💛

[3주차] A Unified Approach to Interpreting Model Preictions

0. Abstract

- **SHAP**: 모델에서 예측을 해석하기 위한 통합된 프레임워크
  - 각 특성에 특정 예측에 대한 중요도 값을 할당
  - **구성요소 1️⃣**: 새로운 클래스의 가법적 특성 중요도 측정법의 식별
    - 가법적 특성 중요도 측정법의 새로운 클래스를 식별
    - 각 특성의 중요도를 수치로 나타내어 복잡한 모델의 예측을 해석하는 데 도움을 줌
  - **구성요소 2️⃣**: 해당 클래스에서 바람직한 성질을 가진 유일한 해결책이 있다는 이론적 결과 포함
    - 이론적으로 바람직한 성질을 가진 유일한 해결책 제공(중립성 >> 어떤 학습 모델에도 사용 가능)
    - SHAP가 다른 해석 방법과 구별되는 중요한 특징
  - 새로운 클래스는 여섯 가지 기존 방법을 통합하며, 몇 가지 최근 방법들이 제안된 바람직한 성질을 갖지 않음
    - 새로운 클래스는 여러 기존 해석 방법의 장점을 하나로 모아 새로운 해석 프레임워크를 제안, 다양한 방법론을 통합하여 더 포괄적이고 효율적인 모델 해석을 가능하게 함
    - 통합된 접근 방식이 모든 이론적 요구사항을 완벽히 충족시키지는 못함 >> 새로운 접근 방식이 여전히 개선의 여지가 있음을 나타냅니다.
  - SHAP의 장점
    - 여러 해석 방법을 하나의 통합된 프레임워크로 결합함으로써 사용자가 복잡한 모델의 예측을 보다 쉽게 이해하고 해석할 수 있게 해줌
    - 계산 성능을 향상시키고 인간의 직관과 더 잘 일치하는 새로운 방법을 제시 >> 복잡한 모델의 예측을 해석할 때 더 나은 결과를 얻을 수 있음을 의미함

1. Introduction

- 모델의 **예측 결과를 정확하게 해석**하는 것은 매우 중요
  - 사용자의 적절한 신뢰를 유도
  - 모델을 개선할 수 있는 통찰 제공
  - 모델링하는 과정을 이해하는 데 도움을 줌
  - 선형 모델과 같은 간단한 모델은 해석하기 쉬운 장점 때문에 복잡한 모델보다 정확도가 떨어질 수 있음에도 불구하고 종종 선호됨
  - 대규모 데이터의 증가로 복잡한 모델을 사용하는 이점이 커지면서 **모델의 정확도와 해석 가능성 사이의 균형**을 맞추는 것이 중요한 과제로 부상함
    - **정확도(Accuracy)**: 모델이 실제 데이터를 얼마나 잘 예측하는지에 대한 척도, 높은 정확도를 가진 모델은 실제 데이터와 예측 데이터가 매우 유사함을 의미
    - **해석 가능성(Interpretability)**: 모델의 예측 결과나 작동 원리를 사용자가 얼마나 쉽게 이해할 수 있는지에 대한 척도, 해석 가능한 모델은 사용자가 모델의 결정 과정을 쉽게 추적하고 이해할 수 있게 해줌
    - **예시 1️⃣**: 간단한 선형 회귀 모델은 해석하기 쉽지만, 복잡한 데이터 패턴을 포착하는 데는 한계가 있을 수 있음
    - **예시 2️⃣**: 딥러닝과 같은 복잡한 모델은 높은 정확도를 제공할 수 있지만, 그 내부 작동 원리를 이해하고 설명하기가 훨씬 어려움

    ➡️ **정확도-해석 가능성 사이의 균형을 맞추는 과제를 해결하기 위한 다양한 방법들이 최근 제안됨**

- 모델 예측을 해석하기 위한 새로운 통합 접근법 소개

  1️⃣ 모델 예측의 설명을 자체 모델로 보는 관점을 도입

  - '설명 모델'이라고 하며, 이를 통해 현재 여섯 가지 방법을 통합하는 '가산 특성 기여 방법(additive feature attribution methods)' 클래스를 정의
  - **설명 모델**: 모델의 예측을 설명하기 위해 사용되는 별도의 모델, 복잡한 머신러닝 모델의 결정 과정을 이해하기 쉽게 만들어 줌
  - **가산 특성 기여 방법**: 모델의 예측에 각 특성이 얼마나 기여하는지를 설명하는 방법으로, 여러 특성의 기여도를 합산하여 모델의 예측을 설명하며 SHAP 값을 포함한

여러 방법을 통합하여 정의됨

2️⃣ **게임 이론 결과가 전체 클래스에 유일한 해결책을 보장함을 보여줌**

- 이를 통해 다양한 방법이 근사하는 특성 중요도에 대한 통합된 측정치인 SHAP 값을 제안
- 게임 이론 결과: 게임 이론은 여러 참여자(또는 특성)가 서로 상호 작용하는 상황을 분석하는 수학적 이론. SHAP 값은 게임 이론의 결과를 바탕으로 모델 예측에 각 특성이 기여하는 정도를 수치화한 것
- SHAP 값: 모델의 예측에 대한 각 특성의 기여도를 나타내는 값. 특성 중요도에 대한 통합된 측정치로, 다양한 방법을 통해 근사할 수 있음

3️⃣ **새로운 SHAP 값 추정 방법을 제안**하고, 이 방법이 사용자 연구를 통해 측정된 **인간의 직관과 더 잘 일치**하며, 여러 기존 방법보다 **모델 출력 클래스를 더 효과적으로 구별함**을 보여줌

- **새로운 추정 방법**: 기존 방법들을 개선하여 사용자의 직관과 더 잘 일치하고 모델 출력 클래스를 더 효과적으로 구별할 수 있는 새로운 SHAP 값 추정 방법을 제안
- **사용자 연구 일치**: 이 새로운 방법은 사용자 연구를 통해 측정된 인간의 직관과 더 잘 일치함을 보여주며 모델의 예측을 설명하는 데 있어 더욱 효과적인 접근 방식을 제공

2. Addative Feature Attribution Methods

- 간단한 모델 vs 복잡한 모델
  - **간단한 모델**: 모델 자체가 가장 좋은 설명, 모델이 자기 자신을 완벽하게 대표하며 이해하기 쉬움
  - **복잡한 모델**: 앙상블 방법이나 심층 네트워크와 같은 복잡한 모델은 원래 모델을 그 자체로 사용하여 설명하기 어려움, 대신 **원본 모델의 해석 가능한 근사치**인 더 간단한 **설명 모델**을 사용하는 것이 좋음
- 설명 모델과 가산 특성 기여 방법

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

- **설명 모델**: 원본 예측 모델 f를 설명하기 위한 모델 g, 특정 입력 x에 대한 예측 f(x)를 설명하기 위해 설계된 지역적 방법에 초점을 맞춤
- **가산 특성 기여 방법**: 설명 모델이 이진 변수의 선형 함수로 정의됨, 즉, $g(z_o) = \varphi_o + \Sigma(\varphi i * z_o i)$로, 여기서 z_o는 {0, 1}의 값을 가지는 간소화된 입력 특성이며, φi는 실수. 이 방법은 각 특성에 대한 효과 φi를 할당하고, 모든 특성 기여도의 합이 원본 모델의 출력 f(x)를 근사함.
  - 설명 모델 $g(z_o)$는 이진 변수의 선형 함수로 정의됨. 여기서 z_o는 간소화된 입력 특성이며, {0, 1}의 값을 가짐. φi는 각 특성의 영향력을 나타내는 실수 값임.
  - 수식 $g(z_o) = \varphi_o + \Sigma(\varphi i * z_o i)$는 각 특성에 할당된 영향력 φi를 모두 더해 원본 모델 f(x)의 출력을 근사함.

  🏠 **예시 상황**
  1️⃣ 집 가격
  예측 모델이 있다고 가정, 위치/크기/방의 수 등 여러 특성을 기반으로 집의 가격을 예측함
  2️⃣
  간소화된 입력 특성 z_o: 각 특성을 이진 값으로 간소화함('위치' 특성이 '도심'인 경우 1, 아닌 경우 0으로 표현)
  3️⃣
  영향력 φi 할당: 각 특성이 집 가격에 미치는 영향력을 수치로 할당('도심 위치'($z_o i=1$)가 집 가격에 미치는 영향력 φi가 50만 원)
  4️⃣
  예측 근사: 모든 특성의 영향력을 합산하여 원본 모델의 예측 가격을 근사함('도심 위치'와 '큰 크기' 두 특성만 고려한다면, 이 두 특성의 영향력 합이 원본 모델이 예측한 집 가격과 유사한 값을 가지게 됨)

- 통합된 접근 방식의 중요성
  - 이전에 제안된 여러 설명 방법이 실제로 동일한 설명 모델을 사용한다는 것을 보여줌

2.1 LIME

LIME(Local Interpretable Model-agnostic Explanations): 복잡한 머신러닝 모델의 예측을 해석하는 방법, 주어진 예측 주변에서 모델을 **국소적으로 근사화**하여 개별 모델 예측을 해석함

$$\xi = \arg\min_{g \in \mathcal{G}} \; L(f, g, \pi_{x'}) + \Omega(g).$$

▼ 국소적 근사화

- 복잡한 모델이 어떤 예측을 내리는지 정확히 이해하기 어렵기 때문에, 해당 예측 주변에서만 모델의 동작을 단순한 모델로 근사화하여 설명
- 주어진 예측에 대해 모델의 예측을 잘 설명할 수 있는 간단한 모델 생성, 이 간단한 모델은 원본 모델의 복잡한 동작을 국소적으로(=특정 예측 주변에서만) 모방함

▼ 개별 예측 해석

- 모델이 특정 입력에 대해 내린 결정이나 예측의 근거를 이해하고 설명하는 과정
  >> 모델의 예측이 단순히 '어떤 결과'를 내놓는 것이 아니라, '
  왜 그런 결과를 내놓았는지'를 분석하는 것
- **LIME**: 복잡한 모델의 예측을 국소적으로 근사화하여 특정 예측에 대한 모델의 동작을 설명함. 예를 들어 특정 텍스트가 긍정적인 예측을 받았다면, LIME은 그 예측에 가장 큰 영향을 준 단어나 구를 식별함.
  - 해당 리뷰 내의 긍정적인 단어들 주변의 국소적 영역을 정의하고, 이 영역에서 원본 모델의 동작을 간단한 모델로 근사화
  - 이를 통해 해당 단어들이 긍정적 예측에 어떻게 기여했는지를 이해할 수 있음
- **SHAP**: 각 입력 특성이 최종 예측에 미치는 영향을 정량화하여 모델의 예측을 해석함. SHAP는 게임 이론에서 영감을 받아 각 특성의 '공정한' 기여도를 계산함.
- 예시: 금융 신용 점수 예측에서, 모델이 특정2 고객에게 낮은 신용 점수를 예측했다면 LIME이나 SHAP를 사용하여 어떤 특성(예: 지불 이력, 대출 금액)이 주로 그 예측에 영향을 미쳤는지 분석할 수 있음

- LIME의 기본 원리
  - **해석 가능한 입력**: LIME에서는 간소화된 입력 x_o를 "해석 가능한 입력"이라고 부르며, 이는 원본 입력 공간으로 변환되기 전의 이진 벡터 형태의 입력을 의미
  - **변환 매핑 x = hx(x_o)**: 이 매핑은 해석 가능한 입력을 원본 입력 공간으로 변환하며, 입력 공간의 유형에 따라 다른 hx 매핑이 사용됨

- 입력 공간에 따른 hx 매핑의 예

    - **텍스트**: 단어의 존재 여부를 나타내는 1 또는 0의 벡터를 원래 단어의 수로 변환
    - **이미지**: 이미지를 슈퍼 픽셀의 집합으로 취급했을 때, 1은 슈퍼 픽셀을 원래 값으로 유지하며 0은 슈퍼 픽셀을 이웃 픽셀의 평균으로 대체

- φ 찾기: 목적 함수 최소화
    - **목적 함수**: $\xi = \arg\min_{g \in G} L(f, g, \pi x_o) + \Omega(g)$. 여기서 L은 간소화된 입력 공간에서의 샘플 집합에 대한 손실을 나타내며, Ω는 g의 복잡성을 감소시킴
    - **해석 모델의 충실도**: $g(z_o)$ 모델이 원본 모델 $f(hx(z_o))$에 충실하도록 하기 위해 손실 L을 통해 국소 커널 πx_o에 의해 가중된 간소화된 입력 공간의 샘플 집합에 대한 손실을 최소화함
    - **해결 방법**: LIME에서 g는 방정식 1을 따르고 L은 제곱 손실이기 때문에, 방정식 2는 패널티가 있는 선형 회귀를 사용하여 해결할 수 있음

- LIME 방법론의 중요성
    - 모델이 특정 예측을 내리는 이유를 이해하는 데 도움을 줌 ➡️ 모델의 해석 가능성을 높이고, 복잡한 모델의 예측을 신뢰할 수 있게 만들어 줌
    - 복잡한 모델의 예측을 해석할 수 있는 강력한 도구 ➡️ 모델의 예측이 어떻게 이루어지는지 더 잘 이해하고, 모델의 신뢰성을 높일 수 있음

2.2 DeepLIFT

DeepLIFT: 딥러닝 모델의 예측을 설명하기 위한 방법 중 하나로, 각 입력값이 모델의 출력에 미치는 영향을 수치로 나타내는 기법

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta o} = \Delta o,$$

- DeepLIFT의 핵심 개념
    - **입력값의 영향력**: DeepLIFT는 각 입력값 (x_i)이 모델의 예측 결과 (y)에 미치는 영향을 (CΔx_iΔy)로 나타내며 여기서 (Δx_i)는 입력값이 참조값(기준값)과 비교했을 때의 변화량을 의미하고, (Δy)는 그 변화가 모델의 출력에 미치는 영향을 나타냄

- **참조값(Reference Value)**: 사용자가 선택하는 이 값은 일반적으로 정보가 없는 배경값을 나타냄(ex. 이미지 처리에서는 검은색 배경이 참조값이 될 수 있음)
- **변환 매핑 (x=hx(x_0))**: 이 매핑은 이진값을 원래의 입력값으로 변환하며, 1은 입력값이 원래의 값이고, 0은 참조값을 취한다는 것을 의미
  - 특정 픽셀이 원래 이미지에서 얼마나 중요한지, 또는 참조 이미지 대비 얼마나 변화를 주는지 나타냄
  - 1로 매핑되는 픽셀은 모델 예측에서 중요한 역할을 함
  - 0으로 매핑되는 픽셀은 참조값으로 간주되어 변화를 주지 않음
- **summation-to-delta 속성**: DeepLIFT는 **모든 입력값의 영향력의 합이 모델 출력의 변화량과 같다**는 속성을 사용함(위 수식 참고). 여기서 (o = f(x))는 모델의 출력, (Δo = f(x) - f(r)), (Δx_i = x_i - r_i), (r)은 참조 입력을 의미함
  -

예시 상황
1️⃣ 고양이와 개를 구분하는 이미지 분류 모델이 있다고 가정. DeepLIFT를 사용하여 이 예측에 가장 큰 영향을 미친 입력 픽셀(또는 이미지의 특정 부분)을 알아낼 수 있음
2️⃣
참조값 설정: 먼저 참조값(기준값)을 설정함. 이는 일반적으로 정보가 없는 상태를 나타내며, 이미지의 경우 검은색 이미지나 평균 이미지가 될 수 있음.
3️⃣
입력값과 참조값의 차이 계산: 고양이 이미지(입력값)와 참조값 사이의 차이를 계산함. 이 차이는 모델의 예측에 어떤 영향을 미쳤는지를 나타내는 기준이 됨.
4️⃣
영향도 계산: 각 입력 픽셀이 모델의 예측 결과(이 경우 고양이라고 예측)에 미친 영향을 계산함. 이를 통해 어떤 픽셀(또는 이미지의 부분)이 예측에 가장 중요한 역할을 했는지 알 수 있음.
예를 들어 모델이 고양이의 귀와 눈을 중요한 특징으로 보고 이를 기반으로 고양이라고 예측했다면, DeepLIFT는 이러한 부분의 픽셀이 높은 영향도를 가지고 있음을 보여줄 것

2.3 Layer-Wise Relevance

Layer-Wise Relevance Propagation(LPR): 딥러닝 모델의 예측을 해석하는 방법 중 하나로, 모델의 예측에 대한 입력의 중요도를 **역추적**하여 표시함. 특히 이미지와 같은 입력 데이터에 대해 모델이 어떤 부분을 중요하게 여기는지를 시각화하는 데 유용

- LRP의 기본 원리
    - **타당성 전파**: 모델의 예측 결과에 대한 각 입력의 기여도를 역추적하는 과정으로, 특정 결과가 나오게 된 원인을 분석하고 그 비중을 각 입력에 분배함
    - **분해**: 타당성 전파를 통해 얻어낸 '원인'을 각 입력의 가중치로 환원하고 해부하는 과정으로, 이를 통해 모델이 어떤 입력을 중요하게 여기는지를 구체적으로 파악할 수 있음.
- LRP와 DeepLIFT의 관계
    - DeepLIFT와의 유사성: LRP는 DeepLIFT와 유사한 방식으로 작동함. **DeepLIFT에서 언급된 "x = hx(x_0)" 변환 매핑은 LRP에서도 적용**되며, 여기서 1은 입력이 원래의 값을 취하고, 0은 입력이 0 값을 취한다는 것을 의미함. 이는 모델의 예측에 대한 입력의 중요도를 결정하는 데 사용됨.
    - DeepLIFT와의 차이점: LRP는 모든 뉴런의 참조 활성화를 0으로 고정한다는 점에서 DeepLIFT와 차이가 있음. 이는 LRP가 모델의 예측을 해석하는 데 있어 더 단순화된 접근 방식을 제공한다는 것을 의미함.

> 👉 **참조 활성화의 의미**
>
> **- LPR**: 신경망의 각 뉴런이 최종 예측에 기여하는 정도를 **역전파**하는 방식으로 해석함. 이 방법은 신경망의 각 계층을 거슬러 올라가며, 최종 출력에서 각 입력 특성까지의 관련성을 전파함.
>
> **- DeepLIFT**: 참조 활성화 상태와 비교하여 입력의 변화가 출력에 미치는 영향을 측정함. 여기서 참조 활성화는 일종의 기준점 또는 비교점을 의미함.
>
> **- 참조 활성화**
> 참조 활성화(reference activation)는 모델의 예측을 해석할 때 기준으로 삼는 뉴런의 활성화 상태. 예를 들어 이미지를 분류하는 신경망에서 특정 픽셀의 중요도를 평가하고자 할 때, 해당 픽셀을 제외한 상태(예: 픽셀 값을 0으로 설정)와 포함한 상태를 비교하여 그 픽셀의 중요도를 평가할 수 있습니다.
>
> **- "모든 뉴런의 참조 활성화를 0으로 고정"**
> LRP는 모든 뉴런의 참조 활성화를 0으로 고정함. 즉, LRP는 각 뉴런의 기여도를 평가할 때, 해당 뉴런이 비활성화된 상태(활성화 값이 0)를 기준으로 삼아 그 뉴런이 최종 예측에 미치는 영향을 계산하며 이는 LRP가 더 단순화된 접근 방식을 제공한다는 것을 의미함. 반면 DeepLIFT는 참조 활성화를 더 유연하게 설정할 수 있으며, 이는 특정 상황이나 데이터에 더 적합한 기준점을 사용할 수 있도록 함.

- LRP의 중요성
  - 시각화를 통한 해석: LRP는 입력 데이터에 대한 히트맵을 생성하여, 모델이 어떤 부분을 주목하는지 시각적으로 표현할 수 있음. 이는 모델의 예측 과정을 이해하고, 모델이 중요하게 여기는 특성을 파악하는 데 도움을 줌.

2.4 Classic Shapley Value Estimation

Classic Shapley Value Estimation(클래식 샤플리 값 추정): 모델 예측의 설명을 위해 협력 게임 이론에서 유래한 세 가지 방법이 사용됨

1️⃣ 샤플리 회귀 값(Shapley Regression Values)

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right].$$

- **다중공선성이 있는 선형 모델에서의 특성 중요도**: 샤플리 회귀 값은 다중공선성이 존재할 때 선형 모델의 특성 중요도를 계산하는 방법으로, 모든 특성 부분 집합 S ⊆ F에 대해 모델을 **재학습**하며 여기서 F는 모든 특성의 집합이 됨. 각 특성에 중요도 값을 할당하여, 해당 특성을 포함시킬 때 모델 예측에 미치는 영향을 나타냄.
- **영향력 계산**: 특정 특성이 포함된 모델 fS∪{i}와 제외된 모델 fS를 학습한 후, 두 모델의 예측을 현재 입력에 대해 비교(fS∪{i}(xS∪{i}) − fS(xS)). 특성을 제외하는 효과는 모델 내 다른 특성에 의존하기 때문에, 가능한 모든 부분 집합 S ⊆ F \ {i}에 대해 이전 차이를 계산함.

2️⃣ 샤플리 샘플링 값(Shapley sampling values)

- **모든 모델에 적용 가능**: 샤플리 샘플링 값은 모델을 재학습할 필요 없이 훈련 데이터셋에서 샘플을 통합하여 모델에서 변수를 제거하는 효과를 근사하는 방법으로, 모든 모델을 설명하기 위해 사용됨. 이는 2^|F|보다 적은 차이를 계산할 수 있게 해줌.

3️⃣ 정량적 입력 영향(Quantitative Input Influence)

- **특성 기여도 이상의 폭넓은 프레임워크**: 정량적 입력 영향은 특성 기여도 이상을 다루는 더 넓은 프레임워크. 그러나 이 방법은 샤플리 샘플링 값과 거의 동일한 샤플리 값의 샘플링 근사를 독립적으로 제안함으로써, 또 다른 가산 특성 기여 방법이 됨.

3. Simple Properties Uniquely Determine Additive Feature Attributions

- additive feature attribution methods의 세 가지 바람직한 속성
- 고전적인 Shapley 값 추정 방법에서는 잘 알려져 있지만 다른 additive feature attribution 방법들에서는 이전에 알려지지 않았던 것들

1️⃣ 지역 정확성(Local Accuracy)

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i$$

- 특정 입력 (x)에 대해 원본 모델 (f)를 근사할 때, 설명 모델이 적어도 간소화된 입력 (x_0) (원본 입력 (x)에 해당)에 대한 (f)의 출력과 일치해야 한다는 요구사항
- 수식에서 (f(x))는 원본 모델의 출력, (g(x_0))은 설명 모델의 출력, (\phi_0)은 기본 출력값, (\phi_i)는 특성 (i)의 기여도, 그리고 (x_{0i})는 간소화된 입력을 의미함

2️⃣ 미싱니스(Missingness)

$$x_i' = 0 \implies \phi_i = 0$$

- 간소화된 입력이 특성의 존재를 나타낼 때 원본 입력에서 누락된 특성은 영향을 미치지 않아야 한다는 요구사항. 이 논문의 2장에서 설명된 모든 방법들은 미싱니스 속성을 준수함
    - 특정 특성이 모델 입력에서 누락되었을 때(즉, 그 값이 0일 때), 그 특성에 할당된 영향력도 0이 되어야 함
- **특성의 누락**: 만약 입력에서 어떤 특성의 값이 0이라면 이는 그 특성이 누락되었거나 '없음'을 의미함
- **영향력의 부재**: 해당 특성이 모델 예측에 미치는 영향력도 0이 되어야 함. 즉, 누락된 특성은 예측 결과에 아무런 영향을 미치지 않아야 함

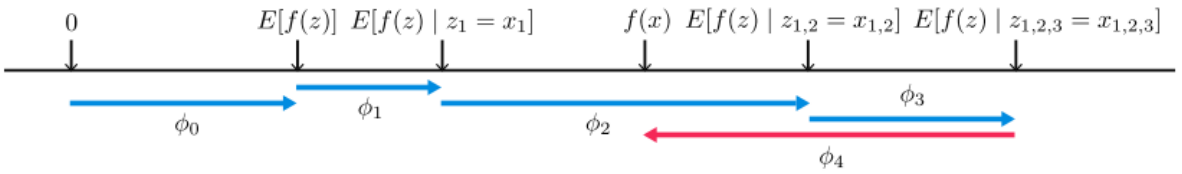
3️⃣ 일관성(Consistency)

- 일관성: 모델이 변경되어 어떤 간소화된 입력의 기여도가 증가하거나 다른 입력에 관계없이 동일하게 유지되는 경우, 해당 입력의 기여도는 감소하지 않아야 함
    - 특정 특성의 중요성이 증가하거나 유지된다면 그 특성의 기여도가 줄어들어서는 안 된다는 것을 의미

<정리>

- 복잡한 모델의 예측을 해석하는 데 도움을 주기 위해 다양한 방법들이 제안됨
- 이 방법들이 어떻게 관련되어 있는지, 어떤 방법이 다른 방법보다 선호되어야 하는지 종종 **불분명**하다는 문제를 해결하기 위해 **SHAP라는 통합된 프레임워크**를 제시
- SHAP는 샤플리 값에 기반한 해석 방법으로, 모델 예측의 해석을 표준화하고 명확하게 하려는 시도라고 볼 수 있음

4. SHAP(SHapley Additive exPlanation) Values



- SHAP 값은 모델의 원래 모델에 대한 조건부 기대 함수의 셰플리 값임. 여기서 (f_x(z_0) = f(h_x(z_0)) = E[f(z) | z_S])이고, (S)는 (z_0)에서 0이 아닌 인덱스의 집합이 됨
- SHAP 값은 속성 1-3을 준수하고 조건부 기대를 사용하여 간소화된 입력을 정의하는 유일한 가법적 특성 중요도 측정값을 제공함
  - SHAP 값의 이 정의에는 간소화된 입력 매핑, (h_x(z_0) = z_S)가 내포되어 있으며, 여기서 (z_S)는 집합 (S)에 없는 특성에 대해 누락된 값을 가짐
- SHAP 값의 정의는 셰플리 회귀, 셰플리 샘플링, 정량적 입력 영향 특성 기여도와 밀접하게 일치하도록 설계되었으며 동시에 LIME, DeepLIFT, 계층별 관련 전파와 같은 연결을 허용함

- SHAP 값의 정확한 계산은 도전적인 부분이 있으나, 현재 가법적 특성 기여도 방법에서 얻은 통찰을 결합함으로써 이들을 근사할 수 있음
- 네 가지 모델 유형별 근사 방법을 설명
  - 특성 독립성과 모델 선형성은 기대값의 계산을 단순화하는 두 가지 가정임

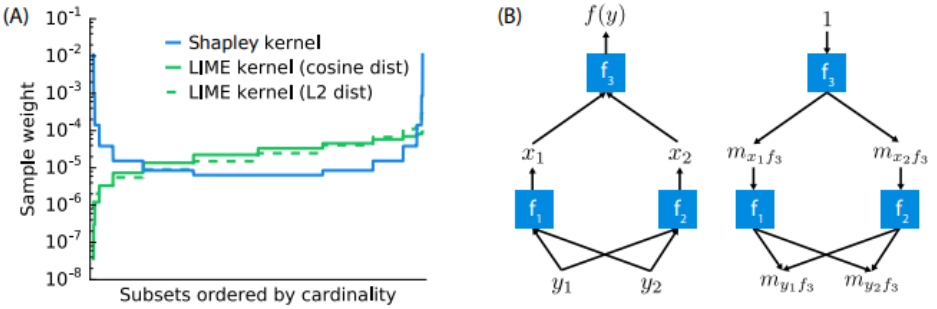


Figure 2: (A) The Shapley kernel weighting is symmetric when all possible z' vectors are ordered by cardinality there are 2^{15} vectors in this example. This is distinctly different from previous heuristically chosen kernels. (B) Compositional models such as deep neural networks are comprised of many simple components. Given analytic solutions for the Shapley values of the components, fast approximations for the full model can be made using DeepLIFT's style of back-propagation.

4.1 Model-Agnostic Approximations

모델 비특정 근사 방법: 모델의 예측을 해석하는 데 있어서 SHAP 값의 중요성을 강조함. 이 섹션에서는 SHAP 값의 추정 방법에 대해 설명함

- 셰플리 샘플링 값 방법과 양적 입력 영향 방법
  - 특성 독립성 가정: 조건부 기대를 근사할 때 특성 독립성을 가정하면 셰플리 샘플링 값 방법이나 양적 입력 영향 방법을 사용하여 SHAP 값이 직접 추정될 수 있음
  - 이 방법들은 셰플리 값 방정식의 순열 버전을 샘플링 근사하며, 각 특성 기여도에 대해 별도의 샘플링 추정이 수행됩니다
- Kernel SHAP(선형 LIME + 셰플리 값)
  - 선형 LIME은 지역적으로 함수 (f)를 근사하기 위해 선형 설명 모델을 사용함
  - 선형 LIME은 가법적 특성 기여도 방법이므로, 셰플리 값이 속성 1~3(지역 정확성, 누락성, 일관성)을 만족함
  - Kernel SHAP은 LIME과 셰플리 값의 아이디어를 사용하여 SHAP 값을 근사하는 모델 비특정 방법임
- 셰플리 커널 정리
  - 셰플리 커널 정리는 손실 함수 (L), 가중치 커널 (\pi_{x_0}), 정규화 항 (\Omega)이 속성 1~3을 만족하는 특정 형태를 제시
  - 선형 회귀를 사용하여 게임 이론에서의 셰플리 값이 계산될 수 있으며, 이는 회귀 기반, 모델 비특정 SHAP 값 추정을 가능하게 함

4.2 Model-Specific Approximations

모델별 근사 방법: 특정 모델 유형에 초점을 맞추어 SHAP 값의 샘플 효율성을 개선하는 방법. 이 섹션에서는 다양한 모델 유형에 대한 근사 방법을 설명함

- 선형 SHAP
  - 선형 모델의 경우 입력 특성의 독립성을 가정하면 모델의 가중치 계수로부터 직접 SHAP 값을 근사할 수 있음
- 저차 SHAP
  - 선형 회귀를 사용하는 저차 SHAP 근사는 (O(2^M + M^3))의 복잡성을 가지며, 조건부 기대치의 근사를 선택할 때 M의 작은 값에 대해 효율적임
- Deep SHAP

- Kernel SHAP은 딥러닝 모델을 포함한 모든 모델에 사용될 수 있지만, 딥 네트워크의 구성적 특성에 대한 추가 지식을 활용하여 계산 성능을 개선할 수 있는 방법이 있는지에 대한 질문이 제기됨
- Deep SHAP은 셰플리 값과 DeepLIFT 사이의 이전에 간과된 연결을 통해 이 질문에 대한 답을 찾음
- Deep SHAP은 네트워크의 작은 구성 요소에 대해 계산된 SHAP 값들을 전체 네트워크에 대한 SHAP 값으로 결합함
- 이는 각 구성 요소에 대해 계산된 SHAP 값으로부터 효과적인 선형화를 도출함으로써, 구성 요소를 선형화하는 방법을 경험적으로 선택할 필요를 피함

5. Computational and User Study Experiments

SHAP 값의 이점에 대한 평가: Kernel SHAP과 Deep SHAP 근사 방법을 사용

1️⃣ Kernel SHAP과 LIME, 그리고 Shapley 샘플링 값의 계산 효율성과 정확성을 비교

- **Kernel SHAP vs. LIME**: Kernel SHAP과 LIME은 모두 모델 설명 가능성을 위한 인기 있는 방법임. Kernel SHAP은 LIME의 블랙 박스 로컬 추정 이점과 함께 일관성과 관련된 이론적 보장을 제공함

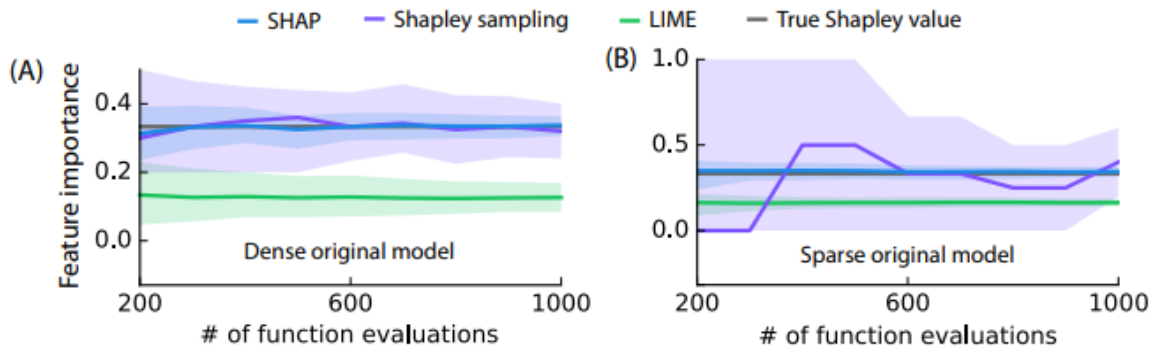
2️⃣ 사용자 연구를 설계하여 SHAP 값과 DeepLIFT 및 LIME에 의해 표현된 대안적 특성 중요도 할당을 비교

- **SHAP 값 vs. DeepLIFT 및 LIME**: SHAP 값은 DeepLIFT 및 LIME과 같은 다른 특성 중요도 할당 방법보다 인간의 직관과 더 일관성이 있음을 보여줌. 이는 SHAP 값이 모델 예측을 해석하는 데 있어 더 신뢰할 수 있는 방법임을 시사함
- 예상대로 SHAP 값은 다른 방법들이 충족하지 못하는 속성 1~3을 만족시키기 때문에 인간의 직관과 더 일관성이 있음이 입증됨

3️⃣ MNIST 숫자 이미지 분류를 사용하여 SHAP과 DeepLIFT 및 LIME을 비교

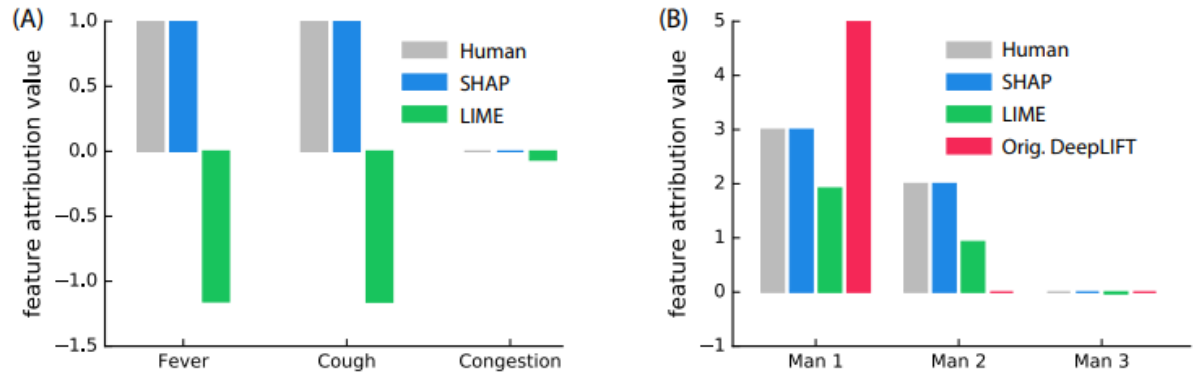
- **SHAP vs. DeepLIFT 및 LIME**: MNIST 숫자 이미지 분류를 사용한 비교를 통해 SHAP 값이 DeepLIFT 및 LIME과 같은 다른 방법들에 비해 어떤 이점을 제공하는지 평가할 수 있음

5.1 Computational Efficiency



- 계산 효율성의 개선
  - **샤플리 값과 가중치가 있는 선형 회귀의 연결**: Theorem2는 게임 이론에서의 샤플리 값과 가중치가 있는 선형 회귀 사이의 연결을 설명하며, Kernel SHAP은 이 연결을 활용하여 특성의 중요도를 계산함
  - **정규화가 추가된 선형 모델의 정확도 향상**: 정규화를 추가할 때 Kernel SHAP은 이전의 샘플링 기반 추정보다 원 모델의 평가 횟수를 줄이면서 더 정확한 추정치를 제공함
- Kernel SHAP, SHAP, LIME 비교
  - **샘플 효율성**: Kernel SHAP은 샘플 효율성이 향상되었음을 보여줍니다
  - **LIME과 SHAP 값의 차이**: LIME에서 나오는 값은 지역 정확성과 일관성을 만족하는 SHAP 값과 상당히 다를 수 있음. 이는 SHAP 값이 더 정확하고 일관된 특성 중요도를 제공할 수 있음을 의미함.

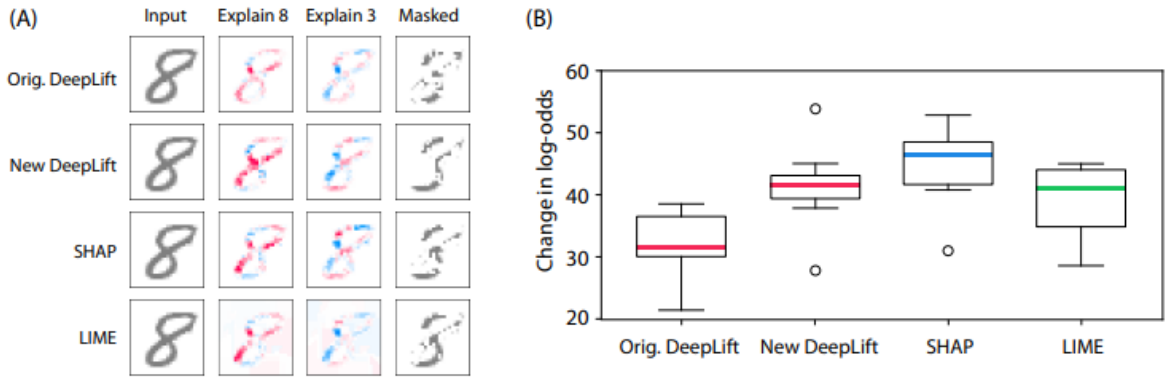
5.2 Consistency with Human Intuition



- 인간 직관과의 일관성 검증

    - **Theorem 1**: 모든 가산 특성 귀속 방법이 SHAP 값을 사용하도록 하며, 이는 LIME과 DeepLIFT와 같은 다른 방법들이 서로 다른 특성 중요도 값을 계산한다는 것을 의미함
    - **사용자 설명과의 비교**: 아마존 메카니컬 터크를 사용하여 간단한 모델에 대한 LIME, DeepLIFT, 그리고 SHAP의 설명을 사용자 설명과 비교했으며, 좋은 모델 설명은 그 모델을 이해하는 사람들의 설명과 일관되어야 함
- 두 가지 설정에서의 비교
    - **병의 점수 설정**: 두 증상 중 하나만 존재할 때 더 높은 병의 점수를 사용하는 설정으로, SHAP은 이 설정에서 인간의 설명과 더 강한 일치를 보였음
    - **최대 할당 문제 설정**: 세 남자가 달성한 최대 점수에 기반하여 돈을 벌었다는 설정으로, 이 경우에도 SHAP은 다른 방법들보다 인간의 설명과 더 일치했음

5.3 Explaining Class Differences



- 클래스 차이 설명의 중요성
    - **Deep SHAP의 도입**: DeepLIFT의 구성적 접근법은 SHAP 값의 구성적 근사치인 Deep SHAP을 제안하며, 이는 DeepLIFT를 개선하고 Shapley 값과 더 잘 일치하도록 새로운 버전을 포함함
    - **성능 향상**: DeepLIFT의 컨볼루션 네트워크 예제는 SHAP 값에 더 가까운 추정치의 성능 향상을 강조하며 이는 사전 훈련된 모델과 정규화된 입력을 기반으로 함
- 모델 설명 방법의 비교
    - **DeepLIFT vs. SHAP과 LIME**: DeepLIFT의 두 버전은 선형 레이어의 정규화된 버전을 설명하는 반면, SHAP과 LIME은 모델의 출력을 설명함. SHAP과 LIME은 모두 성능 향상을 위해 50k 샘플로 실행되었음

6. Conclusion

- SHAP 프레임워크의 중요성
  - **가산적 특성 중요도 방법**: SHAP 프레임워크는 이전의 여러 방법을 포함하는 가산적 특성 중요도 방법의 클래스를 식별합니다. 이는 모델 예측을 해석하는 데 중요한 역할을 함
  - **일관성과 통합**: SHAP은 모델 해석에 대한 공통 원칙을 문헌에 통합함으로써, 향후 방법의 개발을 안내할 수 있는 희망적인 신호를 제공합니다
- SHAP 값 추정 방법의 발전
  - **다양한 추정 방법**: SHAP 값에 대한 여러 가지 추정 방법이 제시되었으며 이러한 방법들은 모델 예측의 해석을 개선하는 데 기여함
  - **향후 발전 방향**: 더 적은 가정을 하는 모델 유형별 추정 방법의 개발, 게임 이론에서 상호 작용 효과 추정의 통합, 새로운 설명 모델 클래스의 정의 등이 유망한 다음 단계로 제시됨

논문에 대한 의견 및 의문점(꼭지)

➡️ 해당 논문에서는 더 적은 가정을 하는 모델 유형별 추정 방법의 개발을 제안함. 이러한 방법들이 실제로 모델 해석을 개선할 수 있을지 의문이 듦. 모델의 유형별로 특화된 추정 방법을 개발하면서도 해석 가능성을 유지하는 것이 필요할 것으로 보임. 또한 특정 모델에 특화된 추정 방법을 개발하는 것은 그 모델에 대해 더 깊은 이해를 가능하게 하지만, 다른 모델로의 적용성이나 일반화에는 한계가 있을 수 있음. 이는 다양한 모델 유형에 대해 유사한 접근법을 개발해야 함을 의미