GLOBAL COUNTERFACTUAL EXPLANATIONS: INVESTIGATIONS, IMPLEMENTATIONS AND IMPROVEMENTS

Dan Ley, Saumitra Mishra, Daniele Magazzeni
J.P. Morgan AI Research, London, UK
{dan.ley, saumitra.mishra, daniele.magazzeni}@jpmorgan.com

ABSTRACT

Counterfactual explanations have been widely studied in explainability, with a range of application dependent methods emerging in fairness, recourse and model understanding. However, the major shortcoming associated with these methods is their inability to provide explanations beyond the local or instance-level. While some works touch upon the notion of a global explanation, typically suggesting to aggregate masses of local explanations in the hope of ascertaining global properties, few provide frameworks that are either reliable or computationally tractable. Meanwhile, practitioners are requesting more efficient and interactive explainability tools. We take this opportunity to investigate existing global methods, with a focus on implementing and improving Actionable Recourse Summaries (AReS), the only known global counterfactual explanation framework for recourse.

1 INTRODUCTION

Counterfactual explanations (CEs) identify input perturbations that result in desired predictions from machine learning (ML) models (Verma et al., 2020; Karimi et al., 2020; Stepin et al., 2021). A key benefit of these explanations is their ability to offer recourse to affected individuals in certain scenarios (e.g., automated credit decisioning). Recent years have witnessed a surge of research therein, with a focus on identifying desirable properties of CEs, developing the methods to model those properties and understanding the weaknesses and vulnerabilities of the proposed methods (Barocas et al., 2020; Venkatasubramanian & Alfano, 2020; Pawelczyk et al., 2021; Slack et al., 2021).

Importantly, the research efforts so far have largely centred around local analysis, generating explanations for individual inputs. Such analyses can help vet model behaviour at an instance-level, though it is seldom obvious if the insights gained therein would generalise globally. For example, a local CE may suggest that a credit decisioning model is not biased against a protected attribute (e.g., gender, race), despite net biases existing across all inputs. A potential way to gain global insights is to aggregate local explanations, but given that the generation of CEs is generally computationally expensive, it is not evident that such an approach would scale well or even retain accuracy.

Rawal & Lakkaraju (2020) investigates this problem, proposing Actionable Recourse Summaries (AReS), a framework that constructs global counterfactual explanations (GCEs). This work reports our attempt to understand and implement AReS. Although a useful and flexible framework, there exist shortcomings that limit its real-world use. Specifically, we find that AReS is a) computationally expensive and b) sensitive to continuous features, due to a dependency on the cardinality of the set used in the selection GCEs. We propose amendments to the algorithm and demonstrate that these lead to significant performance improvements on two benchmarked financial datasets.

2 INVESTIGATIONS: BACKGROUND, MOTIVATION AND EXISTING METHODS

2.1 LOCAL COUNTERFACTUAL EXPLANATIONS

Wachter et al. (2018) is one of the earliest works introducing CEs in the context of understanding black-box ML models. Their approach defines CEs as points that are close to the query input, w.r.t. some distance metric, that result in a desired model prediction. This work inspired several follow-up works where researchers proposed desirable properties of CEs and presented approaches

to generate such CEs. For example, Mothilal et al. (2020) argued that generating diverse CEs is essential for recourse. Other approaches aim to generate plausible CEs by considering proximity to the data manifold (Poyiadzi et al., 2020; Van Looveren & Klaise, 2021; Kanamori et al., 2020) or by taking into account causal relations among input features (Mahajan et al., 2019). Actionability of recourse is another important desideratum, as some features may be non-actionable, and hence should be excluded from the resulting CEs (Ustun et al., 2019). In another direction, some works focused on generating CEs for specific model categories, such as tree-based models (Lucic et al., 2022; Tolomei et al., 2017; Parmentier & Vidal, 2021), or differentiable models (Mothilal et al., 2020). For a detailed survey on CEs, please refer to (Karimi et al., 2020; Verma et al., 2020).

2.2 Beyond Local Counterfactual Explanations: The Curse of Globality

Despite a growing desire from practitioners for global explanation methods that provide summaries of model behaviour (Lakkaraju et al., 2022), the struggles associated with summarising complex, high-dimensional models is yet to be comprehensively solved. Some manner of aggregations of local explanations has been suggested, though no compelling frameworks have been presented that a) are computationally tractable and b) return reliable GCEs. Lakkaraju et al. (2022) also indicates a desire for increased interactivity with explanation tools, alongside global summaries, but these desiderata cannot be paired until the efficiency issues associated with global methods are addressed.

Such works have been few and far between. Plumb et al. (2020) and Ley et al. (2022) have sought global translations which transform each input point within a group to another desired target group, in the context of low-dimensional spaces. Meanwhile, Becker et al. (2021) provides an original method for GCE search, though openly struggles with scalability. To the best of our knowledge, only the aforementioned AReS specifically focuses on finding GCEs in the context of recourse.

2.3 Beyond Individualized Recourse: Actionable Recourse Summaries (AReS)

Recent work (Rawal & Lakkaraju, 2020) proposes AReS, a comprehensive, model-agnostic framework for GCE generation. Building on the previously proposed two level decision sets (Lakkaraju et al., 2019), AReS adopts an original, interpretable structure, termed two level recourse sets.

A two level recourse set contains triples of the form Outer-If/Inner-If/Then conditions, pictured in Figure 1. A frequent itemset mining algorithm such as apriori (Agrawal & Srikant, 1994) is deployed to generate candidate sets of conditions (e.g., Sex = Male, $20 \leq$ Age < 30). These are combined to generate triples, with all valid triples[1] forming the ground set V. The candidate set of Outer-If conditions is termed $\mathcal{SD}$ (the subgroup descriptors), while $\mathcal{RL}$ denotes the candidate set used to select Inner-If or Then conditions. For apriori mining, the probability of an itemset in the data, or support threshold p, determines the size of $\mathcal{SD}$ and $\mathcal{RL}$, and consequently the size of V.

The subgroup descriptors $\mathcal{SD}$ can be set by the user to subgroups of interest, which is shown useful in assessing fairness via the disparate impact of recourses between subgroups. Otherwise, Rawal & Lakkaraju (2020) assign $\mathcal{SD}$ and $\mathcal{RL}$ to the same set generated by apriori. AReS deploys a non-monotone submodular maximization algorithm (Lee et al., 2009) that selects, from the ground set V, a final, smaller set of rules R. Interpretability constraints for the total number of triples ϵ_1, the maximum width of any Outer-If/Inner-If combination ϵ_2 and the number of unique subgroup descriptors ϵ_3 in R are applied throughout. As in AReS, we take $\epsilon_1, \epsilon_2, \epsilon_3 = 20, 7, 10$.

While a novel framework, with an easily interpretable structure, AReS can fall short on two fronts:

Computational Efficiency An extremely low p value is required to achieve high-performance, resulting in an impractically large ground set to optimise. Our work efficiently generates denser, higher-performing ground sets, unlocking the utility that practitioners have expressed desire for.

Continuous Features AReS proposes binning continuous features prior to generating frequent itemsets with apriori. However, we find that for models trained on continuous features, this approach struggles to trade speed with performance. Too few bins results in unrealistic recourses, but too many bins results in excessive computation time for apriori. We propose a modified ground set generation algorithm that demonstrates significant improvements on continuous data.

[1] A valid triple requires that the features in the Outer-If/Inner-If conditions do not match, and the features in the Inner-If/Then conditions match exactly with at least one change in feature value.

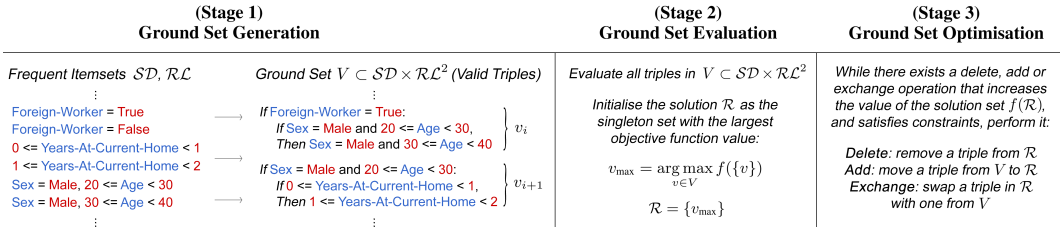

Figure 1: Workflow for our AReS implementation (without improvements). $\mathcal{SD}$ and $\mathcal{RL}$ are assigned to the same set generated by apriori. $\mathcal{SD} \times \mathcal{RL}^2$ is iterated over to compute all valid triples (Outer-If/Inner-If/Then conditions) for the ground set V (Stage 1). Each item in V is evaluated (Stage 2), and the optimisation procedure in Lee et al. (2009) is applied (Stage 3), returning the smaller two level recourse set, R. A more detailed version of the framework can be found in both Rawal & Lakkaraju (2020) and Appendix B.

3 Implementations: Actionable Recourse Summaries (AReS)

Our implementations include the original AReS framework, which follows the workflow demonstrated in Figure 1, as well as optimisations. The ground set V is defined as the set of triples from which the submodular maximisation algorithm (Lee et al., 2009) selects a two level recourse set $R \subset \mathcal{SD} \times \mathcal{RL}^2$, as stated in AReS.[2] We denote the dataset as $\mathcal{X}$, and the set of affected individuals with an unfavourable prediction from the model as $\mathcal{X}_{\text{aff}}$. The objective function $f(R)$ to be maximised is positive, comprising of incorrectness, coverage and cost. The metrics used in evaluating performance are recourse accuracy (the percentage of instances in $\mathcal{X}_{\text{aff}}$ that are provided with a successful recourse), denoted $acc(R)$, and average recourse cost (the average cost of those individuals in $\mathcal{X}_{\text{aff}}$ for whom prescribed recourses results in desired outcomes), denoted $cost(R)$. Owing to space constraints, we refer readers to Rawal & Lakkaraju (2020) and Appendix B for full details.

The overall global counterfactual search in AReS for a two level recourse set, can be partitioned into three stages, as detailed in Figure 1 and Table 1. We generate V, evaluate V, and optimise V (selecting a smaller, more interpretable set, R). We describe each of these stages in detail below, alongside our respective optimisations. R is evaluated in terms of recourse accuracy and average recourse cost, and it should be noted that, since recourse accuracy is monotonic (a new triple cannot invalidate a previous triple), $|R| \leq |V| \implies acc(R) \leq acc(V)$, providing us with an upper-bound.

3.1 Ground Set Generation (Stage 1)

The optimisation algorithm (Lee et al., 2009) requires a ground set V, which is generated by iterating through $\mathcal{SD} \times \mathcal{RL}^2$ and selecting valid triples. To generate larger $\mathcal{SD}$ or $\mathcal{RL}$, and thus larger V, a smaller apriori threshold p is used. With no user input, we assign $\mathcal{SD}$ and $\mathcal{RL}$ to the same set generated by apriori, giving $V \subset \mathcal{RL}^3$, a strict subset.[3] We denote $|\mathcal{RL}| = n \implies |V| < n^3$. Interpretability constraints that are independent of the optimisation, such as ϵ_2, are applied in this stage in $\mathcal{O}(n^2)$ and not $\mathcal{O}(n^3)$ time (see Appendix B.1). We introduce two methods to generate V. The first method computes an identical V more efficiently, while the second computes a different V.

Contribution 1a (*$\mathcal{RL}$-Reduction*) Iterating naively over $\mathcal{SD} \times \mathcal{RL}^2$ is wasteful, as many members of $\mathcal{RL}$ will never form valid "If-Then" conditions. We iterate instead over $\mathcal{RL}$ in $\mathcal{O}(n)$ time and compute feature combinations, before removing any items that contain a feature combination that only occurs once, yielding a new $\mathcal{RL}$ with size αn, where $0 \leq \alpha \leq 1$ (note that $\mathcal{SD} = \mathcal{RL}$ is left untouched). For instance, the item "Foreign-Worker = True, Sex = Male" has a feature combination of "Foreign-Worker, Sex"; if this only occurs once, it can be safely removed. For a given $\mathcal{RL}$, the ground set V is the same as the original method, yet $(1 - \alpha^2)n^3 - n$ iterations are saved.

Contribution 1b (*Then-Generation*, *q*) Instead of searching $\mathcal{SD} \times \mathcal{RL}^2$ for triples, we search $\mathcal{SD} \times \mathcal{RL}$ for If conditions, and deploy a separate method to generate Then conditions. Specifically,

[2] Although Rawal & Lakkaraju (2020) denote the solution to be a subset $R \subset \mathcal{SD} \times \mathcal{RL}$, this is mathematically impossible given that we require three conditions to form a valid triple (unless $\mathcal{RL}$ contains If/Then sets, which cannot be true if $\mathcal{SD} = \mathcal{RL}$, as AReS suggests). Correspondence with the authors confirms this.

[3] We are guaranteed to find invalid triples in $\mathcal{RL}^3$. For example, if the first element of $\mathcal{RL}$ is "Sex = Female", the first iteration generates the triple "If Sex = Female, If Sex = Female, Then Sex = Female", an invalid triple.

	(Stage 1) Ground Set Generation	(Stage 2) Ground Set Evaluation	(Stage 3) Ground Set Optimisation
AReS	n^3 Iterations Performed	Evaluates Full Ground Set	Searches Full Ground Set
Ours	$\alpha^2 n^3 + n$ or $n^2 m$ Iterations Performed	Evaluates and Shrinks Full or Partial Ground Set	Searches Shrunk and Sorted Ground Set

Table 1: A summary of our AReS enhancements w.r.t. each stage of the search. Definitions in Section 3.1.

for each valid element of $\mathcal{SD} \times \mathcal{RL}$, with index i, we compute its feature combination and filter the dataset by these features (also removing inputs that satisfy the initial If conditions), before applying apriori again, with threshold q, to generate a set of Then conditions, denoted $\mathcal{T}_i$. We can lower bound q as $1/|\mathcal{X}|$ (no observed itemset can have frequency < 1), and we find that varying q has little impact on speed but reduces performance (Appendix C). If $m = \max_i |\mathcal{T}_i|$ is the maximum size of any such $\mathcal{T}_i$, the number of iterations has an upper bound of $n^2 m$. The ground set generated differs from the original method and we observe significant improvements on continuous features.

3.2 Ground Set Evaluation (Stage 2)

The submodular maximisation (Lee et al., 2009) first evaluates the objective function f over all triples $v \in V$, before initialising the solution R as the singleton set $\{v\}$ with the maximum $f(\{v\})$. For large $|V|$, this evaluation becomes computationally costly (more-so does the subsequent ground set optimisation), and many triples are also redundant. However, we require large $|V|$ in order to find high-performing triples and achieve an acceptable upper bound[4] on the final set, $R \subseteq V$.

We take advantage of two empirical observations: the generation of a large ground set V is relatively cheap; and the recourse accuracy $acc(V)$ of the full ground set is approached far before the whole set has been evaluated. This allows us to efficiently shrink large ground sets to smaller ones with comparable recourse accuracy. For example, in 40 seconds, the apriori threshold $p = 0.22$ on the German Credit dataset produces a ground set with $|V| = 119708$. While $acc(V) = 84\%$ then takes 300 seconds to evaluate, 84% is converged to after only 5 seconds (Figure 2). The maximum value of a single triple is also seen to converge quickly. We can generate a large ground set, before only evaluating a small portion of this set to yield an equally high-performing yet denser ground set. Note that simply raising p to 0.323 and producing a smaller ground set of equal size does not yield 84% accuracy (instead, it yields 27%).

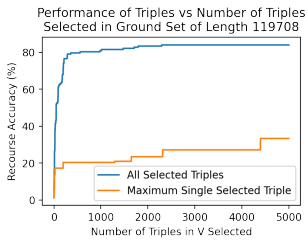

Figure 2: Redundancy in ground set V. German Credit, $p = 0.22$.

Contribution 2 (*V-Reduction*, r, r') We evaluate a fixed number of triples and form a new ground set in one of two ways: by adding each new triple, or by only adding triples that increase the recourse accuracy of the new ground set (i.e. vertical steps in Figure 2, blue). We denote these r and r' respectively. For example, $r' = 2000$ results in 2000 evaluations and less than 2000 triples added.

3.3 Ground Set Optimisation (Stage 3)

The bottleneck in the AReS framework is, however, the submodular maximisation in Lee et al. (2009), which takes the ground set V and returns a reduced set R that satisfies the interpretability constraints. The time taken is a function of the size $|V|$ of the ground set; we can thus achieve speedups by effectively further shrinking the ground set pre-optimisation. The submodular maximisation provides optimality guarantees. As such, we do not modify the algorithm itself.[5] Our ground set modifications instead provide the algorithm with a superior starting point and upper bound.

Contribution 3 (*V-Selection*, s) We propose to sort the (new) ground set by recourse accuracy (already computed), and select the s highest-performing triples. If $s = r$ or r', no sorting occurs.

[4]For instance, if $acc(V) = 25\%$, we cannot achieve $acc(R) > 25\%$; conversely, a ground set with $acc(V) = 80\%$ requires major evaluation and will also include many low-performing, redundant triples.

[5]Importantly, however, with knowledge of our upper bound $acc(R) \leq acc(V)$, optimisation can be terminated if this bound is approached. Such a bound can also be used to determine if Stage 3 is even initiated.

4 IMPROVEMENTS: EXPERIMENTAL RESULTS

We evaluate our methods on two benchmarked financial datasets: the German Credit dataset (Dua & Graff, 2019) classifies credit risk on people described by a set of attributes, consisting mostly of categorical features; the HELOC (Home Equity Line of Credit) dataset (FICO, 2018) includes anonymised credit applications made by real homeowners, and consists solely of continuous features. We train Deep Neural Networks (DNNs) with width 50 and depth 10 and 5 respectively on these datasets, with an 80% training split. Continuous features are binned into 10 equal intervals post-training (see Section 2.3 trade-off), and recourses are constructed on the training set.

We analyse the performance of AReS and our improvements cumulatively, at each stage of the workflow. For various input parameter combinations (p, r, r' and s), the final two level recourse sets returned in Stage 3 achieve significantly higher recourse accuracy within a time frame of 300 seconds (5 minutes), achieving accuracies for which AReS required 45 minutes on German Credit, and over 18 hours on HELOC. Further hyper-parameter details are located in Appendix C.

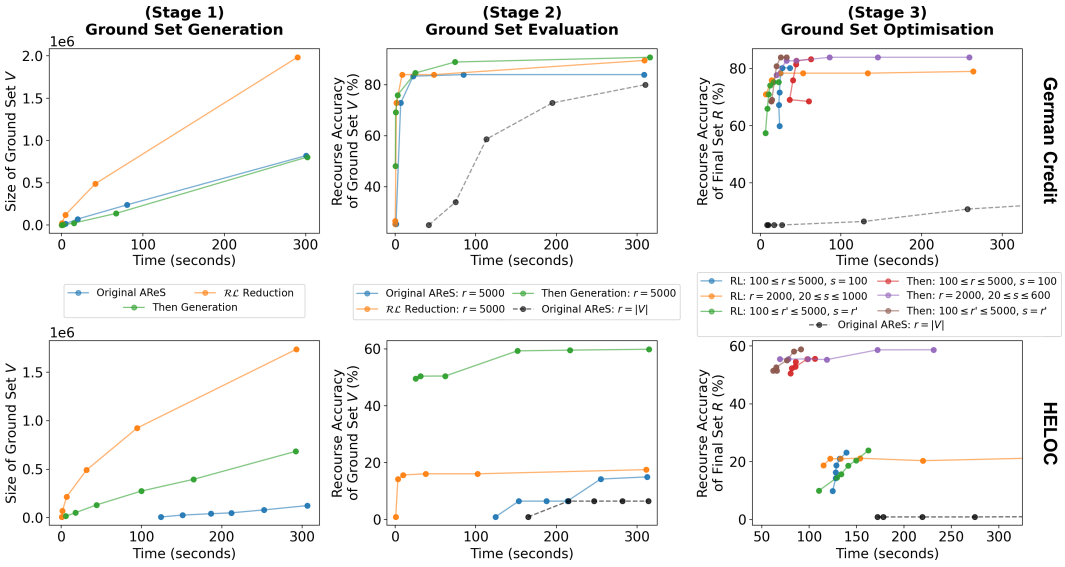

Figure 3: Computational Improvements. Top Row: German Credit. Bottom Row: HELOC. Left: Size of Ground Set V vs Time. Centre: Ground Set $acc(V)$ vs Time. Right: Final Set $acc(R)$ vs Time.

Takeaways In Stage 1, we demonstrate that *$\mathcal{RL}$-Reduction* is capable of generating an equivalent ground set V orders of magnitude faster than the original method. Our *Then-Generation* technique also constructs (different) ground sets rapidly. Stage 2 *V-Reduction* ($r = 5000$) performs significantly better than full evaluation, and *Then-Generation* erases many of the limitations surrounding continuous features. We finally observe vast speedups in Stage 3, owing to the construction of small yet high-performing ground sets: r, r' and s restrict the size of V yet retain a near-optimal $acc(V)$.

We note that the choice of $\mathcal{SD} = \mathcal{RL}$ affects performance (selecting a fixed $\mathcal{SD}$ would reduce the size of $|\mathcal{X}_{\text{aff}}|$ and V) though argue then that we are emphasising the scalability of our new approach.

5 CONCLUSION

This work studies the current state of global counterfactual explanations (GCEs), and addresses in detail the scalability/performance issues we find in the recently proposed AReS framework (Rawal & Lakkaraju, 2020). We investigate works on both global and local counterfactual explanations before implementing and improving AReS. With mounting desire from a practitioner viewpoint for access to fast, interactive explainability tools (Lakkaraju et al., 2022), it is crucial that such methods are not inefficient. We propose improvements to the AReS framework that speed up the generation of GCEs by orders of magnitude, also witnessing significant accuracy improvements on continuous data. Our hope is that this will inspire further research into the particularly under-studied area of GCEs, and prove useful as the development of explainability tools grows in the coming years.

Acknowledgments We thank the original authors Kaivalya Rawal and Himabindu Lakkaraju for their helpful discussion of the proposed AReS framework in Rawal & Lakkaraju (2020).

Disclaimer This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates ("JP Morgan"), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

REFERENCES

Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the International Conference on Very Large Data Bases*, VLDB '94, pp. 487–499, San Francisco, USA, September 1994. Morgan Kaufmann Publishers Inc. ISBN 1558601538. URL https://dl.acm.org/doi/10.5555/645920.672836.

Solon Barocas, Andrew D. Selbst, and Manish Raghavan. The Hidden Assumptions Behind Counterfactual Explanations and Principal Reasons. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, Barcelona, Spain, January 2020. doi: 10.1145/3351095.3372830. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3503019. arXiv: 1912.04930.

Maximilian Becker, Nadia Burkart, Pascal Birnstill, and Jürgen Beyerer. A Step Towards Global Counterfactual Explanations: Approximating the Feature Space Through Hierarchical Division and Graph Search. In *Advances in Artificial Intelligence and Machine Learning*, 2021. URL https://publikationen.bibliothek.kit.edu/1000139219.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2019. URL http://archive.ics.uci.edu/ml. Accessed: 2022-02-26.

FICO. Explainable Machine Learning Challenge, 2018. URL https://community.fico.com/s/explainable-machine-learning-challenge. Accessed: 2022-02-26.

Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2855–2862, Yokohama, Japan, July 2020. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-6-5. doi: 10.24963/ijcai.2020/395. URL https://www.ijcai.org/proceedings/2020/395.

Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A Survey of Algorithmic Recourse: Definitions, Formulations, Solutions, and Prospects. In *NeurIPS Workshop on ML Retrospectives, Surveys & Meta-Analyses (ML-RSA)*, Virtual-Only, December 2020. URL https://ml-retrospectives.github.io/neurips2020/camera_ready/8.pdf. arXiv: 2010.04050.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Faithful and Customizable Explanations of Black Box Models. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, pp. 131–138, Honolulu, USA, January 2019. Association for Computing Machinery. ISBN 9781450363242. doi: 10.1145/3306618.3314229. URL https://doi.org/10.1145/3306618.3314229.

Himabindu Lakkaraju, Dylan Slack, Yuxin Chen, Chenhao Tan, and Sameer Singh. Rethinking Explainability as a Dialogue: A Practitioner's Perspective. 2022. URL https://arxiv.org/abs/2202.01875. arXiv: 2202.01875.

Jon Lee, Vahab Mirrokni, Viswanath Nagarjan, and Maxim Sviridenko. Non-Monotone Submodular Maximization under Matroid and Knapsack Constraints. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, Maryland, USA, May 2009. URL https://dl.acm.org/doi/10.1145/1536414.1536459. arXiv: 0902.0353.

Dan Ley, Umang Bhatt, and Adrian Weller. Diverse, Global and Amortised Counterfactual Explanations for Uncertainty Estimates. In *AAAI Conference on Artificial Intelligence*, Virtual-Only, February 2022. URL http://arxiv.org/abs/2112.02646. arXiv: 2112.02646.

Ana Lucic, Harrie Oosterhuis, Hinda Haned, and Maarten de Rijke. FOCUS: Flexible Optimizable Counterfactual Explanations for Tree Ensembles. In *AAAI Conference on Artificial Intelligence*, Virtual-Only, February 2022. URL http://arxiv.org/abs/1911.12199. arXiv: 1911.12199.

Divyat Mahajan, Chenhao Tan, and Amit Sharma. Preserving Causal Constraints in Counterfactual Explanations for Machine Learning Classifiers. In *NeurIPS Workshop on Do the Right Thing: Machine Learning and Causal Inference for Improved Decision Making*, Vancouver, Canada, December 2019. URL http://arxiv.org/abs/1912.03277. arXiv: 1912.03277.

Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 607–617, Barcelona, Spain, January 2020. doi: 10.1145/3351095.3372850. URL https://dl.acm.org/doi/abs/10.1145/3351095.3372850. arXiv: 1905.07697.

Axel Parmentier and Thibaut Vidal. Optimal Counterfactual Explanations in Tree Ensembles. In *International Conference on Machine Learning*, pp. 8422–8431, Virtual-Only, July 2021. URL http://proceedings.mlr.press/v139/parmentier21a/parmentier21a.pdf. arXiv: 2106.06631.

Martin Pawelczyk, Sascha Bielawski, Johannes van den Heuvel, Tobias Richter, and Gjergji Kasneci. CARLA: A Python Library to Benchmark Algorithmic Recourse and Counterfactual Explanation Algorithms. In *Advances in Neural Information Processing Systems (Datasets & Benchmarks Track)*, August 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/b53b3a3d6ab90ce0268229151c9bde11-Paper-round1.pdf. arXiv: 2108.00783.

Gregory Plumb, Jonathan Terhorst, Sriram Sankararaman, and Ameet Talwalkar. Explaining Groups of Points in Low-Dimensional Representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7762–7771, Virtual-Only, July 2020. PMLR. URL http://proceedings.mlr.press/v119/plumb20a/plumb20a.pdf.

Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. FACE: Feasible and Actionable Counterfactual Explanations. pp. 344–350, New York, USA, February 2020. doi: 10.1145/3375627.3375850. URL https://dl.acm.org/doi/10.1145/3375627.3375850. arXiv: 1909.09369.

Kaivalya Rawal and Himabindu Lakkaraju. Beyond Individualized Recourse: Interpretable and Interactive Summaries of Actionable Recourses. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12187–12198, Virtual-Only, December 2020. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2020/file/8ee7730e97c67473a424ccfeff49ab20-Paper.pdf.

Dylan Slack, Anna Hilgard, Himabindu Lakkaraju, and Sameer Singh. Counterfactual Explanations Can Be Manipulated. In *Advances in Neural Information Processing Systems*, volume 34, Virtual-Only, December 2021. URL https://proceedings.neurips.cc/paper/2021/file/009c434cab57de48a31f6b669e7ba266-Paper.pdf.

Ilia Stepin, Jose M. Alonso, Alejandro Catala, and Martín Pereira-Fariña. A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence. In

IEEE Access, volume 9, pp. 11974–12001, 2021. doi: 10.1109/ACCESS.2021.3051315. URL https://ieeexplore.ieee.org/document/9321372.

Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. Interpretable Predictions of Tree-based Ensembles via Actionable Feature Tweaking. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 465–474, Novia Scotia, Canada, August 2017. doi: 10.1145/3097983.3098039. URL https://dl.acm.org/doi/10.1145/3097983.3098039. arXiv: 1706.06691.

Berk Ustun, Alexander Spangher, and Yang Liu. Actionable Recourse in Linear Classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 10–19, Atlanta, USA, January 2019. doi: 10.1145/3287560.3287566. URL https://dl.acm.org/doi/10.1145/3287560.3287566. arXiv: 1809.06514.

Arnaud Van Looveren and Janis Klaise. Interpretable Counterfactual Explanations Guided by Prototypes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 650–665, Bilbao, Spain, September 2021. Springer. URL https://2021.ecmlpkdd.org/wp-content/uploads/2021/07/sub_352.pdf.

Suresh Venkatasubramanian and Mark Alfano. The Philosophical Basis of Algorithmic Recourse. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 284–293, Barcelona, Spain, January 2020. ACM. ISBN 978-1-4503-6936-7. doi: 10.1145/3351095.3372876. URL https://dl.acm.org/doi/10.1145/3351095.3372876.

Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual Explanations for Machine Learning: A Review. *arXiv:2010.10596 [cs, stat]*, October 2020. URL http://arxiv.org/abs/2010.10596. arXiv: 2010.10596.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. In *Harvard Journal of Law & Technology*, 2018. doi: 10.2139/ssrn.3063289. URL https://www.ssrn.com/abstract=3063289. arXiv: 1711.00399.

APPENDIX

This appendix is formatted as follows.

1. We discuss the *Datasets and Models* used in our work in Appendix A.
2. We discuss the *Implementation Details* of our work in Appendix B.
3. We list the *Experimental Details* of our work and analyse *Further Results* in Appendix C.

A DATASETS AND MODELS

Two benchmarked financial datasets are employed in our experiments, both of which are a) binary classification and b) publicly available. Details are provided below and in Table 2. Our experiments include just one type of model, Deep Neural Networks, which we also describe below and in Table 3.

A.1 DATASETS

The **German Credit** dataset (Dua & Graff, 2019) can be obtained from and is described in detail at the following URL: `https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)`. We augment input dimensions by performing a one-hot encoding over necessary variables (Sex, Foreign-Worker, etc). The documentation for this dataset also details a cost matrix, where false positive predictions induce a higher cost than false negative predictions, but we ignore this in model training. Note that this is distinct from the also common Default Credit dataset.

The **HELOC** (Home Equity Line of Credit) dataset (FICO, 2018) can upon request be obtained from and is described in detail at the following URL: `https://community.fico.com/s/explainable-machine-learning-challenge`. Missing values in the dataset are represented with negative integers; we drop inputs where all feature values are missing, and replace the remaining missing values in the dataset with the median value for that feature. We also drop any duplicate inputs in the dataset. Notably, the majority of features are monotonically increasing/decreasing.

Name	Categorical	Continuous	Input Dim.	No. Train	No. Test
German Credit	17	3	71*	800	200
HELOC	0	23	23	7896*	1975*

*Denotes values post-processing (one-hot encoding inputs, dropping inputs).

Table 2: Summary of the datasets used in our experiments. Although German Credit includes continuous features, we find that they have limited effect on the model both during training and in the resulting explanations.

A.2 MODELS

We train Deep Neural Networks (DNNs) with width 50 and depth 10 and 5 respectively on these datasets, with an 80% to 20% train to test split. Layers include dropout, bias and ReLU activation functions. We map the final layer to the output using softmax, and use Adam to optimise a cross-entropy loss function in the standard manner. Table 3 details various model parameters/behaviours.

Name	Width	Depth	Dropout	Train Acc.	Test Acc	$\lvert\mathcal{X}_{\text{aff}}\rvert$	$\lvert\mathcal{X}_{\text{aff}}\rvert/\lvert\mathcal{X}\rvert$
German Credit	50	10	0.3	82%	79%	162	20%
HELOC	50	5	0.5	74%	73%	3882	49%

Table 3: Summary of the DNNs used in our experiments. The proportion of negative labels in the dataset were 30% and 53% for German Credit and HELOC respectively; our models roughly follow suit (20% and 49%).

Of note is the scalability of AReS, which struggled with HELOC, a dataset that contained significantly more points to explain ($|\mathcal{X}_{\text{aff}}|$) than German Credit. Additionally, the proportion of points with positive predictions (80% for German Credit and 51% for HELOC) influences the ease with which AReS finds recourses. For stringent models (those which scarcely predict positively), it would make

sense that the vast majority of frequent itemsets generated by apriori are representative of feature value combinations that exist in the inputs with negative predictions, and we might therefore expect to need to generate an enormous number of triples before we can identify successful recourses.

B IMPLEMENTATION DETAILS

We use this Appendix to provide further details regarding the implementation of each stage of the AReS workflow. Our implementation of AReS, without improvements, does in fact differ slightly from that proposed in Rawal & Lakkaraju (2020), and as such we will justify our changes herein. We of course acknowledge that this implementation is far from the most efficient possible, though hope that the patterns and improvements we have identified can aid further development of not only this framework, but others in the global counterfactual explanations space also.

B.1 GROUND SET GENERATION (STAGE 1)

As stated in the main text, our implementation applies constraints during ground set generation where possible. AReS includes interpretability constraints for the total number of triples ϵ_1, the maximum width of any Outer-If/Inner-If combination ϵ_2 and the number of unique subgroup descriptors ϵ_3 in R. As in AReS, we take $\epsilon_1, \epsilon_2, \epsilon_3 = 20, 7, 10$. In our implementation, we expedite the ϵ_2 width constraint to the ground set generation process by constraining apriori to only return frequent itemsets that have length $\epsilon_2 - 1$ or less, since those already with width ϵ_2 cannot then be further combined with another itemset to form Outer-If/Inner-If conditions. If the width constraint is not violated for the If conditions, the resulting triple will automatically satisfy the constraint.

The implication of this is that we can apply the constraint in Stage 1 while we generate the ground set (in the first two levels of the iteration through $\mathcal{RL}^3$). This avoids applying the width constraint mid-optimisation in Stage 3, reducing the time complexity of the operation from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$. It also reduces the number of constraints used in Lee et al. (2009), speeding up Stage 3. Since it makes sense that triples which violate the maximum width condition should not be generated in Stage 1, we assume that a similar approach is deployed (though not stated) in Rawal & Lakkaraju (2020).

Then-Generation A lower bound for the threshold q used in Then-Generation was also alluded to in the main text. In fact, there always exists a lower bound when mining frequent itemsets, such as in apriori, since no observed itemset can occur less than once. Thus, setting $q < 1/|\mathcal{X}|$ would be redundant. This allows us to analyse the full effect of $1/|\mathcal{X} \leq q \leq 1$ in Appendix C.

B.2 GROUND SET EVALUATION (STAGE 2)

Our improvement (Contribution 2) evaluates the objective function f (see Section B.3) over a fixed number of triples in V (recall that AReS evaluates the entirety of V). As we've demonstrated empirically, albeit on the two datasets tried in this investigation, evaluating the entire ground set is wasteful, given that performance of the first r elements of V saturates quickly, and more so if one considers that Stage 3 must then perform submodular maximisation over a space potentially hundreds of times as large, and that Lee et al. (2009) only guarantees polynomial time.

However, there is a distinction between evaluating the objective function f and evaluating the acc and $cost$ terms used in evaluation. Fortunately, no extra major computation is required to evaluate the acc and $cost$ terms, since the objective function f returns model predictions and costs, and although the two processes differ, they can be carried out efficiently in tandem. This is promising, as not only does our method allow us to terminate evaluation once saturation has been reached, but it also provides us with the upper bound $acc(R) \leq acc(V)$. In many of our experiments, this upper bound is actually reached in Stage 3 far before the algorithm has finished, presenting us with a straightforward opportunity for early termination of the algorithm. This could further save time dramatically, though was not included in our experiments.

B.3 GROUND SET OPTIMISATION (STAGE 3)

We introduce two key modifications to Stage 3 of our implementation. The first is to the objective function, the second is to the submodular maximisation in Lee et al. (2009).

Objective Function The objective function $f(R)$ in Rawal & Lakkaraju (2020) is designed to be non-normal, non-negative, non-monotone and submodular, and to have constraints that are matroids. These conditions are required for the submodular maximisation in Lee et al. (2009) to have a formal guarantee of convergence. This results in four terms in $f(R)$: *incorrectrecourse*, *cover*, *featurecost*, *featurechange*. Bar the *cover* term, all of these are subtracted from $f(R)$ (i.e., maximising correct recourse by maximising the negative of *incorrectrecourse*). Such an objective function with three adjustable hyperparameters can be very difficult to tune. For that reason, we also trial in our experiments an objective that consists very simply of $acc(R) - \lambda \times cost(R)$, which we maximise. We argue that the formal guarantees of convergence (polynomial time) are largely a misdirection of efforts in the original method. Polynomial time is not particularly helpful when the size of ground sets required for certain datasets/models is huge, and thus we instead focus on reducing the size of the ground set while retaining quality before the submodular maximisation (Lee et al., 2009) is applied.

Submodular Maximisation The algorithm states that, for k constraints, you can exchange up to k elements from your solution set R alongside the addition of one element from V. Stated also is that the optimisation should be repeated $k+1$ times, before the best solution for R is then chosen. In reality, both of these induce high computational costs. Trivially, for the latter, ignoring the maximum width constraint (Appendix B.1) and taking $k + 1 = 3$, we will mostly increase the time taken by AReS three-fold. Having observed that both of these steps do not improve the performance of AReS significantly in our experiments, we omit them from the original and improved implementations.

C Experimental Details and Further Results

We use the training data from each dataset to learn recourses in our experiments (future work could analyse the effectiveness of such rules on unseen test data). Since AReS struggles to achieve sufficient recourse accuracy within reasonable time-frames for our datasets and models, we set the hyperparameters for *featurecost* and *featurechange*, or λ, to 0, also finding that the average cost of recourses were low and did not vary a large amount, justifying the decision to target correctness. The remaining hyperparameters used in the Figure 3 experiments (Section 4) are as detailed in Table 4.

Recall also that we have bounded the range of the apriori threshold q used in *Then-Generation* to $1/|\mathcal{X}| \leq q \leq 1$ (Section 3.1 and Appendix B.1). Figure 4 demonstrates that for $q > 1/|\mathcal{X}|$, we slightly reduce the time taken by the algorithm, at the expense of a much larger drop in performance. Observe that the red and brown lines (where p is held constant and q is varied) converge to the green and purple lines (where $q = 1/|\mathcal{X}|$ and p is varied) respectively. The brown and purple plots also indicate that combining our two improvements *$\mathcal{RL}$-Reduction* and *Then-Generation* performs sub-optimally. We thus decide to evaluate these improvements separately with a fixed $q = 1/|\mathcal{X}|$ threshold used in the *Then-Generation* method.

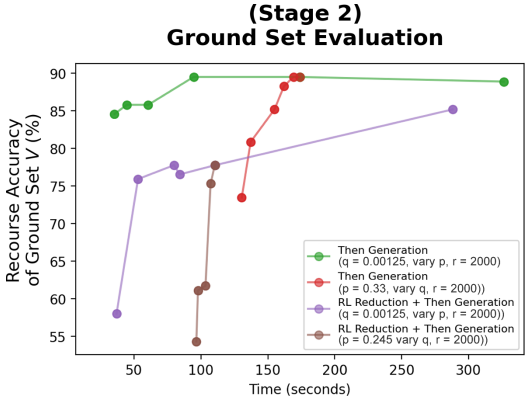

Figure 4: Effect of apriori threshold q in the proposed *Then-Generation* method (German Credit).

	Stage 1	Stage 2	Stage 3
German Credit	**OG**: $0.169 \leq p \leq 0.390 \longrightarrow$ **RL**: $0.39 \leq p \leq 0.149 \longrightarrow$ **Then**: $0.9 \leq p \leq 0.303 \longrightarrow$ $q = 0.00125$	**OG**: $r = 5000$ **RL**: $r = 5000$ **Then**: $r = 5000, q = 0.00125$ **OG**: $0.316 \leq p \leq 0.26, r = \lvert V \rvert$	**OG**: $0.39 \leq p \leq 0.305, r = \lvert V \rvert$ **RL**: $p = 0.245$ **Then**: $p = 0.48$, $q = 0.00125$
HELOC	**OG**: $0.325 \leq p \leq 0.285 \longrightarrow$ **RL**: $0.325 \leq p \leq 0.203 \longrightarrow$ **Then**: $0.75 \leq p \leq 0.563 \longrightarrow$ $q = 0.000127$	**OG**: $r = 5000$ **RL**: $r = 5000$ **Then**: $r = 5000, q = 0.000127$ **OG**: $0.325 \leq p \leq 0.3, r = \lvert V \rvert$	**OG**: $0.324 \leq p \leq 0.318, r = \lvert V \rvert$ **RL**: $p = 0.245$ **Then**: $p = 0.48$, $q = 0.000127$

Table 4: The keys **OG** (*Original AReS*), **RL** (*$\mathcal{RL}$-Reduction*) and **Then** (*Then-Generation)* refer to the generation process of the ground set, as per Section 3.1. Arrows indicate values carried from one stage to the next. Apriori thresholds p and q are listed. Remaining parameters r, r' and s are listed in the original Figure 3 plots.