Research PR 1 - Counterfactual Explanations for ML October 2020

• link betⁿ wt cud happen given i/p changed in specific way

Post-hoc explanations for opaque models → model specific
→ model agnostic

feature importance
→ SHAP
→ QII

Visual
→ Partial Dependency
→ Accumulated Local Effects
→ Individual Condⁿ Expectation

local approx
→ LIME
→ Anchor

local example based
→ Counterfactual Exp

* QII (Quantative Input Influence
sees feat. correlations by randomly changing 1 feat at a time & calc. avg. marginal expected contribution of changed variable on model output over others

* SHAP Shapley Additive exPlanations
game theoretic approach + local explanations
SHAP = f(LIME, Shapley sampling, QII, DeepLIFT, Layerwise Regr propagaⁿ)
→ Tree Explainer
→ Deep
→ Gradient
→ Kernel

* PDP - Expected target response wrt x keeping all else constant

* ALE - small window, handles corrⁿ

* LIME Local Interpretable Mod-agnostic Exp
local surrogate linear model on perterbed data
Local fidelity ⇒ ↑ local fit (loss has complexity penalty)
typically uses exponential smoothening & lasso.

* Anchors - Lime with if-else (nonlinearity to handle unseen test instances)

Why explainability needed?
→ Enemy vs Friendly tank
→ Husky vs Wolf (snow)

Others - Goldeneye, PALM, CAM, Grad-CAM, MES

* Not driver analysis but wt Δinput; pushes record; across decision bondʸ

CF explanations
→ Validity set up as optim problem, dist based
→ Actionability Only change continuous, bias in catᵍ
→ Sparsity Penalty that ↑ sparsity in diff of CF & data
→ Data Manifolds Penalty that ↑ adherence to manifold
→ Causality Retain causal relations
→ Amortized Inference opti ∀x expensive ⇒ generative technique to create CFs
→ Speed for linear/piecewise can use LPP

Similar terms -
→ Recourse (CFs better)
→ Inverse classification ~
→ Contrastive explanation ! ~
→ Adversarial Learning
(CFs more parsimonious
Advs can hv ↑ confidence misclassification)

Validation - ① validity ② Proximity ③ Sparsity ④ Diversity ⑤ Close to train ⑥ Causal IM1 and IM2

① Proportion of CFs actually going across decision boundary
② CF vs record Minkowski or Mahalanobis
③ ↓ feat. & ↓ change
④ atleast 1 CF is useful
⑤ avg dist to k neighbors local outlier factor or reconstruction error wrt VAE (train)
⑥ auto encoders built class wise & check ~ reconstrⁿ errors

Research Ppr 2 - Model agnostic Counterfactual Exp using RL June 2021
Replace expensive opti with end-to-end learnable RL
+ Model Agnostic + target conditional CF instances + Num & Cat while preserving immutability
• should be sparse, indistribution, allow feature conditioning for ↑ wrt actionability

Many Opti → Fewer Opti (Batch level) → Reconstruction losses via AEs → Utilize pre-trained GANs or VAEs to skip opti
Other ways - LORE ↘ class cond^n generative
① Local Surrogate model around instance
② Train on syn. data created by using GA
③ Decision rules from local wan change model prediction leveraged for counterfactual

DiCE
uses determinantal point processes
→ expects diverse CFEs joint intensity equates to determinant of a function
Wud DiCE be overfitted?

Even if u do class conditional generative model for batches of CFEs, this isn't feasible for non-differential models as backprop of gradients needed

RL - usage: Identify optimal Q function (agent reward)
Deep Deterministic Policy Gradient bypasses computation by some approx. (OpenAI ppr)

Say x = instance
M = black box model
$y_M = M(x)$ prediction
$x_{CF} = x + \delta_{CF}$ → sparse perturbation
↳ Counterfactual instance
• optimal conditioning vector c
GA models x_{CF} allowing for c
Generative

x_{CF} depends upon $(x, y_M, y_T, c) = s$ (state) & $\mu(s)$ is actor network ?
Reward classify = 1
Reward regression ∝ proximity
* uses autoencoder help
* perturbations in latent space
* X shift & scale invariant then sampled from Beta(2,2)
IIY categorical from Bern(0.5) [feature conditioning ++]

Baselines for comparison - Model Agnostic (LORE)
? Gradient-free (MO)
- Tabular-oriented (DiCE) → random, → genetic
* DiCE with random perturbations ⇒ better validity, smaller proximity L_0, comparable L_1, worse in-distributionness

Pros:
Better in-distributionness (MMD)
Better transfer, scale aspects
Lesser time compute
Easier feature conditioning

Cons:
No need for smaller problems given DiCE is almost at-par