PNU STAT - LIME Local Interpretable Model-agnostic Explanations(12.27)

Intro

- 짧게 감잡고 가기 좋은 영상

KDD2016 paper 573
https://youtu.be/hUnRCxnydCc

- 뉴스 참고

설명 가능한 인공지능(XAI)... 왜 주목하나?

최창현 기자 | 승인 2019.12.16 07:10 | 댓글 0

> 인공지능이 판단 과정과 결과를 스스로 논리적으로 설명하는 것이 가능해진 것이다.

AI가 내린 결정이나 답을 AI 스스로가 사람이 이해하는 형태로 설명하고 제시할 수 있는 '설명 가능한 AI(eXplainable Artificial Intelligence, XAI)'가 핵심적인 비즈니스에 필수적으로 대두(사진:본지)

왜 모델 설명이 중요한가?

- ML을 실무에 적용하다 보면 만나게 되는 도전과제 → **모형이 내놓는 결과에 대한 해석!**
- 성능이 좋아도 실제 액션을 취하려면 왜 이러한 **예측과정에 대한 이해가 필수적!**
- 모델을 만드는 분석가나 배포나 운영하는 엔지니어의 입장에서도 모형이 왜 이렇게 작동하는지 이해가 반드시 필요! → **결함을 사전에 인지하고 대응하기 가능!**

논문 리뷰 - "Why Should I Trust You?" Explaining the Predictions of Any Classifier

- 2016 논문, 2133회 인용(19.12.07 시점)
  - Attention is all you need, 2017, 4866회 인용
  - BERT 2018, 2905회 인용
- 논문 출처

https://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf

---

ABSTRACT

- 다양한 분야에서 딥러닝과 같은 머신 러닝 기법들이 쓰임에도 불구하고, 여전히 많은 머신 러닝 기법들은 **'black box'**로 불림.
- 모델이 **복잡**해질수록 **모델의 추론 과정을 인간이 이해하기는 난이도가 점점 상승.** 하지만 모델을 이해하는 것은 굉장히 중요.(1, 2차함수 vs 다항함수)
- 모델을 이해할 수 있어야지만 그 모델을 trust 할 수 있기 때문
- 이 논문에서는 모델의 추론 과정을 이해하기 위해 **LIME(Local Interpretable Model-agnostic Explanations) 알고리즘**을 제안

---

locally interpretable(LI)

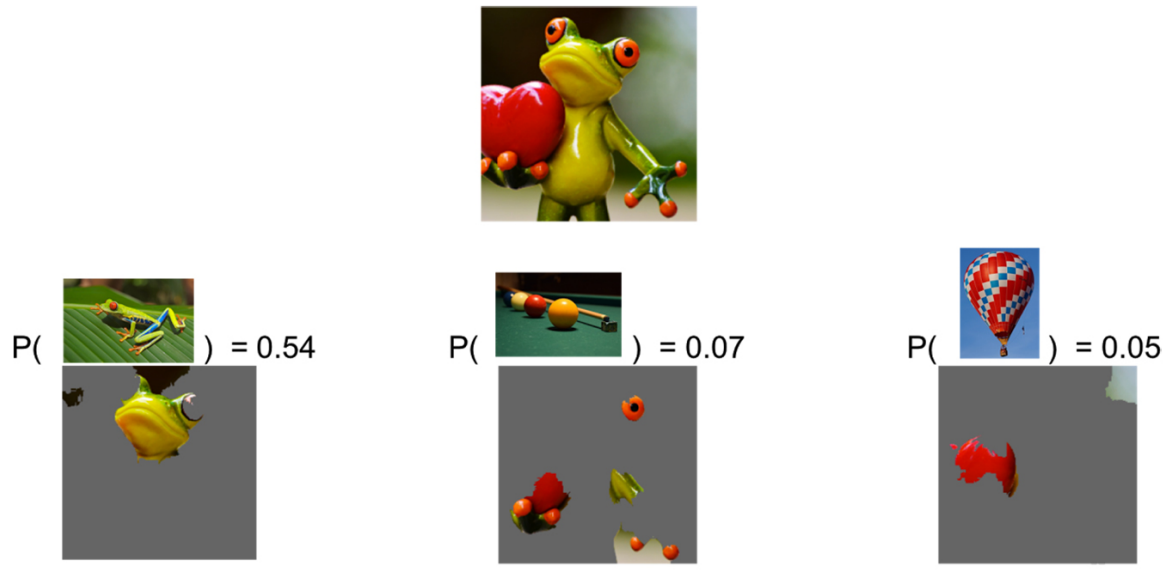
- 모형 자체가 어떠한 방식으로 작동하는지 이해하는 대신 Prediction 값 근방에 모형이 어떻게 작동하는지 설명
- 가정 : 작은 영역에서는 해석이 가능한 단순한 형태를 보일 것!!

model-agnostic(ME)

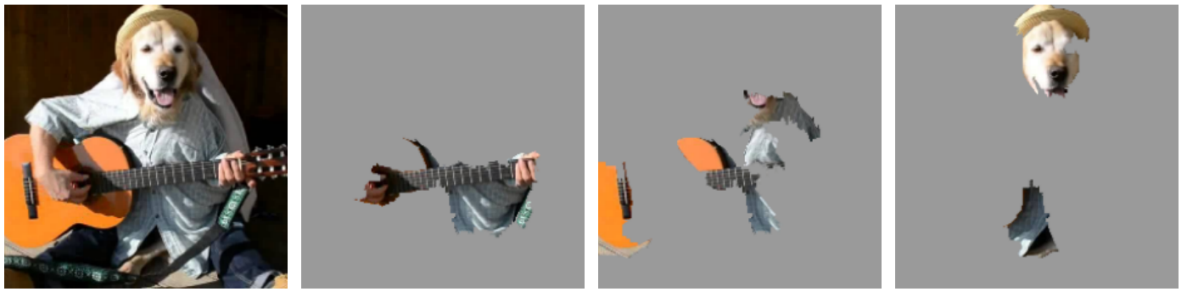
- 어떠한 모델도 설명이 가능하다!!!
- 모형에 대한 가정을 하지 않기 때문에 어떠한 모델에도 적용이 가능한 접근법!

Introduction

- **딥러닝 모델**의 경우 사람이 **레이어, 노드, 파라미터**를 보고 모델의 추론 과정을 해석하기란 매우 어려움
- 하지만 모델의 복잡한 내부가 아닌 주어진 데이터의 "**어떤 부분(local한 특성)**"을 보고 개구리, 당구공, 열기구라고 예측했는지라도 알 수 있다면, 사용자는 모델을 trust 가능



출처: Marco Tulio Ribeiro, Pixabay.



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

LIME의 General Framework

- 논문에서는 예측에 중요한 부분을 찾기 위해 복잡한 모델(원래 모델)을 해석 가능한 '***interpretable representation***' + '***interpretable model***'로 표현하여 해석
  - **interpretable representation** : 실제 사용하는 데이터를 사람이 해석할 수 있는 데이터로 재구성한(자기 자신을 잘 표현하는) 벡터
  - **interpretable model** : linear models, decision trees 등과 같이 사람이 직관적으로 이해할 수 있는 모델
- 아무리 복잡한 모델이라도 이를 사람이 **해석 가능한 데이터와 모델로 근사** 할 수 있다면 → 데이터의 **어떤 부분이 중요한 역할**을 했는지 알아낼 수 있음

---

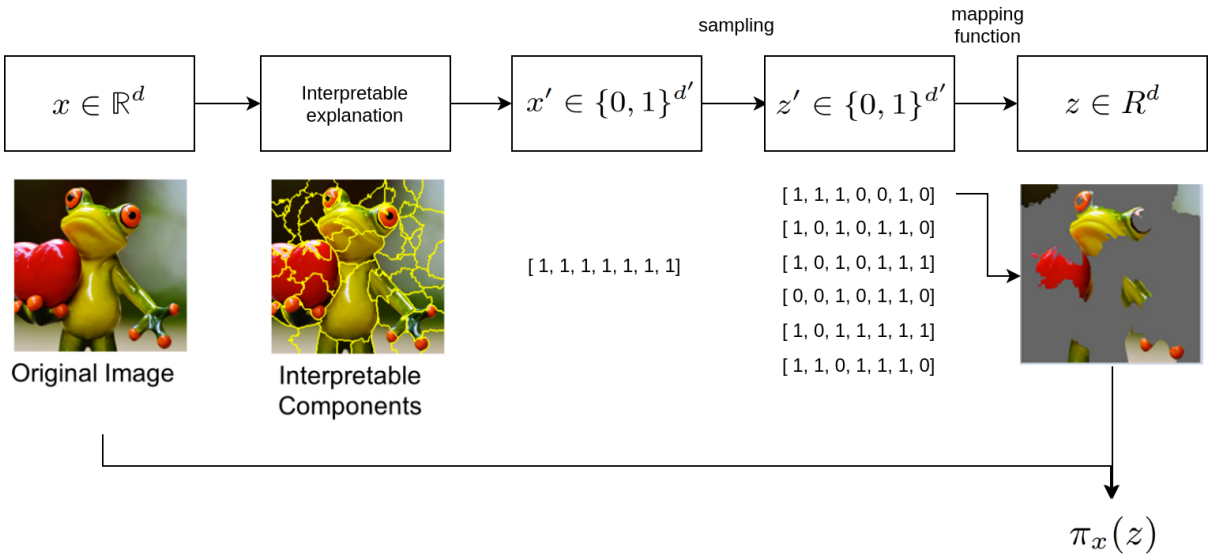
interpretable representation - 해석 가능한 특성(e.g. vision)

- interpretable representation은 사람이 이해할 수 없는 original 데이터를 사람이 이해 할 수 있는 데이터의 존재 유/무 로 표현한 binary vector(vision에서는 픽셀값, nlp에서는 단어)
  - 예를 들어 이미지를 분류하는 경우 original 데이터가 픽셀값
  - 각 픽셀값은 사람이 해석하기 어렵기 때문에 이를 사람이 해석할 수 있는 super-pixel(관련 링크)의 존재 유/무만 표현

---

1. 아래 그림에서 개구리 사진 픽셀(original 데이터)을 super-pixel 단위로 표현(사람이 **인지** 할 수 있도록 함)

2. 이후 super-pixel 단위의 존재 유무를 **binary vector x'**(interpretable representation) 으로 표현(아래 경우 모든 super-pixel이 1로 표현, x'의 정확한 의미는 하단의 암 환자 예시로 이해하는 것이 더 명확)
3. (여기서 부터 LIME idea) 표현한 **x'의 non-zero 부분에 대해서 부분집합을 하이퍼 파라미터 N개 만큼 sampling 하여 z'을 생성**
4. 다음으로 z'을 사용자가 지정한 mapping 함수(e.g. gray)를 통해 z로 재변환
5. 마지막으로 **원본 데이터 x** 와 **만들어진 z** 간의 **유사도를 측정 →** πx(z) 에 저장 이후에 원본 데이터와 가까운 sample에 가중치를 부여하기 위함(유사도는 이미지의 경우 L2 distance)



- Shape(dimension)
  - `x_d 차원` : original 데이터 (e.g. pixel)
  - `interpretable components_d' 차원` : 사람이 해석 가능한 데이터 단위
  - `x'_d'차원` : interpretable representation, x를 interpretable components 존재 유무로 표현한 binary 벡터 (e.g. 1의 의미는 개구리 이미지 super-pixel의 존재를 뜻함)
  - `z'_d'차원` : x' 중 non-zero의 부분집합 sampling, 사용자가 sampling 개수를 정의
  - `z_d차원` : z'을 mapping 함수를 정의하여(e.g. zero super pixel을 gray 로 채움)을 통해 다시 original 데이터의 형태로 변환한 데이터(origianl 데이터와 같은 dim)

- `πx(z)(스칼라)` : x와 z 유사도 (원본 데이터와 가장 가까운 sampling 데이터에 가중치를 주기 위함)

interpretable representation - 해석 가능한 특성(e.g. Text)

interpretable components의 의미를 조금 더 이해해 보자면, interpretable components 주어진 데이터에 한정되지 않습니다. (즉, d > d' or d < d')

- `original 데이터` : 주어진 문서 안에 있는 단어의 embedding vector(d는 주어진 문서 안에 있는 단어 수, d'는 단어 사전의 수)
- `반면 interpretable components` : 주어진 문서의 단어 뿐 아니라 다른 단어들을 포함한 단어 사전
- `Task` : 암 환자의 진단서 분류를 예시

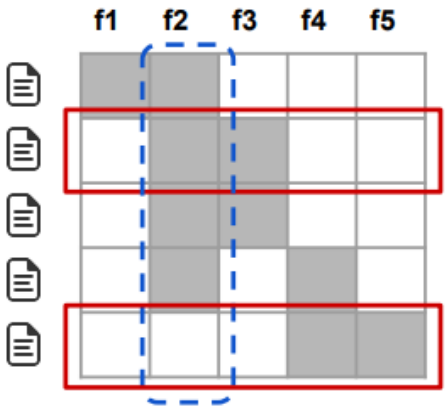


Figure 5: Toy example $\mathcal{W}$. Rows represent instances (documents) and columns represent features (words). Feature f2 (dotted blue) has the highest importance. Rows 2 and 5 (in red) would be selected by the pick procedure, covering all but feature f1.

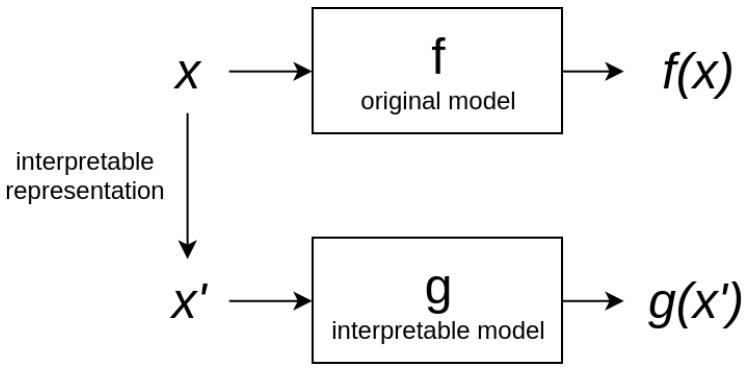
  - 100개의 진단서가 있는 경우 각 진단서에는 **서로 다른 여러개의 증상들이 나열되어 있음**
  - 이 때 첫번째 진단서를 doc1라고 한다면, doc1의 original 데이터 x는 doc1에 적힌 증상의 embedding vector이고 d차원은 doc1에 적힌 증상의 개수
  - 반면 interpretable components은 100개의 진단서에 있는 모든 증상의 binary 벡터이고, d'차원은 **100개 진단서에 있는 모든 증상의 개수**
- d차원과 d'차원의 차이가 하단에 서술할 LIME에 쓰여지는 local fidelity의 핵심!

개구리 예시와 같은 이미지의 경우 super-pixel은 각 이미지 마다 다르게 형성 됩니다. 따라서 텍스트 분류에서 단어 사전과 같이 interpretable representation을 명확히 정의할 수 없습니다. 논문에서도 color histograms or other features of super-pixels 등과 같은 방법으로 정의를 해야하고, 이는 차후 연구 주제 넘긴다고 명시해놨습니다.

interpretable model - 해석 가능한 모델

- 해석 가능한 모델(간단한 모델)의 대표적인 예시로, linear models, decision trees 이 존재.
  - linear models의 경우 각 변수들의 계수로 직관적 해석이 가능
  - decision trees의 경우 분리 기준으로 직관적 해석이 가능
  - 결국 복잡한 모델을 간단한 모델로 근사 시킬 수 있는 interpretable model을 찾는 것과 모델을 해석 할 수 있다는 것은 동일한 의미를 뜻합니다.

- interpretable model은 사람이 해석 가능한 간단한 모델
- 위에 서술한 interpretable representation을 사용하여 변수를 먼저 해석 가능하게 만듬
- 이후 original f(x)값을 해석 가능한 g(x')으로 근사시키는 것이 목적
- 이 때 f(x), g(x')은 분류의 경우 확률값임



- `x` : original 데이터

- `x'` : interpretable representation of x (x를 interpretable components 로 매핑한 binary 벡터)
- `f` : orginal model (복잡한 모델)
- `g` : interpretable model (사람이 해석 가능한 간단한 모델)
- `f(x)` : original 예측치
- `g(x')` : 해석 가능한 변수와 해석 가능한 모델을 쓴 예측치

결국 복잡한 함수 f에 근사 할 수 있는 해석 가능한 함수 g을 찾는게 interpretable model의 목표!!

LIME - Local Interpretable Model-agnostic Explanations

local fidelity

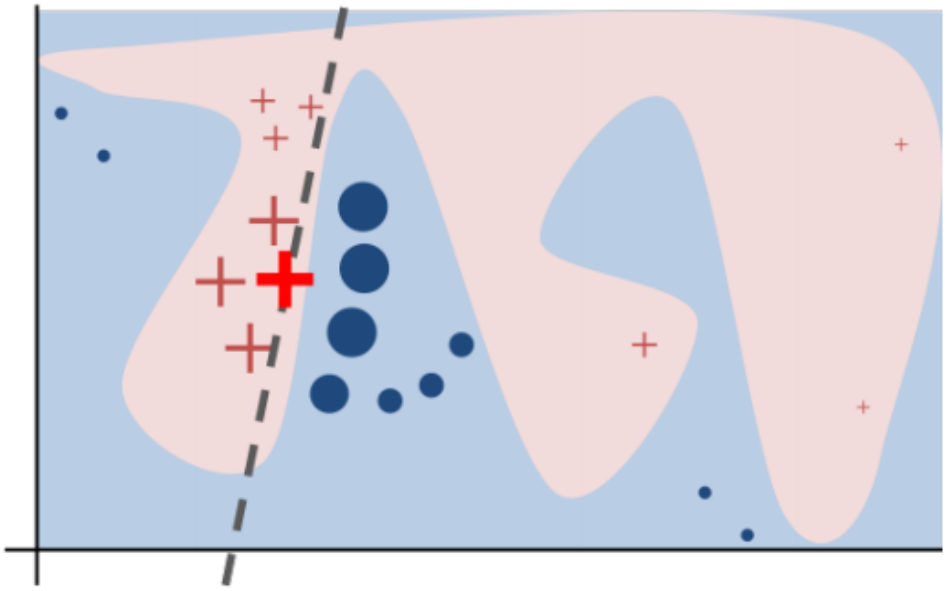
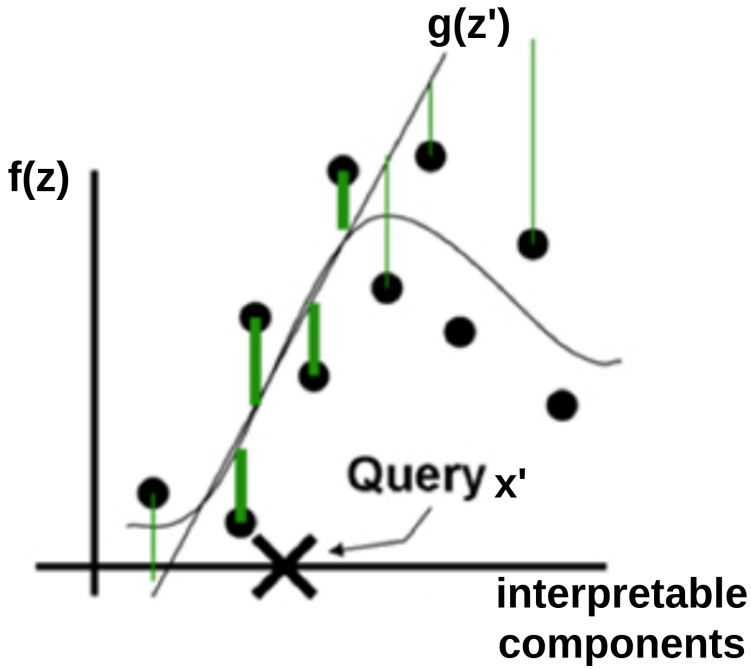


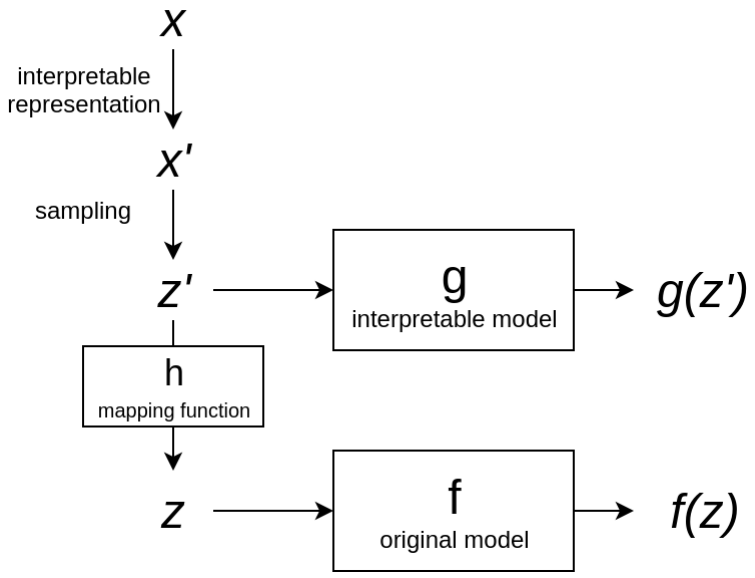
Figure 3: Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

논문에선 LIME에 들어가기 앞서 local fidelity에 대해 정의합니다. 데이터 전체에 대한 **결정면 혹은 분포**를 해석 가능한 함수(e.g. linear)로 표현하는 것은 **불가능** 하지만 **어떤 한 지점(local) 주변의 결정면 혹은 분포는 해석 가능한 함수로 표현 할 수 있다는 의미**입니다. 아래 Toy example 그림을 보면 전체 결정면은 비선형이지만, 특정한 지점(빨간색 굵은 십자가) 주변은 선형으로 표현 할 수 있습니다.

암 환자 예시로 보면, interpretable components은 단어 사전입니다. 각 점의 x좌표는 각 문서를 단어 사전의 binary vector로 표현한 interpretable representation이고, y좌표는 암 환자일 확률입니다. 결국 모든 단어 사전의 vector와 암 환자의 확률을 표현하는 분포는 복잡하지만 내게 주어진 문서를 변환한 x'(Query)과 x' 주변에 있는 z'만으로 한정한다면 선형으로 나타내는 것이 가능하다는 의미 입니다.



LIME Algorithm



1. 주어진 x를 해석 가능한 x' 으로 변환(e.g. 개구리 이미지를 super-pixel 단위의 binary vector로 변환)
2. x'의 non-zero fraction z'을 N개 추출.(N은 하이퍼 파라미터)
3. z'에서 h(z') 매핑 함수를 통해 z를 만듬

4. 원래 함수 f를 통해 z의 예측치 f(z) (e.g. sampling 된 z' 을 변환한 이미지의 개구리 확률) 계산
5. f(z)에 근사 할 수 있는 선형 g(z')을 탐색 → 선형 g(z')의 가중치를 통해 어떤 변수가 중요한 역할을 할 수 있는지 해석 (선형이 아닌 해석 가능한 다른 함수도 가능하지만 논문에선 선형을 사용)

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z_i'$$

Sparse Linear Explanations

- g(z')을 f(z)에 근사시키는 과정
- 먼저 Loss function 을 풀어 설명하면 Z 집합에 속하는 z와 z' 이 주어짐

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) \left(f(z) - g(z')\right)^2$$

- Z는 사용자가 지정한 sampling 개수(N) 개 집합
- f(z)와 g(z')의 오차에 x와 z 유사도 πx(z)를 곱해주어 원본과 가까운 sample에 가중치를 부여

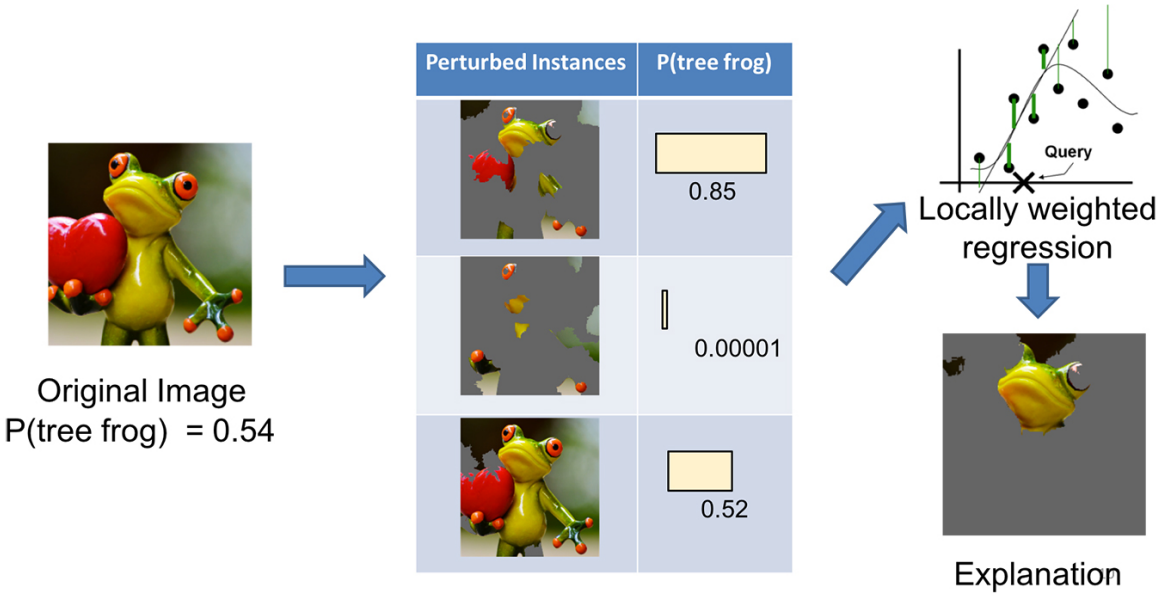
다음은 근사를 일반화 한 수식

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \ \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

- L : x에 대한 설명 모델의 손실(예를 들면, MSE)
- f : 설명하고자 하는 원래 모델
- g : Loss를 최소화 하는 모델(예를 들면, 선형회귀모델)

- G : 설명이 가능한 모든 모델, g의 상위 집합(예를 들면, 설명이 가능한 모든 선형 회귀 모델)
- Omega : 모델 복잡도, 낮게 유지하려고 함
- G 집합은 해석 가능한 모델 (linear, decision tree 등) 의 집합
- loss를 최소화 하는 동시에 g의 복잡도를 낮추는 페널티를 부여
- 논문에서는 Lasso-regression 을 이용

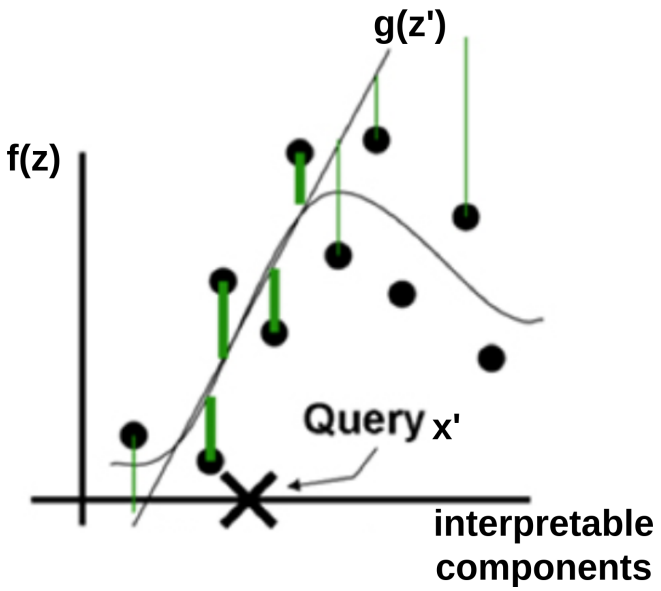
정리



출처: Marco Tulio Ribeiro, Pixabay.

- 개구리 이미지(x)가 들어오면 이는 interpretable representation(x')으로 변환
- 이는 N개의 non-zero fraction sampling z'이 되고 mapping 함수를 통해 부분 이미지(z)가 됨
- x', z'은 그림에서 생략되고 Perturbed Instance가 z → 이를 통해 f(z)를 구함

- 이제 구해진 f(z)를 통해 g(z') 을 찾고 싶어 → 이때 Locally weighted regression 이 사용
- 논문에서는 x와 z의 유사도 weight로 사용하고 Lasso를 통해 g를 찾음



(그림의 초록색 부분이 weighted loss πx(z)*(f(z) – g(z'))^2 를 나타내고 있음)

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z_i'$$

최종적으로 g(z')을 찾을 수 있고 높은 가중치를 가진 부분을 찾아보니 개구리 얼굴이라는 것을 알아 낼 수 있음 → 이를 통해 예측에 중요한 역할을 데이터를 찾고 모델을 해석 할 수 있습니다.

여러분들한테 당부의 말씀

- LIME 말고 SHAP(SHapley Additive exPlanations)라는 방법도 있는데 많이 쓴다고 합니다. 한번 공부해보세요~
- https://pair-code.github.io/what-if-tool/ 여기도 한번 가보세요 (https://youtu.be/ZAS2FwILTNE 송호연님 유튜브)
- 여기는 정말 흥미로운 소재 + 트렌디한 소재이니 공부해보세요~!

Reference

- Main Reference

LIME - why should i trust you
https://www.notion.so/LIME-why-should-i-trust-you-caf64b5b4d5f41ba824c690b718e5f13

- A Unified Approach to Interpreting Model Predictions

https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

- 파이썬 구현 및 한글 설명

LIME (Local Interpretable Model-agnostic Explanation)
https://nhlmary3.tistory.com/entry/LIME-LocalIly-Interpretable-Modelagnostic-Explanation

머신러닝 모델의 블랙박스 속을 들여다보기 : LIME
https://dreamgonfly.github.io/2017/11/05/LIME.html

- 교재가 따로 있을 정도(여기를 눌러서 반드시 읽어보시는 것을 권장)

Interpretable Machine Learning
https://christophm.github.io/interpretable-ml-book/

- 구현 공식 github

marcotcr/lime
https://github.com/marcotcr/lime/tree/ce2db6f20f47c3330beb107bb17fd25840ca4606