Better Luck Next Time: About Robust Recourse in Binary Allocation Problems

Meirav Segal$^{1[0000-0001-7145-9159]}$, Anne-Marie George$^{1[0000-0001-9232-8211]}$,
Ingrid Chieh Yu$^{1[0009-0003-9764-3319]}$, and Christos
Dimitrakakis$^{2[0000-0002-5367-5189]}$

[1] University of Oslo, Oslo, Norway
`{meiravs,annemage,ingridcy}@ifi.uio.no`
[2] University of Neuchatel, Neuchatel, Switzerland
`christos.dimitrakakis@unine.ch`

Abstract. In this work, we present the problem of algorithmic recourse for the setting of binary allocation problems. In this setting, the optimal allocation does not depend only on the prediction model and the individual's features, but also on the current available resources, utility function used by the decision maker and other individuals currently applying for the resource. We provide a method for generating counterfactual explanations under separable utilities that are monotonically increasing with prediction scores. Here, we assume that we can translate probabilities of "success" together with some other parameters into utility, such that the problem can be phrased as a knapsack problem and solved by known allocation policies: optimal $0-1$ knapsack and greedy. We use the two policies respectively in the use cases of loans and college admissions. Moreover, we address the problem of recourse invalidation due to changes in allocation variables, under an unchanged prediction model, by presenting a method for robust recourse under variables' distributions. Finally, we empirically compare our method with perturbation-robust recourse and show that our method can provide higher validity at a lower cost.

Keywords: Counterfactual explanations · Algorithmic recourse · Allocation problems.

1 Introduction

Automated decision-making systems are currently employed in many high-risk applications such as granting loans [49] or admitting students to higher-education programs [45]. As these applications have a great impact on people's lives and future trajectories, it is important to provide individuals with explanations regarding such decisions and algorithmic recourse—actions the individual can take in order to obtain the desired outcome. A widely used approach is counterfactual explanations (CE). With this approach, an explanation model outputs a feature vector that would have obtained the desired outcome, and requires minimal changes to the original feature vector of the individual. For example, an individual who has been denied a loan might be told that they must increase their salary by 1000$ in order to qualify for a loan. As such, CE and algorithmic recourse are of the same format: the description of individual features that would yield the desired outcome. While CEs interpret this as an explanation of what the current individual is lacking, in the view of recourse the action recommendation is to obtain the described features.

The CE and recourse literature is mainly focused on binary classification settings. In these problems, a model M predicts the probability of success (e.g., of repaying the loan or graduating) for each individual and then a function $f : [0, 1] \rightarrow \{0, 1\}$ outputs a decision by setting a threshold over those scores [14, 41]. All individuals with success probability above this threshold are assigned with the desired label ('loan granted' or 'admitted') and the full classifier is $h = f \circ M$. We extend this line of work to allocation problems where budget constraints do not necessarily allow for a threshold policy. Here individuals with lower (prediction) value might be accepted because of their lower cost.

In allocation problems, a decision maker (DM) is allocating *limited* resources among a *population* in order to maximise some *objective* (such as profit to the bank), sometimes under additional constraints. The decision is determined according to the available resources, current population (or applicant pool) and the DM's utility function. Applications such as college admissions and loan granting, which are usually considered as classification problems [18, 32], are in fact dependant on resource constraints (and consequentially the whole population) and are thus better phrased as allocation problems. For example, for the lending use case, we can consider a bank making a decision every time step based on a batch of loan applications. The bank can offer a loan to a limited number of individuals depending on the bank's budget, these individuals are selected among the current set of loan applicants and the bank may have a utility function which takes into account different factors, for instance the current interest rate. Hence, it is insufficient to provide counterfactual explanations with respect to a prediction model (e.g., of the probability of repaying the loan), which usually only considers prediction accuracy. Instead, counterfactual explanations should be made with respect to the entire decision making process, i.e., the allocation problem and its variables.

All three allocation problem variables – resources, population and utility – may change over time. Following the lending example, possible changes of variables could include:

- *Resources.* The bank may have a different budget in the next time step, which would make it easier or harder to be granted a loan.
- *Applicants.* We do not expect to see the exact same population applying again for a loan.
- *Utility.* According to the current market, a bank may change their utility function to be more or less risk averse.

As the goal of acting upon a given recourse is to yield the desired outcome in the future, it is crucial that the recourse remains valid over time. That is, the individual receives the desired outcome at a later time step following the implementation of the recommended recourse. Following the above lending example, a recourse that is based on the current resources, applicants and utility may not be valid at the next time-step.

Previous works that address recourse invalidity, due to changes of the prediction model or retraining following a distribution shift, consider robustness with respect to worst-case perturbations. As a natural result, many works also demonstrate the cost-robustness trade-off, meaning that the cost of a recourse increases with its robustness guarantees. In a work by Pawelczyk et al. [41] the users are granted control over this trade-off by setting a desired validity rate. However, these robust methods do not account for the probability of these perturbations or the likelihood of the current parameters. When taking the distribution of the problem variables into account and providing recourse that is valid with high probability, the cost of the recourse might in fact decrease. A user could be simply "unlucky" in a specific allocation, but could be assigned with a favorable decision given the same features with high probability, depending on the distribution over problem variables. This information is crucial for the user, which might otherwise invest significant efforts to implement an unnecessarily difficult recourse. To this end, we model changes in allocation variables by sampling them from a known distribution and propose a distribution-aware method for robust recourse.

In this paper, we focus on binary allocation problems with monotonic separable utilities (Section 3). We then present two use cases for such problems with different allocation policies: loans and college admissions with optimal $0-1$ knapsack and greedy policies respectively (Section 4). For these problems, we provide a pipeline for generating CE under a black-box prediction model and an allocation policy (Section 5). Here we assume to have access to a CE-generator for classifiers and encapsulate this part in the pipeline. Next, we provide an algorithm for approximating a robust recourse given a distribution over the allocation variables and perform empirical analysis using budget distributions (Section 6).

Our contributions are as follows:

1. We propose allocation problems as a novel setting for considering robust algorithmic recourse.
2. In this setting, under a combination of a classification model and an allocation policy, we show through examples that the counterfactual explanations for allocations can be more reliable for static allocations compared to counterfactual explanations for the associated classification tasks.
3. For algorithmic recourse in repeated allocations, we empirically show that a distribution-aware robust recourse could reduce the cost in some cases while still providing high chances of achieving the desired outcome.

2 Related Work

Recourse for Allocations A recent survey about algorithmic recourse [24] mentions that recourse should be extended to matching problems and allocation problems. Yet, to the best of our knowledge, the problem of robust algorithmic recourse for allocation problems has not been addressed in the literature so far. The literature closest to this problem is from the field of scheduling and routing problems, where several contributions deal with explainability by answering "why-not" and "what-if" questions [9, 15, 29, 31]. Yet, most works address the end-user of the explanation as the scheduler (or employer), and do not consider the individual's point of view (employee who was assigned to tasks). One line of work considers the perspective of the individual and generates CE using inverse optimization [28]. However, this work does not address the problem of algorithmic recourse and possible changes to the problem variables and constraints. Recently, a new work discussing the impact of recourse on the distribution of future population also included a resource constraint and addressed the competition between individuals currently applying for the resource [17]. Here, the recourse definition is limited to a threshold allocation policy, the paper does not regard the DM's utility, and the need for robustness is not addressed.

Recourse Invalidation The problem of algorithmic recourse invalidation and the need for robustness has already been recognised in recent years [35]. The majority of papers consider invalidation due to model retraining with different training data, usually following a distribution shift [5, 6, 14, 20, 27, 37, 38, 42, 46]. We propose that even with the same data distribution, the differences in sampled populations from one allocation to another may lead to recourse invalidation. Moreover, we also address possible invalidation due to change of resources or utility function. The latter was identified as an open problem in a recent survey of causal machine learning [23]. Other studied causes of invalidation are change of prediction model [40] and feature perturbation, which could be due to inaccurate implementation of the recourse [11, 41, 47] or privacy perturbation [36]. We do not address these kinds of invalidation and assume that the recourse is implemented in full.

Robust Recourse Many works try to improve recourse robustness by considering the worst-case adversarial perturbation (e.g. of the data distribution) within a set of plausible changes, usually measured by distance up to a specific value [6, 11, 37, 38, 46, 47]. While these methods indeed improve the robustness of the recourse, they also present a trade-off between robustness and cost (e.g., distance of the counterfactual from the original feature vector) [40, 42, 46]. For deep networks, even if no explicit trade-off exists, the robust recourse is still presented as more costly [5]. Nonetheless, these methods do not take into account the probability of such worst cases or question whether the current variables should be used as a point of reference for increasing robustness. We present a robust recourse under the variables' distribution, which could result in lower costs for the users, compared with the worst-case recourse with respect to current variables. When considering the distribution, we can also provide the user with more control over the robustness-cost trade-off. This was proposed in a recent paper [41] assuming a specific noise distribution over recourse implementation. A similar method was also suggested for generating counterfactual explanations under uncertainty of the causal relations in the data [25]. We facilitate the same kind of control for allocation problems.

Other Approaches Ferrario and Loi [16] suggest a different approach for handling recourse invalidation. They propose a method for retraining the prediction model such that counterfactual explanations generated in the past would still hold. A similar problem to recourse robustness is the uncertainty of counterfactual examples with respect to the data distribution [3, 10, 43]. We do not address this problem, and assume that the black-box explanation model provides a reasonable counterfactual explanation with respect to the data distribution.[3]

3 Binary Allocation Problems

A binary allocation problem is a triple $\langle r, X, U\rangle$ where r represents the available quantity of the resource (such as budget), X is the given population of size n with $x_i \in \mathbb{R}^l$ being the feature vector of individual $i \in \{1, \ldots, n\}$ which includes w_i, the resource amount requested by applicant i, and U is the utility function that the DM is trying to maximise. An allocation policy π outputs a valid allocation or assignment, represented by a binary vector $Y = \{0, 1\}^n$, where $y_i = 1$ means that individual i is assigned with w_i of the resource, and $y_i = 0$ means that they are assigned with none of the resource. A valid allocation is an allocation for which the allocated quantity of the resource does not exceed the available quantity, i.e., $\sum_i y_i w_i \leq r$. In the following sections, we consider the CE, valid recourse and robust recourse to be with respect to the preferred assignment $\hat{y}_i = 1$.

Separable Utility and Prediction Model. In this paper, we focus on settings in which the DM's utility for allocation Y is separable over the population, meaning that it can be decomposed into a sum of individual utilities v_i for each person i to which a resource is allocated, i.e., $U(Y) = \sum_{i:y_i=1} v_i$. The individual utility v is the output of an individual utility function $u : \mathbb{R}^l \rightarrow \mathbb{R}$ which takes the individual's feature vector as input, i.e. $u(x_i) = v_i$. Moreover, we restrict the function u to be of a specific form – a composition of two functions $u = S_\theta \circ M$. The function $M : \mathbb{R}^l \rightarrow [0, 1]$ is a prediction model, which maps a feature vector to a single value. This can, for example, represent the success probability of repaying a loan. The function $S_\theta : [0, 1] \rightarrow \mathbb{R}$ is a monotonically increasing function parameterised by θ. The parameter θ could, for instance, represent the current interest rate. The individual utility can be interpreted as the predicted gain if we allocate the requested resource to the individual.

---

[3] We note that a problem which might be considered as related is the of use counterfactuals to explain classification uncertainty [30]. This is a different objective and in our work we do not account for prediction uncertainty.

4 Use Cases

We present two possible allocation policies for commonly used applications for high-risk decisions: lending and college admissions. Using these examples, we motivate the need for CE for binary allocation problems.

4.1 Lending

The lending use case is often seen as an example of a high-risk application of automated decision making systems [33]. In this problem, individuals apply for a loan by providing information such as requested credit, purpose of the loan, current salary and demographic information. Based on these features, the prediction model employed by the DM (in this case, the bank or lending institute) predicts the individual's probability of repaying the loan. Previous papers consider this as a classification problem, and the allocation policy to be simply setting a constant threshold over these probabilities. We formulate this problem as an allocation problem and describe our concrete modelling choices in the following. This formalism is particularly relevant for student loans in the US, where the Federal Student Aid Programs operate under a limited budget and all applications for the next academic year are submitted up to a set deadline [2].[4]

Utility Function. Following the student loan use-case, we assume that the DM's gain from each successful applicant is twofold: 1) the DM has a (monetary) profit — a constant fraction $G_1 \in [0,1]$ out of the requested credit,[5] and 2) $G_2 \in \mathbb{R}$, a value that represents the social value of granting a loan, e.g., by enabling an educational opportunity to an individual who could not have afforded this otherwise, and allowing them to increase future financial prospects. In case the individual was not able to repay the loan, the DM loses a fraction $C \in [0,1]$ of the loan. For simplicity, we assume that C is constant and has the same value for all applicants. Thus, the expected utility when granting a loan to individual i is

$$u(x_i) = M(x_i)(w_i G_1 + G_2) - (1 - M(x_i))C w_i. \tag{1}$$

Allocation Policy. The DM is trying to maximise utility under budget constraints, where each applicant has individual utility and desired credit. This problem can be translated to the well known $0-1$ knapsack problem [4]. Here, the weight capacity of the knapsack is the budget r, we have n items (individuals), each item i has value $v_i = u(x_i)$ and a weight w_i. Items with negative utility can be removed since including them cannot increase the allocation utility. Considering weights and values to be non-negative, the problem is given

---

[4] Other examples of such allocation problems, with a limited budget and applicants requesting different quantities in batches, include funding agencies and grant applications.

[5] In practice, the utility function also depends on the time for which the loan is requested, but we ignore this component for simplicity.

by $\max \sum_{i=1}^{n} v_i y_i$ s.t. $\sum_{i=1}^{n} w_i y_i \leq r$, i.e., filling the "knapsack" with the most value while respecting its capacity. This constrained optimisation problem is NP-complete, yet solvable in pseudo-polynomial time using dynamic programming. Note that we assume discretisation: the credit has a minimal step size (e.g. 100$). We therefore assume that the DM's allocation policy for this application is determined by the optimal solution.

Table 1. Motivating example: lending use case. Table of applicants with their predicted probability of repaying a loan according to model M and their features, the individual utility for the DM according to equation 1 and the credit each applicant is requesting (in thousands of dollars).

Applicant	$M(x_i)$	$u(x_i)$	**Credit** (w_i)
1	0.8	0.8	4
2	0.7	0.625	3
3	0.6	0.5	2
4	0.5	0.425	1

Motivating Example Consider the applicants described in Table 1 under the utility function $u(x_i) = M(x_i)(w_i(G_1 + C) + G_2) - Cw_i$ (equation 1 rearranged) using the parameters $G_1 = 0.05, G_2 = 1, C = 0.2$ and budget of 6 (thousand dollars). The optimal allocation is $Y = (0, 1, 1, 1)$, meaning approving the loan for applicants $2, 3$ and 4 with utility of 1.55 for the DM. Note that applicant 1 was not selected, even though their probability of repaying the loan is higher than that of the other applicants, as well as their individual utility for the DM. Thus, it would be difficult to explain the decision when only considering the prediction model, without the allocation mechanism, remaining population and budget constraint.

4.2 College Admission (Greedy)

College admission is a highly researched problem [22], and it concerns educational opportunities. As such, it is regarded as a high-risk application for which individuals are entitled to an explanation (according to the European AI Act [33]).

In the case of college admission, individuals apply by providing educational and demographic background information. Based on these features, the prediction model employed by the DM (i.e., the university) predicts the individual's probability of graduation. For college admissions, the weight or requested quantity is identical for all candidates (one admission slot $\forall i, w_i = 1$). Thus, the use of a threshold policy would be optimal, assuming that the utility of the allocation is the sum of individual utilities. Nonetheless, while previous papers set the threshold over the graduation probabilities, for allocation problems the threshold would be over the individual utilities, and applicants are greedily added as long as the utility increases and the assignment does not exceed the budget [8]. This means that individuals with non-positive utility will not be included in the admitted set regardless of the budget constraint. Hence, the decision also depends on the resource and the utility function, which might lead to different results from those we get using only a predictor.

Utility Function. Let us assume that for every admitted student, the university pays a constant cost $C \geq 0$. In addition, for every admitted student who successfully graduates, the university receives a constant reward or gain $G \geq 0$. Thus, the expected (individual) utility for an admitted student i is

$$u(x_i) = M(x_i)(G - C) + (1 - M(x_i))(-C) = M(x_i)G - C. \tag{2}$$

Again, we denote $u(x_i)$ as v_i.

Table 2. Motivating example: college admission. Table of applicants with their predicted probability of graduating according to model M and their features, and the individual utility for the DM.

Applicant	$M(x_i)$	$u(x_i)$
1	0.8	0.2
2	0.7	0.1
3	0.6	0.0
4	0.5	-0.1

Motivating Example Let us consider the utility function from equation 2 with $G = 1$ and $C = 0.6$ as shown in Table 2. For the case of $r = 2$ (two free study slots), the optimal allocation would be to select the top two applicants: 1 and 2 ($Y = (1, 1, 0, 0)$). In this case, in order to be selected, applicant 3 should increase their utility such that it is higher than that of applicant 2. For the case of $r = 3$, the optimal allocation would also be $Y = (1, 1, 0, 0)$, since selecting applicant 3 would not increase the utility. Thus, in order to be selected, it is sufficient for applicant 3 to increase their utility by $\epsilon > 0$. This could be a significant difference of investment cost for the applicant, which could not be captured by considering the prediction model alone.

5 Counterfactual Explanations

We start off by giving a formal definition of counterfactual explanations that is based on the definition of counterfactual explanation for classification problems [19]. We then describe how to generate these in our specific setting.

Definition 1 (Counterfactual Explanation for Binary Allocations). *Given an allocation policy π that outputs the decision Y for population X, utility* *function U and given resource r, a counterfactual explanation for individual $x_i \in X$ consists of an alternative vector of features x' for which the allocation $Y' = \pi(r, X \cup \{x'\} \setminus \{x_i\}, U)$ is different from Y such that $y'_i = 1$. We define such a counterfactual explanation to be minimal if its cost to the individual $d(x_i, x')$ is minimal under some metric $d : \mathbb{R}^l \times \mathbb{R}^l \to \mathbb{R}$.*

Note that here, there could be another individual $j \neq i$ for which $y'_j \neq y_j$, meaning that the CE might change the assignment for other individuals and not only the individual requesting the CE.

Assume we are given a prediction model M, an allocation policy π, an individual utility function $u = S_\theta \circ M$ such that S_θ is a monotonically increasing function, a population X, resources r and a metric d in the feature space. We propose to generate a CE according to the pipeline below.

1) Computing the Minimal Utility-CE. Given an allocation policy, we first produce a *minimal utility-CE* v', i.e., the minimal utility that would have led to a preferred assignment. For the two allocation policies we focus on:

- **Optimal** $0-1$ **knapsack policy.** Intuitively, the individual utility should increase by the difference between the current maximal allocation utility and the maximal allocation utility under the constraint of including individual i. We denote the optimal allocation for applicant set $[n]$ and available resources r as $Y^*([n], r)$. We claim that the minimal utility-CE for individual i is $v'_i = U(Y^*([n], r)) - U(Y^*([n] \setminus i, r - w_i))$, where U is the utility of the allocation. A proof for this result and additional notes can be found in the appendix. We can thus use a dynamic programming algorithm [6] for $0-1$ knapsack, see e.g., [34], to compute the minimal utility-CE. In practice, to avoid ties we set $v'_i = U(Y^*([n], r)) - U(Y^*([n] \setminus i, r - w_i)) + \epsilon$ for some $\epsilon > 0$.
- **Greedy policy.** Following the example in Table 2, the utility-CE for individual i is either 1) larger than the utility of the selected applicant with the smallest utility, in case the budget was fully utilised or 2) larger than 0 in case there are still available vacancies. Formally, we propose for any $\epsilon > 0$:

$$v'_i = \begin{cases} (\min_{j:y_j=1} v_j) + \epsilon, & \text{if } \sum_j y_j = r \\ \epsilon, & \text{otherwise} \end{cases}$$

2) Computing a Prediction-CE. The minimal utility-CE is translated to a *prediction-CE* m', i.e., the minimal success probability that would have led to a preferred assignment. Because S_θ is monotonically increasing, it is also invertible. Then the prediction-CE is $m' = S_\theta^{-1}(v')$.

3) Computing a Minimal (Feature-Based) CE. Using the prediction-CE, a minimal CE x' is generated by solving the following optimisation problem:

$$x' = \arg\min_z d(z, x) \quad \text{s.t.} \quad M(z) \geq m'. \tag{3}$$

---

[6] Simply put, a table V of size $n \times r$ is being filled. Each cell $V[i, j]$ holds the value of the maximal utility that can be obtained given items $1, \ldots, i$ and maximal weight j.

For example, we can construct the function $h_{m'}$ with $h_{m'}(x) = 1$ if $M(x) \geq m'$ and $h_{m'}(x) = 0$ otherwise. Then, one of the many existing explanation models for classifiers [19, 39] (e.g., [48]) can be used on $h_{m'}$ with metric d, which provides x', a minimal CE with respect to the feature-based cost function d. Note that we consider w_i to be fixed.

At the end of this process, x' is minimal with respect to d and $M(x') \geq m'$. Hence, x' is a minimal CE for the allocation problem under the following assumptions: 1) the utility function is monotonic in the prediction scores, and 2) the allocation policy is monotonic in the utility, i.e., increasing the utility for an individual assigned with the resource could never change the allocation such that the individual is not assigned with the resource. Both policies (optimal $0-1$ knapsack and greedy) satisfy these monotonicity assumptions.

To mitigate the effect of specific classification explanation choices in step 3), we can define the CE in terms of success probability or prediction score (prediction-CE). In the remainder of the paper, we assume the cost function is defined with respect to the predicted probability of success: $d_M(M(x_i), m') = |M(x_i) - m'|$.

Using our proposed method, we can see that for the example in table 1 the optimal allocation under the constraint of including applicant 1 is $Y' = (1, 0, 1, 0)$ with utility of 1.3. Hence, applicant 1 should increase their individual utility to be at least $1.55 - 0.5 = 1.05$ which translates to increasing their probability of repaying the loan from 0.8 to 0.925.

6 Robust Recourse for Binary Allocations

Counterfactual explanations are used to explain the current decision, but for repeated settings, we wish to provide recommendations for the future, i.e. recourse. We assume that the available resources, population and utility function may change from one time step to the next, which may lead to invalidation of the recourse. We first define (robust) recourse for binary allocations under variable distributions, then describe how to generate approximate robust recourse and lastly evaluate this in our experiments.

Definition 2 (Valid Recourse for Repeated Binary Allocations). *At time t_1, given the allocation instance $\langle r_{t_1}, X_{t_1}, U_{t_1} \rangle$, a* recourse *for individual $x_i \in X_{t_1}$ consists of an alternative vector of features x'. This recourse is* valid *at time $t_2 > t_1$ if for the new allocation instance $\langle r_{t_2}, X_{t_2} \in \mathbb{R}^{n-1,l}, U_{t_2} \rangle$, the allocation policy outputs the allocation $Y^{t_2} = \pi(r_{t_2}, X_{t_2} \cup \{x'\}, U_{t_2})$ such that $y_i^{t_2} = 1$. A recourse is said to be* minimal *(w.r.t. d) if its cost $d(x_i, x')$ is minimal under the cost metric $d : \mathbb{R}^l \times \mathbb{R}^l \to \mathbb{R}$.*

We assume that at each time step the available resources, applicants and utility function are sampled i.i.d. according to a joint distribution D. Using this distribution, we follow the approach of Pawelczyk et al. [41] and allow the user to control the robustness-cost trade-off by providing a validity probability $\rho \in [0, 1]$.

Definition 3 (ρ-Robust Recourse for Binary Allocations). *Let x' be a recourse generated at time t_1 for individual i given an allocation problem. Given distribution D over resources, applicants and utility function, x' is ρ-robust if the expected validity at time $t_2 > t_1$ is at least ρ, i.e.,*

$$\mathbb{E}_{r_{t_2}, X_{t_2}, U_{t_2} \sim D}[\mathbb{1}_{x' \text{ valid for } \langle r_{t_2}, X_{t_2}, U_{t_2} \rangle}] \geq \rho$$

where $\mathbb{1}[\cdot]$ is an indicator function. Among all ρ-robust recourses, a recourse with minimal cost $d(x_i, x')$ is denoted as a minimal ρ-robust recourse.

Note that we assume here that the allocation policy is constant, yet this could also be relaxed and added to the sampled variables.

Interestingly, under our definition, a robust recourse may be of cost 0, depending on the distribution and the initial allocation variables. For example, the recourse might have been generated under an extremely unlikely combination of variables, so that the individual was simply “unlucky”.

6.1 Approximated Robust Recourse

Algorithm 1 Approximated ρ-Robust Recourse

```
Require: sample size $n > 0$, prediction model $M$, feature-based explanation function $E$, allocation problem $\langle r, X, u \rangle$, allocation policy $\pi$, distribution $D$ over resources, applicants and utility parameters $\{(r_j, X_j, \theta_j)\}_{j \in [n]}$, individual $i$, desired validation level $\rho \in [0, 1]$.
  for $j$ from 1 to $n$ do
    Get sampled variables $(r_j, X_j, \theta_j) \sim D$
    Get utility-CE $v'_j$ with respect to $\langle r_j, X_j, u_j = S_{\theta_j} \circ M \rangle$ and policy $\pi$
    Get prediction-CE $m'_j = S^{-1}_{\theta_j}(v'_j)$
  end for
  Sort all prediction-CE: sorted $\leftarrow$ sort($[m'_j]_{j=1}^n$)
  Get $\rho$-robust prediction-CE $m_\rho \leftarrow$ sorted[$\lceil \rho n \rceil$]
  return feature-based CE $x' = E(M, i, m_\rho)$
```

We approximate the ρ-robust recourse for binary allocations, a monotonic separable utility and a monotonic policy using a Monte-Carlo approximation (see Algorithm 1). Given a prediction model M, an allocation policy π, distribution D over resource r, applicants X and utility function parameter θ, for each allocation problem $\langle r, X, u = S_\theta \circ M \rangle$ such that $(r, X, \theta) \sim D$, we can generate a minimal prediction-CE for individual i as shown in Section 5. Given the minimal prediction-CE for n sampled problems, we can find m_ρ, the prediction-CE that is valid for at least ρ of the sampled allocation problems. Such m_ρ exists as the allocation is monotonic with respect to the prediction score: for every allocation problem which requires individual i to have a prediction score of m in order to receive the resource, any larger prediction score $q > m$ would also guarantee the resource being allocated to i. As we can estimate the distribution's quantiles using Monte Carlo approximation [12], this m_ρ approximates the validity over the entire distribution.[7] This ρ-robust prediction-CE can then be translated to features, as was proposed in step 3 in Section 5. The produced feature vector x' is then the minimiser of

$$\min_z d(z, x_i) \quad \text{s.t.} \quad \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}_{M(z)>m'_j} \geq \rho. \tag{4}$$

Here, m'_j is the j-th prediction-CE. Note that it is sufficient to sort the thresholds, as is done in Algorithm 1. Hence, x' is the feature vector with the lowest cost w.r.t. d which provides individual i with the resource in approximately ρ of the allocations. We note that by using the intermediate step of prediction-CE, we reduce the problem to a one-dimensional monotonic recourse. Without this step, for each sampled allocation problem we would generate a different feature-based CE x'. We do not assume the prediction model M to be monotonic in the features, i.e., a specific value of feature j in x' does not guarantee that all feature vectors with a higher value for feature j would have a greater or equal prediction score.

6.2 Experiments

We empirically evaluate the performance of our robust recourse method in terms of cost and validity. We focus on the case of a changing budget, assuming that the utility of the DM is fixed and the recourse is generated with respect to the current population. In our experiments we produce prediction-CE or prediction-recourse, and measure the recourse cost with respect to the difference in prediction score. As there is no other method for generating a CE for allocation problems, we cannot compare our results with previous methods. Thus, the goal of the empirical results is twofold:

1. Evaluate the cost and validity of the robust-recourse compared to the static CE.
2. Compare our approach of distribution-aware robust-recourse to the previously proposed approach of perturbation-based robust recourse.

Datasets and Preprocessing

Loans We use the German credit dataset [13] which is one of the most common benchmarks used for CE and algorithmic recourse (e.g. [5, 6, 14, 20]). The dataset consists of 1,000 samples with 20 features such as age, marital status, education, savings and requested credit. A binary label indicates whether the

---

[7] The accuracy of the approximation depends on the sample size which we consider to be fixed. However, our method could be extended to include a parameter to control the required sample size.

candidate repaid the loan. We scale numeric features to $[0, 1]$ and encode categorical features as 1-hot vectors. In addition, the requested credit is divided by 100 and rounded. The data is split to train and test sets with the ratio of $70-30$. Then, a random forest classifier with 200 trees is trained on the train set (achieves accuracy of 0.78 on the test set). We construct 20 allocation problems by uniformly sampling 20 individuals from the test set, set the utility function parameters to $G_1 = 0.06, G2 = 4, C = 0.5$ using the utility function in equation 1, and sample a budget from the budget distribution. The utility parameters are set such that the number of individuals with a positive utility is close to the number of individuals who were granted a loan based on the train set.

College Admissions We sample applicant features and their success probability for $10,000$ applicants according to the simulator described in [26], using data from the Norwegian Database for Statistics on Higher Education [1]. We construct 20 allocation problem by uniformly sampling 500 individuals from the simulated data, set the utility function parameters to $G = 2, C = 1$ using the utility function in equation 2, and sample a budget from the budget distribution. The utility parameters are set such that an individual with success probability higher than 0.5 will have a positive utility.

Budget Distributions For both datasets, we sample 50 batches of applicants (20 applicants for German credit and 500 for college admissions) and consider the sum of given credit or admitted students as the current budget or capacity. We then fit a normal distribution to it, and consider this as the budget distribution.

Method and Baselines We test our ρ-robust recourse method, described in Algorithm 1, with $\rho \in \{0.7, 0.9\}$, with 200 budget samples, which we denote as the validation set. We compare our results to the static prediction-CE for allocations. In addition, we implement another recourse method we denote as *p-noisy*. This method is designed to be of a similar nature to perturbation robustness. According to this method, given an allocation problem with a specific budget r, and an individual i, we generate a validation set by sampling 200 values from a truncated normal distribution $\nu_j \sim \mathbb{N}(0, \sigma^2)_{[a,b]}, j \in [200]$. Then, we generate the minimal prediction-CE for all budgets $\{r + \nu_j\}_{j\in[200]}$. The p-noisy robust recourse is the maximal among them. The parameter p controls the range $[a, b]$ such that p of the values lie according to distribution $\mathbb{N}(0, \sigma^2)$ in the range $[a, b]$. We set σ to be the standard deviation of the underlying variable (budget) distribution. In our experiments we use $p \in \{0.7, 0.9\}$. Moreover, we define an optimistic baseline which is a ρ-robust recourse generated based on the test budget samples. For this baseline we set $\rho = 1$, so that the generated recourse is valid for the entire test set.

Our assumption is that the test set would be more similar to the validation set used by the ρ-robust method (sampled from the same distribution), rather than the single sample of the original variable used by the static CE, or the validation set used by the noisy recourse method. Therefore, we expect to see that our method can achieve better results in terms of cost and validity compared to the baselines.

Results For each allocation problem, we find the optimal allocation and provide recourse for all individuals not included in the allocation. The results are described in Table 3. The recourse validity of each individual is measured as the average validity over a test set of 200 samples from the budget distribution. The validity of the method is then the average validity across all individuals. The recourse cost for each method is the average prediction score difference. We normalise all costs by the cost of the optimistic baseline.

Table 3. Robust recourse under resource distribution.

	Loans		Admissions	
Method	**Cost**	**Validity**	**Cost**	**Validity**
Static CE	0.42	0.823	0.756	0.679
0.7-robust	**0.407**	0.84	**0.753**	0.771
0.9-robust	0.51	0.917	0.796	0.922
0.7-noisy	0.571	0.888	0.812	0.845
0.9-noisy	0.649	0.977	0.84	0.952
Optimistic	1	**1**	1	**1**

From the results in table 3, we can observe that as expected, higher ρ or noise values achieve higher validity at a higher cost. We can also observe that the 0.7-robust method Parto-dominates the static-CE for both applications, as it achieves higher validity at a lower cost. This shows that the budgets of some of the allocation problems did not represent the test set and produced a higher-cost prediction-CE. Similarly, the 0.9-robust method Pareto-dominates the 0.7-noisy method in both applications. We can also observe that the ρ-robust methods are never Pareto-dominated by any other. This shows the advantage of our distribution-based robust-method.

When considering a single individual, by increasing the validity we also increase the cost of the recourse. This is due to our monotonicity assumption for the utility function and the allocation policy. However, when considering the average over the population and the test set, we can see it is possible for our method to achieve higher validity at a lower cost. This could be explained by the fact that the validation set is more likely to represent the test set. When the original variable is more permissive, allowing resource allocation to more individuals, our method can provide a recourse that would be valid for more restricting samples of the distribution. Thus increasing the average validity and the average cost. When the original variable hinders resource allocation, our method would be able to find “unlucky” individuals that do not require a costly recourse (or recourse at all) to be allocated with the resource for many variable values. Thus, the average cost would be reduced and the validity would remain high.

Another observation we can make from the experimental results, is the difference between the validity on the test set and the requested validity. This gap can be explained by the fact that the validity is estimated based on the validation set and the final validity is computed based on the test set. Since the two sets are not identical, the recourse for which the estimated validity was ρ (the requested validity) may provide lower or higher validity on the test set. In addition, it is possible that the minimal recourse for the requested validity level already provides a higher validity. For example, let us assume a user is requesting 0.5 validity and the validation set produces the following minimal CE: $(0.1, 0.2, 0.01, 0.2, 0.2, 0.25)$. If we wish to provide 0.5 validity, we must have a recourse of 0.2, but that recourse already provides us with a higher validity of 0.83. This could explain the fact that all methods provide test-set validity that is higher than the requested validity.

7 Generalisation and Open Problems

In this paper we only make the first step in solving this new setting of recourse for allocation problems. We address allocation problems with binary decisions and separable utilities. More complex problems within the scope of allocation problems could be addressed in the future. We propose here more general definitions for CE and recourse for a wider class of allocation problems and point out interesting aspects of these problems.

7.1 Definitions

We start by providing a general definition of an allocation problem.

Definition 4 (Allocation Problem). *An allocation problem is a triple* $\langle R, X, U\rangle$ *where* $R = \{r_j\}_{j=1}^{k}$ *represents the available resources, with* r_j *being the number of units available of resource* j*,* X *is the given population of size* n *with* $x_i \in \mathbb{R}^l$ *being the feature vector of individual* i*, and* U *is the utility function that the DM is trying to maximise.*[8] *An allocation policy* π *outputs a valid allocation or assignment, represented by a matrix* $A \in \mathbb{R}^{n,k}$ *such that* $a_{i,j}$ *is the number of units of resource* j *allocated to individual* i*. A valid allocation is an allocation that satisfies* $\forall j \in [k], \sum_{i=1}^{n} a_{i,j} \leq r_j$*, meaning that the sum of allocated resources is at most the set of available resources. The DM is then trying to find a policy which maximizes the utility function* $U : \mathbb{R}^{n,k} \longrightarrow \mathbb{R}$*.*

Next, we provide a definitions of counterfactual explanations and valid recourse for general allocation problems.

Definition 5 (Counterfactual Explanation for Allocations). *Given an allocation policy* π *that outputs the decision* A *for population* X*, utility function*

---

[8] The allocation problem could also be defined as $\langle R, X, U, C\rangle$ where C represents additional constraints. The feature vector x_i could also include preferences over the resources $[pref(i)]_{i=1}^{k} \in \mathbb{R}^k$.

U and given resources R, a counterfactual explanation for individual $x_i \in X$ with respect to a preferred allocation or assignment for individual i: $\hat{a}_i \in \mathbb{R}^k$ consists of an alternative vector of features x' for which the allocation $A' = \pi(R, X \cup \{x'\} \setminus \{x_i\}, U)$ is different from A such that $a'_i = \hat{a}_i$. We define such a counterfactual explanation to be minimal if its cost $d(x_i, x')$ is minimal under some metric $d : \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}$.

Definition 6 (Valid Recourse for Sequential Allocations). *At time t_1, given the allocation variables $R_{t_1}, X_{t_1}, U_{t_1}$, a recourse for individual $x_i \in X_{t_1}$ with respect to a preferred allocation $\hat{a}_i \in \mathbb{R}^k$ consists of an alternative vector of features x'. This recourse is valid at time $t_2 > t_1$ if given the new set of allocation variables at t_2: $R_{t_2}, X_{t_2} \in \mathbb{R}^{n-1,l}, U_{t_2}$, the allocation policy outputs the allocation $A^{t_2} = \pi(R_{t_2}, X_{t_2} \cup \{x'\}, U_{t_2})$ such that $a_i^{t_2} = \hat{a}_i$. A recourse is said to be minimal if its cost $d(x_i, x')$ is minimal under some metric $d : \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}$.*

Note that if the preferred resources are not in R_{t_2}, a valid recourse for t_2 does not exist.

7.2 Open Problems

Recourse for Non-binary Decisions The problem of recourse for non-binary allocations is closely related to the problem of CE and recourse for regression [44, 21] (for a single resource) and multi-class predictions [7] (for multiple resources). Although some contributions have been made in that respect, these are still open problems. In our definitions, we wish to provide CE and recourse with respect to the individual's preferred outcome. When the allocation problem is not binary, it requires additional information regarding the individual's preferences, and possibly a cost function that takes these preferences as input.

Non-separable Utility In this paper, we only address separable utility functions. Yet, as decisions are being made over batches and not over individuals, the use of non-separable utilities might be needed in some cases. For example, the probabilities of people repaying their loan might not be independent. They might, e.g., be influenced by sectoral or global crises. Thus, a decision maker might assign a higher utility to allocations with a sectoral balance, which cannot be represented by separable utilities. In these cases, it is even more important to consider the rest of the population when providing CE and recourse.

Recourse Feedback and Multi-agent Recourse A mostly unexplored interesting facet of recourse for allocation problems is the fact that an implementation of a recourse by one individual might impact the allocation outcome for other individuals. This question was addressed via an empirical simulation study [17], in which the authors measure the effect of different parameters on recourse validity or "recourse reliability". Yet, the provided recourse did not take into account the feedback effect of the recourse implementation.

8 Discussion

In this paper, we present the first attempt to define robust recourse for allocation problems. Under this setting, we show two examples of allocation policies for which methods for generating CE given a classifier would fail to explain the decision. For repeated allocations, we provide a distribution-aware method for generating robust recourse, as opposed to other methods which consider perturbations of the current problem variables. This approach allows for a recourse which might provide the user with high enough validity at the price of a lower cost. Moreover, our approach grants the user more control over the cost-robustness trade-off by choosing the requested validity probability.

Assumptions and Limitations Our proposed method assumes full knowledge of the utility function structure and the allocation policy. These are reasonable assumptions when considering that the DM is the one providing the recourse. Moreover, we make no assumptions regarding the prediction model and address it as a black-box. In addition, we assume a specific structure of the individual utility function: composition of a parametric function S_θ and a prediction model M, where S is monotonically increasing. As illustrated in Section 4, this structure is reasonable in some applications. However, it fails to capture other interesting applications in which the utility is affected directly by features. For example, for allocating research grants, the utility of a project may depend on the specific topic or planned collaborations, not only on the success probability of the proposed project. Our pipeline for generating CE and robust recourse does not provide a solution for these cases and an extension is left for future work.

We assume the allocation variables are sampled i.i.d. from a static distribution. When the true variable distribution is unknown, we can maintain a belief over the distribution and sample from the posterior to compute the robust recourse. Furthermore, this process can be adapted to consider changes in the underlying distribution over time. We also assume a constant population size, but that could be easily changed.

Our methods and definitions assume that the user's requested resource remains unchanged. Yet, it could be reasonable for an individual to change their requested resource, for example in exchange for increasing their probability of receiving it. A CE which includes change of preferences is left for future work.

Future Work Further research is required to explore the generalisation of our results to settings in which the assumptions mentioned above are relaxed. In addition, more complex problems within the scope of allocation problems, such as matching problems, could be addressed in the future, as well as the open problems mentioned in Section 7.

Societal Impact Lastly, we note that the use of recourse is intended to provide users with more control over aspects in their lives controlled by automated decisions. Ideally, we would like the DMs to be held responsible for their provided recourse, such that the implementation of a recourse would guarantee access to the resource in the future. We choose to provide a probabilistic recourse in order to grant users more control over the robustness-cost trade-off. Yet, when the recourse takes a probabilistic nature, DMs might distance themselves from the responsibility for the robustness of the recourse.

Acknowledgments. This work was supported by the Research Council of Norway under project number 302203.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Appendix

In Section 5 we claim the following:

Lemma 1. *The minimal utility-CE* v'_i *for individual* i *under an optimal* $0-1$ *knapsack policy is*

$$v'_i = U(Y^*([n], r)) - U(Y^*([n] \setminus i, r - w_i)).$$

Proof. We prove this claim by contradiction. Suppose not, and let us assume that there exists $\bar{v_i} < v'_i$ such that for $\bar{v_i}$, individual i is included in the optimal set. We assume that $w_i \leq r$ (otherwise individual i could never be included in the allocation). As the order of the individuals does not change the optimal allocation, let us assume w.l.o.g. that individual i is the last individual inserted into table V $(i = n)$. Thus, when filling the cell $V[n, r]$ we choose whether to include individual n or not: $V[n, r] = \max(V[n-1, W], V[n-1, r - w[n]] + \bar{v_n})$. By assuming that the individual is included, we get that

$$\begin{aligned}
&V[n-1, r] \leq V[n-1, r - w_n] + \bar{v_n} \\
&\Leftrightarrow U(Y^*([n], r)) \leq U(Y^*([n-1], r - w_n)) + \bar{v_n} \\
&\Leftrightarrow U(Y^*([n], r)) - U(Y^*([n-1], r - w_n)) \leq \bar{v_n} \\
&\Leftrightarrow U(Y^*([n], r)) - U(Y^*([n] \setminus i, r - w_i)) \leq \bar{v_n} \\
&\Leftrightarrow v'_n \leq \bar{v_n}
\end{aligned}$$

Which contradicts our assumption of $\bar{v_i} < v'_i$. Note that $U(Y^*([n], r)) = V[n-1, r]$ as the individual was not originally included in the allocation. In practice, we add $\epsilon > 0$ to the utility-CE in order to avoid ties.

Notes:

1. Another approach for generating CE for the $0-1$ knapsack problem was previously proposed [28]. Yet, our approach allows efficient calculation of multiple CE for different budgets by filling the table V (both with and without individual i) for a maximal budget r_{max}, which then provides all solutions for all $r \in [r_{max}]$.

2. In some cases, the required utility-CE would entail a prediction-CE that is grater than 1, which is impossible. Thus, in those cases, the applicant would learn that given the current allocation variables, there is nothing they could have changed in order to have received the requested loan. Nevertheless, we only consider the option to change user features excluding the requested credit, assuming the requested credit cannot be changed.

References

1. Database for statistics on higher education (database for statistikk om høyere utdanning). https://dbh.hkdir.no/, accessed: 2024-04-15
2. Federal student aid in the u.s. department of education website. https://www2.ed.gov/about/offices/list/fsa/index.html?exp=6, accessed: 2024-04-15
3. Ali, G., Al-Obeidat, F., Tubaishat, A., Zia, T., Ilyas, M., Rocha, A.: Counterfactual explanation of bayesian model uncertainty. Neural Computing and Applications pp. 1–8 (2021)
4. Assi, M., Haraty, R.A.: A survey of the knapsack problem. In: 2018 International Arab Conference on Information Technology (ACIT). pp. 1–6. IEEE (2018)
5. Black, E., Wang, Z., Fredrikson, M., Datta, A.: Consistent counterfactuals for deep models. arXiv preprint arXiv:2110.03109 (2021)
6. Bui, N., Nguyen, D., Nguyen, V.A.: Counterfactual plans under distributional ambiguity. arXiv preprint arXiv:2201.12487 (2022)
7. Carlevaro, A., Lenatti, M., Paglialonga, A., Mongelli, M.: Multi-class counterfactual explanations using support vector data description (2023)
8. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms. MIT press (2022)
9. Čyras, K., Letsios, D., Misener, R., Toni, F.: Argumentation for explainable scheduling. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 2752–2759 (2019)
10. Delaney, E., Greene, D., Keane, M.T.: Uncertainty estimation and out-of-distribution detection for counterfactual explanations: Pitfalls and solutions. arXiv preprint arXiv:2107.09734 (2021)
11. Dominguez-Olmedo, R., Karimi, A.H., Schölkopf, B.: On the adversarial robustness of causal algorithmic recourse. In: International Conference on Machine Learning. pp. 5324–5342. PMLR (2022)
12. Dong, H., Nakayama, M.K.: A tutorial on quantile estimation via monte carlo. In: International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing. pp. 3–30. Springer (2018)
13. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
14. Dutta, S., Long, J., Mishra, S., Tilli, C., Magazzeni, D.: Robust counterfactual explanations for tree-based ensembles. In: International Conference on Machine Learning. pp. 5742–5756. PMLR (2022)
15. Eifler, R., Frank, J., Hoffmann, J.: Explaining soft-goal conflicts through constraint relaxations. In: ICAPS 2022 Workshop on Explainable AI Planning (2022)
16. Ferrario, A., Loi, M.: The robustness of counterfactual explanations over time. IEEE Access (2022)

17. Fonseca, J., Bell, A., Abrate, C., Bonchi, F., Stoyanovich, J.: Setting the right expectations: Algorithmic recourse over time. In: Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization. pp. 1–11 (2023)
18. Goyal, A., Kaur, R.: A survey on ensemble model for loan prediction. International Journal of Engineering Trends and Applications (IJETA) **3**(1), 32–37 (2016)
19. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. Data Mining and Knowledge Discovery pp. 1–55 (2022)
20. Guo, H., Jia, F., Chen, J., Squicciarini, A., Yadav, A.: Rocoursenet: Distributionally robust training of a prediction aware recourse model. arXiv preprint arXiv:2206.00700 (2022)
21. Hada, S.S., Carreira-Perpiñán, M.Á.: Exploring counterfactual explanations for classification and regression trees. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 489–504. Springer (2021)
22. Hakimov, R., Kübler, D.: Experiments on centralized school choice and college admissions: a survey. Experimental Economics **24**, 434–488 (2021)
23. Kaddour, J., Lynch, A., Liu, Q., Kusner, M.J., Silva, R.: Causal machine learning: A survey and open problems. arXiv preprint arXiv:2206.15475 (2022)
24. Karimi, A.H., Barthe, G., Schölkopf, B., Valera, I.: A survey of algorithmic recourse: contrastive explanations and consequential recommendations. ACM Computing Surveys (CSUR) (2021)
25. 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)
26. Kleine Buening, T., Segal, M., Basu, D., George, A.M., Dimitrakakis, C.: On meritocracy in optimal set selection. In: Equity and Access in Algorithms, Mechanisms, and Optimization, pp. 1–14 (2022)
27. König, G., Freiesleben, T., Grosse-Wentrup, M.: A causal perspective on meaningful and robust algorithmic recourse. arXiv preprint arXiv:2107.07853 (2021)
28. Korikov, A., Beck, J.C.: Counterfactual explanations via inverse constraint programming. In: 27th International Conference on Principles and Practice of Constraint Programming (CP 2021). Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2021)
29. Lerouge, M., Gicquel, C., Mousseau, V., Ouerdane, W.: Counterfactual explanations for workforce scheduling and routing problems. In: 12th International Conference on Operations Research and Enterprise Systems. pp. 50–61. SCITEPRESS-Science and Technology Publications (2023)
30. Ley, D., Bhatt, U., Weller, A.: Diverse, global and amortised counterfactual explanations for uncertainty estimates. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 7390–7398 (2022)
31. Ludwig, J., Kalton, A., Stottler, R.: Explaining complex scheduling decisions. In: IUI Workshops (2018)
32. Lux, T., Pittman, R., Shende, M., Shende, A.: Applications of supervised learning techniques on undergraduate admissions data. In: Proceedings of the ACM International Conference on Computing Frontiers. pp. 412–417 (2016)
33. Madiega, T.: Artificial intelligence act. European Parliament: European Parliamentary Research Service (2021)
34. Martello, S., Toth, P.: Knapsack problems: algorithms and computer implementations. John Wiley & Sons, Inc. (1990)

35. Mishra, S., Dutta, S., Long, J., Magazzeni, D.: A survey on the robustness of feature importance and counterfactual explanations. arXiv preprint arXiv:2111.00358 (2021)
36. Mochaourab, R., Sinha, S., Greenstein, S., Papapetrou, P.: Robust counterfactual explanations for privacy-preserving svm. In: International Conference on Machine Learning (ICML 2021), Workshop on Socially Responsible Machine Learning (2021)
37. Nguyen, D., Bui, N., Nguyen, V.A.: Distributionally robust recourse action. arXiv preprint arXiv:2302.11211 (2023)
38. Nguyen, T.D.H., Bui, N., Nguyen, D., Yue, M.C., Nguyen, V.A.: Robust bayesian recourse. In: Uncertainty in Artificial Intelligence. pp. 1498–1508. PMLR (2022)
39. Pawelczyk, M., Bielawski, S., Heuvel, J.v.d., Richter, T., Kasneci, G.: Carla: a python library to benchmark algorithmic recourse and counterfactual explanation algorithms. arXiv preprint arXiv:2108.00783 (2021)
40. Pawelczyk, M., Broelemann, K., Kasneci, G.: On counterfactual explanations under predictive multiplicity. In: Conference on Uncertainty in Artificial Intelligence. pp. 809–818. PMLR (2020)
41. Pawelczyk, M., Datta, T., Van den Heuvel, J., Kasneci, G., Lakkaraju, H.: Probabilistically robust recourse: Navigating the trade-offs between costs and robustness in algorithmic recourse. In: The Eleventh International Conference on Learning Representations (2022)
42. Rawal, K., Kamar, E., Lakkaraju, H.: Algorithmic recourse in the wild: Understanding the impact of data and model shifts. arXiv preprint arXiv:2012.11788 (2020)
43. Schut, L., Key, O., Mc Grath, R., Costabello, L., Sacaleanu, B., Gal, Y., et al.: Generating interpretable counterfactual explanations by implicit minimisation of epistemic and aleatoric uncertainties. In: International Conference on Artificial Intelligence and Statistics. pp. 1756–1764. PMLR (2021)
44. Spooner, T., Dervovic, D., Long, J., Shepard, J., Chen, J., Magazzeni, D.: Counterfactual explanations for arbitrary regression models. arXiv preprint arXiv:2106.15212 (2021)
45. Swist, T., Gulson, K.N.: School choice algorithms: Data infrastructures, automation, and inequality. Postdigital Science and Education pp. 1–19 (2022)
46. Upadhyay, S., Joshi, S., Lakkaraju, H.: Towards robust and reliable algorithmic recourse. Advances in Neural Information Processing Systems **34**, 16926–16937 (2021)
47. Virgolin, M., Fracaros, S.: On the robustness of counterfactual explanations to adverse perturbations. arXiv preprint arXiv:2201.09051 (2022)
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. Xiang, J.: Ai in lending. The AI Book: The Artificial Intelligence Handbook for Investors, Entrepreneurs and FinTech Visionaries pp. 34–38 (2020)