Robust Recourse for Binary Allocation Problems

Meirav Segal
University of Oslo
meiravs@ifi.uio.no

Anne-Marie George
University of Oslo
annemage@ifi.uio.no

Ingrid Chieh Yu
University of Oslo
ingridcy@ifi.uio.no

Christos Dimitrakakis
University of Neuchâtel
christos.dimitrakakis@unine.ch

Abstract

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, decision maker's objective and other individuals currently applying for the resource. Specifically, we focus on $0-1$ knapsack problems and in particular the use case of lending. We first provide a method for generating counterfactual explanations and then address the problem of recourse invalidation due to changes in allocation variables. Finally, we empirically compare our method with perturbation-robust recourse and show that our method can provide higher validity at a lower cost.

1 Introduction

Automated decision-making systems are currently employed in many high-risk applications (Xiang, 2020; Swist and Gulson, 2022). As these applications have a great impact on people's lives and future trajectories, it is gravely important to provide individuals with explanations regarding such decisions and algorithmic recourse—actions that allow the individual to obtain the desired outcome. A widely used approach is counterfactual explanations (CE), which provides to an individual a feature vector close to their own that would have obtained the desired outcome. As such, CE and recourse are of the same format: the description of individual's features that would yield the desired outcome. While CEs interpret this as an explanation of what the current individual is lacking, for recourse the action recommendation is to obtain the described features.

The CE and recourse literature is mainly focused on binary classification settings. In contrast, we explore the notions of CE and recourse for *allocation problems*. In allocation problems, a decision maker (DM) is allocating *limited* resources among a *population* in order to maximise some *objective* or utility. Applications such as college admissions and loan granting, which are usually considered as classification problems (Lux et al., 2016; Goyal and Kaur, 2016), are in fact dependant on resource constraints (and consequentially the whole population) and are thus better phrased as allocation problems. As the decision is determined according to the available resources, current population (or applicant pool) and the DM's utility function, it is insufficient to provide CEs with respect to a prediction model. Instead, we define CE with respect to the entire decision-making process, i.e., the allocation problem and its variables (See Section 4).

A recent survey about algorithmic recourse (Karimi et al., 2021) 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

XAI in Action: Past, Present, and Future Applications @ NeurIPS 2023.

routing problems, where several contributions deal with explainability by answering "why-not" and "what-if" questions (Lerouge et al., 2023; Ludwig et al., 2018; Eifler et al., 2022; Čyras et al., 2019). 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 (Korikov and Beck, 2021). However, they do not address the problem of algorithmic recourse and possible changes to the problem variables and constraints.

All three allocation problem variables – resources, population and utility – may change over time. For the lending use case, we can consider a bank making a decision every time step based on a batch of loan applications. In this case, possible changes of variables could include: 1) *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. 2) *Applicants:* We do not expect to see the exact same population applying again for a loan. 3) *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. 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 (Section 5).

In this paper, we focus on binary allocation problems with monotonic separable utilities (Section 2). For these problems, we provide a pipeline for generating CE under a black-box prediction model and a $0-1$ optimal knapsack allocation policy. We assume to have a CE-generator for classifiers and encapsulate this part in the pipeline. Our contributions: 1) We propose allocation problems as a novel setting for considering CE and algorithmic recourse. 2) In this setting, we show through an example that CEs for allocations can be more reliable for static allocations compared to CEs for the associated classification task (Section 3). 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 provide high chances of achieving the desired outcome.

2 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 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 that satisfies $\sum_i y_i w_i \leq r$. 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 $u : \mathbb{R}^l \to \mathbb{R}$ 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 $M : \mathbb{R}^l \to [0,1]$ 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 $S_\theta : [0,1] \to \mathbb{R}$ is a monotonically increasing function parameterised by θ. The parameter θ could for example represent the current interest rate. The individual utility can be interpreted as the predicted gain if we allocate the requested resource to individual i.

Table 1: Motivating example

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

3 Use Case: Lending

The use case of lending is often seen as an example of a high-risk application of automated decision making systems (Kop, 2021). 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 (Web, [n.d.]).[1]

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,[2] 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 $u(x_i) = M(x_i)(w_iG_1 + G_2) - (1 - M(x_i))Cw_i$.

Allocation Policy. As the DM is trying to maximise utility under budget constraints, where each applicant has individual utility and desired credit, we can translate this problem to the well known $0 - 1$ knapsack problem (Assi and Haraty, 2018). 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 by $\max \sum_{i=1}^{n} v_iy_i$ s.t. $\sum_{i=1}^{n} w_iy_i \leq r$, 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.[3] We therefore assume that the DM's allocation policy for this application is determined by the optimal solution.

Motivating Example Consider the applicants described in Table 1 under the utility function $u(x_i) = M(x_i)(w_i(G_1 + C) + G_2) - Cw_i$ 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.

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

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

[3] Note that we assume discretisation: the credit has a minimal step size (e.g. $100\$$).

4 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 (Guidotti, 2022). 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 CE to be minimal if its cost $d(x_i, x')$ is minimal under some metric $d : \mathbb{R}^l \times \mathbb{R}^l \to \mathbb{R}$.*[4]

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. Here we explain how to produce this for the 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 can show 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 Appendix A. We can thus use a dynamic programming algorithm [5] for $0-1$ knapsack, see e.g., Martello and Toth (1990), 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 $\epsilon > 0$.

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: $x' = \arg\min_z d(z, x)$ s.t. $M(z) \geq m'$. 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 (Pawelczyk et al., 2021; Guidotti, 2022) can be used on $h_{m'}$ with metric d, which provides x', a minimal CE with respect to the feature-based cost function d.

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. The $0-1$ optimal knapsack policy satisfies 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.

[4]Note that here, there could be $j \neq i$ 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.

[5]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.

5 Robust Recourse for Binary Allocations

We fist define (robust) recourse for binary allocations under variables distributions. We then describe how to generate approximate robust recourse and evaluate this in our experiments.

Definition 2 (Valid Recourse for Repeated Allocations). *At time t_1, given the allocation variables $r_{t_1}, X_{t_1}, U_{t_1}$, a* recourse *for individual $x_i \in X_{t_1}$ 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 for t_2: $r_{t_2}, X_{t_2} \in \mathbb{R}^{n-1,l}, U_{t_2}$, 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 *if its cost $d(x_i, x')$ is minimal under some 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. (2022) 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 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., $\mathbb{E}_D[\mathbb{I}\{x' \text{ valid for } \langle R_{t_2}, X_{t_2}, U_{t_2}\rangle\}] \geq \rho$, where $\mathbb{I}[\cdot]$ 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.*

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".

5.1 Approximated Robust Recourse

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 in Appendix C). 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 4. 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 (Dong and Nakayama, 2018), this m_ρ approximates the validity over the entire distribution.[6] This ρ-robust prediction-CE can then be translated to features, as was proposed in step 3 in Section 4. The produced feature vector x' is then the minimiser of $\min_z d(z, x_i) \quad \text{s.t. } \frac{1}{n}\sum_{j=1}^{n} \mathbb{I}[M(z) > m'_j] \geq \rho$. Here, m'_j is the j-th prediction-CE. Note that it is sufficient to sort the thresholds, as is done in Algorithm 1 in Appendix C. 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 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.

5.2 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

[6] 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.

Table 2: Empirical results for robust recourse under resource distribution

Method	**Cost**	**Validity**
Static CE	0.42	0.823
0.7-robust	**0.407**	0.84
0.9-robust	0.51	0.917
0.7-noisy	0.571	0.888
0.9-noisy	0.649	0.977
Optimistic	1	**1**

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 (Virgolin and Fracaros, 2022; Dominguez-Olmedo et al., 2022; Nguyen et al., 2022, 2023; Bui et al., 2022; Upadhyay et al., 2021). See Appendix B for additional related work.

Dataset We use the German credit dataset (Dua and Graff, 2017) which is one of the most common benchmarks used for CE and algorithmic recourse (e.g. Dutta et al. (2022); Black et al. (2021); Bui et al. (2022); Guo et al. (2022)). 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. We construct 20 allocation problems by uniformly sampling 20 individuals from the test set, set the utility function parameters to $G_1 = 0.06, G_2 = 4, C = 0.5$, and sample a budget from the budget distribution. See additional preprocessing details in Appendix D.1.

Method and Baselines We test our ρ-robust recourse method, described in Algorithm 1 in Appendix C, 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. See additional details in Appendix D.2. 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.

Evaluation For each allocation problem, we find the optimal allocation via the optimal $0-1$ knapsack and provide recourse for all individuals not included in the allocation. The results are described in Table 2. 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.

Results From table 2, 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 Pareto-dominates the static-CE, 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. 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. See Appendix D.3 for additional discussion.

6 Discussion

In this paper, we present the first attempt to define robust recourse for binary allocation problems. Under this setting, we provide a use case of lending 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.

In this paper we only make the first step in solving this new setting of recourse for allocation problems. We addressed allocation problems with binary decisions and separable utilities. More complex problems within the scope of allocation problems could be addressed in the future. 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. Additional limitations and possible extensions can be found in Appendix E.

Acknowledgments and Disclosure of Funding

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

References

[n.d.]. Federal Student Aid in the U.S. Department of Education website. `https://www2.ed.gov/about/offices/list/fsa/index.html?exp=6`. Accessed: 2023-02-17.

Gohar Ali, Feras Al-Obeidat, Abdallah Tubaishat, Tehseen Zia, Muhammad Ilyas, and Alvaro Rocha. 2021. Counterfactual explanation of Bayesian model uncertainty. *Neural Computing and Applications* (2021), 1–8.

Maram Assi and Ramzi A Haraty. 2018. A survey of the knapsack problem. In *2018 International Arab Conference on Information Technology (ACIT)*. IEEE, 1–6.

Emily Black, Zifan Wang, Matt Fredrikson, and Anupam Datta. 2021. Consistent counterfactuals for deep models. *arXiv preprint arXiv:2110.03109* (2021).

Ngoc Bui, Duy Nguyen, and Viet Anh Nguyen. 2022. Counterfactual Plans under Distributional Ambiguity. *arXiv preprint arXiv:2201.12487* (2022).

Kristijonas Čyras, Dimitrios Letsios, Ruth Misener, and Francesca Toni. 2019. Argumentation for explainable scheduling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 2752–2759.

Eoin Delaney, Derek Greene, and Mark T Keane. 2021. Uncertainty estimation and out-of-distribution detection for counterfactual explanations: Pitfalls and solutions. *arXiv preprint arXiv:2107.09734* (2021).

Ricardo Dominguez-Olmedo, Amir H Karimi, and Bernhard Schölkopf. 2022. On the adversarial robustness of causal algorithmic recourse. In *International Conference on Machine Learning*. PMLR, 5324–5342.

Hui Dong and Marvin K Nakayama. 2018. A tutorial on quantile estimation via Monte Carlo. In *International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. Springer, 3–30.

Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. `http://archive.ics.uci.edu/ml`

Sanghamitra Dutta, Jason Long, Saumitra Mishra, Cecilia Tilli, and Daniele Magazzeni. 2022. Robust Counterfactual Explanations for Tree-Based Ensembles. In *International Conference on Machine Learning*. PMLR, 5742–5756.

Rebecca Eifler, Jeremy Frank, and Joerg Hoffmann. 2022. Explaining Soft-Goal Conflicts through Constraint Relaxations. In *ICAPS 2022 Workshop on Explainable AI Planning*.

Andrea Ferrario and Michele Loi. 2022. The Robustness of Counterfactual Explanations over Time. *IEEE Access* (2022).

Anchal Goyal and Ranpreet Kaur. 2016. A survey on ensemble model for loan prediction. *International Journal of Engineering Trends and Applications (IJETA)* 3, 1 (2016), 32–37.

Riccardo Guidotti. 2022. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery* (2022), 1–55.

Hangzhi Guo, Feiran Jia, Jinghui Chen, Anna Squicciarini, and Amulya Yadav. 2022. RoCourseNet: Distributionally Robust Training of a Prediction Aware Recourse Model. *arXiv preprint arXiv:2206.00700* (2022).

Jean Kaddour, Aengus Lynch, Qi Liu, Matt J Kusner, and Ricardo Silva. 2022. Causal Machine Learning: A Survey and Open Problems. *arXiv preprint arXiv:2206.15475* (2022).

Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. 2021. A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys (CSUR)* (2021).

Amir-Hossein Karimi, Julius Von Kügelgen, Bernhard Schölkopf, and Isabel Valera. 2020. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. *Advances in neural information processing systems* 33 (2020), 265–277.

Gunnar König, Timo Freiesleben, and Moritz Grosse-Wentrup. 2021. A causal perspective on meaningful and robust algorithmic recourse. *arXiv preprint arXiv:2107.07853* (2021).

Mauritz Kop. 2021. EU artificial intelligence act: the European approach to AI. Stanford-Vienna Transatlantic Technology Law Forum, Transatlantic Antitrust ....

Anton Korikov and J Christopher Beck. 2021. 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.

Mathieu Lerouge, Céline Gicquel, Vincent Mousseau, and Wassila Ouerdane. 2023. Counterfactual Explanations for Workforce Scheduling and Routing Problems. In *12th International Conference on Operations Research and Enterprise Systems*. SCITEPRESS-Science and Technology Publications, 50–61.

Dan Ley, Umang Bhatt, and Adrian Weller. 2022. Diverse, Global and Amortised Counterfactual Explanations for Uncertainty Estimates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 7390–7398.

Jeremy Ludwig, Annaka Kalton, and Richard Stottler. 2018. Explaining Complex Scheduling Decisions.. In *IUI Workshops*.

Thomas Lux, Randall Pittman, Maya Shende, and Anil Shende. 2016. Applications of supervised learning techniques on undergraduate admissions data. In *Proceedings of the ACM International Conference on Computing Frontiers*. 412–417.

Silvano Martello and Paolo Toth. 1990. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc.

Saumitra Mishra, Sanghamitra Dutta, Jason Long, and Daniele Magazzeni. 2021. A survey on the robustness of feature importance and counterfactual explanations. *arXiv preprint arXiv:2111.00358* (2021).

Rami Mochaourab, Sugandh Sinha, Stanley Greenstein, and Panagiotis Papapetrou. 2021. Robust Counterfactual Explanations for Privacy-Preserving SVM. In *International Conference on Machine Learning (ICML 2021), Workshop on Socially Responsible Machine Learning*.

Duy Nguyen, Ngoc Bui, and Viet Anh Nguyen. 2023. Distributionally robust recourse action. *arXiv preprint arXiv:2302.11211* (2023).

Tuan-Duy H Nguyen, Ngoc Bui, Duy Nguyen, Man-Chung Yue, and Viet Anh Nguyen. 2022. Robust Bayesian Recourse. In *Uncertainty in Artificial Intelligence*. PMLR, 1498–1508.

Martin Pawelczyk, Sascha Bielawski, Johannes van den Heuvel, Tobias Richter, and Gjergji Kasneci. 2021. Carla: a python library to benchmark algorithmic recourse and counterfactual explanation algorithms. *arXiv preprint arXiv:2108.00783* (2021).

Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. 2020. On counterfactual explanations under predictive multiplicity. In *Conference on Uncertainty in Artificial Intelligence*. PMLR, 809–818.

Martin Pawelczyk, Teresa Datta, Johannes van-den Heuvel, Gjergji Kasneci, and Himabindu Lakkaraju. 2022. Probabilistically Robust Recourse: Navigating the Trade-offs between Costs and Robustness in Algorithmic Recourse. *arXiv preprint arXiv:2203.06768* (2022).

Kaivalya Rawal, Ece Kamar, and Himabindu Lakkaraju. 2020. Algorithmic recourse in the wild: Understanding the impact of data and model shifts. *arXiv preprint arXiv:2012.11788* (2020).

Lisa Schut, Oscar Key, Rory Mc Grath, Luca Costabello, Bogdan Sacaleanu, Yarin Gal, et al. 2021. Generating interpretable counterfactual explanations by implicit minimisation of epistemic and aleatoric uncertainties. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1756–1764.

Teresa Swist and Kalervo N Gulson. 2022. School Choice Algorithms: Data Infrastructures, Automation, and Inequality. *Postdigital Science and Education* (2022), 1–19.

Sohini Upadhyay, Shalmali Joshi, and Himabindu Lakkaraju. 2021. Towards robust and reliable algorithmic recourse. *Advances in Neural Information Processing Systems* 34 (2021), 16926–16937.

Marco Virgolin and Saverio Fracaros. 2022. On the Robustness of Counterfactual Explanations to Adverse Perturbations. *arXiv preprint arXiv:2201.09051* (2022).

Joshua Xiang. 2020. AI in Lending. *The AI Book: The Artificial Intelligence Handbook for Investors, Entrepreneurs and FinTech Visionaries* (2020), 34–38.

A Proof of Minimal Utility-CE for 0-1 Knapsack

In Section 4 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 (Korikov and Beck, 2021). 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 receive the requested loan. Nevertheless, we only consider here the option to change user features (excluding the requested credit), assuming the requested credit cannot be changed.

B Additional Related Work

Recourse Invalidation The problem of algorithmic recourse invalidation (or invalidation of counterfactual explanations) and the need for robustness has already been recognised in recent years (Mishra et al., 2021). The majority of papers consider invalidation due to model retraining with different training data, usually following a distribution shift (Nguyen et al., 2022, 2023; Guo et al., 2022; Rawal et al., 2020; König et al., 2021; Bui et al., 2022; Black et al., 2021; Upadhyay et al., 2021; Dutta et al., 2022). 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 (Kaddour et al., 2022). Other studied causes of invalidation are change of prediction model (Pawelczyk et al., 2020) and feature perturbation, which could be due to inaccurate implementation of the recourse (Dominguez-Olmedo et al., 2022; Pawelczyk et al., 2022; Virgolin and Fracaros, 2022) or privacy perturbation (Mochaourab et al., 2021). 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 (Virgolin and Fracaros, 2022; Dominguez-Olmedo et al., 2022; Nguyen et al., 2022, 2023; Bui et al., 2022; Upadhyay et al., 2021). While these methods indeed improve the robustness of the recourse, they also present a trade-off between robustness and cost (distance of the counterfactual from the original feature vector) (Rawal et al., 2020; Pawelczyk et al., 2020; Upadhyay et al., 2021). For deep networks, even if no explicit trade-off exists, the robust recourse is still presented as more costly (Black et al., 2021). 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, not the worst-case with respect to current variables, which might result in a lower cost for the user. 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 (Pawelczyk et al., 2022) 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 (Karimi et al., 2020). We facilitate the same kind of control for allocation problems.

Other approaches Ferrario and Loi (2022) 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 (Delaney et al., 2021; Schut et al., 2021; Ali et al., 2021). 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[7].

[7]We note that a problem which might be considered as related is the of use counterfactuals to explain classification uncertainty (Ley et al., 2022). This is a different objective and in our work we do not account for prediction uncertainty.

C Algorithm

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 π, n samples from 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 $AP_j = \langle r_j, X_j, u_j = S_{\theta_j} \circ M\rangle$ and policy π

  Get prediction-CE $m'_j = S^{-1}_{\theta_j}(v'_j)$

end for

Sort all prediction-CE: $\mathrm{sorted} \leftarrow \mathrm{sort}([m'_j]^n_{j=1})$

Get ρ-robust prediction-CE $m_\rho \leftarrow \mathrm{sorted}[\lceil \rho n\rceil]$

return Feature-based CE $x' = E(M, i, m_\rho)$

D Additional Experimental Details

D.1 Dataset and Preprocessing

We scale numeric features to $[0, 1]$ and encode categorical features as 1-hot vectors. In addition, the requested credit is divided by 100. The random forest classifier achieves accuracy of 0.78 on the test set. We sample 50 batches of 20 applicants and consider the sum of given credit as the current budget. We then fit a normal distribution to it, and consider this as the budget distribution.

D.2 Noisy Robust Recourse

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.

D.3 Result Discussion

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: $(0.01, 0.1, 0.2, 0.2, 0.2, 0.25)$. 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.

E Limitations and Additional Future Work

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 3 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. As previously mentioned, 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 assumed 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.

An 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 in the current population. This calls for fairness considerations. Minimal (negative) impact on other individuals could be defined as a new desired property of CE and recourse, similarly to other properties in the literature (e.g. diversity, efficiency and stability). Furthermore, recourse models could includes possible feedback as a result of recourse implementation. We do not address this direction in the paper and leave it for future work.