RAGUEL: Recourse-Aware Group Unfairness Elimination

Aparajita Haldar
University of Warwick
Coventry, United Kingdom
aparajita.haldar@warwick.ac.uk

Teddy Cunningham
University of Warwick
Coventry, United Kingdom
teddy.cunningham@warwick.ac.uk

Hakan Ferhatosmanoglu*
University of Warwick
Coventry, United Kingdom
hakan.f@warwick.ac.uk

ABSTRACT

While machine learning and ranking-based systems are in widespread use for sensitive decision-making processes (e.g., determining job candidates, assigning credit scores), they are rife with concerns over unintended biases in their outcomes, which makes algorithmic fairness (e.g., demographic parity, equal opportunity) an objective of interest. 'Algorithmic recourse' offers feasible recovery actions to change unwanted outcomes through the modification of attributes. We introduce the notion of ranked group-level recourse fairness, and develop a 'recourse-aware ranking' solution that satisfies ranked recourse fairness constraints while minimizing the cost of suggested modifications. Our solution suggests interventions that can reorder the ranked list of database records and mitigate group-level unfairness; specifically, disproportionate representation of sub-groups and recourse cost imbalance. This re-ranking identifies the minimum modifications to data points, with these attribute modifications weighted according to their ease of recourse. We then present an efficient block-based extension that enables re-ranking at any granularity (e.g., multiple brackets of bank loan interest rates, multiple pages of search engine results). Evaluation on real datasets shows that, while existing methods may even exacerbate recourse unfairness, our solution – RAGUEL – significantly improves recourse-aware fairness. RAGUEL outperforms alternatives at improving recourse fairness, through a combined process of counterfactual generation and re-ranking, whilst remaining efficient for large-scale datasets.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

KEYWORDS

Fairness; Ranking; Algorithmic Recourse; Recourse-Aware Ranking; Classification; Machine Learning

ACM Reference Format:
Aparajita Haldar, Teddy Cunningham, and Hakan Ferhatosmanoglu. 2022. RAGUEL: Recourse-Aware Group Unfairness Elimination. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557424

*Also with Amazon Web Services. This publication presents work performed at the University of Warwick and is not associated with Amazon.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557424

1 INTRODUCTION

Machine learning techniques are being used increasingly for a wide range of decision-making. This includes classification-based decisions, such as medical diagnoses [20], judicial verdicts [9], financial risk assessments [39], as well as ranking-based decisions, such as exam results [42] and job applications [33]. Despite their influence on the real world, automated decision-making systems have come under scrutiny for potentially unfair outcomes between sub-groups separated based on protected data attributes, such as race, gender, and marital status [7, 11, 18, 22, 24, 43]. The concept of fairness is applicable more broadly, including technical settings such as fair resource allocation in computer networks [3] and fair task assignment in crowdsourcing [6].

Numerous definitions of algorithmic fairness have gained popularity, each of which has associated bias mitigation strategies, such as ignoring protected attributes in the model and enforcing balanced success/error rates across sub-groups [31, 34, 50]. However, none of the traditional unfairness mitigation strategies for ranking address the impact of imbalances in 'algorithmic recourse'. Recourse aims to provide users with a set of feasible actions that can be taken to recover from an unwanted outcome [41]. Recourse fairness implies that the opportunity for, and cost of, improvement or recovery should not favor any sub-group over another. Even if a model is fair in its expected outputs (e.g., by ensuring equal success rates through demographic parity), failing to consider recourse can lead to issues such as inequity across society due to imbalances in the cost of recovery. 'Group-level recourse fairness' (i.e., minimizing the difference in recourse across groups) is thus desirable in many settings, and has been included in recent classification models [17, 45]. Recourse fairness in ranking remains unexplored despite a range of real-world applications where entities should enjoy comparable costs of recovery to improve their ranking (e.g., job applicants, web search results, online dating profiles, e-commerce product listings [8, 16, 40]). Recourse-aware ranking is thus beneficial to identify the minimal adjustments that individuals need to make so that there is fairer representation throughout the ranked list. For example, measures have been prescribed to help low-income individuals demonstrate creditworthiness based on steady income rather than owning credit cards [4]. There is a need to identify which of these alternatives is most suitable to offer to a given individual to improve their chances for recourse.

In this paper, we introduce the notion of *ranked* group-level recourse fairness, which can be applied to classification models and ranking-based problems. For ranking problems, there is a recourse cost to reach some defined 'ideal' point that all database members aspire to achieve, whereas for classification problems, it is the cost

to reach the classifier boundary, modeled as a hyperplane of target query points that have the desired classifier outcome.

We propose a strategy that ranks records according to each record's (recourse) cost to reach a target point and, given any such ranked list, offers improved ranked recourse fairness with minimal re-ranking. Thus, we aim to minimize the cost of adjustment while satisfying fairness constraints through our re-ranking strategy, **RAGUEL**: **R**ecourse-**A**ware **G**roup **U**nfairness **El**imination. RAGUEL has two steps: i) computing the 'ranked group-level recourse fairness ratio' for sub-groups based on a given set of protected attributes (e.g., gender), and ii) adjusting the ranked list such that recourse fairness is improved across sub-groups. To achieve the former, RAGUEL determines 'counterfactuals' and aggregates the group-wise cost of recourse to reach the counterfactual(s). Here, we use the general term 'counterfactual' to denote any target point into which a record can feasibly be transformed. For the latter, RAGUEL first identifies the disadvantaged data points that require minimal perturbation towards their counterfactuals. RAGUEL then iteratively adjusts the attributes of a point until its new rank satisfies fair representation and recourse fairness at the group-level. RAGUEL recommends these changes to clients as potential recourse steps. Therefore, our solution considers recourse during the point selection process as well as the re-ranking process to minimize the cost of the recommended interventions. Unlike many existing strategies [17, 37, 38], RAGUEL does not require re-weighting or retraining the model after database repair.

We then extend RAGUEL to handle multiple ranked 'blocks', which occurs when multiple batches or brackets are used in ranking or classification outcomes. For example, loan application screening may require multiple different acceptance thresholds corresponding to different interest rate brackets offered. Similarly, information retrieval and recommendation systems typically present results in batches, such as search engine results pages. For such situations, one can model each bracket/batch to have its own threshold boundary, and divide the ranking into multiple blocks to match these boundaries. In the block-based approach, points are re-ranked to improve proportional representation and ensure ranked group-level recourse fairness of sub-groups within every block.

The contributions of RAGUEL are summarized as follows.

1. Ranked Group-Level Recourse Fairness. We introduce 'ranked group-level recourse fairness', which measures fairness in recourse actions in ranked lists; alongside the need for fair representation in ranking, this forms the basis for our problem. In contrast to existing fair ranking approaches, RAGUEL i) provides a set of feasible actions to the *user* (e.g., loan applicant) that would result in the desired outcome, and ii) enables the *platform* (e.g., bank) to consider recourse fairness via minimal cost of opportunities presented to disadvantaged sub-groups.

2. Recourse-Aware Fair Ranking. RAGUEL's iterative and minimally invasive approach considers the cost associated with the re-ranking while improving recourse fairness. The experiments show an improvement on recourse fairness by 15% compared to the initial ranking, by up to 200% compared to traditional fair classifiers (which actually exacerbate recourse unfairness due to the differences in objectives), and by 37% compared to the fair ranking method FA*IR [49].

Table 1: Summary of limitations in related work

Desideratum	MACE [25]	AR [41]	FA*IR [49]	FoEiR [40]	RAGUEL
Counterfactual generation	✓	✓	✗	✗	✓
Distance-agnostic	✓	✓	✗	✓	✓
Fair ranking	✗	✗	✓	✓	✓
Recourse consideration	✗	✓	✗	✗	✓
Adjustable granularity	✗	✗	✗	✗	✓

3. Efficient Counterfactual Generation and Re-Ranking. When multiple counterfactual points exist (e.g., when there is a classifier decision boundary), our inverse classification-based solution is around 10x faster on large datasets compared to the baseline [41]. RAGUEL also re-ranks large datasets with ease, whereas FA*IR [49] and FoEiR [40] could not handle more than 400 and 50 records respectively in practice.

4. Block-Based Solution. To the best of our knowledge, no prior ranking solution adjusts the granularity of the fairness measure. The proposed block-based re-ranking better preserves similarity to the original list compared to the non-block version. Besides practical use cases, RAGUEL-Block shows significant performance benefits. It requires fewer modifications to the ranked list with a 1.3x lower cost for achieving recourse fairness, and it is up to 30x faster in unfairness mitigation in our experiments. The solution is shown to be scalable for datasets with millions of records.

2 RELATED WORK

We now review recent literature that relates to our problem. Table 1 summarizes the main limitations of the most relevant work.

There has been an increased focus on understanding whether, and how, inherent biases result in decision-making systems being unfair to individuals or groups [5, 10, 31]. There have been several unfairness mitigation measures, such as simply omitting sensitive attributes and pre-processing the data to mask such attributes [13, 23]. Counterfactual fairness and explicit causal models capture the intuition that a decision outcome should not be influenced by sensitive attributes [27, 46, 51]. Interventions are used to reassign specific values to attributes of points to generate counterfactual explanations of models by asking "what if things had been different?". The term 'counterfactual' here indicates a point having different attribute values, leading to a different model prediction. LIME [36] offers local explanations for individual predictions of black-box models. CLEAR [47] extends this to compute the nearest counterfactuals and measure their fidelity to the underlying model. MACE [25] generates plausible and diverse nearest counterfactuals in a model-agnostic manner.

'Recourse' is defined as an individual's ability to change their outcome by altering their attributes, in a recovery process that is similar to the interventions to generate counterfactuals [41]. However, most counterfactual generation methods do not incorporate the cost (financial or otherwise) of these recovery actions. Ustun et al. [41] focus on the viability of providing recourse to individuals based on the interventions required and the effect of immutable attributes. Their study has inspired a number of strategies for equalizing recourse between sub-groups in classification models. One approach re-weights groups with large recourse costs relative to

groups with small recourse costs [17]. Another method applies causal-based "equality of effort" [21] for potential "discrimination removal" by adding new optimization constraints to the classifier. Karimi et al. [26] explore probabilistic methods to determine recourse and counterfactuals given limited causal knowledge.

Fairness in ranked lists has attracted attention with a focus on mitigating the position bias that leads to unfair representation in ranking [e.g., 8, 16, 32, 40, 49]. FA*IR [49] is a top-k ranking algorithm that aims to ensure that the proportion of protected individuals in every subset of a top-k ranking remains above some minimum threshold while maintaining the utility of the ranking. Singh and Joachims [40] compare exposure allocation with query relevance, motivated by different fairness constraints, to examine the utility trade-offs. Similarly, "equity of attention" aims to optimize fairness versus utility trade-offs by amortizing fairness accumulated across a series of rankings [8].

We introduce the notion of recourse fairness for ranked lists by imposing a constraint to ensure 'ranked group-level recourse fairness'. Current methods (e.g., FA*IR [49], FoEiR [40]) consider utility trade-offs without incorporating the costs incurred during re-ranking. We also introduce a block-based solution for recourse-aware re-ranking. This approach simultaneously processes multiple blocks, each with a different acceptance threshold.

3 FAIR RECOURSE-AWARE RANKING

Here, we introduce necessary notation and define the fairness constraints for our problem before outlining our re-ranking solution RAGUEL, its block-based variant, and extensions to our work.

3.1 Preliminaries

We define a database $\mathcal{D}$, with D records, in which each record is represented as a vector $\mathbf{x} \in \mathbb{R}^m$ with m attributes such that $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$. A fundamental part of our setting is the notion of recourse cost, which is the difficulty of changing a prediction by taking feasible actions to alter attribute values [41]. When we extend this to ranked lists, recourse cost is the difficulty of the actions to reach some ideal point or hyperplane (e.g., the top of the ranking). This chosen target is the counterfactual point $\mathbf{x}'$, calculated as: $\mathbf{x}' = \arg\min_{\mathbf{x}^* \in \mathcal{X}^*} d(\mathbf{x}, \mathbf{x}^*)$. Here, $d(\mathbf{x}, \mathbf{x}^*)$ denotes the distance function that quantifies the cost of the actions required to transfer from $\mathbf{x}$ to $\mathbf{x}^*$. That is, given a set of candidate counterfactual points $\mathcal{X}^*$, the counterfactual point to $\mathbf{x}$ is the point $\mathbf{x}' \in \mathcal{X}^*$ that has the minimum distance to $\mathbf{x}$.

For settings involving a classifier, we generate counterfactuals by considering the minimal actions to reach the decision boundary (e.g., negatively classified points trying to change their outcome). We use $f : \mathbb{R}^m \rightarrow \{-1, 1\}$ to denote the decision-making model that classifies records into two classes. This assumption is without loss of generality since any multi-class classifier can be considered as a stack of several binary classifiers. In this setting, $\mathcal{X}^*$ is the set of all candidate counterfactual points lying on the classifier boundary. For general ranking problems, we assume a single counterfactual point – the point at the top of the ranking – therefore we have $\mathcal{X}^* = \{\mathbf{x}'\}$. This can be generalized in future work by determining recourse with respect to multiple possible counterfactual points.

The recourse cost $c(\mathbf{x})$ is defined as the cost to reach $\mathbf{x}'$ from $\mathbf{x}$. That is, $c(\mathbf{x}) = d(\mathbf{x}, \mathbf{x}')$. In Example 1 (Section 3.4), the 'Cost' column of Table 2 reflects these recourse values based on the weighted Euclidean distance to the corresponding counterfactual on the boundary line, as seen in Figure 1. In our method for increasing fairness in representation and recourse, a chosen point, $\mathbf{x}$, is modified towards its counterfactual point, $\mathbf{x}'$, and $\widetilde{\mathbf{x}}$ denotes this modified point.

The database (or any subset thereof) can be divided into sub-groups, denoted as S_j, according to some protected attribute (e.g., gender, race, marital status) for which we are interested in evaluating fairness of representation and recourse. Without loss of generality and as is common practice in literature [21, 34], we illustrate our approach for two sub-groups, S_1 and S_2, and they are composed such that $|S_1 \cup S_2| = D$ and $|S_1 \cap S_2| = 0$. In any pair of sub-groups, where one sub-group contains records with a protected attribute, p denotes the proportion of the global database with this protected characteristic. That is, $p = |S_2|/D$, assuming S_2 contains the records associated with the protected characteristic. This represents the ideal proportion of protected records in any subset of the database.

X_k denotes some ordered subset of $\mathcal{D}$ containing $k \leq D$ points and the proportion of X_k that consists of records with the protected attribute is p_k. Finally, in our block-based approach, we segment the database into B blocks, with each block b as part of the set $\mathcal{B}$.

3.2 Fairness

We first introduce the simplest definition of fairness given two sub-groups, which ensures that their proportional representation in any subset correctly reflects that in the database.

Definition 3.1. Fair Representation Any subset $X \subseteq \mathcal{D}$ represents the protected sub-group fairly if, given some tolerance ϵ, $|p_X - p| \leq \epsilon$, where p_X is the proportion of protected records in X.

Given an ordered subset X_k, we can extend this definition to recursively hold for ordered subsets of the database.

Definition 3.2. Ranked Group-level Fair Representation An ordered subset X_k satisfies ranked group-level fair representation if, for every X_i (where $1 \leq i < k, k \geq 2$), it satisfies fair representation. Any singleton set X_1 is considered un-ranked and always fair.

Definition 3.2 is a reformulation of the sufficient condition for ranked group fairness in FA*IR [49]. However, while this definition aims to achieve balanced representation for every subset X_i, it does not take the cost of recourse into account. Therefore, we propose 'ranked group-level recourse fairness' as a constraint in the ranking.

To measure ranked fairness with respect to recourse, we first formalize the notion of recourse fairness between two sub-groups. The 'group-level recourse fairness ratio', r, assesses how balanced two sub-groups are in terms of the average cost of recovery actions:

$$r = \frac{\min(M_1, M_2)}{\max(M_1, M_2)} \qquad 0 \leq r \leq 1 \tag{1}$$

where M_1 and M_2 are the mean recourse costs for S_1 and S_2 respectively, defined as: $M_j = \frac{1}{|S_j|} \sum_{\mathbf{x}_i \in S_j} c(\mathbf{x}_i)$. Near-ideal fairness is represented by r-values close to 1, as the mean recourse costs for both sub-groups are similar, whereas r-values close to zero show that one sub-group is severely disadvantaged when trying to recover from an unwanted outcome.

Table 2: Original and re-ranked example data points

Name and Gender	Before Re-Ranking				Counter-factual		After Re-Ranking			
	LA	LD	Cost	Rk.	LA	LD	LA	LD	Cost	Rk.
Abdul M	3.5	6	0.33	1	3.06	6.11	3.5	6	0.33	1
Bogdan M	2	1	1.00	2	0.67	1.33	2	1	1.00	3
Chiara F+	4	4	1.33	3	2.22	4.44	**3.45**	**4**	**0.97**	**2**
Diana F+	5	4	2.00	4	2.33	4.67	5	4	2.00	4

M = male, F+ = female and all other gender identities, LA = loan amount in $'000, LD = loan duration in years, Cost = recourse cost, Rk. = rank

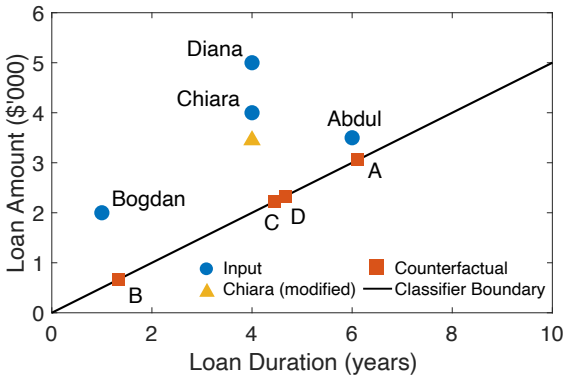


Figure 1: Example data points plotted

Definition 3.3. Group-Level Recourse Fairness Any subset $X \subseteq \mathcal{D}$ satisfies group-level recourse fairness if $r_i \geq 1 - \phi$.

Recourse fairness therefore ensures that the two sub-groups maintain a ratio of their mean recourse costs that is within some tolerance ϕ (distinct from ϵ). This may be extended to an ordered subset X_k in a manner similar to Definition 3.2, as described below.

Definition 3.4. Ranked Group-Level Recourse Fairness An ordered subset X_k satisfies ranked group-level recourse fairness if $r_i \geq 1 - \phi$, for every X_i (where $1 \leq i < k$). For any singleton set X_1, we always have $r_1 = 1$.

Definition 3.4 acts as a constraint for balancing recourse, in addition to Definition 3.2, which balances demographic parity.

3.3 Problem Statement

Given a dataset $\mathcal{D}$, we first obtain the ordered set X_D by ranking records according to their recourse costs to reach the chosen counterfactual points. The goal is to identify the actions needed to construct an ordered set of modified points $\widetilde{X}_D$ such that $\widetilde{X}_D$ satisfies Definitions 3.2 and 3.4, whilst ensuring that we minimize the extent of the modification, i.e., min $\frac{1}{D} \sum_i d(\mathbf{x}_i, \widetilde{\mathbf{x}}_i)$.

3.4 Examples

We illustrate our solution in the context of loan applications. We also discuss how RAGUEL can be applied to ranking-based problems such as fair webpage ranking.

1. Recourse-aware Loan Evaluations. Consider four loan applicants who have been rejected and placed onto a ranked waiting list. Abdul (rank #1) and Bogdan (#2) identify as male whereas Chiara (#3) and Diana (#4) identify differently. Table 2 and Figure 1 show this ranked list and the subsequent recourse-fair re-ranking after applying RAGUEL. In this example, the recourse cost is the cost for negatively classified points to reach their counterfactual points on the classifier boundary (i.e., the loan being granted) considering only the loan amount (LA) and loan duration (LD). Each applicant (*user*) is capable of specific actionable changes, such as changing the requested loan amount in $50 increments, or changing the duration of the requested loan in one year increments. Initially, despite fair representation, the average cost of recourse is higher for non-male customers, which may be due to them facing greater barriers towards gaining credit (as previously claimed in the real world [e.g., 2, 15, 28]). As a result, the bank (*platform*) may be interested in making conditional loan offers to lessen the extent of this disparity, subject to regulations. The platform's goal here would be to re-rank the list to achieve recourse fairness via minimal interventions (e.g., for widening participation of disadvantaged customers, or for inclusive representation due to policies and regulations). Since the top of the ranking shows disproportionate under-representation of female customers, RAGUEL identifies potential actions for Chiara (e.g., providing proof of her ability to cover $550 from other sources) that would move her up in the ranked list (e.g., a better chance of acceptance with a slightly smaller loan amount offered). In doing so, the bank improves ranked recourse fairness by enabling the disadvantaged group to perform actions to meet their conditional acceptance criteria. An alternative outcome is that the bank chooses to offer loans to applicants from the list (e.g., due to an availability of more funds), where candidates higher up in the list (now including Chiara) would secure their requested loans. RAGUEL provides a mechanism for both fair representation and fair recourse for all sub-groups with minimal re-ranking.

To utilize the block-based solution, the bank can generate equally-sized 'blocks' containing two individuals each, with individuals from the first block (Abdul and Bogdan) being offered a lower interest rate. Chiara might be offered conditional acceptance at the lower rate, provided she requests a shorter loan amount, moving her into the first block. If the bank has blocks containing four individuals, the block would already appear balanced with respect to gender, and no recourse would need to be offered.

2. Recourse-Aware Webpage Rankings. RAGUEL can be applied to any ranked list, such as top-k query results. Consider a set of web-pages (*users*), all of which would like to be listed on the first page of the search engine (*platform*) results (i.e., the top-10 web-pages). Unlike the case of loan applications, there is no classifier boundary, and the counterfactual is the query itself. Web-pages are ranked by their recourse costs, derived simply based on the difficulty of reaching the counterfactual. It is in the search engine's best interests to strive for "fair" ranking across all categories of web-pages (e.g., business, education, entertainment), while the organizations running the web-pages would like to know what recourse options they have to move up in the ranking (e.g., to help guide their search engine optimization). Considering recourse costs helps to identify the interventions (e.g., mobile-friendly formatting, faster page loading, adding hyperlinks) that can enable re-ordering with minimal cost. If the search engine were only concerned with

achieving recourse fairness on each page of results, the less granular block-based solution can be applied, with each block representing a page of search engine results.

3.5 Methodology

RAGUEL produces a 'recourse-aware ranking' that corrects group-level imbalances in terms of recourse and representation. The first step computes the recourse cost for each record to reach the relevant counterfactual point. Second, the records are ranked, in ascending order, according to their recourse costs. RAGUEL then identifies the minimal interventions applied to points such that the re-ranked list mitigates (or eliminates) any (group-level) unfairness. RAGUEL can also be extended to a block-based approach in which the ranking is subdivided into a finite number of blocks.

3.5.1 Identifying Counterfactuals and Recourse Costs. In the classification setting, such as Example 1, we treat the decision boundary as a query and use integer programming-based inverse classification [1, 29] to identify the feasible actions for a point to reach the boundary and flip its classification outcome. Scenarios where decisions are influenced by multiple factors (e.g., where the classification boundary is a hyperplane) require non-trivial solutions to efficiently find the counterfactual points for large datasets. By using generalized inverse classification, it is possible to include bounds on the allowable interventions, add a sparsity constraint that ensures fewer attributes are changed, and support non-linearity in recourse weights [29, 30]. Minimizing the cost of such interventions thereby produces the counterfactual point. For the general case of ranking, such as Example 2, we identify the interventions needed for each record to reach the top of the ranking, the nearest representative, or other appropriately defined counterfactual points.

We use weighted Euclidean distance in our experiments. This assumption is useful where no information exists about the classifier or ranking process, and only the 'recourse weight' for each attribute is known. Thereby, utility can be maintained by minimizing the cost of recourse interventions. Any other function for $d(\mathbf{x}, \mathbf{x}')$ may be used, if available. Hence, for each $\mathbf{x}_i \in \mathcal{D}$, our counterfactual $\mathbf{x}'_i$ is computed as:

$$\mathbf{x}'_i = \arg\min_{\mathbf{x}^*_j \in \mathcal{X}^*} \sqrt{\sum_{k=1}^{m} w_k |x^*_{j,k} - x_{i,k}|^2} \tag{2}$$

The recourse cost $c(\mathbf{x}_i)$ is thus $c(\mathbf{x}_i) = d(\mathbf{x}_i, \mathbf{x}'_i)$. Recourse weights (w_k) can be user-defined, assigned by experts, or learned from data. These weights are customizable to reflect the ease with which certain actions can be taken, and should be normalized. In Example 1, decreasing the requested loan amount ($w_{LA} = 0.5$) is easier than increasing the duration of the loan ($w_{LD} = 1$). One cannot change immutable attributes (e.g., race) or conditionally immutable attributes (e.g., marital status), which is reflected in the weights.

3.5.2 Recourse-Aware Re-Ranking. The next step is to apply interventions to ensure fairness in the ranked list. Definition 3.4 helps to identify minimal modifications that can re-rank the records towards more balanced group-level recourse costs alongside other fairness constraints. In this way, RAGUEL improves group-level recourse fairness by suggesting actions that expend minimal resources.

The steps are outlined in Algorithm 1. We start by defining $\widetilde{X}$, which grows one record at a time and will eventually represent the ranked database of modified points (Lines 2–3). If X_i satisfies Definition 3.2, $\mathbf{x}_i$ is added to $\widetilde{X}$ (i.e., we maintain the status quo) (Lines 4–5). Otherwise, as the addition of $\mathbf{x}_i$ would result in $\widetilde{X}$ no longer satisfying ranked fair representation, we consider substituting $\mathbf{x}_i$ for a lower ranked point. Substitution is possible if interventions on a lower ranked point decrease its recourse cost such that it gains a higher rank than $\mathbf{x}_i$. Hence, the algorithm iterates through the remaining records and, if Definition 3.2 is satisfied by one of these points $\mathbf{x}_j$, it proceeds to modification (Lines 6-8). Modification occurs by iteratively and minimally modifying attribute values of $\mathbf{x}_j$ into $\widetilde{\mathbf{x}}_j$ until Definitions 3.2 and 3.4 are satisfied by the resulting new ordered subset of points (Line 9-11).

Algorithm 1 Recourse-Aware Re-Ranking

```
1:  function RE-RANKING(𝒟)
2:    X̃ ← ∅
3:    for 1 ≤ i < D do
4:      if X_i ⊭ Definition 3.2 then
5:        X̃ ← X̃ ∪ x_i
6:      else
7:        for i + 1 ≤ j ≤ D do
8:          if X̃ ∪ x_j ⊨ Definition 3.2 then
9:            Modify x_j into x̃_j until X̃ ∪ x̃_j ⊨ Definition 3.4
10:           X̃ ← X̃ ∪ x̃_j
11:           break
```

Attributes are modified in order of their recourse weights to prioritize interventions with lowest recourse weights. This is because sparsity of the recourse vector can make recourse options more understandable to users [30]. If no amount of modification leads to the desired ranking, the attribute is reset to its original value and the next feasible attribute is modified, and so on, followed by combinations of attributes in the same order, if necessary. That is, if three attributes (A, B, C) exist with $w_A < w_B < w_C$, modifications are attempted in the following order: A, B, C, A&B, A&C, B&C, and A&B&C. In Example 1, updating Chiara's loan amount in \$50 increments leads to the given re-ranking with minimal cost of intervention. In the unlikely event that no re-ranking is possible at any stage (e.g., when tolerance margins are tight), the algorithm fails to satisfy the fairness constraints and copies the remaining ranked list as is. This exit strategy is rarely needed (on average, 10% of the blocks at the strictest tolerance setting in our experiments require it). At any stage during re-ranking, we expect at least $p - \epsilon$ points to be available to be re-ranked since this proportion was successfully maintained by the fairness constraint so far.

3.5.3 Block-Based Recourse-Aware Re-Ranking. Until now, RAGUEL has focused on ranked lists with recourse costs relative to a *single* query (e.g., classifier boundary, top-k query). There are applications where the ranking needs to be done with respect to *multiple* queries that have an inherent ranking (e.g., interest rate brackets in Example 1, search engine result pages in Example 2). To enable diversity in the set of records that are offered recourse interventions, it is useful to consider fairness within groups of points that are similar/clustered with respect to a classification model or search query, by considering these as independent blocks for re-ranking. Also, it is often not possible to modify an attribute by the precise amount needed to re-rank it such that Definition 3.2 is satisfied.

In these cases, a block-based approach is more suitable as it can handle multiple queries, compare clusters separately, and perform less granular re-ranking.

Algorithm 2 outlines the block-based re-ranking process that handles these cases. As a pre-processing step, we subdivide the ranking into a set of blocks, $\mathcal{B}$. If there are B blocks each with n records, b_1 contains the first n records, b_2 contains the records with ranks $n+1$ to $2n$, etc. In our experiments, the number of blocks determines the number of points in each block, which is kept constant throughout re-ranking. We assume that recourse costs within a block (and thus the ranking order) are close enough to one another that we can focus on satisfying our fairness conditions at the less-granular block-level. And so, although irregularly sized blocks (e.g., variable-width histogram bins, clusters) can be used, the recourse cost range within each block should be minimized to ensure high accuracy and fairness.

If b_i does not satisfy fair representation (Definition 3.1), its lowest ranked point block is moved into b_{i+1} (Lines 1–4). The highest ranking feasible point from b_{i+1} replaces it, with interventions performed to modify its ranking (Lines 4–6).

This overall approach for RAGUEL-Block is similar to Algorithm 1, with an additional constraint, which is:

$$\begin{aligned} c(\widetilde{\mathbf{x}}) &\leq \beta \qquad \text{if } M_1 \leq M_2 \\ c(\widetilde{\mathbf{x}}) &\geq \beta \qquad \text{if } M_1 \geq M_2 \end{aligned} \tag{3}$$

where $\bar{\mathbf{x}}$ is removed from b and the bound, β, is defined as:

$$\beta = (M_1|S_1| - c(\bar{\mathbf{x}}))\frac{M_2}{M_1}\frac{|S_2|+1}{|S_1|-1} - M_2|S_2|$$

This constraint guarantees that the interventions used to improve fair representation also improve recourse fairness.

Theorem 3.1. *Minimal modification of a data point that results in improved fair representation of the block is guaranteed to improve recourse fairness of the block if the constraint in Equation 3 is met.*

Proof. Consider data points $\mathbf{x}_i$ in a block of size n. Each $\mathbf{x}_i \in b$ has a recourse cost $c(\mathbf{x}_i)$ to its respective target counterfactual point $\mathbf{x}'_i$. Assume the block, b_i, is divided into subgroups S_1 and S_2 based on some sensitive attribute, with S_2 being under-represented (i.e., $|S_2| < |S_1|$). Within b, the mean costs are:

$$M_1 = \frac{1}{|S_1|}\sum_{\mathbf{x}_i \in S_1} c(\mathbf{x}_i) \qquad M_2 = \frac{1}{|S_2|}\sum_{\mathbf{x}_i \in S_2} c(\mathbf{x}_i)$$

Then, with the modified point $\widetilde{\mathbf{x}}$ (of the under-represented subgroup) from b_{i+1}, we aim to improve the proportion $\frac{|S_2|}{n}$ as per Definition 3.1. The block size is maintained, thus the lowest ranked point $\bar{\mathbf{x}}$ (of the over-represented sub-group) is pushed into the next block, b_{i+1}. The new mean total costs are:

$$M'_1 = \frac{-c(\bar{\mathbf{x}})+\sum_{i\in S_1} c(\mathbf{x}_i)}{|S_1|-1} \qquad M'_2 = \frac{c(\widetilde{\mathbf{x}})+\sum_{i\in S_2} c(\mathbf{x}_i)}{|S_2|+1}$$

In the trivial case where $|S_1| = n$ and $|S_2| = 0$, we know that $M_2 = 0$ and any choice of $\widetilde{\mathbf{x}}$ will improve the ratio $\frac{M'_2}{M'_1}$. In cases where $|S_1| > |S_2| > 0$, we have two situations: $\bar{\mathbf{x}} \in S_1$ or $\bar{\mathbf{x}} \in S_2$. The latter is not a permitted perturbation as it makes no improvement to fair representation. When $\bar{\mathbf{x}} \in S_1$, by substituting M_1 and M_2 into the expression for $\frac{M'_2}{M'_1}$, we obtain:

$$\begin{aligned} \frac{M'_2}{M'_1} &= \frac{c(\widetilde{\mathbf{x}}) + \sum_{i\in S_2} c(\mathbf{x}_i)}{-c(\bar{\mathbf{x}}) + \sum_{i\in S_1} c(\mathbf{x}_i)} \times \frac{|S_1|-1}{|S_2|+1} \\ &= \frac{M_2|S_2| + c(\widetilde{\mathbf{x}})}{M_1|S_1| - c(\bar{\mathbf{x}})} \times \frac{|S_1|-1}{|S_2|+1} \end{aligned}$$

Thus, $$c(\widetilde{\mathbf{x}}) = (M_1|S_1| - c(\bar{\mathbf{x}}))\frac{M'_2}{M'_1}\frac{|S_2|+1}{|S_1|-1} - M_2|S_2|$$

Consider the bound given by: $\beta = (M_1|S_1| - c(\bar{\mathbf{x}}))\frac{M_2}{M_1}\frac{|S_2|+1}{|S_1|-1} - M_2|S_2|$. If $M_1 \leq M_2$ and $c(\widetilde{\mathbf{x}}) \leq \beta$, it follows that $\frac{M'_2}{M'_1} < \frac{M_2}{M_1}$. Similarly, if $M_1 \geq M_2$ and $c(\widetilde{\mathbf{x}}) \geq \beta$, it follows that $\frac{M'_2}{M'_1} > \frac{M_2}{M_1}$. In either case, the new ratio $r' = \frac{\min(M'_1, M'_2)}{\max(M'_1, M'_2)}$ moves closer to 1 and thus recourse fairness improves within the block. □

Algorithm 2 Block-Based Recourse-Aware Re-ranking

1: **function** BlockBased($\mathcal{D}$, $\mathcal{B}$)
2: **for** $1 \leq i < B$ **do**
3: **while** $b_i \not\models$ Definition 3.1 **do**
4: Move last element of b_i into b_{i+1}
5: Find highest ranked $\mathbf{x} \in b_{i+1}$ s.t. $b_i \cup \mathbf{x} \models$ Definition 3.1
6: **repeat**
7: Modify $\mathbf{x}$ into $\widetilde{\mathbf{x}}$ while satisfying Equation 3
8: **until** $\widetilde{\mathbf{x}} \in b_i$ and $b_i \cup \widetilde{\mathbf{x}} \models$ Definition 3.4

Note that points can only move up/down by one block when they are re-ranked. This restricts the permitted cost of modification, and means that the exit strategy (introduced in Section 3.5.2) is more relevant in RAGUEL-Block. In situations where no feasible modification is possible, the block is copied as is and we proceed to the next block (illustrated in Figure 3 and Table 6).

3.6 Extensions

RAGUEL can be extended in a number of directions, which we briefly discuss here. First, as alluded to in Section 3.1, to cater for multi-class classifiers, one can stack several binary classifiers. That is, fairness can be ensured across all outcomes by considering all possible one-versus-rest binary classifiers.

Second, our fairness definitions (and mechanism) can be generalized for more than two sub-groups. For example, a company may want to consider fairness between under-18s, adults, and over-65s. In this case, Definition 3.1 can be formalized such that X only offers fair representation if $|p_X - p_i| \leq \epsilon$ for all protected sub-groups. A revised Definition 3.2 follows naturally. For Definition 3.4, the redefined r must consider fairness between multiple sub-groups. For n sub-groups, the group-level recourse fairness ratio is:

$$r = \frac{\min(M_1, \ldots, M_n)}{\max(M_1, \ldots, M_n)} \tag{4}$$

Finally, RAGUEL can also be applied to non-linear models (e.g., neural networks) when recourse-unaware methods are used to compute counterfactuals and initial rankings. Although methods are yet to be defined that can declare unfairness of recourse in non-linear settings [41], doing so, and incorporating recourse weights to determine these rankings, are interesting research challenges.

4 EXPERIMENTAL EVALUATION

We evaluate RAGUEL using three real-world datasets. We first examine the recourse fairness of both the initial data and the post-processed data using the current 'fair classifiers' (Section 4.2). We then study the effectiveness of RAGUEL at generating counterfactuals (4.3) and in re-ranking (4.4), and compare them with competitive baselines. This is followed by an analysis of RAGUEL-Block (4.5).

All code is written in Python 3 and we use Gurobi for the optimization problem of counterfactual generation. Experiments are conducted on macOS 10.15 with 2.4 GHz CPU and 8 GB RAM.

4.1 Data

We apply RAGUEL to the challenge of automated customer credit and liability decision-making using the 'German Credit' [12], 'Default of Credit Card Clients' (DC3) [12], and HMDA [14] datasets. The German Credit dataset contains 1,000 records and has 20 features relating to individuals' financial and personal details, such as purpose of loan, missed payments, and marital status. A binary class label indicates individuals who are (un)successful in their loan application. The DC3 dataset has 30,000 records and 24 features, and class labels indicating individuals who default on their payments. For both datasets, we convert categorical attributes into actionable numerical attributes (e.g., length of employment) or binarize them (e.g., has guarantor), and aggregate some columns (e.g., months with zero balance) to construct more relevant features that can easily reflect infeasible recourse weights. After these modifications, the German Credit and DC3 datasets have 18 and 10 features, respectively. The HMDA dataset is based on the Home Mortgage Disclosure Act by the US government, and reports public loan data that can be filtered by year, geography, financial institution, and features such as the type/purpose/duration of loan, gender/race of applicants, or property type. The original data comprises over 25M records, with 5M of these loans being clearly labeled as accepted/rejected. The 93 feature columns are filtered down to 26 where only the columns that reflect aggregated data are preserved. This large HMDA dataset is only used for scalability experiments on RAGUEL, since none of the baselines can operate on this many records. We assume that all attributes are independent of each other in these datasets. We consider marital status as the protected attribute from literature for German Credit and DC3 [7, 24] and use gender for HMDA (as the marital status attribute is unavailable). As the classifier boundary (i.e., threshold) is defined based on the accept/reject labels in the real data, we only consider this boundary. Alternative thresholds are possible (as are other types of classifiers).

4.2 Group-Level Recourse Fairness Analysis

We first examine the effect of existing post-processing techniques, which are designed to improve fairness by re-classifying data points. 'Demographic parity' ensures that the acceptance rate is equal across sub-groups. 'Equalized odds' (and its relaxed version, 'equal opportunity') instead enforces fairness only among individuals who reach similar outcomes, by equalizing false positive/negative error rates or both ('weighted') [19, 35]. While these methods do have different objectives, we include them to indicate how existing fairness definitions do not subsume the recourse-based definition and, as shown in Table 3, when these methods fail to improve recourse fairness, they actually exacerbate the problem significantly. The 'Recourse Cost' column demonstrates how RAGUEL's aggregate weighted distance measure across sub-groups is lowered compared to the initial ranking, whereas measures that shift the classifier boundary worsen this distance. This means that disadvantaged individuals under the post-processed classifier model would face even greater difficulty in improving their classification outcome. RAGUEL, however, increases recourse fairness by 15% compared to the initial data, and up to 200% compared to the alternatives, whilst also needing fewer points to be modified. RAGUEL-Block reduces the cost of interventions (difference in recourse cost) by 130% compared to RAGUEL. This guarantees that the re-ranked list remains closer in accuracy to the original ranked list, due to looser, less granular fairness constraints.

Table 3: Recourse cost and fairness disparities (DC3)

Post-Processed Data	Sub-Group	# Points Changed	Recourse Cost	r
Initial	Single	0	7.688	0.759
	Married	0	5.837	
Demographic Parity	Single	2771	6.023	0.472
	Married	7	2.843	
Equalized Odds (false negative)	Single	474	6.238	0.511
	Married	101	3.189	
Equalized Odds (false positive)	Single	76	6.296	0.542
	Married	4910	3.412	
Equalized Odds (weighted)	Single	76	6.296	0.422
	Married	4944	2.658	
FA*IR	Single	0	7.611	0.554
	Married	57	4.217	
FoEiR-DP	Single	12	6.788	0.793
	Married	7	5.383	
RAGUEL	Single	135	6.663	0.876
	Married	23	5.837	
RAGUEL-Block ($B = 25, \tau = \frac{1}{3}$)	Single	18	6.891	0.847
	Married	49	5.837	

Table 3 also includes results for two fair-ranking methods – FA*IR [49] and FoEiR [40]. FA*IR hurts recourse fairness and can only select up to 400 points, with RAGUEL attaining a ratio that is 37% better. FoEiR can only re-rank up to 50 points due to inefficiency, but uses an exposure utility measure that can slightly improve recourse, although not as well as RAGUEL. Average results over 50 trials are presented for both. The smaller German Credit data exhibited similar characteristics and is therefore omitted.

4.3 Counterfactual Analysis

Table 4 compares our explainable and efficient approach against two alternatives for generating counterfactuals. MACE [25] uses formal verification techniques and satisfiability solvers to generate counterfactuals, whereas AR [41] enumerates feasible "flipsets" (counterfactuals) for recommending and auditing recourse. We present a qualitative comparison in terms of average closeness to the training data (Euclidean distance of counterfactual to nearest neighbor), average sparsity (number of attributes modified),

Table 4: Comparison of counterfactual generation methods

CF Method	German Close.	German Spars.	German Time (s)	DC3 Close.	DC3 Spars.	DC3 Time (s)
MACE	3095.47	2.23	1.713	2082.83	3.84	0.642
AR	9.47	1.60	0.003	192.51	1.65	0.003
RAGUEL-CF	1.42	1.36	0.003	72.04	2.68	0.004

and average runtime (per counterfactual generated) [44]. RAGUEL sometimes shows higher sparsity as different recourse weights may mean that smaller modifications are made to a higher number of attributes. However, our resulting counterfactuals are much closer to the original distribution of points, which is ultimately more important. The efficiency of RAGUEL is comparable with that of AR, and both outperform MACE. Note that MACE and AR merely identify counterfactual points but do not modify the data, classification, or ranking. Hence, their influence on ranking fairness can only be measured in tandem with other methods for fair ranking.

4.4 Re-Ranking Analysis

We now consider RAGUEL's effectiveness at re-ranking $\mathcal{D}$ to achieve improved recourse fairness. Before doing so, we introduce the user-specified 'tolerance' parameter, τ, which is used to determine the value for ϵ, where $\epsilon = \tau \times p$. Our default setting is $\tau = \frac{1}{3}$. For example, if 30% of the database belongs to a protected sub-group and $\tau = \frac{1}{3}$, then $\epsilon = 30 \times \frac{1}{3} = 10\%$. This means that the proportion of protected individuals in each block should be in the range 30±10%.

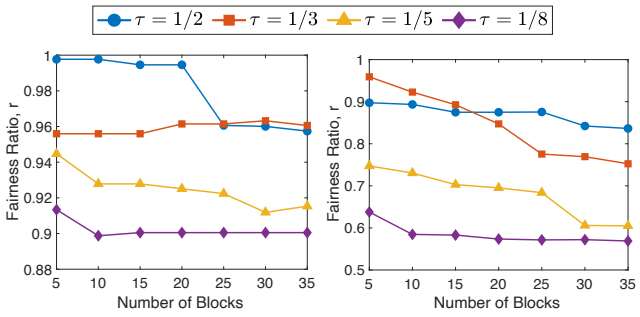
We compare RAGUEL with state-of-the-art fair ranking baselines, where points are originally ranked according to their distance to counterfactuals, as computed by each of the counterfactual generation methods from Section 4.3. FA*IR [49] selects the top-k points ranked by relevance scores, to satisfy demographic parity without compromising on selection utility. FoEiR [40] balances exposure allocation in terms of three different fairness definitions (demographic parity, DP; disparate treatment, DT; and disparate impact, DI). Since FA*IR was infeasible on larger datasets, we report the average values over 50 trials, sampling 400 points each time. Empirically, we find that setting α, which is a FA*IR-specific parameter, to 0.15 offers the best results and we use this setting throughout. Similarly, FoEiR could only handle 50 points, and so we similarly obtain average results after 50 trials. In comparison, RAGUEL runs efficiently (<2.5 seconds) on both full datasets, and it is 4x faster than FoEiR on its smaller sample.

We evaluate these results with respect to different ranking quality metrics (normalized discounted KL-divergence, rKL; normalized discounted difference, rND; normalized discounted ratio, rRD) [48]. rND indicates the difference between the protected sub-group's proportion among top-k records and overall, while rKL uses Kullback-Leibler divergence for the expectation of this difference. rRD computes a similar difference to rND but between the ratios of the (minority) protected sub-group to the majority.

Table 5 shows that RAGUEL generally outperforms the alternatives on both datasets, consistently maintaining or improving these ranking metrics compared to the initial ranked list. FoEiR generally achieves better scores with the recourse-unaware MACE, while FA*IR performs comparatively well with AR. RAGUEL allows any combination of techniques to be applied for re-ranking.

Table 5: Comparison of fair ranking methods

CF	Method for... Ranking	German rKL	German rND	German rRD	DC3 rKL	DC3 rND	DC3 rRD
MACE	Initial	0.051	0.194	**0.000**	0.129	**0.248**	**0.042**
	FA*IR	0.051	0.196	**0.000**	0.124	0.249	**0.042**
	FoEiR-DP	0.049	0.188	0.009	0.070	0.229	0.061
	FoEiR-DT	0.049	0.188	0.009	**0.069**	0.229	0.061
	FoEiR-DI	0.048	0.186	0.009	**0.069**	0.229	0.061
	RAGUEL-Rk	**0.048**	**0.185**	**0.000**	0.129	**0.248**	**0.042**
AR	Initial	0.054	0.223	**0.000**	**0.013**	0.112	0.223
	FA*IR	0.066	0.265	**0.000**	0.021	0.125	**0.205**
	FoEiR-DP	0.080	0.231	0.035	0.038	0.183	0.246
	FoEiR-DT	0.080	0.232	0.035	0.038	0.183	0.246
	FoEiR-DI	0.079	0.229	0.039	0.036	0.185	0.247
	RAGUEL-Rk	**0.051**	**0.218**	**0.000**	**0.013**	**0.111**	0.221
RAGUEL-CF	Initial	0.026	0.085	0.121	0.025	**0.153**	0.299
	FA*IR	0.118	0.316	0.033	0.026	0.148	**0.262**
	FoEiR-DP	0.052	0.165	0.170	0.039	0.190	0.276
	FoEiR-DT	0.052	0.165	0.170	0.041	0.190	0.276
	FoEiR-DI	0.052	0.172	0.168	0.040	0.191	0.273
	RAGUEL-Rk	**0.005**	**0.047**	**0.032**	**0.023**	**0.153**	0.298



(a) German Credit dataset **(b) DC3 dataset**

Figure 2: Variation of group-level recourse fairness with τ and B. Note the false origins in both plots.

4.5 Block-Based Re-Ranking Analysis

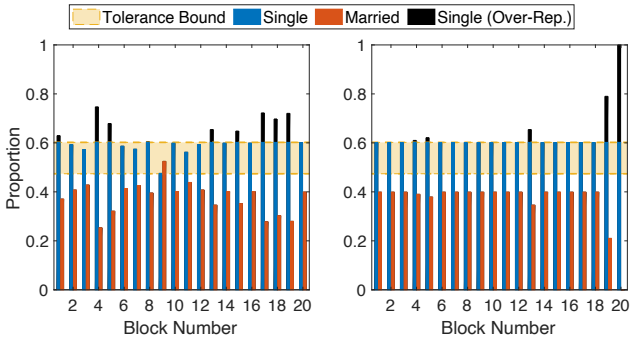
We now examine the effectiveness of performing block-based re-ranking by considering different granularities and strictness. We assume all blocks are equal in size with block size being governed by the number of blocks into which the ranked list is sub-divided.

Figure 2 shows how group-level recourse fairness changes with the number of blocks and the tolerance. Recourse fairness decreases as the number of blocks increases as there is a more restricted range of values into which attributes can be perturbed for re-ranking. RAGUEL-Block is relatively stable to an increase in B, although there is an expected decrease in r as it becomes harder to satisfy the fairness constraints for small blocks. Similarly, while low τ-values lead to recourse fairness being more strictly enforced within each block, a very low τ-value results in increased unfairness by forcing corrections to more blocks (especially those encountered early).

We also study the effectiveness of our re-ranking approach by considering the number of unfair blocks that are transformed into fair blocks through re-ranking (Table 6). As expected, with high

Table 6: Effectiveness of RAGUEL-Block (DC3); initial and final number of unfair blocks shown; number in brackets denotes number of blocks made fair

B	Tolerance, τ			
	1/2	1/3	1/5	1/8
5	0 → 0 (0)	1 → 0 (1)	2 → 1 (1)	1 → 1 (0)
10	0 → 0 (0)	3 → 0 (3)	5 → 1 (4)	2 → 2 (0)
15	2 → 0 (2)	6 → 1 (5)	8 → 2 (6)	5 → 4 (1)
20	2 → 0 (2)	8 → 1 (7)	10 → 2 (8)	9 → 5 (4)
25	3 → 0 (3)	13 → 1 (12)	12 → 3 (9)	14 → 7 (7)
30	6 → 0 (6)	13 → 1 (12)	14 → 3 (11)	13 → 9 (4)
35	7 → 0 (7)	14 → 1 (13)	19 → 5 (14)	21 → 11 (10)



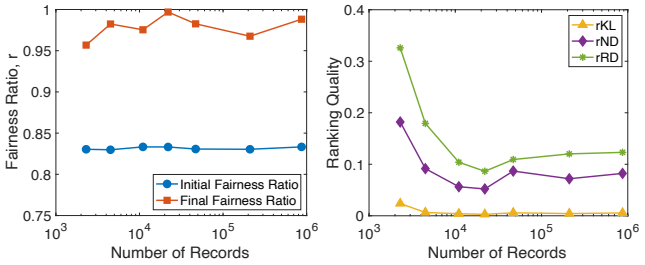
(a) Before **(b) After**

Figure 3: Group proportions before and after re-ranking

tolerance and very few (large) blocks, the dataset is often already deemed to be fair or it is easily made fair. As the number of blocks is increased and each block thereby becomes smaller, re-ranking has a greater impact. Notably, it is more effective to reduce the size of blocks than to reduce the tolerance, as a lower tolerance leads RAGUEL-Block to be less likely to create fair blocks.

Figure 3 shows the proportion of the two sub-groups in each block before and after re-ranking ($B = 20, \tau = \frac{1}{8}$, DC3). The allowable tolerance margin around the ideal proportion, based on the overall dataset, is also indicated. Given the tolerance bound, the original dataset shows that nine of the 20 blocks are unfair initially with only five blocks remaining unfair after re-ranking. This highlights the impact of using a stricter tolerance as more unfair blocks remain, compared to that at lower τ values. However, these blocks become concentrated at the bottom of the ranking, with the last block in particular becoming more unfair as there is no opportunity for its fairness to be corrected. With a looser tolerance, earlier blocks containing slightly unfair representations can be ignored during re-ranking, which prevents later blocks from accumulating highly imbalanced outliers.

In terms of runtime, when B is low and/or τ is large, RAGUEL-Block runs in less than two seconds on the larger DC3 dataset. As B increases and/or τ decreases, runtime increases by up to nine times – 14.3s ($B = 35, \tau = \frac{1}{5}$) vs. 1.7s ($B = 5, \tau = \frac{1}{2}$). This is to be expected as the mechanism needs to make more satisfiability checks due to the tighter constraints and the larger number of blocks. Moreover, RAGUEL-Block is nearly 30x faster, while changing the fairness ratio by no more than 10% (up to $B = 25$ and $\tau = \frac{1}{3}$ on DC3).



(a) Fairness Ratio **(b) rKL, rND, and rRD**

Figure 4: Effect on recourse fairness and ranking quality as the number of records in the dataset varies

4.6 Scalability Analysis

Finally, we demonstrate that RAGUEL-Block runs efficiently on large-scale data such as the HMDA dataset, with performance results that remain stable even as the number of records increases. None of the re-ranking alternatives (e.g., FA*IR and FoEiR) could handle large datasets, so they are excluded here. We sample between 2,500 and 1M records HMDA dataset to maintain an initial fairness ratio of 0.83. We run RAGUEL-Block using a fixed block size of 100 records (i.e., the total number of blocks is variable) on these samples, presenting results averaged over 10 trials. Figure 4a shows that the final fairness ratio attained is at least 0.95 and does not exhibit much variability as dataset size increases. The ranking quality metrics (Figure 4b), which are high when the number of records is small, quickly decrease and remain low for larger datasets.

These results show that our solution can easily handle any arbitrarily large dataset that is sub-divided into smaller blocks within which group-level recourse fairness is being enforced. Our method essentially uses sliding window processing, as a block only requires the next adjoining block of candidate records for RAGUEL-Block to improve its fairness (see Algorithm 2 and Section 3.5.3).

5 CONCLUSION

RAGUEL is a solution for improving group-level fairness in ranked lists with respect to both representation- *and* recourse-based constraints. RAGUEL performs iterative computations to identify feasible recourse actions that require minimal cost and can result in a fairer ranked list. A block-based extension can handle adjustable granularities and multiple target points or boundaries. While these new recourse-based approaches have recently been considered more in the context of societal fairness where appropriate policy measures are in place for disadvantaged groups, they are more broadly applicable to technical optimization tasks where fairness is a constraint or part of the objective, such as resource allocation in computer systems or distributing funding to organizations. We leave these applications to future work.

ACKNOWLEDGMENTS

This work is supported in part by the UK Engineering and Physical Sciences Research Council under Grant No. EP/L016400/1. Aparajita is supported via a Feuer International Scholarship in Artificial Intelligence. We thank Efehan Madran, Aaron MacFarlane, and Amina Tkhamokova for their valuable contributions.

REFERENCES

[1] Charu C. Aggarwal, Chen Chen, and Jiawei Han. 2010. The inverse classification problem. *Journal of Computer Science and Technology* 25, 3 (2010), 458–468.

[2] Alberto F. Alesina, Francesca Lotti, and Paolo Emilio Mistrulli. 2013. Do Women Pay More for Credit? Evidence From Italy. *Journal of the European Economic Association* 11 (2013), 45–66.

[3] Nasim Arianpoo and Victor Leung. 2016. How network monitoring and reinforcement learning can improve TCP fairness in wireless multi-hop networks. *EURASIP Journal on Wireless Communications and Networking* 3 (2016), 1–15.

[4] Boston Federal Reserve Bank. 1997. Closing the gap: a guide to equal opportunity lending.

[5] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2017. Fairness in machine learning. *NeurIPS Tutorial* 1 (2017), 2017.

[6] Fuat Basık, Bugra Gedik, Hakan Ferhatosmanoglu, and Kun-Lung Wu. 2018. Fair task allocation in crowdsourced delivery. *IEEE Transactions on Services Computing* (2018), 1040–1053.

[7] Vaishnavi Bhargava, Miguel Couceiro, and Amedeo Napoli. 2020. LimeOut: An Ensemble Approach To Improve Process Fairness. In *ECML PKDD International Workshop on eXplainable Knowledge Discovery in Data Mining (XKDD 2020)*.

[8] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *ACM SIGIR*. 405–414.

[9] Nicola Capuano, Carmen De Maio, Saverio Salerno, and Daniele Toti. 2014. A methodology based on commonsense knowledge and ontologies for the automatic classification of legal cases. In *ACM Conference on Web Intelligence, Mining and Semantics*. 27.

[10] Sam Corbett-Davies and Sharad Goel. 2018. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023* (2018).

[11] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. In *ACM SIGKDD*. 797–806.

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

[13] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *ACM SIGKDD*. 259–268.

[14] Federal Financial Institutions Examination Council's (FFIEC). 2021. HMDA - Home Mortgage Disclosure act. https://ffiec.cfpb.gov/

[15] Martha L. Garrison. 1976. Credit-Ability for Women. *The Family Coordinator* 25, 3 (1976), 241–248.

[16] Ahmad Ghizzawi, Julien Marinescu, Shady Elbassuoni, Sihem Amer-Yahia, and Gilles Bisson. 2019. Fairank: An interactive system to explore fairness of ranking in online job marketplaces. In *EDBT*.

[17] Vivek Gupta, Pegah Nokhiz, Chitradeep Dutta Roy, and Suresh Venkatasubramanian. 2019. Equalizing recourse across groups. *arXiv preprint arXiv:1909.03166* (2019).

[18] Sara Hajian, Francesco Bonchi, and Carlos Castillo. 2016. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *ACM SIGKDD*. 2125–2126.

[19] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *NeurIPS*. 3315–3323.

[20] Paul R. Harper. 2005. A review and comparison of classification algorithms for medical decision making. *Health Policy* 71, 3 (2005), 315–331.

[21] Wen Huan, Yongkai Wu, Lu Zhang, and Xintao Wu. 2020. Fairness through equality of effort. In *Companion Proceedings of the Web Conference 2020*. 743–751.

[22] Zhongjun Jin, Mengjing Xu, Chenkai Sun, Abolfazl Asudeh, and HV Jagadish. 2020. Mithracoverage: a system for investigating population bias for intersectional fairness. In *ACM SIGMOD*. 2721–2724.

[23] Faisal Kamiran and Toon Calders. 2009. Classifying without discriminating. In *IEEE Conference on Computer, Control and Communication*. 1–6.

[24] Jian Kang, Tiankai Xie, Xintao Wu, Ross Maciejewski, and Hanghang Tong. 2021. MultiFair: Multi-Group Fairness in Machine Learning. *arXiv preprint arXiv:2105.11069* (2021).

[25] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. 2020. Model-agnostic counterfactual explanations for consequential decisions. In *AISTATS*. PMLR, 895–905.

[26] Amir-Hossein Karimi, Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. 2020. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. In *NeurIPS*.

[27] Matt J. Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. In *NeurIPS*. 4066–4076.

[28] Helen F. Ladd. 1982. Equal Credit Opportunity: Women and Mortgage Credit. *The American Economic Review* 72, 2 (1982), 166–170.

[29] Michael T. Lash, Qihang Lin, Nick Street, Jennifer G. Robinson, and Jeffrey Ohlmann. 2017. Generalized inverse classification. In *SIAM International Conference on Data Mining*. SIAM, 162–170.

[30] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2018. Comparison-based inverse classification for interpretability in machine learning. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. 100–111.

[31] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–35.

[32] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. 2018. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *ACM CIKM*. 2243–2251.

[33] Ioannis Paparrizos, B Barla Cambazoglu, and Aristides Gionis. 2011. Machine learned job recommendation. In *Proceedings of the fifth ACM Conference on Recommender Systems*. 325–328.

[34] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. 2021. Fairness in rankings and recommendations: An overview. *The VLDB Journal* (2021), 1–28.

[35] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q. Weinberger. 2017. On fairness and calibration. In *NeurIPS*. 5680–5689.

[36] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *ACM SIGKDD*. 1135–1144.

[37] Babak Salimi, Bill Howe, and Dan Suciu. 2020. Database repair meets algorithmic fairness. *ACM SIGMOD Record* 49, 1 (2020), 34–41.

[38] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. 2019. Interventional fairness: Causal database repair for algorithmic fairness. In *ACM SIGMOD*. 793–810.

[39] Suproteem K. Sarkar, Kojin Oshiba, Daniel Giebisch, and Yaron Singer. 2018. Robust Classification of Financial Risk. *arXiv preprint arXiv:1811.11079* (2018).

[40] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *ACM SIGKDD*. 2219–2228.

[41] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable recourse in linear classification. In *ACM Conference on Fairness, Accountability, and Transparency*. 10–19.

[42] Salvatore Valenti, Francesca Neri, and Alessandro Cucchiarelli. 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education: Research* 2, 1 (2003), 319–330.

[43] Suresh Venkatasubramanian. 2019. Algorithmic fairness: Measures, methods and representations. In *ACM PODS*. 481–481.

[44] Sahil Verma, John Dickerson, and Keegan Hines. 2020. Counterfactual Explanations for Machine Learning: A Review. *arXiv:2010.10596* (2020).

[45] Julius von Kügelgen, Amir-Hossein Karimi, Umang Bhatt, Isabel Valera, Adrian Weller, and Bernhard Schölkopf. 2020. On the fairness of causal algorithmic recourse. *arXiv:2010.06529* (2020).

[46] Hao Wang, Berk Ustun, and Flavio Calmon. 2019. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. In *ICML*. PMLR, 6618–6627.

[47] Adam White and Artur d'Avila Garcez. 2019. Measurable counterfactual local explanations for any classifier. *arXiv:1908.03020* (2019).

[48] Ke Yang and Julia Stoyanovich. 2017. Measuring fairness in ranked outputs. In *SSDBM*. 1–6.

[49] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. Fa* ir: A fair top-k ranking algorithm. In *ACM CIKM*. 1569–1578.

[50] Hantian Zhang, Xu Chu, Abolfazl Asudeh, and Shamkant B. Navathe. 2021. OmniFair: A Declarative System for Model-Agnostic Group Fairness in Machine Learning. In *ACM SIGMOD*. 2076–2088.

[51] Junzhe Zhang and Elias Bareinboim. 2018. Fairness in decision-making—the causal explanation formula. In *AAAI*.