

Threshold Recourse for Dynamic Allocations

Meirav Segal¹, Anne-Marie George¹ and Christos Dimitrakakis²

¹University of Oslo, Oslo, Norway

²University of Neuchatel, Neuchatel, Switzerland

Abstract

In this paper, we focus on recourse for the setting of dynamic sequential allocations with limited resources, in which agents are competing against each other. In particular, we assume the decision maker is using a threshold policy and demonstrate the problem of recourse invalidation under the use case of college admissions. We then provide a first attempt to generate a valid threshold recourse that takes into account the feedback effect of recourse implementation.

Keywords

Recourse, allocation problems, feedback loops, multi-agent recourse

1. Introduction

As automated decision-making systems are increasingly embedded in high-risk applications, it is highly important to maintain human agency. In practice, this need is met by providing recourse to users who received a negative decision (e.g., are not admitted to university). The recourse is a recommendation for actions the users should take in order to qualify for a positive decision in the future. In many cases, the recourse takes the form of an alternative feature vector that requires minimal changes from the user's current feature vector (minimal cost).


A key property of a recourse is validity. A recourse is valid if the user is indeed granted the desired decision following the implementation of the recourse. It has been shown in recent years that the validity of a recourse might be impacted by different factors such as changes in the model due to retraining after a distribution shift [1, 2, 3, 4, 5, 6, 7, 8, 9]. Nonetheless, only recently has the community began to address invalidation due to feedback effects.


When generating a recourse, most methods consider the point of view of the individual. Yet, for allocation problems in particular, in which resources are limited, it is crucial to take into account the competition between users and the impact of recourse implementation. For example, applicants who were not admitted in the current year for a university study program, might be told which admission threshold was used. In the following year, let us assume that k rejected candidates managed to increase their GPA to a value higher than that threshold. If the number of admission slots r is lower ($r < k$), that means that the recourse is not valid, since some people who implemented the recourse do not receive the beneficial decision.

EWAF'24: European Workshop on Algorithmic Fairness, July 01–03, 2024, Mainz, Germany

✉ meiravs@ifi.uio.no (M. Segal); annemage@ifi.uio.no (A. George); christos.dimitrakakis@unine.ch (C. Dimitrakakis)

ORCID 0000-0001-7145-9159 (M. Segal); 0000-0001-9232-8211 (A. George); 0000-0002-5367-5189 (C. Dimitrakakis)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

This problem was addressed via an empirical simulation study [10], in which the authors measured the effect of different parameters on recourse validity. The actions taken by individuals following a recourse were simulated and the validity of the recourse in the next step was measured. Yet, the provided recourse itself did not take into account the feedback effect of the recourse implementation. A multi-agent point of view of recourse was presented in an earlier work [11]. The authors provide two new recourse definitions: 1) Social-Welfare-Efficient Recourse: a minimal-cost recourse that also increases the sum of predictions, and 2) Pareto-Efficient Recourse: a minimal-cost recourse in which none of the individuals receiving the recourse is worse off. Other papers [12, 13] aim to design an optimal policy considering the users' best response when acting strategically, but do not generate an explicit recourse.

In this paper, we present a setting of a dynamic allocation problem based on the college admission use-case. We provide an algorithm for generating all possible valid recourse values. Finally, we show that the current recourse properties may not be sufficient to select the recourse which benefits users the most. There is a need to specify the recourse objective.

2. Dynamic college admission setting

Let us consider the use case of college admissions. At each time step t , a population of candidates apply for available seats. Each candidate i has a 1-dimensional feature x_i that represents their value to the decision maker (DM, university in this case) if admitted. The available resource quantity (number of seats) is r . The DM uses a threshold allocation policy and sets a threshold τ_t such that at most r candidates are admitted. A candidate i is admitted if $x_i > \tau_t$. The overall utility of the allocation to the DM, U_{DM} , is the sum of features of all admitted applicants.

The population of candidates is composed of two groups:

- **New candidates** X . A set of n candidates who are applying for the first time. We assume that the features of the new population arriving are the same at each round, e.g. $\{x_1 = 0.9, x_2 = 0.7, \dots, x_n = 0.5\}$.¹ For simplicity, we assume the candidates are sorted such that x_1 has the highest value and x_n has the lowest value.
- **Reapplying candidates** X^{RC} . A set of l_t candidates who were rejected in the previous round and attempt to be admitted in the current round. For the first round, this set is empty. Again, we assume the candidates are sorted such that x_1^{RC} has the highest value and $x_{l_t}^{RC}$ has the lowest value. The size of this set l_t may change from round to round.

After setting the admission threshold τ , the DM provides the rejected candidates $X^- = \{x \in X | x \leq \tau\}$ a recourse \hat{x} . In this setting, all candidates are provided with the same recourse, which is the feature value they should have in the next round in order to be admitted. The rejected candidates from group X^{RC} cannot reapply (we allow only a second chance).

We consider the decision making process of the users regarding recourse implementation. All individuals share the same cost function $d : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$. We assume the candidates choose whether to implement the recourse based on their expected utility. For this example, we set G , the gain or reward given to each individual, to 1 if admitted ($G^+ = 1$) and 0 if rejected ($G^- = 0$). Given that the recourse is valid (acceptance is guaranteed upon implementation of

¹Note that this is a more restrictive assumption than the setting proposed in [10].

the recourse), the candidate will choose to implement the recourse only if $1 - d(x_i, \hat{x}) > \mathbb{1}_{x_i \geq \hat{x}}$, i.e., the expected utility from implementing the recourse is higher than that of not implementing the recourse. The set of reapplying candidates after implementing the recourse is denoted X^{RC} .

3. Validity

Our goal is to provide a valid recourse, i.e. $\hat{x}_t > \tau_{t+1}$. For this condition to hold, the total number of candidates above the value \hat{x}_t should be at most r . We demonstrate the problem of invalid recourse using the following example. We consider an allocation problem at $t = 0$ with resources $r = 2$, cost function $d(x, \hat{x}) = 2 \cdot |\hat{x} - x|$, new candidates $X = \{x_1 = 0.8, x_2 = 0.7, x_3 = 0.5, x_4 = 0.4, x_5 = 0.3, x_6 = 0.1\}$ and no current reapplying candidates $X^{RC} = \{\}$. The admission threshold is $\tau_0 = 0.6$, i.e., the admitted set is $X_0^+ = \{x_1 = 0.8, x_2 = 0.7\}$.

After setting the threshold, we would like to provide a recourse for the set of rejected candidates $X^- = \{x_1^- = 0.5, x_2^- = 0.4, x_3^- = 0.3, x_4^- = 0.1\}$. As mentioned above, we assume that the set of new candidates X is identical at each time step. Let us consider the possible recourse value 0.75. The costs of implementing this recourse for the rejected candidates are $\{d(x_1^-, 0.75) = 0.5, d(x_2^-, 0.75) = 0.7, d(x_3^-, 0.75) = 0.9, d(x_4^-, 0.75) = 1.3\}$. Thus, three out of the four rejected candidates will choose to implement the recourse, as their cost is lower than the reward of being admitted (1). Yet, there are only two available slots ($r = 2$), so the recourse will not be valid. The threshold would be at least 0.75 and none of the candidates implementing the recourse would be admitted.

If we assume the feature space is unbounded, a naive valid recourse always exists - setting the recourse to a high enough value such that no individual chooses to implement it. In case of a bounded feature space, there may not be a valid recourse, e.g., when too many individuals would choose to implement a recourse of the maximal feature value.

4. Multi-agent recourse

In some cases, several values could be used as a valid recourse. Following our example, we can identify two ranges of recourse values. A recourse in the range $[0.9, 1)$ would result in only one candidate implementing the recourse, x_1^- . The costs in the range $[d(x_1^-, 0.9) = 0.8, d(x_1^-, 1) = 1)$ provide the candidate with a positive gain, while the costs for the other candidates would result in a non-positive utility, e.g., $d(x_2^-, 0.9) = 1$. Therefore, the rest of the rejected candidates will not implement the recourse. Similarly, a recourse in the range $(0.8, 0.9)$ would cause exactly two candidates to implement the recourse (x_1^-, x_2^-). Each option is represented by a range and not by a single value. Clearly, the minimal value in this range would benefit the individuals implementing the recourse, as their cost would be minimised. We wish to provide a recourse of minimal cost, so we can always choose the minimal value as recourse.

5. Multi-agent recourse algorithm

Algorithm 1 generates all minimal valid recourse values. First, for each individual i of the top $r + 1$ rejected candidates, we find μ_i , the minimal recourse value for which the candidate will

Algorithm 1 Find all minimal valid recourse values (bounded features)

Require: new candidates X (of the next round, sorted in descending order), rejected candidates X^- (sorted in descending order) with $|X^-| = l$, cost function d , resource amount r , $\epsilon > 0$, feature upper bound B , gain to the individual if accepted G^+

▷ Find the minimal recourse value for which the individual will choose not to implement the recourse.

for i from 1 to $\min(r + 1, l)$ **do**
 find μ_i such that $d(x_i^-, \mu_i) = G^+$

end for

$\tau \leftarrow \{\}$

▷ A possible recourse value to allow the top j candidates to reapply. This value guarantees that the rest of the reapplying candidates will choose not to implement the recourse (too costly), and that all individuals implementing the recourse will be admitted.

for j from 1 to $\min(r, l - 1)$ **do**

$\hat{x}_j \leftarrow \max(\mu_{j+1}, x_{r-j+1} + \epsilon)$

▷ The recourse is added only if it is not too costly for the individual with the lowest feature among the top j reapplying candidates (hence not too costly for all top j), and the recourse does not exceed the feature upper bound.

if $\hat{x}_j < \mu_j$ and $\hat{x}_j \leq B$ **then**

 Add \hat{x}_j to τ

end if

end for

▷ The case of providing a recourse that allows all rejected candidates to reapply (if the number of available seats is at least the number of rejected candidates).

if $l \leq r$ **then**

$\hat{x}_l \leftarrow x_{r-l+1} + \epsilon$

▷ There is no need to check whether the recourse is too costly for the rest of the rejected candidates.

if $\hat{x}_l < \mu_l$ and $\hat{x}_l \leq B$ **then**

 Add \hat{x}_l to τ

end if

end if

return τ

choose to not implement the recourse. Second, for each number j of reapplying candidates from 1 to r , we set the minimal recourse value \hat{x}^j to $\max(\mu_{j+1}, x_{r-j+1} + \epsilon]$, $\epsilon > 0$. This recourse is guaranteed to admit at most j reapplying candidates, because this recourse is too costly for the rest of the rejected candidates. In addition, all reapplying candidates implementing this recourse are guaranteed to be allocated with the resource (their new feature value is larger than that of the r -th new candidate). We then check two conditions: 1) The recourse is not too costly for all top j rejected individuals. That is, the recourse value \hat{x}_j must be lower than μ_j .² 2) In

²Note that as the feature increases, the cost decreases. Therefore, it is sufficient to verify this condition for the candidate with the lowest feature in the top j rejected candidates.

Table 1

Recourse utilities for the DM and the reapplying candidates.

Recourse value	U_{DM}	U_{RC}
0.9	$0.9+0.8=1.7$	$1-0.8=0.2$
0.81	$0.81+0.81=1.62$	$(1-0.62)+(1-0.82)=0.56$

case of bounded features, the recourse must not exceed the maximal feature value.

Utility-maximising recourse Given several values of a minimal valid recourse, there is a need to determine which value is preferred. Following our example, we are left with two options for a recourse: $\hat{x}^1 = 0.9$, $\hat{x}^2 = 0.8 + \epsilon$ for some $\epsilon > 0$ (here, $\epsilon = 0.01$). The decision of which specific value to choose depends on the objective. We consider two viewpoints:

1. *The DM's objective.* Maximising the sum of features of admitted individuals (U_{DM}).
2. *The objective of the reapplying population.* Maximising the sum of individual rewards (to the individuals) for the set of the reapplying population (U_{RC}).

From the utilities for the DM and the reapplying candidates in table 1, we can observe that the DM would prefer to provide the recourse $\hat{x}^1 = 0.9$ while the reapplying candidates would prefer the recourse $\hat{x}^2 = 0.81$. Hence, for recourse generation for dynamic allocation problems, it is not sufficient to specify that the recourse should be valid and minimal, there should also be a recourse objective in order to select one value from a set of possible recourse values.

6. Discussion and future work

The problem of recourse invalidation due to recourse implementation was only recently discussed in the literature. To the best of our knowledge, this is the first attempt to provide a valid feedback-aware recourse in dynamic settings. Using an example, we show that this problem could have several solutions, or not at all. In order to select one solution, new recourse desiderata should be in place, to prevent misuse of recourse by the decision makers.

The setting described in this paper has several limitations that can be relaxed in future work. For example, a more realistic setting could include applicants sampled from a population distribution. In addition, we only consider a threshold recourse, but the DM could also provide a different recourse to each individual. Moreover, we consider validity in a binary way, but in practice, most methods provide a recourse that is valid with high probability. When the recourse is not guaranteed to be valid, the estimation of expected utility for the users would change, and consequently the recourse should take that into consideration. Lastly, we do not consider the utility for new candidates, and the impact of intentionally providing a costly recourse to some individuals. Hence, some additional fairness notions for recourse may be required.

Acknowledgments

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

References

- [1] T.-D. H. Nguyen, N. Bui, D. Nguyen, M.-C. Yue, V. A. Nguyen, Robust bayesian recourse, in: *Uncertainty in Artificial Intelligence*, PMLR, 2022, pp. 1498–1508.
- [2] D. Nguyen, N. Bui, V. A. Nguyen, Distributionally robust recourse action, *arXiv preprint arXiv:2302.11211* (2023).
- [3] H. Guo, F. Jia, J. Chen, A. Squicciarini, A. Yadav, Rocoursenet: Distributionally robust training of a prediction aware recourse model, *arXiv preprint arXiv:2206.00700* (2022).
- [4] K. Rawal, E. Kamar, H. Lakkaraju, Algorithmic recourse in the wild: Understanding the impact of data and model shifts, *arXiv preprint arXiv:2012.11788* (2020).
- [5] G. König, T. Freiesleben, M. Grosse-Wentrup, A causal perspective on meaningful and robust algorithmic recourse, *arXiv preprint arXiv:2107.07853* (2021).
- [6] N. Bui, D. Nguyen, V. A. Nguyen, Counterfactual plans under distributional ambiguity, *arXiv preprint arXiv:2201.12487* (2022).
- [7] E. Black, Z. Wang, M. Fredrikson, A. Datta, Consistent counterfactuals for deep models, *arXiv preprint arXiv:2110.03109* (2021).
- [8] S. Upadhyay, S. Joshi, H. Lakkaraju, Towards robust and reliable algorithmic recourse, *Advances in Neural Information Processing Systems* 34 (2021) 16926–16937.
- [9] S. Dutta, J. Long, S. Mishra, C. Tilli, D. Magazzeni, Robust counterfactual explanations for tree-based ensembles, in: *International Conference on Machine Learning*, PMLR, 2022, pp. 5742–5756.
- [10] J. Fonseca, A. Bell, C. Abrate, F. Bonchi, J. Stoyanovich, Setting the right expectations: Algorithmic recourse over time, in: *Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, 2023, pp. 1–11.
- [11] A. O’Brien, E. Kim, Toward multi-agent algorithmic recourse: Challenges from a game-theoretic perspective, in: *The International FLAIRS Conference Proceedings*, volume 35, 2022.
- [12] S. Tsirtsis, B. Tabibian, M. Khajehnejad, A. Singla, B. Schölkopf, M. Gomez-Rodriguez, Optimal decision making under strategic behavior, *Management Science* (2024).
- [13] Y. Chen, J. Wang, Y. Liu, Strategic recourse in linear classification, *arXiv preprint arXiv:2011.00355* 236 (2020).