Comentarios sobre el artículo “Inductive Logic Programming”

Mikel Escobar de Carlos
Luis Alexander Zelada Medina
Karen Oliveros Félez

Diciembre 2021

1. Introducción: procesos inductivos y deductivos. Motivación y aplicaciones de la programación lógica inductiva

En este apartado vamos a describir los procesos inductivos y deductivos y vamos a mostrar algunos ejemplos clarificatorios.

En un **razonamiento deductivo** las premisas apoyan de forma absoluta a la conclusión, es decir si las premisas son verdaderas la conclusión también lo es, este razonamiento no aporta nueva información.

En un **razonamiento inductivo** las premisas apoyan de forma parcial a la conclusión, es decir si las premisas son verdaderas la conclusión no necesariamente tiene que serlo, sin embargo siempre nos aporta nueva información. El proceso de inducción consta de dos elementos, el primero es la abducción o formación de hipótesis a partir de casos particulares y el segundo elemento es la justificación o el grado de certeza de la hipótesis generada.

Veamos ejemplos de procesos deductivos e inductivos:

Ejemplo 1.0.1 *Razonamiento deductivo. Sean A, B formulas lógicas,*

$$(A \to B) \equiv (\neg B \to \neg A)$$

Demostración: escribimos las tablas de verdad de cada una de las implicaciones y observamos que ambas fórmulas son equivalentes.

Ejemplo 1.0.2 *Razonamiento inductivo. Tenemos que hacer la suma de todos los números positivos hasta n, $S = 1 + 2 + 3 + ... + n = \sum_{i=1}^{n} i$. Tenemos que:*

$$\begin{array}{l} \quad 1 + 2 + 3 + ... + (n-1) + n \\ + \\ \quad n + (n-1) + ... + 3 + 2 + 1 \\ \hline \quad (n+1) + (n+1) + ... + (n+1) = 2S \end{array}$$

Y por tanto,

$$S = \frac{n(n+1)}{2}$$

Ejemplo 1.0.3 *Razonamiento deductivo. Tenemos las siguientes afirmaciones.*

- El hijo de mi abuelo es mi tío
- Raúl es hijo de mi abuelo
Conclusión: ***Raúl es mi tío***

En muchas ocasiones es muy difícil establecer las leyes o teorías generales para seguir un proceso deductivo hacia casos mas particulares.

La programación lógica inductiva utiliza casos muy específicos (conocimientos previos) para intentar generalizar la realidad (la búsqueda y elección de hipótesis). Por lo tanto es un método que nos ayuda a establecer las posibles hipótesis y elige la hipótesis generada con más certeza, en vez de tener que establecer las hipótesis de antemano e intentar demostrarla después. Uno de los motivos para usar la lógica inductiva es la limitación humana que tenemos de generar hipótesis en las que intervienen grandes cantidades de variables.

Ejemplo 1.0.4 *Ejemplo de razonamiento inductivo en el día a día. Los niños se valen de procesos inductivos para entender el mundo que les rodea, por ejemplo a la hora de identificar platos en una comida:*

- Las naranjas la comemos en el postre.
- Las fresas las comemos en el postre.
- Las peras las comemos en el postre.
- Las naranjas, las fresas y las peras son fruta.
La hipotesis sería que la ***fruta se come en el postre*** *.*

En nuestro día a día los adultos también nos servimos de procesos inductivos sencillos para entender como funciona el mundo que nos rodea. Por otro lado existen aplicaciones de la programación lógica inductiva [3] en las ciencias muy interesantes, como por ejemplo se pueden deducir reglas de inferencia sobre una base de datos de conocimiento [4], el aprendizaje de distintas reglas para la detección temprana de un diagnóstico reumático, o deducir un set de reglas para el diagnóstico de fallos de energía en los satélites [3].

2. Relación entre la programación lógica inductiva y la probabilidad

Partimos sabiendo que, al contrario de lo que sucede con la inferencia deductiva, no podemos asegurar que las conclusiones que extraemos de la inferencia inductiva sean la explicación correcta de aquello que sabemos. Por este motivo cada declaración lógica que hemos inferido inductivamente debemos acompañarla con un grado de confianza, o lo que es lo mismo, un valor de probabilidad.

En concreto, el Teorema de Bayes nos permite calcular la probabilidad condicional de un evento P dado otro evento Q en términos de la probabilidad condicional de Q dado P y las probabilidades marginales de P y Q.

Teorema 2.0.1 *Teorema de Bayes. Sean P,Q dos sucesos del espacio muestral, $P, Q \subseteq \Omega$, con una probabilidad p asociada a Ω, se tiene:*

$$p(P|Q) = \frac{p(Q|P) \cdot p(P)}{p(Q)}$$

Este resultado es de gran ayuda cuando tratamos de aportar este valor de confianza, o justificación, de una hipótesis. Además, es ampliamente utilizado en cualquier problema de probabilidad o estadística. Para el tema que nos concierne, si tomamos que nuestra clausula definida de teoría $\mathcal{T}$ se extrae de la unión de nuestro conocimiento previo $\mathcal{B}$ y nuestra hipótesis $\mathcal{H}$:

$$\mathcal{T} = \mathcal{B} \wedge \mathcal{H}$$

Podemos asumir que nuestro grado de confianza en dicha teoría puede representarse como probabilidad subjetiva de $\mathcal{T}$ dadas nuestras evidencias $\mathbf{E}$. Así, empleando el Teorema de Bayes tal que:

$$p(\mathcal{T}|\mathbf{E}) = \frac{p(\mathcal{T}) \cdot p(\mathbf{E}|\mathcal{T})}{p(\mathbf{E})}$$

Asumiremos que nuestra evidencia es correcta, es decir que $p(\mathbf{E}) = 1$. Podemos reinterpretar estos términos de probabilidad en términos de teoría de la información de Shannon donde el contenido en información de un programa lógico P es

$$I(P) = -log_2 p(P)$$

Con esta definición, y volviendo a nuestra expresión del Teorema de Bayes para el grado de confianza, si $\mathcal{T} \models \mathbf{E}$ entonces, aplicando propiedades de logaritmos, llegamos a

$$I(\mathcal{T}|\mathbf{E}) = I(\mathcal{T}) + I(\mathbf{E}|\mathcal{T}) - I(\mathbf{E})$$

Y a raíz de esta relación podemos llegar al resultado de que elegir la teoría con la mínima descripción es equivalente a elegir la teoría con la mayor probabilidad posterior. Es decir, buscamos minimizar $I(\mathbf{E}|\mathcal{T})$ sabiendo que esto equivale a maximizar la probabilidad $p(\mathcal{T}|\mathbf{E})$ Esto también se conoce como principio de mínima descripción de Rissanen.

Matemáticamente es fácil de ver esta equivalencia ya que la función $\log_2$ que hemos aplicado para poner la probabilidad en términos de información de Shannon es creciente en todo su dominio y se extrae que:

$$\min_{\mathcal{T}\in\Sigma} I(\mathcal{T}|\mathbf{E}) = -\log_2 \max_{\mathcal{T}\in\Sigma} p(\mathcal{T}|\mathbf{E})$$

Donde Σ es el conjuto de teorías plausibles. Vemos de nuevo que la teoría que minimiza la información de $I(\mathcal{T}|\mathbf{E})$ equivale a la teoria de que maximiza $-\log_2(p(\mathcal{T}|\mathbf{E}))$.

Es decir, hay una equivalencia entre la teoría de mínima descripción y la teoría que tiene una mayor probabilidad posterior, que es lo que estamos buscando. Así podremos elegir la teoría $\mathcal{T}$ que tiene la máxima probabilidad dada la evidencia $\mathbf{E}$. **De manera más coloquial podemos simplemente decir que la teoría con una descripción más corta es la que tiene más probabilidades de ser la acertada, o que la manera más sencilla de describir nuestros datos constituye el mejor modelo.**

3. Conclusiones y relación entre la programación lógica inductiva y la inteligencia artificial

Del texto principal [3] y de otros textos [2] sacamos la conclusión de que la programación lógica inductiva utiliza técnicas de Machine Learning y programación lógica; del Machine Learning inductivo la programación lógica inductiva (ILP) hereda el objetivo de desarrollar herramientas y técnicas para inducir hipótesis (en ML lo llamamos modelo) a partir de observaciones y sintetizar nuevos conocimientos de la experiencia. Además, hace uso del principio MDL que es clásicamente usado en Machine Learning.

La programación lógica inductiva también ha tenido problemas que han podido ser resueltos empleando soluciones desarrolladas en Machine Learning. Por otro lado las redes neuronales pueden ser vistas como una forma de inferencia inductiva, aunque los algoritmos ILP generan outputs que pueden ser más comprensibles por las personas.

Históricamente, hemos podido comprobar cómo han avanzado de la mano los sistemas de lógica inductiva con los de aprendizaje automático. Uno de los puntos donde hemos observado más convergencia es el de los sistemas de clasificación, como es el caso de los árboles de decisión. En modelos como el expuesto en [1], de 1995, donde vemos que la estrategia de aprendizaje de reglas para clasificación comienza por crear una hipótesis a partir de un árbol de decisión muy complejo que overfittea los datos y a partir de ahí se va simplificando el árbol para generalizar dicha hipótesis. Esto es, en esencia, el principio de mínima descripción del que hablábamos en el apartado anterior. En [5] se habla de dos corrientes principales dentro del Machine Learning: por una parte, aquellos sistemas que tratan de sintetizar reglas capaces de clasificar casos no vistos en un número finito de categorías y por otra, aquellos que intentan inducir funciones que modelan los datos. El ejemplo anterior caería dentro de esta primera corriente mientras que modelos de redes neuronales como las Restricted Boltzmann Machines se usan en numerosas aplicaciones para tratar de inducir la distribución de probabilidad subyacente a los datos de entrada, es decir, estaríamos induciendo funciones de probabilidad.

Referencias

[1] William W Cohen. Fast effective rule induction. In *Machine learning proceedings 1995*, pages 115–123. Elsevier, 1995.

[2] Andrew Cropper and Sebastijan Dumančić. Inductive logic programming at 30: a new introduction, 2020.

[3] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19:629–679, 1994.

[4] Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. Rnnlogic: Learning logic rules for reasoning on knowledge graphs, 2021.

[5] José Ramón Quevedo and Antonio Bahamonde. Aprendizaje de funciones usando inducción sobre clasificaciones discretas. In *Proceedings CAEPIA*, volume 99, pages 64–71, 1999.