European Journal of Control 66 (2022) 100647

Contents lists available at ScienceDirect

European Journal of Control

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

Active preference-based optimization for human-in-the-loop feature selection☆

Federico Bianchi[a,*], Luigi Piroddi[a], Alberto Bemporad[b], Geza Halasz[c], Matteo Villani[d], Dario Piga[e]

[a] *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano 20133, Italy*
[b] *IMT School for Advanced Studies Lucca, Lucca 55100, Italy*
[c] *Cardiology Department Guglielmo Da Saliceto Hospital, Piacenza 29121, Italy*
[d] *Anesthesiology and ICU Department Guglielmo da Saliceto Hospital, Piacenza 29121, Italy*
[e] *IDSIA - Dalle Molle Institute for Artificial Intelligence Research - USI/SUPSI, Lugano-Viganello CH - 6962, Switzerland*

ARTICLE INFO

Article history:
Received 3 February 2022
Accepted 14 April 2022
Available online 2 May 2022

Recommended by Prof. T Parisini

Keywords:
Feature selection
Preference-based learning
Randomized algorithms
Human-in-the-loop

ABSTRACT

In various classification problems characterized by a large number of features, feature selection (FS) is essential to guarantee generalization capabilities. The FS problem is often ill-posed due to significant correlations among features, which may lead to several different feature subsets with comparable scores in terms of classification performance. However, not all these subsets are equivalent from a domain-oriented point of view due to known relationships among features and their different acquisition costs in production to deploy the trained classifier. In this paper, we consider the potential benefits of including the domain expert's preferences in the FS task, thus integrating both objective elements (*e.g.*, classification accuracy) and subjective (often not quantifiable) considerations in the selection process. This goes in the direction of increasing the interpretability and the trustworthiness of the machine learning model, which is an often desired property in many application domains such as in medicine. The proposed method consists of an iterative procedure. At each iteration, the expert is asked to express a "human" preference on pairs of classifiers, each one trained from a different subset of features. The expressed preferences are used algorithmically to update a suitable surrogate function that mimics the latent subjective expert's objective function, and then to propose a new classifier for testing and comparison. The proposed method has been tested on academic and experimental FS problems, and notably, on a COVID'19 patients record. The preliminary experimental results are promising, in that a parsimonious and accurate solution is obtained after a relatively short number of iterations.

1. Introduction

Feature selection (FS) is of paramount importance in classification problems characterized by a large number of features, in particular in the presence of small-size training datasets. FS amounts to a combinatorial problem that aims at extracting the relevant features from a given set of candidate ones. FS methods can be broadly classified into filter and wrapper methods [27,29,30]. In filter methods, FS is performed independently of the classifier design, based only on the intrinsic properties of the features. In wrapper methods, the criterion for selecting features is based on the performance of the classifier *i.e.*, the classifier is "wrapped" on a search algorithm that seeks the subset of features which results in the highest classification accuracy. Wrapper methods are typically more accurate, but also computationally intensive. Besides in machine learning applications, the problem of selecting the most relevant terms/samples for a given set has been addressed in different research areas. For example, in the systems and control area, the techniques answering to this selection problem are known under the collective term *model structure selection*, while in the signal processing community they are known as *sparse approximation and compressed sensing*, see *e.g,* [22].

FS from a large set of features is often an ill-posed problem, due to significant correlation among features. As a result, it is a common experience that several equivalent classifiers can be obtained based on different sets of selected features, with equivalence measured in terms of classification accuracy or other discrete performance-oriented indicators. It follows that the FS algorithm cannot pick one feature set over the others based on ob-

☆ Prof. T. Parisini

* Corresponding author.
E-mail addresses: luigi.piroddi@polimi.it, federico.bianchi@polimi.it (L. Piroddi).

https://doi.org/10.1016/j.ejcon.2022.100647

jective and quantitative elements. However, models with equivalent performance do not necessarily convey the same level of information to the domain expert regarding selected features, interactions among them, and model interpretability. For example, some features might be associated with the effects of the modeled phenomenon rather than its causes. Furthermore, some features may be costly to obtain in practice (for example, those associated with invasive clinical analysis, as opposed to other features obtained by standard non-invasive exams). Other features may be associated with noisy and unreliable measurements. Ultimately, the domain expert may prefer classifiers based on specific combinations of features more often associated with the modeled phenomenon, or for classifiers whose performance is more accurate on specific, critical samples. From these considerations, it is apparent that the FS problem ultimately amounts to a multi-objective optimization problem that accounts for both performance-oriented and subjective criteria, the latter being related to model interpretability and explainability and thus often not easy to mathematically formalize, [23]. At the same time, introducing human feedbacks implies new sources of complexity related to variability of the expert's current focus, [11].

In this work, we investigate the potential benefits of including the domain expert's preferences in the FS task, thus integrating both objective elements (*e.g.*, classification accuracy) and subjective considerations in the selection process. This "human-in-the-loop" can be beneficial in solving the above mentioned ambiguities, [20]. In short, the proposed algorithm suitably generates candidate solutions to submit to the expert, who is occasionally required to express a preference within pairs of solutions. This coarse information is then employed to "alter" the objectives of the FS algorithm, to drive it towards solutions that guarantee a required level of accuracy but at the same time are agreeable to the expert according to his or her subjective preferences. While human-AI interaction in machine learning has been deeply exploited for data producing, labeling, and pre-processing, relative few works deals with human-AI interaction in machine learning modeling tasks, such as FS (see [23] for details). In [14] the authors suggest to use domain human expert knowledge to select among equally important features in the proposed wrapper FS method, but no method is proposed based on this idea. In [13], a Reinforcement Learning method for FS is described where human experts only provide some initial information regarding the most relevant features, while during the learning process no human intervention is exploited.

The preference-based FS method described in this paper builds upon a tailored extension of the GLISp algorithm proposed by some of the authors for real-valued black-box global optimization through active preference learning [3]. The primary motivation behind using GLISp and other preference-based optimization algorithms (see, e.g., [1,4,10,17]) is that many real-world problems require optimizing a qualitative objective function. The function may be difficult to quantify, as a human decision-maker can only qualitatively assess the "goodness" of a solution. In this case, it is well known that humans are better at expressing a preference between two options ("A is better than B") rather than defining a fictitious metric to assess multiple solutions quantitatively [12].

In preference-based optimization, the expert's preferences are used to build a *surrogate* cost function describing his/her evaluation of different solutions. In turn, this surrogate function is used to build an *acquisition* function, which is optimized to select the next candidate solution to propose to the user for comparison with the current best. The acquisition function balances exploitation (optimization only based on the surrogate cost function describing the observed preferences) and exploration (searching unexplored areas of the solution domain). In the present research endeavor, the GLISp approach is reformulated for a combinatorial optimization framework and tailored to the FS task. The discrete nature of the optimization problem is explicitly accounted for in the construction of both the surrogate and acquisition functions. Notice also that, again due to the discrete nature of the problem, the optimization of the acquisition function may occasionally yield a previously explored solution, which can never occur in the continuous setting. To avoid presenting to the expert already seen solutions, a heuristic method is applied to locally perturb the solution. To optimize the acquisition function we here employ the Randomized FS and Classification (RFSC) algorithm described in [9].

The RFSC is a wrapper algorithm, that employs a multi-model criterion for assessing the importance of each feature, for increased robustness. More specifically, at each iteration of the RFSC algorithm a set of models is extracted from a probability distribution defined over all the possible feature subsets. These models are estimated and evaluated, and the aggregate information regarding their performances is used to update the probability distribution, by reinforcing the probability to extract features that appear in successful models more often than not. Ultimately, the distribution converges to a limit distribution corresponding to a single model. The RFSC has several desirable features: a) it only requires the evaluation of the cost function; b) it generally provides an excellent tradeoff between model complexity and classification accuracy; c) it is not prone to error accumulation problems (as sequential methods); d) it operates the selection based on robust evidence gathered on a population of models; e) thanks to the randomization it can occasionally escape from local minima. All these features, and especially its robustness (due to the multi-model criterion for FS), make the RFSC well-suited for the GLISp framework. Besides, the sample-and-evaluate strategy exploited by the RFSC has been successfully applied for feature selection in several discrete and continuous problems, as discussed in [6–8,16].

While the effectiveness of the GLISp and the RFSC has been already analyzed in [3] and [9] with reference to several numerical data sets taken from public available repositories, in this paper we are mainly interested in investigating their combination to solve FS problems within a human-in-the-loop framework [20].

The proposed method has been tested on both academic and experimental FS problems, and notably, a COVID'19 patients record, demonstrating its ability to drive the selection process towards solutions that optimize an unknown criterion, manifested to the algorithm only utilizing the expert preferences. Regarding the COVID'19 dataset, classifiers for mortality prediction in patients with COVID-19 pneumonia have been trained. A human-in-the-loop experiment is also documented where the trained classifiers are proposed to a human medical expert, who iteratively expresses pairwise preferences between two classifiers according to his (unknown) subjective understanding and model interpretability.

The main contributions of this work are:

- A novel FS approach that accounts for human-AI interaction, by resorting to the expert advise for better tuning of the optimization process;
- A tailored extension of the GLISp algorithm for discrete optimization;
- Presentation and discussion of a medical human-in-the-loop experiment related to prognosis for COVID'19.

We stress again that this work is not meant to make a comparison with existing classical FS approaches, but rather to discuss the potential benefit of considering human-AI interaction for ill-posed FS problems, where classical methods are at a loss.

The rest of the paper is organized as follows. Section 2 presents the active preference-based FS problem. Section 3 reviews the GLISp framework upon which the proposed algorithm detailed in Section 4 is built. A brief description of the RFSC algorithm used to maximize the acquisition function is provided in Section 5. Section 6 reports the results in applying the proposed procedure to an illustrative example and to a case study dealing with predicting mortality in COVID-19 pneumonia. Some concluding remarks end the paper.

2. Problem statement

We here consider a multi-class classification problem, where a training set is given with T input-output pairs $\mathcal{D} = \{(\boldsymbol{x}(k), c(k))\}_{k=1}^{T}$, with $\boldsymbol{x}(k) \in \mathbb{R}^{N_f}$ denoting the kth input (or feature vector) and $c(k) = \{1, 2, \ldots, N_c\}$ the corresponding output label (or observed class).

In many classification problems, the size N_f of the feature vector $\boldsymbol{x}$ (*i.e.*, the number of features) can be very large, which makes the estimation of the full model awkward, since overparametrization and overfitting issues are likely to ensue, unless a prior selection of the features is carried out. The robustness and reliability of the model, *i.e.* the capability of generalizing the prediction performances to unseen observations, are greatly improved if the number of features is kept low, including in the model only a small subset of meaningful features. This has also an important practical consequence, given that the actual obtainment of the feature values is often not devoid of cost, as in the case, *e.g.*, of features associated to clinical tests that a patient has to undergo. Finally, the interpretability of the model is also increased by focusing on few features. For all these reasons, a FS procedure must be put in place, as discussed in the following.

A classifier $g_{\boldsymbol{s},\boldsymbol{\vartheta}} : \mathbb{R}^{N_f} \to \{1, 2, \ldots, N_c\}$, maps features to classes, $\boldsymbol{s} \in \mathcal{S} = \{0, 1\}^{N_f}$ coding its *structure*, such that $s_i = 1$ if the ith feature $\boldsymbol{x}_i$ enters the model and $s_i = 0$ otherwise, and $\boldsymbol{\vartheta} \in \Theta$ being a set of parameters. The classifier can be trained on the dataset $\mathcal{D}$ by minimizing a loss function $\mathcal{L} : \mathcal{S} \times \Theta \to \mathbb{R}$ (*e.g.*, minus the log-likelihood of the data). The minimal loss $\mathcal{L}$ achieved by a classifier with structure $\boldsymbol{s}$ can be thus computed as

$$\mathcal{J}(\boldsymbol{s}) = \min_{\boldsymbol{\vartheta} \in \Theta} \mathcal{L}(\boldsymbol{s}, \boldsymbol{\vartheta}) \tag{1}$$

and $\boldsymbol{\vartheta}^{\star}_{\boldsymbol{s}} = \arg\min_{\boldsymbol{\vartheta} \in \Theta} \mathcal{L}(\boldsymbol{s}, \boldsymbol{\vartheta})$ denotes the corresponding parametrization. Accordingly, we denote by $g_{\boldsymbol{s},\boldsymbol{\vartheta}^{\star}_{\boldsymbol{s}}}$ the classifier with structure $\boldsymbol{s}$ and corresponding optimal parameters $\boldsymbol{\vartheta}^{\star}_{\boldsymbol{s}}$.

We here investigate the possibility to include in the FS procedure the subjective criteria employed by a domain expert by taking into account his/her preferences, occasionally expressed over pairs of suggested classifiers. The intuition behind this idea is that the expert's preferences may convey useful and subjective information to allow the FS algorithm to balance model accuracy and other not formalized – but nonetheless important – requirements. This ultimately brings the algorithm to select those features (or combinations of features) that ensure high classification accuracy and at the same time are meaningful from a domain-oriented point of view.

Formally, we state the FS problem as follows:

$$\min_{\boldsymbol{s} \in \mathcal{S}} p(\boldsymbol{s}; g_{\boldsymbol{s},\boldsymbol{\vartheta}^{\star}_{\boldsymbol{s}}}), \tag{2a}$$

$$\text{s.t.} \quad f(\boldsymbol{s}; g_{\boldsymbol{s},\boldsymbol{\vartheta}^{\star}_{\boldsymbol{s}}}) \leq 0 \tag{2b}$$

where $p(\boldsymbol{s}; g_{\boldsymbol{s},\boldsymbol{\vartheta}^{\star}_{\boldsymbol{s}}}) : \mathcal{S} \to \mathbb{R}$ is an unknown cost function which depends on the classifier $g_{\boldsymbol{s},\boldsymbol{\vartheta}^{\star}_{\boldsymbol{s}}}$ and describes the subjective opinion of an external expert about the feature subset s and the corresponding classifier performance. Instead, we assume that the constraint function $f(\boldsymbol{s}; g_{\boldsymbol{s},\boldsymbol{\vartheta}^{\star}_{\boldsymbol{s}}})$ in (2b), with $f : \mathcal{S} \to \mathbb{R}^{n_c}$, captures possible quantifiable and objective properties that the classifier must satisfy, *e.g.*, minimum classification accuracy, specificity, sensitivity, or model size. However, we assume that f can be only evaluated after $g_{\boldsymbol{s},\boldsymbol{\vartheta}}$ has been trained. In the following, for ease of notation, we omit the dependence of the functions p and f on the classifier $g_{\boldsymbol{s},\boldsymbol{\vartheta}}$.

Since function p is not directly available to the FS procedure, to solve problem (2) the expert should in principle rate all the possible structures in $\mathcal{S}$. This is generally not affordable due to the large number 2^{N_f} of combinations, where the dimension N_f of $\mathcal{S}$ can be also large. Instead, we propose an iterative procedure to solve (2) where at each iteration the expert is asked to give some preferences between pairs of candidate model structures[1], as discussed in the following.

Given two candidate model structures $\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)} \in \mathcal{S}$, the *preference* function $\pi : \mathcal{S} \times \mathcal{S} \to \{-1, 0, 1\}$ expressed by the expert is defined as

$$\pi(\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}) = \begin{cases} -1 & \text{if } \boldsymbol{s}^{(1)} \text{ is "better" than } \boldsymbol{s}^{(2)} \\ 0 & \text{if } \boldsymbol{s}^{(1)} \text{ is "as good as" } \boldsymbol{s}^{(2)} \\ +1 & \text{if } \boldsymbol{s}^{(1)} \text{ is "worse" than } \boldsymbol{s}^{(2)} \end{cases}, \tag{3}$$

where for all $\boldsymbol{s}^{(j)}, \boldsymbol{s}^{(k)}, \boldsymbol{s}^{(l)} \in \mathcal{S}$ it holds that:

1. $\pi(\boldsymbol{s}^{(j)}, \boldsymbol{s}^{(j)}) = 0$,
2. $\pi(\boldsymbol{s}^{(j)}, \boldsymbol{s}^{(k)}) = -\pi(\boldsymbol{s}^{(k)}, \boldsymbol{s}^{(j)})$,
3. $\pi(\boldsymbol{s}^{(j)}, \boldsymbol{s}^{(k)}) = \pi(\boldsymbol{s}^{(k)}, \boldsymbol{s}^{(l)}) = -1 \Rightarrow \pi(\boldsymbol{s}^{(j)}, \boldsymbol{s}^{(l)}) = -1$ (transitive property).

Note that π is a black-box function that can be evaluated on pairs $(\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)})$ by querying the expert. In particular, we assume that the value $\pi(\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)})$ is returned by the user according to his or her underlying function p as follows:

1. $p(\boldsymbol{s}^{(j)}) < p(\boldsymbol{s}^{(k)}) \to \pi(\boldsymbol{s}^{(j)}, \boldsymbol{s}^{(k)}) = -1$,
2. $p(\boldsymbol{s}^{(j)}) = p(\boldsymbol{s}^{(k)}) \to \pi(\boldsymbol{s}^{(j)}, \boldsymbol{s}^{(k)}) = 0$,
3. $p(\boldsymbol{s}^{(j)}) > p(\boldsymbol{s}^{(k)}) \to \pi(\boldsymbol{s}^{(j)}, \boldsymbol{s}^{(k)}) = 1$,

for all $\boldsymbol{s}^{(j)}, \boldsymbol{s}^{(k)} \in \mathcal{S}$ satisfying constraint (2b). Without loss of generality, we assume that combinations $\boldsymbol{s} \in \mathcal{S}$ such that constraint $f(\boldsymbol{s}) > 0$ holds are implicitly always not preferred to feasible solutions. In other words:

$$\pi(\boldsymbol{s}^{(j)}, \boldsymbol{s}^{(k)}) = -1, \quad \forall \boldsymbol{s}^{(j)}, \boldsymbol{s}^{(k)} : f(\boldsymbol{s}^{(j)}) \leq 0, f(\boldsymbol{s}^{(k)}) > 0. \tag{4}$$

This allows us model the constraint $f(s) \leq 0$ directly as a known property of the cost function p, that is

$$f(\boldsymbol{s}^{(j)}) \leq 0, f(\boldsymbol{s}^{(k)}) > 0 \to p(\boldsymbol{s}^{(j)}) < p(\boldsymbol{s}^{(k)}), \forall \boldsymbol{s}^{(j)}, \boldsymbol{s}^{(k)} \in \mathcal{S} \tag{5}$$

To conclude, Problem (2) ultimately amounts to finding a model structure $\boldsymbol{s}_{\star}$ that is better (or at least not worse) than all other acceptable model structures, *i.e.* such that

$$\pi(\boldsymbol{s}_{\star}, \boldsymbol{s}) \leq 0, \quad \forall \boldsymbol{s} \in \mathcal{S}. \tag{6}$$

3. The GLISp framework

The preference-based FS method described in this paper follows the GLISp scheme [3], in that it iteratively suggests a sequence of model structures $\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(N)}$ to the user to test and compare. The user preferences gathered in the process are then exploited to collect information regarding the unknown cost function $p(\boldsymbol{s})$.

More precisely, at each iteration of the GLISp scheme, a *surrogate* preference function $\hat{p}(\boldsymbol{s}; g_{\boldsymbol{\vartheta}(\boldsymbol{s})}) : \mathcal{S} \to \mathbb{R}$ is trained to approximate the latent function $p(\boldsymbol{s}; g_{\boldsymbol{\vartheta}(\boldsymbol{s})})$. The set of observed pairwise preferences expressed by the user is taken into account by trying to preserve the relations: $\hat{p}(\boldsymbol{s}^{(1)}) < \hat{p}(\boldsymbol{s}^{(2)})$ if $\pi(\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}) = -1, \hat{p}(\boldsymbol{s}^{(1)}) > \hat{p}(\boldsymbol{s}^{(2)})$ if $\pi(\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}) = 1$, and $\hat{p}(\boldsymbol{s}^{(1)}) = \hat{p}(\boldsymbol{s}^{(2)})$ if $\pi(\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}) = 0$. The surrogate $\hat{p}$ is then used to build an *acquisition function* that is minimized to select the next point $s \in \mathcal{S}$ for evaluation, thus proposing a new comparison to the user between $g_{\boldsymbol{\vartheta}(\boldsymbol{s})}$ and the current best classifier. The acquisition function realizes a trade-off between *exploitation* (optimization only based on

[1] In the following, when we refer to a model structure $\boldsymbol{s}$, we also implicitly refer to the corresponding classifier $g_{\boldsymbol{s},\boldsymbol{\vartheta}^{\star}_{\boldsymbol{s}}}$, with parameters $\boldsymbol{\vartheta}^{\star}_{\boldsymbol{s}}$ obtained by solving (1).

the surrogate function describing the observed preferences) and *exploration* (searching unexplored areas of the solution domain $\mathcal{S}$). This iterative algorithm terminates when the maximum number of user queries is reached or when a satisfactory solution is obtained. Overall, the goal of GLISp is to approach the optimal solution $\boldsymbol{s}_\star$ within a small number of experiments, in order to minimize the number of expert interventions.

In the following subsections we discuss in detail how the surrogate and acquisition functions can be constructed in the context of FS.

3.1. Building the surrogate preference function

In this section we summarize the approach proposed in [3] to construct the surrogate preference function $\hat{p}$. Assume that we have trained $N \geq 2$ classifiers from the dataset $\mathcal{D}$ for N different model structures $\boldsymbol{s}^{(j)} \in \mathcal{S}$, $j = 1, \dots, N$. Assume also that the expert user has expressed M (with $1 \leq M \leq \binom{N}{2}$) pairwise preferences between model structures (evaluated based on the comparison of the respective classifiers), in the form:

$$b_h = \pi(\boldsymbol{s}^{(i(h))}, \boldsymbol{s}^{(j(h))}), \tag{7}$$

with $h = 1, \dots, M$, $i(h), j(h) \in \{1, \dots, N\}$, $i(h) \neq j(h)$. According to the definition of preference, it holds that $b_h \in \{-1, 0, 1\}$. The user preferences are collected in a *preference vector* $B = [b_1 \ \dots \ b_M]^T \in \{-1, 0, 1\}^M$, along with the compared structures indexed by $i(h), j(h)$, with $h = 1, \dots, M$.

Let us parameterize the surrogate function $\hat{p}$ to be estimated as the following linear combination of *Radial Basis Functions* (RBFs) [18,24]:

$$\hat{p}(\boldsymbol{s}) = \sum_{k=1}^{N} \beta_k \phi(\epsilon d(\boldsymbol{s}, \boldsymbol{s}^{(k)})), \tag{8}$$

where $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ is a distance measure between two model structures $\boldsymbol{s}^{(i)}$ and $\boldsymbol{s}^{(j)}$, such as the Euclidean distance

$$d(\boldsymbol{s}^{(i)}, \boldsymbol{s}^{(j)}) = \|\boldsymbol{s}^{(i)} - \boldsymbol{s}^{(j)}\|_2^2, \tag{9}$$

$\epsilon > 0$ is a scalar parameter, $\phi : \mathbb{R} \to \mathbb{R}$ is an RBF, and $\beta = [\beta_1 \ \dots \ \beta_N]^T$ are the unknown coefficients to be trained from the available preference vector B. Some examples of RBFs are the inverse quadratic $\phi(\epsilon d) = \frac{1}{1+(\epsilon d)^2}$, the Gaussian $\phi(\epsilon d) = e^{-(\epsilon d)^2}$, and the thin plate spline $\phi(\epsilon d) = (\epsilon d)^2 \log(\epsilon d)$ functions (see more examples in [2,18]).

Based on the observed preference vector B, the surrogate preference function $\hat{p}$ is constructed by imposing the constraints

$$\hat{p}(\boldsymbol{s}^{(i)}) \leq \hat{p}(\boldsymbol{s}^{(j)}) - \sigma + \varepsilon_h \quad \text{if } \pi(\boldsymbol{s}^{(i)}, \boldsymbol{s}^{(j)}) = -1 \tag{10a}$$

$$\hat{p}(\boldsymbol{s}^{(i)}) \geq \hat{p}(\boldsymbol{s}^{(j)}) + \sigma - \varepsilon_h \quad \text{if } \pi(\boldsymbol{s}^{(i)}, \boldsymbol{s}^{(j)}) = 1 \tag{10b}$$

$$|\hat{p}(\boldsymbol{s}^{(i)}) - \hat{p}(\boldsymbol{s}^{(j)})| \leq \sigma + \varepsilon_h \quad \text{if } \pi(\boldsymbol{s}^{(i)}, \boldsymbol{s}^{(j)}) = 0 \tag{10c}$$

for all $h = 1, \dots, M$, where $\sigma > 0$ is a given tolerance and $\varepsilon = [\varepsilon_1, \dots, \varepsilon_M]$ are positive slack variables. More specifically, the coefficients β in (8) are computed by solving the convex quadratic programming (QP) problem

$$\begin{aligned}
\min_{\beta,\varepsilon} \quad & \sum_{h=1}^{M} c_h \varepsilon_h + \frac{\lambda}{2} \sum_{k=1}^{N} \beta_k^2 \\
\text{s.t.} \quad & \sum_{k=1}^{N} (\phi(\epsilon d(\boldsymbol{s}^{(i(h))}, \boldsymbol{s}^{(k)}) - \phi(\epsilon d(\boldsymbol{s}^{(j(h))}, \boldsymbol{s}^{(k)}))\beta_k \\
& \leq -\sigma + \varepsilon_h, \qquad \forall h : \ b_h = -1 \\
& \sum_{k=1}^{N} (\phi(\epsilon d(\boldsymbol{s}^{(i(h))}, \boldsymbol{s}^{(k)}) - \phi(\epsilon d(\boldsymbol{s}^{(j(h))}, \boldsymbol{s}^{(k)}))\beta_k \\
& \geq \sigma - \varepsilon_h, \qquad \forall h : \ b_h = 1 \\
& \left| \sum_{k=1}^{N} (\phi(\epsilon d(\boldsymbol{s}^{(i(h))}, \boldsymbol{s}^{(k)}) - \phi(\epsilon d(\boldsymbol{s}^{(j(h))}, \boldsymbol{s}^{(k)}))\beta_k \right| \\
& \leq \sigma + \varepsilon_h, \qquad \forall h : \ b_h = 0 \\
& h = 1, \dots, M
\end{aligned} \tag{11}$$

where c_h are positive weights, *e.g.* $c_h = 1$, $\forall h = 1, \dots, M$, and $\lambda > 0$ is a regularization hyperparameter. Note that the slack variables ε_h in (10) and (11) are used to relax the constraints imposed by the preference vector B. Infeasibility of the constraints may be due to inconsistent assessments done by the user or to the poor flexibility of the basis functions used to parameterize $\hat{p}$.

Remark 1. As stated in [11], a potential risk in using human feedback is the confirmation bias. Indeed, in expressing preferences, experts may track the likelihood of a hypothesis, which could lead to bias if the experts only acknowledge evidence that is consistent with their existing beliefs. This calls for a proper balancing of the exploitation of the current available observations *vs.* the exploration when generating the new structures. The risk for confirmation bias will be considered in future work.

Finally, we remark that the computation of the surrogate function $\hat{p}$ requires to set the hyperparameter ϵ defining the shape of the RBFs ϕ in (8). The simplest way to choose ϵ is by K-fold cross-validation [28], by testing the capabilities of $\hat{p}$ in reconstructing the preferences in parts of the dataset not used to estimate $\hat{p}$.

3.2. Building the acquisition function

Once the surrogate function $\hat{p}$ is estimated, one could in principle minimize it to find the model structure (and corresponding classifier) that represents the best selection for the user, according to definition (6). More specifically, the following steps could be iteratively followed:

i) Propose a new model structure by minimizing $\hat{p}$, *i.e.*,

$$\boldsymbol{s}^{(N+1)} = \arg\min_{\boldsymbol{s} \in \mathcal{S}} \hat{p}(\boldsymbol{s}); \tag{12}$$

ii) Ask the user to express the preference $\pi(\boldsymbol{s}^{(N+1)}, \boldsymbol{s}_\star^{(N)})$, with $\boldsymbol{s}_\star^{(N)}$ being the best model structure found so far, corresponding to the smallest index $i_\star$ such that

$$\pi(\boldsymbol{s}^{(i_\star)}, \boldsymbol{s}^{(i)}) \leq 0, \ \ \forall i = 1, \dots, N; \tag{13}$$

iii) Update the estimate of $\hat{p}$ through (11).

Unfortunately, by *exploiting* only the current available observations in the model structure selection process, one may easily miss the global optimum $\boldsymbol{s}_\star$ in (6), as the proposed candidate solutions only rely on the available observations, leaving regions of the search space $\mathcal{S}$ unexplored. A term promoting the *exploration* of the space $\mathcal{S}$ should thus be considered, along with the surrogate $\hat{p}$, in proposing the next model structure $\boldsymbol{s}^{(N+1)}$. As proposed in [2,3], the exploration term is constructed based on the *inverse distance weighting* (IDW) function $z : \mathcal{S} \to [0,\ 1]$ defined as

$$z(\boldsymbol{s}) = \begin{cases} 0 & \text{if } \boldsymbol{s} \in \{\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(N)}\} \\ \frac{2}{\pi}\tan^{-1}\left(\frac{1}{\sum_{i=1}^{N} w_i(\boldsymbol{s})}\right) & \text{otherwise} \end{cases} \tag{14}$$

where $w_i(\boldsymbol{s}) = \frac{1}{d(\boldsymbol{s}, \boldsymbol{s}^{(i)})^2}$. In other words, $z(\boldsymbol{s}) = 0$ for all already tested structures, and $z(\boldsymbol{s}) > 0$ otherwise. The inverse tangent function in (14) prevents $z(\boldsymbol{s})$ from getting excessively large far away from all sampled points.

In the GLISp algorithm [3], an acquisition function is employed to balance exploitation *vs.* exploration when generating the new sample $\boldsymbol{s}^{(N+1)}$. Given an exploration hyperparameter $\delta \geq 0$, the *acquisition function* $a : \mathcal{S} \to \mathbb{R}$ is constructed as

$$a(\boldsymbol{s}) = \frac{\hat{p}(\boldsymbol{s})}{\Delta\hat{p}} - \delta z(\boldsymbol{s}), \tag{15}$$

where

$$\Delta\hat{p} = \max_i\{\hat{p}(\boldsymbol{s}^{(i)})\} - \min_i\{\hat{p}(\boldsymbol{s}^{(i)})\}$$

is the range of the surrogate function values on the samples $\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(N)}$ and acts as a normalization constant in (15) to simplify the choice of the exploration parameter δ. Note that $\Delta\hat{p} \geq \sigma$ (where σ is the tolerance introduced in (10)) if there is at least one comparison such that $b_h = \pi(\boldsymbol{s}^{(i(h))}, \boldsymbol{s}^{(j(h))}) \neq 0$.

Given a set $\{\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(N)}\}$ of samples and a vector B of preferences defined by (7), the next model structure $\boldsymbol{s}^{(N+1)}$ (and corresponding classifier) to propose to the user is computed as the solution of the following optimization problem with binary variables:

$$\boldsymbol{s}^{(N+1)} = \arg\min_{\boldsymbol{s} \in \mathcal{S}} a(\boldsymbol{s}). \tag{16}$$

In the acquisition function (15), the exploration term promotes sampling the space $\mathcal{S}$ in areas that have not been explored yet. Parameter δ balances the exploitation and exploration terms in $a(\boldsymbol{s})$. Setting $\delta = 0$ makes the GLISp algorithm rely only on the surrogate function $\hat{p}$ as in (12), whereas setting $\delta \gg 1$ makes it explore the entire input space regardless of the results of the comparisons.

We finally remark that, in executing the GLISp algorithm, a new candidate classifier $\boldsymbol{s}^{(N+1)}$ may not satisfy $f(\boldsymbol{s}^{(N+1)}) \leq 0$ as in (2b) after computing $g_{\boldsymbol{\vartheta}(\boldsymbol{s})}$. In this case, because of (4), there is no need to ask a preference to the user between the sample $\boldsymbol{s}^{(N+1)}$ proposed in (16) and the best model structure $\boldsymbol{s}_\star^{(N)}$ achieved up to iteration N. It is also possible that in the first iterations of the algorithm a comparison should be performed over two classifiers that both violate the constraint (2b). In this case, a remedy is to set π automatically so that the model structure with the highest classifier accuracy (or another quantitative performance metric) is preferred.

4. A preference-based feature selection algorithm

Algorithm 1 summarizes the steps required to compute the optimal structure $\boldsymbol{s}_\star$ and the associated classifier $g_{\boldsymbol{\vartheta}(\boldsymbol{s}_\star)}$, based on user preferences modeled using RBF interpolants (8) and the acquisition function (15). Throughout the algorithm, the classifier $g_{\boldsymbol{\vartheta}(\boldsymbol{s})}$ associated with a given $\boldsymbol{s}$ is computed by exploiting the linear programming based classification method proposed in [5]. Other classifiers, such as *Gaussian Process classifiers* or *Support Vector Machines* can be alternatively used.

In the initialization phase (cf. Algorithm 1, Step 1), N_{init} structures are generated randomly, possibly imposing *a priori* requirements on the resulting classifier such as *e.g.*, a desired minimum level of accuracy, sensitivity, or specificity.

The main cycle of Algorithm 1 consists of two main phases: generation and observation. During the generation phase (cf. Algorithm 1, Steps 5 – 18), which applies only for iterations $N \geq N_{init}$, an approximate solution to Problem (16) is generated as explained in Section 5.

Algorithm 1 Preference-based FS algorithm.

Input: Number $N_{init} \geq 2$ of initial structures, maximum number $N_{\max} \geq N_{init}$ of preference observations, hyper-parameters $\delta \geq 0$, $\sigma > 0$, $\epsilon > 0$, self-calibration index set $\mathcal{I}_{sc} \subseteq \{1, \ldots, N_{\max} - 1\}$.
Output: Optimal structure $\boldsymbol{s}_\star$.

```
1:  Generate $N_{init}$ random structures $\{\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(N_{init})}\}$;
2:  $N \leftarrow 1$, $i^\star \leftarrow 1$, CONTINUE $\leftarrow$ True; OBSERVE $\leftarrow$ True;
3:  while $N < N_{\max}$ and CONTINUE do
4:      if $N \geq N_{init}$ then
5:          if $N \in \mathcal{I}_{sc}$ then
6:              Recalibrate $\epsilon$ by K-fold cross-validation;
7:          end if
8:          Solve optimization problem (11) and get $\beta$;
9:          Solve optimization problem (16) and get $\boldsymbol{s}^{(N+1)}$ (Algorithm 2);
10:         if $\boldsymbol{s}^{(N+1)} \in \{\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(N)}\}$ then
11:             $(\boldsymbol{s}^{(N+1)}, \texttt{CONTINUE}) \leftarrow \texttt{flip}(\boldsymbol{s}^{(N+1)}, \{\boldsymbol{s}^{(k)}\}_{k=1}^{N})$;
12:         end if
13:         Compute classifier associated to $\boldsymbol{s}^{(N+1)}$;
14:         if $f(\boldsymbol{s}^{(N+1)}) \leq 0$ then
15:             OBSERVE $\leftarrow$ True;
16:         else
17:             OBSERVE $\leftarrow$ False;
18:         end if
19:     end if
20:     $i(N) \leftarrow i^\star$, $j(N) \leftarrow N+1$;
21:     if OBSERVE then
22:         Observe preference $b_N = \pi(\boldsymbol{s}^{(i(N))}, \boldsymbol{s}^{(j(N))})$;
23:         if $b_N = 1$ then
24:             $i^\star \leftarrow j(N)$;
25:         end if
26:     end if
27:     $N \leftarrow N+1$;
28: end while
```

Once the candidate solution $\boldsymbol{s}^{(N+1)}$ has been retrieved, it undergoes a test (cf. Algorithm 1, Step 11) to establish whether it has been already explored (to avoid unnecessary queries to the expert). This may happen due to the randomized nature of the RFSC and the discrete nature of $\mathcal{S}$. If $\boldsymbol{s}^{(N+1)} \in \{\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(N)}\}$, we perturb the solution as follows. All the model structures at a Hamming distance of 1 from $\boldsymbol{s}^{(N+1)}$ are sorted according to their surrogate function values, and the best unexplored solution is returned. If all the new generated structures have been already explored, we exit from Algorithm 1. The generation phase concludes by verifying whether or not the generated candidate solution $\boldsymbol{s}^{(N+1)}$ satisfies the constraint $f(\boldsymbol{s}) \leq 0$ in (2b) (cf. Algorithm 1, Step 14). In the affirmative case, the algorithm proceeds with the preference observation phase (cf. Algorithm 1, Step 22). In this phase, the expert is asked to provide a pairwise preference between the sample $\boldsymbol{s}^{(N+1)}$ proposed in (16) and the best model structure $\boldsymbol{s}_\star^{(N)}$ achieved up to iteration N. If the proposed candidate solution is preferred, the best model structure is updated accordingly (cf. Algorithm 1, Step 23).

5. The generation phase

In the generation phase the algorithm selects a new structure $\boldsymbol{s}^{(N+1)}$ to be proposed to the expert for comparison with the best one obtained so far $\boldsymbol{s}_\star^{(N)}$. The new structure is obtained by minimizing the acquisition function $a(\boldsymbol{s})$. To solve this combinatorial problem over the space of structures $\mathcal{S}$, we apply the RFSC algorithm [9]. The RFSC employs a probabilistic reformulation of the optimization problem, by introducing the random variable ϕ which takes values in $\mathcal{S}$ according to a probability distribution $\mathcal{P}_\phi$. The performance of ϕ is also a random variable, and its expectation is given by

$$\mathbb{E}[\mathcal{J}(\phi)] = \sum_{\boldsymbol{s}\in\mathcal{S}} \mathcal{J}(\boldsymbol{s})\mathcal{P}_\phi(\boldsymbol{s}), \tag{17}$$

where $\mathcal{J}(\boldsymbol{s}) = e^{-K\cdot a(\boldsymbol{s})}$, so that performance is graded from 0 to 1. Index (17) is maximized when the probability mass concentrates on a feature subset with minimum value of a. Accordingly, the optimization problem can be solved by searching for the limit distribution

$$\mathcal{P}_\phi^* = \arg\min_{\mathcal{P}_\phi} \mathbb{E}[\mathcal{J}(\phi)]. \tag{18}$$

To address this problem in practice, $\mathcal{P}_\phi$ is parameterized by associating a Bernoulli random variable ρ_j to each feature $\boldsymbol{x}_j$, that models the belief that $\boldsymbol{x}_j$ belongs to the target feature subset:

$$\rho_j \sim Be(\mu_j), \tag{19}$$

$j = 1,\dots,N_f$, where $\mu_j \in [0,1]$ is the success probability. A feature subset can then be extracted from this distribution, by extracting a value from the Bernoullian distribution associated to each feature $\boldsymbol{x}_j$, $j = 1,\dots,N_f$, and including the latter in the feature subset if the outcome is 1. This event has probability μ_j, which is accordingly denoted *Feature Inclusion Probability* (FIP) of the jth feature. For simplicity, all random variables ρ_j, $j = 1,\dots,N_r$ are assumed independent. The probability of a feature subset $\boldsymbol{s}\in\mathcal{S}$ can be expressed as

$$\mathcal{P}_\phi(\boldsymbol{s}) = \prod_{j:\boldsymbol{s}_j=1} \mu_j \prod_{j:\boldsymbol{s}_j=0} (1-\mu_j). \tag{20}$$

The RFSC operates by adapting the FIPs until convergence to a target limit distribution (*i.e.*, such that all FIPs are valued 0 or 1, which corresponds to assigning probability 1 to a specific feature subset). The adaptation of $\mathcal{P}_\phi$ is carried out by repeating the following tasks at each iteration: a) extract a set of feature subsets, b) evaluate the corresponding values of the acquisition function, c) estimate the importance of each feature, d) update the FIP of each feature. The importance of a feature $\boldsymbol{x}_j$ is calculated by means of an aggregate indicator $\mathcal{I}_j$ that compares the average performance of the feature subsets including the said feature with that of the remaining ones:

$$\mathcal{I}_j = \mathbb{E}[\mathcal{J}(\phi)|\phi_j = 1] - \mathbb{E}[\mathcal{J}(\phi)|\phi_j = 0], \tag{21}$$

where $j = 1,\dots,N_f$. Indicator $\mathcal{I}_j$ averages over all structures in $\mathcal{S}$ and can therefore be considered a global measure of the regressor importance. In task (c) of the main loop of the algorithm, $\mathcal{I}_j$ is estimated based on the sampled feature subsets. Then, in task (d), the FIPs are updated as follows

$$\mu_j(t+1) = \mu_j(t) + \chi\hat{\mathcal{I}}_j \tag{22}$$

for $j = 1,\cdots,N_f$, where $\hat{\mathcal{I}}_j$ is the sampled estimate of $\mathcal{I}_j$ and χ is a gain factor (or step size). The value of χ balances algorithm speed and robustness, and reflects the reliability that the user can assume on the sampled estimate of the importance indicator.

The structure generation procedure is summarized in Algorithm 2. We address the reader to [9] for all technical details of the algorithm and for a comprehensive numerical analysis on several numerical data sets from the UCI machine learning repository, [25]. Notice that the original version of the method includes a feature pre-processing step that is here omitted.

Algorithm 2 Randomized algorithm for structure generation.

Input: Number N_p of structures to extract at each iteration, number N_f of features, initial Bernoullian success probabilities $\boldsymbol{\mu}$, probability saturation values $\mu_{\min}$ and $\mu_{\max}$, scaling factor K, RBF coefficients β, scalar parameter ϵ.
Output: Proposed structure $\boldsymbol{s}$.

```
repeat
    for $p = 1$ to $N_p$ do
        Extract non-empty structure $\boldsymbol{s}^{(p)}$ from Bernoullian($\boldsymbol{\mu}$);
        Evaluate the surrogate function $\hat{p}(\boldsymbol{s}^{(p)})$ as in (8);
        Define acquisition function $a(\boldsymbol{s}^{(p)})$ as in (15);
        $\mathcal{J}^{(p)} \leftarrow e^{-K\cdot a(\boldsymbol{s}^{(p)})}$;
    end for
    for $j = 1$ to $n$ do
        $\mathcal{J}^{\oplus} \leftarrow 0$; $n^{\oplus} \leftarrow 0$; $\mathcal{J}^{\ominus} \leftarrow 0$; $n^{\ominus} \leftarrow 0$;
        for $p = 1$ to $N_p$ do
            if $s_j^{(p)} = 1$ then
                $\mathcal{J}^{\oplus} \leftarrow \mathcal{J}^{\oplus} + \mathcal{J}^{(p)}$; $n^{\oplus} \leftarrow n^{\oplus} + 1$;
            else
                $\mathcal{J}^{\ominus} \leftarrow \mathcal{J}^{\ominus} + \mathcal{J}^{(p)}$; $n^{\ominus} \leftarrow n^{\ominus} + 1$;
            end if
        end for
        $\chi \leftarrow \frac{1}{10(\mathcal{J}_{\text{best}} - \mathcal{J}_{\text{mean}}) + 0.1}$;
        $\mu_j \leftarrow \mu_j + \chi\left(\frac{\mathcal{J}^{\oplus}}{\max(n^{\oplus},1)} - \frac{\mathcal{J}^{\ominus}}{\max(n^{\ominus},1)}\right)$;
        $\mu_j \leftarrow \max\left(\min\left(\mu_j, \mu_{\max}\right), \mu_{\min}\right)$;
    end for
until Stopping criterion
$\boldsymbol{s} \leftarrow \texttt{round}(\boldsymbol{\mu})$;
```

6. Examples

6.1. Illustrative example

We first show the performance of the proposed scheme in optimizing an (unknown) numerical cost function through user preferences. The expert employs the following *latent* cost function to rate model structures:

$$p(\boldsymbol{s}) = \|\boldsymbol{s} - \boldsymbol{s}^\circ\|_1^2 + \mathcal{P}(\boldsymbol{s}), \tag{23}$$

where

$$s_i^\circ = \begin{cases} 0, & i \in \{3,\dots,10\}, \\ 1, & i \in \{1,2,11,12,\dots,20\}, \end{cases}$$

and $\mathcal{P}(\boldsymbol{s}) = 100\|\boldsymbol{s}_{\{3,\dots,10\}}\|_1$. In other words, the expert's subjective criterion penalizes structures different from $\boldsymbol{s}^\circ$. However, the FS algorithm gets this information only indirectly and partially, by way of pairwise comparisons between structures.

Algorithm 1 has been applied to this FS problem, assuming that the expert preferences are given according to the latent cost p. The parameter settings for Algorithms 1 and 2 are reported in Table 1.

Fig. 1 shows the value of the latent function p and of the surrogate $\hat{p}$ as a function of the number of queried preferences (number of iterations). Apparently, the constructed surrogate function $\hat{p}$ is capable of driving the algorithm toward the global minimum (represented by the dashed red line) which is reached after 40 queried preferences (excluding the N_{init} ones), despite the fact that p and $\hat{p}$ have very different shapes. Indeed, $\hat{p}$ has been constructed only to honor the preference constraints (10) given by the user, which account for the relative relationships (in terms of the preference function) of a small number of model structure pairs.

Fig. 2 provides a full pictorial representation of the pairwise preferences among the structures proposed at each iteration. The

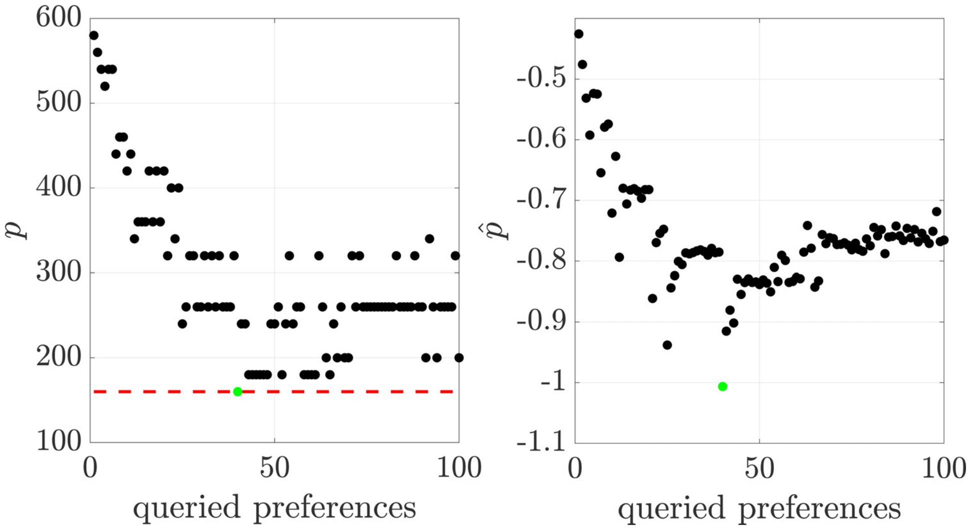

Fig. 1. Value of the latent function p and of the surrogate $\hat{p}$ *vs* number of queried preferences (number of iterations). The green marker denotes the optimizer found by the proposed algorithm. The red dashed line indicates the true optimal cost $p_\star$.

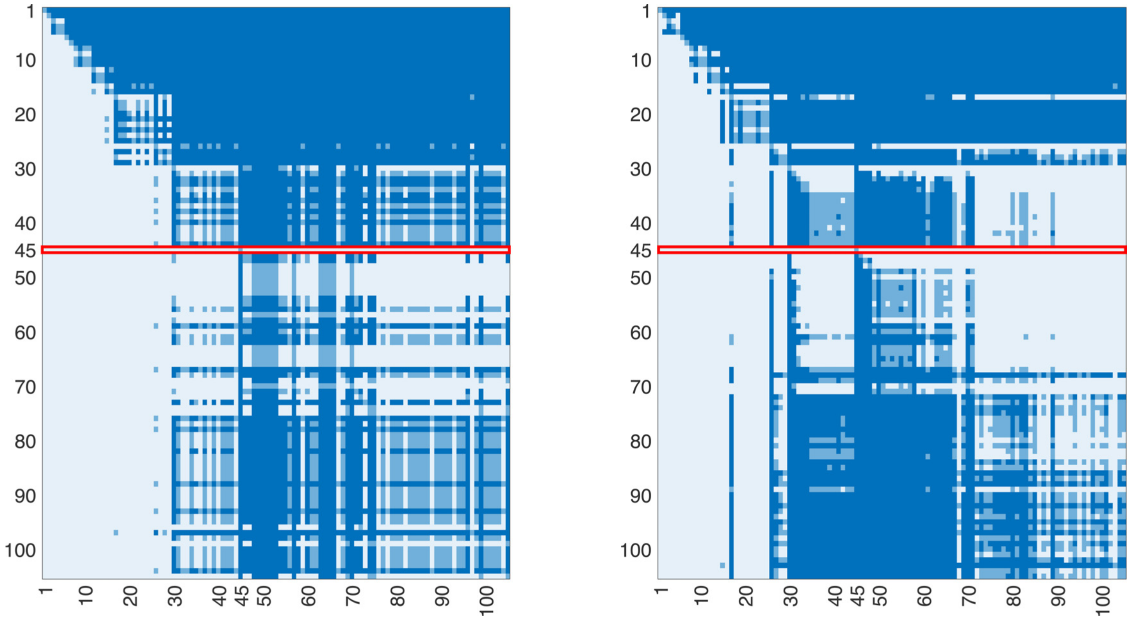

Fig. 2. Preferences (green: 0, blue: 1, white: −1) based on the latent function p (left) and its constructed surrogate $\hat{p}$ (right). Optimal structure $\boldsymbol{s}_\star$ at row 45.

Table 1
Example 1: parameter setting for Algorithm 1 and 2.

Param	Value
N_{init}	5
$N_{\max}$	100
$\mathcal{I}_{sc}$	$\{10, 20, \ldots, 90\}$
δ	1
σ	0.001
ϵ	1
N_p	100
μ_j	0.5
$\mu_{\min}$	0.001
$\mu_{\max}$	0.999
K	1

left picture shows the preferences calculated according to the latent function p, while the right picture considers the surrogate function $\hat{p}$. It is evident that the constructed surrogate correctly reconstructs the preferences between the identified optimal structure $\boldsymbol{s}_\star$ (row 45) and all the other explored structures, thus fulfilling its purpose. Finally, note from Fig. 1 that, as expected, the algorithm proceeds in exploring the solution space $\mathcal{S}$ until the maximum number $N_{\max}$ of preference observations is reached. In this example, the flip routine has never been executed.

6.2. Case study - Predicting mortality in COVID-19 pneumonia

To apply the proposed FS algorithm in a real-world context, we consider the problem of training a classifier to predict 30-day mortality in patients with COVID-19 pneumonia. We remark that this case study is reported only for illustrative purposes and it only aims at showing the effectiveness and potential of the proposed FS approach in a clinical application using real data and experts (clinicians). Extensive validation is not performed for this case study, and thus the models presented in this paper should not be used by clinicians to fight against COVID-19 infections. The interested reader is referred to [15,19,31] and the references therein for studies on data-driven development of mortality predictors in COVID-19 pneumonia.

Table 2
Case study: parameter setting for Algorithm 1 and 2. Within brackets the values used in Section 6.2.4, if changed.

Param	Value
N_{init}	5
$N_{\max}$	150 (60)
$\mathcal{I}_{sc}$	$\{10, 20, \ldots, 140\}$ ($\{1, 2, \ldots, 59\}$)
δ	1
σ	0.0001
ϵ	1
N_p	100
μ_j	0.016
$\mu_{\min}$	0.001
$\mu_{\max}$	0.999
K	1

6.2.1. Dataset

The dataset consists of 704 patients diagnosed with COVID-19 pneumonia admitted from February to November 2020 to the Guglielmo da Saliceto Hospital, Piacenza, in northern Italy. Among the considered patients, 438 (62%) were discharged, while the remaining 266 (38%) deceased. Data characterizing the patients includes demographic information, comorbidities, laboratory tests, symptoms and blood examinations at hospital admission, etc., for a total of 64 features (see Table 4 for the complete list). Continuous features are normalized in the $[0, 1]$ range, and a nearest-neighbour method is used to fill in missing data. The overall patient data set is randomly split into training (599) and test (105) sets.

6.2.2. Fictitious quantitative cost function

For the sake of illustration, we first test the algorithm by defining preferences based on a fictitious quantitative cost function, to verify that it is able to reach good solutions although it ignores the cost function employed by the expert and only employs the coarse information provided by the given preferences. To this aim, we employ the following multi-objective fictitious cost to rate classifiers:

$$p(\boldsymbol{s}; g_{\vartheta(\boldsymbol{s})}) = (1 - \mathtt{acc}(g_{\vartheta(\boldsymbol{s})})) + \frac{\|\boldsymbol{s}\|_1^2}{64} + f_{sens}(g_{\vartheta(\boldsymbol{s})}) + f_{spec}(g_{\vartheta(\boldsymbol{s})}),$$

where $\mathtt{acc}(g_{\vartheta(\boldsymbol{s})})$, $\mathtt{sens}(g_{\vartheta(\boldsymbol{s})})$, and $\mathtt{spec}(g_{\vartheta(\boldsymbol{s})})$ measure respectively the accuracy, sensitivity and specificity of the classifier $g_{\vartheta(\boldsymbol{s})}$, and

$$f_{sens}(g_{\vartheta(\boldsymbol{s})}) = \begin{cases} \exp(-10(\mathtt{sens}(g_{\vartheta(\boldsymbol{s})}) - 0.6)), & \text{if } \mathtt{sens}(g_{\vartheta(\boldsymbol{s})}) \geq 0.6 \\ (1 + 10|\mathtt{sens}(g_{\vartheta(\boldsymbol{s})}) - 0.6|), & \text{otherwise} \end{cases}$$

$$f_{spec}(g_{\vartheta(\boldsymbol{s})}) = \begin{cases} \exp(-10(\mathtt{spec}(g_{\vartheta(\boldsymbol{s})}) - 0.85)), & \text{if } \mathtt{spec}(g_{\vartheta(\boldsymbol{s})}) \geq 0.85 \\ (1 + 10|\mathtt{spec}(g_{\vartheta(\boldsymbol{s})}) - 0.85|), & \text{otherwise} \end{cases}$$

The rationale behind the designed cost function $p(\boldsymbol{s}; g_{\vartheta(\boldsymbol{s})})$ is to maximize the overall classifier performance and to comply with a desired minimum level of sensitivity, *i.e.*, 60%, and specificity, *i.e.*, 85%. Classifier complexity is penalized as well. The parameter settings for Algorithms 1 and 2 are reported in Table 2.

Fig. 3 shows the designed cost function $p(\boldsymbol{s}; g_{\vartheta(\boldsymbol{s})})$ and its contribution as a function of the number of queried preferences (number of iterations) for an execution of the proposed algorithm. Apparently, the designed cost function fulfills its purposes, as the preferences defined based on it suffice to steer the algorithm towards a parsimonious solution (9 features) with the desired overall performance. As can be observed in the figure, the optimal solution is found after 60 iterations.

6.2.3. Effects of feature correlation

Table 3 reports the classifier performance assessed in terms of model accuracy, specificity, sensitivity, and the selected features obtained by executing three times the presented algorithm with the designed cost function $p(\boldsymbol{s}; g_{\vartheta(\boldsymbol{s})})$. In this way, three different classifiers, denoted as $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ are obtained. Although the three classifiers show similar performance, their structure is extremely different. This is due to the fact that many features are correlated (see Fig. 4), which implies that multiple equivalent classifiers can be obtained. Nonetheless, the obtained classifiers $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$ are not equivalent from a clinical point of view, as here discussed.

The *PaO$_2$-to-FiO$_2$ ratio* is known to be one of the most important predictor of mortality in COVID-19 pneumonia, and is present in all three models. *Creatinine* is present in the first and second classifier, and not in the third one. Nevertheless, the third classifier comprises the *urea* level, which is strongly correlated with the creatinine, as also observed experimentally, with a linear correlation coefficient equal to 0.72. It is worth remarking that a third of patients with severe COVID-19 pneumonia presents also an acute kidney injury [26], and thus creatinine/urea turns out to be strongly related to the worse outcome in these patients. However, high creatinine/urea levels is consistently found not only in COVID-19 pneumonia, but also in other diseases that compromise kidney function.

Looking at the single classifiers, all features involved in $\mathcal{C}_1$ can be easily collected at the hospital admission. The only exception is the *respiratory rate*, which is mostly measured manually and it is believed to waste valuable time for clinicians, especially in emergency settings. Furthermore, although the accuracy in the measurements of respiratory rate by healthcare professionals has been reported to be fairly high, minor changes in this variable may have an important effect in risk assessment in critically ill COVID-19 patients [21]. Overall, classifier $\mathcal{C}_1$ shows good performance, but lacks inflammatory parameters (such as the *neutrophil-to-lymphocyte ratio* and the *C-reactive protein*) which are the most predictive laboratory variables in COVID-19 pneumonia.

Classifier $\mathcal{C}_2$, is the simplest in terms of required predictors, but it provides information only on kidney and pulmonary functions. Furthermore, it comprises the *sodium* level as a predictor, whose correlation with prognosis in COVID-19 is still a matter of debate.

The last classifier $\mathcal{C}_3$ is the most complete and informative from a physician perspective, as it comprises clinical and laboratory parameters, such as: *PaO$_2$-to-FiO$_2$ ratio*; *symptom duration*; and *neutrophil-to-lymphocyte ratio*. Notably, $\mathcal{C}_3$ is the only classifier considering the *age* as a feature. However, it comprises the *PaO$_2$* (*i.e.*, partial pressure of oxygen dissolved in plasma), that is redundant with respect to the *PaO$_2$-to-FiO$_2$ ratio.* Overall, the two clinicians involved in this study (Dr. Geza Halasz and Dr. Matteo Villani), agree to consider $\mathcal{C}_3$ as the most valuable classifier among the three in predicting COVID-19 mortality.

The above discussion highlights the differences among the three classifiers, and shows the importance of involving experts to drive the construction of prognostic models for clinical practices.

6.2.4. Clinician-in-the-loop decision making

The clinician Dr. Geza Halasz was asked to act as expert in the application of the proposed preference-based algorithm.

To start the experiment, five initial classifiers are randomly generated with the following constraints: maximum number of features equal to 15; accuracy larger than 0.7, sensitivity and specificity larger than 0.5. An initial comparison between these five classifiers is then performed, as detailed in Algorithm 1.

New models are then iteratively proposed according to Algorithm 1. To avoid unnecessary queries to the expert, only models with an accuracy $\mathtt{acc}(g_{\vartheta(\boldsymbol{s})})$ higher than 0.7 are proposed for

Table 4
Complete list of patients' characteristics available in the COVID-19 dataset.

Number	Feature	Number	Feature
1	glucose level	33	prothrombin time
2	urea level	34	prothrombin activity percentage
3	creatinine level	35	prothrombin time - INR
4	sodium level	36	partial thromboplastin time
5	potassium level	37	activated partial thromboplastin time
6	chloride level	38	C-reactive protein
7	conjugated total	39	age
8	conjugated bilirubin	40	gender
9	aspartate aminotransferase	41	systolic blood pressure
10	alanine aminotransferase	42	heart rate
11	lactate dehydrogenase	43	oxygen saturation
12	creatine kinase	44	respiratory rate
13	amilase	45	temperature
14	lipase	46	PaO_2-to-FiO_2 ratio
15	cholinesterase	47	symptoms
16	white blood cells count	48	hypertension
17	red blood cells count	49	atrial fibrillation
18	haemoglobin	50	chronic obstructive pulmonary disease
19	hematocrit	51	dislypidemia
20	mean corpuscular volume	52	chronic kidney disease
21	mean hemoglobin concentration	53	diabetes
22	mean corpuscular hemoglobin concentration	54	malignancy (active or previously treated)
23	platelets count	55	previous stroke
24	red cell distribution width	56	peripheral artery disease
25	neutrophils percentage	57	comorbidities
26	lymphocytes percentage	58	neutrophil-to-lymphocyte ratio
27	monocytes percentage	59	coronary artery disease
28	eosinophil percentage	60	arterial pH
29	lymphocytes count	61	PaO_2
30	monocytes count	62	$PaCO_2$
31	eosinophil count	63	HCO_3
32	neutrophils count	64	glasgow coma scale

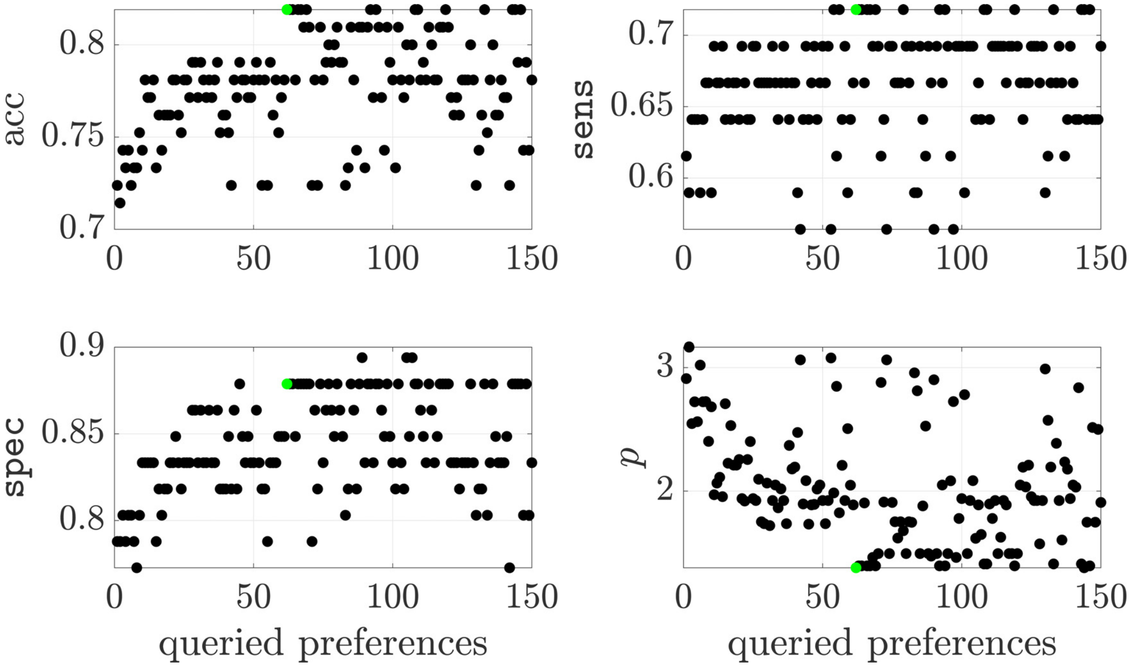

Fig. 3. Cost function $p(\boldsymbol{s}; g_{\vartheta(\boldsymbol{s})})$ and its contributions as a function of the number of queried preferences (number of iterations). The green marker denotes the found optimal solution.

comparison. Classifiers not satisfying this constraint are thus automatically "labelled" as worse than the previous best classifier. Besides quantitative metrics such as accuracy, sensitivity, specificity, and required features, the clinician implicitly considered the following criteria in expressing his preference: clinical interpretability of the model; cost and difficulty in obtaining the features; presence of variables typically associated with mortality in COVID-19 pneumonia. For instance, at iteration $N = 12$ the comparison between the following two models is proposed:

- best model $g_{\vartheta(\boldsymbol{s}_\star^{(N)})}$ achieved up to iteration $N = 12$: $\mathtt{acc}(g_{\vartheta(\boldsymbol{s}^{(N)})}) = 77.14\%$; $\mathtt{spec}(g_{\vartheta(\boldsymbol{s}^{(N)})}) = 64.10\%$; $\mathtt{sens}(g_{\vartheta(\boldsymbol{s}^{(N)})}) = 84.85\%$; features = { *neutrophil-to-lymphocyte ratio*, *white blood cells count*, *monocytes percentage*, *monocytes count*, *prothrombin*

Table 3
Performance and selected features of three different classifiers.

	Accuracy [%]	Sensitivity [%]	Specificity [%]	Selected features
$\mathcal{C}_1$	81.91	71.80	87.88	creatinine level cholinesterase haemoglobin red cell distribution width monocytes percentage prothrombin activity percentage respiratory rate PaO_2-to-FiO_2 ratio
$\mathcal{C}_2$	80.95	64.10	90.91	creatinine level sodium level oxygen saturation PaO_2-to-FiO_2 ratio
$\mathcal{C}_3$	80.00	66.667	87.88	urea level neutrophils count age PaO_2-to-FiO_2 ratio symptoms peripheral artery disease neutrophil-to-lymphocyte ratio PaO_2

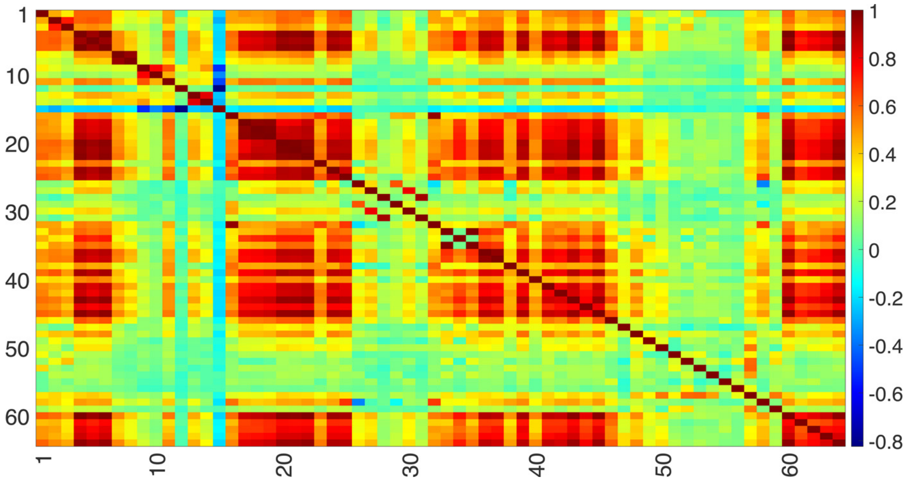

Fig. 4. Linear correlation coefficients between pairs of features: x and y axis represent the feature number.

time, *age*, *PaO_2-to-FiO_2 ratio*, *chronic obstructive pulmonary disease*, *chronic kidney disease*, *coronary artery disease*}.

- new candidate model $g_{\boldsymbol{\vartheta}(\boldsymbol{s}^{(N)})}$: $\mathtt{acc}(g_{\boldsymbol{\vartheta}(\boldsymbol{s}^{(N)})}) = 76.20\%$; $\mathtt{spec}(g_{\boldsymbol{\vartheta}(\boldsymbol{s}^{(N)})}) = 58.98\%$; $\mathtt{sens}(g_{\boldsymbol{\vartheta}(\boldsymbol{s}^{(N)})}) = 86.36\%$; features = { *neutrophil-to-lymphocyte ratio*, *white blood cells count*, *monocytes percentage*, *eosinophil count*, *prothrombin time*, *age*, *PaO_2-to-FiO_2 ratio*, *chronic obstructive pulmonary disease*, *chronic kidney disease*}.

Although accuracy and sensitivity of the new candidate model are lower than the best model proposed so far, the former is preferred since it involves less features and also includes the *eosinophil count* which, according to the literature, is strongly related to mortality in COVID-19 pneumonia.

It is interesting to discuss also the comparison proposed at the next iteration (i.e, $N = 13$), where the best model is the one just reported, while the new candidate model has the following characteristics: $\mathtt{acc}(g_{\boldsymbol{\vartheta}(\boldsymbol{s}^{(N)})}) = 74.28\%$; $\mathtt{spec}(g_{\boldsymbol{\vartheta}(\boldsymbol{s}^{(N)})}) = 56.41\%$; $\mathtt{sens}(g_{\boldsymbol{\vartheta}(\boldsymbol{s}^{(N)})}) = 84.85\%$; features = { *neutrophil-to-lymphocyte ratio*, *monocytes percentage*, *prothrombin time*, *chronic obstructive pulmonary disease*, *chronic kidney disease*}. The two models are different in terms of selected features, and the first one outperforms the second one. However, the latter has a similar clinical interpretability, although it contains less variables. In this case, the clinician defines the two models as “comparable”.

At iteration $N = 20$, the model with the following characteristics is proposed and selected: $\mathtt{acc}(g_{\boldsymbol{\vartheta}(\boldsymbol{s}^{(N)})}) = 79.05\%$; $\mathtt{spec}(g_{\boldsymbol{\vartheta}(\boldsymbol{s}^{(N)})}) = 64.10\%$; $\mathtt{sens}(g_{\boldsymbol{\vartheta}(\boldsymbol{s}^{(N)})}) = 87.88\%$; features = { *neutrophil-to-lymphocyte ratio*, *white blood cells count*, *prothrombin time*, *age*, *PaO_2-to-FiO_2 ratio*, *symptoms*, *chronic obstructive pulmonary disease*, *chronic kidney disease*}. The procedure keeps going until $N_{\max} = 60$ iterations, but no better models are selected. This model contains a “reasonable” number of variables, which turns out to be quite informative from a clinical perspective. In fact, this model includes: laboratory parameters; symptom duration before hospital admission; clinical variables as the *PaO_2-to-FiO_2 ratio*; and coexisting pathological conditions. For the above reasons, both clinicians involved in this study agree that this model model is better than the three ones discussed in Section 6.2.3 and reported in Table 3.

7. Conclusion

A novel algorithm for active preference-based FS in classification problems has been discussed. It relies on a suitable formulation of the FS problem based on the optimization of a latent cost function describing the subjective opinion of an external expert about the selected feature subset and about the classifier performance. Since this term is not directly available to the algorithm, a surrogate of it is iteratively trained based on binary preferences expressed by the expert on pairs of candidate feature subsets. The proposed method has been tested on both synthetic and experimental FS problems, proving its effectiveness in selecting the relevant features. Notably, the potentiality of the proposed approach has been validated by two clinicians involved in the study dealing with predicting mortality in COVID-19 pneumonia. The preliminary experimental results are promising, in that a parsimonious and accurate solution is obtained after a relatively short exploration phase. Future research will focus on deriving alternative parameterizations of the surrogate function, as well as addressing the confirmation bias issue.

Declaration of Competing Interest

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

References

[1] M. Abdolshah, A. Shilton, S. Rana, S. Gupta, S. Venkatesh, Multi-objective Bayesian optimisation with preferences over objectives, arXiv:1902.04228 (2019).
[2] A. Bemporad, Global optimization via inverse distance weighting and radial basis functions, Computational Optimization and Applications 77 (2020) 571–595. Code available at http://cse.lab.imtlucca.it/bemporad/glis
[3] A. Bemporad, D. Piga, Global optimization based on active preference learning with radial basis functions, Machine Learning 110 (2021) 417–448.
[4] A. Benavoli, D. Azzimonti, D. Piga, Preferential Bayesian Optimisation with Skew Gaussian Processes, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, in: GECCO '21, 2021, pp. 1842–1850.
[5] K.P. Bennett, O.L. Mangasarian, Multicategory discrimination via linear programming, Optimization methods and Software 3 (1994) 27–39.
[6] F. Bianchi, V. Breschi, D. Piga, L. Piroddi, Model structure selection for switched NARX system identification: a randomized approach, Automatica 125 (2021) 109415.
[7] F. Bianchi, A. Falsone, M. Prandini, L. Piroddi, A randomised approach for NARX model identification based on a multivariate bernoulli distribution, International Journal of Systems Science 48 (6) (2017) 1203–1216.
[8] F. Bianchi, M. Prandini, L. Piroddi, A randomized two-stage iterative method for switched nonlinear systems identification, Nonlinear Analysis: Hybrid Systems 35 (2020) 100818.
[9] A. Brankovic, A. Falsone, M. Prandini, L. Piroddi, A feature selection and classification algorithm based on randomized extraction of model populations, IEEE Transactions on Cybernetics 48 (4) (2018) 1151–1162.
[10] E. Brochu, N. de Freitas, A. Ghosh, Active preference learning with discrete choice data, in: Advances in neural information processing systems, 2008, pp. 409–416.
[11] C.J. Cai, E. Reif, N. Hegde, J. Hipp, B. Kim, D. Smilkov, M. Wattenberg, F. Viegas, G.S. Corrado, M.C. Stumpe, et al., Human-centered tools for coping with imperfect algorithms during medical decision-making, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019, pp. 1–14.
[12] B. Chau, N. Kolling, L. Hunt, M. Walton, M. Rushworth, A neural mechanism underlying failure of optimal choice with multiple alternatives, Nature neuroscience 17 (3) (2014) 463.
[13] A.H. Correia, F. Lecue, Human-in-the-loop feature selection, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 2438–2445.
[14] F. Drobnič, A. Kos, M. Pustišek, On the interpretability of machine learning models and experimental feature selection in case of multicollinear data, Electronics 9 (5) (2020) 761.
[15] H. Estiri, Z. Strasser, J. Klann, P. Naseri, K. Wagholikar, S. Murphy, Predicting COVID-19 mortality with electronic medical records, NPJ digital medicine 4 (1) (2021) 1–10.
[16] A. Falsone, L. Piroddi, M. Prandini, A randomized algorithm for nonlinear model structure selection, Automatica 60 (2015) 227–238.
[17] A. González, Z. Dai, A. Damianou, N.D. Lawrence, Preferential Bayesian optimization, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 1282–1291.
[18] H. Gutmann, A radial basis function method for global Optimization, Journal of Global Optimization 19 (2001) 201–2227.
[19] G. Halasz, M. Sperti, M. Villani, U. Michelucci, P. Agostoni, A. Biagi, L. Rossi, A. Botti, C. Mari, M. Maccarini, F. Pura, L. Roveda, A. Nardecchia, E. Mottola, M. Nolli, E. Salvioni, M. Mapelli, M. Deriu, D. Piga, M. Piepoli, Predicting outcomes in the machine learning era: the Piacenza score a purely data driven approach for mortality prediction in COVID-19 pneumonia, Journal of Medical Internet Research 23 (5) (2021).
[20] A. Holzinger, Interactive machine learning for health informatics: when do we need the human-in-the-loop? Brain Informatics 3 (2) (2016) 119–131.
[21] G. Latten, M. Spek, J. Muris, J. Cals, P. Stassen, Accuracy and interobserver-agreement of respiratory rate measurements by healthcare professionals, and its effect on the outcomes of clinical prediction/diagnostic rules, Public Library of Science San Francisco 14 (10) (2019).
[22] L. Ljung, H. Hjalmarsson, H. Ohlsson, Four encounters with system identification, European Journal of Control 17 (5-6) (2011) 449–471.
[23] M. Maadi, H. Akbarzadeh Khorshidi, U. Aickelin, A review on human–ai interaction in machine learning and insights for medical applications, International Journal of Environmental Research and Public Health 18 (4) (2021) 2121.
[24] D. McDonald, W. Grantham, W. Tabor, M. Murphy, Global and local optimization using radial basis function response surface models, Applied Mathematical Modelling 31 (10) (2007) 2095–2110.
[25] D.J. Newman, Uci repository of machine learning database, http://www.ics.uci.edu/mlearn/MLRepository.html (1998).
[26] A. Rodriguez-Morales, J. Cardona-Ospina, E. Gutiérrez-Ocampo, R. Villamizar-Peña, et al., Clinical, laboratory and imaging features of COVID-19: a systematic review and meta-analysis, Travel medicine and infectious disease 34 (2020) 101623.
[27] Y. Saeys, I. Inza, P. Larranaga, A review of feature selection techniques in bioinformatics, bioinformatics 23 (19) (2007) 2507–2517.
[28] M. Stone, Cross-validatory choice and assessment of statistical predictions, Journal of the Royal Statistical Society: Series B (Methodological) 36 (2) (1974) 111–133.
[29] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: A review, Data classification: algorithms and applications (2014) 37.
[30] Ö. Uncu, I. Türkşen, A novel feature selection approach: combining feature wrappers and filters, Information Sciences 177 (2) (2007) 449–466.
[31] S. Varela-Santos, P. Melin, A new approach for classifying coronavirus COVID-19 based on its manifestation on chest x-rays using texture features and neural networks, Information sciences 545 (2021) 403–414.