Counter Factual Explanations

Explainable Artificial Intelligence

A. Caley, L. Conran, N. Cannon, W. H. Lima

CITS4404 Project Assignment

October 2020

Table of Contents

Table of Contents

Explainable AI or 'XAI'

Complex Machine Learning models are making predictions with ever increasing accuracy and complexity.

- Support good decision making processes (confidence in the Model)
- Comply with legal and regulatory requirements (industry regulations)
- Predictions can have real life and important impact

Using Explainable AI or 'XAI' can:

- uncover correlations and patterns in the data
- assist in Model de-bugging

Table of Contents

Counterfactual Explanations

"A counterfactual explanation of a prediction describes the smallest change to the feature values that changes the prediction to a predefined output." - Christoph Molnar

- 'What if' explanations, or comparable with cause-and-effect relationships.
- How the world would have to change to achieve a new prediction
- 'If X had occurred, Y would not'
- Multiple truths ("Rashomon effect")

Table of Contents

Datasets used

- 'Breast cancer wisconsin dataset (classification)' available from UCI Machine Learning repository. Numeric dataset with 569 instances of 30 numerical features. Benign / Malignant.
- 'Adult dataset' available from UCI Machine Learning repository. It has 8 features (categorical and numeric). Labelled to indicate whether that Adult's income above or below USD50k.
- 'Fashion MNIST' available at https://github.com/zalandoresearch/fashion-mnist. Consisting of 70,000 greyscale images of clothing articles, with 10 classes.

Table of Contents

Modelling

Selected methods of producing Counterfactual explanations:

- Alibi - diverse library of methods which includes a counterfactual explanation method
- DiCE - Diverse Counterfactual Explanations
- CEML - Counterfactuals for Explaining Machine Learning models

We have applied these methods to the predictions of neural networks and examined the results.

Alibi

Alibi is a Python package designed to help explain the predictions of machine learning models and gauge the confidence of predictions.

- Takes inspiration from scikit-learn (initialize, fit and explain steps)
  - import the explainer
  - initialize passing a prediction function
  - fit the explainer (if necessary)
  - call explainer
- Returns Explanation object has meta and data attributes which are dictionaries containing any explanation metadata (e.g. parameters, type of explanation) and the explanation itself respectively

DiCE

- DiCE implements counterfactual (CF) explanations by showing feature-perturbed versions of the same case, comparable to 'what-if' explanations for model output.
- The core idea is to work as an optimization problem, similar to finding adversarial examples.
  - import helper functions
  - load dataset form CF Explanations
  - load pre-trained ML model
  - Run explanation instance
- Returns generated counterfactual examples based on a query list
- Visualize counterfactual explanation in a friendly table

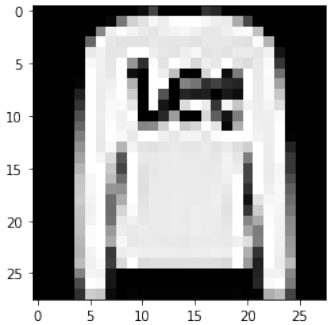
CEML

- CEML is a Python toolbox for computing counterfactuals.
- must specify the model, the input whose prediction we want to explain and the requested target prediction (prediction of the counterfactual).
  - Load data
  - Whitelist of features (list of features we can change/use when computing a counterfactual)
  - Create and fit model
  - Select data point for explaining its prediction
  - Compute counterfactual
- Returns the counterfactual, its prediction and the needed changes to the input as dictionary

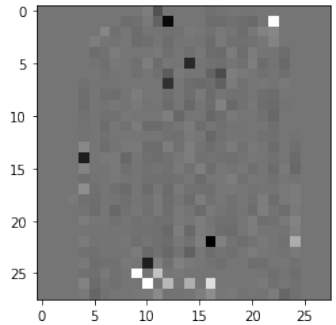
Table of Contents

Results - Alibi

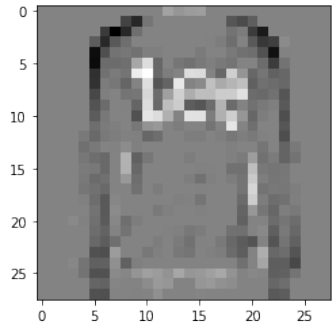
- Convolutional Neural Network ('CNN') on the Fashion MNIST and applied the Alibi package. The CNN predicts class 'Shirt'.



(a) Predicted 'Shirt'



(b) Without prototype



(c) With prototype

Figure: Alibi - Counterfactual Explanations

- non-prototype highlights the pixels that would need to change intensity in order to flip the predicted class.
- with prototype first creates a 'prototype' for each class using an autoencoder.

Results - DiCE

- Trained on the Adult Income dataset, we 'query' a specific instance in the data.
  - The model predicts income <USD50k

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	30.0	Private	HS-grad	Single	White-Collar	White	Male	45.0	0.067473

Figure: An example of instance that has been 'queried'

  - It generates five counterfactual examples.

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	88.0	Private	Doctorate	Single	White-Collar	White	Male	45.0	0.766
1	39.0	Private	Doctorate	Married	White-Collar	White	Male	40.0	0.935
2	42.0	Self-Employed	HS-grad	Married	White-Collar	White	Female	45.0	0.747
3	44.0	Private	Prof-school	Single	Professional	Other	Male	45.0	0.608
4	52.0	Private	HS-grad	Married	White-Collar	White	Male	65.0	0.807

Figure: Five counterfactual explanations for the prediction above

  - DiCE suggests some counterfactuals that are impossible to achieve

Results - CEML

- Worked well for simple models (Decision Trees, small Random Forests) on breast cancer dataset
  - Results hard to interpret, requires extra effort to make sense of
- Unable to find any counterfactuals for complex Neural Network models
  - Long time to compute ($>$ 1000 sec)
  - Poor results possibly due to unsuitable loss function and optimization technique for the problem
- Ran experiment to explore the relationship between model complexity and CEML counterfactual compute time

CEML Random Forest Results

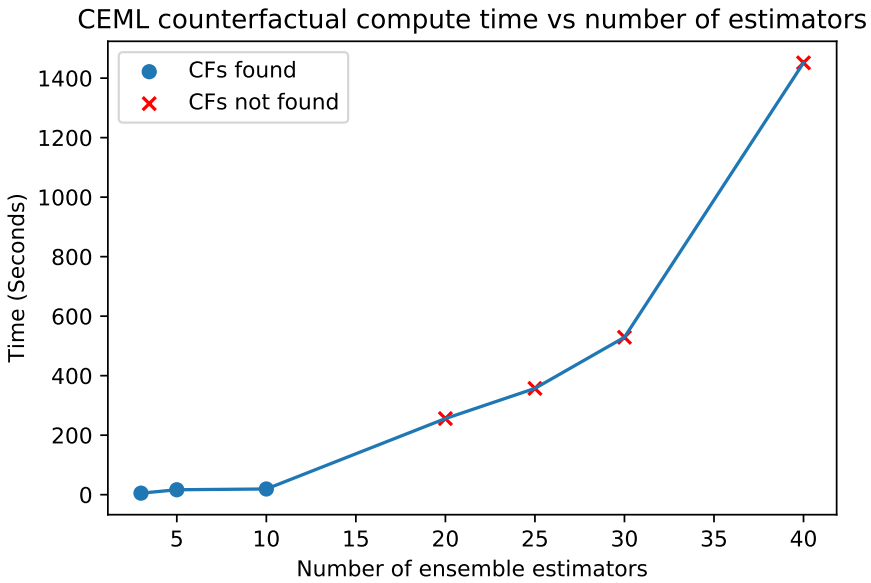


Figure: CEML Random Forest counterfactual computation time vs complexity.

Table of Contents

Conclusion

- Alibi - Well optimised with performance provided by class prototyping method. Simple, but inflexible.
- DiCE - Simple to implement Produces clean tabular output.
- CEML - Good for simple models. Not for complex DNNs.