

Análisis de artículos científicos

ILP, BLP, PLIB y su rol en el Aprendizaje Automático

Curso: 2025/2026 [Convocatoria Abril]

Grupo de Trabajo:

Michael Laudrup Luis González

Pol Jorba Lloses

Profesor/a:

Amílcar J. Pérez A.

Fecha de Entrega: 11/06/2025

Institución Educativa: Universidad Internacional de Valencia

Índice

1. Proceso deductivo e inductivo	3
2. Motivación y aplicaciones de la programación lógica inductiva (PLI)	6
3. Interacción con la probabilidad y la inferencia bayesiana.....	8
4. Conexión con la inteligencia artificial y las redes neuronales	10
5.1 Conexión entre Machine Learning (ML), redes neuronales y Programación lógica inductiva (ILP).....	10
5.2 Programación lógica Bayesiana (BLP).	12
5.3 Programación lógica Inductiva Bayesiana	14
1. Conclusiones.....	15
5. Referencias	16

1. Proceso deductivo e inductivo

Explíquese cuáles son las diferencias clave entre un proceso deductivo e inductivo. Indique algunos ejemplos de cada uno de ellos, más allá de los que ya se presentan en el artículo.

Un proceso deductivo parte de conocimientos preestablecidos para llegar a una conclusión específica, tomando reglas lógicas y premisas ya existentes. Esto implica que un proceso deductivo es válido solo si las premisas y reglas lógicas que se han tomado son ciertas. Por lo tanto, son estas premisas las que garantizan la verdad de la conclusión, y no ninguna nueva evidencia empírica.

Los procesos deductivos son un tipo de inferencia muy utilizada históricamente y en la actualidad en campos como las matemáticas y la filosofía.

Por otro lado, un proceso inductivo parte de observaciones concretas y particulares para definir una regla o conclusión. Por lo tanto, no puede garantizar la verdad de las conclusiones obtenidas, ya que solo se basa en observaciones y no en reglas o premisas existentes.

Es por este motivo que un proceso inductivo requiere ser justificado para confirmar si la hipótesis obtenida es correcta, y una de las formas de justificar la hipótesis es mediante soporte probabilístico bayesiano, donde la hipótesis se justifica dándole una probabilidad previa (prior) y actualizándola según las evidencias observadas (posterior). De manera que se puede calcular cuán probable es que la hipótesis sea cierta.

Otra de las formas de justificar una hipótesis es mediante restricciones sobre el espacio de hipótesis, definiendo límites sobre las hipótesis consideradas válidas o aceptadas, evitando generar hipótesis demasiado complejas o que se ajusten solo a la muestra tomada.

Es importante la justificación, ya que sin ella podemos llegar a tener hipótesis accidentales (válidas solo para la muestra observada), también podemos sobreajustar (definir una regla demasiado específica que no generalice y se centre en los casos específicos), o no podremos garantizar que la hipótesis sea válida para nuevas muestras o situaciones reales.

Por lo tanto, las principales diferencias son que un proceso deductivo no necesita una validación adicional y, si las premisas son ciertas, podemos garantizar que las conclusiones también lo son. Pero en los procesos inductivos necesitamos una validación adicional, que puede ser a partir de un soporte probabilístico bayesiano o mediante restricciones sobre el espacio de hipótesis. Otro aspecto diferencial es que en los procesos inductivos no se puede garantizar la certeza de la conclusión, sino que tendrá una probabilidad.

Un ejemplo de proceso deductivo puede ser la asistencia obligatoria a las clases de una asignatura, tomando las siguientes premisas:

Premisa 1: Todos los estudiantes deben asistir como mínimo al 90% de las sesiones para aprobar.

Premisa 2: María ha asistido al 65% de las sesiones.

Conclusión: María va a suspender la asignatura.

Y un ejemplo de proceso inductivo puede ser el siguiente:

Los alumnos que participan más en clase tienen mejores notas → Participar en clase ayuda a tener buenos resultados.

Otro ejemplo podría ser la afirmación de Eduard Punset durante una entrevista en el programa “El convidat” de TV3, donde Punset afirma:

«Yo, probablemente, no me moriré nunca. No está demostrado que yo me tenga que morir.»

Donde el entrevistador le respondió que hasta el día de hoy, se ha muerto todo el mundo.

En esta entrevista se puede ver como ejemplo que el entrevistador analiza la muerte del ser humano mediante un proceso deductivo de la siguiente manera:

Premisa 1: Todos los humanos son mortales.

Premisa 2: Eduard Punset es humano.

Conclusión: Eduard Punset es mortal y, por lo tanto, morirá.

Sin embargo, el entrevistado cuestiona la validez de una inferencia inductiva, rechazando la generalización del razonamiento popular: “como todos los humanos hasta ahora han muerto, todos los humanos morirán”.

Desde este punto de vista, Punset sugiere que, aunque toda la muestra conocida de la humanidad haya muerto, sigue existiendo una probabilidad, por pequeña que sea, de que él no muera. Y esto es así porque, por definición, un proceso inductivo nunca puede garantizar una conclusión con certeza absoluta, sino únicamente con cierto grado de probabilidad.

2. Motivación y aplicaciones de la programación lógica inductiva (PLI)

Una vez establecidas las diferencias, indíquese cuál es la motivación para trabajar con programación lógica inductiva, exponiendo algunas aplicaciones y/o situaciones en las que nos valemos de un proceso inductivo.

La programación lógica inductiva (PLI) es una técnica que combina la lógica de primer orden para representar conocimiento estructurado y relacional con los métodos inductivos para aprender reglas generales a partir de ejemplos. El objetivo de este tipo de programación es crear programas lógicos que expliquen o generalicen los ejemplos positivos o negativos dados utilizando también conocimiento de fondo.

Este tipo de proceso inductivo no garantiza una certeza absoluta debido a que se basa en datos empíricos, pero permite obtener nuevo conocimiento donde no existen reglas previas.

El principal motivo para utilizar PLI es que genera reglas lógicas explícitas, que son comprensibles y se pueden verificar y reutilizar estas reglas. En cambio, en modelos estadísticos o de aprendizaje profundo, las reglas obtenidas son opacas y por lo tanto difíciles de verificar.

Otro motivo es que la programación lógica inductiva permite utilizar relaciones entre elementos complejas. En algunos entornos, los datos no vienen en forma de tabla o como variables independientes, sino que están relacionados entre sí, lo que hace a la PLI una buena opción para manejar este tipo de datos.

También se destaca por poder utilizar conocimientos previos (background knowledge) para aprender de forma más eficiente utilizando las reglas previamente conocidas, cosa que no es posible con muchos sistemas estadísticos.

Los procesos inductivos son muy utilizados en entornos donde es necesario entender el funcionamiento interno de los algoritmos, como puede ser en

decisiones médicas o legales. La PLI aporta unos modelos que son interpretables y por lo tanto se puede verificar el resultado.

En el entorno médico se usa para diagnosticar a partir de historiales o informes clínicos. La PLI es buena en este tipo de casos ya que mezcla conocimientos previos (background knowledge) con los datos empíricos, y además permite entender el proceso lógico que utiliza para llegar a una conclusión.

La PLI también es utilizada en procesamiento de lenguaje natural para aprender reglas lógicas que ayuden a entender o interpretar el lenguaje humano de forma automática. De la misma forma que en el entorno médico, se incorpora conocimiento previo con reglas lingüísticas ya existentes y posteriormente aprende a partir de ejemplos obteniendo nuevas reglas comprensibles por el ser humano, a diferencia de las redes neuronales.

3. Interacción con la probabilidad y la inferencia bayesiana

Expóngase la interacción entre esta disciplina y la probabilidad, así como el rol que juega el teorema de Bayes y, en general, la inferencia bayesiana.

Una de las limitaciones de la ILP es lidiar con la incertidumbre, siempre está buscando la hipótesis perfecta que tenga la verdad absoluta, sin embargo, esto está muy alejado de problemas del mundo real, por ello, se rompe esta limitación introduciendo componentes de probabilidad en ILP.

Al añadir componentes de la probabilidad se puede lidiar con más eficacia con la incertidumbre haciendo que nuestro nuevo modelo ILP asigne un grado de probabilidad a las hipótesis que va estudiando a partir de una base de conocimiento compuesta por (hechos, ejemplos y contraejemplos).

Hay una pequeña redundancia. Podrías simplificarlo a: "Una de las herramientas clave para este cálculo es el Teorema de Bayes, que establece:

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

Tal que:

- **H:** es la hipótesis.
- **E** es la evidencia (los ejemplos observados).
- **P(H):** es la probabilidad previa de la hipótesis (que tan creíble era antes de ver los ejemplos).
- **P(E | H):** es que tan bien la hipótesis explica la evidencia (conocido como verosimilitud).
- **P(H | E):** cuánto creo que **H** después de ver **E**.

Cada hipótesis genera un conjunto de modelos posibles, tras esto se calcula que proporción de esos modelos coinciden con la evidencia, esa misma proporción es la probabilidad de que la hipótesis sea correcta. Otro aspecto interesante es que se tiene en cuenta la probabilidad previa a haber visto los ejemplos y contraejemplos

Este último aspecto es relevante porque tiene mucha relación con la inferencia bayesiana, la cual da ese grado de flexibilidad de que la experiencia (ver ejemplos y contraejemplos) moldee y cambie la probabilidad llegando a conclusiones más refinadas.

Dos ramificaciones de la lógica asociadas a la probabilidad son BLP (Programación lógica Bayesiana) y PLIB (Programación Lógica inductiva Bayesiana), veamos más al detalle cada una de ellas:

- En el caso de BLP se fusiona las redes bayesianas con la lógica de predicados, pudiendo estudiar la relación entre diferentes predicados no solo eso, sino en que grado o con que fuerza unas influyen en otras, además posee herramientas como la representación gráfica que permite hacer interpretable y entendible por el lenguaje humano el conocimiento e hipótesis aprendido.
- En el caso de PLIB uno de los aspectos más interesantes es que no solo obtenemos la hipótesis más prometedora, sino que podremos apreciar que tan probables son otras hipótesis.

En conclusión, gracias a la probabilidad, el teorema de Bayes y la inferencia Bayesiana, podemos dotar de mucha más flexibilidad a la programación lógica inductiva y que está pueda lidiar con la naturaleza intrínseca de los problemas reales, no buscando doctrinas o verdades absolutas, sino flexibilizándose al dinamismo natural de nuestro mundo.

4. Conexión con la inteligencia artificial y las redes neuronales

Detállese la conexión que existe entre estas ramas de la lógica y estadística con la inteligencia artificial, en particular el aprendizaje automático y las redes neuronales.

Para dar respuesta a esta pregunta, subdividiremos sus respuestas en 4 apartados, los 3 primeros asociados a cada una de las ramas de la lógica que hemos estado estudiando en este trabajo académico y el ultimo reflejará las conclusiones a las que se han llegado.

4.1 Conexión entre Machine Learning (ML), redes neuronales y Programación lógica inductiva (ILP)

La primera conexión que se aprecia entre la programación lógica inductiva, la cual, busca aprender reglas lógicas o hipótesis a partir del conocimiento previo y las redes neuronales es que ambas aprenden de conocimiento previo y ejemplos. Esto tiene mucho que ver con una de las características esenciales del aprendizaje automático que es aprender patrones o reglas a partir de los datos (conocimiento previo), que en el caso más concreto de ILP es aprender hipótesis o formulaciones lógicas. Sin embargo, hay que subrayar que en el caso de ILP además de tener ejemplos y contraejemplos, también se le da hechos, es decir, se le definen patrones o predicados, que la hacen tener un conocimiento base explícito, sin embargo, en redes neuronales este conocimiento está implícito en los datos y no se expresa formalmente en forma de un conjunto de patrones previamente aprendidos, sino que ella tiene que aprender todos los patrones de los datos desde cero.

Adicionalmente, uno de los aspectos más importantes que tienen en contraste, es que la ILP genera reglas lógicas o hipótesis que se pueden entender e interpretar, sin embargo, las redes neuronales actúan más como una caja negra donde cuesta entender como aprende o que patrones le llevan a sacar ciertas conclusiones. Es decir, mientras que la ILP no solo se centra en dar resultados sino también en

indicar ¿Cómo genera esos resultados? Las redes neuronales no nos explican como llegan a ese resultado en sí, sino que nos lo arrojan directamente.

Otra conexión es que en ILP también se aprenden los conceptos de generalización y especialización muy similares a conceptos de generalización y especialización machine Learning, en el caso concreto de Deep Learning si se sobrentrena la red neuronal se llega a que el modelo que infiere conclusiones se apega demasiado a los datos y se especializa mucho en lo que ha aprendido, sin llegar a conclusiones más generales, por el contrario, si la red neuronal no se entrena lo suficiente con los datos, perderá sutilezas o especializaciones y será más generalista en sus conclusiones.

Tanto en ILP como en Machine Learning o Redes neuronales se busca un punto intermedio entre generalización y especialización.

Además, tanto en ILP como en Redes neuronales se aplican restricciones que guían el aprendizaje (bias), aunque en el caso de redes neuronales no hay que confundirlo con el parámetro libre “b” de las regresiones lineales que hay en cada neurona, en este contexto nos referimos más a que tanto ILP como redes neuronales establecen restricciones para sesgar o dirigir el aprendizaje de una manera determinada.

Concretamente en el caso de ILP se establece sesgos en forma sintáctica (limitamos solo usar ciertos predicados, limitar el número de variables que puede tener un predicado, etc.) o semántica (que controla el significado o comportamiento de las reglas).

Mientras que en redes neuronales el sesgo se establece suponiendo que las funciones son lineales dentro de las neuronas, establecer número de capas, tipo de funciones de activación, etc. En conclusión, sesgamos el aprendizaje hacia una determinada dirección estableciendo la arquitectura de la red neuronal.

Además, otra conexión que tienen ILP y las redes neuronales es que ambas se pueden entrenar inyectando uno a uno los ejemplos y por cada ejemplo que se introduce se actualizan las hipótesis progresivamente en ILP, o en el caso de las redes neuronales, los pesos y el sesgo se actualizan, en el caso de ILP a esto se le

denomina incremental mientras que en machine learning se le denomina proceso estocástico o con todos los ejemplos. Lo mismo sucede en el caso contrario, entrenarlo con todo el conjunto de datos; mientras que en ILP se denomina “no incremental” en el caso de las redes neuronales se denomina “Batch training”.

Adicionalmente, aunque no es exactamente lo mismo, el concepto de que un ILP puede ser interactivo o no interactivo, tiene aspectos similares al aprendizaje por refuerzo de Machine Learning, aunque en el caso de ILP es más un humano experto el que guía el aprendizaje y responde a preguntas y en el caso de machine Learning es el propio entorno el que da recompensas o no si que podemos quedarnos con el concepto de que ambos, con estas técnicas demuestran ser dinámicos, adaptables y flexible antes nuevos escenarios, de los cuales, aprenden.

En resumen, ILP y redes neuronales pertenecen a enfoques distintos, pero comparten principios fundamentales como aprender de datos, aplicar restricciones y ajustarse a entornos cambiantes, aunque lo hagan con herramientas conceptualmente distintas.

4.2 Programación lógica Bayesiana (BLP).

En el caso de la programación lógica Bayesiana se fusionan las ventajas tanto de lógica proposicional como de las redes bayesianas, es decir, se puede estructurar la información en forma de predicados lógicos aumentando la capacidad de expresividad y a cada predicado o hecho lógico se le asocia una probabilidad de que sea cierta, pudiendo lidiar con la incertidumbre y dando más flexibilidad al sistema.

Nuevamente, como se explicó en el apartado anterior en este caso obtenemos hipótesis y una probabilidad asociada, podemos interpretar los datos y entender cómo se infiere o se llega a conclusiones, sin embargo, en redes neuronales no podemos interpretar como se han generado las conclusiones o que patrones o reglas se han aprendido.

Además, el concepto de probabilidad intrínseco de estadística se trata diferente en BLP que en redes neuronales, en el caso de BLP podemos definir de manera

explícita hechos que reflejen probabilidades y tenemos más flexibilidad para controlar y definir que probabilidades hay de ciertos hecho, sin embargo, en redes neuronales, no tenemos esa versatilidad, por ejemplo, si una red neuronal está especializada en decir en una foto hay un gato o no hay un gato, probablemente la red neuronal nos arrojará un valor numérico 0.97 que podría reflejar que hay un 97% de probabilidad de que sea un gato, es decir, a pesar de que si tenemos esa capacidad de trabajar con la incertidumbre como en BLP, no podemos controlarla, manipularla ni sesgarla hacia una dirección determinada de manera directa sino de manera indirecta influyendo en la arquitectura de la red neuronal, es más, ni si quiera podemos entender como ha llegado a ese valor 0,97. En contraste, en BLP sí que podemos explicar cómo se ha llegado a una conclusión.

Otra facilidad que nos ofrece la BLP es que podemos ver de manera gráfica las relaciones y sus probabilidades, permitiéndonos comprender de manera más intuitiva los que se está aprendiendo, es decir, la interpretabilidad de las hipótesis, conclusiones y como se interrelacionan junto con su incertidumbre es más sencilla y cercana al lenguaje humano, sin embargo, en redes neuronales, esto está mucho más limitado.

A pesar de todo esto, un símil entre BLP y las redes neuronales, es que ambas técnicas tienen la capacidad de tener en cuenta que un hecho puede depender de varios factores, por ejemplo, en una red neuronal el precio de una casa depende del número de habitaciones, su posición geográfica, servicios que tiene, conexiones con red de transporte, etc. En una red neuronal muy simple, cada neurona puede estar especializada en ver cómo influye cada una de las características en el precio de la casa, es decir, tenemos en cuenta que un hecho depende de varios factores, esto también se puede reflejar en el BLP con las funciones de agregación que permiten combinar o resumir información de un conjunto de hechos o entidades para influir en una conclusión probabilística.

La diferencia clave es que en BLP el propio diseñador del sistema tiene que establecer las funciones de agregación y en redes neuronales, es la propia red la

que aprende automáticamente las relaciones entre diferentes factores y su grado de influencia en el resultado final.

En conclusión, la distinción fundamental entre la Programación Lógica Bayesiana y las redes neuronales radica en el compromiso entre la interpretabilidad y la automatización. Mientras que los BLP ofrecen modelos transparentes y auditables ("cajas blancas") donde la lógica es explícita, las redes neuronales son potentes "cajas negras" que aprenden patrones complejos de forma autónoma. La elección entre ambos dependerá de si la prioridad es entender el "porqué" de una decisión o maximizar la capacidad predictiva del modelo a partir de los datos.

4.3 Programación lógica Inductiva Bayesiana

Podemos entender la programación lógica inductiva Bayesiana como una evolución de ILP, en la que podemos asignar probabilidades a ciertas reglas, afirmando que unas son más probables que otra. Como resultado se presenta una hipótesis que es la que mayor probabilidad ha conseguido y además calcula una distribución de probabilidad sobre todas las hipótesis posibles.

En este caso concreto heredamos todas las conexiones que tiene la programación lógica inductiva con el machine Learning y las redes neuronales explicadas en el apartado 4.a., adicionalmente, otro aspecto que conecta la BILP con las redes neuronales es que estas dan prioridad a las hipótesis más simples sobre las más complejas, busca que se generalice sin sobre ajustarse a los datos, algo muy parecido sucede en redes neuronales donde a través de técnicas de regularización se busca un equilibrio entre especialización y generalización, y también se busca simplificar el modelo.

Finalmente, al añadir probabilidades en ILP derivando en BILP podemos apreciar que el aprendizaje es más robusto y menos sensible a datos puntuales que generan ruido, algo intrínseco de las redes neuronales.

4. 4 Conclusiones

Podemos concluir que las técnicas de aprendizaje asociadas a la lógica (ILP, BLP y BILP) pueden considerarse una forma de machine Learning, pero más orientada al razonamiento simbólico mediante lógica de predicados y otros componentes estructurados.

El aspecto más relevante que diferencia estos enfoques lógicos del aprendizaje automático basado en redes neuronales es la interpretabilidad: en ILP, BLP y BILP es posible comprender, analizar y justificar las hipótesis aprendidas, e incluso conocer cómo ciertos hechos o reglas influyen sobre otros. Por el contrario, en las redes neuronales el proceso de aprendizaje ocurre de forma distribuida en los pesos y capas, lo que dificulta entender cómo se ha llegado a una conclusión o qué conocimiento se ha adquirido. Es lo que se conoce como una arquitectura de tipo “caja negra”.

Teniendo en consideración todo esto, podemos apreciar que, dependiendo de nuestras necesidades y del dominio específico, será más conveniente una estrategia de aprendizaje u otra. Uno de los muchos criterios que puede haber es que, en el caso de que queramos deducir verdades, hechos o hipótesis de un área determinada, es decir, que nos interese más el “cómo” que el “qué”, nos decantaremos por las ramas de la lógica. Por el contrario, si estamos más centrados en los resultados y en el “qué”, sin darle especial relevancia a cómo se ha llegado a ellos, una estrategia más adecuada es la del aprendizaje profundo (Deep Learning).

5. Referencias

- Muggleton, S. (1994). *Bayesian Inductive Logic Programming*.
- Kersting, K., & De Raedt, L. (s.f.). *Bayesian Logic Programming: Theory and Tool*.
- Schaeffer, H., & McCalla, S. G. (2020). *Extending the Step-Size Restriction for Gradient Descent to Avoid Strict Saddle Points*.
- Muggleton, S., & De Raedt, L. (1994). *Inductive Logic Programming: Theory and Methods*.