Endogenous Macrodynamics in Algorithmic Recourse

***Abstract*—Existing work on Counterfactual Explanations (CE) and Algorithmic Recourse (AR) has largely been limited to the static setting and focused on single individuals: given some estimated model the goal is to find valid counterfactuals for individual instance that fulfill various desiderata. The ability of such counterfactuals to handle dynamics like data and model drift remains a largely unexplored research challenge at this point. There has also been surprisingly little work on the related question of how the actual implementation of recourse by one individual may affect other individuals. Through this work we aim to close that gap by systematizing and extending existing knowledge. We first show that many of the existing methodologies can be collectively described by a generalized framework. We then argue that the existing framework fails to account for a hidden external cost of recourse, that only reveals itself when studying the endogenous dynamics of recourse at the group level. Through simulation experiments involving various popular counterfactual generators and several benchmark datasets, we generate a total XX million Counterfactual Explanations and study the resulting domain and model shifts. We find that the induced shifts are substantial enough to likely impede the applicability Algorithmic Recourse in practice. Fortunately, we find various potential mitigation strategies that can be used in combination with existing approaches. Our simulation framework for studying recourse dynamics is fast and open-sourced.**

I. Introduction

Recent advances in Artificial Intelligence (AI) have propelled its adoption in scientific domains outside of Computer Science including Healthcare, Bioinformatics, Genetics and the Social Sciences. While this has in many cases brought benefits in terms of efficiency, state-of-the-art models like Deep Neural Networks (DNN) have also given rise a new type of principal-agent problem in the context of data-driven decision-making. It involves a group of **principals** - i.e. human stakeholders - that fail to understand the behaviour of their **agent** - i.e. the model used for automated decision-making [1].

Models or algorithms that fall into this category are typically referred to **black-box** models. Despite their shortcomings, black-box models have grown in popularity in recent years and have at times created undesirable societal outcomes [2]. The scientific community has tackled this issue from two different angles: while some have appealed for a strict focus on inherently interpretable models [3], others have investigated different ways to explain the behaviour of black-box models. These two sub-domains can be broadly referred to as **interpretable AI** and **explainable AI** (XAI), respectively.

Among the approaches to XAI that have recently grown in popularity are **Counterfactual Explanations** (CE). They explain how inputs into a model need to change for it to produce different outputs. Counterfactual Explanations that involve realistic and actionable changes can be used for the purpose of **Algorithmic Recourse** (AR) to help individuals who face adverse outcomes. An example relevant to the Social Sciences is consumer credit: in this context, AR can be used to guide individuals in improving their creditworthiness, should they have previously been denied access to credit based on an automated decision-making system. A meaningful recourse recommendation for a denied applicant could be: *"If your net savings rate had been 10% of your monthly income instead of the actual 8%, your application would have been successful. See if you can temporarily cut down on consumption."* In the remainder of this paper we will use both terminologies - recourse and counterfactual - interchangeably to refer to situations where counterfactuals are generated with the intent to provide individual recourse.

Existing work in this field has largely worked in a static setting: various approaches have been proposed to generate counterfactuals for a given individual that is subject to some pre-trained model. More recent work has compared different approaches within this static setting [4]. In this work, we go one step further and ask ourselves: what happens if recourse is provided and implemented repeatedly? What types of dynamics are introduced and how do different counterfactual generators compare in this context?

Research on Algorithmic Recourse has also so far typically addressed the issue from the perspective of one single individual and has indeed been referred to as **individual recourse** in some places. Arguably though, most real-world applications that warrant Algorithmic Recourse involve potentially large groups of individuals typically competing for scarce resources. Our work demonstrates that in such scenarios, choices made by or for one single individual are likely to affect the broader collective of individuals in ways that current approaches to AR fail to account for. More specifically, we argue that a strict focus on minimizing the private costs faced by individuals may be too narrow an objective.

Figure 1 illustrates this idea for a binary problem involving a probabilistic classifier and the counterfactual generator proposed by [5]: the implementation of AR for a subset of individuals leads to a domain shift (b), which in turn triggers a model shift (c). As this game of

implementing AR and updating the classifier is repeated, the decision boundary moves away from training samples that were originally in the target class (d). We refer to these types of dynamics as **endogenous** because they are induced by the implementation of recourse itself. The term **macrodynamics** is borrowed from the economics literature and used to describe processes involving whole groups or societies.

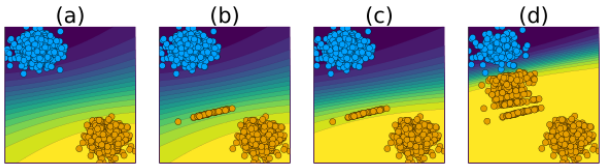

Figure 1: Dynamics in Algorithmic Recourse: we have a simple Bayesian model trained for binary classification (a); the implementation of AR for a random subset of individuals leads to a domain shift (b); as the classifier is retrained we observe a model shift (c); as this process is repeated, the decision boundary moves away from the target class (d).

We think that these types of endogenous dynamics may be problematic and warrant our attention. Firstly, model shifts may inadvertently change classification outcomes for individuals who never received and implemented recourse. Secondly and relatedly, we observe in Figure 1 that as the decision boundary moves in the direction of the non-target class, counterfactual paths become shorter: in the consumer credit example, individuals that previously would have been denied credit based on their input features are suddenly considered as creditworthy. Average default risk across all borrowers can therefore be expected to increase. Conversely, lenders that anticipate such dynamics may choose to refrain from offering recourse (and hence credit) to more than just a tiny share of individuals. In that latter and perhaps more likely scenario, the probability of being offered recourse decreases with every individual that implements recourse: in other words, the actions of first-movers exert a negative externality on future would-be borrowers.

To the best of our knowledge this is the first work investigating endogenous macrodynamics in AR. Our contributions to the state of knowledge are as follows: firstly, we posit a compelling argument that calls for a novel perspective on Algorithmic Recourse extending our focus from single individuals to groups. Secondly, we introduce an experimental framework extending previous work by [6], which enables us to study macrodynamics of Algorithmic Recourse through simulations that can be fully parallelized. Thirdly, we use this framework to provide a first in-depth analysis of endogenous recourse dynamics induced by various popular counterfactual generators including [5], [7], [8], [9] and [10]. To this end we propose a number of novel evaluation metrics that can be used to quantify and benchmark the macrodynamics introduced by the different generators. Finally, we also discuss what drives endogenous dynamics and propose strategies to mitigate them.

The remainder of the paper is structured as follows: Section II places our work in the broader context of related literature. Section III posits a generalized framework for gradient-based Algorithmic Recourse and introduces the notion of hidden external costs. Section IV sets out our experimental framework for modeling endogenous macrodynamics in AR. Section V presents our experimental results for synthetic and real-world datasets. We also present evidence for the effectiveness of various proposed mitigation strategies in that section. Our findings are then discussed in the broader context of the literature in Section VII. We also point to some of the limitations of our work as well as avenues for future research in Section VIII. Finally, Section IX concludes.

II. Background

In this Section we provide a review of the relevant literature. First, Section II-A discusses the existing research within the domain of Counterfactual Explanations and Algorithmic Recourse. Then, Section II-B presents some of the previous work on the measurement of dataset and model shifts.

A. Algorithmic Recourse

A framework for Counterfactual Explanations was first proposed in 2017 by [5] and has served as the baseline for most methodologies that have been proposed since then. Let $M : \mathcal{X} \mapsto \mathcal{Y}$ denote some pre-trained model that maps from inputs $X \in \mathcal{X}$ to outputs $Y \in \mathcal{Y}$. Then we are interested in minimizing the cost[1] $H = h(x')$ incurred by individual x when moving to a counterfactual state x' such that the predicted outcome $M(x')$ corresponds to some target outcome y^*:

$$\min_{x' \in x} h(x') \quad \text{s. t.} \quad M(x') = t \tag{1}$$

For implementation purposes, Equation 1 is typically approximated through regularization:

$$x' = \arg\min_{x'} \ell(M(x'), y^*) + \lambda h(x') \tag{2}$$

In the baseline work [5], the cost function $h : \mathcal{X} \mapsto \mathbb{R}$ is proxied by some distance metric based on the simple intuition that perturbations of x are costly to the individual. For models that are differentiable and produce smooth predictions, Equation 2 can be solved through gradient descent. This summarizes the approach followed in [5]

[1] Equivalently, others have referred to this quantity as *complexity* or simply *distance*

which we shall refer to simply as **Wachter** - the name of the first author - in the remainder of this paper.

Many approaches for the generation of Algorithmic Recourse have been described in the literature since 2017. An October 2020 survey by Karimi et al. layed out 60 algorithms that have been proposed since 2014 [11]. Another survey published around the same time by Verma et al. described 29 algorithms [12]. Different approaches vary primarily in terms of the objective functions they impose, how they optimize said objective (from brute force through gradient-based approaches to graph traversal algorithms), and how the ensure that certain requirements for CE are met. Regarding the latter, the literature has produced an extensive list of desiderata each addressing different needs. To name but a few, we are interested in generating counterfactuals that close [5], actionable [13], realistic [7], sparse, diverse [9] and if possible causally founded [14].

Efforts so far have largely been directed at improving the quality of Counterfactual Explanations within a static context: given some pre-trained classifier $M : \mathcal{X} \mapsto \mathcal{Y}$, we are interested in generating one or multiple meaningful Counterfactual Explanations for some individual characterized by x. The ability of Counterfactual Explanations to handle dynamics like data and model shifts remains a largely unexplored research challenge at this point [12]. We have been able to identify only one recent work that considers the implications of **exogenous** domain and model shifts in the context of AR [15]. Exogenous shifts are strictly of external origin. For example, they might stem from data correction, temporal shifts or geospatial changes [15]. The authors of [15] propose ROAR - a framework for Algorithmic Recourse that evidently improves robustness to such exogenous shifts.

As mentioned earlier, research has so far also generally focused on generating counterfactuals for single individuals or instances. We have been able to identify only one existing work that investigates black-box model behavior towards a group of individuals [16]. The authors propose an optimization framework that generates collective counterfactuals. We provide a motivation for doing so from the perspective of endogenous macrodynamics of Algorithmic Recourse.

B. Domain and Model Shifts

Much attention has been paid to the detection of dataset shifts – situations where the distribution of data changes over time. Rabanser et al. suggest a framework to detect data drift from a minimal number of samples through the application of two-sample tests [17]. This task is a generalization of the anomaly detection problem for large datasets, which aims to answer the question if two sets of samples could have been generated from the same probability distribution. Numerous approaches to anomaly detection have been summarized [18]. Another well-established research topic is that of concept drift – situations where external variables influence the patterns between the input and the output of a model [19]. For instance, Gama et al. offer a review of the adaptive learning techniques which can handle concept drift [20]. Less previous work is available on the related topic of model drift - changes in model performance over time. Nelson et al. review how resistant different machine learning models are to the model drift [21]. Ackerman et al. offer a method to detect changes in model performance when ground truth is not available [22].

In the context of Algorithmic Recourse, domain and model shifts were first brought up by the authors behind ROAR [15]. In their work they refer to model shifts as simply any perturbation Δ to the parameters of the model in question: M. While this also sets the baseline for our analysis here, it is worth noting that in [15] these perturbations are mechanically introduced. In contrast we are interested in quantifying model shifts that arise endogenously as part of a dynamic recourse process. In addition to quantifying the magnitude of shifts Δ, we aim to also analyse the characteristics of changes to the model, such as the position of the decision boundary and the overall decisiveness of the model. We have not been able to identify previous work on this topic.

C. Benchmarking Counterfactual Generators

Despite the large and growing number of approaches to counterfactual search, there have been surprisingly few benchmark studies that compare different methodologies. This may be partially due to limited software availability in this space. Recent work has started to address this gap: firstly, [23] run a large benchmarking study using different algorithmic aproaches and numerous tabular datasets; secondly, [4] introduce a Python framework - CARLA - that can be used to apply and benchmark different methodologies; finally, `CounterfactualExplanations.jl` [6] provides an extensible, fast and language-agnostic implementation in Julia. Since the experiments presented here involve extensive simulations, we have relied on and extended the Julia implementation due to the associated performance benefits. In particular, we have built a framework on top of `CounterfactualExplanations.jl` that extends the functionality from static benchmarks to simulation experiments: `AlgorithmicRecourseDynamics.jl`[2]. The core concepts implemented in that package reflect what is presented in section Section IV of this paper.

III. Gradient-Based Recourse Revisited

In this section we first set out a generalized framework for gradient-based counterfactual search in Section III-A that encapsulates the various individual recourse methods we have chosen to use in our experiments. We then introduce the notion of a hidden external cost in algorithmic recourse

[2]The package is available from …

and extend the existing framework to explicitly address this cost in the counterfactual search objective.

A. From individual recourse …

We have chosen to focus on gradient-based counterfactual search for two reasons: firstly, they can be seen as direct descendants of our baseline method - Wachter; secondly, gradient-based search is particularly well-suited for differentiable black-box models like deep neural networks, which we focus on in this work. In particular, we include include the following generators in our simulation experiments below: **REVISE** [8], **CLUE** [10], **DiCE** [9] and a greedy approach that relies on probabilistic models [7]. Our motivation for including these different generators in our analysis, is that they all offer slightly different approaches to generate meaningful counterfactuals for differentiable black-box models. We hypothesize that generating more **meaningful** counterfactuals should mitigate the endogenous dynamics illustrated in Figure 1 in Section I. This intuition stems from the underlying idea that more meaningful counterfactuals are generated by the same or at least a very similar data generating process as the training data. All else equal, counterfactuals that fulfill this basic requirement should be less prone to trigger domain and model shifts.

As we will see next, all of them can be described by the following generalized form of Equation 2:

$$\mathbf{s}' = \arg\min_{\mathbf{s}' \in \mathcal{S}} \left\{ \sum_{k=1}^{K} \ell(M(f(s_k')), y^*) + \lambda h(f(s_k')) \right\} \tag{3}$$

Here $\mathbf{s}' = \{s_k'\}_K$ is the stacked K-dimensional array of counterfactual states and $f : \mathcal{S} \mapsto \mathcal{X}$ maps from the counterfactual state space to the feature space. In Wachter, the state space is the feature space: f is just the identity function and the number of counterfactuals K is equal to one. Both REVISE and CLUE search counterfactuals in some latent embedding $S \subset \mathcal{S}$ instead of the feature space directly. The latent embedding is learned by a separate generative model that is tasked with learning the data generating process (DGP) of X. In this case f in Equation 3 corresponds to the decoder part of the generative model, in other words the function that maps back from the latent embedding to the feature space. Provided the generative model is well-specified, traversing the latent embedding typically results in realistic and plausible counterfactuals, because they are implicitly generated by the (learned) DGP [8].

CLUE distinguishes itself from REVISE and other counterfactual generators in that it aims to minimize the predictive uncertainty of the model in question, M. To quantify predictive uncertainty the authors rely on entropy estimates for probabilistic models. The approach proposed by [7], which we shall refer to as **Greedy**, also works with the subclass of models $\tilde{\mathcal{M}} \subset \mathcal{M}$ that can produce predictive uncertainty estimates. The authors show that in this setting the cost function $h(\cdot)$ in Equation 3 is redundant and meaningful counterfactuals can be generated in a fast and efficient manner through a modified Jacobian-based Saliency Map Attack (JSMA). Schut et al. [7] also show that by maximizing the predicted probability of x' being assigned to target class y^* we also implicitly minimize predictive entropy - as in CLUE. In that sense, CLUE can be seen as equivalent to REVISE in the Bayesian context and we shall therefore refer to both approaches collectively as **Latent Space** generators[3].

Finally, DiCE [9] distinguishes itself from all other generators considered here in that it aims to generate a diverse set of $K > 1$ counterfactuals. Wachter et al. [5] show that diverse outcomes can in principal be achieved simply rerunning counterfactual search multiple times using stochastic gradient descent (or by randomly initializing the counterfactual). In [9] diversity is explicitly proxied via Determinantal Point Processes (DDP): the authors simply introduce DDP as a component of the cost function $h(\mathbf{s}')$ and therebt produce counterfactuals $s_1, ..., s_K$ that look as different from each other as possible. The implementation of DiCE in the our library of choice - `CounterfactualExplanations.jl` - uses that exact approach. It is worth noting that for $k = 1$, DiCE reduces to Wachter since the DDP is constant and therefore does not affect the objective function Equation 3.

B. … towards collective recourse

All of the different approaches introduced above tackle the problem of Algorithmic Recourse from the perspective of one single individual[4]. To explicitly address the issue that individual recourse may affect the outcome and prospect of other individuals, we propose to extend Equation 3 as follows:

$$\begin{aligned} \mathbf{s}' = \arg\min_{\mathbf{s}' \in \mathcal{S}} \sum_{k=1}^{K} \ell(M(f(s_k')), y^*) \\ + \lambda_1 h(f(s_k')) + \lambda_2 g(f(s_k')) \end{aligned} \tag{4}$$

Here $h(f(s_k'))$ denotes the proxy for private costs faced by the individual as before and λ_1 governs to what extent that private cost ought to be penalized. The newly introduced term $g(f(s_k'))$ is meant to capture and address external costs incurred by the collective of individuals in response to changes in $\mathbf{s}'$. The underlying concept of private and external costs is borrowed from Economics

[3] In fact, there are a number of other recently proposed approaches to counterfactual search that also broadly fall in this same category. They largely differ with respect to the chosen generative model: for example, the generator proposed by [24] relies on normalizing flows.

[4] DiCE recognizes that different individuals may have different objective functions, but it does not address the interdependencies between different individuals.

and well-established in that field: when the decisions or actions by some individual market participant generate external costs, then the market is said to suffer from negative externalities and considered inefficient [25]. We think that this concept describes the endogenous dynamics of algorithmic recourse oberserved here very well. As with individual recourse, the exact choice of $g(\cdot)$ is not obvious, nor do we intend to provide a definite answer in this work, if such even exists. That being said, we do propose a few potential mitigation strategies in Section VI-B.

IV. Modeling Endogenous Macrodynamics in Algorithmic Recourse

In the following we describe the framework we propose for modeling and analysing endogenous macrodynamics in Algorithmic Recourse. We first describe the basic simulations that were generated to produce the findings in this work and also constitute the core of `AlgorithmicRecourseDynamics.jl` - the Julia package we introduced earlier. The remainder of this section then introduces various evaluation metrics that can be used to benchmark different counterfactual generators with respect to how they perform in the dynamic setting.

A. Simulations

The dynamics illustrated in Figure 1 in Section I were generated through a simple experiment that aims to simulate the process of Algorithmic Recourse in practice. We begin in the static setting at time time $t = 0$: given some classifier M that was pre-trained on data $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1$ we generate recourse for a random batch of B individuals in the non-target class ($\mathcal{D}_0$). Note that we focus our attention on classification problems, since classification poses the most common practical use-case for Algorithmic Recourse. In order to simulate the dynamic process, we suppose that the model M is retrained following the actual implementation of recourse in time $t = 0$. Following the update to the model, we assume that at time $t = 1$ recourse is generated for yet another random subset of individuals in the non-target class. This process is repeated for a number of time periods T. To get a clean read on endogenous dynamics we keep the total population of samples closed: we allow existing samples to move from factual to counterfactual states, but do not allow any entirely new samples to enter the population. The experimental setup is summarized in Algorithm 1

A noteworthy practical consideration is the choice of T and B. The higher these values, the more factual instances undergo recourse throughout the entire experiment. Of course, this is likely to lead to more pronounced domain and model shifts by time T. At the same time, it is generally improbable that a very large part of the population would request an explanation of the algorithm's decisions. In our experiments, we choose the values such that $T \cdot B$ corresponds to the application of recourse on $\approx 50\%$ of the negative instances from the initial dataset. As we collect data at each time t, we can also verify the impact of recourse when it is implemented for a smaller number of individuals.

Algorithm 1 Simulation Experiment

procedure Experiment($M, \mathcal{D}, G$)
  $t \leftarrow 0$
  while $t < T$ **do**
    $\mathcal{D}_B \subset \mathcal{D}_0$ ▷ Sample from $\mathcal{D}_0$.
    $\mathcal{D}_B \leftarrow G(\mathcal{D}_B)$ ▷ Generate counterfactuals.
    $M \leftarrow M(\mathcal{D})$ ▷ Retrain model.
  end while
  return $M, \mathcal{D}$
end procedure

Algorithm 1 summarizes the proposed simulation experiment for a given dataset $\mathcal{D}$, model M and generator G, but naturally we are interested in comparing simulation outcomes for different sources of data, models and generators. The framework we have built facilitates this, making use of multi-threading in order to speed up computations. Holding the initial model and dataset constant the experiments are run for all generators, since our primary concern is to benchmark different recourse methods. To ensure that each generator is faced with the exact same initial conditions in each round t, the candidate batch of individuals from the non-target class is randomly drawn from the intersection of all individuals in the non-target class across all experiments $\{\text{Experiment}(M, \mathcal{D}, G)\}_{j=1}^{J}$ where J is the total number of generators.

B. Evaluation Metrics

We formulate two desiderata for the set of metrics used to measure domain and model shifts induced by recourse. First, the metrics should be applicable regardless of the dataset or classification technique so that they allow for the meaningful comparison of the generators in various scenarios. As the knowledge of the underlying probability distribution is rarely available, the metrics should be empirical and non-parametric. This further ensures that we can also measure large datasets by sampling from the available data. Moreover, while our study was conducted in a two-class classification setting, our choice of metrics should remain applicable in the future research on multi- class recourse problems. Second, the set of metrics should allow to capture various aspects of the previously mentioned magnitude, path, and tempo of changes while remaining as small as possible.

1) Domain Shifts: To quantify the magnitude of domain shifts we rely on an unbiased estimate of the squared population **Maximum Mean Discrepancy (MMD)** given as:

$$MMD_u^2[F, X', \tilde{X}'] = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(\tilde{x}_i, \tilde{x}_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, \tilde{x}_j) \quad (5)$$

where $\mathcal{F}$ is a unit ball in a Reproducing Kernel Hilbert Space $\mathcal{H}$ [26], and X, $\tilde{X}$ represent independent and identically distributed samples drawn from probability distributions $\mathcal{X}$ and $\tilde{\mathcal{X}}$ respectively [27]. MMD is a measure of the distance between the kernel mean embeddings of $\mathcal{X}$ and $\tilde{\mathcal{X}}$ in RKHS $\mathcal{H}$. An important consideration is the choice of the kernel function $k(\cdot, \cdot)$. In our implementation we make use of Gaussian kernel with a constant length-scale parameter of 0.5. As the Gaussian kernel captures all moments of distributions $\mathcal{X}$ and $\tilde{\mathcal{X}}$, we have that $MMD_u^2[F, X, \tilde{X}] = 0$ if and only if $X = \tilde{X}$.

The evaluation metric in Equation 5 is computed after every round $t = 1, ..., T$ of the experiment. To assess the statistical significance of the observed shifts under the null hypothesis that samples X and $\tilde{X}$ were drawn from the same probability distribution, we follow [28]. To that end, we combine the two samples and generate a large number of permutations of $X + \tilde{X}$. Then, we split the permuted data into two new samples X' and $\tilde{X}'$ having the same size as the original samples. Then under the null hypothesis we should have that $MMD_u^2[F, X', \tilde{X}']$ be approximately equal to $MMD_u^2[F, X, \tilde{X}]$. The corresponding p-value can then be calculated by counting how these two quantities are not equal.

We calculate the MMD for both classes individually based on the ground truth labels, i.e. the labels that samples were assigned in time $t = 0$. Throughout our experiments, we generally do not expect the distribution of the negative class to change over time – application of recourse reduces the size of this class, but since individuals are sampled uniformly the distribution should remain unaffected. Conversely, unless a recourse generator can perfectly replicate the original probability distribution, we expect the MMD of the positive class to increase. Thus, when discussing MMD, we generally mean the shift in the distribution of the positive class.

2) Model Shifts: As our baseline for quantifying model shifts we measure perturbations to the model parameters at each point in time t following [15]. We define $\Delta = ||\theta_{t+1} - \theta_t||^2$, that is the euclidean distance between the vectors of parameters before and after retraining the model M. We shall refer to this baseline metric simply as **Perturbations**.

We extend the metric in Equation 5 for the purpose of quantifying model shifts. Specifically, we introduce **Predicted Probability MMD (PP MMD)**: instead of applying Equation 5 to features directly, we apply it to the predicted probabilities assigned to a set of samples by the model M. If the model shifts, the probabilities assigned to each sample will change; again, this metric will equal 0 only if the two classifiers are the same. We compute PP MMD in two ways: firstly, we compute it over samples drawn uniformly from the dataset, and, secondly, we compute it over points spanning a meshgrid over a subspace of the entire feature space. For the latter approach we bound the subspace by the extrema of each feature. While this approach is theoretically more robust, unfortunately, it suffers from the curse of dimensionality, since it becomes increasingly difficult to select enough points to overcome noise as the dimension D grows.

As an alternative to PP MMD we use a pseudo-distance for the **Disagreement Coefficient** (Disagreement). This metric was introduced in [29] and estimates $p(M(x) \neq M'(x))$, that is the probability that two classifiers do not agree on the predicted outcome for a randomly chosen sample. Thus, it is not relevant whether the classification is correct according to the ground truth, but only whether the sample lies on the same side of the two respective decision boundaries. In our context, this metric quantifies the overlap between the initial model (trained before the application of recourse) and the updated model. A Disagreement Coefficient unequal to zero is indicative of a model shift. The opposite is not true: even if the Disagreement Coefficient is equal to zero a model shift may still have occured. This is one reason for why PP MMD is our our preferred metric.

Finally, we introduce **Decisiveness** as a metric that quantifies the likelihood that a model assigns a high probability to its classification of any given sample. We define the metric simply as $\frac{1}{N} \sum_{i=0}^{N} (\sigma(M(x)) - 0.5)^2$ where $M(x)$ are predicted logits from a binary classifier and σ denotes the sigmoid function. This metric provides an unbiased estimate of the binary classifier's tendency to produce high-confidence predictions in either one of the two classes. Although the exact values for this metric are not important for our study, they can be used to detect model shifts. If decisiveness changes over time, then this is indicative of the decision boundary moves towards either one of the two classes.

V. Experiment Setup

This section presents the exact ingredients and parameter choices describing the simulation experiments we ran to produce the findings presented in the next section (Section VI). For convenience, we use Algorithm 1 as a template to guide us through this section. A few high-level details upfront: each experiment is run for a total of $T = 50$ rounds, where in each round we provide recourse to five percent of all individuals in the non-target class, so

Table I: Model Architectures

(a) MLP

	Hidden Dim.	Hidden Layers	Batch	Dropout
Synthetic	32	1	-	-
Real-World	32	2	50	0.25

(b) Variational Autoencoder

	Hidden Dim.	Epochs
Synthetic	2	100
Real-World	8	250

$B_t = 0.05 * N_t^{\mathcal{D}_0}$[5]. All classifiers and generative models are retrained for 10 epochs in each round t of the experiment. Rather than retraining models from scratch, we initialize all parameters at their previous levels $(t-1)$ and compute backpropagate for 10 epochs using the new training data as inputs into the existing model. Evaluation metrics are computed and stored every 10 rounds.

A. M – Classifiers and Generative Models

For each dataset and generator we look at three different types of classifiers all of them built and trained using `Flux.jl` [30]: firstly, a simple linear classifier - **Logistic Regression** - implemented as single linear layer with sigmoid activation; secondly, a multilayer perceptron (**MLP**); and finally, a **Deep Ensemble** composed of five MLPs following [31] that serves as our only probabilistic classifier. We have chosen to work with deep ensembles both for their simplicity and effectiveness at modelling predictive uncertainty. They are also the model of choice in [7]. The actual neural network architectures are kept simple (Table Ia), since we are only marginally concerned with achieving good initial classifier performance. For the real-world datasets we using mini-batch training and dropout regularization.

The Latent Space generators rely on separate generative models. Following the authors of both REVISE and CLUE we use Variational Autoencoders (**VAE**) for this purpose. As with the classifiers, we deliberately choose to work with fairly simple architectures (Table Ib). More expressive generative models generally also lead to more meaningful counterfactuals produced by Latent Space generators. But in our view this should simply be considered as a vulnerability of counterfactual generators that rely on surrogate models to learn what realistic representations of the underlying data.

B. $\mathcal{D}$ – Data

We have chosen to work with both synthetic and real-world datasets. Using synthetic data allows us to impose distributional properties that may affect the resulting recourse dynamics. Following [15], we generate synthetic data in $\mathbb{R}^2$ to also allow for a visual interpretation of the results. Real-world data is used in order to assess if endogenous dynamics also occur in higher-dimensional settings.

1) Synthetic data: We use four synthetic binary classification datasets consisting of 1000 samples each. The datasets are presented in Figure 2 (see also Appendix A for a formal description). Samples from the negative class are marked in blue while samples of the positive class are marked in orange.

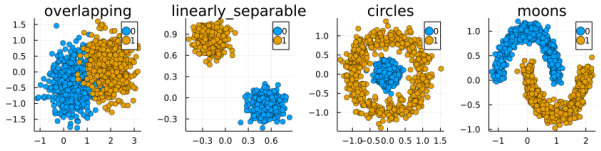

Figure 2: Synthetic classification datasets used in our experiments.

Ex-ante we expect to see that by construction Wachter will create a new cluster of counterfactual instances in the proximity of the initial decision boundary. Thus, the choice of a black-box model may have an impact on the paths of the recourse. For generators that use latent space search (REVISE [8], CLUE [10]) or rely on (and have access to) probabilistic models (CLUE [10], Greedy [7]) we expect that counterfactuals will end up in regions of the target domain that are densely populated by training samples. Of course, this is expectation hinges on how effective said probabilistic models are at capturing predictive uncertainty. Finally, we expect to see the counterfactuals generated by DiCE to be uniformly spread around the feature space inside the target class[6]. In summary, we expect that the endogenous shifts induced by Wachter outsize those induced by all other generators, since Wachter is the only approach that is not concered with generating what we have defined as meaningful counterfactuals.

2) Real-world data: We use three different real-world datasets from the Finance and Economics domain, all of which are tabular and can be used for binary classification. Firstly, we use the **Give Me Some Credit** dataset which was open-sourced on Kaggle for the task to predict whether a borrower is likely to experience financial difficulties in the next two years [32]. Originally consisting of 250,000 instances with 11 numerical attributes. Secondly, we use the **UCI defaultCredit** dataset [33], a benchmark dataset that can be used to train binary classifiers to predict the binary outcome variable, whether credit card clients default on their payment. In its raw form it consists of 23 explanatory variables - 4 categorical features relating to demographic attributes[7] and 19 continuous features

[5]As mentioned in the previous section, we end up providing recourse to a total of $\approx 50\%$ by the end of round $T = 50$.

[6]As we mentioned earlier, the diversity constraint used by DiCE is only effective for when at least two counterfactuals are being generated. We have therefore decided to always generate 5 counterfactuals for each generator and randomly pick one of them.

[7]These have been ommitted from the analysis. See Section VIII-B for details.

largely relating to individuals' payment histories and amount of credit outstanding. Both of these datasets have been used in the literature on Algorithmic Recourse before (see for example [4], [8] and [13]), presumably because they constitute real-world classification tasks involving individuals that compete for access to credit.

As a third dataset we include the **California Housing** dataset derived from the 1990 U.S. census and sourced through scikit-learn [35]. It consists of 8 continuous features that can be used to predict the median house price for California districts. The continuous outcome variable is binarized as $\tilde{y} = \mathbb{I}_{y>\text{median}(Y)}$ indicating whether or not the median house price of a given district is above or below the median of all districts. While we have not seen this dataset used in the previous literature on AR, others have used the Boston Housing dataset in a similar fashion (see for example [7]). While we initially also conducted experiments on that dataset, we eventually discarded this dataset, since it has been found to suffer from an ethical problem [36].

Since the simulations involve generating counterfactuals for a significant proportion of the entire sample of individuals, we have randomly undersampled each dataset to yield balanced subsamples consisting of 10,000 individuals each. We have also standardized all explanatory features since our chosen classifiers are sensetive to scale.

C. G – Generators

All generators introduced earlier are included in the experiments: Wachter [5], REVISE [8], CLUE [10], DiCE [9] and Greedy [7]. In addition, we introduce two new generators in Section VI-B that directly address the issue of endogenous domain and model shifts. We also test to what extent it may be beneficial to combine ideas underlying the various generators.

VI. Experiments

A. Endogenous Macrodynamics

B. Potential Mitigation Strategies

In the following we explore several ideas for strategies that may help to mitigate those types of endogenous recourse dynamics we observed in the previous subsection. All of them essentially boil down to one simple principle: to avoid substantial domain and model shifts, the generated counterfactuals should comply as much as possible with the true data generating process. This principle is really at the core of Latent Space generators, and hence it is not surprising that we have found these types of generators to perform comparably well in the previous subsection. But as we have mentioned earlier, generators that rely on separate generative models carry an additional computational cost and - perhaps more importantly - their performance hinges on the performance of said generative models. Fortunately, it turns out that we can use a number of other, much simpler strategies, which we will discuss now.

1) More Conservative Decision Thresholds: The most obvious and trivial mitigation strategy is to simply choose a higher decision threshold γ. Under $\gamma = 0.5$, counterfactuals will end up near the decision boundary by construction. Since this is the region of maximal aleotoric uncertainty, the classifier is bound to be thrown off. By simply setting a more conservative decision threshold, we can avoid this issue to some extent. A potential drawback of this approach is that a classifier with high decisiveness may classify samples with high confidence even far away from the training data.

2) Classifier Preserving ROAR: Another potential strategy draws inspiration from ROAR [15]: to preserve the classifier, we propose to simply explicitly penalize the loss it incurs when evaluated on the counterfactual x' at given parameter values. Recall that $g(\cdot)$ denotes what we had defined as the external cost in Equation 4. Then formally we let

$$g(f(s'_k)) = l(M(f(s'_k)), y') \tag{6}$$

for each counterfactual k where l denotes the loss function used to train M. This approach, which we shall refer to as **ClapROAR**, is based on the intuition that (endogenous) model shifts will be triggered by counterfactuals that increase classifier loss. It is closely linked to the idea of choosing higher decision threshold, but likely better at avoiding the potential pitfalls associated with highly decisive classifiers. It also makes the private vs. external cost trade-off more explicit and hence manageable.

3) Gravitational Counterfactual Explanations: Yet another strategy simply extends Wachter as follows: instead of only penalizing the distance of the individuals' counterfactual to its factual, we propose penalizing its distance to some sensible point in the target domain, for example the sample average: $\bar{x}$:

$$g(f(s'_k)) = \text{dist}(f(s'_k), \bar{x}) \tag{7}$$

Once again we can putting this in the context of Equation 4, the former penalty can be thought of here as the private cost incurred by the individual, while the latter reflects the external cost incurred by other individuals. Higher choices of λ_2 relative to λ_1 will lead counterfactuals to gravitate towards the specified point $\bar{x}$ in the target domain. In the remainder of this paper we will therefore refer to this approach as **Gravitational** generator, when we investigate its potential usefulness for mitigating endogenous macrodynamics[8].

[8]Note that despite the naming convention our goal here is not to provide yet another counterfactual generator, but merely investigate the most simple penalty we can think of with respect to its effectiveness.

Figure 3 shows an illustrative example that demonstrates the differences in counterfactual outcomes when using the various mitigation strategies compared to the baseline approach, that is Wachter with $\gamma = 0.5$: choosing a higher decision threshold pushes the counterfactual a little further into the target domain; this effect is even stronger for ClapROAR; finally, using the Gravitational generator the counterfactual ends up all the way inside the target domain in the neighbourhood of $\bar{x}$[9].

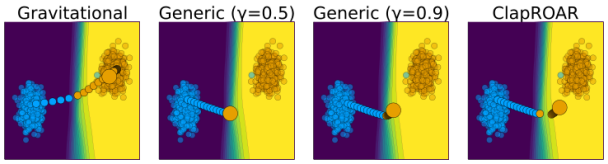

Figure 3: The differences in counterfactual outcomes when using the various mitigation strategies compared to the baseline approach, that is Wachter with $\gamma = 0.5$

Our findings ...

VII. Discussion

1. Shift of focus from individual to group of individuals (related: https://www.researchgate.net/publication/353073138_Generating_Collective_Counterfactual_Explanations_in_Score-Based_Classification_via_Mathematical_Optimization)
2. Convergence criterium matters: terminating once threshold probability is reached may not be optimal (see e.g. REVISE)
3. Optimizer choice matters: dimensionality is typically low, so no obvious benefit to using ADAM.
   - This might be better placed in JuliaCon proceedings, perhaps backed by small blog post on the matter.
4. Mitigating strategy: penaliye distance from centroid.

VIII. Limitations and Future Work

While we believe that this work constitutes a valuable starting point for addressing existing issues in Algorithmic Recourse from a fresh perspective, we are aware of several of its limitations. In the following we highlight some of these limitations and point to avenues for future research.

A. Experimental Setup

The experimental setup proposed here is designed to mimic a real-world recourse process in a simple fashion. In practice, models are in fact updated on a regular basis [15]. We also find it plausible to assume that the implementation of recourse happens periodically for different individuals, rather that all at once at time $t = 0$. That being said, our experimental design is a vast over-simplification of potential real-world scenarios. In practice, any endogenous shifts that may occur can be expected to be entangled with exogenous shifts of the nature investigated in [15]. We also make implicit assumptions about the utility functions of the involved agents that may well be too simple: individuals seeking recourse are assumed to always implement the proposed Counterfactual Explanations; conversely, the agent in charge of the model M is assumed to always treat individuals that have implemented valid recourse as if they were truly now in the target class. Relating this back to the consumer credit example, we assume that the would-be borrowers are always willing and able to implement recourse and the bank is always willing to provide credit as would-be borrowers move across the decision boundary. In practice it is doubtful that agents behave according to such simple rules. Nonetheless, we think that our simple framework offers a starting point for future work on recourse dynamics (both endogenous and exogenous dynamics).

B. Data

Largely in line with the existing literature on Algorithmic Recourse, we have limited our analysis of real-world data to three commonly used benchmark datasets that involve binary prediction tasks. Future work may benefit from including novel datasets or extending the analysis to multi-class or regression problems, the latter arguably representing the most common objective in Finance and Economics. It is also worth mentioning that the use of real-world datasets considered in this work is constrained by the fact that at the time of writing `CounterfactualExplanations.jl` only supports continuous features, at least of some of the counterfactual generators considered here. The fact that we therefore had to discard discrete features led to relatively poor initial performance of our classifiers in some cases. While this is indeed a limitation we intend to address in future and derivative work, our findings with respect to endogenous macrodynamics do not hinge on strong classifier performance.

C. Classifiers

For reasons stated earlier we have limited our analysis to differentiable linear and non-linear classifiers, in particular logistic regression and deep neural networks. While these sorts of classifiers have also typically been analyzed in the existing literature on Counterfactual Explanations and Algorithmic Recourse, they represent only a subset of popular machine learning models employed in practice - both black-box and glass-box. Despite the success and popularity of deep learning in the context of high-dimensional data such as image, audio and video, empirical evidence suggests that other models such as boosted decision trees may have an edge when it comes to lower-dimensional tabular datasets, such as the ones considered here [38].

[9] In order for the Graviational generator and ClapROAR to work as expected, one needs to ensure that counterfactual search continues, independent of the threshold probability γ.

IX. Concluding Remarks

Acknowledgment

P. A. thanks …

References

[1] C. Borch, "Machine learning, knowledge risk, and principal-agent problems in automated trading," *Technology in Society*, p. 101852, 2022.

[2] C. O'neil, *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown, 2016.

[3] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[4] M. Pawelczyk, S. Bielawski, J. van den Heuvel, T. Richter, and G. Kasneci, "Carla: A python library to benchmark algorithmic recourse and counterfactual explanation algorithms," *arXiv preprint arXiv:2108.00783*, 2021.

[5] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[6] P. Altmeyer, *CounterfactualExplanations.jl - a Julia package for Counterfactual Explanations and Algorithmic Recourse*. 2022. Available: https://github.com/pat-alt/CounterfactualExplanations.jl

[7] L. Schut *et al.*, "Generating interpretable counterfactual explanations by implicit minimisation of epistemic and aleatoric uncertainties," in *International conference on artificial intelligence and statistics*, 2021, pp. 1756–1764.

[8] S. Joshi, O. Koyejo, W. Vijitbenjaronk, B. Kim, and J. Ghosh, "Towards realistic individual recourse and actionable explanations in black-box decision making systems," *arXiv preprint arXiv:1907.09615*, 2019.

[9] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 607–617.

[10] J. Antorán, U. Bhatt, T. Adel, A. Weller, and J. M. Hernández-Lobato, "Getting a clue: A method for explaining uncertainty estimates," *arXiv preprint arXiv:2006.06848*, 2020.

[11] A.-H. Karimi, G. Barthe, B. Schölkopf, and I. Valera, "A survey of algorithmic recourse: Definitions, formulations, solutions, and prospects," *arXiv preprint arXiv:2010.04050*, 2020.

[12] S. Verma, J. Dickerson, and K. Hines, "Counterfactual explanations for machine learning: A review," *arXiv preprint arXiv:2010.10596*, 2020.

[13] B. Ustun, A. Spangher, and Y. Liu, "Actionable recourse in linear classification," in *Proceedings of the conference on fairness, accountability, and transparency*, 2019, pp. 10–19.

[14] A.-H. Karimi, B. Schölkopf, and I. Valera, "Algorithmic recourse: From counterfactual explanations to interventions," in *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 2021, pp. 353–362.

[15] S. Upadhyay, S. Joshi, and H. Lakkaraju, "Towards robust and reliable algorithmic recourse," *arXiv preprint arXiv:2102.13620*, 2021.

[16] E. Carrizosa, J. Ramırez-Ayerbe, and D. Romero, "Generating collective counterfactual explanations in score-based classification via mathematical optimization," 2021.

[17] S. Rabanser, S. Günnemann, and Z. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[18] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[19] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine learning*, vol. 23, no. 1, pp. 69–101, 1996.

[20] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[21] K. Nelson, G. Corbin, M. Anania, M. Kovacs, J. Tobias, and M. Blowers, "Evaluating model drift in machine learning algorithms," in *2015 IEEE symposium on computational intelligence for security and defense applications (CISDA)*, 2015, pp. 1–8.

[22] S. Ackerman, P. Dube, E. Farchi, O. Raz, and M. Zalmanovici, "Machine learning model drift detection via weak data slices," in *2021 IEEE/ACM third international workshop on deep learning for testing and testing for deep learning (DeepTest)*, 2021, pp. 1–8.

[23] R. M. B. de Oliveira and D. Martens, "A framework and benchmarking study for counterfactual generating methods on tabular data," *Applied Sciences*, vol. 11, no. 16, p. 7274, 2021.

[24] A.-K. Dombrowski, J. E. Gerken, and P. Kessel, "Diffeomorphic explanations with normalizing flows," 2021.

[25] R. S. Pindyck and D. L. Rubinfeld, *Microeconomics*. Pearson Education, 2014.

[26] A. Berlinet and C. Thomas-Agnan, *Reproducing kernel hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[27] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[28] M. A. Arcones and E. Gine, "On the bootstrap of u and v statistics," *The Annals of Statistics*, pp. 655–674, 1992.

[29] S. Hanneke, "A bound on the label complexity of agnostic active learning," in *Proceedings of the 24th international conference on machine learning*, 2007, pp. 353–360.

[30] M. Innes *et al.*, "Fashionable modelling with flux," *arXiv preprint arXiv:1811.01457*, 2018.

[31] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *arXiv preprint arXiv:1612.01474*, 2016.

[32] K. Competition, "Give me some credit, improve on the state of the art in credit scoring by predicting the probability that somebody will experience financial distress in the next two years." https://www.kaggle.com/c/GiveMeSomeCredit

[33] I.-C. Yeh and C. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert systems with applications*, vol. 36, no. 2, pp. 2473–2480, 2009.

[34] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[35] R. K. Pace and R. Barry, "Sparse spatial autoregressions," *Statistics & Probability Letters*, vol. 33, no. 3, pp. 291–297, 1997.

[36] C. M., "Racist data destruction? - a boston housing dataset controversy," 2019, Available: https://medium.com/@docintangible/racist-data-destruction-113e3eff54a8

[37] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," *arXiv preprint arXiv:2110.01889*, 2021.

[38] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on tabular data?" *arXiv preprint arXiv:2207.08815*, 2022.

X. Tables

...

XI. Figures

...

XII. Code

...