Counterfactual Training

Update Meeting Dec 2024

Patrick Altmeyer Arie van Deursen Cynthia C. S. Liem

Delft University of Technology

2024-12-18

Status

- *Code base*: In place and streamlined for reproducibility and configuration.
- *Experiments*: Lots of work done and results largely supportive of idea.
  - Ran into problems on DelftBlue, which has set me back about 2 weeks.
- *Paper*: Still bare-bones.
- *ICML*: Potentially still possible to submit something, but this will be rushed and not "finished".

Problems on Cluster

- Trying to distribute:
  1. Models/experiments across processes.
  2. For each model/experiment distribute the counterfactual search across processes.
- Out-of-memory issues, data races, ...
- Multi-processing for models & multi-threading for counterfactual search: low CPU efficiency on DelftBlue (jobs get cancelled).

High-Level Idea

Counterfactual Training (CT) combines ideas from Energy-Based Models and Adversarial Training:

$$\ell_{\text{clf}}(f_\theta(x), y) + \lambda_{\text{gen}} \ell_{\text{gen}}({x'}_t, x_t; \theta) + \lambda_{\text{adv}} \ell_{\text{clf}}(f_\theta({x'}_t), y)$$

- ${x'}_t$ are counterfactuals of $x_s \subseteq x$ with target class t.
- ℓ_{gen} is the difference in energies between observed samples in target class x_t and counterfactuals.
- Counterfactuals are recycled as adversarial examples.

Training Details

During each `EPOCH`:

1. Generate `nce` counterfactuals and distribute across mini-batches.
2. For each batch compute:
   - Classifier loss: $\ell_{\text{clf}}(f_\theta(x), y)$
   - Generator loss: $\ell_{\text{gen}}({x'}_t, x_t; \theta)$
   - Adversarial loss: $\lambda_{\text{adv}} \ell_{\text{clf}}(f_\theta({x'}_t), y)$
3. Backpropagate all losses and update parameters.

Motivation and Intuition

- Instead of using SGLD to sample from $p(x|t;\theta)$, we use counterfactual generators.
- The idea is to align counterfactual explanations with observed data to induce plausibility.
- This should only work if counterfactuals are generated faithfully (favorable evidence).
- Approach can be leveraged to impicitly encode mutability and domain constraints in model.

Encoding Domain Knowledge

Let $f_\theta(x) = \theta^T x$ be a linear classifier:

$$\nabla_\theta \ell_{\text{gen}}({x'}_t, x_t; \theta) = \nabla_\theta(\theta^T x_t - \theta^T {x'}_t)$$
$$\frac{\partial \ell_{\text{gen}}}{\partial \theta[1]}(x', x; \theta) = x_t[1] - x'_t[1]$$

Suppose that featue $x[1]$ is immutable (e.g. 'age'), so $x'_t[1] = x_s[1]$ where $s \neq t$. If $x_t[1] > x_s[1]$:

- ℓ_{gen} induces lower values of $\theta[1]$, acting as a hedge against ℓ_{clf}, which favours higher $\theta[1]$.

Findings

Moons (Plausibility)

- All counterfactuals at test time generated using *ECCo*.
- Penalty on energy differential increases from *l.* to *r.*

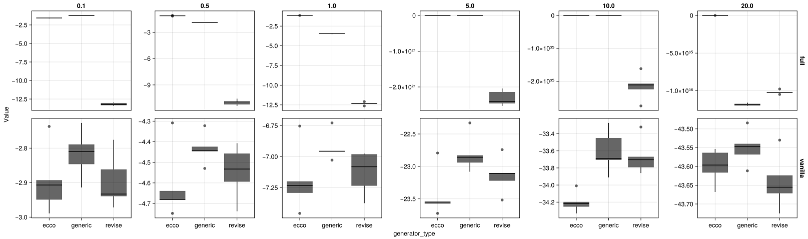

Figure 1: Plausibility of faithful counterfactuals x_t' measured in terms of their distance from x_t.

Moons (Example)

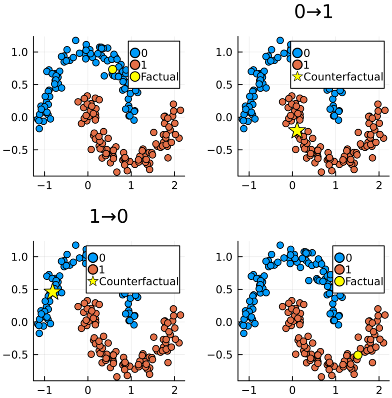

Figure 2: Counterfactual explanations for model trained with CT (*ECCo*).

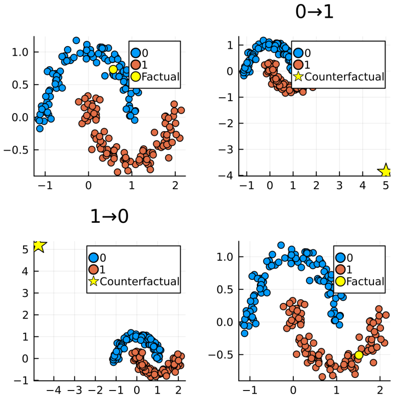

Figure 3: Counterfactual explanations for conventionally trained model.

Moons (Validation Accuracy)

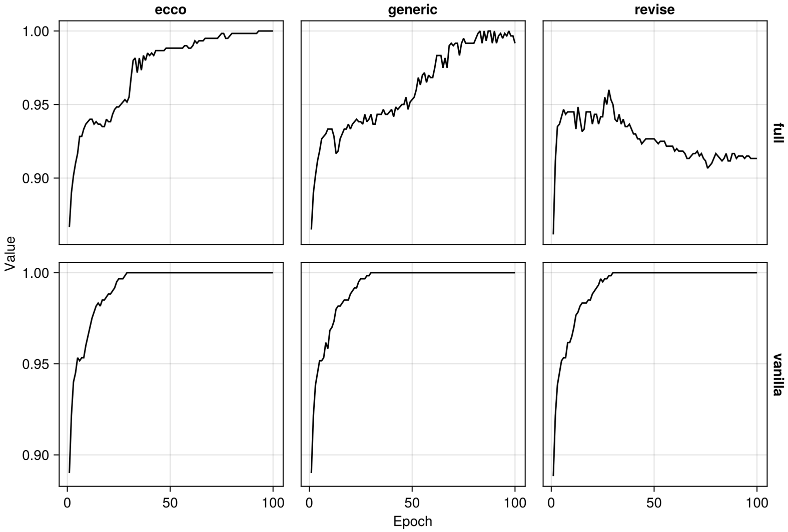

Figure 4: Validation accuracy for different models.

Planning Ahead