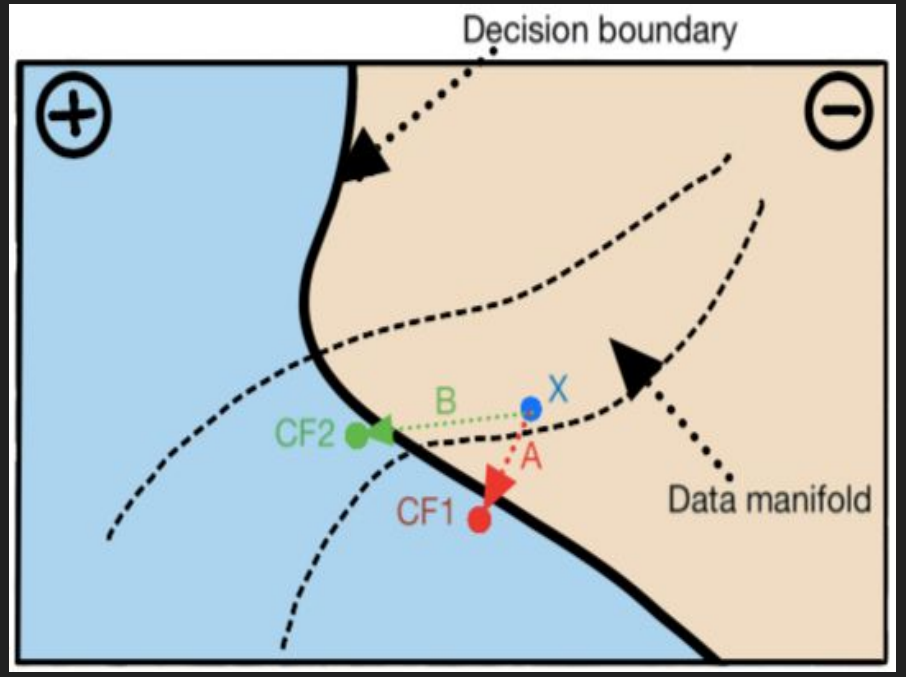
Amortized Generation of Sequential Counterfactual Explanations for Black-box Models

What are Counterfactual Explanations?

- Emerging explainability technique in Machine Learning.
- For a datapoint and a model, a counter factual explanation is a datapoint in the vicinity, but with a different prediction (additional desiderata).
- Helpful in answering questions about the changes required to be brought for getting to other side of decision boundary.



Verma, S., Dickerson, J., & Hines, K. (2020). Counterfactual Explanations for Machine Learning: A Review.

Karimi, A. H., Barthe, G., Schölkopf, B., & Valera, I. (2020). A survey of algorithmic recourse: definitions, formulations, solutions, and prospects.

Desiderata of Counterfactual Explanations

- Actionability and Mutability of features.
- Sparsity
- Adherence to Data Manifold
- Respect for causal relations

- Model-agnostic
- Black-box approach
- Amortized Inference

FastCF

- We propose a novel stochastic control based approach for generating counterfactual explanations (CFEs).
- We translate a counterfactual problem into a Markov Decision Process (MDP) and use a RL algorithm to learn a policy that generates counterfactuals for datapoints from a distribution.
- Our approach is model-agnostic and works for black-box models.
- It provides amortized inference (as it learns a policy).
- It only changes the actionable features and generates sparse CFEs close to training data manifold while respecting causal relations.

Qualitative Comparison with Previous Approaches

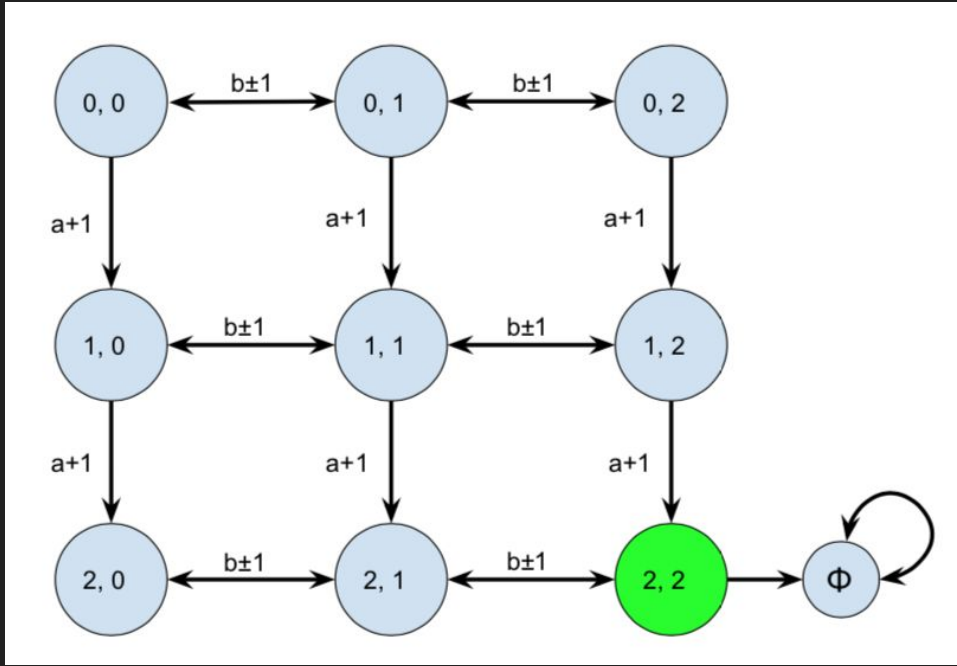
FASTCF *is the* ***first and only one*** *which satisfies all desiderata.*

Approach	Action'ty	Sparse	Agnostic	Black-box	Amortize	Manifold	Causality
CF Expl. [65]	✗	✓	✗	✗	✗	✗	✗
Recourse [61]	✓	✓	✗	✗	✗	✗	✗
CEM [11]	✗	✓	✗	✗	✗	✓	✗
MACE [26]	✓	✓	✗	✗	✗	✗	✗
DACE [25]	✓	✗	✗	✗	✗	✓	✗
DICE [39]	✓	✓	✗	✗	✗	✗	✗
VAE CFs [36]	✓	✗	✗	✗	✓	✓	✓
Spheres [34]	✗	✓	✓	✓	✗	✗	✗
LORE [20]	✗	✓	✓	✓	✗	✗	✗
Weighted [19]	✗	✗	✓	✓	✗	✗	✗
CERTIFAI [55]	✓	✗	✓	✓	✗	✗	✗
Prototypes [62]	✗	✓	✓	✓	✗	✓	✗
MOC [7]	✓	✓	✓	✓	✗	✓	✗
FASTCF	✓	✓	✓	✓	✓	✓	✓

MDP Formulation

MDP is a tuple consisting of:

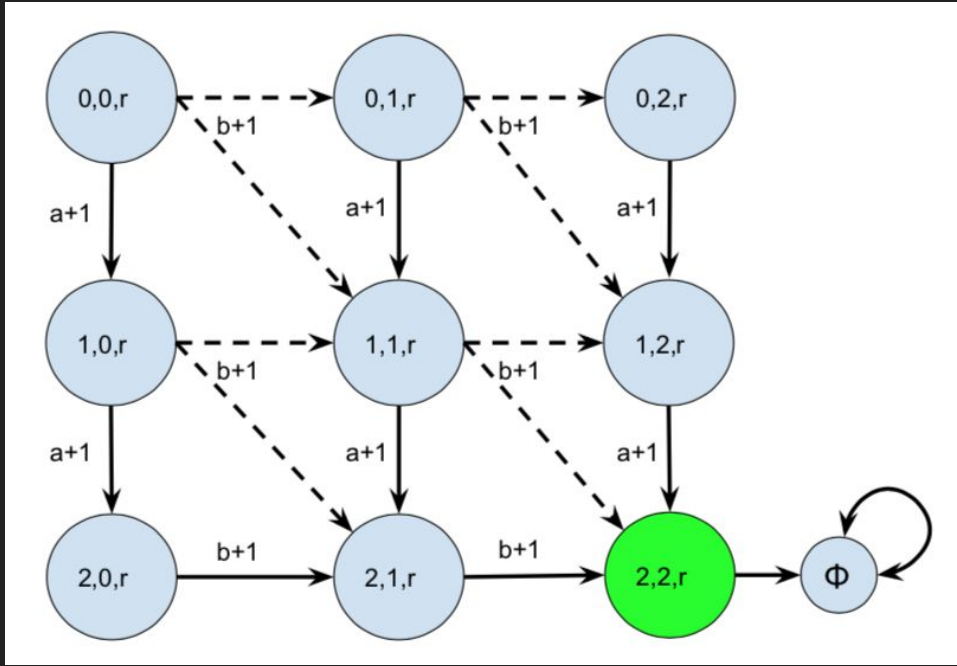
- State Space
  {0,0},{0,1},{0,2},{1,0},...}
- Action Space
  {a+1, a-1, b+1, b-1}
- Transition Function T :S×A→S
- Reward Function r:S×A→R.
- Discount Factor γ ∈ [0, 1]



MDP Formulation

MDP is a tuple consisting of:

- State Space
  {0,0,0},{0,1,0},{0,2,1},{1,0,0},...
- Action Space
  {a+1, a-1, b+1, b-1}
- Transition Function T:S×A×S′→
  {0,1}
- Reward Function r:S×A→R
- Discount Factor γ ∈ [0, 1]



Algorithm

ALGORITHM 1: Generate MDP from a Counterfactual Explanation Problem

Input : Training Dataset (D), ML model (f), Structural Causal Model (SCM), Actionable features (ActF), Data Manifold distance function (DistD), Data Manifold adherence (λ), Desired Label (L), Distance Function (DistF), Discount Factor (γ)

Output : MDP

// States consist of numerical (Num) and categorical (Cat) features.

State space $\mathcal{S} \subseteq \mathbb{R}^{|\text{Num}|} \times \mathbb{Z}^{|\text{Cat}|}$

// Actions change an actionable feature by some amount.

Action space $\mathcal{A} \subseteq \mathbb{R}^{|\text{ActF}|}$; denote actions $A \in \mathcal{A}$

Function Reward(f, L, CurrState, A, D, λ, DistD, SCM)

  NextState $\leftarrow$ Transition(CurrState, A, SCM)

  if argmax(f(NextState)) $=$ L **then**

    CFReward $\leftarrow$ Pos // High positive reward

  else

    CFReward $\leftarrow$ f(NextState)[L] // Probability of classification in the desired class

  return DistF(CurrState, A, D) // cost of an action

  $+\lambda *$ DistD(NextState, D) // Manifold distance cost

  +CFReward // Counterfactual label reward

Function Transition(CurrState, A, SCM)

  // Action does not violate feature domain and unary constraints

  if Allowed (A) & InDomain (A) **then**

    NextState $\leftarrow$ CurrState $+ A$ // Modify features

  else

    return CurrState

  // Modify the endogenous features

  for V $\in$ SCM **do**

    if $A \in$ Parent(V) **then**

      NextState[V] $\leftarrow$ F(U) // Stochastic or deterministic update of endogenous features

  return NextState

MDP $\leftarrow \{\mathcal{S}, \mathcal{A}, \text{Transition}, \text{Reward}, \gamma\}$

Evaluation

We performed experiments to answer the following research questions:

1. Does FastCF successfully generate CFEs for the input datapoints (validity)?
2. How much change is required to reach a counterfactual state (proximity)?
3. How many features are changed to reach a counterfactual state (sparsity)?
4. Do the generated CFEs adhere to the data manifold (realisticness)?
5. Do the generated CFEs respect causal relations (feasibility)?
6. How much time does FastCF take to generate CFEs (amortizability)?

Datasets

We use three datasets in our evaluation (all from UCI):

1. German Credit: 20 features.
2. Adult Income: 13 features.
3. Credit Default: 23 features.

Dataset	Causal relations	Immutable features
German Credit	Age and Job cannot decrease	Foreign worker, Number of liable people, Personal status, Purpose
Adult Income	Age and Education cannot decrease, increasing Education increases Age	Marital-status, Race, Native-country, Sex
Credit Default	Age and Education cannot decrease, increasing Education increases Age	Sex, Marital status

Baselines

Previous CFE generating approaches:

- Complete model access: MACE
- Gradient access: DiCE
- Black-box access: Model-agnostic versions of DiCE: random, genetic, kdtree.

Simple baselines we developed (black-box and amortized inference):

- Random approach
- Greedy approach

Implementation

Any approach that given an MDP, learns a policy would be appropriate. We choose PPO+GAE algorithm for our implementation.

Results

Dataset	Approach	#DataPts.	Validity	Prox-Num	Prox-Cat	Sparsity	Manifold dist.	Causality	Time (s)
German Credit	Random	257	23.7	0.17	0.57	11.33	1.08	41.0	0.31
	Greedy	257	49.8	**0.07**	0.087	1.81	0.48	**100.0**	4.59
	DiCE-Genetic	257	98.1	0.67	0.26	6.52	2.39	45.6	1.71
	DiCE-KDTree	257	0.0	N/A	N/A	N/A	N/A	N/A	0.17
	DiCE-Random	257	**100.0**	0.33	0.10	1.93	2.40	93.4	0.17
	DiCE-Gradient	257	**100.0**	0.27	0.29	6.33	2.19	82.9	7.10
	MACE (LR)	210	**100.0**	0.36	**0.017**	1.99	0.60	97.1	38.45
	MACE (RF)	287	**100.0**	0.22	0.02	2.64	**0.38**	74.2	101.29
	FastCF	257	97.3	0.10	0.063	**1.22**	0.72	**100.0**	**0.07**
Adult Income	Random	7229	80.9	0.56	0.77	10.07	1.00	29.0	0.25
	Greedy	7229	97.7	**0.04**	0.02	1.18	**0.17**	95.0	0.27
	DiCE-Genetic	7229	89.5	0.71	0.27	4.43	0.46	23.0	3.43
	DiCE-KDTree	7229	0.0	N/A	N/A	N/A	N/A	N/A	0.59
	DiCE-Random	7229	**100.0**	0.82	0.04	1.64	1.24	90.0	0.22
	DiCE-Gradient	500	84.0	0.18	0.012	2.78	0.51	82.4	59.75
	FastCF	7229	**100.0**	**0.04**	**0.0**	**1.00**	0.18	**100.0**	**0.015**
Credit Default	Random	5363	12.8	4.85	0.68	14.54	1.30	41.5	0.63
	Greedy	5363	65.1	0.15	**0.072**	1.25	**0.22**	99.9	4.67
	DiCE-Genetic	5363	92.6	3.93	0.49	16.67	2.75	27.9	3.58
	DiCE-KDTree	5363	0.0	N/A	N/A	N/A	N/A	N/A	0.45
	DiCE-Random	5363	**100.0**	5.80	0.20	2.33	3.09	97.7	0.39
	DiCE-Gradient	100	81.0	0.77	0.40	15.98	1.35	85.2	479.17
	FastCF	5363	99.9	**0.01**	0.11	**1.00**	0.32	**100.0**	**0.051**

Results

- RQ1: FastCF achieves high validity, 100% for Adult and 99.9% for Default. DiCE-Random, DiCE-genetic, DiCE-Gradient, and MACE also have high validity, but at the cost of other metrics. DiCE-KDTree fails for all datapoints. Greedy and Random approach have low validity.
- RQ2: FastCF produces most proximal CFEs. MACE's performance is average. Greedy approach performs good but has a low validity.
- RQ3: FastCF achieves the lowest sparsity. Greedy, MACE, and DiCE-Random's performance is about average. Other baselines are abysmal.

Results

- RQ4: FastCF has a low manifold distance (best for Adult and Default, average for German). Greedy and MACE also do well on this metric.
- RQ5: FastCF respects all causal relations in all generated CFEs. All other baselines apart from greedy perform abysmally.
- RQ6: FastCF is very fast in generating CFEs. It is **8X** faster than the DiCE-Random and **11X** faster than Random, which are the next fastest. FastCF is about **1000X** faster than MACE and **4500X** faster than DiCE-Gradient. For other baselines it is somewhere in the middle. The one-time cost to learn a policy ranged between 2 to 12 hours.

Conclusions and Future Work

- FastCF qualitatively and quantitatively performs much better than most previous popular CFE generating approaches.
- Future work avenues might include modelling CFEs as SSPs (instead of current MDPs). This will force the agent to prescribe a CFE for any input datapoints, however high the cost might be. In the current scenario, of MDP, for specific datapoints the optimal path might be to not act at all, leading to no CFE. SSPs can alleviate this problem.