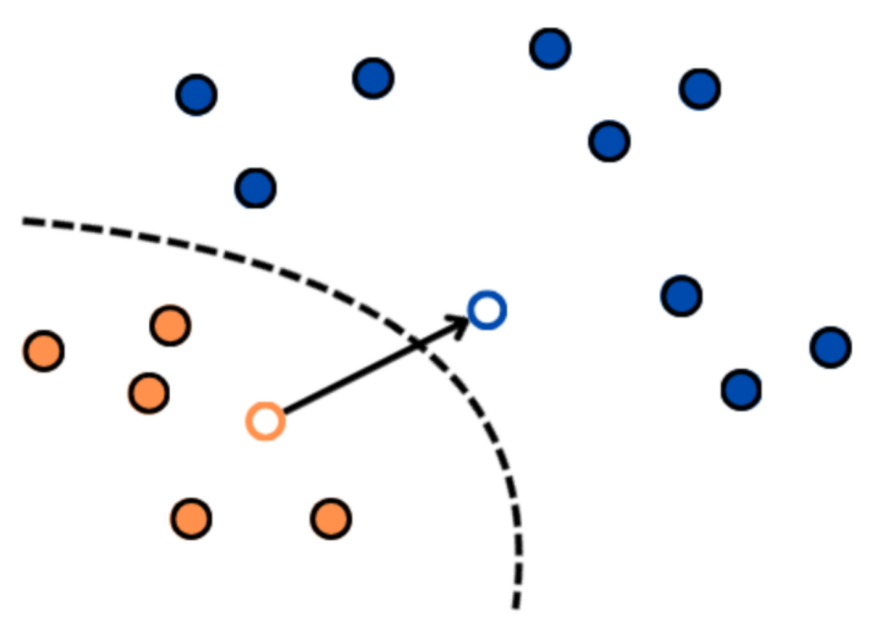
Make your counterfactual explanations robust to model change!

We introduce BetaRCE, a Bayesian-inspired method for generating counterfactual explanations that are robust to model change. It offers probabilistic guarantees for robustness, works with any model type and allows you to control the robustness-cost trade-off according to your needs!

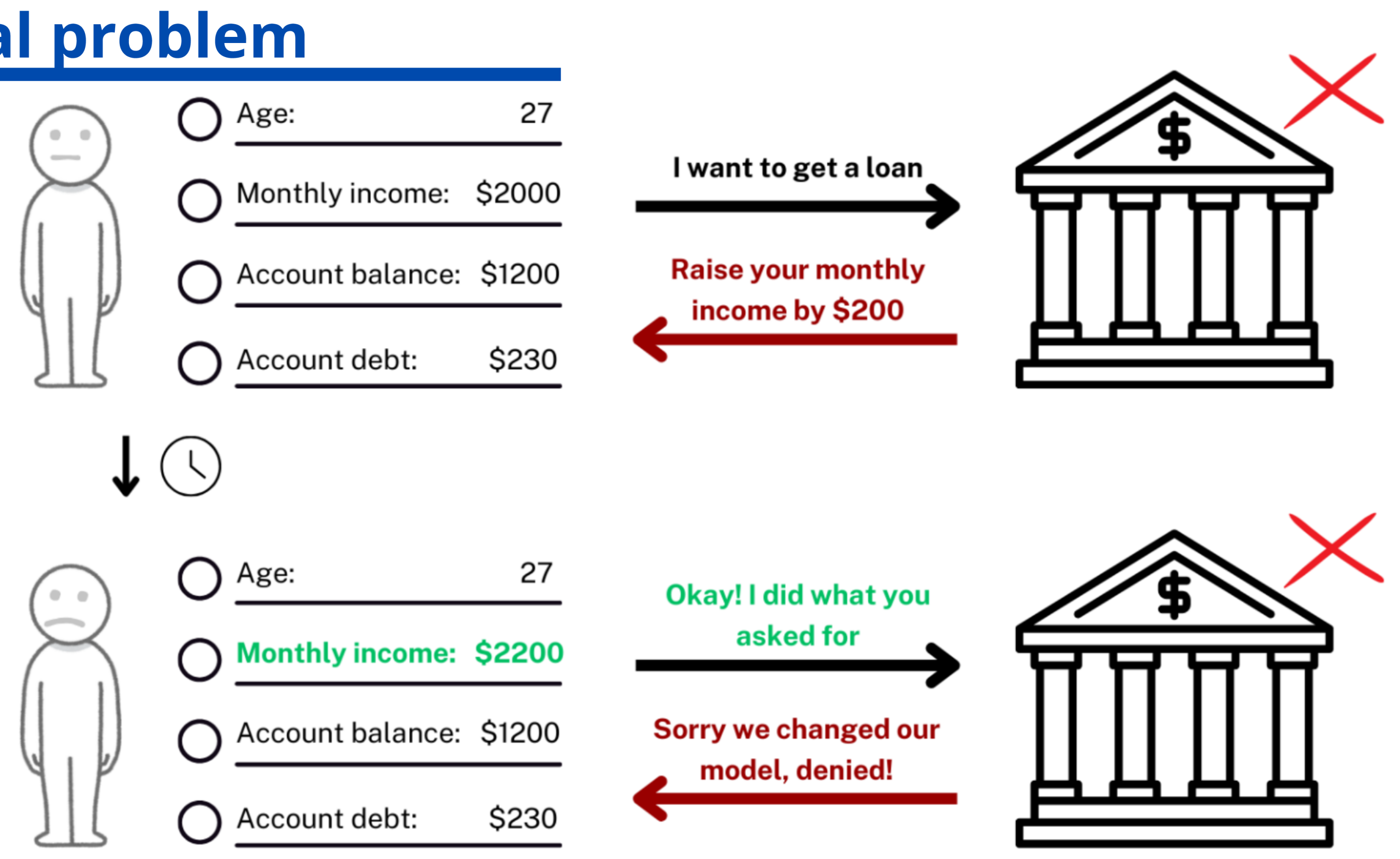
A quick primer on counterfactuals

Counterfactual explanations (CFs) help users understand and change the outcome of a classification model by showing how to move an input across the model's decision boundary. For example, if a model classifies a loan application as "rejected," a CF might say, "If your monthly income were $200 higher, the model would classify you as 'approved'." CFs highlight the minimal changes needed to flip the model's decision, giving clear, actionable feedback



Robust counterfactual problem

Imagine that you apply for a loan, get rejected, and the CF issued by the bank suggests: 'Increase your monthly income by $200 to get approved'. Cool, you follow this advice, but some time later, when you reapply, the model has been updated, and the original counterfactual turned out to be now invalid. This is the problem of non-robust counterfactuals. Real-world AI systems change, and we need explanations that can survive these updates.



○ Age: 27

○ Monthly income: $2000

○ Account balance: $1200

○ Account debt: $230

I want to get a loan

Raise your monthly income by $200

○ Age: 27

○ Monthly income: $2200

○ Account balance: $1200

○ Account debt: $230

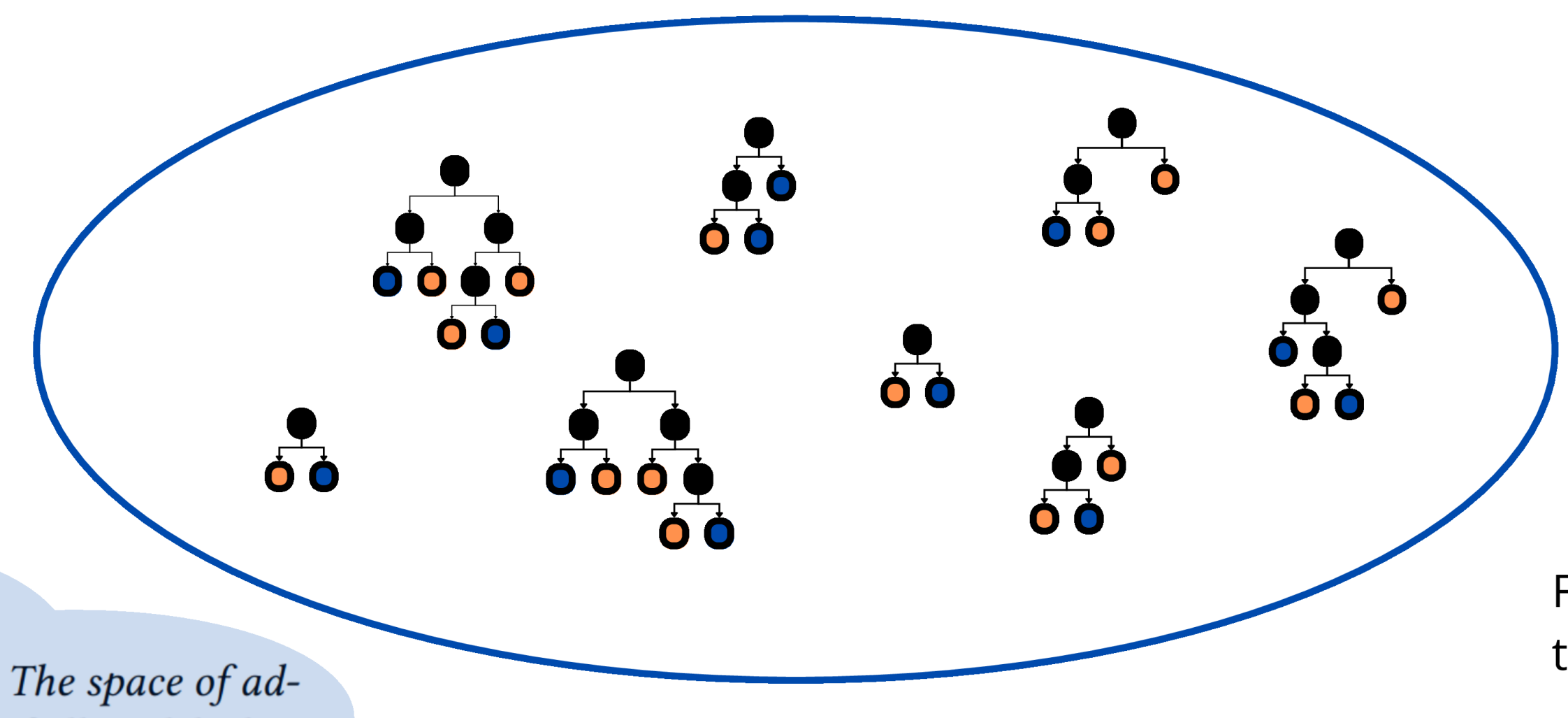
Okay! I did what you asked for

Sorry we changed our model, denied!

Definition 1 (Robust counterfactual). *A counterfactual x^{cf} explaining the prediction of a model M is robust to its change to a model M' if x^{cf} is classified identically by the original and changed model: $M(x^{cf}) = M'(x^{cf})$.*

Space of Admissible Models (SAM)

We start from defining SAM, a space that contains models which you expect to might see in the future. For instance these changes can be as simple as model retraining + architecture modifications / data shift / seed change etc.



Definition 2 (Space of admissible models). *The space of admissible models $\mathcal{M}_M$ is the probabilistic distribution of all models that are the result of a complete retraining of the model M using arbitrary settings from the predefined set of model changes.*

Feel free to skip the theory clouds!

Defining robustness

Definition 3 (δ-robust counterfactual). *A counterfactual x^{cf} is said to be δ-robust if and only if it is robust to change to a model randomly drawn from the given admissible model space $\mathcal{M}_M$ with probability at least δ.*

$$P(M'(x^{cf}) = M(x^{cf})) \geq \delta \qquad M' \sim \mathcal{M}_M \tag{1}$$

Next, we formally define robustness in the following way:

if M(x) = M'(x) **then robust** // a CF is robust when it preserves it's class for some new model

robust ~ B(**p**) // robustness is either true or false, so it follows a Binomial distr.

p ~ **Beta**(**a**, **b**) // parameter p in Binomial distr. has a conjugate Beta prior

P(**robust**) = **p** > δ // **δ-robust** if it is robust with probability over **δ**

P(**p** > δ) > α // **(δ,α)-robust** if it is **δ-robust** with **α** confidence

Definition 4 ((δ,α)-robust counterfactual). *A counterfactual x^{cf} is said to be (δ,α)-robust if and only if it is robust to change to a model randomly drawn from the admissible model space $\mathcal{M}_M$ with probability at least δ given the confidence level α.*

$$P(\hat{\delta} > \delta) > \alpha \tag{2}$$

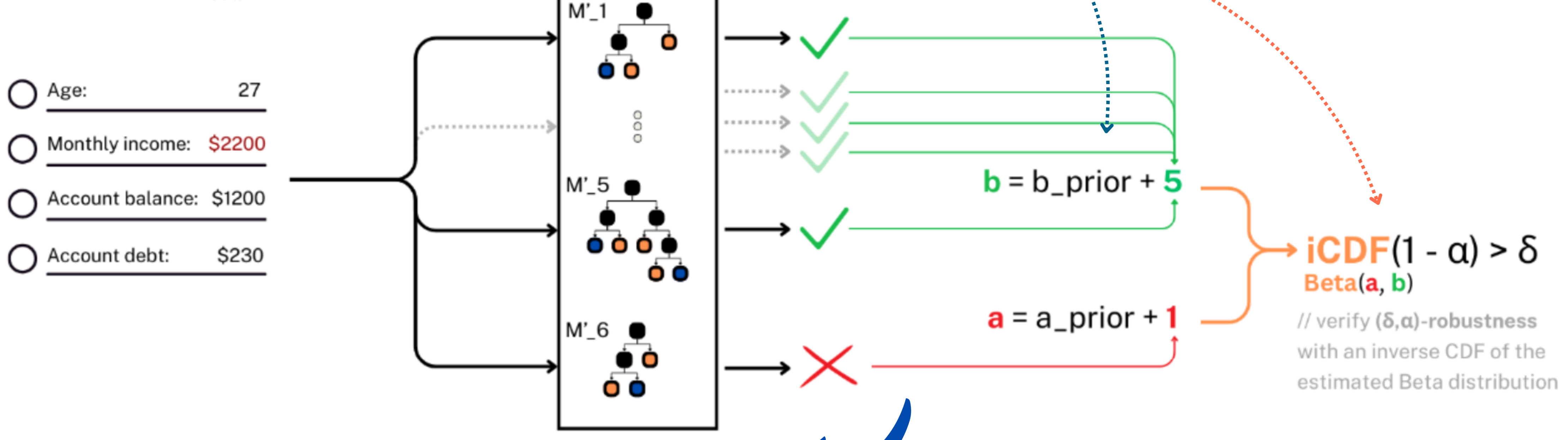
where $\hat{\delta}$ follows the a posteriori distribution representing the uncertainty regarding the estimated probability of a binary random event $[M'(x^{cf}) = M(x^{cf})]$.

Verification

We perform a simple bootstrap verification to check whether a counterfactual is (δ, α)-robust (Theorem 1)

1. Sample *k* models from SAM
2. Classify *x* with these models
3. For each prediction check if *M(x) == M'(x)*
4. Use this evidence to update the noninformative Jeffreys prior for \hat$\delta$ that follows Beta distribution
5. Finally, verify if the counterfactual is (δ, α)-robust

$$F^{-1}_{Beta}(1-\alpha) \geq \delta$$



○ Age: 27

○ Monthly income: $2200

○ Account balance: $1200

○ Account debt: $230

b = b_prior + **5**

a = a_prior + **1**

iCDF(1 - α) > δ
Beta(**a**, **b**)

// verify **(δ,α)-robustness** with an inverse CDF of the estimated Beta distribution

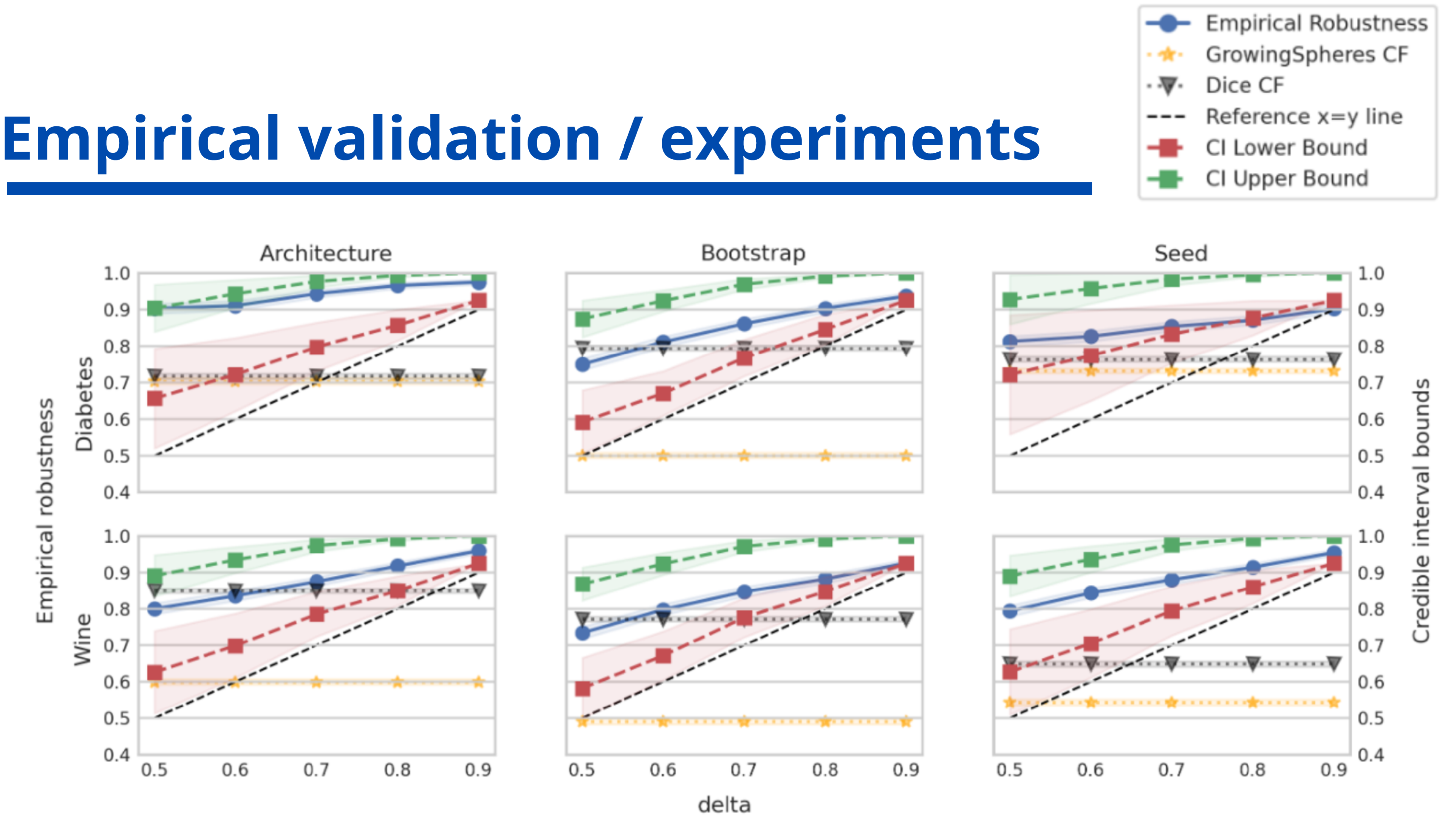
Generating counterfactuals

- BetaRCE is designed to be a post-hoc method, that is take in a counterfactual and update it to meet the robustness criteria
- Optimization uses the following objective function

$$x^* = \underset{x' \in \mathcal{X}}{\operatorname{argmin}}\, d(x^{cf}, x') \quad \text{s.t.} \quad valid(x', x^{orig}) \wedge robust(x') \tag{6}$$

  where d is the distance to the base counterfactual
- We utilize GrowingSpheres [1] to optimize this objective, but we note that any zeroth-order optimizer could be used

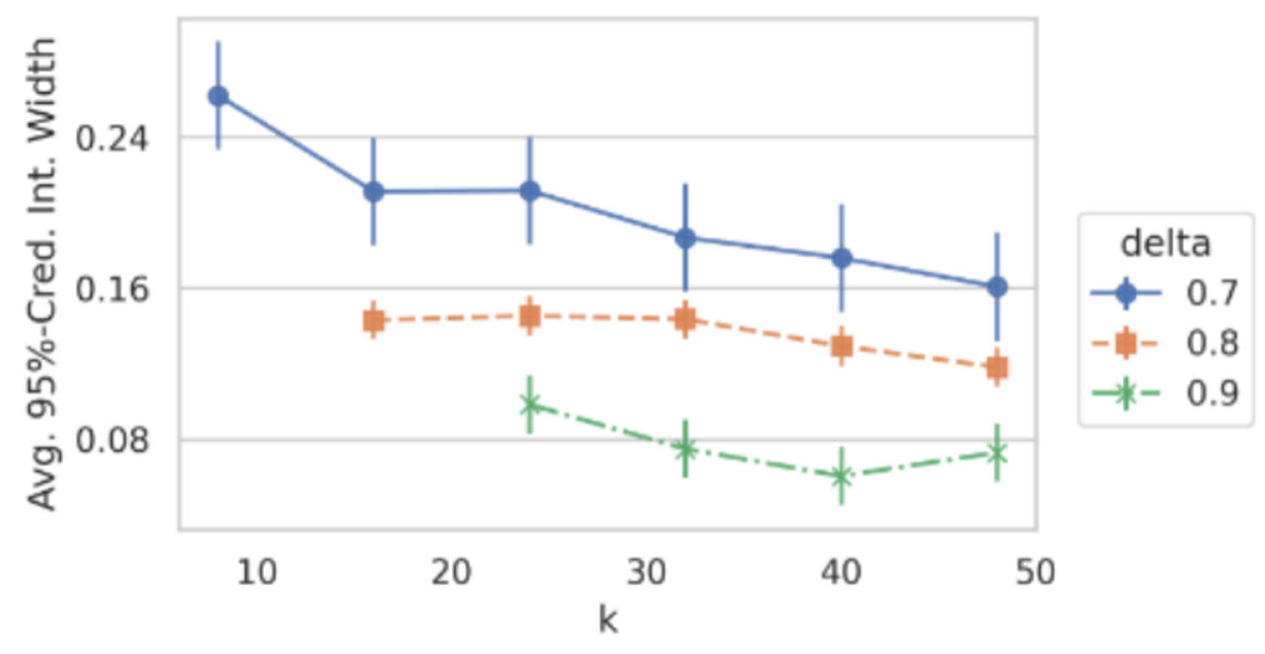
Empirical validation / experiments



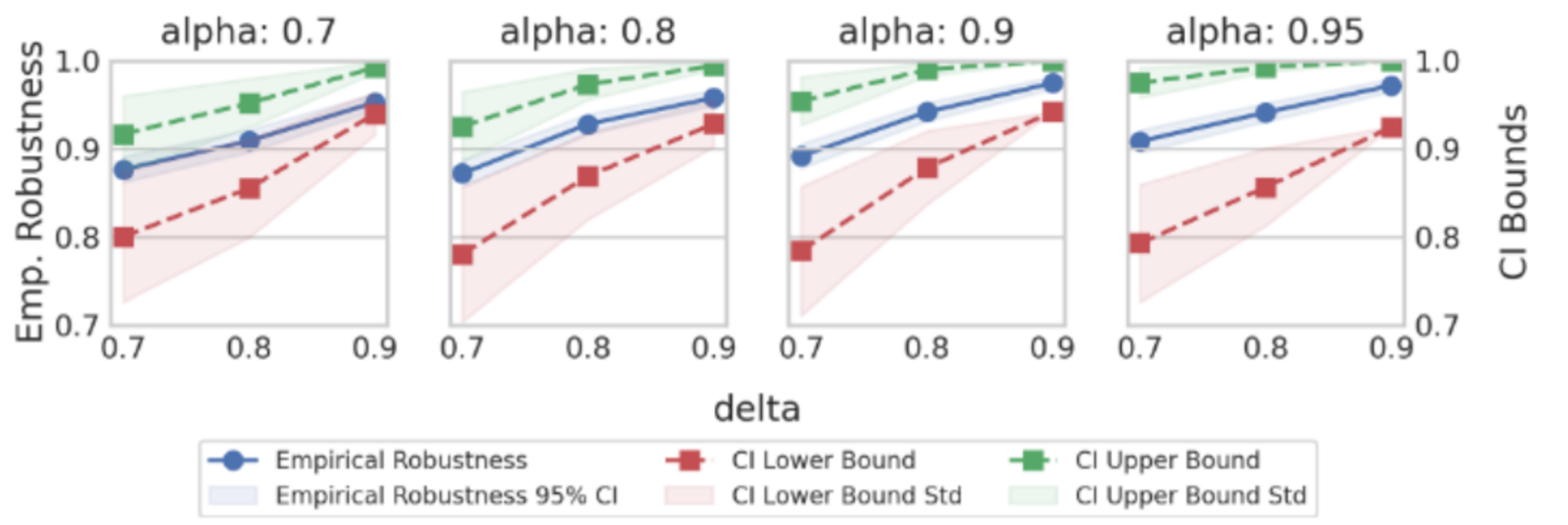
$$\text{Empirical Robustness} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{M(x_i^{cf})=M'(x_i^{cf})}$$

See our paper for quite a lot more visualizations and results!

The impact of *k*



The impact of δ



Generalization - out-of-distribution SAM

Original	Generalization Architecture	 Bootstrap	 Seed
Architecture	0.913 ± 0.007	0.865 ± 0.009	0.923 ± 0.007
Bootstrap	0.939 ± 0.006	0.877 ± 0.008	0.909 ± 0.007
Seed	0.927 ± 0.007	0.866 ± 0.009	0.890 ± 0.008

What about BetaRCE hyperparameters?

We found (Theorem 2) that all three hyperparameters introduced to derive BetaRCE are entangled by the following formula, which defines the maximum attainable delta, given fixed k and alpha. This enables user to set two of these parameters, likely alpha and delta, and have the third one set automatically to the most convenient value

$$\delta_{max} = F^{-1}_{Beta_{(a+k,b)}}(1-\alpha)$$

Main takeaways

- BetaRCE is a post-hoc robust CF generation method providing probabilistic robustness guarantees in a model-agnostic manner with interpretable hyperparameters (δ and α)
- Experiments show that target robustness levels are consistently achieved, validating the theorethical framework
- Empirically we see that BetaRCE outperforms other robustness-focused methods in robustness-cost tradeoff (see the extended evaluation in the paper)

Key references

[1] Laguel et al. 2018 "Comparison-based Inverse Classification for Interpretability in Machine Learning

Counterfactual Explanations with Probabilistic Guarantees on their Robustness to Model Change

Ignacy Stępka [a], Jerzy Stefanowski [a], Mateusz Lango [a b]

[a] Poznan University of Technology, Poznan, Poland
[b] Charles University, Prague, Czech Republic

This research was funded in whole by National Science Centre, Poland OPUS grant no. 2023/51/B/ST6/00545

Check out our project website!