h = \left\langle F, S, q, t \right\rangle$, where $F = (f_1, \dots, f_K)$ is a sequence of (pairwise disjunct) features $f_j \in \mathcal{F}$, $S = (s_1, \dots, s_K) \in \mathcal{S}^K$, and q is a mapping $\Sigma \longrightarrow [0,1]$, where

$$\Sigma := \left\{ T = \sum_{i=1}^{K} s_i \, x_i \;\middle|\; s_1, \dots, s_K \in \mathcal{S},\, x_1, \dots, x_K \in \{0,1\} \right\}$$

is the set of possible values for the total score that can be obtained by any instance $\boldsymbol{x} \in \mathcal{X}$, and $q(T) = p(y = 1 \mid T)$ is the (estimated) probability for the positive class $(y = 1)$ given that the total score is T (and hence $1 - q(T)$ the probability for the negative class). Moreover, t is a suitably chosen probability threshold prescribing a (deterministic) decision (5).

An increase in the total score should naturally increase but not decrease the probability of the positive decision, so that probabilistic scoring systems should satisfy the following monotonicity constraint:

$$\forall T, T' \in \Sigma : (T < T') \Rightarrow q(T) \leq q(T') . \tag{6}$$

This property is in line with standard scoring systems and appears to be important from an interpretability perspective: A violation of (6) would compromise the acceptance of the decision model. Therefore, we consider only monotonic probabilistic scoring systems.

The decision threshold t is needed for taking action, i.e., whenever a concrete decision (under uncertainty) has to be made. Given an underlying loss function ℓ as performance measure, t is naturally chosen to minimize this loss in expectation: Given total score T, the expected loss of a positive and negative decision is given by $(1 - q(T)) \cdot \ell(0,1)$ and $q(T) \cdot \ell(1,0)$, respectively (assuming that $\ell(0,0) = \ell(1,1) = 0$). Under the assumption (6), the former is monotone decreasing and the latter is monotone increasing in T. Thus, t can be defined by the smallest score T such that $(1 - q(T)) \cdot \ell(0,1) < q(T) \cdot \ell(1,0)$.

4.3. Learning catalogues of scoring systems: a greedy approach

Let us return to the problem of learning decision catalogues (2). More specifically, we consider the case where each decision model h_j in a catalogue $H = (h_1, \dots, h_M)$ is a probabilistic scoring system $h_j = \langle F_j, S_j, q_j, t_j \rangle$, and where the systems are monotonic in the sense of Definition 4.2. In this case, the search for an optimal scoring system closely resembles the problem of feature selection [25], which is known to be computationally challenging because of the exponential size of the search space. Therefore, analogous to forward selection in feature subset search [31], we propose a simple greedy strategy for constructing a decision catalogue.

Our strategy starts with the empty feature set $F_1 = \emptyset$, i.e., the first model $h_1 = \langle \emptyset, \{0\}, q_1, t_1 \rangle$ in the catalogue suggests a default decision, either positive ($t_1 = 0$) or negative ($t_1 > 0$), which does not use any feature. The decision depends on the probability $q_1(0)$, which is (an estimate of) the marginal probability of the positive class.

Then, a single feature f_j with a corresponding score s_j is added at each iteration, such that the resulting extension leads to the maximal improvement in terms of a given loss L. Note that, to determine the improvement, each combination of candidate feature and corresponding score must be combined with a mapping q_j, wherefore the outer search for optimal features includes an inner optimization over mappings from the set of total scores to probability estimates. Adopting a standard frequentist approach, probabilities $q_j(T)$ are estimated in terms of relative frequencies P_T / N_T, where N_T is the number of training examples with total score T, and P_T is the number of examples with total score T and positive class $y = 1$. However, as these estimates are obtained independently for each score T, they may violate the monotonicity condition (6). A better idea, therefore, is to estimate them jointly using a probability calibration method [14]. To this end, the original data $\mathcal{D}$ is first mapped to the data,

$$\mathcal{C} := \left\{ (T(\boldsymbol{x}), y) \mid (\boldsymbol{x}, y) \in \mathcal{D} \right\} \subset \Sigma_j \times \mathcal{Y} ,$$

to which any calibration method can then be applied. Following Hanselle et al. [26], we make use of isotonic regression [40] for that purpose, which amounts to finding values $q_j(T)$ solving the following constrained optimization problem:

$$\begin{aligned} &\text{minimise} \quad \sum_{(T,y) \in \mathcal{C}} \left(q_j(T) - y \right)^2 \\ &\text{s. t.} \quad \forall T, T' \in \Sigma_j : (T < T') \Rightarrow (q_j(T) \leq q_j(T')) \end{aligned}$$

Algorithm 1 outlines the greedy procedure for constructing a catalogue of scoring systems as described above. Greedy algorithms make irrevocable locally optimal decisions in each step, which may come at the risk of missing globally optimal solutions. To allow for a less greedy behaviour, we employ a l-step lookahead search. Here, we consider catalogue extensions of length l in each stage, which comes at considerable computational cost but mitigates the aforementioned risk. In addition to the training examples and their corresponding labels, the algorithm also takes a subset of features F and scores S that may be associated with them as input arguments. This allows one to construct scoring systems that are restricted to a predefined set of scores, including $S = \{-1, +1\}$ as used in the previous example, but also more flexible choices of score sets. The complexity function c and the loss function L can also be chosen freely. Hanselle et al. [26] propose the use of an expected entropy measure for assessing the impurity of probabilistic scoring systems:

$$\ell(h_j) = \sum_{T \in \Sigma_j} \frac{N_T}{N} \cdot \mathrm{Ent}\left(q_j(T) \right) \tag{7}$$

where N_T denotes the number of examples with total score T among all N examples in the training data, and the Shannon entropy

$$\mathrm{Ent}(p) = -p \cdot \log(p) - (1-p) \cdot \log(1-p) .$$

The corresponding global loss L is then given by summing up the individual local losses $L(H) = \sum_{h_j \in H} \ell(h_j)$ and the complexity of a scoring system is given by the number of its features $c(h) = |F|$. However, other instantiations of Algorithm 1 are possible and will be discussed in our experimental evaluation. An implementation of Algorithm 1, which we subsequently refer to as GreedyCatalogue, is available at GitHub.[5]

As already mentioned, even an optimal catalogue may not necessarily contain an optimal model $h_k^* \in \mathcal{H}_k$, where $\mathcal{H}_k = \bigcup_{|F|=k} \mathcal{H}_F$, i.e., a model minimizing the loss among those with a predefined complexity k. Nevertheless, there are good reasons to believe that

[5] https://github.com/TRR318/scikit-psl/tree/v0.6.1.

Algorithm 1: GreedyCatalogue.

```
input  : dataset $X$ and target labels $y$,
         set of all features $\mathcal{F}$ and available scores $\mathcal{S}$,
         lookahead length $l$, loss function $\ell$ to evaluate a hypothesis
output : decision catalogue $H$
$F, S, H = (), (), ()$
# While not all features have been used. For set difference and inequality operators we treat $F$ as a
  set for ease of notation.
while $F \neq \mathcal{F}$ do
    # $\bar{F}$ contains all remaining features
    $\bar{F} = \mathcal{F} \setminus F$
    $l' = \min(l, |\bar{F}|)$
    # Evaluate all possible extensions of length $l$
    $F^+ = \left\{(f_i) \in \bar{F}^{l'} \mid \forall k, j : f_k \neq f_j\right\}$
    $(f_i), (s_i) = \arg\min_{(f_i) \in F^+ \times (s_i) \in \mathcal{S}^{l'}} \left(\text{FITSCORE}(F \parallel (f_i), S \parallel (s_i))\right)$
    $F = F \parallel (f_1)$
    $S = S \parallel (s_1)$
    $H = H \parallel \left((F, S, q, t)\right)$
return $H$

Function FITSCORE (F,S):
    $L = 0$
    for $i \leftarrow 0$ to $|F|$ do
        $h = \left\langle (f_k \mid k \leq i), (s_k \mid k \leq i), q \right\rangle$
        $L \mathrel{+}= \ell(y, h(X), t)$
    return $L$
```

the catalogues found by our greedy search will come close to h_k^*: Calinescu et al. [6] show that a greedy algorithm achieves an $1-1/e$ approximation to the best possible solution when maximizing a monotone submodular function over a uniform matroid. In our case, $(\mathbb{H}, \sqsubseteq)$ does have the structure of a uniform matroid, and the performance measure is at least likely to be submodular. Recall that a function g on subsets of a set U is submodular if $g(A \cup \{u\}) - g(A) \geq g(B \cup \{u\}) - g(B)$ for all elements $u \in U$, whenever $A \subset B \subseteq U$. In our case, this means that the increase in performance due to adding a feature f_j to a smaller feature subset F is at least as big as the gain due to adding it to a bigger superset B. This is normally the case, and would be even provably true if we defined $\mathcal{H}_F$ as the set of scoring systems using *at most* (instead of *exactly*) the features in F.

5. Experimental evaluation

In this section, we present different experimental studies meant as a first evaluation of our approach. We start with two case studies in medical decision-making, one aimed at the diagnosis of coronary heart disease, and another one concerned with the detection of SARS-Cov-2 cases. In addition, we present experimental results on benchmark datasets from the UCI machine learning repository.

All experiments were conducted by a 50-fold Monte-Carlo cross validation, trained on 2/3 of the available data and evaluated on the remaining third. The greedy search was configured with a lookahead parameter of $l = 2$. The error bands in the following plots always show the 95% confidence interval of the mean. The set of scores is restricted to $\{\pm 1, \pm 2, \pm 3\}$ for all learning algorithms.

5.1. Datasets

Thorax This case study is based on a dataset originally used to evaluate the diagnostic accuracy of symptoms and signs for coronary heart disease (CHD) in patients presenting with chest pain in primary care. Chest pain is a common complaint in primary care, with CHD being the most concerning of many potential causes. Based on the medical history and physical examination, general practitioners (GPs) have to classify patients into two classes: patients in whom an underlying CHD can be safely ruled out and patients in whom chest pain is probably caused by CHD.

Briefly, 74 GPs recruited consecutively patients aged ≥ 35 who presented with chest pain as a primary or secondary complaint. GPs took a standardized history and performed a physical examination. Patients and GPs were contacted six weeks and six months after the consultation. All relevant information about the course of chest pain, diagnostic procedures and treatments had been gathered during six months. An independent expert panel of one cardiologist, one GP and one research staff member reviewed each patient's data and established the reference diagnosis by deciding whether CHD was the underlying reason of chest pain. For details about the design and conduct of the study, we refer to Bösner et al. [4].

Overall, the dataset consists of 1199 (135 CHD and 1064 non-CHD) patients described by ten binary attributes: patient assumes pain is of cardiac origin, muscle tension, age gender compound, pain is sharp, pain depends on exercise, known clinical vascular

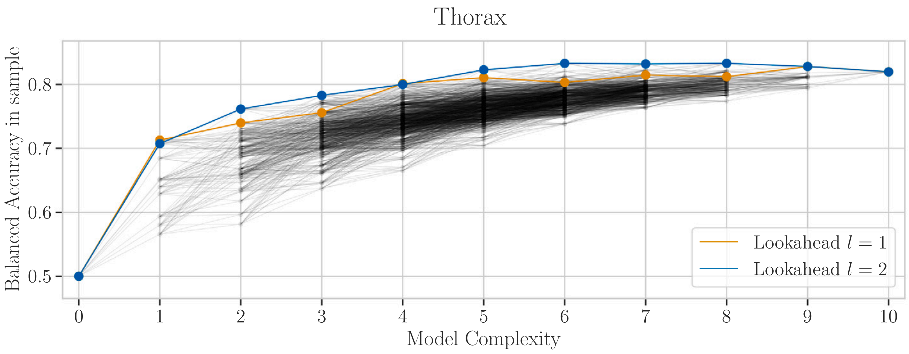

Fig. 3. Comparison of final catalogues found by the proposed algorithm and all possible probabilistic scoring models for the Thorax dataset. The score set was reduced to $\{1\}$. The greedy search with a lookahead of $l = 2$ selects almost exclusively models on the Pareto front. All coherent models that could be part of the same catalogue are connected with an edge.

disease, diabetes, heart failure, pain is not reproducible by palpation, patient has cough. Note that, by way of domain knowledge, all these features can be encoded in such a way that the presence of a feature does always increase the likelihood of the positive class. Therefore, scoring systems could be restricted to positive scores. This in turn means that scoring systems, in which all features have the same influence (i.e., have score +1 if being included), effectively reduce to m-of-n classifiers as discussed in Section 4.1.

Covid For our second case study, we rely on routinely collected and fully anonymized data from 12 German emergency departments [30], which has previously been used by Rapp et al. [44]. Each example in the dataset corresponds to a patient who has consulted one of the emergency departments between 2020 and 2021. The dataset contains 11 binary features that indicate the presence or absence of different symptoms, e.g., fever, cough, loss of smell, or fatigue. In addition, for each patient, the result of a PCR test has been recorded. Our goal is to detect possible cases of *SARS-Cov-2* based on the available indicators and using the result of the PCR tests as the ground truth. The dataset was filtered by removing examples with missing feature values or ground truth labels, resulting in nearly 700 examples with less than 10% Covid positive cases.

UCI datasets In addition to the two previously introduced dataset, we evaluated our approach on the following two datasets: Adult (id = 2) and Heart (id = 45). The Heart dataset is similar to the already presented Thorax dataset.[6] For all datasets, we removed entries with missing features. The datasets have been preprocessed to encode categorical features with a one-hot vector and by binarizing or quantizing continuous features. Quantized features were also one-hot encoded. Experiments on the Adult dataset have been downsampled to 1000 datapoints for each cross-validation split. All details on the encoding process can be found in the experimental repository.[7]

5.2. Solution quality of greedy search and comparison with SLIM

As the distribution of positive and negative examples in many of the previously described datasets is very imbalanced (many negative examples and few positive examples), we analyze performance with respect to *balanced accuracy* metric [5]:

$$\frac{1}{\sum_{i=1}^{N} \hat{w}(y_i)} \sum_{i=1}^{N} [\![h(\boldsymbol{x}_i) = y_i]\!]\, \hat{w}(y_i), \qquad \text{with} \qquad \hat{w}(y) = \frac{1}{\sum_{i=1}^{N} [\![y = y_i]\!]}.$$

For these experiments, we instantiate Algorithm 1 with a loss function ℓ that corresponds to the balanced accuracy, i.e., optimizing the metric we also use for evaluation (other choices are investigated in the next section). It conducts a greedy lookahead search through the space of monotonic decisions catalogues. This search space is induced by considering all feature permutations together with all possible score assignments. Due to the greediness of our approach, it is an interesting question how well it is able to identify optimal catalogues.

Fig. 3 depicts the balanced accuracy for all possible monotonic decision catalogues of probabilistic scoring systems for the Thorax dataset. The blue and orange lines indicate the catalogues that have been identified by GREEDYCATALOGUE (see Algorithm 1) when setting the lookahead parameter to $l = 1$ (i.e. a purely greedy configuration) and $l = 2$. While the greedy configuration manages to find a fairly good catalogue, making irrevocable locally optimal decisions on each stage leads to missing globally optimal solutions. This is exemplified by considering stage 1 in Fig. 3. Unlike the greedy configuration, the search with lookahead $l = 2$ chooses a locally suboptimal feature-score-pair at stage 1, which is the only way to achieve optimal performance at stage 2. Overall, we observe that greedy search with lookahead leads to almost exclusively selecting points on (or close to) the Pareto front of performance and complexity. This supports our conjecture that greedy search yields good approximations in this setting.

[6] https://archive.ics.uci.edu.

[7] https://github.com/TRR318/pub-ijar-learning_cascades/tree/v1.2.0.

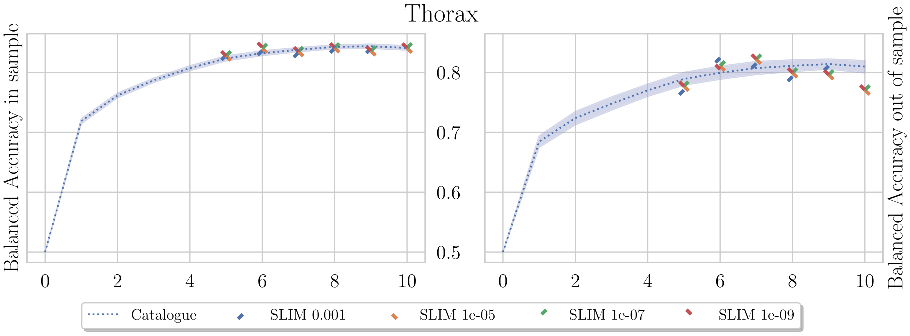

Fig. 4. Comparison between SLIM models and the greedy catalogue with respect to accuracy on the Thorax dataset. The error band shows the 95% confidence interval of the mean catalogues performance over all cross-validation folds. For SLIM, we display the median model for each stage and hyperparameter.

In addition to GreedyCatalogue, we also used SLIM [59] as a baseline to compare with. SLIM is an approach for learning sparse scoring systems via mixed-integer programming, i.e., it first learns a linear model with real-valued scores that minimizes the regularized training objective

$$\frac{w^+}{N} \sum_{i \in \{i | y_i = 1\}} \left[\!\left[y_i S^\top \boldsymbol{x}_i \leq 0 \right]\!\right] + \frac{w^-}{N} \sum_{i \in \{i | y_i = 0\}} \left[\!\left[y_i S^\top \boldsymbol{x}_i \leq 0 \right]\!\right] + C_0 \|S\|_0 + \epsilon \|S\|_1 ,$$

and subsequently converts it into a scoring system with discrete scores by conducting a best-first search. The weights w^+ and w^- can be chosen freely to either optimize for accuracy ($w^+ = w^- = 0.5$) or balanced accuracy ($w^+ = \frac{N^-}{N}, w^- = \frac{N^+}{N}$). N denotes the total number of instances, while N^+ and N^- are the number of positive ($y_i = 1$) and negative ($y_i = 0$) instances, respectively. However, because the search for an optimal solution can require a significant amount of time, we restricted the maximum runtime to 15 Minutes, after which the best model found so far is returned.

SLIM provides implicit control over model complexity (sparsity) via the regularization parameter C_0, which specifies the maximum absolute decline in terms of the target measure that is acceptable in favour of omitting a feature from the final model. We tested different values $C_0 \in \{10^{-3}, 10^{-5}, 10^{-7}, 10^{-9}\}$ for this trade-off between sparsity and predictive performance. The weight of SLIM's L_1-regularization term was kept at $\epsilon = 0.001$.[8] Fig. 4 shows a comparison of the catalogues constructed with GreedyCatalogue and scoring systems that were built using these configurations of SLIM on the Thorax dataset in terms of balanced accuracy. As we can see, the catalogues built by the greedy method are on average on par with those induced by SLIM. To obtain a deterministic prediction for GreedyCatalogue, the positive class was predicted once the $P(y = 1|x) > w^-$.

Looking at the plot in Fig. 4, one may wonder whether decision catalogues could not also be constructed using SLIM, i.e., by indirect control of complexity through regularization. This is in general quite difficult, for at least two reasons. First, it is not possible to reliably control the number of features, i.e., the complexity of models that SLIM constructs. The regularization parameter C_0 does trade off sparsity and performance, but is doing so only indirectly, and its effect on the number of features is hard to anticipate. Consider for example the SLIM configuration with $C_0 = 10^{-9}$ in Fig. 4. Such a low regularization parameter only imposes a very small penalty for complex models. Yet, we observe models of all complexity levels between 5 and 10, depending on the fold of the cross-validation. Across all 50 cross-validation runs, we only see one model for each regularization parameter $C_0 \leq 10^{-5}$ that uses all ten features. Secondly, even if it would be possible to reliably configure the amount of features SLIM uses, the individual SLIM models are built independently. Thus, there is no guarantee that the resulting catalogue fulfils the coherence (monotonicity) property in Definition 4.2, and in fact it is quite unlikely to do so. GreedyCatalogue satisfies these coherence constraints by design while still achieving competitive predictive performance.

5.3. Model performance with respect to optimization goal

As already discussed, GreedyCatalogue can be instantiated with any loss function. Fig. 5 shows the in-sample and out-of-sample balanced accuracy of decision catalogues when instantiating the learning algorithm with various loss functions. Similar to the previous section, we selected w^- as the decision threshold for the positive class. We observe that the best in-sample balanced accuracy is achieved when we use the balanced accuracy as an optimization objective, which is what one would expect. This is however not the case for the out-of-sample data. Here, the minimization of expected entropy (on the training data) leads to a similar predictive performance as directly optimizing for balanced accuracy. Overall, expected entropy seems to be a good and relatively robust measure for training decision catalogues (comparable to expected information gain for decision trees).

As an aside, note that catalogues with more features may exhibit worse performance, even on the training data. This is because the learner can only choose among a small set of discrete scores, which may prevent it from modulating the effect of individual

[8] Ustun and Rudin [59] recommend this parameter to be chosen small enough, for reasons explained in Section 2 of their paper. They effectively avoid using this parameter for regularization purposes.

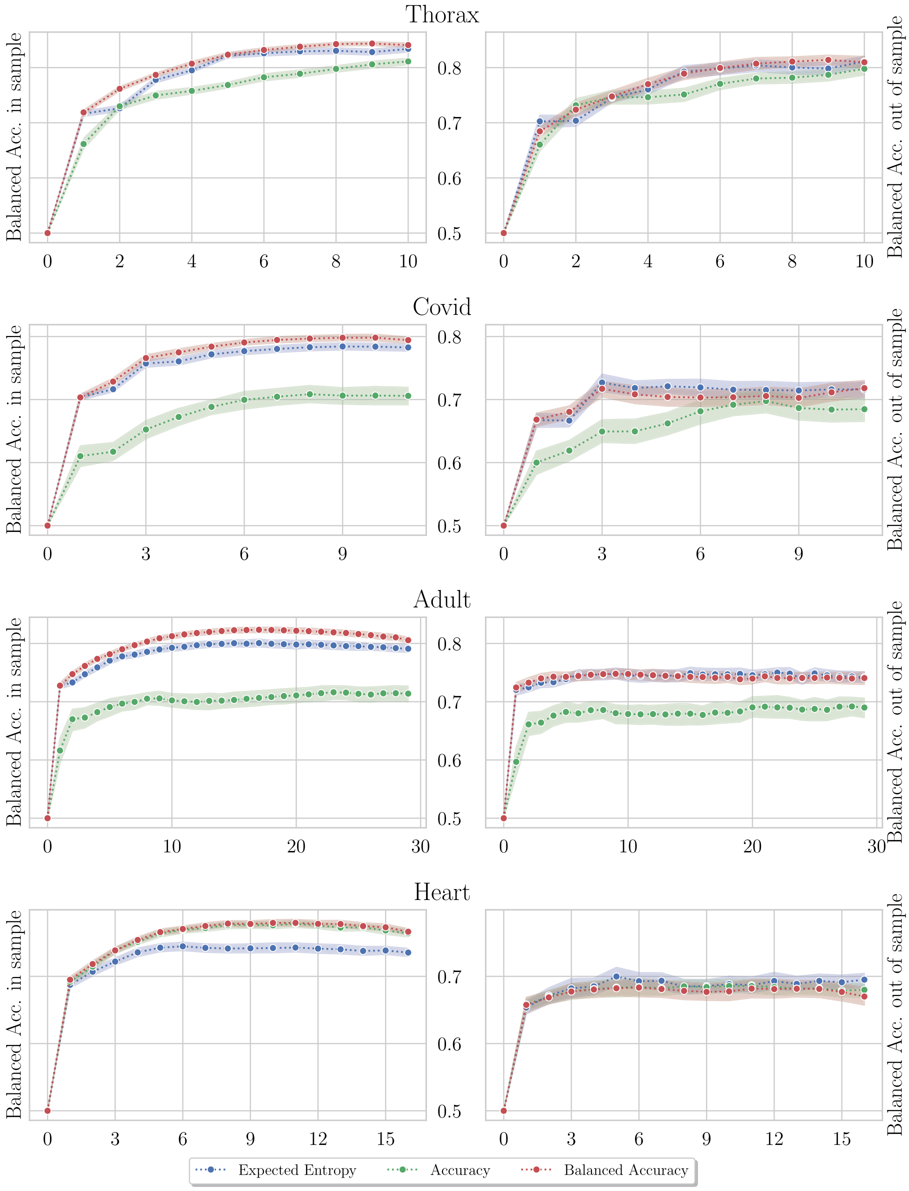

Fig. 5. In-sample and out-of-sample accuracy of the catalogue optimized with respect to the criteria *expected entropy*, *accuracy* and *balanced accuracy*.

features in a sufficiently fine-granular manner. Thus, instead of including a relevant but less important feature, and giving it the same or almost the same influence as more important features, it might be better to omit that feature altogether. In practice, it can therefore be reasonable to cut a catalogue when observing a significant decline in performance. On the other hand, keeping all features can increase robustness. In the Adult dataset, the performance out-of-sample does not decrease even with more than 20 features, and although the in-sample performances already decline.

Besides measures like expected entropy and (balanced) accuracy, one may also be interested in more application-specific metrics. In applications such as medical diagnoses, for example, it is often important to guarantee that positive cases are indeed identified as such. In other words, the recall of a predictor should be sufficiently high. To capture this idea, we define the *precision@recall* measure as follows:

$$precision@r(h) = \begin{cases} precision(h) & \text{if } recall(h) > r \\ 0 & \text{otherwise} \end{cases},$$

where precision(h) and recall(h) denote the precision and recall of the scoring system h, respectively. Note that this measure, which is parameterized by the recall-level r, is a utility rather than a loss function, i.e., it ought to be maximized rather than minimized. Moreover, like other measures such as AUC or F-measure, it is not defined per-instance but rather for a complete set of data (e.g., the training data and test data).

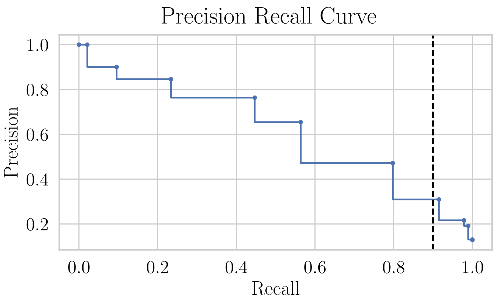

Fig. 6. Example of a precision-recall curve of a probabilistic scoring system on the Thorax dataset.

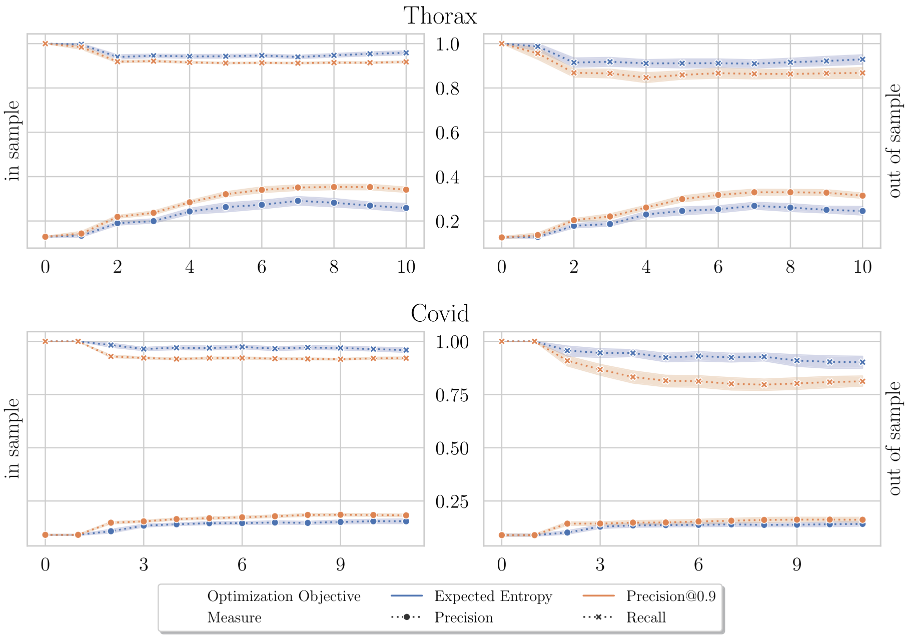

Fig. 7. In-sample and out-of-sample precision and recall of the decision catalogue when using precision@0.9 and expected entropy as the optimization objective of our algorithm.

Fig. 6 depicts a precision-recall curve of a probabilistic scoring system on the Thorax dataset. For a recall of $r = 0.9$, highlighted with the dashed line, the precision@0.9 ≈ 0.3. While the level of true positives is fixed, the refinements lead to fewer and fewer false positives as we progress to higher stages in the decision catalogue.

In Fig. 7, we report the precision and recall when instantiating our proposed algorithm with precision@0.9 and the expected entropy as optimization objective. Regardless of the optimization objective, the decision threshold for the predictions was chosen to maximize precision while keeping a recall of at least 0.9. More precisely, we calculated the optimal precision@0.9 in-sample and kept the decision threshold to measure precision and recall in- and out-of-sample. Respecting the in-sample performance, precision@0.9 leads to better values of precision while strictly keeping recall values close to 0.9. Expected entropy leads to more conservative decisions, classifying more instances as positive, and hence leading to higher recall and lower precision. For the out-of-sample case, there is no guarantee that a certain recall level is met. This can be seen both when optimizing for precision@0.9 as well as expected entropy. However, due to the conservative decisions of expected entropy, we achieve higher recall levels in the out-of-sample data as well. Thus, we suppose that instantiating the algorithm with optimization objectives that correspond to expected entropy minimization leads to a less myopic learning behaviour and is not as much prone to overfitting as precision@0.9. The precision curves look quite similar for in-sample and out-of-sample data. Again, expected entropy minimization seems to be an adequate default choice in this setting.

6. Conclusion and outlook

We presented a general search-based framework for learning decision catalogues, i.e., coherent sequences of decision models that are increasing in complexity, as well as an instantiation of this framework for the case of scoring systems defined on an increasing subset of features. Moreover, to demonstrate the potential of this approach, we presented first experimental studies on two real use cases from the medical domain as well as benchmark data commonly used in machine learning. Our results suggest that a simple greedy strategy for constructing a decision catalogue of scoring systems often performs quite well or even close to optimal, although still leaving space for improvement.

Needless to say, there are various ways in which this framework can be further explored, and many directions for future work. In the following, we list a few extensions that appear to be quite sensible:

- Scoring systems check conditions in the form of binary features, which necessitates a binarization of numerical or categorical features; Ideally, this binarization is not done independently as a preprocessing step, but rather integrated with the learning of scoring systems [51–53,2].
- So far, we only considered the case of binary decisions, though an extension to decision spaces of higher cardinality is practically relevant.
- Despite the relatively strong performance of greedy search for the cases considered in this paper, more sophisticated search and optimization methods for learning decision catalogues should be developed on the basis of techniques such as Markov decision processes [9] or Monte Carlo tree search. This also includes strategies that are specifically tailored to a given loss L on catalogues. The design of such algorithms should go hand in hand with their theoretical analysis, e.g., regarding computational complexity.
- It would also be interesting to instantiate our general framework with decision models other than scoring systems.

CRediT authorship contribution statement

Stefan Heid: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation. **Jonas Hanselle:** Validation, Software, Methodology, Investigation, Data curation. **Johannes Fürnkranz:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Eyke Hüllermeier:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

We gratefully acknowledge funding by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG): TRR 318/1 2021 – 438445824.

References

[1] M. Aggarwal, A. Tehrani, E. Hüllermeier, Preference-based learning of ideal solutions in TOPSIS-like decision models, J. Multi-Criteria Decis. Anal. 22 (2014).

[2] M. Alaya, S. Bussy, S. Gaiffas, A. Guilloux, Binarsity: a penalization for one-hot encoded features in linear supervised learning, J. Mach. Learn. Res. 20 (2019) 1–34.

[3] F. Beck, J. Fürnkranz, P.V.Q. Huynh, Generalizing conjunctive and disjunctive rule learning to learning m-of-n concepts, in: Proceedings of the 23rd Conference Information Technologies - Applications and Theory (ITAT 2023), CEUR-WS.Org, Tatranské Matliare, Slovakia, 2023, pp. 8–13.

[4] S. Bösner, A. Becker, M. Hani, H. Keller, A. Sonnichsen, J. Haasenritter, K. Karatolios, J. Schäfer, E. Baum, N. Donner-Banzhoff, Accuracy of symptoms and signs for coronary heart disease assessed in primary care, Br. J. Gen. Pract. 60 (2010).

[5] K.H. Brodersen, C.S. Ong, K.E. Stephan, J.M. Buhmann, The balanced accuracy and its posterior distribution, in: 2010 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 3121–3124.

[6] G. Calinescu, C. Chekuri, M. Pal, J. Vondrak, Maximizing a monotone submodular function subject to a matroid constraint, SIAM J. Comput. 40 (2011).

[7] Y. Chevaleyre, F. Koriche, J. Zucker, Rounding methods for discrete linear classification, in: Proc. ICML, International Conference on Machine Learning, 2013, pp. 651–659.

[8] G. Choquet, Theory of capacities, Ann. Inst. Fourier 5 (1953) 131–295.

[9] M. Clertant, N. Sokolovska, Y. Chevaleyre, B. Hanczar, Interpretable cascade classifiers with abstention, in: Proc. AISTATS, 22nd Int. Conf. on Artificial Intelligence and Statistics, 2019.

[10] Y. Crama, P. Hammer, Boolean Functions: Theory, Algorithms and Applications, Cambridge University Press, 2011.

[11] J. Czerlinski, G. Gigerenzer, D. Goldstein, How good are simple heuristics?, in: Simple Heuristics That Make Us Smart, Oxford University Press, New York, 1999, pp. 97–118.

[12] J.V. Davis, J. Ha, C.J. Rossbach, H.E. Ramadan, E. Witchel, Cost-sensitive decision tree learning for forensic classification, in: J. Fürnkranz, T. Scheffer, M. Spiliopoulou (Eds.), Proc. 17th European Conference on Machine Learning (ECML), Springer, Berlin, Germany, 2006, pp. 622–629.

[13] L. De Raedt, M. Bruynooghe, Towards friendly concept-learners, in: N.S. Sridharan (Ed.), Proc. 11th International Joint Conference on Artificial Intelligence (IJCAI), Morgan Kaufmann, Detroit, MI, USA, 1989, pp. 849–858.

[14] T.S. Filho, H. Song, M. Perelló-Nieto, R. Santos-Rodríguez, M. Kull, P. Flach, Classifier calibration: how to assess and improve predicted class probabilities: a survey, Mach. Learn. 112 (2023) 3211–3260.

[15] P.C. Fishburn, Exceptional paper—lexicographic orders, utilities and decision rules: a survey, Manag. Sci. 20 (1974) 1442–1471.

[16] J. Fürnkranz, Top-down pruning in relational learning, in: A.G. Cohn (Ed.), Proc. 11th European Conference on Artificial Intelligence (ECAI-94), John Wiley & Sons, Amsterdam, the Netherlands, 1994, pp. 453–457.

[17] J. Fürnkranz, Pruning algorithms for rule learning, Mach. Learn. 27 (1997) 139–171.

[18] J. Fürnkranz, T. Kliegr, H. Paulheim, On cognitive preferences and the plausibility of rule-based models, Mach. Learn. 109 (2020) 853–898.

[19] J. Fürnkranz, E. Hüllermeier (Eds.), Preference Learning, Springer-Verlag, Berlin, Heidelberg, 2011.

[20] B. Gage, A. Waterman, W. Shannon, M. Boechler, M. Rich, M. Radford, Validation of clinical classification schemes for predicting stroke, J. Am. Med. Assoc. 285 (2001) 2864–2870.

[21] J. Gama, P. Brazdil, Cascade generalization, Mach. Learn. 41 (2000) 315–343, https://doi.org/10.1023/A:1007652114878.

[22] M. Grabisch, Modelling data by the Choquet integral, in: Information Fusion in Data Mining, Springer, 2003, pp. 135–148.

[23] M. Grabisch, C. Labreuche, Fuzzy measures and integrals in MCDA, in: Multiple Criteria Decision Analysis: State of the Art Surveys, Springer, New York, NY, 2005.

[24] M. Grabisch, C. Labreuche, A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid, Ann. Oper. Res. 175 (2010) 247–290.

[25] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.

[26] J. Hanselle, J. Fürnkranz, E. Hüllermeier, Probabilistic scoring lists for interpretable machine learning, in: Proc. DS, 23rd Int. Conference on Discovery Science, Springer, Porto, Portugal, 2023.

[27] T. Hastie, S. Rosset, R. Tibshirani, J. Zhu, The entire regularization path for the support vector machine, J. Mach. Learn. Res. 5 (2004) 1391–1415.

[28] T.J. Hastie, Generalized Additive Models, Routledge, 2017.

[29] A. Holzinger, Interactive machine learning for health informatics: when do we need the human-in-the-loop?, Brain Inform. 3 (2016) 119–131.

[30] A. Hüfner, D. Kiefl, M. Baacke, R. Zöllner, E. Loza Mencía, O. Schellein, N. Avan, S. Pemmerl, Risikostratifizierung durch implementierung und evaluation eines Covid-19-scores, Med. Klin. Intensivmed. Notfmed. 115 (2020) 132–138, https://doi.org/10.1007/s00063-020-00754-4.

[31] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: W.W. Cohen, H. Hirsh (Eds.), Proc. 11th International Conference on Machine Learning, Morgan Kaufmann, Rutgers University, New Brunswick, NJ, USA, 1994, pp. 121–129.

[32] S. Kauschke, J. Fürnkranz, Batchwise patching of classifiers, in: Proc. 32nd AAAI Conference on Artificial Intelligence (AAAI-18), 2018, pp. 3374–3381.

[33] R.L. Keeney, H. Raiffa, Decision with Multiple Objectives, Wiley, New York, 1976.

[34] J. Kleinberg, H. Lakkaraju, J. Leskovec, J. Ludwig, S. Mullainathan, Human decisions and machine predictions, Q. J. Econ. 133 (2018) 237–293.

[35] D. Krantz, R. Luce, P. Suppes, A. Tversky, Foundations of Measurement. Vol. 1: Additive and Polynomial Representations, Academic Press, 1971.

[36] C. Lindig-Leon, N. Kaur, D. Braun, From Bayes-optimal to heuristic decision-making in a two-alternative forced choice task with an information-theoretic bounded rationality model, Front. Neurosci. 16 (2022), https://doi.org/10.3389/fnins.2022.906198.

[37] V. Melnikov, E. Hüllermeier, Learning to aggregate: tackling the aggregation/disaggregation problem for OWA, in: Proc. Asian Conference on Machine Learning (ACML), in: Proc. of Machine Learning Research., 2019.

[38] T. Miller, Explanation in artificial intelligence: insights from the social sciences, Artif. Intell. 267 (2019) 1–38, https://doi.org/10.1016/j.artint.2018.07.007.

[39] P.M. Murphy, M.J. Pazzani, ID2-of-3: constructive induction of m-of-n concepts for discriminators in decision trees, in: L. Birnbaum, G. Collins (Eds.), Proc. Eighth International Workshop (ML91), Morgan Kaufmann, Northwestern University, Evanston, Illinois, USA, 1991, pp. 183–187.

[40] A. Niculescu-Mizil, R. Caruana, Predicting good probabilities with supervised learning, in: Proc. ICML, 22nd International Conference on Machine Learning, New York, USA, 2005, pp. 625–632.

[41] J. Ortega, M. Koppel, S. Argamon, Arbitrating among competing classifiers using learned referees, Knowl. Inf. Syst. 3 (2001) 470–490.

[42] P. Perny, P. Viappiani, A. Boukhatem, Incremental preference elicitation for decision making under risk with the rank-dependent utility model, in: Proc. Uncertainty in Artificial Intelligence (UAI), New York, United States, 2016.

[43] D. Pessach, G. Singer, D. Avrahamia, H.C. Ben-Gal, E. Shmueli, I. Ben-Gala, Employees recruitment: a prescriptive analytics approach via machine learning and mathematical programming, Decis. Support Syst. 134 (2020).

[44] M. Rapp, M. Kulessa, E. Loza Mencía, J. Fürnkranz, Correlation-based discovery of disease patterns for syndromic surveillance, Front. Big Data 4 (2022) 128.

[45] R.L. Rivest, Learning decision lists, Mach. Learn. 2 (1987) 229–246.

[46] A.K. Seewald, J. Fürnkranz, An evaluation of grading classifiers, in: F. Hoffmann, D.J. Hand, N. Adams, D. Fisher, G. Guimaraes (Eds.), Advances in Intelligent Data Analysis: Proc. 4th International Conference (IDA-01), Springer-Verlag, Cascais, Portugal, 2001, pp. 115–124.

[47] H.A. Simon, A behavioral model of rational choice, Q. J. Econ. 69 (1955) 99–118.

[48] H.A. Simon, Rational choice and the structure of the environment, Psychol. Rev. 63 (1956) 129–138.

[49] O. Simsek, M. Buckmann, On learning decision heuristics, in: Imperfect Decision Makers: Admitting Real-World Rationality, 2017, pp. 75–85.

[50] A. Six, B. Backus, J. Kelder, Chest pain in the emergency room: value of the heart score, Neth. Heart J. 16 (2008) 191–196.

[51] O. Sobrie, V. Mousseau, M. Pirlot, Learning a majority rule model from large sets of assignment examples, in: Proc. 3rd Int. Conference on Algorithmic Decision Theory (ADT), Bruxelles, Belgium, 2013, pp. 336–350.

[52] O. Sobrie, V. Mousseau, M. Pirlot, Learning the parameters of a non compensatory sorting model, in: Proc. 4th Int. Conference on Algorithmic Decision Theory (ADT), Lexington, KY, USA, 2015, pp. 153–170.

[53] N. Sokolovska, Y. Chevaleyre, J. Zucker, A provable algorithm for learning interpretable scoring systems, in: Proc. AISTATS, 21st Int. Conf. on Artificial Intelligence and Statistics, 2018.

[54] V. Subramanian, E. Mascha, M. Kattan, Developing a clinical prediction score: comparing prediction accuracy of integer scores to statistical regression models, Anesth. Analg. 132 (2021) 1603–1613.

[55] V. Torra, Y. Narukawa, Modeling Decisions: Information Fusion and Aggregation Operators, Springer, 2007.

[56] G. Towell, J. Shavlik, Extracting refined rules from knowledge-based neural networks, Mach. Learn. 13 (1993) 71–101.

[57] P.D. Turney, Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm, J. Artif. Intell. Res. 2 (1995) 369–409.

[58] P.D. Turney, Types of cost in inductive concept learning, in: Proc. ICML Workshop on Cost-Sensitive Learning, Stanford University, California, 2000, pp. 15–21, http://arxiv.org/abs/cs/0212034.

[59] B. Ustun, C. Rudin, Supersparse linear integer models for optimized medical scoring systems, Mach. Learn. 102 (2016) 349–391.

[60] B. Ustun, C. Rudin, Optimized risk scores, in: Proc. 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 1125–1134.

[61] B. Ustun, C. Rudin, Learning optimized risk scores, J. Mach. Learn. Res. 20 (2019) 1–75.

[62] V. Vapnik, Statistical Learning Theory, John Wiley & Sons, 1998.

[63] C. Wang, B. Han, B. Patel, C. Rudin, In pursuit of interpretable, fair and accurate machine learning for criminal recidivism prediction, J. Quant. Criminol. 39 (2023) 519–581.

[64] J. Wnek, R.S. Michalski, Discovering representation space transformations for learning concept descriptions combining DNF and M-of-N rules, in: Working Notes of the ML-COLT'94 Workshop on Constructive Induction and Change of Representation, New Brunswick, NJ, 1994, https://hdl.handle.net/1920/1809.

[65] A. Zagorecki, D. Johnson, J. Ristvej, Data mining and machine learning in the context of disaster and crisis management, Int. J. Emerg. Manag. 9 (2013).

[66] Y. Zhao, D. Zeng, A. Rush, M. Kosorok, Estimating individualized treatment rules using outcome weighted learning, J. Am. Stat. Assoc. 107 (2012) 1106–1118, https://doi.org/10.1080/01621459.2012.695674.