Machine Learning (2025) 114:55
https://doi.org/10.1007/s10994-024-06705-w

Probabilistic scoring lists for interpretable machine learning

Jonas Hanselle[1,2] · **Stefan Heid**[1,2] · **Johannes Fürnkranz**[3] · **Eyke Hüllermeier**[1,2]

Received: 27 March 2024 / Revised: 23 October 2024 / Accepted: 12 November 2024 /
Published online: 6 February 2025

Abstract
A scoring system is a simple decision model that checks a set of features, adds a certain number of points to a total score for each feature that is satisfied, and finally makes a decision by comparing the total score to a threshold. Scoring systems have a long history of active use in safety-critical domains such as healthcare and justice, where they provide guidance for making objective and accurate decisions. Given their genuine interpretability, the idea of learning scoring systems from data is obviously appealing from the perspective of explainable AI. In this paper, we propose a practically motivated extension of scoring systems called probabilistic scoring lists (PSL), as well as a method for learning PSLs from data. Instead of making a deterministic decision, a PSL represents uncertainty in the form of probability distributions, or, more generally, probability intervals. Moreover, in the spirit of decision lists, a PSL evaluates features one by one and stops as soon as a decision can be made with enough confidence. To evaluate our approach, we conduct case studies in the medical domain and on standard benchmark data.

Keywords Machine learning · Decision support · Scoring systems · Uncertainty representation · Calibration

Editors: Albert Bifet, Rita Ribeiro, Ana Carolina Lorena.

Jonas Hanselle and Stefan Heid have contributed equally to this work.

✉ Jonas Hanselle
jonas.hanselle@ifi.lmu.de

✉ Stefan Heid
stefan.heid@ifi.lmu.de

✉ Johannes Fürnkranz
juffi@faw.jku.at

✉ Eyke Hüllermeier
eyke@lmu.de

[1] Institute of Informatics, LMU Munich, Akademiestr. 7, 80799 Munich, Germany

[2] Munich Center for Machine Learning, Munich, Germany

[3] Institute for Application-Oriented Knowledge Processing, Johannes Kepler Universität Linz, Altenberger Straße 69, 4040 Linz, Austria

1 Introduction

Predictive models generated by modern machine learning algorithms, such as deep neural networks, tend to be complex and difficult to comprehend, and may not be appropriate in applications where a certain degree of transparency of a model and explainability of decisions are desirable. Besides, depending on the situation and application context, time and computational 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 even by the human herself without any technical device.

So-called *scoring systems* provide a simple, genuinely interpretable model class as an alternative. In a nutshell, a scoring system is a decision model that checks a set of features, adds (or subtracts) a certain number of points to a total score for each feature that is satisfied, and finally makes a decision by comparing the total score to a threshold. Scoring systems have a long history of active use in safety-critical domains such as healthcare (Six et al., 2008) and justice (Wang et al., 2023), where they provide guidance for making objective and accurate decisions. Given their genuine interpretability, scoring systems are appealing from the perspective of explainable AI, which is why the idea of learning such systems from data has recently attracted attention in machine learning.

Building on our previous work (Hanselle et al., 2023), this paper contributes to existing methodology for scoring systems as follows:

- We propose a practically motivated extension of scoring systems called *probabilistic scoring lists* (PSL), as well as a method for learning PSLs from data.
- To increase uncertainty-awareness, a PSL produces predictions in the form of probability distributions (instead of making deterministic decisions).
- Moreover, to increase cost-efficiency, a PSL is conceptualized as a *decision list*: It evaluates features one by one and stops as soon as a decision can be made with enough confidence.

Moreover, we extend (Hanselle et al., 2023) in various ways:

- We make the PSL method amenable to continuous variables by developing *discretization techniques* for turning numerical attributes into binary features, either in a preprocessing step or progressively in the course of the PSL procedure.
- In order to calibrate probability estimates, we consider beta calibration in addition to isotonic regression.
- We propose a method for quantifying *epistemic uncertainty*, i.e., uncertainty about the probability estimates in a PSL.
- Going beyond simple decisions, we propose a variant of PSL that is appropriate for the task of *ranking*, i.e., sorting a set of instances from most likely positive to most likely negative, leveraging training data in the form of relative comparisons.
- We also expand the empirical evaluation by adding additional datasets and experimental studies.

Following a brief overview of related work in the next section, we introduce PSLs in Sect. 3 and address the problem of learning such models from data in Sect. 4. To evaluate our approach, we conduct as series of experimental studies in Sect. 5f, prior to concluding the paper with an outlook on extensions and future work in Sect. 6.

2 Related work

The use of machine learning models for decision support has recently gained considerable attention. In many safety-critical application domains such as healthcare or justice, the interpretability of the predictions made by such models is an essential requirement, because a decision maker who is unable to understand a prediction is unlikely to trust and base their decision on the model.

Methods such as LIME (Ribeiro et al., 2016) and SHAP (Lundberg & Lee, 2017) can be used to create explanations for any model, including powerful black-box models like deep neural networks. An alternative to using such post-hoc explainers is the use of inherently interpretable models, which is of particular interest in high stakes decision-making (Rudin, 2019). Models such as linear regression or decision trees are considered inherently interpretable. By inspecting their simple structure, one can see how different features influence the predictions across instances and gain a global understanding of the model. This interpretability comes at the cost of reduced flexibility, potentially leading to a loss in predictive accuracy. However, if genuine interpretability is a must, this loss has to be accepted.

One of the practically most relevant model classes of this kind are scoring systems, which are the de facto standard tool in clinical decision support but also used in other domains. Scoring systems assign integer weights to all features, which are added up for all positively evaluated features of a particular patient. Typically, this total score is a degree of severity which is compared with a threshold to form a decision. Over the past decades, a plethora of scoring systems has been developed which are in active use in healthcare. Prominent examples include the SAPS (Metnitz et al., 2005; Moreno et al., 2005; Le Gall et al., 1993, 1984) and APACHE (Knaus et al., 1981, 1985, 1991) scoring systems for assessing mortality in intensive care, the SOFA score (Vincent et al., 1996) for sepsis-related organ failure assessment or the CHA_2DS_2-VASc score (Lip et al., 2010) for predicting stroke risk of patients with atrial fibrillation.

The scoring systems in healthcare commonly assign small integer scores to the features. This simple structure makes them applicable in practice, imposing low cognitive load for the practitioners and making predictive outcomes easy to understand and communicate.

While scoring systems are typically handcrafted by domain experts, there has been increasing interest in algorithmically deriving scoring systems from data using machine learning methods. 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 (Ustun & Rudin, 2016, 2017, 2019). 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 integer linear program solvers.

In several applied fields, one also finds methods of a more heuristic nature. Typically, 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 (Subramanian et al., 2021). From a theoretical

perspective, certain guarantees for the rounded solutions can nevertheless be given (Chevaleyre et al., 2013).

A related research direction is the learning of simple decision heuristics that are considered plausible from the perspective of cognitive psychology. Again, however, this is a relatively unexplored field, in which only a few publications can be found so far—Simsek and Buckmann (2016) collect and empirically compare some of these heuristics.

Decision lists have been primarily used in inductive rule learning (Fürnkranz et al., 2012), where each term consists of a conjunction of conditions, which are sufficient to make a prediction in case the conditions are satisfied, or else continue with the next rule. They have been shown to generalize both, k-term CNF and DNF expressions, as well as decision trees with a fixed depth k (Rivest, 1987). Practically, they represent a simple way for tie-breaking in situations where multiple rules cover the same example: in that case, the first rule in the list is given priority. They can be easily learned, as their structure mirrors the commonly covering or separate-and-conquer strategy (Fürnkranz, 1999), which learns one rule at a time, typically by appending rules to the list, assuming that most important rules are tried first, but prepending has also been tried (Webb, 1994). While rules are typically used for classification, they may also be viewed as simple probability estimators, using the class distribution among the covered examples as the basis for various estimation techniques (Sulzmann & Fürnkranz, 2009). However, these are known to be overly optimistic, because the way the conditions are selected results in a bias towards the positive examples during learning (Možina et al., 2019). Also, in decision lists in rule learning, the probability estimates are derived from the last rule in isolation, practically ignoring all previous rules, whereas, as will be seen later, the probability distributions in PSLs are successively refined.

3 Probabilistic scoring lists

Consider a scenario where decisions need to be made in different contexts, which are characterized in terms of a set of variables or features $\mathcal{F} = \{f_1, \ldots, f_K\}$. A concrete situation is specified by a vector $\boldsymbol{x} = (x_1, \ldots, x_K)$, where x_i is the value observed for the feature f_i, and 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 is normally finite, typically comprising a small to moderate number of alternatives to choose from.

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 $\boldsymbol{x}$. Note that such models can be represented in different ways. For the reasons already explained, we shall focus on scoring systems in this paper. In a nutshell, scoring systems 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 (Hastie, 1990) defined over a set of features.

Definition 1 (*Scoring system*). A *scoring system* over a set of (binary) candidate features $\mathcal{F}$ and score set $\mathcal{S} \subset \mathbb{Z}$ is a triple $h = \langle F, S, t \rangle$, where $F = \{f_1, \ldots, f_K\} \subset \mathcal{F}$ is a subset of the candidate features, $S = (s_1, \ldots, s_K) \in \mathcal{S}^K$ are scores assigned to the corresponding features,

and $t \in \mathbb{Z}$ is a decision threshold. For a given decision context $\boldsymbol{x} = (x_1, \ldots, x_K) \in \{0,1\}^K$, i.e., the projection of an instance to the feature set F, the decision prescribed by h is given by

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

where $[\![\cdot]\!]$ is the indicator function.[1]

Note that, according to this definition, scoring systems are binary classifiers ($\mathcal{Y} = \{0,1\}$). They can also be seen as linear classifiers with weights s_i restricted to be elements of the score set $\mathcal{S}$. When features are strictly binary, scoring systems essentially resemble Boolean threshold functions and share many of their properties (Crama & Hammer, 2011). Thus, scoring systems are clearly limited in terms of expressivity. In particular, note that they are not capable of modeling feature interactions unless these are manually added as additional (compound) features (for example, the synergy/redundancy between two features x_i and x_j can be modeled by adding the logical conjunction $x_{i,j} = x_i \wedge x_j$ as an additional feature and assigning it a positive/negative score).

In the following, we generalize such scoring systems in two ways: from deterministic to probabilistic, and from a single decision model to a decision list.

As for the first extension, the idea is to return a probability distribution over $\mathcal{Y}$ instead of a binary decision (1), i.e., to assign a probability $p(y)$ to each decision $y \in \mathcal{Y}$. The latter can be interpreted as the probability that y is the best or correct decision, which (implicitly) presupposes the existence of a kind of ground truth. Without loss of generality, we can assume that the ground truth distinguishes between a class of positive cases and a class of negative cases, and that the decision is a prediction of the correct class. Therefore, we shall use the terms "decision" and "class" interchangeably.

We contextualize the distribution p, not directly with $\boldsymbol{x}$, but rather with the total score $T(\boldsymbol{x})$ assigned to $\boldsymbol{x}$. In other words, we consider conditional probabilities $p(\cdot \mid T(\boldsymbol{x}))$ on $\mathcal{Y}$. This appears meaningful and is in line with the assumption that the total score is indicative of the class — in fact, standard scoring systems can be seen as a special case, returning probability 1 for the positive class when exceeding the threshold and probability 0 otherwise.

Definition 2 (*Probabilistic scoring system, PSS*). A *probabilistic scoring system* (PSS) over candidate features $\mathcal{F}$ and score set $\mathcal{S} \subset \mathbb{Z}$ is a triple $h = \langle F, S, q \rangle$, where $F = \{f_1, \ldots, f_K\} \subset \mathcal{F}$, $S = (s_1, \ldots, s_K) \in \mathcal{S}^K$, and q is a mapping $\Sigma \longrightarrow [0,1]$, where

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

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 $1 - q(T) = p(y = 0 \mid T)$ the probability for the negative class).

[1] $[\![P]\!] = 1$ if predicate P is true (positive decision) and $[\![P]\!] = 0$ if P is false (negative decision).

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

$$\forall\, T, T' \in \Sigma : \; (T < T') \Rightarrow q(T) \leq q(T')\,. \tag{2}$$

This property is again in line with standard scoring systems and appears to be important from an interpretability perspective: A violation of (2) would be considered as an inconsistency and compromise the acceptance of the decision model. Therefore, in the remainder of the paper, we consider only monotonic probabilistic scoring systems.

Our second extension combines probabilistic scoring systems with the notion of decision lists. The underlying idea is as follows: Instead of determining all K feature values x_i right away, these values are determined successively, one after the other, in a predefined order. Each time a new feature is added, the total score T is updated, and the probability $q(T)$ of the positive class is determined. Depending on the latter, the process is then continued or stopped: If the probability is sufficiently high or sufficiently low, the process is stopped, because a decision can be made with enough confidence; otherwise, the process is continued by adding the next feature. In summary, a PSL consists of i) an ordering of features, ii) the associated scores and iii) a probability function mapping total scores to probability estimates at each stage.

Table 1 depicts a PSL with four features $F = \{f_1, f_2, f_3, f_4\}$. As can be seen from the assigned scores, all features except f_1 are indicative of the positive class, i.e., the presence of f_2, f_3 or f_4 increases the probability of the positive class, whereas the presence of f_1 decreases the probability.

The decision process starts with an empty feature set and a prior probability of 0.3 for the positive class. After seeing the first feature f_3 with a weight $s_3 = +1$, the possible scores are $T = 0$ if the feature does not hold (the value of the feature is $x_3 = 0$), or $T = +1$, if $x_3 = 1$. In the former case, the probability for the positive decision decreases to 0.2, in the latter case it increases to 0.4. The next feature is f_1 with a weight of $s_1 = -2$, resulting in a total of four possible scores, ranging from $T = -2$ (if $x_3 = 0$ and $x_1 = 1$) to $T = +1$ (if $x_3 = 1$ and $x_1 = 0$). Note that the absence of f_1, in this case, may increase the probability of a positive score to 0.6. Adding the remaining features continues this process, until we get a diverse set of seven probability estimates (five of which are different) corresponding to the seven different score values we can obtain for the

Table 1 Example of a PSL with feature set $\mathcal{F} = \{f_1, f_2, f_3, f_4\}$ and score set $\mathcal{S} = \{0, \pm 1, \pm 2\}$

Feature added	Feature list	Score	Total score						
			−2	−1	0	+1	+2	+3	+4
–	–	–	–	–	0.3	–	–	–	–
f_3	$\{f_3\}$	+1	–	–	0.2	0.4	–	–	–
f_1	$\{f_1, f_3\}$	−2	0.1	0.2	0.5	0.6	–	–	–
f_2	$\{f_1, f_2, f_3\}$	+1	0.1	0.2	0.6	0.7	0.9	–	–
f_4	$\{f_1, f_2, f_3, f_4\}$	+2	0.1	0.1	0.2	0.6	0.7	0.9	0.9

The first two columns show the newly selected feature and the complete features list at that stage, while the third column shows the score assigned to the newly added feature. The remaining columns show the probabilities assigned to the total scores $q(T) = p(y = 1 \mid T)$ for each stage

$2^4 = 16$ possible instances. For example, the instance $\boldsymbol{x} = (1, 1, 1, 1)$ would be assigned a probability of $q(2) = 0.7$, based on its total score of $T(\boldsymbol{x}) = +2$.

Note the monotonicity (2) in the scores in each row (higher score values result in higher probabilities for the positive decision). Also note that if the final maximal probability of 0.9 is considered to be sufficiently high for making a positive decision, the process could already have been stopped after seeing the first three features for any instance $\boldsymbol{x} = (0, 1, 1, *)$, irrespective of its value x_4 for the fourth feature f_4.

Formally, we can define a probabilistic scoring list as follows:

Definition 3 (*Probabilistic scoring list, PSL*). A *probabilistic scoring list* over candidate features $\mathcal{F}$ and score set $\mathcal{S} \subset \mathbb{Z}$ is a triple $h = \langle F, S, p \rangle$, where $F = (f_1, \dots, f_K)$ is a list of (distinctive) features from $\mathcal{F}$, $S = (s_1, \dots, s_K) \in \mathcal{S}^K$, and q is a mapping

$$q : \bigcup_{k=0}^{K} (k, \Sigma_k) \longrightarrow [0, 1] \tag{3}$$

such that

$$\forall\, k \in \{0, 1, \dots, K\}, T, T' \in \Sigma_k : (T < T') \Rightarrow q(k, T) \leq q(k, T') . \tag{4}$$

Here, Σ_k is the set of possible values for the total score at stage k, i.e.,

$$\Sigma_k = \left\{ T = \sum_{i=1}^{k} s_i x_i \;\middle|\; s_1, \dots, s_k \in \mathcal{S},\, x_1, \dots, x_k \in \{0, 1\} \right\} . \tag{5}$$

A value $q(k, T)$ is interpreted as the probability of the positive decision if the total score at stage k is given by T.

Note that $k = 0$ is included in (3). This case corresponds to the empty list, where no feature has been determined at all. The corresponding value $q(0, 0)$ can be considered as a default probability of the positive class. In general, the score set $\mathcal{S}$ can be any subset of $\mathbb{Z}$. However, to guarantee good interpretability, it is common to chose small sets with small values. Moreover, the score of 0 is usually omitted, because features with this score are simply disabled and hence effectively removed from the model.

4 Learning probabilistic scoring lists

While standard 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 (Ustun & Rudin, 2016). 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.

Instead of learning standard scoring systems, we are interested in the task of learning probabilistic scoring lists, i.e., in constructing a PSL h from training data

$$\mathcal{D} = \left\{ (\boldsymbol{x}_i, y_i) \right\}_{i=1}^{N} \subset \mathcal{X} \times \mathcal{Y} . \tag{6}$$

This essentially means determining the following components:

- the subset of features to be included and the order of these features;
- the score assigned to each individual feature;
- the probabilities for the resulting combinations of stage and total score.

A first question in this regard concerns the quality of a model h: What do we actually mean by a "good" probabilistic scoring list? Intuitively, a good PSL allows for making decisions that are quick and confident at the same time. Thus, we would like to optimize two criteria simultaneously, namely, to minimize the number of features that need to be determined before a decision is made, and to maximize the confidence of the resulting decision. This compromise could be formalized in different ways, but regardless of how an overall performance measure is defined, the problem of optimizing that measure over the space of possible PSLs will be computationally hard (Chevaleyre et al., 2013).

4.1 A greedy learning algorithm

As a first attempt, we therefore propose a heuristic learning procedure that is somewhat inspired by decision tree learning. Starting with the empty list, the next feature/score combination (x_k, s_k) is added greedily so as to improve performance the most,[2] and this is continued until no further improvement can be obtained.

To this end, each (remaining) feature/score combination is tried and evaluated as follows: Let Σ_k be the set of total scores T in stage k as defined in (5), and $Q = \{(N_T, \hat{q}_T) \mid T \in \Sigma_k\}$ the set of probability estimates $\hat{q}_T = \hat{q}(k, T)$ for total scores $T \in \Sigma_k$, together with the number N_T of training examples being assigned this score. The feature/score combination is then evaluated in terms of the *expected entropy*:

$$E(Q) = \sum_{(N_T, \hat{q}_T) \in Q} \frac{N_T}{N} H\big(\hat{q}_T\big)\,, \tag{7}$$

where H is the Shannon entropy

$$H(q) = -q \cdot \log(q) - (1-q)\log(1-q)\,.$$

Thus, according to (7), the entropy of each distribution $\hat{q}_T$ is weighted by the probability that this distribution occurs.

4.2 Probability estimation

As for the estimation of the probabilities $q(k, T)$, the most obvious idea would be a standard frequentist approach, i.e., to estimate them in terms of relative frequencies P_T/N_T, where N_T is again the number of training examples with total score T, and P_T is the number of examples with total score T and class $y = 1$ (in stage k). However, as these estimates are obtained independently for each score T, they may violate the

[2] As the importance of a feature x_k, and hence the score s_k, can only be decided relative to other features, the choice of the score for the first feature is ambiguous; assuming this feature to be important, we have given it the largest score possible.

monotonicity condition (2). A better idea, therefore, is to estimate them jointly using a probability calibration method (de Menezes e Silva Filho et al., 2023). To this end, the original data $\mathcal{D}$, or a subset $\mathcal{D}_{cal}$ specifically reserved for calibration (and not used for training), is first mapped to the data

$$\mathcal{C} := \left\{ (T(\boldsymbol{x}), y) \mid (\boldsymbol{x}, y) \in \mathcal{D}_{cal} \right\} \subset \Sigma_k \times \mathcal{Y}, \tag{8}$$

to which any calibration method can then be applied. One of the most popular techniques, *isotonic regression* (Niculescu-Mizil & Caruana, 2005), amounts to finding values $\hat{q}(k, T)$ as solutions to the following constrained optimization problem:

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

Another common calibration method is the use of a logistic regression, which, however, assumes class-wise normally distributed scores. Kull et al. (2017) show that this assumption does not hold for common classifiers and propose the more flexible *beta calibration*, which defines a mapping $[0, 1] \longrightarrow [0, 1]$ of the following form:

$$\hat{q}(k, \tau) = \frac{1}{1 + m^a (1-m)^{-b}\, \tau^{-a}\, (1-\tau)^b}\,, \tag{9}$$

where $a, b \geq 0$ and $m \in [0, 1]$ are parameters. These parameters are fitted by minimizing log-loss on the calibration data (8). In our case, the τ-values are (linear) transformations of the total scores T from Σ_k to [0, 1]. Both approaches, isotonic regression and beta calibration, are illustrated in Fig. 1.

Note that, from a probability estimation point of view, the estimation of one distribution per total score $T \in \Sigma_k$ is a meaningful compromise between a global probability estimate (not taking any context features into account) and a *per-instance* estimation, i.e., the prediction of an individual distribution $p(\cdot \mid \boldsymbol{x})$ tailored to any specific instance $\boldsymbol{x}$. Obviously, the former is not informative enough, while the latter is very difficult to obtain, due to a lack of statistical information related to a single point (Foygel Barber et al., 2021). According to our assumption, all instances with the same total score T share the same probability. Therefore, those instances in the training data with the same score form a homogeneous statistical subgroup

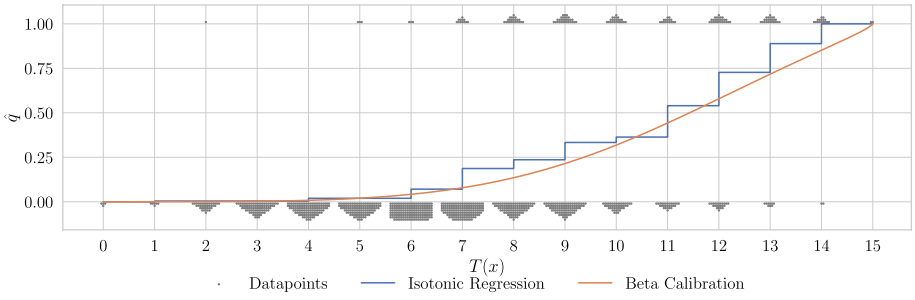

Fig. 1 Example of calibration with isotonic regression and beta calibration, using the medical dataset introduced in Sect. 5. The values on the x-axis correspond to the total scores. As class labels are either 0 or 1, the data points $\mathcal{C}$ are plotted with jittering for better visualization

$$\mathcal{D}_T := \left\{ (\boldsymbol{x}_i, y_i) \in \mathcal{D} \mid T(\boldsymbol{x}) = T \right\},$$

to which statistical estimation methods can be applied. While this is in line with other local prediction methods, such as probability estimation trees (PETs) (Provost & Domingos, 2003), the distinguishing feature here is the way in which the instance space $\mathcal{X}$ is partitioned. For example, compared to PETs, PSLs appear to have a more rigid structure, because the succession of tests (features) is fixed and can not vary depending on the value of the features (like in trees). Moreover, the size of the partition, $|\Sigma_k|$, will normally be smaller than the up to 2^k different leaf nodes in a tree (leaves with same scores are merged). Overall, we consider a single sequence of k feature evaluations more interpretable than navigating through up to 2^k different paths through a PET.

4.3 Feature binarization

The features $\mathcal{F}$ in a PSL are assumed to be binary. However, in many practical applications, instances are also characterized by continuous attributes. To make scoring systems amenable to these kinds of data, continuous attributes need to be turned into binary features through *binarization*. This is done by selecting a threshold t_j for each feature $f_j \in \mathcal{F}$ and assigning a value of 0 for values below the threshold and 1 otherwise. Such a discretization from numerical values into binary features is unavoidably accompanied by a loss of information. To lose as little information as possible about the connection between the feature value and the class label, we employ the *entropy minimization heuristic* (Fayyad & Irani, 1993). We can make use of this heuristic either for preparing the data in a preprocessing step (*preprocessing*) or as a subroutine that is used during the greedy construction of the PSL (*in-search*).

When feature binarization is carried out as a preprocessing step, all features are treated independently. For each (numerical) feature f_j, we consider all possible bisections of the dataset when thresholding with t_j, where t_j is any mid-point between two consecutive values assumed by that feature in the training data:

$$Y_{t_j}^{\leq} = \{ y_i \mid (\boldsymbol{x}_i, y_i) \in \mathcal{D}, \boldsymbol{x}_{i,j} \leq t_j \}$$
$$Y_{t_j}^{>} = \{ y_i \mid (\boldsymbol{x}_i, y_i) \in \mathcal{D}, \boldsymbol{x}_{i,j} > t_j \}$$

All possible bisections are enumerated, and we finally select the threshold that leads to the minimal expected entropy by considering the cardinality and relative frequency of the positive class in each bisection:

$$t_j^* = \arg\min_{t_j} E\left(\left\{ \left(|Y_{t_j}^{>}|, \sum_{y \in Y_{t_j}^{>}} \frac{y}{|Y_{t_j}^{>}|} \right), \left(|Y_{t_j}^{\leq}|, \sum_{y \in Y_{t_j}^{\leq}} \frac{y}{|Y_{t_j}^{\leq}|} \right) \right\} \right),$$

with E defined according to (7).

Instead of finding the optimal threshold t_j^* by ordering all $\boldsymbol{x}_{i,j}$ and exhaustively testing every candidate threshold between two consecutive values in a brute-force manner, one can also make use of a heuristic that is computationally less costly. By assuming quasi-convexity of the expected entropy with respect to the threshold point, hierarchical binary search can be used instead. In practice, this assumption does not necessarily hold, but is computationally less expensive, as only a logarithmic number of candidate thresholds need

to be considered. In the empirical evaluation in Sect. 5, both the heuristic and brute-force threshold selection are evaluated.

As an alternative to preprocessing, feature binarization can also be carried out *in-search*, i.e., during the proposed greedy learning algorithm. Here, the features are not treated independently, but rather sequentially. Recall that when constructing a PSL, the greedy learning algorithm chooses a feature and an associated score in each stage k. Alongside these choices, it now also chooses the binarization threshold for continuous features. Thus, at construction in stage k, the features selected at stages $1, \ldots, k-1$ have already been binarized. Again, binarization is done by enumerating all candidate thresholds t_k and selecting the one that minimizes expected entropy of the resulting partitioning:

$$t_k^* = \arg\min_{t_k} E\left(\{(N_T, \hat{q}_{t_k}(k,T)) \mid T \in \Sigma_k\}\right),$$

with $\hat{q}_{t_k}(k,T)$ the probability estimate for total score T at stage k when the k^{th} feature is binarized by threshold t_k.

4.4 Beyond probabilities: capturing epistemic uncertainty

Going beyond standard probabilistic prediction, various methods have recently been proposed in machine learning that seek to distinguish between so-called aleatoric and epistemic uncertainty (Senge et al., 2014; Hüllermeier & Waegeman, 2021). Broadly speaking, aleatoric uncertainty refers to inherent randomness and stochasticity of the underlying data-generating process. This type of uncertainty is relevant in our case, because the dependence between total score T and decision/class assignment y is presumably non-deterministic. Aleatoric uncertainty is properly captured in terms of probabilities, i.e., by the approach introduced above.

Epistemic uncertainty, on the other side, refers to uncertainty caused by a lack of knowledge, e.g., the learner's uncertainty about the true distribution $p = p(\cdot \mid T)$. In a machine learning context, this uncertainty could be caused by insufficient or low-quality training data. Obviously, it is relevant in our case, too: Proceeding further in the decision list, the training data will become more and more fragmented, because the number of possible values for the total score increases. Consequently, the estimation $\hat{q}_T$ of a conditional probability $p(y = 1 \mid T)$ will be based on fewer and fewer data points, so that the epistemic uncertainty increases (even if the joint estimation of these probabilities for all scores T alleviates this effect to some extent).

Representing this uncertainty is arguably important from a decision-making point of view. For example, proceeding in the list and adding another variable may imply that the (predicted) distribution becomes better in the sense of having lower entropy, but at the same time, the prediction itself may become more uncertain. In that case, it is not clear whether the current stage should be preferred or maybe the next one — the answer to this question will depend on the attitude of the decision maker (toward risk), and probably also on the application.

A natural approach to capturing epistemic uncertainty is to replace point estimates $\hat{q}_T$ of $p_T = p(y = 1 \mid T)$ by interval estimates — epistemic uncertainty is then reflected by the interval widths. Formally, we can view the true probability p_T as the (unknown) parameter of a Bernoulli distribution (binomial proportion). There is vast statistical literature on estimating confidence intervals for binomial proportions, and various constructions of such

intervals have been proposed. For example, the Clopper–Pearson interval (Clopper & Pearson, 1934) with confidence level $1-\alpha$ can be expressed as $l_T \leq p_T \leq u_T$ with

$$l_T = \left(1 + \frac{N_T + 1}{P_T \, F[\alpha/2; 2P_T, 2(N_T+1)]}\right)^{-1},$$
$$u_T = \left(1 + \frac{N_T}{(P_T+1) \, F[1-\alpha/2; 2(P_T+1), 2N_T]}\right)^{-1},$$

where N_T is the number of negative examples, P_T the number of positive examples, and $F[c;d,d']$ is the c-quantile from an F-distribution with d and d' degrees of freedom.

On the basis of individual confidence intervals of that kind, a complete *confidence band* $\left\{[l_T^*, u_T^*] \mid T \in \Sigma_T\right\}$, i.e., a sequence of intervals for all total score values, can be constructed as follows:

- First, one has to guarantee a simultaneous confidence of $1-\alpha$. The simplest way to do so is to apply Bonferroni correction, i.e., to compute individual intervals $[l_T, u_T]$ for confidence level $1-\alpha/|\Sigma_k|$.
- Second, monotonicity constraints can be incorporated by correcting the intervals as follows:

$$l_T^* \leftarrow \max\{l_V \mid V \in \Sigma_k,\, V \leq T\} \tag{10}$$

$$u_T^* \leftarrow \min\{u_V \mid V \in \Sigma_k,\, V \geq T\} \tag{11}$$

Note that, although the correction (10–11) may lead to inconsistencies (empty intervals), it still guarantees the $1-\alpha$ confidence (under the assumption of monotonicity): With probability (at least) $1-\alpha$, we have $p_T \in [l_T, u_T]$ simultaneously for all $T \in \Sigma_k$, which in turn implies $p_T \geq p_V \geq l_V$ for all $V \leq T$ and $p_T \leq p_V \leq u_V$ for all $V \geq T$.

In order to assure that the confidence band covers the calibrated (point) estimates $\hat{q}(k,T)$, one may consider another correction step (which obviously maintains monotonicity):

$$l_T^\star \leftarrow \min\{\, l_T^*,\, \hat{q}(k,T)\,\} \tag{12}$$

$$u_T^\star \leftarrow \max\{\, u_T^*,\, \hat{q}(k,T)\,\} \tag{13}$$

4.5 Ranking

In addition to the standard probabilistic binary classification task, we also consider PSL for the task of ranking. Instead of a model that assigns each instance to the positive or the negative class, we now seek a model that is able to prioritize instances from most likely positive to most likely negative—in the literature, this problem is known as the bipartite ranking problem (Kotlowski et al., 2011). Again, models of that kind are highly relevant and have many practical applications.

When having access to standard training data, i.e., a set of instances labelled positive or negative, this can essentially be accomplished using the PSL as is. More specifically, a PSL can be trained in exactly the same way as in the case of binary classification. Then, given a set $X \subset \mathcal{X}$ of instances to be ranked, the PSL can be used to predict a probability $\hat{q}(\boldsymbol{x})$ for

each $\boldsymbol{x} \in X$, and the ranking $\hat{\boldsymbol{\pi}}$ of X is obtained by sorting all $\boldsymbol{x} \in X$ in decreasing order of their (predicted) probabilities. Indeed, it can be shown that most common loss functions for bipartite ranking, comparing a predicted ranking $\hat{\boldsymbol{\pi}}$ with a binary ground-truth, are minimized (in expectation) by sorting instances in decreasing order of their probability of being positive (Kotlowski et al., 2011). An important example of such a loss is the well-known AUC measure[3] (Fawcett, 2006).

However, in the realm of ranking, training data is often given in the form of relative comparisons $\boldsymbol{x} \succ \boldsymbol{x'}$ between instances $\boldsymbol{x}, \boldsymbol{x'} \in \mathcal{X}$, suggesting that $\boldsymbol{x}$ is preferred to (should be ranked higher than) $\boldsymbol{x}'$. Then, in contrast to the first scenario, the PSL algorithm cannot be applied in a straightforward way. In particular, the expected entropy (7) cannot be computed as an impurity measure and selection criterion in the greedy learning algorithm. Instead, one has to refer to a ranking loss.

Given a set of pairwise comparisons $\mathcal{D} := \{\boldsymbol{x}_i \succ \boldsymbol{x}'_i\}_{i=1}^N$, the *pairwise soft rank loss* is defined as follows:

$$\mathrm{SRL}(\mathcal{D}) := \frac{1}{N} \sum_{i=1}^{N} \ell(\boldsymbol{x}_i, \boldsymbol{x}'_i) \tag{14}$$

with

$$\ell(\boldsymbol{x}, \boldsymbol{x'}) = \begin{cases} 0 & \text{if } \hat{q}(\boldsymbol{x}) > \hat{q}(\boldsymbol{x'}) \\ 0.5 & \text{if } \hat{q}(\boldsymbol{x}) = \hat{q}(\boldsymbol{x'}) \\ 1 & \text{if } \hat{q}(\boldsymbol{x}) < \hat{q}(\boldsymbol{x'}) \end{cases} .$$

This loss, which is related to the Kendall correlation coefficient, imposes a penalty of 0 for pairs that are correctly ordered, 0.5 for ties, and 1 for incorrect orderings. It can be computed in each stage k of the PSL algorithm—the probabilities $\hat{q}(\boldsymbol{x})$ are then given by $\hat{q}\big(k, T(\boldsymbol{x})\big)$. The PSL algorithm itself is then modified in the sense that, in every stage k, it finds the feature/score combination that yields the smallest SRL instead of the smallest expected entropy.

PSL has an interesting and intuitive interpretation in the context of ranking: Starting with the entire set X as a single tie group, it successively refines a ranking by splitting such groups into smaller subgroups and sorting these subgroups. In the first stage, all instances are assigned the same probability, i.e., there are only ties, and we start with an AUC of 0.5. As we progress throughout the stages, the set of total scores becomes larger, and more and more ties are being resolved. The process can then be stopped as soon as a sufficient resolution has been reached. This is particularly useful in scenarios in which one is not interested in retrieving the complete ordering of alternatives, but rather in eliciting the top (or bottom) m alternatives.

[3] Actually, AUC in an accuracy measure, so the loss would be obtained by 1−AUC.

5 Empirical evaluation

In this section, we present an experimental evaluation of our proposed method. Unless stated differently, the figures show the models' mean performance and a 95% confidence interval of the mean, aggregated over 100 Monte Carlo cross-validation (MCCV) splits with 2/3 used for training and 1/3 used in testing. Additionally, the score set $\{\pm 1, \pm 2, \pm 3\}$ was chosen. As PSL (per default) assumes features to be binary, numerical features are binarized in a preprocessing step as discussed in Sect. 4.3.

The detailed experimental setup and implementation is publicly available[4] as is the implementation of the learning algorithm.[5]

In the next sections, we will attempt to answer the following research questions:

RQ1: Is greedy search sufficient to find a good model?
RQ2: Are the probability estimates of PSL well calibrated?
RQ3: How can the epistemic uncertainty in PSL stages be quantified?
RQ4: How does PSL perform when considering real-world attribute deployment costs?
RQ5: How does binarization as preprocessing compare with in-search binarization? Do the heuristic simplifications in the binarization hamper PSL performance?
RQ6: Is PSL applicable for ranking tasks? How well does it perform when being trained on class labels versus preference data?

First, however, we briefly summarize the datasets and baseline methods used in this study.

5.1 Datasets

5.1.1 Coronary heart disease data

The dataset for this case study has originally been 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 (the negative class) and patients in whom chest pain is probably caused by CHD (the positive class).

Briefly, 74 general practitioners (GP) recruited consecutively patients aged ≥ 35 who presented with chest pain as primary or secondary complaint. GPs took a standardized history and performed a physical examination. Patients and GPs were contacted 6 weeks and 6 months after the consultation. All relevant information about course of chest pain, diagnostic procedures and treatments had been gathered during 6 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. (2010).

[4] https://github.com/TRR318/pub-ml-psl/releases/tag/v1.1.0.
[5] https://github.com/TRR318/scikit-psl/releases/tag/v0.7.2.

Overall, the dataset comprises 1199 (135 CHD and 1064 non-CHD) patients described by ten binary attributes: (f_1) patient assumes pain is of cardiac origin, (f_2) muscle tension, (f_3) age gender compound, (f_4) pain is sharp, (f_5) pain depends on exercise, (f_6) known clinical vascular disease, (f_7) diabetes, (f_8) heart failure, (f_9) pain is not reproducible by palpation, (f_{10}) 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 can be restricted to positive scores. For the following experiments, the missing feature values have been imputed using the mode, representing the most frequent value of each feature.

5.1.2 Further datasets

In addition to the *CHD* dataset, we also consider two datasets from the UCI repository (Patrício et al., 2018; Ramana & Venkateswarlu, 2012). The first is concerned with the prediction of *breast cancer* for patients in Coimbra, Portugal. It comprises 9 features including resistin, glucose, age, and BMI. The dataset is of small size, with only 116 instances, of which 64 are positive. The second UCI dataset is concerned with *Indian liver patients* (*I-Liver*). It contains 583 instances, of which 416 are positive, i.e., the majority of patients are positive. The dataset contains 10 features including age, sex, total proteins. In contrast to the *CHD* data, the UCI datasets contain numeric features, which have to be binarized, e.g., by using the methods described in Sect. 4.3.

Furthermore, we employ two larger datasets. The first one is the *California Housing* dataset, which is available in the StatLib repository (Pace & Barry, 1990). This dataset with 20,460 entries and 8 features is originally a regression dataset regarding house prices, which we transformed into a classification dataset, where the target column indicates whether the house price is above (1) or below (0) the median. The fourth dataset in our study is the Fairlearn *ACSIncome* dataset (Ding et al., 2021). The original dataset consists of 10 features and overall 1,664,500 entries, of which we subsampled 50,000 for computational reasons. Additionally, the categorical features have been transformed to binary features. The detailed datapreparation process can be found in the experiment repository.

Finally, we consider two datasets which include feature acquisition costs: *BUPA* liver disorder and the *Thyroid* dataset (Forsyth, 2016; Quinlan, 1986). *BUPA* contains 345 observations and consists of 5 features that result from blood tests sensitive to liver disorders caused by excessive alcohol consumption. The cost of these features range from \$5.17 to \$9.86. The target variable indicates the number of half-pint equivalents of alcoholic beverages consumed per day, where class 0 is defined as "<3" and class 1 "≥ 3". The *Thyroid* dataset originally consists of 7200 data points of three classes indicating whether a patient is hypothyroid, hyperthyroid or has balanced thyroid levels. As PSL is a binary classifier, we removed the hypothyroid patients (smallest class) from the dataset, resulting in a binary classification dataset of 7034 observations with 20 features. The features range from simple information like the patient's age (imposing cost of \$1) and blood tests for thyroid hormones TSH, T3, TT4, T4U with cost ranging from \$9.31 up to \$22.75. In both datasets, the cost of a feature that results from a blood test is discounted if a blood test has already been conducted for another feature. For a detailed description of the data, we refer to Turney (1995).

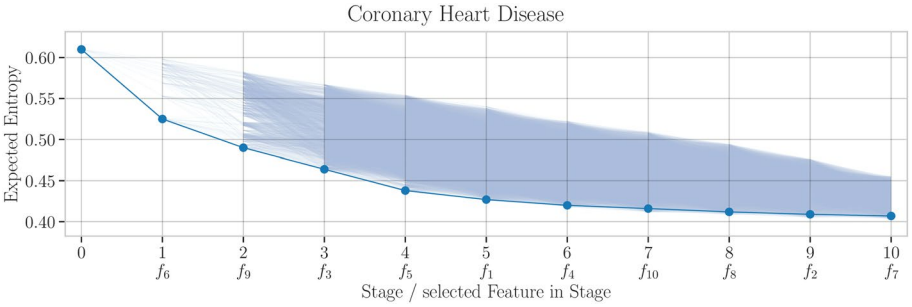

Fig. 2 Evaluation of the greedy learning algorithm (blue line) on the coronary heart disease dataset. The light blue lines show the complete search space induced by all feature permutations and possible score assignments. The features, selected by the greedy algorithm in every stage, are also labelled on the x-axis. The visualization was created for a score set $\mathcal{S} = \{1, 2, 3\}$ (Color figure online)

5.2 Baseline methods

Our method is compared with logistic regression (LR) as a baseline. We consider LR a sensible comparison for two reasons: First, LR is a linear model and thus comes with a high degree of interpretability: The influence of the features on the predictions is clearly determined by their associated coefficients in the regression model. Additionally, LR is known to yield well-calibrated probability estimates, which is essential for a comparison to PSL in order to assess the trustworthiness of its probability estimates.

For specific research questions, we use additional baselines as needed. In order to assess the early stopping capabilities at prediction time, we compare PSL with a decision tree (DT). Unlike LR, DTs do typically not yield well-calibrated probability estimates, but they can be queried sequentially, allowing prediction with feature subsets similar to PSL. This allows for a comparison relating attribute deployment cost to predictive performance. For evaluating probabilistic predictions, or, more specifically, a possible loss in predictive accuracy as a price to pay for interpretability, we additionally include a random forest (RF) and an XGBoost classifier. Although both of them are ensemble methods and exhibit a complex structure hindering inherent interpretability, they are considered state-of-the-art methods for probabilistic tabular classification and are thus an interesting comparison.

5.3 RQ1: Expected entropy minimization

The introduced algorithm iteratively selects the feature/score pair that minimizes the expected entropy (7) for each stage. As can be seen in Fig. 2, entropy continues to decrease, but the improvements diminish stage by stage and almost vanish after the addition of the fifth feature. Interestingly, this result is very much in agreement with previous studies on this data, and the top-5 features in Fig. 2 exactly correspond to those features that have eventually been included in the "Marburg Heart Score", a decision rule that is now in practical use.[6]

As our algorithm minimizes expected entropy on the training data greedily, one may wonder to what extent expected entropy is also minimized globally, i.e., across all stages.

[6] https://www.mdcalc.com/calc/4022/marburg-heart-score-mhs.

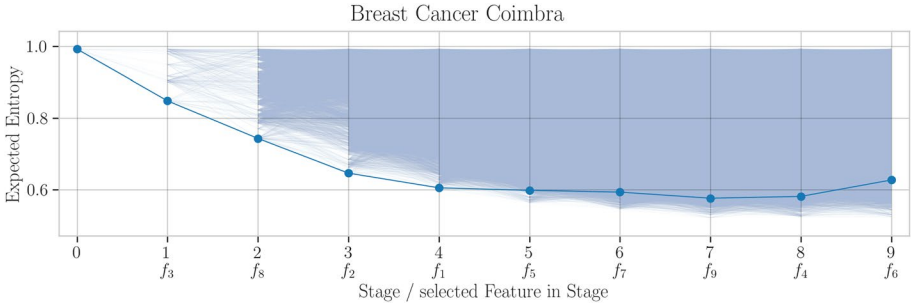

Fig. 3 Evaluation of the greedy algorithm on the breast cancer Coimbra dataset with a score set of $\mathcal{S} = \{-2, -1, +1, +2\}$ (refer to Fig. 2 for details)

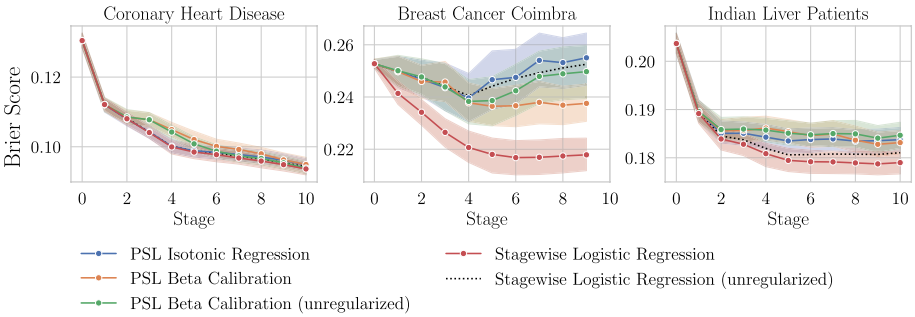

Fig. 4 Stagewise Brier score for PSLs on the test datasets

To get an idea, we compared the expected entropy curve produced by the greedy algorithm with the curves produced by all other PSLs. With the score set $\mathcal{S} = \{1, 2, 3\}$—a complete enumeration of the resulting set of PSLs is still feasible. As can be seen from Fig. 2, the greedy approach (shown in solid blue) performs well, at least on the *CHD* dataset. Figure 3 illustrates the result of the greedy parameter search on the *BCC* dataset. As we can see here, the curve of the selected model is clearly not the lower envelope, i.e. it does not achieve stagewise optimal performance. This may be due to the greedy search approach, which acts myopically in the sense that it irrevocably selects the locally best option and therefore could be missing out on global optima. However, note that stagewise optimal performance may also be impossible to achieve with a single scoring list, as the stagewise best performing models may stem from separate lists having different prefixes of feature/score pairs.

5.4 RQ2: Investigating probability estimates

Next, we investigate the trustworthiness of the probability (point) estimates of our proposed method. As already outlined in Sect. 4.4, the training data becomes more and more fragmented when progressing in the PSL, as the set of possible total scores grows. Consequently, the probability estimates for the individual instances are based on fewer and fewer training data. While the final probabilistic predictions are based on a joint estimation in

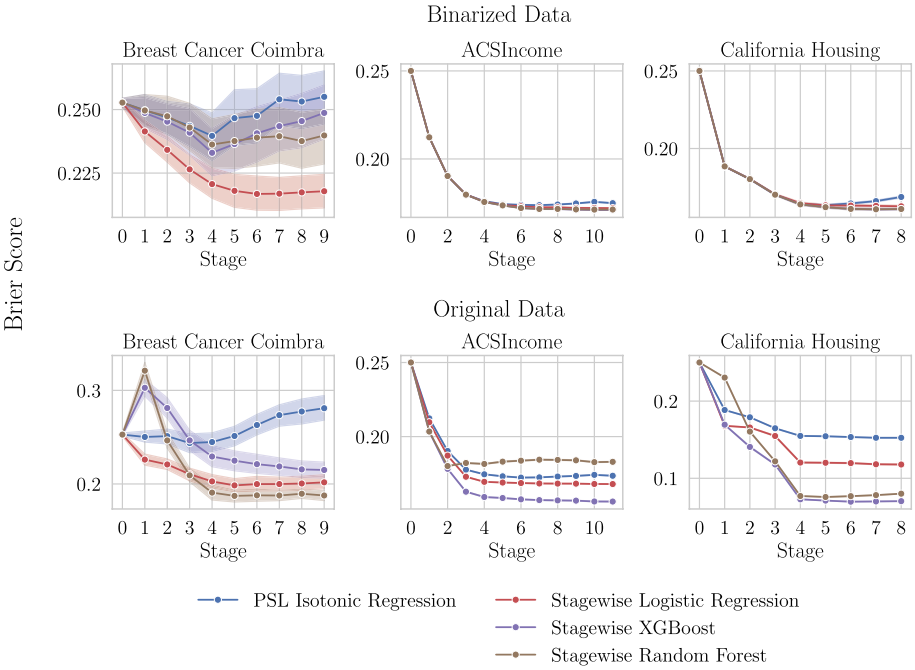

Fig. 5 Stagewise Brier score for PSLs on the *BCC* as well as large datasets

terms of probability calibration, it is not clear how the quality of the estimations develops throughout the PSL stages.

We evaluate this quality by computing the stagewise *Brier score* (Brier, 1950). Let $T_k(\boldsymbol{x}_i)$ denote the total score of some instance $\boldsymbol{x}_i$ at stage k. Having access to a set of test data $\mathcal{D}_{test} := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, the Brier score at stage k is given by

$$BS(k) := \frac{1}{N} \sum_{i=1}^{N} \left(\hat{q}(k, T_k(\boldsymbol{x}_i)) - y_i\right)^2.$$

Fig. 4 shows the stagewise Brier score for all considered datasets when using isotonic regression or beta calibration for PSL. As a baseline, we also train a LR model, using the same features as PSL in each stage. Note that, compared to PSL, LR is more flexible in the sense that scores are real-valued and not restricted to (small) integers. Additionally, the learning algorithm of LR employs L_2-regularization in order to prevent the model from overfitting. For all approaches, the features were binarized in advance (*preprocessing*) using the bisect heuristic (cf. Sect. 4.3).

As we can see, isotonic regression and beta calibration perform quite similarly for each of the considered datasets. On the *CHD* dataset, both are also on par with the stagewise logistic regression, which is known to produce well-calibrated models. However, on the *BCC* as well as the *I-Liver* dataset, the PSL variants achieve worse Brier scores than the logistic regression baseline. Their Brier scores even increase for higher stages, which is likely to be an overfitting effect: The dotted line indicates the performance of an unregularized LR, whose performance is very similar to the two considered PSL variants. Recall that the *BCC* dataset only contains 116 instances. The regularized LR manages to avoid

overfitting the training data and achieves better generalization performance in terms of stagewise Brier score. Although PSL is a quite restricted model with only a few integer weights, it tends to overfit if there is not enough training data available, e.g., in the case of the *BCC* dataset. Thus, when relying on point estimates and having only access to small amounts of training data, PSL needs to be regularized to circumvent this problem. We employed a regularized Beta calibration in order to avoid this overfitting effect. Although it outperforms the other PSL variants on the *BCC* dataset, it still falls significantly short of the LR baseline in terms of performance. Another alternative is to go from point to interval estimates, e.g., by using the Clopper–Pearson confidence intervals introduced in Sect. 4.4, whose application will be examined in Sect. 5.5.

To estimate the cost of interpretability in terms of predictive performance, we also compared PSL to state-of-the-art probabilistic classifiers XGBoost and RF. Figure 5 shows the stagewise Brier score of these predictors for the large dataset *ACSIncome* and *California Housing* in comparison to *BCC*. Here, we consider both binarized datasets and the original datasets including numerical features and use the PSL variant employing isotonic regression for porbability estimation. Although PSL is designed for binary features, we want to estimate how much performance is lost due to binarization. This is particularly relevant for the RF and XGBoost baselines, which can split several times on numerical features, effectively partitioning the feature space into smaller, more homogeneous regions. For binary features this is not possible. A full comparison of train and test performance of all considered methods on all datasets is depicted in Appendix A and a comparison of binarization techniques will be presented in Sect. 5.7.

For the binarized version, the two baseline models and and PSL deteriorate after 5 features, while LR remain improving their performance on the small *BCC* dataset. This is likely due to an overfitting effect of the flexible baseline models as well as overfitting of PSLs isotonic regression as discussed above. For the two large datasets, the baseline models dominate the performance of PSL, although not always by a large margin. PSLs performance deteriorates after stage 5 on the *California Housing* dataset. As more features increase the capacity of the learner, this may look like a standard overfitting effect. However, there is also an alternative, in a sense even opposite explanation. Note that we do not observe this deterioration for LR, which can modulate the influence of any additional feature in a very flexible way, by appropriately tuning the weight coefficient—up to completely ignoring a presumably unuseful feature by setting its weight to 0. PSL does not have this ability. Instead, it can only weight all features in (more or less) the same way. Therefore, in cases where adding another feature might be useful, but with a weight much smaller than the others, it might be better to omit it completely instead of giving it the same influence as the more important features. Seen from this perspective, the class of scoring systems is simply not flexible enough, and the deterioration might be due to a problem of underfitting rather than overfitting.

We observe a less homogeneous behavior for the original datasets including numerical features. On the *BCC* dataset, the flexible baseline methods RF and XGBoost are disadvantageous for lower stages. This is most likely due to an overfitting effect, as the feature space is very low dimensional at these stages. This causes the flexible models to make overly specific splits, that do not capture the underlying data distribution. In the higher stages, this disadvantage vanishes and PSLs performance deteriorates similar to the binarized case. For the *ACSIncome* dataset, XGBoost is the favorable method, dominating the other methods stagewise. Interestingly, RF is suffering from an overfitting effect here, although the dataset is much larger than *BCC* with 50,000 samples. XGBoost again dominates the other method on the *California Housing* dataset, whereas RF achieves similar performance.

Here, we observe the most pronounced performance gap to PSL and LR, as opposed to the binary case. One key consideration here, is that *California Housing* is known for exhibiting higher order feature interactions, e.g. between longitude and latitude of the house locations. Recall that linear models like PSL and LR are not able to capture these interactions, while XGBoost and RF can. While this might explain a significant part of the performance gap between XGBoost/RF and LR/PSL, these interactions seem to be of limited importance on already binarized data for this particular dataset. Nevertheless, LR utilizes numerical features more extensively than PSL, resulting in lower Brier scores throughout the stages.

5.5 RQ3: Uncertainty quantification and decision-making

In the previous section, we have discussed that the probability estimates in higher stages of the model are based on less and less data points as they get more fragmented. In this section, we investigate the applicability of the Clopper–Pearson confidence interval introduced in Sect. 4.4 in medical decision-making.

Figure 6 illustrates the point estimates as well as the confidence intervals exemplarily for several stages of a PSL trained on the *CHD* data. As expected, we observe an increased size of the confidence intervals in higher stages. This is not only caused by the increasingly fewer data points the relative frequency estimate is based on, but also by the Bonferroni correction, that grows in the size of the set of total scores $|\Sigma_k|$. For example, in stage 4, there is not a single data point that exhibits a total score of 7, hence the confidence bounds are fully determined by the neighboring total scores.

Since the *CHD* dataset is quite imbalanced, there are fewer data points for positive samples, hence the confidence intervals for predicting high true-class probabilities are wider, as there are fewer data points for the respective total scores. For stage 6 and beyond (see Fig. 6) the lower confidence bound never exceeds 0.5, meaning that the positive class can never be predicted with high confidence. Note, that the Clopper–Pearson confidence interval is a conservative guarantee of the probability estimate, i.e., when computing an 95% interval, the probability of the true parameter p_T laying outside the interval is *at most* 5%.

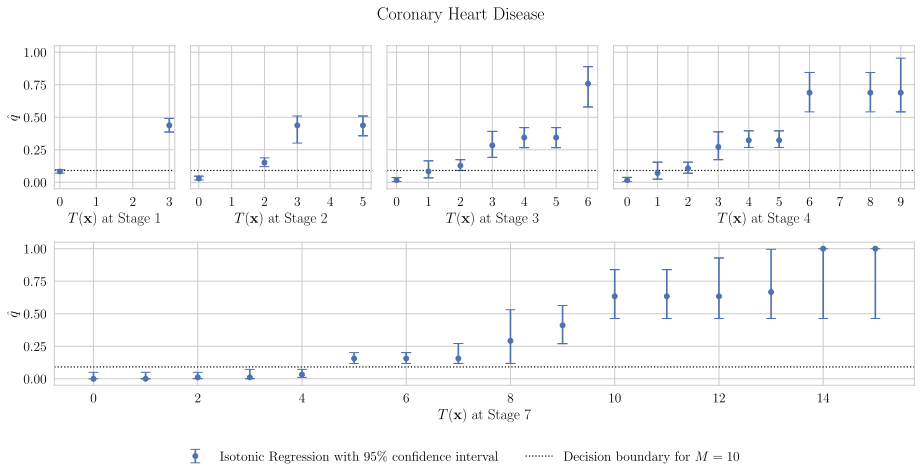

Fig. 6 Probability estimates of all possible total scores for the first 4 stages and stage 7 of the PSL, trained on the full *CHD* dataset. The error bars show the 95% confidence interval described in Sect. 4.4

Following this illustration of the Clopper–Pearson confidence intervals, we will showcase their usefulness in the context of risk-averse decision-making in medicine. In medical diagnosis, the consequences of a false negative prediction, i.e., not treating an ill patient, are typically far more severe than of a false positive. This asymmetry can be captured by a loss function that assigns a loss of 1 to a false positive and a loss of $M \gg 1$ to a false negative:

	Predicted Positive	Predicted Negative
Actual Positive	0 (TP)	M (FN)
Actual Negative	1 (FP)	0 (TN)

In the medical domain, this also goes under the notion of "diagnostic regret", and various empirical methods for eliciting preferences in decision-making (i.e., the cost factor M) have been proposed in the literature (Tsalatsanis et al., 2010; Moreira et al., 2009).

To minimize the risk of the decision, the negative class should only be predicted if its probability $(1 - \hat{p})$ is M times as high as the probability for the positive $\hat{p}$ class:

$$\hat{y} = \begin{cases} 1 & \text{if } 1 - \hat{p} < M \cdot \hat{p} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

The (estimated) expected loss for this risk-minimizing decision is therefore $E(\hat{y}) = \min\{1 - \hat{p}, M \cdot \hat{p}\}$. This decision boundary for $M = 10$ is visualized in Fig. 6. This decision strategy nicely emphasizes the importance of (accurate) probabilistic predictions and, more generally, uncertainty-awareness, in safety-critical domains.

To incorporate risk-awareness into the decision-making process, we propose to select not with respect to the point estimate of the probability, but the upper confidence bound. Again, referring to Fig. 6, we can observe that deciding with respect to the probability point estimate, data points with score 0 at stage 0 are classified negatively as they lay below the decision boundary. When using the upper confidence bound instead, all data points are classified negatively, as the upper confidence bound is slightly above the decision threshold.

Again, we compare the PSL variants to an LR model that is trained on the same features as the PSL on each stage. Accounting for uncertainty, we do not use the point estimate for the predicting the positive class but rather the upper bound of the corresponding 50% Clopper–Pearson confidence interval of the PSL estimate.

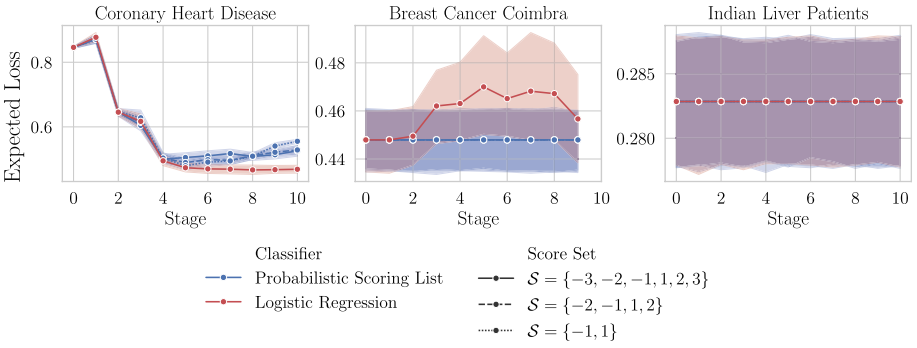

Fig. 7 Expected loss, calculated using the upper confidence bound of the 50% confidence interval

Figure 7 shows the loss for $M = 10$. The PSL has been configured with three different score sets. For the *CHD* dataset, we can see that all three PSL variants perform quite similarly, with small improvements for larger score sets. Moreover, they are all on a par with LR, sometimes even a bit better, which is quite remarkable. For all variants, we observe a monotonic decrease in loss until the fifth feature is added. Again, in the large majority of cases, the five top-features correspond to the features also included in the Marburg heart score. Adding further features leads to a slight deterioration for PSL.

The performance decrease after stage 5 can be explained similarly to the observations in Fig. 5.

For the *BCC* and the *I-Liver* dataset, the plots look drastically different. Recall, that at stage 0, the probability estimate is simply the relative frequency of the positive class in the training data. Thus, all instances are classified identically. Employing the risk-averse decision rule, we classify all patients as positive. Advancing from this is only possible if we make true negative predictions. For each false negative prediction, we need to make at least $M = 10$ true negative predictions, otherwise the expected loss will increase. Depending on the classification task at hand, this may be a very difficult problem. We observe that the PSL with 50% confidence intervals refrains from changing the initial classification, which results in a constant value of expected loss throughout the stages. LR on the other hand acts less risk-averse and introduces negative predictions for the *BCC* dataset. However, it does not introduce enough true negative predictions in order to compensate for the false negatives, resulting in a deterioration of the expected loss. For the *I-Liver* dataset, both methods refrain from changing the initial classification when more features become available.

5.6 RQ4: Attribute deployment cost

Designed as a decision list and providing probability estimates after each evaluated feature, PSL supports an early stopping strategy: predictions can possibly be made early without acquiring the full set of features. This allows for situation-adapted and possibly less costly predictions sensitive to the decision context at hand. Strictly speaking, PSL is not cost-sensitive, because the selection of features is solely based on the criterion of informativeness and does not consider any notion of cost. Or, stated differently, it implicitly assumes that all features have the same cost.

Nevertheless, we conducted a first study to assess the performance of PSL with regard to the trade-off between accuracy and cost. To this end, we used data from Turney (1995), which comes with feature acquisition costs, namely the *BUPA* liver disorder dataset and the *Thyroid* dataset. For an overview of the features and the associated costs, we refer to the appendix of Turney (1995). As a baseline to compare with, we used a decision tree classifier[7] (DT), because it can be applied with a varying number of features in a way similar to PSL: One simply makes a prediction based on the class distribution in the node reached after splitting on k features, even if that node is not yet a leaf node (in case a leaf node is reached earlier, a decision is made at that node).

For this experiment, we use the train-test-split that is associated with the individual datasets. Each instance was evaluated with an increasing amount of at most k features for both models. PSL was only stopped early if the probability collapses to extreme 0 or 1 estimates or if k features were used. This way, we can compare model sequences of increasing cost and complexity.

[7] The DT was trained with entropy as a splitting criterion.

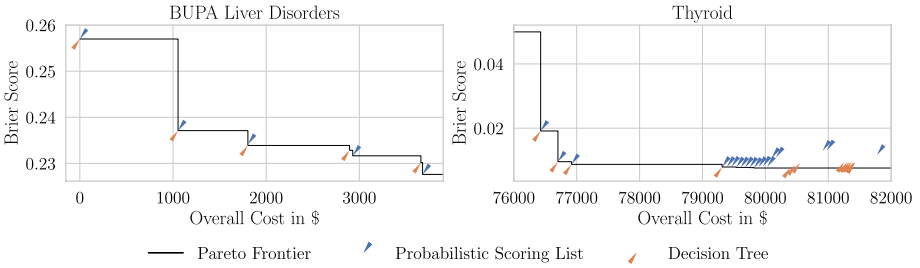

Fig. 8 Brier score achieved by PSL and DT with increasing amount of features used. The dashed line indicates the Pareto frontier of costs and Brier score with respect to both methods. Note that for the *Thyroid* dataset the first point at cost 0 was cut off, as this point is the same for PSL and DT and removing it yields a more insightful plot for the remaining points

Figure 8 shows the Brier score for PSL and DT at different cost levels induced by the maximum number of features k. Note that these cost levels do not necessarily coincide for DT and PSL, as DT's leafs may be reached earlier than PSL's final prediction. The overall cost is the sum of attribute costs spent for making predictions for all instances in the test dataset. For the *BUPA* dataset, we observe that the Brier score decreases for each added feature. However, DT yields worse probability calibration at higher cost levels. A possible explanation is that DT bases its probabilistic predictions solely on the data points in the current node, whereas PSL employs isotonic regression for probability estimation and thus also incorporates information from the neighboring buckets due to the monotonicity constraint. Overall, in this example, PSL yields highly competitive results despite being less flexible than DTs.

For the much larger *Thyroid* dataset, we observe that PSL and DT perform very similar and also at very similar cost points at the early stages. While both methods deteriorate in performance at higher cost levels, the deterioration is more pronounced for PSL. This may be explained by effects similar to those observed in Sects. 5.4 and 5.5, as PSL cannot freely adjust the influence of additional features due to the discrete score set.

5.7 RQ5: Binarization

As described in Sect. 4.3, dealing with numerical features is of great practical importance. We compare the *in-search* binarization of numerical features with the case in which the features are binarized independently in a *preprocessing* step. Additionally, we evaluate our heuristic optimization method, with which only a logarithmic number of candidate binarization thresholds need to be checked.

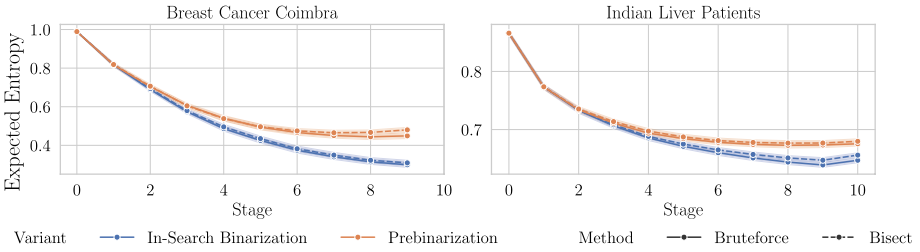

Fig. 9 Comparison of pre-binarization and in-search binarization, with heuristic and brute-force threshold optimization

Figure 9 shows the comparison on the *BCC* and the *I-Liver* in terms of the stagewise expected entropy. In both datasets, the *in-search* binarization is advantageous. The advantage becomes more pronounced throughout the stages of the PSL. Intuitively, this can be explained by the fact that in *preprocessing*, the features are binarized individually. In contrast to that, the *in-search* procedure binarizes the features sequentially during the greedy construction of the PSL. Thus, when adding a feature/score pair to the PSL, it can take into account dependencies between already selected features and the newly chosen feature by setting the binarization threshold accordingly. As discussed previously, in situations where adding another feature with the same weight as previous (more important) features is undesirable, abstaining from using a feature is a sensible option and avoids performance deterioration. *In-search* binarization can enable this, by setting the threshold higher than the maximum value of the feature, effectively ignoring it even if a non-zero score is assigned.

For both methods, we observe that the bisecting search heuristic leads to a negligible deterioration of performance in terms of expected entropy. Consequently, we consider *in-search* binarization with the bisect search a sensible default configuration and use it for the remaining experiments of this paper.

5.8 RQ6: Ranking

In the following, we will evaluate the ranking performance of our proposed method. To this end, we consider the AUC (Area under the ROC Curve) (Fawcett, 2006). For a binary scoring classifier, the AUC can be interpreted as the probability of ranking a randomly chosen positive example before a randomly chosen negative example.

When fitting the classifier, we consider two settings: In the first one, we assume to have a training dataset as described in the previous experiments of this paper. Here, we use the standard greedy learning algorithm optimizing the expected entropy (7) (blue line). In the second scenario, we exclusively require access to pairwise comparisons $x \succ x'$ with $x, x' \in \mathcal{X}$ and instantiate the PSL with the pairwise soft rank loss (14) (orange line).

Figure 10 shows the stagewise AUC. For the *CHD* dataset, we observe a monotonic increase for both settings. For the other two datasets, the values increase in the beginning until a plateau is reached at stage 4. The two settings lead to very similar results for the *BCC* and *I-Liver* datasets. For the breast cancer dataset, optimizing expected entropy leads to better AUC values than the soft rank loss. However, as already mentioned, the *BCC* dataset is very small. We conclude that the PSL is generally applicable for the task of ranking.

6 Summary and conclusion

In this paper, we introduced probabilistic scoring lists, a probabilistic extension of scoring systems. While retaining the genuine interpretability of standard scoring systems, PSL provides additional information about uncertainty by turning scores into well-calibrated probability estimates. Moreover, these estimates can be gradually refined by adding more features. This may be important if features are expensive or time-consuming to obtain, so that rough estimates can be obtained quickly and decisions can be made early without the need to obtain scores for all features.

Our experiments on binary and binarized datasets showed that our greedy learning algorithm manages to construct well-performing models. We compared PSL with sensible

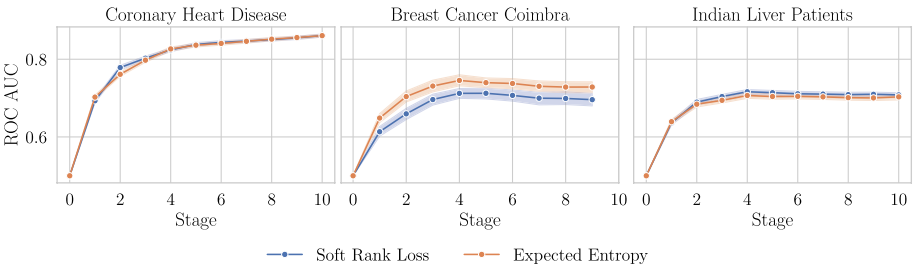

Fig. 10 Stagewise AUC (area under the ROC curve) for all datasets. The blue line illustrates learning directly using the original expected entropy loss, while the orange line corresponds to learning from preference data derived from the dataset using SRL (Color figure online)

baseline methods, demonstrating its suitability for situated decision support and combining the following strengths:

- Interpretability and transparency,
- well-calibrated probability estimates,
- uncertainty-aware interval estimates.
- early stopping at prediction time,

We additionally evaluated a method for feature binarization within the learning procedure of PSL and showcased its applicability for ranking tasks.

Building on the approach presented in this paper, we plan to address the following extensions in future work:

- Although the greedy learning algorithm proposed in this paper seems to perform quite well, more sophisticated algorithms for learning PSLs should be developed, including algorithms tailored to specific loss functions.
- So far, our proposed learning algorithm does not consider feature costs. In practice, the cost of feature acquisition may vary drastically for different features. For example, measuring the temperature of a patient can be done much more easily than a sophisticated blood test. An interesting direction for future work is to consider feature costs explicitly and develop a learning algorithm for cost-effective PSLs (Clertant et al., 2019).
- In practice, some features may be not available at inference time, e.g., because there is no access to laboratory equipment in the field. Here, it would be interesting to consider default scores for missing features and develop a more sensitive version of PSL that produces models that are robust to missing features.
- So far, we have only examined PSL from an algorithmic point of view. Conducting a user study and investigating how suitable it is for the task of supporting human decision makers is of major interest. This also involves the development of usability features such as advanced visualizations of PSL stages.
- In the current implementation, the score set is a hyperparameter that has to be set in advance. We would like to investigate ways to automatically select score sets that lead to performant yet meaningful and interpretable scoring lists.

Appendix

A Train and test performance on all datasets

Figure 11 shows the brier score of several PSLs configured with different score sets as well as various baselines on non-binarized data. Figure 12 shows evaluations for the same

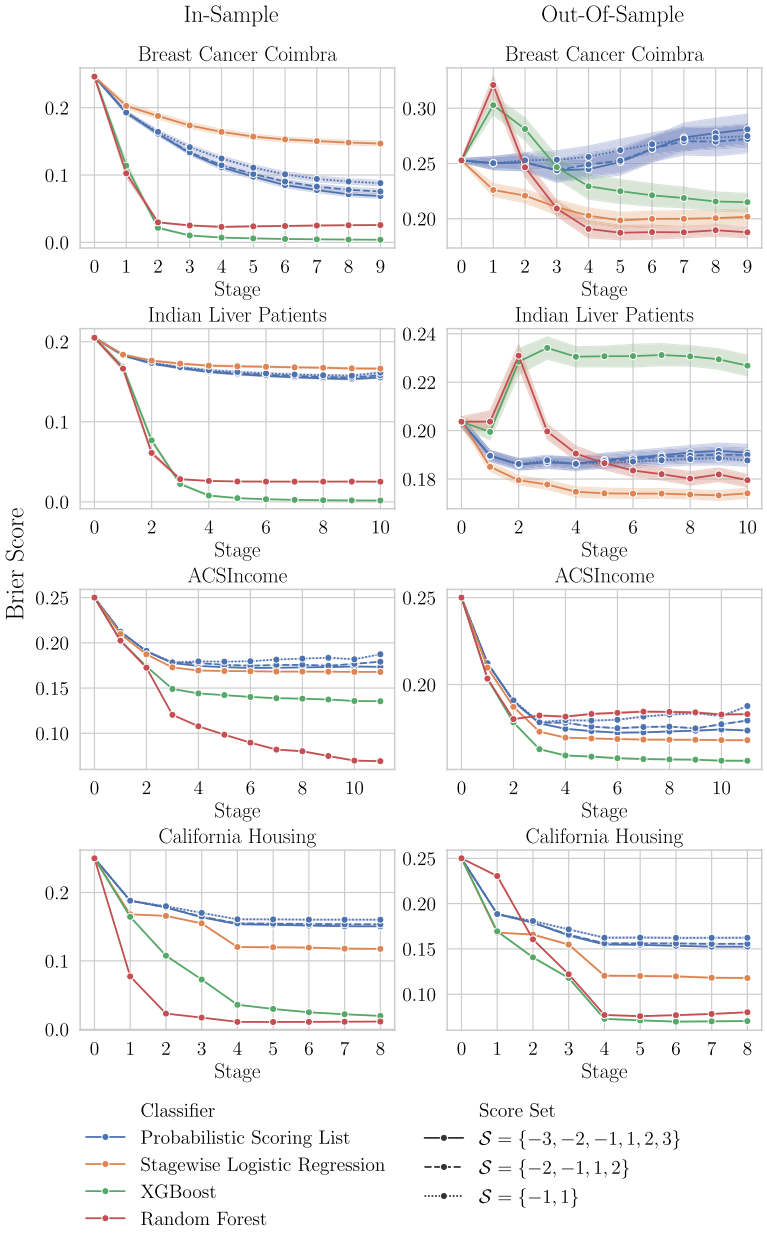

Fig. 11 Brier score for baseline methods and PSL instantiations with different score sets for all datasets without binarization

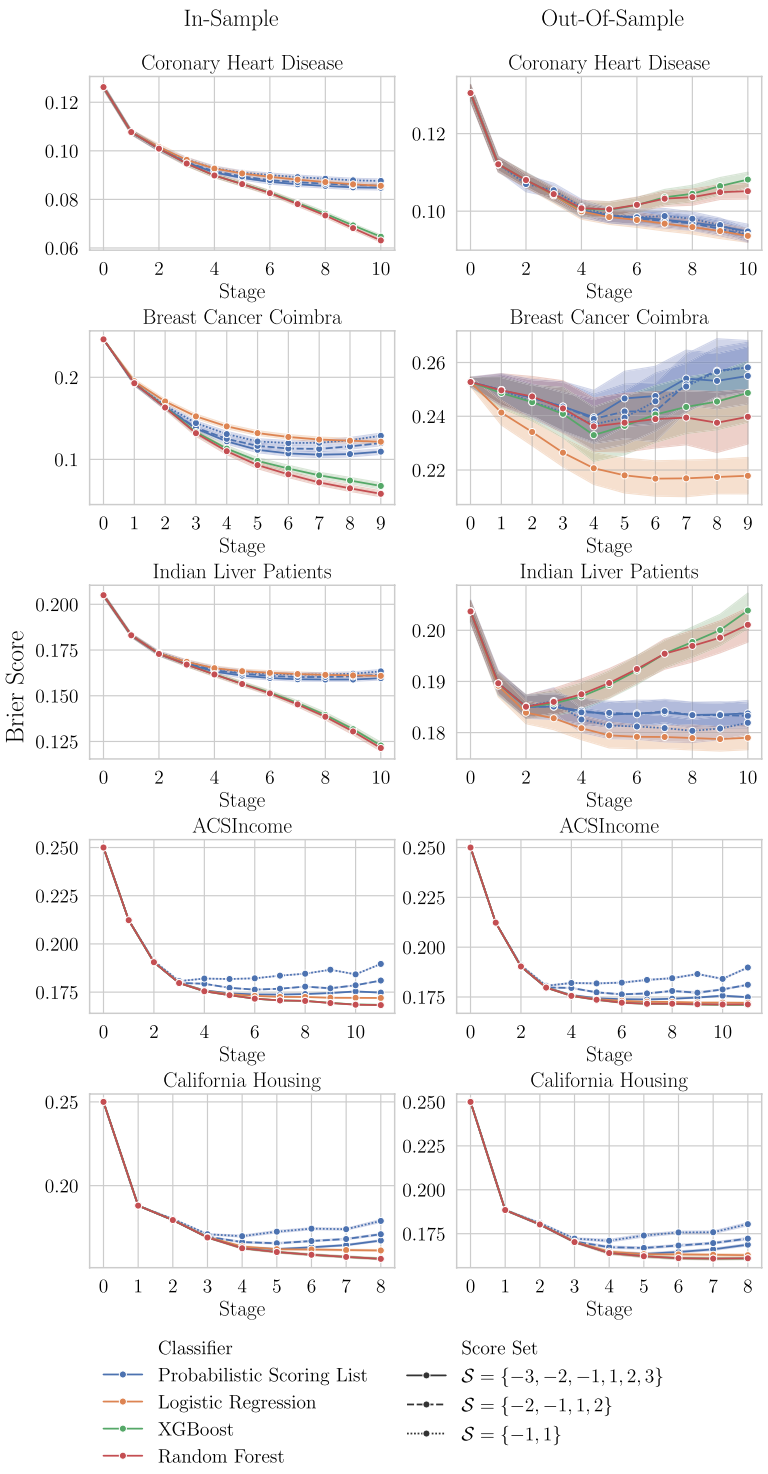

Fig. 12 Brier score for baseline methods and PSL instantiations with different score sets for all binarized datasets

learning algorithms, however, on the binarized datasets. Note, that the *CHD* dataset only consists of binary features and is thus omitted from the non-binarized plot. Train and test performance are reported, in order to identify overfitting.

Acknowledgements We gratefully acknowledge funding by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG): TRR 318/1 2021 – 438445824 and the German Research Foundation (DFG) within the Collaborative Research Center "On-The-Fly Computing" (SFB 901/3 Project No. 160364472).

Author contributions J.F. and E.H. conceptualized the proposed model. J.H., S.H. and E.H. wrote the main manuscript text, J.H. and S.H. implemented the model, conducted experiments and analyzed empirical results. All authors reviewed and finalized the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The Coronary Heart Disease dataset cannot be shared due to legal restrictions.All other datasets are publicly available and references are given in the manuscript.

Declarations

Competing interests The authors declare no competing interests.

References

Bösner, S., Becker, A., Hani, M., Keller, H., Sonnichsen, A., Haasenritter, J., Karatolios, K., Schäfer, J., Baum, E., & Donner-Banzhoff, N. (2010). Accuracy of symptoms and signs for coronary heart disease assessed in primary care. *British Journal of General Practice, 60*(575), 246–257.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review, 78*(1), 1–3.

Chevaleyre, Y., Koriche, F., & Zucker, J. (2013). Rounding methods for discrete linear classification. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16–21 June 2013, Volume 28 of JMLR Workshop and Conference Proceedings* (pp. 651–659). JMLR.

Clertant, M., Sokolovska, N., Chevaleyre, Y., & Hanczar, B. (2019). Interpretable cascade classifiers with abstention. In Chaudhuri, K., Sugiyama, M., (Eds.), *The 22nd international conference on artificial intelligence and statistics, AISTATS 2019, 16–18 April 2019, Naha, Okinawa, Japan, Volume 89 of Proceedings of Machine Learning Research* (pp. 2312–2320). PMLR.

Clopper, C. J., & Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika, 26*(4), 404–413. https://doi.org/10.1093/biomet/26.4.404

Crama, Y., & Hammer, P. L. (2011). *Boolean Functions - Theory, Algorithms, and Applications, Volume 142 of Encyclopedia of Mathematics and its Applications*. Cambridge University Press.

de Menezes e Silva Filho, T., Song, H., Perelló-Nieto, M., Santos-Rodríguez, R., Kull, M., & Flach, P. A. (2023). Classifier calibration: A survey on how to assess and improve predicted class probabilities. *Machine Learning, 112*(9), 3211–3260. https://doi.org/10.1007/S10994-023-06336-7

Ding, F., Hardt, M., Miller, J., & Schmidt, L. (2021). Retiring adult: New datasets for fair machine learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., Vaughan, J. W. (Eds.), *Advances in*

neural information processing systems 34: Annual conference on neural information processing systems 2021, NeurIPS 2021, December 6–14, 2021, virtual (pp. 6478–6490).

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861–874. https://doi.org/10.1016/J.PATREC.2005.10.010

Fayyad, U. M., & Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In Bajcsy, R. (Ed.), *Proceedings of the 13th international joint conference on artificial intelligence. Chambéry, France, August 28–September 3, 1993* (pp. 1022–1029). Morgan Kaufmann.

Forsyth, R. (2016). Liver disorders. *UCI Machine Learning Repository*. https://doi.org/10.24432/C54G67

Foygel Barber, R., Candes, E. J., Ramdas, A., & Tibshirani, R. J. (2021). The limits of distribution-free conditional predictive inference. *Information and Inference, 10*(2), 455–482. https://doi.org/10.1093/imaiai/iaaa017

Fürnkranz, J. (1999). Separate-and-conquer rule learning. *Artificial Intelligence Review, 13*(1), 3–54. https://doi.org/10.1023/A:1006524209794

Fürnkranz, J., Gamberger, D., & Lavrac, N. (2012). *Foundations of rule learning*. Berlin: Springer.

Hanselle, J., Fürnkranz, J., & Hüllermeier, E. (2023). Probabilistic scoring lists for interpretable machine learning. Lecture Notes in Computer ScienceIn A. Bifet, A. C. Lorena, R. P. Ribeiro, J. Gama, & P. H. Abreu (Eds.), *Discovery science - 26th international conference, DS 2023, Porto, Portugal, October 9–11, 2023, Proceedings* (Vol. 14276, pp. 189–203). Springer.

Hastie, T. J. (1990). *Generalized additive models*. New York: Routledge.

Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning, 110*(3), 457–506. https://doi.org/10.1007/S10994-021-05946-3

Knaus, W. A., Draper, E. A., Wagner, D. P., & Zimmerman, J. E. (1985). APACHE II: A severity of disease classification system. *Critical Care Medicine, 13*(10), 818–829.

Knaus, W. A., Wagner, D. P., Draper, E. A., Zimmerman, J. E., Bergner, M., Bastos, P. G., Sirio, C. A., Murphy, D. J., Lotring, T., & Damiano, A. (1991). The APACHE III prognostic system: Risk prediction of hospital mortality for critically III hospitalized adults. *Chest, 100*(6), 1619–1636.

Knaus, W. A., Zimmerman, J. E., Wagner, D. P., Draper, E. A., & Lawrence, D. E. (1981). APACHE-acute physiology and chronic health evaluation: A physiologically based classification system. *Critical Care Medicine, 9*(8), 591–597.

Kotlowski, W., Dembczynski, K., & Hüllermeier, E. (2011). Bipartite ranking through minimization of univariate loss. In Getoor, L., Scheffer, T. (Eds.), *Proceedings of the 28th international conference on machine learning, ICML 2011, Bellevue, Washington, USA, June 28–July 2, 2011* (pp. 1113–1120). Omnipress.

Kull, M., de Menezes e Silva Filho, T., & Flach, P. A. (2017). Beta calibration: A well-founded and easily implemented improvement on logistic calibration for binary classifiers. In Singh, A., Zhu, X. J. (Eds.), *Proceedings of the 20th international conference on artificial intelligence and statistics, AISTATS 2017, 20–22 April 2017, Fort Lauderdale, FL, USA, Volume 54 of proceedings of machine learning research* (pp. 623–631(. PMLR.

Le Gall, J. R., Lemeshow, S., & Saulnier, F. (1993). A new simplified acute physiology score (SAPS II) based on a European/North American multicenter study. *JAMA, 270*(24), 2957–2963.

Le Gall, J. R., Loirat, P., Alperovitch, A., Glaser, P., Granthil, C., Mathieu, D., Mercier, P., Thomas, R., & Villers, D. (1984). A simplified acute physiology score for ICU patients. *Critical Care Medicine, 12*(11), 975–977. https://doi.org/10.1097/00003246-198411000-00012

Lip, G. Y., Nieuwlaat, R., Pisters, R., Lane, D. A., & Crijns, H. J. (2010). Refining clinical risk stratification for predicting stroke and thromboembolism in atrial fibrillation using a novel risk factor-based approach: The euro heart survey on atrial fibrillation. *Chest, 137*(2), 263–272.

Lundberg, S. M., & Lee, S. (2017). A unified approach to interpreting model predictions. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., Garnett, R. (Eds.), *Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, December 4–9, 2017, Long Beach, CA, USA* (pp. 4765–4774).

Metnitz, P. G. H., Moreno, R. P., Almeida, E., Jordan, B., Bauer, P., Campos, R. A., Iapichino, G., Edbrooke, D., Capuzzo, M., Le Gall, J. R., and on behalf of the SAPS 3 Investigators. (2005). SAPS 3-From evaluation of the patient to evaluation of the intensive care unit. Part 1: Objectives, methods and cohort description. *Intensive Care Medicine, 31*(10), 1336–1344. https://doi.org/10.1007/s00134-005-2762-6

Moreira, J., Bisig, B., Muwawenimana, P., Basinga, P., Bisoffi, Z., Haegeman, F., Kishore, P., & Van den Ende, J. (2009). Weighing harm in therapeutic decisions of smear-negative pulmonary tuberculosis. *Medical Decision Making, 29*(3), 380–390.

Moreno, R. P., Metnitz, P. G. H., Almeida, E., Jordan, B., Bauer, P., Campos, R. A., Iapichino, G., Edbrooke, D., Capuzzo, M., Le Gall, J. R., and on behalf of the SAPS 3 Investigators. (2005). SAPS 3-From evaluation

of the patient to evaluation of the intensive care unit. Part 2: Development of a prognostic model for hospital mortality at ICU admission. *Intensive Care Medicine,31*(10), 1345–1355. https://doi.org/10.1007/s00134-005-2763-5

Možina, M., Demšar, J., Bratko, I., & Žabkar, J. (2019). Extreme value correction: A method for correcting optimistic estimations in rule learning. *Machine Learning, 108*(2), 297–329. https://doi.org/10.1007/S10994-018-5731-3

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In Raedt, L. D., Wrobel, S. (Eds.), *Machine learning, proceedings of the twenty-second international conference (ICML 2005), Bonn, Germany, August 7–11, 2005, Volume 119 of ACM international conference proceeding series* (pp. 625–632). ACM.

Pace, R. K., & Barry, R. (1990). *California housing dataset*. StatLib.

Patrício, M., Pereira, J., Crisstomo, J., Matafome, P., Seiça, R., & Caramelo, F. (2018). Breast cancer coimbra. https://doi.org/10.24432/C52P59.

Provost, F. J., & Domingos, P. M. (2003). Tree induction for probability-based ranking. *Machine Learning, 52*(3), 199–215. https://doi.org/10.1023/A:1024099825458

Quinlan, J. R. (1986). Thyroid disease. https://doi.org/10.24432/C5D010.

Ramana, B., & Venkateswarlu, N. B. (2012). ILPD (Indian liver patient dataset). https://doi.org/10.24432/C5D02C.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In Krishnapuram, B., Shah, M., Smola, A. J., Aggarwal, C. C., Shen, D., Rastogi, R. (Eds.), *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA, August 13–17, 2016* (pp. 1135–1144). ACM.

Rivest, R. L. (1987). Learning decision lists. *Machine Learning, 2*(3), 229–246. https://doi.org/10.1007/BF00058680

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*(5), 206–215. https://doi.org/10.1038/S42256-019-0048-X

Senge, R., Bösner, S., Dembczynski, K., Haasenritter, J., Hirsch, O., Donner-Banzhoff, N., & Hüllermeier, E. (2014). Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. *Information Science, 255*, 16–29. https://doi.org/10.1016/J.INS.2013.07.030

Simsek, Ö., & Buckmann, M. (2016). On learning decision heuristics. In Guy, T. V., Kárný, M., Insua, D. R., Wolpert, D. H. (Eds.), *Proceedings of the NIPS 2016 workshop on imperfect decision makers: Admitting real-world rationality, Barcelona, Spain, December 9, 2016, Volume 58 of proceedings of machine learning research* (pp. 75–85). PMLR.

Six, A. J., Backus, B. E., & Kelder, J. C. (2008). Chest pain in the emergency room: Value of the heart score. *Netherlands Heart Journal, 16*(6), 191–196.

Subramanian, V., Mascha, E. J., & Kattan, M. W. (2021). Developing a clinical prediction score: Comparing prediction accuracy of integer scores to statistical regression models. *Anesthesia and Analgesia, 132*(6), 1603–1613.

Sulzmann, J., & Fürnkranz, J. (2009). An empirical comparison of probability estimation techniques for probabilistic rules. Lecture notes in computer science. In J. Gama, V. S. Costa, A. M. Jorge, & P. Brazdil (Eds.), *Discovery science, 12th international conference, DS 2009, Porto, Portugal, October 3–5, 2009* (Vol. 5808, pp. 317–331). Springer.

Tsalatsanis, A., Hozo, I., Vickers, A. J., & Djulbegovic, B. (2010). A regret theory approach to decision curve analysis: A novel method for eliciting decision makers' preferences and decision-making. *BMC Medical Informatics Decision Making, 10*, 51. https://doi.org/10.1186/1472-6947-10-51

Turney, P. D. (1995). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research, 2*, 369–409. https://doi.org/10.1613/JAIR.120

Ustun, B., & Rudin, C. (2017). Optimized risk scores. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, Halifax, NS, Canada, August 13–17, 2017* (pp. 1125–1134). ACM.

Ustun, B., & Rudin, C. (2016). Supersparse linear integer models for optimized medical scoring systems. *Machine Learning, 102*(3), 349–391. https://doi.org/10.1007/S10994-015-5528-6

Ustun, B., & Rudin, C. (2019). Learning optimized risk scores. *Journal of Machine Learning Research, 20*, 1501–15075.

Vincent, J. L., Moreno, R., Takala, J., Willatts, S., De Mendonça, A., Bruining, H., Reinhart, C. K., Suter, P. M., & Thijs, L. G. (1996). The SOFA (sepsis-related organ failure assessment) score to describe organ dysfunction/failure. On behalf of the Working Group on Sepsis-Related Problems of the European Society of Intensive Care Medicine. *Intensive Care Medicine,22*(7), 707–710. https://doi.org/10.1007/BF01709751

Wang, C., Han, B., Patel, B., & Rudin, C. (2023). In pursuit of interpretable, fair and accurate machine learning for criminal recidivism prediction. *Journal of Quantitative Criminology, 39*(2), 519–581.
Webb, G. I. (1994). Recent progress in learning decision lists by prepending inferred rules. In *Proceedings of the 2nd Singapore international conference on intelligent systems* (pp. 280–285).