Personalized Algorithmic Recourse with Preference Elicitation

Giovanni De Toni[1,2], Paolo Viappiani[3], Stefano Teso[2,4], Bruno Lepri[1], and Andrea Passerini[2]

[1] Fondazione Bruno Kessler, Trento, Italy
[2] DISI, University of Trento, Trento, Italy
[3] LAMSADE, CNRS, Université Paris-Dauphine, PSL
[4] CIMeC, University of Trento, Trento, Italy

Abstract. *Algorithmic Recourse* (AR) is the problem of computing a *sequence of actions* that – once performed by a user – overturns an undesirable machine decision. It is paramount that the sequence of actions does not require too much effort for users to implement. Yet, most approaches to AR assume that actions cost the same for all users, and thus may recommend unfairly expensive recourse plans to certain users. Prompted by this observation, we introduce PEAR, the first human-in-the-loop approach capable of providing *personalized* algorithmic recourse tailored to the needs of any end-user. PEAR builds on insights from Bayesian Preference Elicitation to iteratively refine an estimate of the costs of actions by asking *choice set* queries to the target user. The queries themselves are computed by maximizing the *Expected Utility of Selection*, a principled measure of information gain accounting for uncertainty on both the cost estimate and the user's responses. PEAR integrates elicitation into a Reinforcement Learning agent coupled with Monte Carlo Tree Search to quickly identify promising recourse plans. Our empirical evaluation on real-world datasets highlights how PEAR produces high-quality personalized recourse in only a handful of iterations.

Keywords: Algorithmic Recourse · Counterfactuals · Preference Elicitation · Explainable AI

1 Introduction

Automated decision support systems are increasingly employed in high-risk decision tasks with the aim of empowering human decision-makers and improving the quality of their decisions. Example applications include bail requests [9], loan approvals [37], job applications [24], and prescription of medications and treatments [52]. Despite their promise, often these systems are opaque – meaning that users, and even engineers, have trouble understanding and controlling their decision process – and provide no means for overturning unwanted outcomes, such as denied loan requests. One way of addressing these issues is through the lens of *Algorithmic Recourse* (AR) [42,20]. In AR, given an undesirable machine-generated decision, the goal is to identify a sequence of actions – or *interventions* for short – that once implemented by the user overturns said decision, for instance, changing jobs or obtaining a master's degree. Motivated by this, a number of approaches have been recently proposed for computing AR [7,29,19,34,51,41,18,40,36,28,49,6].

It is critical that the suggested recourse plans are not too difficult or expensive to carry out. This entails that recourse should be *personalized*, because different users in the same situation may *need* substantially different recourse plans. To see this, consider a user who is denied a loan. Based on a profile made by the financial institution, an AR algorithm might suggest the user to reduce their monthly expenses. However, unlike the "average" customer, our user is incurring high medical expenses because they recently contracted an invalidating illness. Thus, the AR suggestion is highly inappropriate. Clearly, it is impossible to infer such a constraint from their profile alone. Equally importantly, actions influence each other's costs – *e.g.*, obtaining an additional degree can dramatically lower the cost of landing a better-payed job – and we need to account for this if we wish to minimize the cost of recourse. Most approaches, however, completely neglect the user's own preferences. The few that do require feedback that is difficult to obtain in practice, *e.g.*, preferences over a large pool of alternatives [36,28,49,6] or upfront quantification of action costs [50,19,35,49,26].

We argue that algorithmic recourse should make users first-class citizens in the recourse generation process rather than viewing them as passive observers. To this end, we introduce `PEAR` (Preference Elicitation for Algorithmic Recourse), the first human-in-the-loop approach for generating *personalized* recourse tailored for a target end-user. Our algorithm integrates AR and ideas from interactive Preference Elicitation (PE) [5,2,3,13,45,8] in a fully Bayesian setup. `PEAR` goes beyond existing approaches in that the costs of actions are estimated from user feedback and prior information. In each iteration, `PEAR` identifies a *small* selection of alternative interventions – a *choice set* – that optimizes a sound measure of information gain (the *Expected Utility of Selection* (EUS) [45,46]) and then asks the user to pick their preferred option. Using this feedback, `PEAR` quickly improves its estimate of the user's preferences and generates interventions that get progressively closer to the user's ideal. See Fig. 1 for an overview.

Contributions: Summarizing, we:

- Introduce the problem of *personalized algorithmic recourse*, and show that existing approaches are insufficient to solve it.
- Develop `PEAR`, the first human-in-the-loop, Bayesian approach for computing *personalized* interventions that is robust to noise in user feedback and minimizes user effort.
- Evaluate `PEAR` on synthetic and real-world datasets and show that it can generate substantially – up to 50% – cheaper interventions than user-agnostic competitors after only a handful of queries.

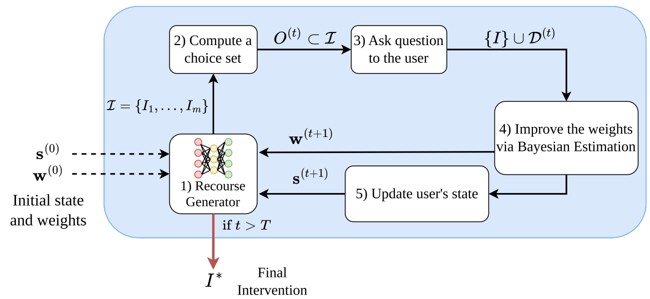

Fig. 1: Overview of `PEAR`. (1) Given the initial state $\mathbf{s}^{(0)}$ of the user and weights $\mathbf{w}^{(0)}$, `PEAR` computes a pool of candidate interventions achieving recourse. (2) A choice set $O^{(t)}$ is selected from the pool and presented to the user. (3) The user picks their preferred intervention from the set. (4) An improved estimate of the weights $\mathbf{w}^{(t+1)}$ is computed using this feedback, and (5) the user's state $\mathbf{s}^{(t+1)}$ is updated. After T rounds, the estimated weights are used to compute a final intervention I^*.

2 Problem Statement

The user *state* $\mathbf{s} \in \mathcal{S} \subseteq \mathbb{R}^d$ is a vector of d categorical and real-valued features encoding, *e.g.*, instruction level and income. An *action* $a \in \mathcal{A}$ is a map that takes a state $\mathbf{s}$ and changes a *single* feature, yielding a new state $\mathbf{s}' = a(\mathbf{s})$, and expresses a recommendation of the form "*Increase your income by $100.*" Given a (black-box) binary classifier[5] $h : \mathcal{S} \to \{0, 1\}$ and a user state $\mathbf{s}$ leading to an undesirable decision $y = h(\mathbf{s})$, AR computes an *intervention* – *i.e.*, an ordered set of actions $I = \{a^{(1)}, \dots, a^{(|I|)}\}$ – that can be applied to $\mathbf{s}$ to obtain a counterfactual state $\mathbf{s}'$ associated to a more desirable outcome $h(\mathbf{s}') \neq y$, *all while minimizing user effort*.

Approaches to AR assume the user effort is proportional to the number of actions that need to be carried out, and minimize it by searching for *short* interventions I. However, this is unrealistic and impractical. For instance, changing job into a highly skilled one may not be realistic without obtaining a Master's degree first. Motivated by this, we introduce a new problem setting, denoted *personalized algorithmic recourse*:

Definition 1 (Personalized Algorithmic Recourse). *Given a black-box binary classifier h and a user state $\mathbf{s}$, acquire a cost function $C(I \mid \mathbf{w})$ for interventions such that the intervention I^* obtained by solving the following optimization problem:*

$$I^* \in \text{argmin}_I \; C(I \mid \mathbf{w}) \quad \text{s.t.} \quad h(I(\mathbf{s}^{(0)})) \neq h(\mathbf{s}^{(0)}) \tag{1}$$

[5] It is straightforward to adapt our approach to deal with multiclass classification problems.

has ***minimal regret*** *for the target user, defined as:*

$$Reg(I^*, I^{GT}) = C(I^* \mid \mathbf{w}^{GT}) - C(I^{GT} \mid \mathbf{w}^{GT}) \tag{2}$$

where $\mathbf{w}^{GT}$ *encodes the ground-truth but unobservable preferences of the user and* I^{GT} *is the "ideal" intervention that would be obtained by solving Eq. (1) using* $\mathbf{w}^{GT}$.

The similarity of Eq. (1) to existing formulations of AR can be misleading, as here the key challenge is that of obtaining weights $\mathbf{w}$ that reflect the user's own preferences. We discuss how `PEAR` does so in Section 4.

3 Related work

Counterfactual explanations (CEs) are a class of local, human-understandable explanations [48,4] that convey information about changes to input variables that overturn a machine decision [12,39]. AR aims to identify *actionable* CEs that attain recourse for the user [43,19]. Existing approaches to AR solve Eq. (1) via diverse optimization methods [48,34,32,29,36,28,49,6] or by learning a general policy [51,7,44]. Most methods simply return a set of actions, disregarding their order. However, recent research showed that ignoring the causal relationship between features prevents reaching optimal recourse [20]. Some methods thus optimize for recourse plans, *i.e.*, *sequences* of actions attaining recourse, following a causal setup [41,20,21,7,29,26]. `PEAR` follows this paradigm and considers the interplay between features when finding recourse.

Most AR approaches assume that the cost function is fully specified beforehand [48,34,35,49,26], ignoring the problem of modelling user preferences altogether. The few that explicitly deal with user preferences do so in a naïve manner. Some of them [36,28,6,50] ask users to pick their preferred option from a *large* pool of user-agnostic recourse plans, that is not guaranteed to contain a low-cost option for the user. This is also impractical, as users can only properly evaluate a limited number of alternatives at a time [38,27]. Others require users to *quantify* the cost of each possible action upfront [19,35,49,26], or via numerical constraints [32,49], yet end-users can rarely articulate their preferences in a quantitative manner [22]. `PEAR` sidesteps this issue by learning preferences from ranking data, *i.e.*, relative judgments of the form "I prefer option A to option B" [11].

AR is specifically concerned with high-stakes scenarios, such as loan requests, that users face at most a handful of times in their lifetime. In these settings, it is impossible to estimate user preferences from historical data. In recommender systems, this issue has been solved through *preference elicitation*, whereby the user's preferences are estimated through a human-friendly interaction protocol [2]. In order to converge to high-quality options with minimal user effort, PE algorithms select queries (*i.e.*, questions to the user) that maximize information gain and that are easy to answer. A popular option is *choice queries*, in which the user has to select a preferred item from a small set of alternatives. `PEAR` builds on Bayesian PE methods, as they account for imprecision in users' answers

[5,13,45] in a principled way by measuring the information gain of choice sets in terms of Expected Utility of Selection [46]. Similarly to PEAR, [35] estimate cost weights from preference feedback. However, they collect feedback from *domain experts* in a non-interactive fashion and learn *population-level* preferences that are not personalized, thus failing to minimize Eq. (2). Population-level estimates can lead to recourse that is largely suboptimal for specific individuals, as will be shown in our experimental evaluation.

4 Personalized Algorithmic Recourse with PEAR

Before describing PEAR, we discuss how we model the user's preferences and how these impact the costs of actions and interventions. We formalize the user effort required to perform an action as a *cost function* $C : \mathcal{A} \times \mathcal{S} \to \mathbb{R}^+$. In order to account for interactions between action costs, we model C using a linear *Structural Causal Model* (SCM) [31] parameterized by weights $\mathbf{w}$. This SCM is a directed graph where each node is associated with the cost w_k of changing a single feature s_k and each edge represents a causal relationship between two costs, parameterized by $w_{jk} \in \mathbb{R}$.

Then, following the structural assignments induced by the SCM, we define the cost of applying an action a_k to a state $\mathbf{s}$ to change the k-th feature from s_k to s'_k as a linear combination:

$$C(a_k, \mathbf{s} \mid \mathbf{w}) = w_k |s'_k - s_k| + \textstyle\sum_{j \in Pa_k} w_{jk} s_j \tag{3}$$

Here, Pa_k are the *parents* of the k-th node in the graph. An example of cost function is shown in Fig. 2. The cost of an intervention is the sum of the costs of all actions it contains, that is:

$$C(I \mid \mathbf{w}) = \textstyle\sum_{i=0}^{|I|} C(a^{(i)}, \mathbf{s}^{(i)} \mid \mathbf{w}) \tag{4}$$

Here, $\mathbf{s}^{(i)}$ is the state obtained by applying action $a^{(i-1)}$ to state $\mathbf{s}^{(i-1)}$, and $\mathbf{s}^{(0)} = \mathbf{s}$ is the initial state. We denote with $I(\mathbf{s}^{(0)})$ the operation of applying each action $a \in I$ sequentially. Eq. (3) does not define a simple linear model, but it accounts for a richer set of interactions. For example, given two actions a_1 and a_2, their cost is not additive, meaning $C(a_1, \mathbf{s} \mid \mathbf{w}) + C(a_2, \mathbf{s} \mid \mathbf{w}) \neq C(\{a_1, a_2\}, \mathbf{s} \mid \mathbf{w})$. We can show it with a simple example.

Example of non-additivity of Eq. (3). Consider a simple SCM with two nodes $s_1 \to s_2$, with $\mathbf{s} = (s_1, s_2)$ and $\mathbf{w} = (w_1, w_2, w_{12})$, and an instance $\mathbf{s} = (1, 1)$ and $\mathbf{w} = (1, 0.5, 1)$. We define two actions, a_1 and a_2, which simply set the corresponding features s_i to 2. Using Eq. (3), we can compute the cost of applying the single action as

$$C(a_1, \mathbf{s} \mid \mathbf{w}) = 1|2-1| = 1 \qquad C(a_2, \mathbf{s} \mid \mathbf{w}) = 0.5|2-1| + 1 = 1.5 \tag{5}$$

By Eq. (4), the cost of the intervention $\{a_1, a_2\}$ is $C(\{a_1, a_2\}, \mathbf{s} \mid \mathbf{w}) = 3.5 \neq 1.5{+}1$, hence the costs are *not* additively independent. Moreover, Eq. (4) considers the *order* in which actions are applied, *i.e.*, $\exists\, \mathbf{w}$ such that $C(\{a_1, a_2\}, \mathbf{s} \mid \mathbf{w}) \neq C(\{a_2, a_1\}, \mathbf{s} \mid \mathbf{w})$, as shown in Fig. 2(b).

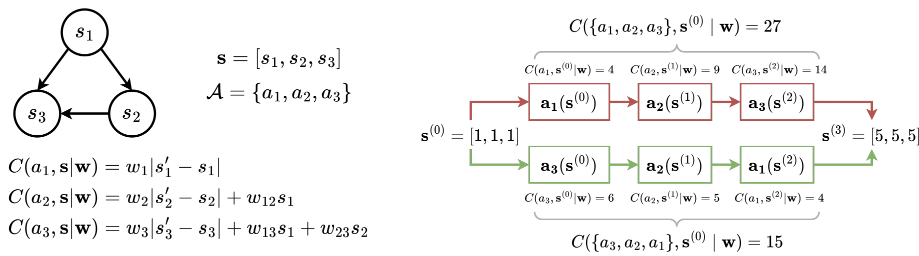

Fig. 2: **The cost model** (Left) A *Linear Structural Causal Model* for cost modelling. (Right) Given $\mathbf{s}^{(0)} = [1, 1, 1]$ and unit $\mathbf{w}$, let us imagine we want to reach $\mathbf{s}^{(3)} = [5, 5, 5]$ by following the presented SCM and the intervention $I = \{a_1, a_2, a_3\}$, where $\forall\ i\ a_i$ assign $s_i \leftarrow 5$. Clearly, we incur different costs by applying permuted versions of I. The green path indicates the lower-cost intervention.

4.1 The `PEAR` Algorithm

In order to account for uncertainty over the user's weights $\mathbf{w}$, `PEAR` explicitly models a distribution $P(\mathbf{w})$ over them and progressively refines it by interacting with a target user. A high-level overview of `PEAR` is given in Fig. 1 and the pseudo-code is listed in Algorithm 1.

In each iteration $t = 1, \ldots, T$, where T is the iteration budget, `PEAR` computes a *choice set* $O^{(t)} \in \mathcal{I}^k$ containing k candidate interventions achieving recourse (for a small k, *e.g.*, 2 to 4) and asks the user to indicate their most preferred option in the set. Importantly, $O^{(t)}$ is chosen so as to maximize the (expected) information gained from the user, and in a way that is robust to noise in their feedback. We detail the exact procedure used by `PEAR` in Section 4.3. These user choices are stored in an initially empty dataset $\mathcal{D}^{(t)}$. In each step, `PEAR` integrates the user's feedback by inferring a posterior over the weights $P(\mathbf{w} \mid \mathcal{D}^{(t)}) \propto P(\mathcal{D}^{(t)} \mid \mathbf{w})P(\mathbf{w})$ using Bayesian inference, and updates the user state by applying the first action $\hat{I}_1$ of the chosen intervention. We apply a single action so as to elicit user preferences in all intermediate states. If a state achieving recourse is reached, the user state is reinitialized. After T rounds,[6] `PEAR` computes a low-cost personalized intervention by applying the intervention generation procedure described in Section 4.2, biased according to the latest posterior $p(\mathbf{w} \mid \mathcal{D}^{(t)})$.

`PEAR` makes no assumption on the form of the prior $P(\mathbf{w})$, meaning that the prior can be adjusted based on the application. In order to model both variances across the preferences of individuals and for sub-groups in the population, in this work we model it as a mixture of Gaussians with M components $\mathcal{N}_i(\mu_i, \mathbf{\Sigma}_i)$, for $i = 1, \ldots, M$. This choice works well in our experiments, see Section 5. Note that,

[6] In practice, the loop can be terminated as soon as the user is satisfied with one of the interventions in $O^{(t)}$.

Algorithm 1 The PEAR algorithm: $h : \mathcal{S} \to \{0, 1\}$ is a classifier, $\mathbf{s}^{(0)} \in \mathcal{S}$ the initial state, $\mathcal{A}$ the available actions, $p(\mathbf{w})$ the prior, $T \geq 1$ the query budget, $k \geq 2$ is the size of choice sets.

procedure PEAR($h, \mathbf{s}^{(0)}, \mathcal{A}, T, k$)
  Initialize $t \leftarrow 0$, $\mathcal{D}^{(0)} \leftarrow \varnothing$
  for $t = 1, \ldots, T$ **do**
    $O^{(t)} \leftarrow$ SUBMOD-CHOICE($h, \mathbf{s}^{(t-1)}, \mathcal{A}, k, \mathcal{D}^{(t-1)}$) ▷ Algorithm 2
    Ask the user to pick the best intervention $\hat{I} \in O^{(t)}$
    $\mathcal{D}^{(t)} \leftarrow \mathcal{D}^{(t-1)} \cup \{\hat{I}\}$
    Update weight estimate $p(\mathbf{w} \mid \mathcal{D}^{(t)})$
    $\mathbf{s}^{(t)} \leftarrow \hat{I}_1(\mathbf{s}^{(t-1)})$
    if $h(\mathbf{s}^{(t)}) \neq h(\mathbf{s}^{(0)})$ **then**
      $\mathbf{s}^{(t)} \leftarrow \mathbf{s}^{(0)}$
  $I^* =$ W-FARE($h, \mathbf{s}^{(0)}, \mathbf{w}^*$) with $\mathbf{w}^* = \mathbb{E}_{P(\mathbf{w}|\mathcal{D}^{(T)})}[\mathbf{w}]$ ▷ Section 4.2
  return I^*

analogously to [35], it is also possible to fit the prior on population-level preference data or domain expert input, whenever this is available. In our experiments, we do this for *all* competitors.

4.2 Generating Personalized Interventions with W-FARE

PEAR generates personalized interventions by leveraging a novel, *user-aware* extension of FARE [7], a state-of-the-art algorithm for generating short – but *user-agnostic* – interventions, which we briefly outline next. In FARE, each action $a \in \mathcal{A}$ is implemented as a tuple (f, x), where f is a function changing *one* feature and x is the value that feature takes, *e.g.*, (*change_income*, \$1000). Given an initial state $\mathbf{s}$, FARE uses reinforcement learning to learn two probabilistic policies $\pi_f(\mathbf{s})$ and $\pi_x(\mathbf{s})$, which are used as priors to guide a Monte Carlo Tree Search procedure that incrementally builds an intervention I by selecting actions $a^{(i)} \in \mathcal{A}$. In order to ensure interventions are *actionable*, actions a are only chosen if they satisfy given preconditions. The reward used by FARE is $r(I) = \rho^{|I|} \cdot \mathbb{1}\{h(I(\mathbf{s}^{(0)})) \neq h(\mathbf{s}^{(0)})\}$, where $\rho > 0$ is a discount factor and the indicator evaluates to 1 if I attains recourse and to 0 otherwise. FARE is highly scalable and very effective at identifying counterfactual interventions even under a minimal training budget [7].

FARE is user-agnostic, while PEAR needs to generate *personalized* interventions. We fill this gap by introducing W-FARE, a novel extension of FARE that integrates the user's costs into the reward while inheriting all benefits of the latter. Recall that PEAR maintains a posterior over the weights. The *expected cost of an action* can thus be obtained by marginalizing over the posterior:

$$\mathbb{E}[C(a, \mathbf{s}) \mid \mathcal{D}^{(t)}] = \int_{\mathbf{w}} C(a, \mathbf{s} \mid \mathbf{w}) P(\mathbf{w} \mid \mathcal{D}^{(t)})\, d\mathbf{w} \tag{6}$$

Analogously, the cost of an intervention I is replaced by the expectation:

$$\mathbb{E}[C(I) \mid \mathcal{D}^{(t)}] = \sum_{i=0}^{|I|} \mathbb{E}[C(a^{(i)}, \mathbf{s}^{(i)}) \mid \mathcal{D}^{(t)}] \tag{7}$$

The W-FARE reward function $r(I \mid \mathbf{w})$ is then proportional to $\rho^{\mathbb{E}[C(I|\mathcal{D}^{(t)})]} \cdot \mathbb{1}\{h(I(\mathbf{s}^{(0)})) \neq h(\mathbf{s}^{(0)})\}$. This explicitly drives RL to learn policies that optimize user-specific action costs and that, therefore, help MCTS to more quickly converge to *personalized* interventions. We show empirically that PEAR is substantially more effective than FARE at computing personalized interventions in Section 5.

4.3 Computing Informative Choice Sets

Given the current posterior $P(\mathbf{w} \mid \mathcal{D}^{(t)})$, PEAR computes a *choice set* containing k interventions I that maximizes information gain [5]. We measure the latter using the *Expected Utility of Selection* (EUS) [46], a measure of the goodness of a set defined as the expectation, under the uncertainty over $\mathbf{w}$, of the utility of its most preferred element. EUS is closely related to the Expected Value of Information (EVOI), and frequently used in Bayesian PE [33,45,1,46]. The EUS builds on the notion of *expected utility of an intervention* I, which is defined as:

$$\mathsf{EU}(I \mid \mathcal{D}^{(t)}) = \mathbb{E}[-C(I) \mid \mathcal{D}^{(t)}] = -\int_{\mathbf{w}} C(I \mid \mathbf{w}) P(\mathbf{w} \mid \mathcal{D}^{(t)})\, d\mathbf{w} \qquad (8)$$

The EUS is of a choice set O can then be defined as:

$$\begin{aligned}\mathsf{EUS}_R(O \mid \mathcal{D}^{(t)}) &= \textstyle\sum_{I \in O} P_R(O \rightsquigarrow I) \mathsf{EU}(I \mid \mathcal{D}^{(t)}) \\ &= -\textstyle\int_{\mathbf{w}} \left[\sum_{I \in O} P_R(O \rightsquigarrow I \mid \mathbf{w}) C(I \mid \mathbf{w})\right] P(\mathbf{w} \mid \mathcal{D}^{(t)})\, d\mathbf{w} \qquad (9)\end{aligned}$$

Here, $P_R(O \rightsquigarrow I \mid \mathbf{w})$ is the probability that a user with weights $\mathbf{w}$ picks I from O, under a specific choice of *response model* R modelling noise in user choices. Intuitively, we expect users tend to prefer interventions $I \in O$ that cost less than that of the other interventions in O and that interventions with similar costs have a similar probability of being chosen. Motivated by this, and following common practice in choice modelling [25], in PEAR we implement a *logistic response model* (L), defined as:

$$P_L(O \rightsquigarrow I \mid \mathbf{w}) = \frac{\exp(-\lambda C(I \mid \mathbf{w}))}{\sum_{I \in O} \exp(-\lambda C(I \mid \mathbf{w}))} \qquad (10)$$

Here, $\lambda \in \mathbb{R}$ is a temperature parameter. Finding a choice set O maximizing the EUS is intractable in general – *NP-hard* [30,23], in fact – and computationally intensive in practice, and risks slowing down the interaction loop to the point of estranging users. We observe that, however, under some response models R, the EUS becomes *submodular* [23] and *monotonic*. This is the case for the *noiseless* response model (NL), according to which the user always prefers the lowest-cost option, *i.e.*,[7]

$$P_{NL}(O \rightsquigarrow I \mid \mathbf{w}) = \textstyle\prod_{I, I' \in O \,:\, I \neq I'} \mathbb{1}\{C(I \mid \mathbf{w}) < C(I' \mid \mathbf{w})\} \qquad (11)$$

[7] For the sake of presentation, we assume that there are no ties. Note that the EUS formula is invariant to the way ties are broken. In our implementation, ties are broken uniformly at random.

Algorithm 2 Greedy procedure to efficiently compute a choice set O: $\mathbf{s}^{(t)} \in \mathcal{S}$ the current state, $\mathcal{A}$ the available actions, $k \geq 2$ is the size of choice sets, $\mathcal{D}^{(t)}$ the user choices so far.

procedure SUBMOD-CHOICE($\mathbf{s}^{(t)}, k, \mathcal{A}, \mathcal{D}^{(t)}$)
  $O \leftarrow \varnothing$
  $\bar{\mathbf{w}} \leftarrow \mathbb{E}_{p(\mathbf{w} \mid \mathcal{D}^{(t)})}[\mathbf{w}]$
  while $|O| < k$ **do**
    Generate the candidate interventions $\mathcal{I}$ with W-FARE using $\mathcal{A}$ and $\bar{\mathbf{w}}$
    $\hat{I} \leftarrow \text{argmax}_{\hat{I}}\ \mathsf{EUS}_{NL}(O \cup \hat{I} \mid \mathcal{D}^{(t)}) - \mathsf{EUS}_{NL}(O \mid \mathcal{D}^{(t)})$
    $O \leftarrow O \cup \{\hat{I}\}$
  return O

This means that, for NL, greedy optimization is sufficient to find a choice set O that achieves high EUS_{NL} with approximation guarantees. Formally, it holds that for choice sets O found via greedy optimization, $\mathsf{EUS}_{NL}(O \mid \mathcal{D}) \geq (1 - e^{-1})\mathsf{EUS}_{NL}(O^* \mid \mathcal{D})$, where O^* is the truly optimal choice set [45,30,23]. In PEAR, we leverage the fact that $\mathsf{EUS}_L - \mathsf{EUS}_{NL}$ is always smaller than a problem independent (tight) bound [45], meaning that instead of minimizing EUS_L directly, we can compute a high-quality choice set by greedily maximizing EUS_{NL} instead. This immediately leads to a practical algorithm for the logistic response model L, listed in Algorithm 2.

4.4 Benefits and limitations.

PEAR is designed to facilitate the application of AR to practical high-stakes tasks like loan approval. The main benefit of PEAR is that it provides *personalized* algorithmic recourse, which existing approaches are not capable of. Also, it follows a fully Bayesian setup for handling uncertainty over the estimated preferences and noise in user feedback. It also leverages ideas from preference elicitation – such as small *choice sets* and elicitation of relative preferences – to ensure the interaction is cognitively affordable.

Several steps of the algorithm – namely, evaluating the EUS (Eq. (9)) and the expected cost of interventions (Eq. (7)), and updating the posterior $p(\mathbf{w} \mid \mathcal{D}^{(t)})$ – require marginalizing over the weights. Doing so involves evaluating a complex integral that cannot be solved analytically. We sidestep this issue by leveraging an efficient Monte Carlo approximation, and specifically *ensemble split sampling* [16,17] and then averaging over the samples with the highest likelihood. We find that, empirically, this procedure is efficient – so much that it can support interactive usage – and leads to competitive results in our experiments, cf. Section 5.

5 Experiments

Our experimental evaluation is aimed at answering the following research questions:

Q1 Does PEAR succeed in minimizing regret for increasing amount of user feedback?
Q2 Does PEAR outperform competitors in terms of validity and cost?
Q3 Is PEAR robust to imprecise knowledge of the causal graph of the user?

Datasets and Classifiers. We evaluated our approach on two real-world datasets taken from the relevant literature: GiveMeSomeCredit [15] and Adult [10]. They are (unbalanced) binary classification problems for income prediction and loan assignment respectively. The datasets have both categorical and numerical features. Some of these features are actionable (e.g., *occupation*, *education*), while others are immutable (e.g., *age*, *sex*, *native_ country*). These datasets come without a causal graph and users' preferences over the features. Following previous work [20,7,29], we manually defined fixed SCMs for both. We randomly generated user-specific weights for each instance by sampling from a (dataset-specific) mixture of Gaussians with $M = 6$ components. We then split the data into training (70%), validation (10%) and test (20%) sets. For each dataset, we designed the black-box classifier h as an MLP with two hidden layers. We trained it by cross-entropy minimization, selecting the hyperparameters which maximise the F_1 score on the validation set.

"Easy" vs. "Hard" Users. Intuitively, users close to the decision boundary of the black-box h will require few actions to achieve recourse, while users to whom h assigns a low score might need longer and more complex interventions. Understanding the preferences of these "hard" users is crucial since a wrong suggestion might increase sensibly the overall cost for them. For each dataset, we thus built two separate testing sets. The first one, named All, is obtained by sampling 300 users $\mathbf{s}$ with an unfavourable classification ($h(\mathbf{s}) < 0.5$), regardless of the actual value of h. The second one, named Hard, is obtained by sampling 300 users with an unfavourable classification having a score in the lower quartile of the black-box score distribution.

Competitors. We compare PEAR against several baselines: FARE and its explainable version EFARE [7], CSCF [29], an evolutionary algorithm which, similarly to FARE, generates recourse options by considering causal cost functions and action sets $\mathcal{A}$, and FACE [32], a well-know AR algorithm, which optimizes for population-based "feasible paths" to achieve recourse. We also consider two simpler baselines, a brute-force search (MCTS) and a vanilla reinforcement learning agent (RL), trained in a similar way as [14,44]. Note that all the competitors are model-agnostic and *not* interactive, since they assume the users' costs to be fixed.

Experimental Protocol. For PEAR, we vary the number of questions T to the user from 0 to 10. For $T = 0$, we initialize the weights with the expected value of the prior, $\mathbb{E}_{P(\mathbf{w})}[\mathbf{w}]$, that represents a user-independent population-based prior. Moreover, we employ two user response models, the *noiseless* model (Eq. (11)), to check the effectiveness of our approach in the best-case scenario where the user can perfectly express their preferences, and the *logistic* model (Eq. (10)), to challenge our approach in a more realistic scenario. To provide a fair comparison, we equip the competitors with the causal cost function and set their weights to the expected value of the prior.

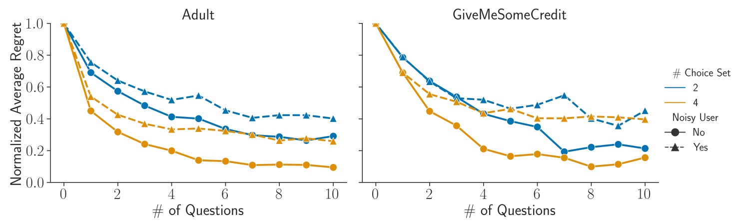

Fig. 3: Normalized Average Regret for PEAR when varying the number of questions, the choice set size and the user response model on both datasets (sampled from All users).

Q1: PEAR successfully minimizes the regret. Fig. 3 shows the evaluation of the regret as a function of the number of queries to the user. Here the ground-truth intervention I^{GT} (which is unknown) is approximated by running PEAR with the correct user costs $\mathbf{w}^{GT}$, and the regret is normalized by rescaling the costs between $C(I^{\mathrm{GT}} \mid \mathbf{w}^{GT})$ and $C(I^{(0)} \mid \mathbf{w}^{GT})$ where we generate $I^{(0)}$ using the expectation of the prior. We run PEAR with two different choice set dimensions, $k = 2$ and $k = 4$, and for both noiseless and logistic response models. After a few questions, PEAR reaches a low regret in all settings. Generally, a larger choice set produces a lower regret, irrespective of the response model, with the downside of increasing the cognitive burden for the user. We now briefly summarize the results when $T = 10$. For the Adult dataset, the best regret is ≈ 0.09 for the noiseless user and $k = 4$, while the worst regret is ≈ 0.40 for the logistic response model and $k = 2$. For GiveMeSomeCredit, we get ≈ 0.15 (noiseless, $k = 4$) and ≈ 0.45 (logistic, $k = 2$). Overall, we can provide interventions which are at least 50% cheaper than their preference-agnostic counterparts.

Q2: PEAR outperforms competitors in terms of validity and cost. Following the AR literature [19,43], we compare PEAR (with $T = 10$) and all competitors in terms of average *validity*, *i.e.*, fraction of users for which we obtained recourse, intervention *cost* and *length* (or *sparsity*), *i.e.*, the number of features that have to be changed. Intervention costs are computed by using the true weights $\mathbf{w}^{GT}$. Table 1 shows the results. PEAR manages to achieve the highest validity while also providing substantially cheaper interventions than the non-personalized competitors on average. This is true both for the noiseless and logistic response models. While CSCF tends to produce shorter interventions, these are in general more costly and have a larger cost variance with respect to those found by PEAR, confirming the intuition that length is a suboptimal proxy of intervention complexity. The only exception is the Hard setting of GiveMeSomeCredit, where however CSCF manages to achieve recourse for only 14% of the users, whereas PEAR achieves recourse in 58% of the cases. The difficulty of CSCF in achieving recourse is visible in all settings and severely limits its applicability. Furthermore, CSCF is 10 to 50 times more computationally expensive than PEAR, making it unsuitable for real-time interactive scenarios. The MCTS baseline has rather poor performance both in terms of validity and

Table 1: Performance of all competitors averaged over 10 runs. A '-' indicates that the method did not find *any* successful intervention for *any* user. $\texttt{PEAR}_{NL}$ and $\texttt{PEAR}_{L}$ indicate PEAR associated with the noiseless and logistic response model, respectively. The best results are boldfaced.

Users	Method	Adult			GiveMeSomeCredit		
		Validity	*Cost*	*Length*	*Validity*	*Cost*	*Length*
All	FARE	0.97 ± 0.16	257.48 ± 191.58	3.04 ± 1.06	0.83 ± 0.27	132.52 ± 97.26	2.97 ± 1.08
	RL	0.86 ± 0.31	247.84 ± 190.58	2.94 ± 1.06	-	-	-
	EFARE	0.85 ± 0.33	267.08 ± 192.45	3.10 ± 1.14	0.67 ± 0.37	123.18 ± 96.57	2.84 ± 1.06
	CSCF	0.80 ± 0.30	171.69 ± 140.69	$\mathbf{2.64 \pm 0.66}$	0.60 ± 0.40	101.31 ± 113.46	$\mathbf{2.52 \pm 1.10}$
	MCTS	0.44 ± 0.44	439.78 ± 202.44	4.55 ± 1.26	0.72 ± 0.41	220.98 ± 105.71	4.18 ± 1.24
	FACE	0.14 ± 0.19	410.79 ± 96.77	4.00 ± 0.59	0.23 ± 0.36	328.68 ± 46.15	6.09 ± 0.52
	$\texttt{PEAR}_{NL}$ **(ours)**	$\mathbf{1.00 \pm 0.03}$	$\mathbf{135.22 \pm 49.87}$	2.90 ± 0.52	$\mathbf{0.89 \pm 0.00}$	$\mathbf{92.81 \pm 26.27}$	2.85 ± 0.35
	$\texttt{PEAR}_{L}$ **(ours)**	$\mathbf{1.00 \pm 0.03}$	$\mathbf{140.18 \pm 52.35}$	2.88 ± 0.49	$\mathbf{0.89 \pm 0.02}$	$\mathbf{98.03 \pm 29.23}$	2.94 ± 0.45
Hard	FARE	0.92 ± 0.28	445.17 ± 210.11	4.33 ± 1.31	0.35 ± 0.42	297.03 ± 91.19	4.82 ± 0.70
	RL	0.71 ± 0.44	421.39 ± 209.66	4.11 ± 1.34	-	-	-
	EFARE	0.69 ± 0.45	438.38 ± 213.34	4.24 ± 1.35	0.13 ± 0.33	305.16 ± 75.11	4.75 ± 0.57
	CSCF	0.32 ± 0.43	386.73 ± 132.16	3.87 ± 0.60	0.14 ± 0.33	$\mathbf{189.26 \pm 113.89}$	$\mathbf{3.33 \pm 1.08}$
	MCTS	0.21 ± 0.40	573.09 ± 140.24	5.56 ± 0.70	0.40 ± 0.44	346.73 ± 97.86	5.43 ± 0.87
	FACE	0.00 ± 0.04	448.72 ± 0.00	5.20 ± 0.00	0.20 ± 0.34	441.98 ± 41.74	7.09 ± 0.46
	$\texttt{PEAR}_{NL}$ **(ours)**	$\mathbf{0.99 \pm 0.09}$	$\mathbf{299.03 \pm 43.62}$	$\mathbf{3.41 \pm 0.50}$	$\mathbf{0.58 \pm 0.04}$	242.43 ± 45.53	4.59 ± 0.38
	$\texttt{PEAR}_{L}$ **(ours)**	$\mathbf{0.98 \pm 0.09}$	$\mathbf{305.24 \pm 54.99}$	$\mathbf{3.40 \pm 0.54}$	$\mathbf{0.58 \pm 0.03}$	256.81 ± 46.07	4.64 ± 0.36

cost in all settings, while the `RL` baseline has a reasonably high validity on `Adult` but it completely fails to learn a policy achieving recourse on `GiveMeSomeCredit`. On the other hand, methods which combine `MCTS` and `RL` (`FARE` and `EFARE`) give better performance, which is aligned with previous results [7], but are still suboptimal with respect to `PEAR` in terms of both validity and cost. Finally, `FACE` struggles to achieve recourse since it needs to find a "feasible path" from the current user to a similar one *in the training set*, which is favourably classified.

Q3: `PEAR` is robust to misspecifications of the causal graph. In the previous experiments, following other research works [20,7,29], we assumed to know the structure of the SCM a-priori. However, in a real scenario, we might have instead an *approximate* causal graph. Table 2 shows the validity, cost and length of the interventions found by removing $X\%$ of edges from the causal graph, with $X \in \{0.15, 0.25, 0.50, 1.00\}$. Validity is almost unaffected by corruption in all settings since it only impacts the computation of the cost. On the other hand, as expected, increasing the amount of graph corruption reduces the effectiveness of user feedback. However, the degradation is not dramatic. Indeed, if we look at the `Hard` evaluation, the increase in cost is negligible (around 2%) with up to 25% randomly removed edges. When considering `All` users, the degradation is more evident, but still within 10% for `GiveMeSomeCredit`, while for `Adult` it goes up to 60%. The setting $X = 1.0$ is equivalent to a non-causal cost function, in which acting on a feature has always the same cost, irrespective of the others. It is the common choice of many works dealing with AR [48,34,32]. Under such a setting, even with elicitation, the interventions are up to 70% more expensive than if we were to consider interactions between features. Overall, results clearly indicate that `PEAR` can suggest reasonable cost interventions even with a largely misspecified causal graph. This is apparent when comparing these results with

Table 2: Evaluation of PEAR (with $q = 10$ and a logistic noise model) for increasing amount of causal graph corruption, averaged over 10 runs. "None" indicates that the correct causal graph is being used.

Users	Corruption	Adult			GiveMeSomeCredit		
		Validity	*Cost*	*Length*	*Validity*	*Cost*	*Length*
All	**None**	1.00 ± 0.03	140.18 ± 52.35	2.88 ± 0.49	0.89 ± 0.02	98.03 ± 29.23	2.94 ± 0.45
	0.15	1.00 ± 0.04	225.90 ± 66.95	3.27 ± 0.41	0.89 ± 0.00	105.45 ± 30.89	2.75 ± 0.54
	0.25	1.00 ± 0.00	226.25 ± 72.53	3.33 ± 0.43	0.89 ± 0.00	107.31 ± 25.81	2.75 ± 0.45
	0.5	1.00 ± 0.03	236.05 ± 66.11	3.28 ± 0.38	0.89 ± 0.00	114.45 ± 38.69	2.81 ± 0.53
	1.0	0.98 ± 0.03	243.86 ± 96.10	3.69 ± 0.62	0.89 ± 0.00	130.78 ± 51.86	3.01 ± 0.63
Hard	**None**	0.98 ± 0.09	305.24 ± 54.99	3.40 ± 0.54	0.58 ± 0.03	256.81 ± 46.07	4.64 ± 0.36
	0.15	0.99 ± 0.00	307.71 ± 46.34	3.04 ± 0.61	0.58 ± 0.00	262.26 ± 48.36	4.65 ± 0.36
	0.25	0.99 ± 0.00	312.40 ± 47.78	3.02 ± 0.57	0.59 ± 0.09	254.10 ± 60.96	4.55 ± 0.45
	0.5	0.99 ± 0.00	310.71 ± 48.30	3.19 ± 0.65	0.58 ± 0.06	276.56 ± 38.22	4.68 ± 0.39
	1.0	0.99 ± 0.03	363.70 ± 61.18	3.23 ± 0.74	0.58 ± 0.00	282.36 ± 26.96	4.64 ± 0.31

those in Table 1. Even with 50% randomly removed edges, PEAR recommends interventions that are cheaper than all competitors but CSCF, that however has a substantially lower validity.

6 Conclusion

In this work, we identify the problem of *personalized* algorithmic recourse as a fundamental stepping stone for ensuring recourse is usable in real-world applications, and develop PEAR, the first algorithm able to provide *personalized* interventions. Our experimental evaluation shows that PEAR substantially outperforms existing (non-personalized) solutions in terms of both validity and intervention cost with only a handful of queries to the user. We hope that this initial contribution can foster further research in the community to work towards a more realistic form of algorithmic recourse that can be successfully deployed in real-world scenarios. As for all methods dealing with algorithmic recourse, the effectiveness of the approach should, in principle, be evaluated on real users. However, this evaluation is highly non-trivial (and thus still missing in the algorithmic recourse literature) because it requires the creation of a realistic scenario where a user feels to be *unfairly treated* in some machine-driven decision involving her life. The legal requirements that are progressively being introduced to regulate AI systems [47] could contribute to making the information needed to set up such a scenario available in the near future.

Broader Impact. In principle, we develop these methods to increase the fairness of the current machine learning systems. However, we need to consider the potential bad ethical ramifications of these technologies. Eliciting users' preferences might entail asking sensitive questions, or malicious entities could exploit these procedures to "hack" and twist the intervention generation. These considerations can be mitigated by research on adversarial attacks to ensure the method's robustness. Moreover, legal advice might be needed to manage personal user data.

References

1. Akrour, R., Schoenauer, M., Sebag, M.: APRIL: active preference learning-based reinforcement learning. In: Flach, P.A., Bie, T.D., Cristianini, N. (eds.) Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II. Lecture Notes in Computer Science, vol. 7524, pp. 116–131. Springer (2012). https://doi.org/10.1007/978-3-642-33486-3_8, https://doi.org/10.1007/978-3-642-33486-3_8
2. Boutilier, C.: A pomdp formulation of preference elicitation problems. In: AAAI. pp. 239–246 (2002)
3. Braziunas, D., Boutilier, C.: Minimax regret based elicitation of generalized additive utilities. In: UAI. pp. 25–32 (2007)
4. Byrne, R.M.: Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning (2019)
5. Chajewska, U., Koller, D., Parr, R.: Making rational decisions using adaptive utility elicitation. In: AAAI/IAAI. pp. 363–369 (2000)
6. Dandl, S., Molnar, C., Binder, M., Bischl, B.: Multi-objective counterfactual explanations. In: PPSN. pp. 448–469. Springer (2020)
7. De Toni, G., Lepri, B., Passerini, A.: Synthesizing explainable counterfactual policies for algorithmic recourse with program synthesis. Machine Learning pp. 1–21 (2023)
8. Dragone, P., Teso, S., Passerini, A.: Constructive preference elicitation. Frontiers in Robotics and AI **4**, 71 (2018)
9. Dressel, J., Farid, H.: The accuracy, fairness, and limits of predicting recidivism. Science advances **4**(1), eaao5580 (2018)
10. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
11. Furnkranz, J., Hullermeier, E.: Preference Learning. Springer-Verlag, Berlin, Heidelberg, 1st edn. (2010)
12. Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., Giannotti, F.: Local rule-based explanations of black box decision systems. CoRR **abs/1805.10820** (2018), http://arxiv.org/abs/1805.10820
13. Guo, S., Sanner, S.: Real-time multiattribute bayesian preference elicitation with pairwise comparison queries. In: AISTATS. pp. 289–296 (2010)
14. Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Pfaff, T., Weber, T., Buesing, L., Battaglia, P.W.: Combining q-learning and search with amortized value estimates. arXiv preprint arXiv:1912.02807 (2019)
15. Kaggle: Give me some credit (Sep 2011), https://www.kaggle.com/competitions/GiveMeSomeCredit/overview
16. Karamanis, M., Beutler, F.: Ensemble slice sampling: Parallel, black-box and gradient-free inference for correlated & multimodal distributions. arXiv preprint arXiv: 2002.06212 (2020)
17. Karamanis, M., Beutler, F., Peacock, J.A.: zeus: A python implementation of ensemble slice sampling for efficient bayesian parameter inference. arXiv preprint arXiv:2105.03468 (2021)

18. Karimi, A., von Kügelgen, J., Schölkopf, B., Valera, I.: Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. In: NeurIPS (2020), https://proceedings.neurips.cc/paper/2020/file/02a3c7fb3f489288ae6942498498db20-Paper.pdf
19. Karimi, A.H., Barthe, G., Schölkopf, B., Valera, I.: A survey of algorithmic recourse: Contrastive explanations and consequential recommendations. ACM Comput. Surv. **55**(5) (dec 2022). https://doi.org/10.1145/3527848, https://doi.org/10.1145/3527848
20. Karimi, A.H., Schölkopf, B., Valera, I.: Algorithmic recourse: from counterfactual explanations to interventions. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. pp. 353–362 (2021)
21. Karimi, A.H., Von Kügelgen, J., Schölkopf, B., Valera, I.: Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. Advances in Neural Information Processing Systems **33**, 265–277 (2020)
22. Keeney, R.L., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Trade-Offs. Cambridge University Press (1993). https://doi.org/10.1017/CBO9781139174084
23. Krause, A., Golovin, D.: Submodular function maximization. Tractability **3**, 71–104 (2014)
24. Liem, C., Langer, M., Demetriou, A., Hiemstra, A.M., Sukma Wicaksana, A., Born, M.P., König, C.J.: Psychology meets machine learning: Interdisciplinary perspectives on algorithmic job candidate screening. In: Explainable and interpretable models in computer vision and machine learning, pp. 197–253. Springer (2018)
25. Luce, R.D.: Individual choice behavior: A theoretical analysis. Courier Corporation (2012)
26. Mahajan, D., Tan, C., Sharma, A.: Preserving causal constraints in counterfactual explanations for machine learning classifiers. arXiv preprint arXiv:1912.03277 (2019)
27. March, J.G.: Bounded rationality, ambiguity, and the engineering of choice. The bell journal of economics pp. 587–608 (1978)
28. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: FAT*. pp. 607–617 (2020)
29. Naumann, P., Ntoutsi, E.: Consequence-aware sequential counterfactual generation. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 682–698. Springer (2021)
30. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions—i. Mathematical programming **14**(1), 265–294 (1978)
31. Pearl, J.: Causality. Cambridge university press (2009)
32. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., Flach, P.: Face: feasible and actionable counterfactual explanations. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. pp. 344–350 (2020)
33. Price, R., Messinger, P.R.: Optimal recommendation sets: Covering uncertainty over user preferences. In: Veloso, M.M., Kambhampati, S. (eds.) Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA. pp. 541–548. AAAI Press / The MIT Press (2005), http://www.aaai.org/Library/AAAI/2005/aaai05-085.php
34. Ramakrishnan, G., Lee, Y.C., Albarghouthi, A.: Synthesizing action sequences for modifying model decisions. In: AAAI. vol. 34, pp. 5462–5469 (2020)

35. Rawal, K., Lakkaraju, H.: Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. Advances in Neural Information Processing Systems **33**, 12187–12198 (2020)
36. Russell, C.: Efficient search for diverse coherent explanations. FAT* '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3287560.3287569, https://doi.org/10.1145/3287560.3287569
37. Sheikh, M.A., Goel, A.K., Kumar, T.: An approach for prediction of loan approval using machine learning algorithm. In: 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). pp. 490–494. IEEE (2020)
38. Simon, H.A.: A behavioral model of rational choice. The quarterly journal of economics pp. 99–118 (1955)
39. Stepin, I., Alonso, J.M., Pereira-Fariña, A.C.M.: A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. IEEE Access **9**, 11974–12001 (2021)
40. Tsirtsis, S., Rodriguez, M.: Decisions, counterfactual explanations and strategic behavior. In: NeurIPS (2020), https://proceedings.neurips.cc/paper/2020/hash/c2ba1bc54b239208cb37b901c0d3b363-Abstract.html
41. Ustun, B., Spangher, A., Liu, Y.: Actionable recourse in linear classification. In: FAT*. pp. 10–19 (2019)
42. Venkatasubramanian, S., Alfano, M.: The philosophical basis of algorithmic recourse. In: Proceedings of the 2020 conference on fairness, accountability, and transparency. pp. 284–293 (2020)
43. Verma, S., Boonsanong, V., Hoang, M., Hines, K.E., Dickerson, J.P., Shah, C.: Counterfactual explanations and algorithmic recourses for machine learning: a review
44. Verma, S., Hines, K., Dickerson, J.P.: Amortized generation of sequential algorithmic recourses for black-box models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 8512–8519 (2022)
45. Viappiani, P., Boutilier, C.: Optimal Bayesian recommendation sets and myopically optimal choice query sets. Advances in neural information processing systems **23** (2010)
46. Viappiani, P., Boutilier, C.: On the equivalence of optimal recommendation sets and myopically optimal query sets. Artificial Intelligence **286**, 103328 (2020)
47. Voigt, P., Von dem Bussche, A.: The EU general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing **10**(3152676), 10–5555 (2017)
48. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harv. JL & Tech. **31**, 841 (2017)
49. Wang, Z.J., Wortman Vaughan, J., Caruana, R., Chau, D.H.: Gam coach: Towards interactive and user-centered algorithmic recourse. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. pp. 1–20 (2023)
50. Yadav, P., Hase, P., Bansal, M.: Low-cost algorithmic recourse for users with uncertain cost functions. arXiv preprint arXiv:2111.01235 (2021)
51. Yonadav, S., Moses, W.S.: Extracting incentives from black-box decisions. CoRR **abs/1910.05664** (2019), http://arxiv.org/abs/1910.05664
52. Yoo, T.K., Ryu, I.H., Lee, G., Kim, Y., Kim, J.K., Lee, I.S., Kim, J.S., Rim, T.H.: Adopting machine learning to automatically identify candidate patients for corneal refractive surgery. NPJ digital medicine **2**(1), 1–9 (2019)