**Explainable AI for Classification using Probabilistic Logic Inference**

[**https://arxiv.org/pdf/2005.02074.pdf**](https://arxiv.org/pdf/2005.02074.pdf)

**0. TERMS**

|  |  |
| --- | --- |
|  | 각 instance를 characterise하는 **feature의 집합** |
|  | **Data** |
| POS | **POSITIVE** |
| L | **Labeling function** |
| KB () | Probabilistic **Knowledge Base**  **KB** is in the form of where (는 **clause**) |
| D | **Training data** |
|  | D를 **probabilistic KB로 mapping**시키는 함수 |
|  | Query x에 대해 **와 x가 TRUE를 수반**하는지 확인한다. |
| T, | **T:** **decision tree** / (**feature-value pair**, feature a having value v)  T의 root-to-leaf path의 집합, is in the form of   * 이때 T에 의해 생성되는 KB는 where each 이다.   + **:** 이라는 label이 있는 node 중 **POSITIVE의 비율** |
| F | Data(D)의 **features** (taking values from )  , 즉 **F의 멱집합에서 공집합을 제외**한 것  where   * 각 에 대하여 는 **feature 가 value 를 갖는 샘플**이다. * 이면 **가 에서의 POSITIVE sample**의 비율이라고 할 때, 는 **data에 의해 direct하게 생성되는 KB**에 있다. |
|  | **Complete conjunctions set over**  (각 conjunction이 n개의 서로 다른 propositional variable을 가짐) |
|  | **개의 확률의 집합,**   * 는 **각각의** 에 대하여 **모든 w**에 대해 **을 만족시킬 때 를 만족**시킨다. |
|  | **Clauses** |
|  | **literals** |
|  | 에 대한 **linear program** |
|  | **Unknowns** where |

**1. DEFINITIONS AND EXAMPLES**

|  |
| --- |
| **DEFINITION 1: A Knowledge Base (KB)** |
| Knowledge Base KB 는 clause 와 그에 대한 확률 의 pair이다.   * **각 clause는** **literal의 disjunction(논리합)**이며, 각 literal은 propositional variable(명제 변수, 논리식) 또는 그 negation(부정)이다. |
| **EXAMPLE 1** |
| 2개의 propositional variable 에 대하여 는 2개의 clause를 포함한 simple KB이며 확률은 각각 0.6, 0.8이다. |

|  |
| --- |
| **DEFINITION 2: decision tree T** |
| T가 decision tree일 때 T에 있는 각각의 non-root node는 **feature-value pair 로 라벨링**된다. (feature **a** having value **v**)  나머지는 **0. TERMS**의 **T, , ,**  부분 참고  **<data D로부터 KB drawn from T 를 생성하는 알고리즘>**   |  |  | | --- | --- | |  |  | | Decision Tree로부터 **root-to-leaf path를 통해 clause를 생성**한다.   * Feature **에 대한 path** (각 feature의 값은 중 하나)는 로 해석 | **Decision Tree를 생성**한 다음 **그 tree의 path로부터 clause를 생성**한다. | |
| **EXAMPLE 2** |
| 4개의 string **0000, 1111, 1010, 1100**은 **POSITIVE**이고 4개의 string **0010, 0100, 1110, 1000**은 **NEGATIVE**이면 **4개의 feature (bit)**가 있고 **각 feature는 {0, 1}**에 있는 값을 갖는다. 이때 생성되는 decision tree는 오른쪽 그림과 같다.  예를 들어 이때의 **path** 은 **clause** 을 나타낸다.  또한 이때의 는 오른쪽 그림과 같다. |

|  |
| --- |
| **DEFINITION 3: Given data D with features F taking values from V** |
| **0. TERMS**의 **F,**  부분과 같음 |
| **EXAMPLE 3** |
| 이고 이면 이다. 여기서 이면 이다.  이때 이고 **을 에 추가**한다고 하면 는 **서로 다른 와 c를 반복적으로 추가**하여 얻어진다고 할 수 있다.  이때 오른쪽 그림의 **Algorithm 3**을 이용하여 를 구성할 수 있다. |

|  |
| --- |
| **DEFINITION 4: Given a KB K with clauses C over literals Z** |
| Literal 에 대해, clause가 인 에 대하여 **unknowns** 에 대한 의 **linear program** 는 다음과 같다.   * **Clause로부터 Literal probability**를 전체 conjunction set의 분포 계산 없이 계산 가능 |
| **EXAMPLE 4** |
| (Example 1 cont.) 이고 이면 다음과 같다.   * 에 대한 Truth Assignment는 을 만족시킨다. * 에 대한 Truth Assignment는 를 만족시킨다.   는 일 때 consistent하다. 이때 **, ,** 이다. 이때 는 다음과 같다. |
| **EXAMPLE 5** |
| 이고 다음과 같다고 하면,  이때 가 의 하나의 해가 된다. |

**2. PROPOSITIONS, LEMMAS, COROLLARIES AND EXAMPLES**

|  |
| --- |
| **PROPOSITION 1** |
| 어떤 dataset D에 대해 |
| **PROPOSITION 2** |
| 어떤 dataset D의 **각 clause** 에 대해 **D에 의해 생성되는 decision tree T**가 존재한다.   * 이때 T에는 **p에 의해 생성되는 clause가 c**인 path p가 있다. |
| **LEMMA 1** |
| 가 consistent하면 모든 **w(z)에 대한 solution** 가 존재한다.   * 이때 z는 **의 값을 0으로 최소화**한다. |
| **COROLLARY 1** |
| 어떤 KB 에 대해 **가 0으로 최소화되지 않으면 는 not consistent**하다. |
| **PROPOSITION 3** |
| n개의 propositional variable 이 있는 어떤 KB 에 대해 **에 있는 각각의 는 polynomial time O(n)에 계산** 가능하다. |

|  |
| --- |
| **EXAMPLE 6~7** |
| (Example 2 cont.) query 0101에 대해 **의 제약 조건으로 다음 equation을 추가**한다.  이때 계산된 **의 값이 0.5보다 크지 않으면 negative classification을 표시**한다.   |  |  | | --- | --- | |  |  |     이때 query 0101에 대해 **Algorithm 5**에 의해 도출된 relevantKB는 다음과 같다. |

**3. EXPLANATION AND KNOWLEDGE INCORPORATION**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Explanation을 도출하는 알고리즘**은 오른쪽 그림과 같다.  (Example 6 cont.) 가장 결정적인 feature를 계산하려면 **k=1**로 놓는다. 이때 S는 다음과 같이 4개의 feature-value pair를 갖는다.  이때 **는 에 의해 계산되는 w(POS)**이며, 여기서 이 도출된다. 따라서 classification에 대한 computed explanation은 이다.  따라서 설명은 **“0 - - - is responsible for 0101 being negative”** 라고 읽을 수 있다.  **[Example 6~7 풀이] (query=0101)**   |  |  | | --- | --- | | Decision tree |  | |  |  | | POSITIVE | NEGATIVE | | 0000, 1111, 1010, 1100 | 0010, 0100, 1110, 1000 |   1. 먼저 **를 Algorithm 3을 이용하여 구하면** 다음과 같다. (중간의 \_ 기호는 생략)   |  |  |  |  | | --- | --- | --- | --- | | 0000 |  | 0010 |  | | 1111 |  | 0100 |  | | 1010 |  | 1110 |  | | 1100 |  | 1000 |  |   여기서 로 축약하여 표현하면  0000:  1010:  등이 된다.  따라서 에 대해 이므로 에 를 삽입한다. 마찬가지로 해서 를 구하면 다음과 같다. |

**3. EXPLANATION AND KNOWLEDGE INCORPORATION**

|  |
| --- |
| 이와 같은 방법으로 **최종적으로 를 도출**하면 다음과 같다.  2. **Relevant Knowledge Base (Relevant KB)** 를 **Algorithm 5**를 이용하여 **로부터 도출**하면 다음과 같다. **(query=0101)**    S의 각 **key k**와 의 각 **clause c**에 대해 **c가 k를 포함하면 추가**한다. 따라서 Relevant KB는 오른쪽과 같다.  3. **Algorithm 4, Algorithm 6**을 이용하여 **Explanation을 도출**한다. (k=1)  각 feature 와 각 possible value 에 대하여 **query에서** **feature a가 value v를 가지면** , 그렇지 않으면 이므로 에는 다음이 추가된다.  이 **를 이용하여 w(POS)를 계산**하고, 그 결과는 다음과 같다.  따라서 QueryKB 함수의 결과값이 0.5보다 크지 않으므로 **0.333에 해당하는 이 도출**된다. |