Counterfactual Explanations of (some) ML Models

SAPIENZA
UNIVERSITÀ DI ROMA

Fabrizio Silvestri

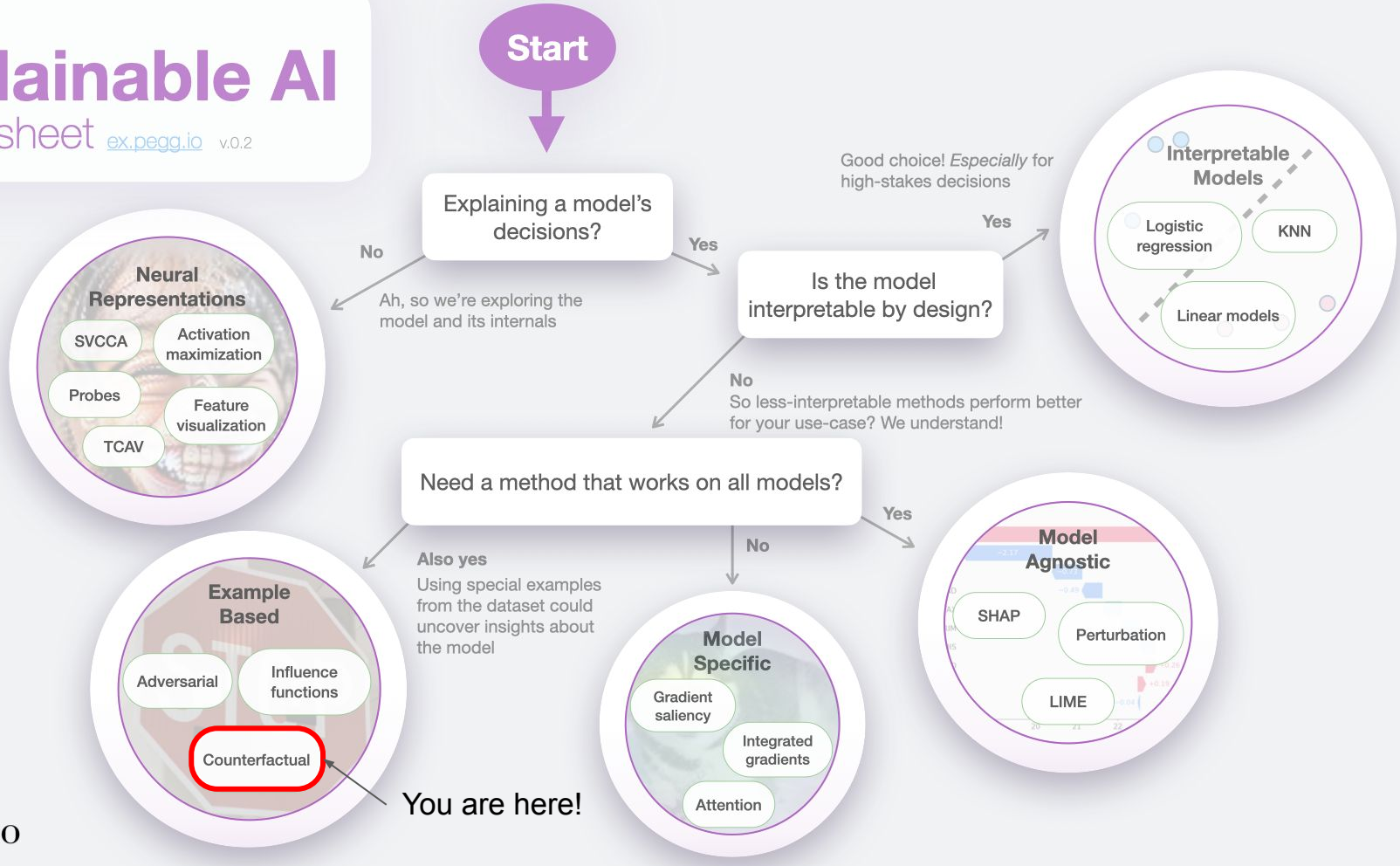
Introduction

Explainable AI (XAI)

https://medium.com/@BonsaiAI/what-do-we-want-from-explainable-ai-5ed12cb36c07

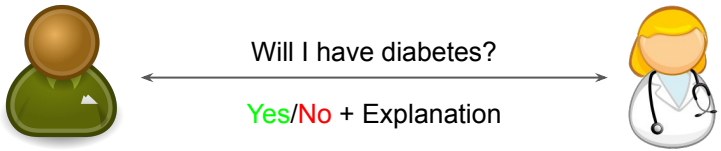
Explainable AI

Cheat sheet ex.pegg.io v.0.2



Arpeggio

Counterfactual Explanations



Factual

Age	Gender	Exercise Level	Fat Level
45	M	Low	High

f( ) = Yes

Counterfactual

Age	Gender	Exercise Level	Fat Level
45	M	**Medium**	**Average**

f( ) = No

Explanation: You will not develop diabetes if you ***lower your fat level*** and you ***increase your exercise level***.

SAPIENZA
UNIVERSITÀ DI ROMA

Agenda

- Counterfactual Explanations for Random Forest Classifiers
- Counterfactual Explanations for Node Classifiers in GNNs

  - Motivations
  - Problem definition
  - Our solution
  - Experiments

SAPIENZA
UNIVERSITÀ DI ROMA

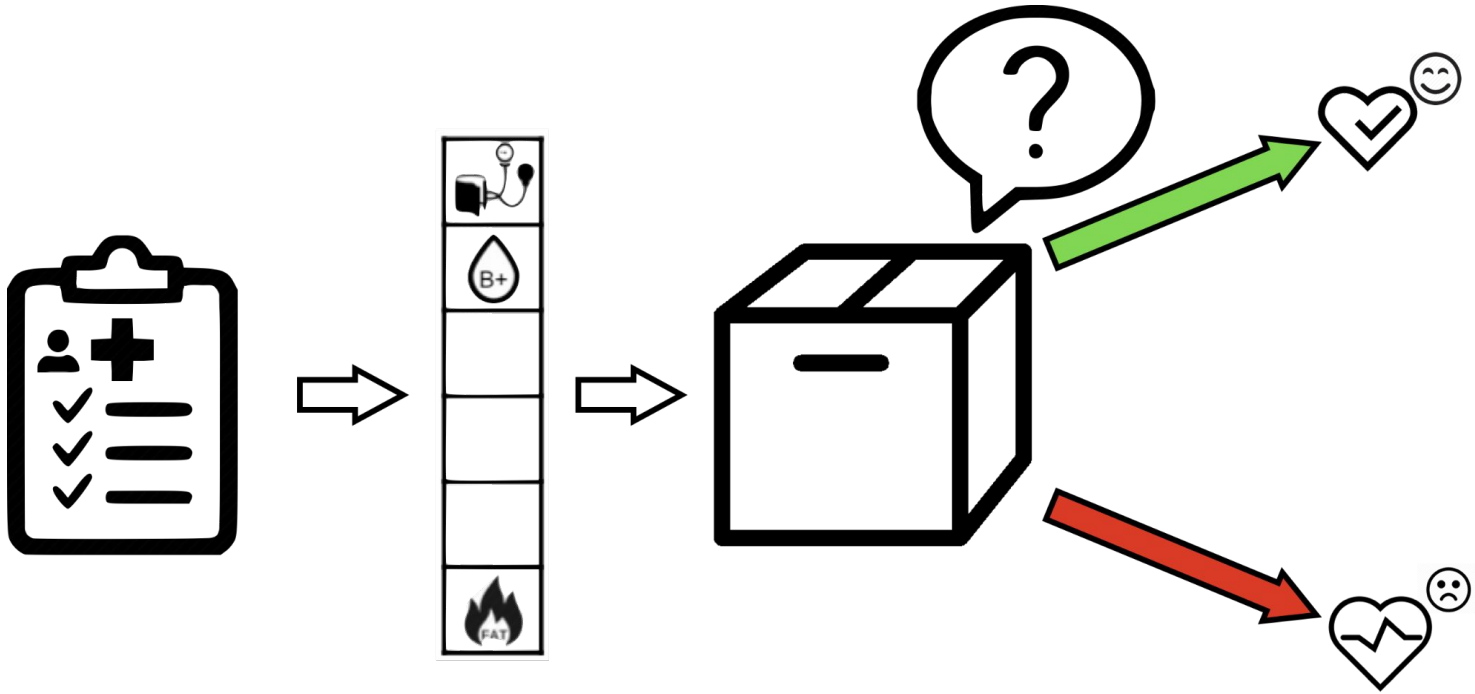
Slides mostly based on:
Tolomei et al. Interpretable predictions of tree-based ensembles via actionable feature tweaking. *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017.

Patent Application:
Tolomei, G., Haines, A., Lalmas, M., & Silvestri, F. (2018). *U.S. Patent Application No. 15/444,912*.

Generating Actionable Interpretations from Ensembles of Decision Trees

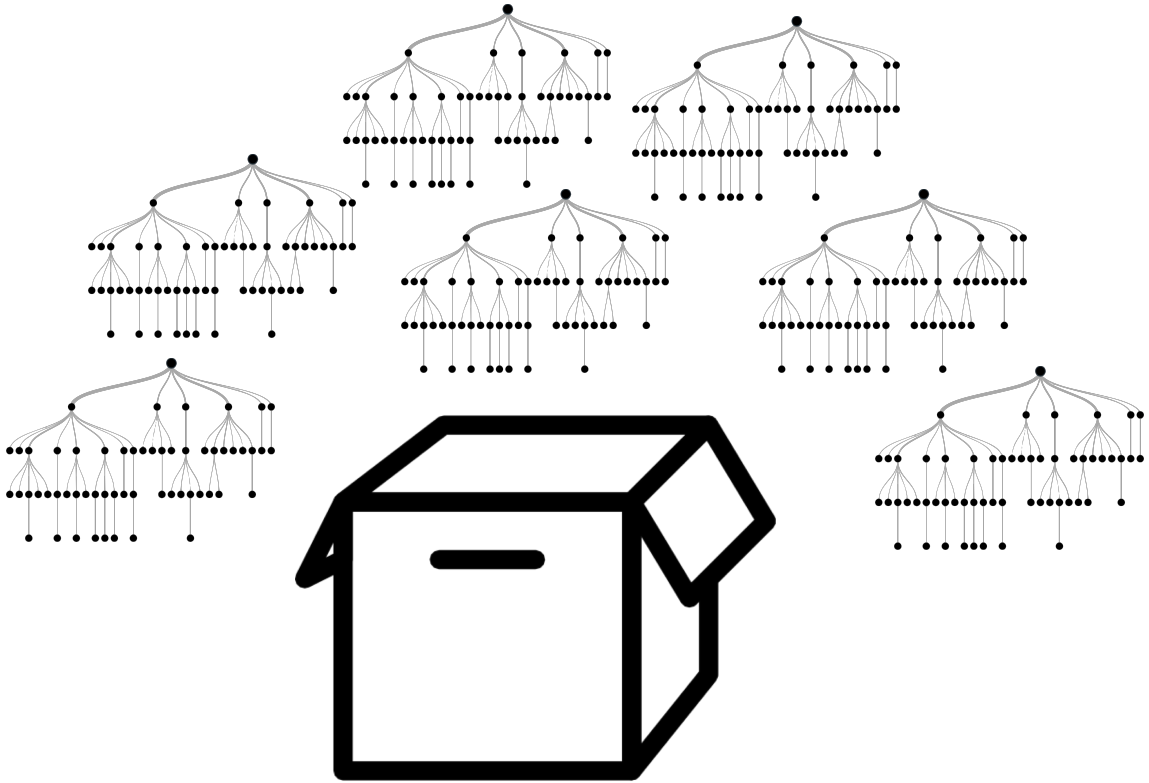
Tolomei & Silvestri. Generating Actionable Interpretations from Ensembles of Decision Trees. *IEEE Transactions on Knowledge and Data Engineering*. 2021.

Predictive Models as Black Boxes



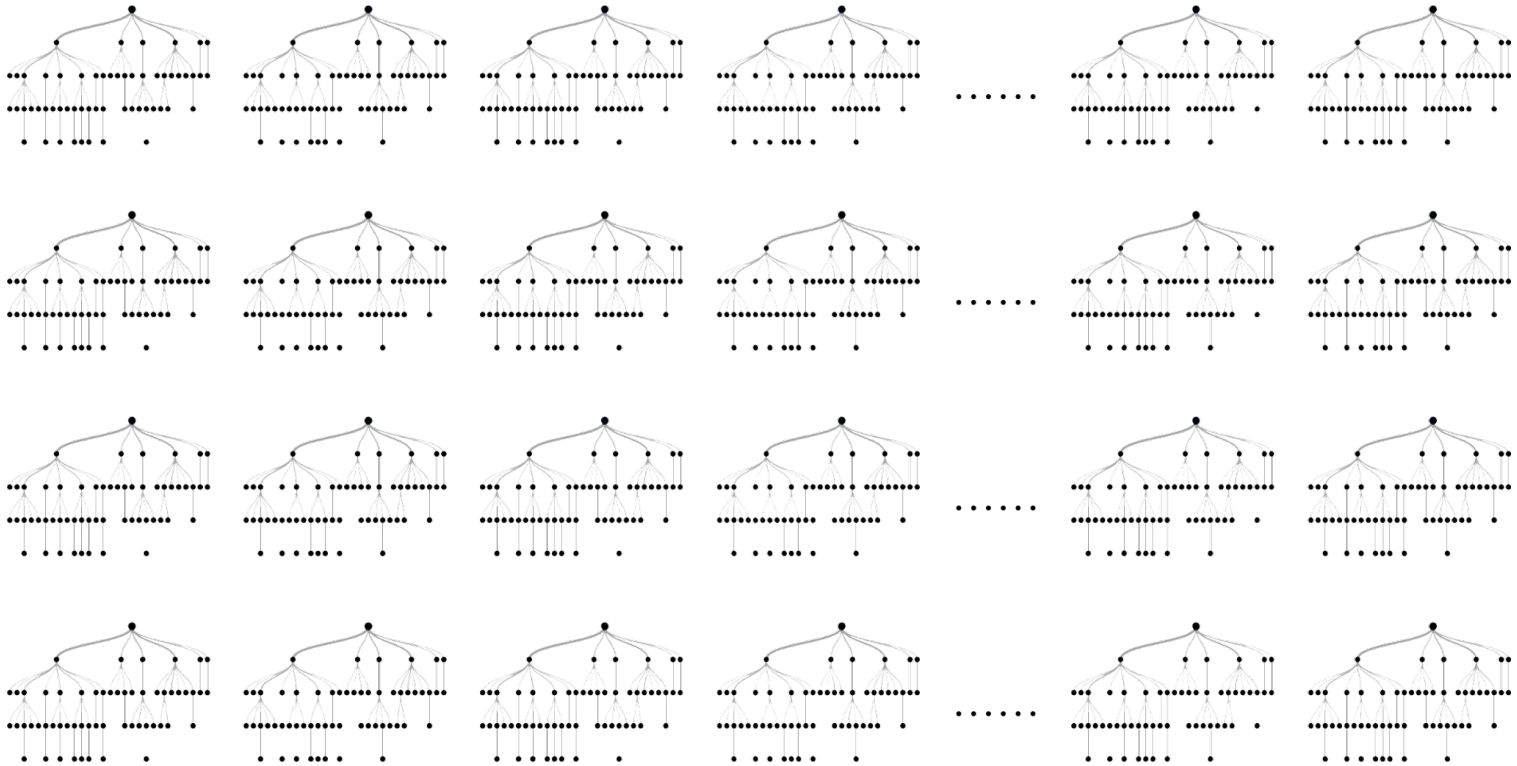
SAPIENZA
UNIVERSITÀ DI ROMA

Let’s Open the Black Box



SAPIENZA
UNIVERSITÀ DI ROMA

Let’s Open the Black Box



SAPIENZA
UNIVERSITÀ DI ROMA

Preliminaries

- We focus on classification problems.
- Let $\mathbf{X} \subseteq R_n$ be an n-dimensional vector space of real-valued features.
- $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ represents any objects we want to classify as a vector in $\mathbf{X}$.
- Each $\mathbf{x}$ is associated with a binary class label {-1, +1}.

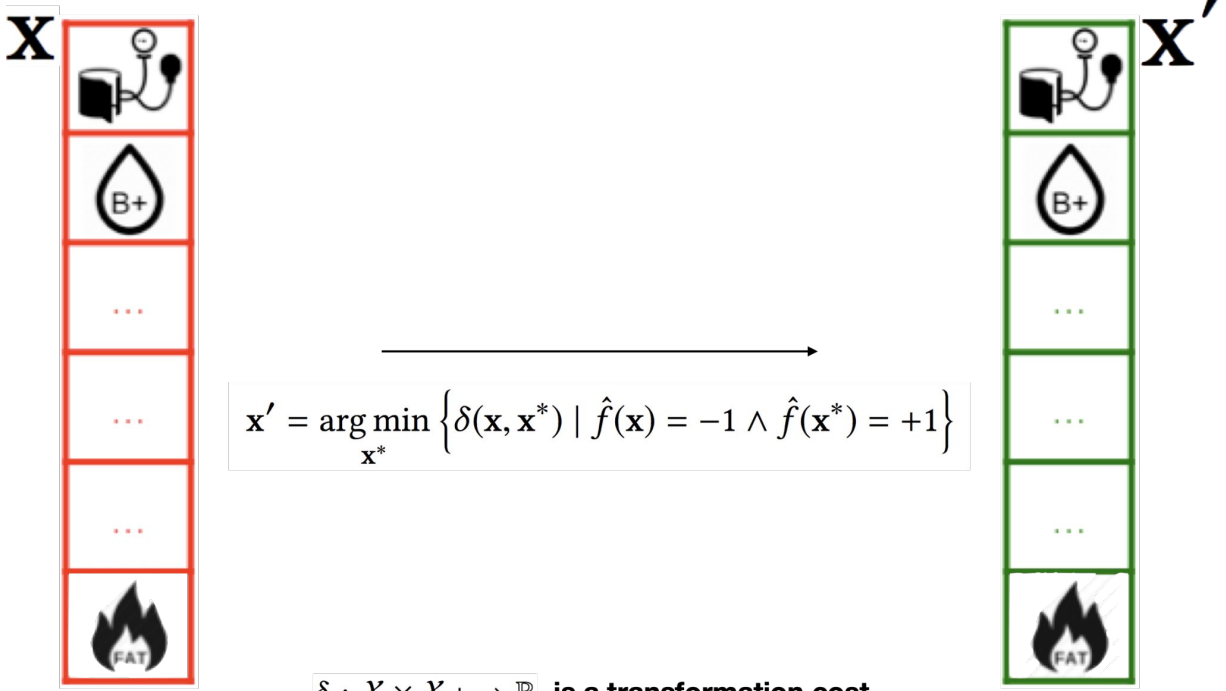
SAPIENZA
UNIVERSITÀ DI ROMA

Preliminaries

- We are given a learnt function f: $X \rightarrow Y$ that we assume to be represented as an ensemble of K trees: $f = \phi(h_1, h_2, ..., h_K)$
  - each h_i: $X \rightarrow Y$ is a base learner estimated on some samples of X.
- We assume the final predicted label is given by a majority voting approach, i.e., $\phi = \text{sign}(\text{sum}(h_i))$.

SAPIENZA
UNIVERSITÀ DI ROMA

Problem Definition

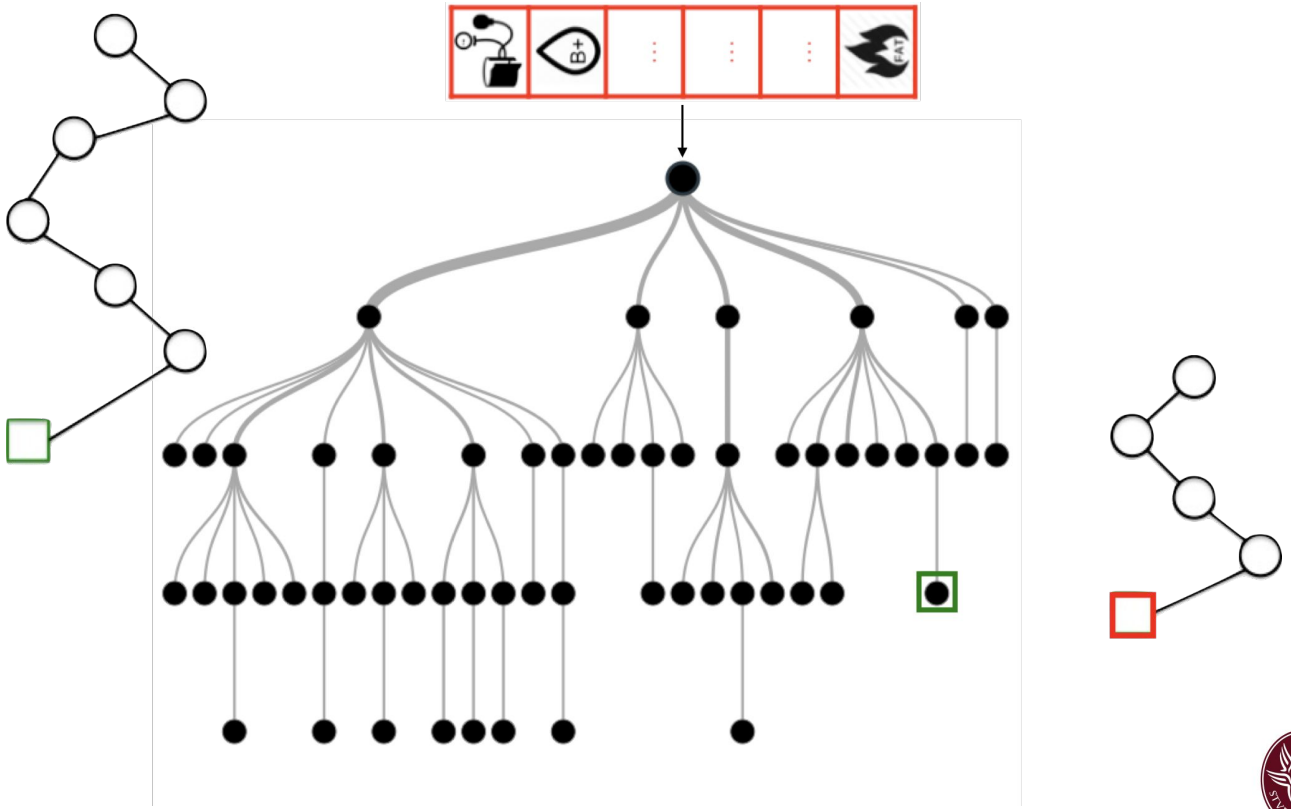


$$\mathbf{x}' = \arg\min_{\mathbf{x}^*} \left\{ \delta(\mathbf{x}, \mathbf{x}^*) \mid \hat{f}(\mathbf{x}) = -1 \wedge \hat{f}(\mathbf{x}^*) = +1 \right\}$$

$\delta : \mathcal{X} \times \mathcal{X} \longmapsto \mathbb{R}$ **is a transformation cost function measuring the effort necessary to go from x to x'.**

SAPIENZA
UNIVERSITÀ DI ROMA

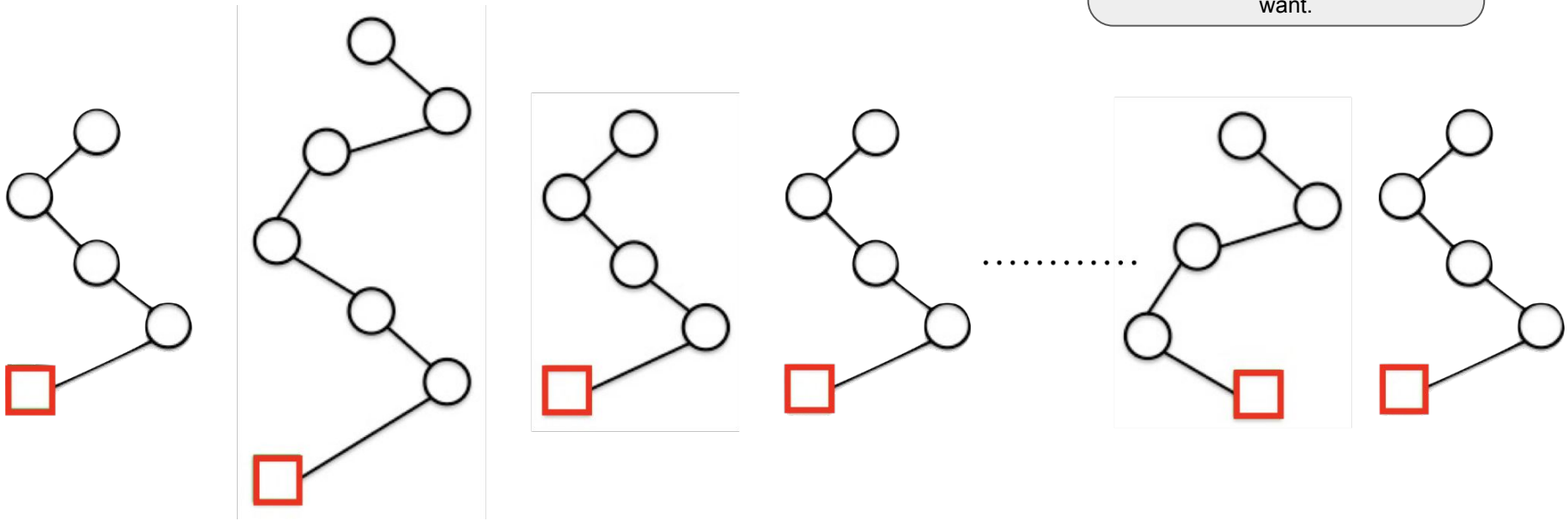
Positive/Negative Paths



SAPIENZA
UNIVERSITÀ DI ROMA

Positive/Negative Paths

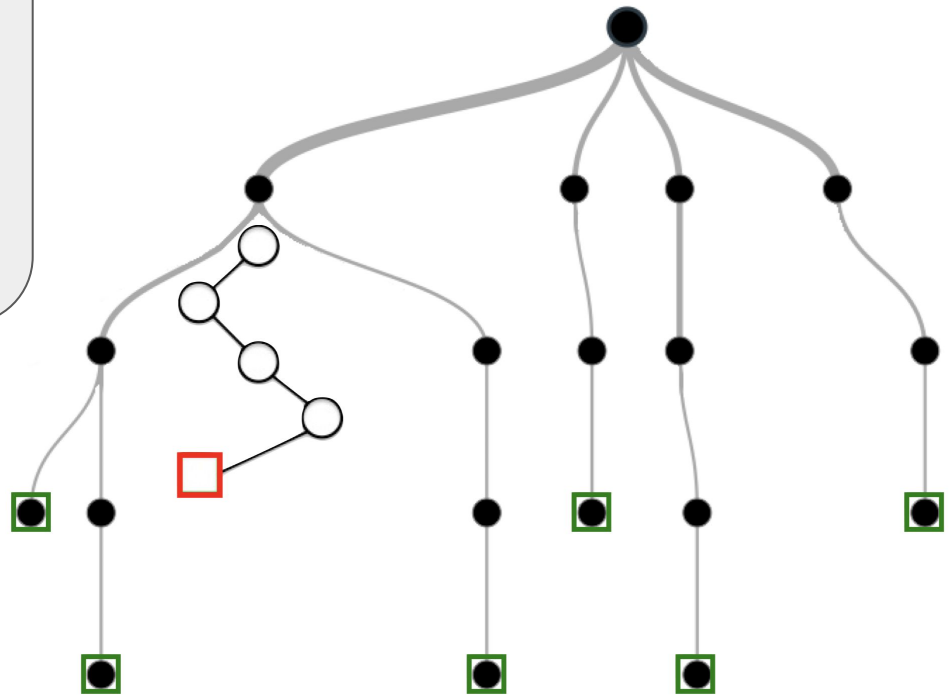
We can skip all the trees containing the positive paths, as these are already encoding the (positive) prediction we ultimately want.



SAPIENZA
UNIVERSITÀ DI ROMA

Positive/Negative Paths

- We consider the set of positive paths in the tree T.
- We modify a negative instance in order to go through a positive path in T.
- Each feature is modified by at most ε.
- An instance meeting the ε constraint and satisfying the tree is called: ε-satisfactory.



SAPIENZA
UNIVERSITÀ DI ROMA

Building ε-satisfactory Instances

- Let us consider a positive path as represented by the conditions from root to leaves

$$p_{k,j} = \{(x_1 \lesseqgtr \theta_1), (x_2 \lesseqgtr \theta_2), \ldots, (x_n \lesseqgtr \theta_n)\}$$

- for any (small) fixed ε > 0, we build a positive feature vector x+ dimension-by-dimension as follows:

$$\begin{cases} \theta_i - \epsilon & \text{if the } i\text{-th condition is } (x_i \leq \theta_i) \\ \theta_i + \epsilon & \text{if the } i\text{-th condition is } (x_i > \theta_i) \end{cases}$$

SAPIENZA
UNIVERSITÀ DI ROMA

Finding a Valid Tweaking

- For each positive path in our model we transform our input feature vector **x** into the ε-satisfactory instance for that path.
- This leads us to a set of tweakings Γ_K associated with each tree T_k. The set of all tweakings is then given by $\Gamma = \cup \Gamma_K$
- The problem of feature tweaking then becomes that of finding:

$$\mathbf{x}' = \underset{\mathbf{x}^+ \in \Gamma_K}{\operatorname{argmin}} \{\delta(\mathbf{x}, \mathbf{x}^+)\}$$

NP-Hard. Reduction to DNF-MAXSAT

- Solution based on Spatial Indexes for logarithmic-time search

SAPIENZA UNIVERSITÀ DI ROMA

Experiments

- Random sample of 1,500 ads out of those served on mobile app by Yahoo Gemini during one month
- 50÷50 positive (high quality) vs. negative (low quality) instances using median dwell time (62.5 secs.) for labelling
- 80÷20 training/test splitting for learning a binary classifier (Decision Tree, GBDT, Random Forest)
- 10-fold cross validation on the training portion to select the best hyper-parameters for each model
- Eventually, the best-performing model on the test set (offline) is Random Forest with 1,000 trees and maximum depth = 16

Ad Quality Experiments

- A use case coming from a real need while working at Yahoo's Gemini Project
- How to suggest advertisers features to modify in order to have their ads being classified as "good-looking"?
- Classifier described in:
  - Barbieri, Silvestri, Lalmas: *Improving Post-Click User Engagement on Native Ads via Survival Analysis*. WWW 2016

SAPIENZA
UNIVERSITÀ DI ROMA

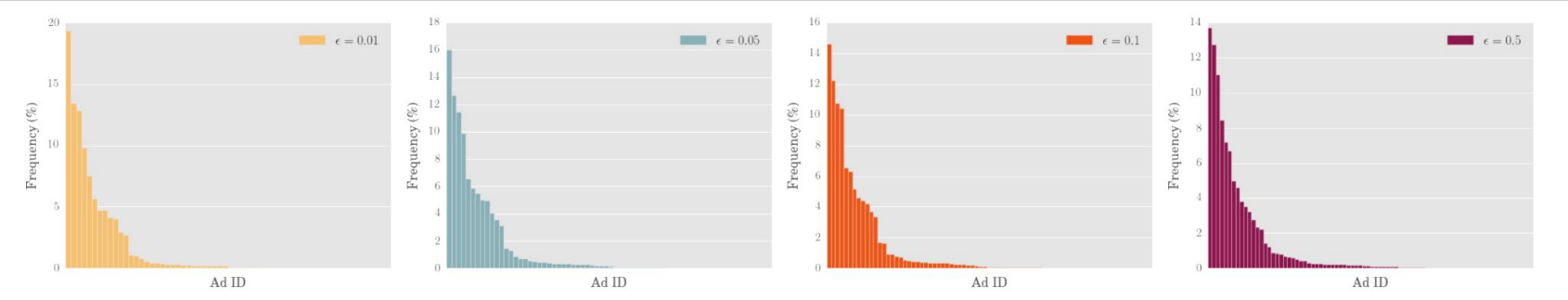
Ads Features

Category	Source	Description
Language†	CR	This set of features capture the extent to which the text of the ad creative may include adult, violent, or spam content (*e.g.*, ADULT_SCORE, HATE_SCORE, and SPAM_SCORE)
DOM	LP	This set of features are derived from the elements extracted from the HTML DOM of the ad landing page, such as the main textual content (LANDING_MAIN_TEXT_LENGTH), the total number of internal and external hyperlinks (LINKS_TOTAL_COUNT), the ratio of main text length to the total number of hyperlinks on the page (LINKS_MAIN_LENGTH_TOTAL_RATIO), *etc.*
Readability	CR-LP	These features range from a simple count of tokens (words) in the text of the ad creative and landing page to well-known scores for measuring the summarisability/readability of a text (*e.g.*, READABILITY_SUMMARY_SCORE), *etc.*
Mobile Optimising	LP	This set of features describe the degree of mobile optimisation of the ad landing page by measuring the ability of it to be tuned to different screen sizes (VIEW_PORT), testing for the presence of a click-to-call button (CLICK_TO_CALL), *etc.*
Media	LP	These features refer to any media content displayed within the ad landing page, such as the number of images (NUM_IMAGES), *etc.*
Input	LP	This set of features represent all the possible input types available on the ad landing page, such as the number of checkboxes, drop-down menus, and radio buttons (NUM_INPUT_CHECKBOX, NUM_INPUT_DROPDOWN, NUM_INPUT_RADIO), *etc.*
Content & Similarity	CR-LP	These features extract the set of Wikipedia *entities* from the ad creative and landing page (NUM_CONCEPT_ANNOTATION), and measure the *Jaccard* similarity between those two sets (SIMILARITY_WIKI_IDS), *etc.*
History	LP	These features measure historical indicators, such as the median *dwell time* as computed from the last 28 days of observed ad clicks (HISTORICAL_DWELLTIME), and the *bounce rate* – *i.e.*, the proportion of ad clicks whose dwell time is below 5 seconds (HISTORICAL_BOUNCE_RATE), *etc.*

SAPIENZA
UNIVERSITÀ DI ROMA

Impact of ε

ε	0,01	0,05	0,10	0,50	1,00
ad coverage (%)	58,5	64,2	72,3	**77,4**	63,2

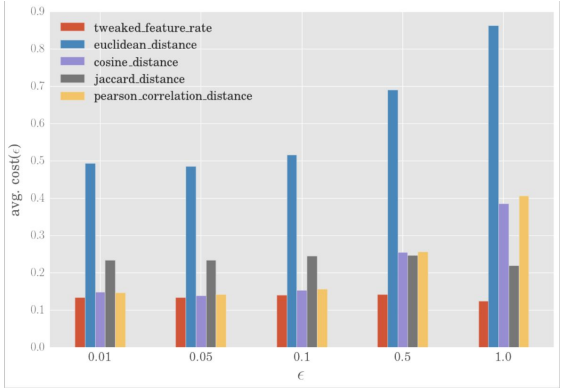


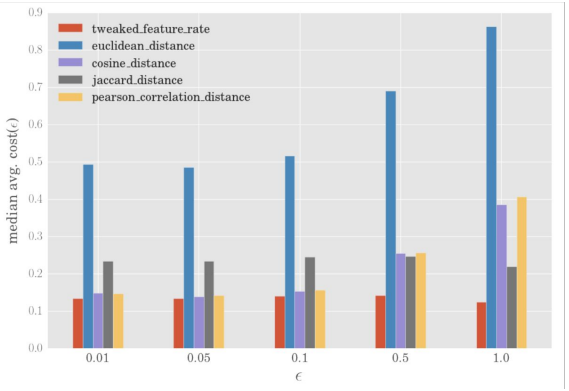
Distribution of per-ad ε-transformations

SAPIENZA
UNIVERSITÀ DI ROMA

Impact of ε on cost δ

- tweaked feature rate:
  - proportion of features affected by the transformation of x into x' (range = [0, 1]);
- euclidean distance:
  - euclidean distance between x and x' (range = R);
- cosine distance:
  - 1 minus the cosine of the angle between x and x' (range = [0, 2]);
- jaccard distance:
  - one's complement of the Jaccard similarity between x and x' (range = [0, 1]);
- pearson correlation distance:
  - 1 minus the Pearson's correlation coefficient between x and x' (range = [0, 2]).





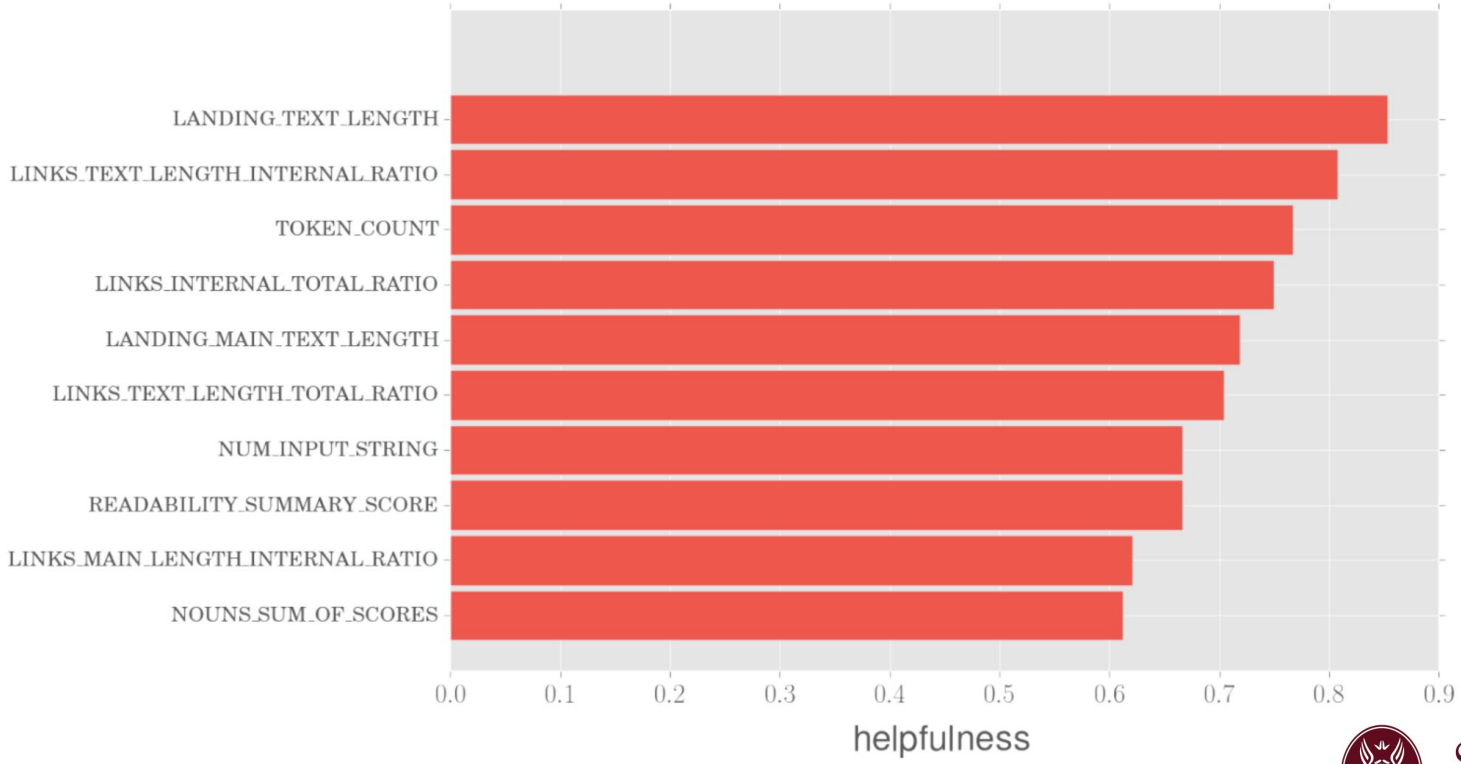
SAPIENZA
UNIVERSITÀ DI ROMA

Qualitative Evaluation

- We asked a team of creative strategist to evaluate tweakings for 100 randomly selected ads that our classifier scored -1 (Bad Ads). Ratings:
  - *helpful*, *non-helpful*, or *non-actionable*
- 57.3% ↦ helpful (inter-agreement: 60.4%)
- 0.4% ↦ non-actionable suggestion.
- about 25% of the 42.3% non-helpful recommendations were considered “neutral”:
  - not hurting the user experience if discarded and not adding any positive value.

Features Helpfulness

$$\text{helpfulness}(i) = \frac{|\text{helpful}(i)|}{|\text{helpful}(i)| + |\neg\text{helpful}(i)|}$$

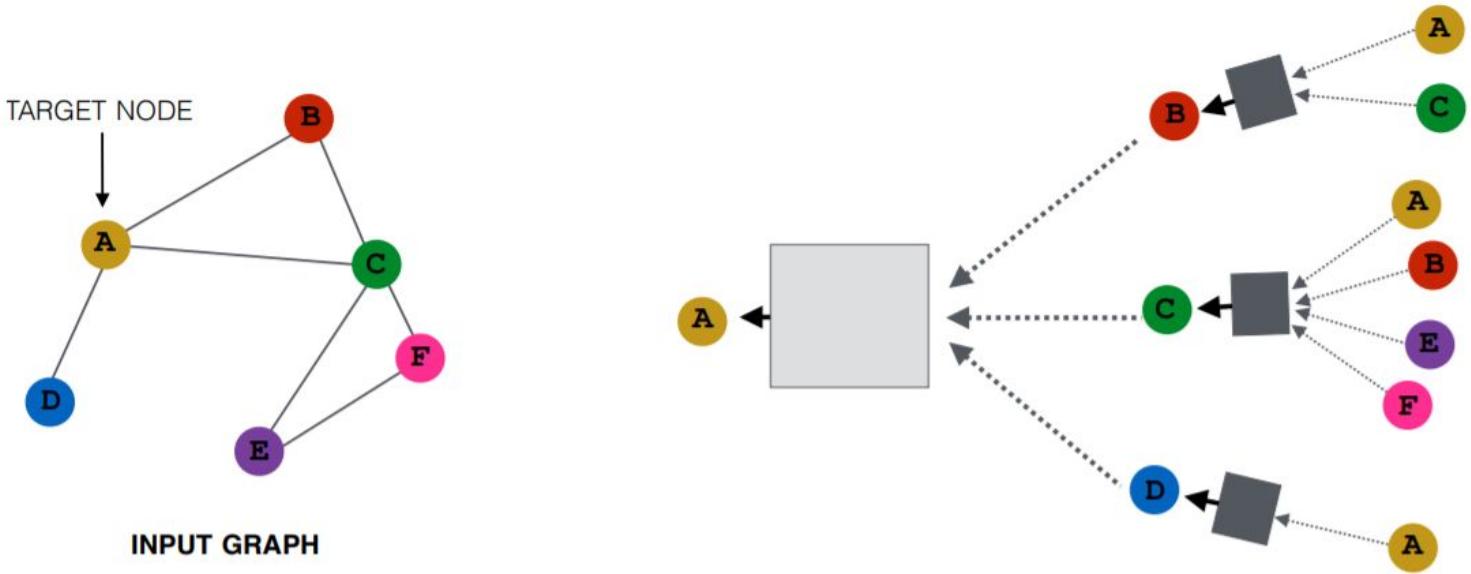


SAPIENZA
UNIVERSITÀ DI ROMA

CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks

Lucic, A., ter Hoeve, M., Tolomei, G., de Rijke, M., & Silvestri, F. (2021). CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks. arXiv preprint arXiv:2102.03322.

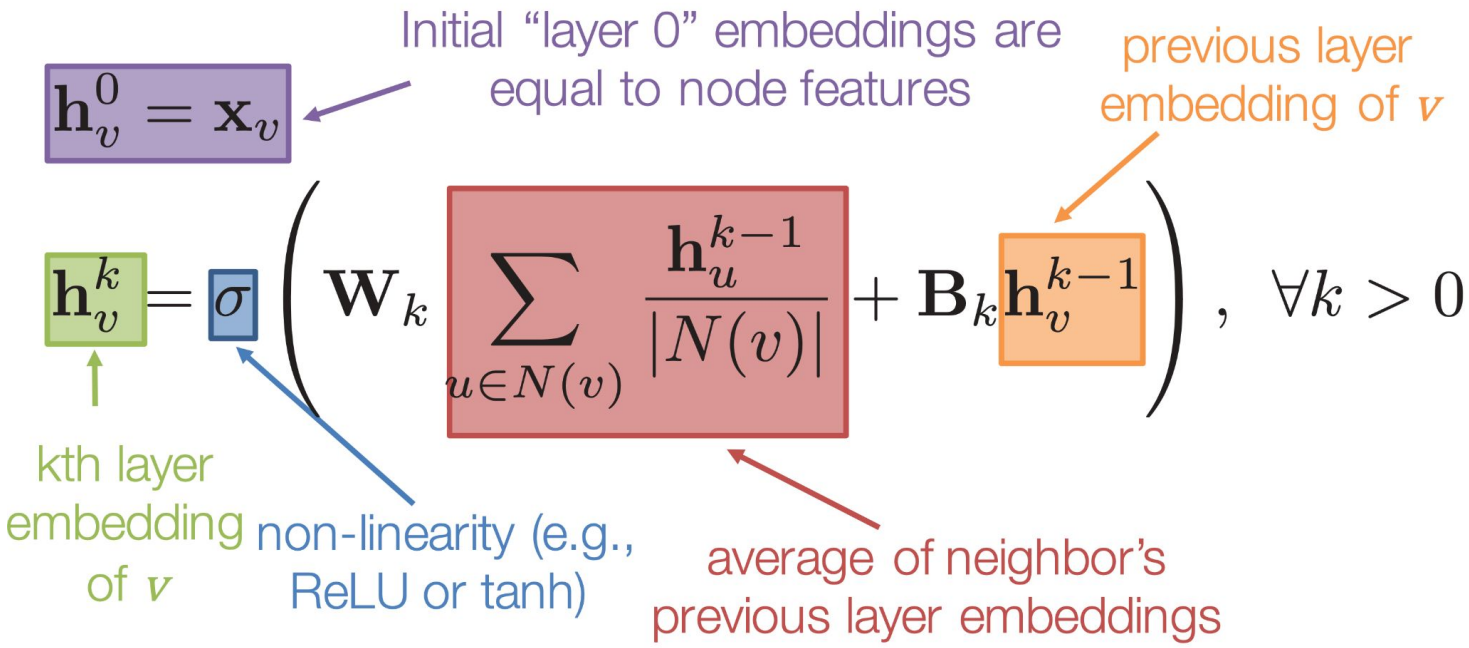
Motivations: Graph Neural Networks



Curtesy of https://heartbeat.fritz.ai/introduction-to-graph-neural-networks-c5a9f4aa9e99

SAPIENZA
UNIVERSITÀ DI ROMA

Graph Neural Networks for Node Classification



Curtesy of http://snap.stanford.edu/proj/embeddings-www/files/nrltutorial-part2-gnns.pdf

SAPIENZA
UNIVERSITÀ DI ROMA

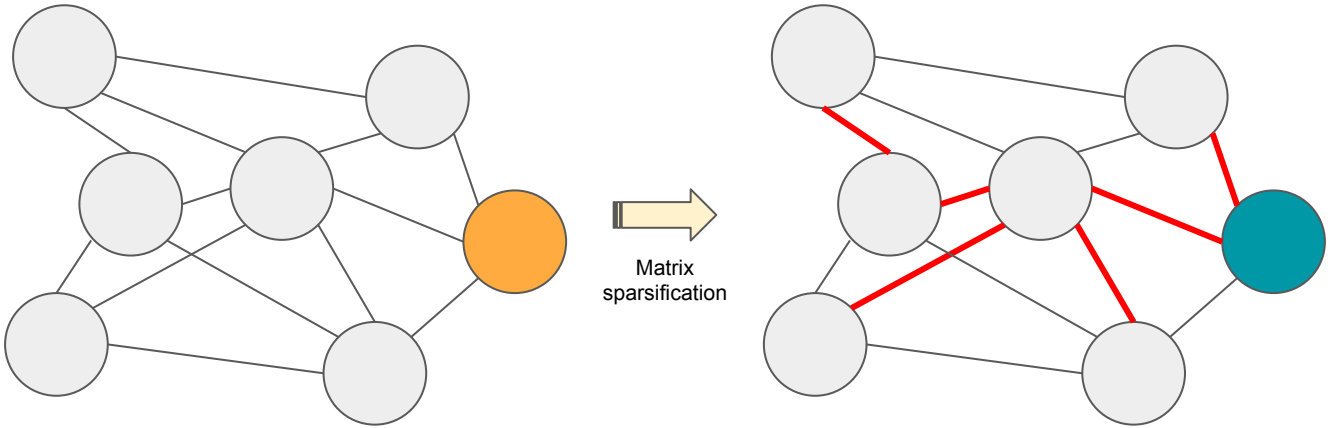
CF-Example for Graph NNs

- A CF example x* for an instance x according to a trained classifier f is found by perturbing the features of x such that f(x*) != f(x)
  - An optimal CF example x^o is one that minimizes the distance between the original instance and the CF example, according to some distance function d, and the resulting optimal CF explanation is $\Delta x^o = x^* - x$
- For graph data, it may not be enough to simply perturb node features, especially since they are not always available.

SAPIENZA
UNIVERSITÀ DI ROMA

Counterfactual Explanations for Node Prediction

- The goal is to find the smallest subgraph that if removed the model will predict a different class for a node.



$$\mathcal{L} = \mathcal{L}_{pred}(v, \bar{v} \mid f, g) + \beta \mathcal{L}_{dist}(v, \bar{v})$$

original

Counterfactual model

SAPIENZA
UNIVERSITÀ DI ROMA

Adjacency Matrix Perturbation

- For a node v, we consider $G_v=(A_v, X_v)$
  - A_v is the restriction of A (the adjacency matrix) to the l-hop neighborhood of v (X_v) is the corresponding edge set.
- $A_v^* = P \odot A_v$
  - P is a perturbation matrix
- To generate P we first generate an intermediate real-valued P* which we threshold in order to get P*

SAPIENZA
UNIVERSITÀ DI ROMA

An example for GCN

- We start from the traditional GCN formulation

$$f(A, X; W) = \text{softmax}\left[\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}XW\right]$$

- We modify it to explicitly isolate A_v

$$f(A_v, X_v; W) = \text{softmax}\left[(D_v + I)^{-1/2}(A_v + I)(D_v + I)^{-1/2}X_vW\right]$$

- We define a new model g depending on P

$$g(A_v, X_v, W; P) = \text{softmax}\left[\bar{D_v}^{-1/2}(P \odot A_v + I)\bar{D_v}^{-1/2}X_vW\right]$$

SAPIENZA
UNIVERSITÀ DI ROMA

CF-GNNExplainer

Algorithm 1 CF-GNNEXPLAINER: given a node $v = (A_v, x)$ where $f(v) = y$, generate the minimal perturbation, $\bar{v} = (\bar{A}_v, x)$, such that $f(\bar{v}) \neq y$.

Input: node $v = (x, A_v)$, trained GNN model f, CF model g, loss function $\mathcal{L}$, learning rate α, trade-off parameter β, number of iterations K, distance function d.

$f(v) = y$ *# Get GNN prediction*
$\hat{P} \leftarrow J_n$ *# Initialization*

for $k \in range(K)$ **do**
  $\bar{v}^{(k)}$ = GET_CF_EXAMPLE()
  $\mathcal{L} \leftarrow \mathcal{L}(v, \bar{v}^{(k)})$ *# Eq 1 & Eq 5*
  $\hat{P} \leftarrow P^{(k)} + \alpha \nabla_{\hat{P}} \mathcal{L}$ *# Update* $\hat{P}$
end for

Function GET_CF_EXAMPLE()
  $P \leftarrow \text{threshold}(\sigma(\hat{P}^{(k)}))$
  $\bar{A}_v \leftarrow P \odot A_v$
  $\bar{v}^{(k)}_{cand} \leftarrow (\bar{A}_v, x)$
  if $f(v) \neq f(\bar{v}^{(k)}_{cand})$ **then**
    $\bar{v}^{(k)} \leftarrow \bar{v}^{(k)}_{cand}$
    if $\mathcal{L}_{dist}(v, \bar{v}) \leq \mathcal{L}_{dist}(v, \bar{v}^{(k)})$ **then**
      $\bar{v}^* \leftarrow \bar{v}^{(k)}$ *# Keep track of best CF*
    end if
  end if
  return $\bar{v}^*$

SAPIENZA
UNIVERSITÀ DI ROMA

Experiments

- Baselines:
  - RANDOM
  - 1-HOP
    - All edges in the ego-graph
  - RM-1HOP
    - All edges *not* in the ego-graph
  - (Modified) GNNExplainer
- Metrics
  - Fidelity
  - Explanation Size
  - Sparsity
  - Accuracy

SAPIENZA
UNIVERSITÀ DI ROMA

Datasets

- TREE-CYCLES
  - binary tree (base graph), with cycle-shaped motifs
- TREE-GRIDS
  - binary tree, with 3×3 grids as the motifs
- BA-SHAPES
  - Barabasi-Albert (BA) with house-shaped motifs, where each motif consists of 5 nodes (one for the top of the house, two in the middle, and two on the bottom).
- Task:
  - Node classification. Classes: not-in-motif, in-motif: top, middle, bottom.
- Baseline Model (f) we want to explain:
  - 3-layer GCN (hidden size = 20) for each task.
  - Our GCNs have at least 87% accuracy on the test set.

SAPIENZA
UNIVERSITÀ DI ROMA

Experiments: Results

Metric	TREE-CYCLES				TREE-GRID				BA-SHAPES			
	Fid. ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲	*Fid.* ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲	*Fid.* ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲
RANDOM	**0.00**	4.70	0.79	0.63	**0.00**	9.06	0.75	0.77	**0.00**	503.31	0.58	0.17
1HOP	0.32	15.64	0.13	0.45	0.32	29.30	0.09	0.72	0.60	504.18	0.05	0.18
RM-1HOP	0.46	2.11	0.89	—	0.61	2.27	0.92	—	0.21	10.56	0.97	**0.99**
GNNEXP	0.55	6.00	0.57	0.46	0.34	8.00	0.68	0.74	0.81	6.00	0.81	0.27
CF-GNN	0.21	**2.09**	**0.90**	**0.94**	0.07	**1.47**	**0.94**	**0.96**	0.39	**2.39**	**0.99**	0.96

SAPIENZA
UNIVERSITÀ DI ROMA

Experiments: Explanation Size

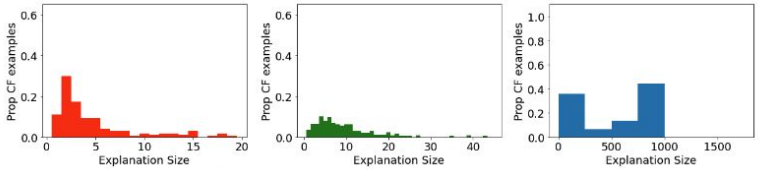


Figure 1: Histograms showing *Explanation Size* from RANDOM. Note the x-axis for BA-SHAPES goes up to 1500. Left: TREE-CYCLES, Middle: TREE-GRID, Right: BA-SHAPES.

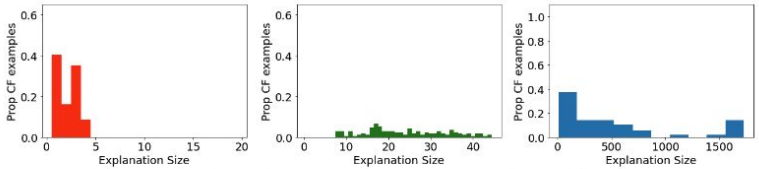


Figure 2: Histograms showing *Explanation Size* from 1HOP. Note the x-axis for BA-SHAPES goes up to 1500. Left: TREE-CYCLES, Middle: TREE-GRID, Right: BA-SHAPES.

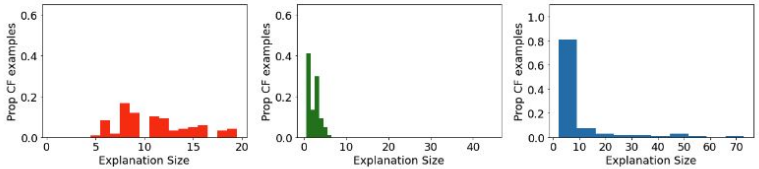


Figure 3: Histograms showing *Explanation Size* from RM-1HOP. Note the x-axis for BA-SHAPES goes up to 70. Left: TREE-CYCLES, Middle: TREE-GRID, Right: BA-SHAPES.

SAPIENZA
UNIVERSITÀ DI ROMA

Experiments: Explanation Size

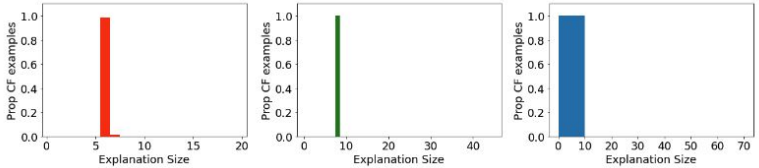


Figure 4: Histograms showing *Explanation Size* from GNNEXPLAINER. Note that the y-axis goes up to 1. Left: TREE-CYCLES, Middle: TREE-GRID, Right: BA-SHAPES.

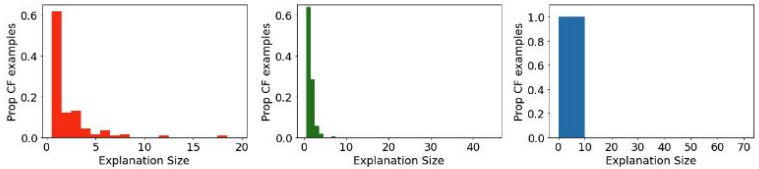


Figure 5: Histograms showing *Explanation Size* from CF-GNNEXPLAINER. Note the x-axis for BA-SHAPES goes up to 70. Left: TREE-CYCLES, Middle: TREE-GRID, Right: BA-SHAPES.

SAPIENZA
UNIVERSITÀ DI ROMA

Conclusions and Future Directions

- Explaining the decisions of black-box models is of paramount importance.
  - Why didn't you finance my mortgage?
- We presented two pioneering works in those directions:
  - The first CF explanation model for an ensemble of trees
  - The first CF explanation model for GNNs
- Future work: Can we design a generic model-agnostic mechanism to explain a black box?
  - We are working on a RL-based solution that is able to tweak features until the prediction changes
  - Using an anomaly-detection model to train a generative model to transform an input until the label changes